Innovative Technology for Computer Professionals

Computer

April 2008

Data-Intensive Computing

http://www.computer.org

IEEE computer society

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society. **IEEE Headquarters**, Three Park Avenue, 17th Floor, New York, NY 10016-5997; **IEEE Computer Society Publications Office**, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; voice +1 714 821 8380; fax +1 714 821 4010; **IEEE Computer Society Headquarters**,1730 Massachusetts Ave. NW, Washington, DC 20036-1903. IEEE Computer Society membership includes $19 for a subscription to *Computer* magazine. Nonmember subscription rate available upon request. Single-copy prices: members $20.00; nonmembers $99.00.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in USA.

Innovative Technology for Computer Professionals
Computer

April 2008, Volume 41, Number 4

IEEE Computer Society: http://computer.org
Computer: http://computer.org/computer
computer@computer.org
IEEE Computer Society Publications Office: +1 714 821 8380

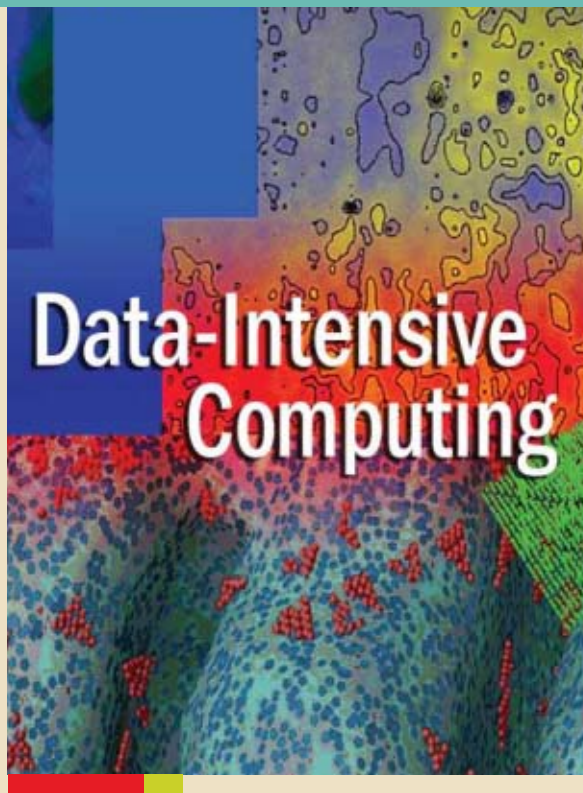
Data-Intensive Computing

Cover design and artwork by Dirk Hagner

ABOUT THIS ISSUE

The breakthrough technologies needed to address many of the critical problems in data-intensive computing will come from collaborative efforts involving several disciplines, including computer science, engineering, mathematics, and the sciences. This special issue on data-intensive computing presents five cover features that address some of these challenges.

Flagship Publication of the IEEE Computer Society



IEEE

IEEE computer society

ARTICLE SUMMARIES

Using String Matching for Deep Packet Inspection
pp. 23-28

Po-Ching Lin, Ying-Dar Lin, Tsern-Huei Lee, and Yuan-Cheng Lai

String matching has recently proven useful for deep packet inspection (DPI) to detect intrusions, scan for viruses, and filter Internet content. However, the algorithm must still overcome some hurdles, including becoming efficient at multigigabit processing speeds and scaling to handle large volumes of signatures.

Before 2001, researchers in packet processing were most interested in *longest-prefix matching* in the routing table on Internet routers and *multifield packet classification* in the packet header for firewalls and quality-of-service applications. However, DPI for various signatures is now of greater interest.

Quantitative Retrieval of Geophysical Parameters Using Satellite Data
pp. 33-40

Yong Xue, Wei Wan, Yingjie Li, Jie Guang, Linyian Bai, Ying Wang, and Jianwen Ai

A reliable atmospheric remote-sensing monitor uses physical or statistical models for which the parameters must be retrieved quantitatively. However, such retrieval is data-intensive. High-resolution, wide-range, and long-duration observations produce several terabytes of data each day.

Processing such massive volumes of data into scientific aerosol products involves addressing several computational problems. Processing Level 1B data for Level 2 aerosol products for a single day requires 13 Gbytes total to achieve full coverage of China's main land surface.

Accelerating Real-Time String Searching with Multicore Processors
pp. 42-50

Oreste Villa, Daniele Paolo Scarpazza, and Fabrizio Petrini

String-searching algorithms are at the core of search engines, intrusion detection systems, virus scanners, spam filters, and content-monitoring systems. Fast string-searching implementations traditionally have been based on specialized hardware like FPGAs and application-specific instruction-set processors, but the advent of multicore architectures such as IBM's Cell Broadband Engine is adding new players to the game.

The authors developed a parallelization strategy for the Aho-Corasick algorithm that achieves performance comparable to other results in the literature with small data dictionaries but exploits the Cell's sophisticated memory subsystem to effectively handle large dictionaries.

Analysis and Semantic Querying in Large Biomedical Image Datasets
pp. 52-59

Vijay S. Kumar, Sivaramakrishnan Narayanan, Tahsin Kurc, Jun Kong, Metin N. Gurcan, and Joel Saltz

Digital microscopy opens new opportunities to study a disease's tissue characteristics at the cellular level. Traditionally, human experts visually examine tissue and classify images, then make a diagnosis. This process is time-consuming and the sheer size of image datasets makes gleaning information from digital microscopy slides data-intensive.

The authors' work addresses problems in two areas: the processing of large digitized slides for analysis and the semantic query of annotated images and image regions in a large image dataset.

Hardware Technologies for High-Performance Data-Intensive Computing
pp. 60-68

Maya Gokhale, Jonathan Cohen, Andy Yoo, W. Marcus Miller, Arpith Jacob, Craig Ulmer, and Roger Pearce

Data-intensive problems challenge conventional computing architectures with demanding CPU, memory, and I/O requirements. Using benchmarks that draw on three data types—scientific imagery, unstructured text, and semantic graphs representing networks of relationships—the authors demonstrate that emerging hardware technologies to augment traditional microprocessor-based computing systems can deliver 2 to 17 times the performance of general-purpose computers on a wide range of data-intensive applications by increasing compute cycles and bandwidth and reducing latency.

ProDA: An End-to-End Wavelet-Based OLAP System for Massive Datasets
pp. 69-77

Cyrus Shahabi, Mehrdad Jahangiri, and Farnoush Banaei-Kashani

By design, developers optimize traditional databases for transactional rather than analytical query processing. These databases support only a few basic analytical queries with nonoptimal performance and, therefore, provide inappropriate tools for analyzing massive datasets.

Online analytical processing tools have emerged to address the limitations of traditional databases and spreadsheet applications. OLAP tools support complex analytical queries and handle massive datasets. The authors' ProDA system uses these tools to enable exploratory analysis of massive multidimensional datasets.

LETTERS

SPACE PENS AND PENCILS

Editor's note: The following is representative of several letters we received regarding the urban myth referred to in the The Profession column in *Computer*'s February 2008 issue.

I was disappointed to see that *Computer* printed an urban myth as a fact in the The Profession column in its February issue (R. Natarajan, "On Attending Conferences," pp. 108, 107). It is ironic that this inaccurate statement is printed right after the author states that "there is a great danger of becoming shallow if we abstain from research." Although it might not have been the author's intent, this unfortunate occurrence serves only to fuel a false perception of NASA's incompetence as well as an inflammatory indication of the agency's carelessness at US taxpayer's expense.

Here are the facts:

- During the first NASA space missions, the astronauts used pencils.
- In 1965, NASA ordered 34 pencils from Tycam Engineering Manufacturing at $128.80 per unit.
- Paul C. Fisher developed the Fisher Space Pen and offered it to NASA.
- Mr. Fisher developed the pen without NASA funding.
- NASA purchased approximately 400 pens from Fisher in 1967 at $6 per unit for Project Apollo.
- The Soviet Union also purchased 100 of the Fisher pens.
- Both American astronauts and Soviet/Russian cosmonauts have continued to use these pens.

The problem with pencils is that they are hazardous items in weightless conditions and pure oxygen atmosphere. It took only a few minutes of "research" to obtain reliable facts. They are available from http://history.nasa.gov/spacepen.html as well as from www.spacepen.com/Public/History/index.cfm. It took about the same amount of time to track down the myth sources (www.snopes.com/business/genius/spacepen.asp).

Aleksandar Fabijanic
alex.fabijanic@computer.org

The author responds:

Thanks to Aleksandar Fabijanic and the other correspondents who pointed out that the NASA pen and pencil story is not to be taken as a real historical episode. This could be helpful to readers who might have failed to see that it was just an anecdote narrated in a lighter vein with the purpose of setting the mood before describing more true-to-life examples of innovations brought about by outsiders to various disciplines. In fact, a search for the keyword phrase "space pencil" on Google immediately reveals that this is a well-known joke, and the top hit http://www.thespacereview.com/article/613/1 provides a clear explanation of the real facts.

Raja Natarajan
raja@tifr.res.in

E-VOTING

I read "Secure and Easy Internet Voting" by Giampiero E.G. Beroggi (Feb. 2008, pp. 52-56) with interest. It's great to see a system that encourages greater participation through convenience while ensuring the integrity of the election. The only concern that I did not see fully addressed was the point that "no one can intercept, change, or reroute electronically cast votes." This requirement might not be met for votes cast on a computer with malware present at time of voting.

The "Silent Banker" trojan is an example of this danger; it places itself between the browser and the SSL layer and does not need to break an SSL session to steal money from the victim. Similar trojan malware would have full control over a voting session by modifying the vote results as they are submitted and faking confirmation response screens to make the voter believe the vote was cast as intended. Even if the voting system uses a secure applet, a trojan having full control over the screen, mouse, and keyboard could tamper with the vote.

Antivirus researchers commonly place the number of infected computers worldwide at around 11 percent. If a foreign government or organization were to control a large percentage of these computers, the corrupted votes would have a significant effect on the election results.

Michael Nice
niceman@att.net

I read with interest the February article on Swiss "Secure and Easy Internet Voting." I was hoping that someone had solved the problems, but sadly, that is not the case. Easy, perhaps, but secure?

Nothing in this system prevents man-in-the-middle attacks, and recent discussions of malware that modifies a cracked system' DNS server settings shows this is critical. Nothing prevents one person from being registered multiple times or one person from voting for all the registered voters in one household. The sale of votes is not prevented. It is trivial to monitor the voter's display to make sure he's voting the way he is supposed to, and he can't change his vote later. These all seem to violate at least three of the four Federal Chancellery rules.

Further, recounts are meaningless since the same software will do the recount that did the first, and a modified database of votes will count the same twice, thrice, or more no matter whose software counts things.

The author claims that the "e-voting hardware itself is in a steel cage," but surely none of the home systems that connect to it and are part of the system as a whole are. The system's security relies on tens or eventually hundreds of thousands of home computers.

The most serious statement comes just prior to that, referring to the decision to keep the code from the public. Attackers "with such access could modify voting and auditing

records." Security through obscurity is a dangerous way to operate, since clearly someone does have access to the code and thus could find the means to modify voting records.

And yet, this system has received an award for its "remarkably high security standard." If this is true, then I have an encryption routine I'd like to sell you. You can't see the code for it, but I promise it's very secure. In fact, I've used it to encrypt this e-mail, not just once, but twice. I call it "ROT13."

Mark Kramer
c28f62@theworld.com

"Secure and Easy Internet Voting" contains a glaring contradiction. The author concedes that the Swiss e-voting system he describes falls short of the ACM's recommendations in that it "does not lend itself to a reproducible recording of each voter's actions." He purports to excuse this shortcoming on the grounds that "a paper trail is … dangerous in that it provides a visible receipt [which] could subject voters to bribery from those seeking to buy and sell votes." Yet he overlooks the obvious fact that when a Swiss e-voter uses a computer or a mobile phone, there's no way to know who might be looking over the e-voter's shoulder, buying or coercing his vote.

What Mr Beroggi doesn't understand about the ACM's recommendations is that the "visible receipt" can't be taken away from the polling place, and it is in fact treated by the voting authorities like a paper ballot. It thus poses far less risk of vote buying or coercion than the Swiss e-voting system, which might be easy but is far from secure.

Hamilton Richards
ham@cs.utexas.edu

TYPOGRAPHICAL POVERTY

Catching up on my reading, I found "The Profession as a Culture Killer" (Sept. 2007, pp. 112, 110-111) of particular interest in its remarks on typographical poverty in personal computing and its development.

This poverty was not universal in the early days of personal computing. The Amstrad PCW 8256 word processing computer of two decades ago (www.old-computers.com/museum/computer.asp?c=189) used a special keyboard with a program called LocoScript that allowed applying all accents to any character by simple and obvious keying. It was very popular at the time.

What a pity we can no longer do this.

Piotr Karocki
pkar@ieee.org

We welcome your letters. Send them to computer@computer.org.

Thinking Locally, Acting Globally

David Alan Grier
George Washington University

Even though a project achieves its goal, it also can have unintended consequences.

I live in an activist neighborhood, one of those areas where individuals take up causes in the name of doing good for the broader community.

Sally, who lives three blocks to the west, organized a neighborhood patrol to walk the streets at night and discourage crime. Karl has been worrying about a different kind of crime, a gang of rogue possums that took up residence in one of the alleys and demonstrated remarkable skill at ransacking garbage cans. When they lived on our block, Jeff and Marissa raised funds to build a playground in a nearby park. Tim and Caroline, long before they were divorced, spent untold hours trying to encourage homeowners to celebrate Halloween by decorating their homes in purple and orange lights.

UNINTENDED CONSEQUENCES

All these actions, and many more, were taken with the honest and sincere intent of making our neighborhood a better place to live. Although all the projects achieved their goal, each had unintended consequences. After it cleared the area of lurking burglars, Sally's patrol proved to be a highly successful social event in which the participants eventually spent more time in a local diner than on the streets. Karl expelled the possums only to find that a family of raccoons had taken their place. It appears that, in addition to being good at overturning garbage cans, raccoons are also skilled at picking kitchen locks and emptying refrigerators.

Jeff and Marissa easily collected the money they needed to build the playground, as our neighborhood was in the midst of a baby boomlet, but they found their plans thwarted when they discovered that the playground's proposed location was a historic site. For a weekend in 1932, the park had been a camp for the World War I Bonus Marchers. Jeff and Marissa ultimately needed the equivalent of an act of Congress to put a children's slide and jungle gym on the land.

The Halloween festival, of course, went far beyond Tim and Caroline's intent. It is now a major civic attraction and draws visitors from other parts of the city. Families bring their children to parade down the streets. Cars of teenagers come to party. Costumed revelers dance and preen. The local residents look at the mobs of people and wonder what has happened to the event that was once a little neighborhood party. Last fall, my neighbor Yousef watched the crowds and shook his head. "This is what happens," he said, "from thinking locally but acting globally."

Of course, the computer industry can provide multiple examples of unintended consequences, of actions that were designed to meet a local need but ultimately influenced a much larger community. John Mauchly described an electronic machine to compute ballistics tables, and the offspring of his idea created a multitrillion-dollar industry. The US Department of Defense asked for a communications network that would make it easy to share computing resources, and this idea evolved into the Internet with all of its services. A Cornell graduate student created a small program that could travel across that network, and suddenly no computer in the world was safe.

COMMUNITY NETWORKS

In my neighborhood, the intersection of the local and global is seen in the proliferation of broadband routers. On most nights, I can sit in my bedroom and find a dozen routers with enough strength to reach my house. They range from Yousef's Boccinet (named for his dog) to Xshdict381, which is either the router's serial number or the name of a Klingon starship—I'm not sure which. A few are secured, but most are open to the world.

Lauren, a resident of a nearby basement apartment, once told me how she would borrow bandwidth from the house next door in much the same way that neighbors used to borrow cups of sugar from each other. She would scan the offerings and pick one that appealed to her. She was quite pleased that she knew how to connect to other networks, but she was also concerned about the effect of her action. "Won't the signals get mixed up?" she asked. "How do the bits know which computer is which?" Little did she know that all the bits pass through a common router that is bolted to a pole in the

alley in the same way that the sewage from our homes passes through a common pipe under the street.

If you follow the data line from our alley router, tracing it through a conduit under the street, bypassing the local telephone office and following the main trunk line toward downtown, you arrive at an office that wants to eliminate all of our individual networks and replace them with a single regional or community network. This office is occupied by an energetic young man named Eli, who believes that community networks are going to change the world. "They are the real thing," he said, "Community networks will give people access to hundreds of new services and new activities."

To Eli, a community network is a civic organization, usually run by a government or a cooperative, that delivers high-speed wireless data connections for free or for very low prices. As he sees it, we need these networks to address five serious problems facing our world.

First, they will allow city governments to make their services more efficient. "We can't have digital services until everyone is online," he notes.

Second, they will bridge the digital divide that separates rich and poor householders by reaching those homeowners who can't afford broadband access. "Not everyone can pay for a high-speed cable connection," Eli says.

Third, these networks will improve education. "Think what will happen when every high school teacher can be available 24 hours a day," he pontificates, forgetting that many high school teachers might resist being available to their students every moment of the day.

Finally, Eli claims that these networks will spur local economic development and promote regional tourism.

"Tourism?" I queried.

"Yes, tourism," he replied. "Every neighborhood can promote its cultural and historic sites."

I started pondering how we might present our neighborhood. "Hoover Park: the neighborhood of community activists, five raccoons, and a historically significant jungle gym" didn't seem like the kind of slogan that would attract many visitors during the non-Halloween season.

WIRELESS LEIDEN

Beyond his list of five talking points, Eli believes that community nets will revitalize urban life, a belief that he illustrated with the story of Wireless Leiden, a cooperative that serves a dozen cities and 400,000 residents of the western Netherlands.

The task of building any piece of municipal infrastructure is not easy, even in the best of circumstances.

Wireless Leiden was started in 2002 by three residents of Leiden: Huub Schuurmans, Jasper Koolhaas, and Marten Vijn. Initially, these three wanted nothing more than to build a network that merely connected their own computers. Using standard IEEE 802.11b technology, they quickly assembled a straightforward network. Once they had completed their project, they found that neighbors and friends were interested in using their net, so they decided to "build a citywide wireless network for internal and Internet communication with free access for everybody."

Like many community network promoters, they argued that theirs would improve the local economy. They placed their first nodes so that local businesses could allow their employees to work at home. Next, they deployed routers in public buildings: libraries, schools, and healthcare facilities. Finally, they began placing nodes so that local residents could use the net for their own purposes.

Their business model was quite simple—they financed the first nodes themselves. After the network started to grow, they convinced a few computer vendors to donate servers. For the network nodes, they asked organizations to sponsor connection points for 1,200 euros each. For all users, Wireless Lieden is open and free of charge.

Wireless Lieden has no paid employees. The support staff is a group of 70 volunteers. A wiki contains the technical and operational information. The network has had a few problems that tested the technical skill of its volunteer staff, but as a whole, it has operated well.

Eli finished telling this story with the pride of the righteous. This was the model for future rural networks, he said. The system was fully operational more than 99 percent of the time and had already been copied in a rural region of Turkey.

I noted that Wireless Leiden might be successful because it is located in a prosperous region and because it carries only a relatively small amount of traffic. Each node gets only 5 to 10 connections per day. Eli has waived this objection away. "It's based on standard 802.11b technology," he asserted. "and it is a community effort." I wanted to ask if the staff could also expand, but held my tongue.

MUNICIPAL NETWORKS

The task of building any piece of municipal infrastructure is not easy, even in the best of circumstances. Many a valiant city project has ended in failure with millions wasted, officials resigning, and contractors indicted. Cities pack so many conflicting interests into such a small area that they seem to thwart any attempt to improve urban life, especially when the promised improvements come with the aura of idealism: universal service, clean air, improved education for poor citizens, a more equitable environment for small business, and a better environment for dogs.

During the past five years, local governments and civic organizations

have become increasingly interested in community networks as a way of providing broadband services to areas that fall outside the wealthy urban areas or the regions that are already served by high-tech infrastructure. These efforts include the Champaign-Urbana Community Wireless Network, the Dutch/German Internet Exchange, the San Diego area's Tribal Digital Village, NYCwireless.net, Axia, the Citizens Communications Corporation in Indiana, Infratel, and Prairie iNet in the midwest.

Some of these organizations are cooperatives. Some are ordinary corporations. Some are public/private partnerships. Some are government agencies. They employ technology ranging from commercial Wi-Fi, to WiMAX, to custom meshnets, to experimental white space networks. All of them share a deep faith in the importance of community. They "are still in the nascent stages of the community wireless movement," wrote one commentator, "but the social benefits of ubiquitous, community broadband are becoming obvious."

The benefits of community networks might be obvious, but so are the problems of building a robust wireless network for a modern, urban environment that might cover 150 square miles and two million residents. Such work can't easily be done by three visionaries, fortified with commercial Wi-Fi routers and financed by small donations. Furthermore, the established telecommunications firms have come to view municipal networks as lost business or even as potential competition. In at least 17 states, cities are barred from building or operating a municipal network.

WIRELESS PHILADELPHIA

In Pennsylvania, the legislature barred its cities from operating municipal networks just as the state's largest city, Philadelphia, was starting to design one. This project had begun in the mayor's office and had been promoted with a demonstration network in a small park near city hall. Although the network was described as a way of providing universal Internet access, it was criticized as principally benefiting the rich and well educated. "We're just taking money from hardworking families and giving it to people who can afford [personal digital assistants] and laptops," wrote one critic. "If you don't have a job where you can use your laptop to do your work in a city park, it's not going to benefit you."

The Pennsylvania restrictions ultimately allowed Philadelphia to build its network, but they pushed the project in a more conventional direction. The city began planning the network in fall 2004, with civic meetings and focus groups and other public discussions. As a result of these meetings, city officials initially decided that they would build their network in partnership with private industry. The city would finance, construct, and own the basic infrastructure, while private firms would sell connections to the network at a reduced rate.

The established telecommunications firms have come to view municipal networks as lost business or even as potential competition.

As they continued with their work, the Philadelphia officials began to modify their plans. Rather than employing volunteers or even a local firm to build the network, they hired a large national company to wire the city and deploy wireless nodes. This firm proposed that it, not the city, should finance the construction. In return for sparing the city from selling bonds or raising taxes, the firm would own and market the network. City officials accepted the idea, knowing that Philadelphia voters, like so much of the American electorate, have little faith in the fiscal wisdom of their elected officials. As a result, the network that had begun as a community effort started to look more like a private organization.

The first sectors of Wireless Philadelphia, encompassing 15 square miles, became operational in January 2007. "This is a major step toward achieving our vision of The Entire City Connected," boasted the project manager.

By this time, many observers felt that the city chose wisely when it gave ownership of the network to a private company, as the costs of the network changed substantially during construction. Originally, engineers estimated that they would need 20 to 25 wireless nodes per square mile, but their plan left areas of weak or limited coverage. In the end, the network required 40 or more nodes per square mile. That number might have seemed high to the engineers in Philadelphia, but it is small when you compare it to the number of routers in my neighborhood.

WAR CHALKING

One cold morning, I hurried out the front door to get the morning newspaper to prevent it from being snatched by Joe, who lives in the alley a couple blocks to the north of us. In the morning, Joe will pick up a paper from a neighborhood doorstep and read it, or perhaps pretend to read it, as he walks down the street. As he goes along, he drops pages. First the ads, then the sports, next the style section, then metro, and finally the front page. When Joe is done, the sidewalk is littered with a stream of blowing newsprint leading toward the doorstep where he found the paper.

As I stooped to pick up the paper, I saw a chalk circle, neatly drawn on the sidewalk with a few notes scribbled next to it. I thought it odd that someone would have drawn such a thing in the night. I looked up and saw similar circles in front of the other homes. The ones by Karl's door and at Jeff and Marissa's house had no writing. Tim and Caroline's house was marked by a circle like ours. I stood outside long enough

to realize that my feet were getting chilled and returned inside unsure of what I'd seen.

"It's called war chalking," explained Eli when I described the circle. "Those are marks that show the location of wireless routers in your neighborhood. Someone has walked up and down your block with a cellphone or a laptop checking for routers."

"What does it mean?" I asked.

"Some proponents of community networks believe that we should build large urban nets by opening our personal wireless routers to friends, neighbors, and pedestrians on the street."

"They can't be serious," I said.

"Yes they are," he replied before starting to discuss how such a strategy would have trouble with signal strength and scope of coverage.

As Eli talked, I recalled Lauren's ability to borrow bandwidth from the neighborhood. I wondered who else had tapped into our routers. Without any plans or coordination on our part, we had apparently created the beginning of a community network. Hoover Park: good people, nice raccoons, and wireless broadband for those who can find it. We will now have to see what the unintended consequences might be.

David Alan Grier is *an associate professor of International Science and Technology Policy at George Washington University and the author of* When Computers Were Human *(Princeton University Press, 2005). Contact him at grier@gwu.edu.*

APRIL 1976

ELECTRONIC FUNDS TRANSFER (pp. 14-15). "Neither cash nor checks are about to become obsolete, but the white collar worker is an increasingly threatened species. That is one of the major findings of a just-released Arthur D. Little report on the sociological and economic impacts of electronic funds transfer."

"Principal social issues which ADL believes require regulatory action concern freedom of choice and possible misuses of EFT. The potential for invasion of privacy and new types of crime will exist within the increasingly interconnected masses of financial information. Delays in error detection and liability (when more parties than the payor and payee are involved) will compound the problems, but not make them insurmountable."

[Update: "EFT Hastens White Collar Obsolescence, Says Arthur D. Little Study."]

GRAPHIC TERMINALS (p. 18). "Graphic terminals allow on-line interaction between the user and his program, thereby enabling him to alter the displayed picture according to his requirements. However, a major problem with the use of a graphic terminal is the cost associated with it. There is no doubt that low-cost graphic terminals are available, but their utility is very limited since they have no processing capability (or intelligence) and must depend on a host processor for all processing. … On the other hand, graphic terminals which have local processing capability are very expensive. With the availability of microprocessors, it is now possible to resolve this dilemma: a reasonable amount of intelligence can be added to the terminal at a very reasonable cost. Such a terminal has been developed at the University of Ottawa."

[J. Raymond and D.K. Bannerji, "Using a Microprocessor in an Intelligent Graphics Terminal."]

LEARNING MICROPROCESSING (p. 56). "EBKA Industries has introduced the 'Familiarizor,' a complete microcomputer system for use in learning microprocessing theory and operation. The system comes with its own hexidecimal keyboard and display built into a single PC board. No teletype or other terminal is required. With the addition of a power supply, it becomes a complete, simple-to-operate microcomputer."

"The Familiarizor uses an 8-bit MOS Technology 6502 microprocessor, which can address up to 65K bytes of memory. On-board memory consists of 1K bytes of RAM for user programs and two 8-bit I/O ports. A 256-byte monitor program, supplied in one 1702A erasable PROM, and an on-board terminal replaces more complex lights and switches to permit simple loading, examination, running, debugging, and modification of programs.

"The Familiarizer is available both in kit form, containing all parts, manuals, and documentation, for $229, or completely assembled for $285. Optional power supply is available at $58."

[New Products: "EBKA 'Familiarizor' Microcomputer."]

NEW COMPUTER FAMILY (p. 59). "Digital Systems Corporation has announced the introduction of its new GALAXY/5 family of computers. Designed primarily as a medium-sized multiprocessing system for use in teleprocessing, timesharing, data base management, and similar interactive applications, the GALAXY/5 is available with from one to four central processing elements. Each processing element is a completely independent CPU with its own registers and arithmetic units. All processing units, however, share the main memory and the input/output equipment.

"The main memory is constructed from 16,384 byte modules, with 1,048,576 bytes of main memory constituting the maximum configuration. …"

"The manufacturer claims that the GALAXY/5 is the first large scale computer to use many micro computers to perform functions (such as in the disk controller and the communications controller) previously performed only by hardware logic."

[New Products: "New Medium-Sized Computer System."]

TOXICOLOGY (pp. 63-64). "The Los Angeles County Coroner is using a computerized analysis system for solving pathological mysteries that previously would have been filed away as 'unknown.' With its recently installed gas chromatograph/mass spectrometer, the Coroner's toxicological laboratory has been able to determine the exact cause of death in many cases where conventional equipment would not have been able to do the job."

"The system works this way: a toxicologist enters the extracted specimen into the unit. Then, a gas chromatograph separates chemical mixtures into constituents for analysis.

"The constituents then are electronically scanned, and converted into digital representations [of mass spectra] that are stored in memory. …

"Through a keyboard, the toxicologist can ask the NAKED MINI to search its memory to identify the compound most like a given mass spectrum. The minicomputer provides an immediate printout of all compounds resembling the unknown, and indicates the degree of match."

[New Applications: "Computerized System Solves Complex Toxicology Problems."]

CERTIFICATION (p. 82). "The CDP (Certificate in Data Processing) examination has been expanded to include six test sites beyond the continental limits of the United States and Canada, the Institute for Certification of Computer Professionals announced during its recent annual meeting.

32 & 16 YEARS AGO

"In addition to the 88 U.S. and Canadian testing locations, the CDP exam was administered on February 21 in London, Kaiserslautern, Singapore, Murdoch (Australia), the Canal Zone, and Puerto Rico."

[Update: "CDP Examination Goes International."]

APRIL 1992

WAFER-SCALE INTEGRATION (p. 6). "Traditional very large scale integration (VLSI) circuits are created by fabricating a wafer, testing the individual die, dicing the wafer, and packaging the defect-free chips. A microelectronic system is implemented by mounting the individually packaged chips on printed circuit boards. By contrast, in monolithic WSI [wafer-scale integration] a wafer may be fabricated with several types of circuits; the circuits are tested and the defect-free circuits are interconnected to realize the system on the wafer."

TASK-FLOW ARCHITECTURE (pp. 10-11). "The major difference between task-flow and other machines is the concept of sending computations to stationary data objects rather than sending data from memory to stationary processors. Task-flow machines do not contain separate processors and memories; instead, multiple cells contain both computing and memory elements. Data arrays are partitioned across multiple interconnected cells. Computation is performed by a set of tasks flowing through the network. Each task is executed by sending *transmission packets* (TPs) along linked lists of *memory packets* (MPs). Each MP contains a data element, the next instruction to perform, and a link to the next MP. Each cell's memory is organized as sequentially addressable MPs. The linked list nature of the MPs is useful in sparse matrix computations, database query processing, distributed reduction operations, associative memory, and coarse-grained application synchronization."

WAFER-SCALE MEMORY (p. 27). "Wafer-scale memory using a 4-Mbit DRAM … has hundreds of lines so that it can achieve parallel access for high data throughput. Estimating the defect tolerance scheme gives us an area-efficient system with optimized additional circuits.

"In our system, access is 20 times faster than in serial-access memory, and throughput is 10 times higher. This parallel-access architecture for wafer-scale memory is thus advantageous for high-performance, high-density, and low-cost memory storage."

LASER RESTRUCTURING (p. 41). "The restructurable very large scale integration (RVLSI) program at MIT Lincoln Laboratory has established the viability of using a laser to restructure wafer-scale circuits for customization and defect avoidance. Wafer-scale circuits are built with a standard integrated circuit fabrication process when the diffused-link restructuring device is used. … Wafer-scale implementation requires system, architecture, and technology trade-offs, as demonstrated by nine wafer-scale systems we have built.

"To use the RVLSI technique, we first fabricate wafers with redundant circuit modules, called cells, and uncommitted interconnects. After fabrication, we test cells and interconnects, and connect operable cells with a laser to build the desired system. We can also use the laser to customize circuitry and perform diagnostic testing."

WSI PROSPECTS (p. 58). "The concept of wafer-scale integration is already more than 20 years old. Yet despite many clever innovations, years of development, and millions of dollars invested in WSI, a large digital system implemented on a single silicon wafer is still more an academician's dream or a costly exhibit on the wall of a government-funded research lab than a commercially viable product delivering all the theoretically possible performance, reliability, and economic gains.

"Why this discrepancy between hopes and realities? When will WSI finally prove its usefulness? Do recent promising announcements about WSI memories indicate the beginning of a new era of WSI microelectronics?"

TABLET PC (p. 90). "CalComp Digitizer Products Group has developed DisplayPad, which lets users turn a standard desktop or portable computer into a pen-based system. DisplayPad uses company electromagnetic digitizer technology in a cordless stylus that features tip-pressure sensing, tilt-angle sensing, tip-height sensing, and graphic-effects controls. The tablet is a high-resolution LCD grid printed on ultrathin transparent film that acts as an antenna for pen-transmitted signals. The tablet features 640 × 480-dot resolution with 64 shades of gray."

USENET (p. 112). "No other public bulletin board comes close to Usenet in breadth and bandwidth. It has nearly 2 million readers and carries about 25 megabytes per day, and it is still growing exponentially. Although it uses traditional international networks like Internet and Bitnet for data transport, Usenet itself has no central organization. A few sites sell news feeds and for-profit data, but Usenet mostly consists of volunteers exchanging free data. …"

PDFs of the articles and departments from the April 1992 issue of Computer *are available through the Computer Society's website: www.computer.org/computer.*

Editor: Neville Holmes; neville.holmes@utas.edu.au

INDUSTRY TRENDS

The Move to Make Social Data Portable

Karen Heyman

During the past few years, social networks such as Bebo, Flickr, Facebook, Friendster, LinkedIn, MySpace, and Orkut have attracted millions of users who are involved in blogging, participatory book reviewing, personal networking, photo sharing, and other similar activities.

The users have freely submitted personal data to proprietary databases so that they can communicate and share with other participants. However, some have begun to grumble that the networks' proprietary code and lack of APIs don't allow them to export data from one site to another.

The data in question includes identity-related information such as membership numbers; contacts; relationships; personal details; and media such as photos and video, noted Michael Pick, a social-media consultant.

For social-network operators, this information lock-in was supposed to be an advantage. Forcing participants to re-enter data for each service ostensibly would encourage them to stay with the same service.

Now, though, as the battle between the big social-network players heats up, users are demanding some level of data portability between the different sites so that they can move or transfer their information without re-entering it every time. Proponents say the failure to provide portability could cause many users to lose interest in social networking. They are calling for open standards and other technologies to unlock much of the information they enter into the networks.

"Data portability will allow me to take my data with me, increasing competition [among sites] and creating a better environment for everyone using the Web," said Duncan Riley, veteran Web developer and the b5media blog network's founder.

Proponents also advocate data portability because it could let multiple social Internet services work together using the same data.

Some social-network operators and other Internet players are expressing interest in data portability, although they haven't taken many steps to implement it yet.

"[Letting] users have ownership and control of their data is the right thing to do for the users," said John McCrea, vice president of marketing for Plaxo, an online address-book synchronization service. "We'll all make a lot more money if we collectively remove the pain that users are currently experiencing as they attempt to use a bunch of different socially enabled services."

However, some sources say proponents must be careful with data portability because the process could cause security and privacy problems.

TECHNOLOGIES

Many social networks do not enable data portability because they use proprietary code and don't provide APIs, which let multiple programs talk to one another.

The issue is complex because programs often describe data differently, explained research professor Craig Knoblock, a senior project leader at the University of Southern California's Information Sciences Institute.

Essentially, he explained, the solution will involve information integration and the creation of universal formats to which data can be written. Of course, he noted, the ongoing challenge will be legacy data that doesn't conform to these formats.

Proposed solutions combine markup languages and various tools. However, even proponents say it is early in the process and thus unclear whether and how any or all of the proposals would ultimately be integrated into a data-portability standard.

Several technologies could be part of the solution.

RSS. Really simple syndication has become a de facto standard. The technology, which the RSS Advisory Board and the RSS-DEV Working Group support, has evolved and become the leading XML-driven format for exchanging syndicated content and various types of media—such as audio, images, and video—between platforms.

OpenID. This is an open and decentralized identity system managed by the OpenID Foundation and supported by companies such as Google, Microsoft, and Yahoo! The increasingly popular system is designed to enable collaboration and interoperation between the Web's different login and registration systems. Basically, OpenID lets users log in to different sites without reregistering basic information such as first name, last name, and e-mail address.

OAuth. This open protocol allows secure API-based authentication in a standardized way from desktop

or Web-based applications. Via the technology, participants authorize websites and applications to act on their behalf on target sites, which might or might not be trustworthy, without having to give up usernames and passwords.

Microformats. The many types of microformats are simple conventions for embedding XML-based semantics in webpages so that machines can extract, index, search, save, or cross-reference them. Microformats identify specific kinds of data, such as people or events.

They let users, for example, copy and paste contact information from a blog or other webpage to their address book, instead of manually retyping it. Eventually, microformats could make published information easily sharable and searchable.

RDF. The World Wide Web Consortium's XML-based Resource Description Framework provides a way to add a substantial amount of metadata to content. This would let machines exchange information without human intervention, thereby improving data portability.

APML. The Attention Profiling Markup Language lets users selectively record their *attention data*—such as the sites they visit, the search terms that interest them most, the content they commonly link to—and share it with their favorite websites and services via a portable file format. Proponents say APML, sponsored by Faraday Media and the APML Workgroup, will enable sites and services to provide visitors with material they are most interested in seeing.

SIOC. The Semantically-Interlinked Online Communities Project has submitted this technology as a proposed standard to the World Wide Web Consortium. SIOC (pronounced "shock"), which Figure 1 shows, provides ways to interconnect discussion platforms such as blogs, forums, and mailing lists. It consists of the SIOC ontology, an open, machine-readable format for expressing the information contained in these platforms; metadata producers for numerous popular platforms; and storage and browsing systems for leveraging the data they contain.

FOAF. Friend of a Friend technology provides a machine-readable ontology describing people, their activities, and their relation to other people and objects. In essence, it creates machine-readable webpages describing people, the links between them, the things they create and do, and their social networks. Users generate FOAF files on their Web server and share the appropriate URLs so that machines can use the information in the files, thereby enhancing data portability.

The technology, which the FOAF Project supports, was developed in 2000 but never really took hold. Now, though, it has attracted new interest.

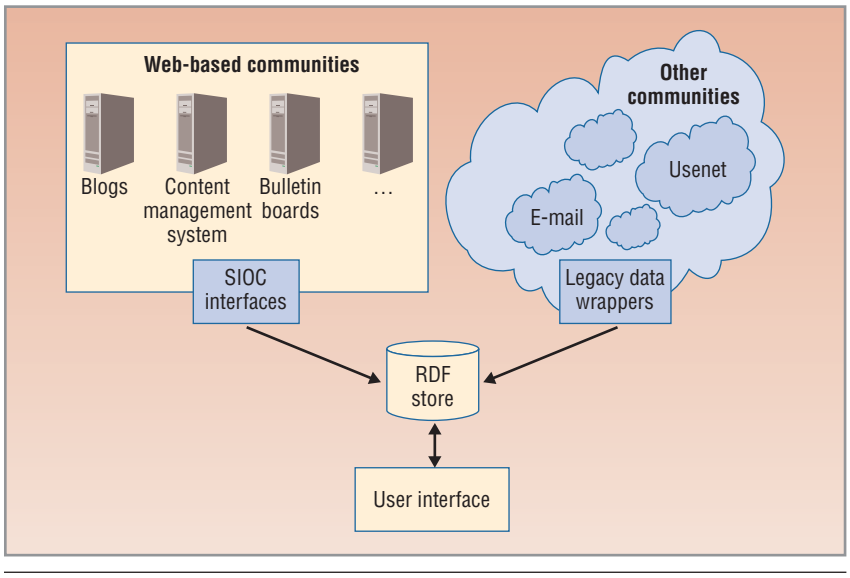


Figure 1. The Semantically-Interlinked Online Communities Project has developed an approach that interconnects discussion platforms such as blogs, forums, and mailing lists. It uses SIO interfaces and legacy data wrappers to bring data into a single, user-accessible store that works with the World Wide Web Consortium's XML-based Resource Description Framework. RDF lets users add metadata to content, enabling machines to exchange information without human intervention and thereby improving data portability.

ORGANIZATIONAL SUPPORT

Several organizations are participating in data-portability activities.

Data Portability Workgroup

The DPWG (www.dataportability.org) is trying to coordinate and formalize the process of developing data-portability technologies. The organization is attempting to define the nature of data portability on social-networking sites, promote best data-portability practices, and advocate open standards for implementing these practices.

"Open standards are the key building blocks," explained workgroup chair Chris Saad, who is also CEO of Faraday Media, which builds tools for observing users' online activities and delivering relevant information to them.

The DPWG is not promoting one approach or developing a new technology but instead is working as a facilitator and moderator for the various data-portability efforts, he said.

Large companies

Major Internet players—such as Digg, Facebook, Google, Microsoft, and Yahoo!—are starting to participate in data-portability-related activities.

For example, Microsoft joined the

DPWG this past January and has since publicly committed to opening up many of its APIs and communications protocols.

"The logical evolution of the Internet is to enable the removal of barriers to provide integrated, seamless experiences but to do so in a manner that ensures that users retain full control over the security and privacy of their information," said Dave Treadwell, Microsoft's corporate vice president of Windows Live platform services.

"When information is portable and transferable, users will gravitate to the services and applications that best meet their needs and preferences," said Google developer advocate Kevin Marks. "We view this as a positive development. Portability encourages innovation and improvement by fostering competition and user choice."

He noted that Google, which joined the DPWG, has released its Social Graph API. The API makes it easy for social-application developers to access information that Google has extracted about publicly available connections between participants in different Web services. The developers can let users find data on their social connections across the Web. Users can then easily add their contacts when starting to work with a new social application.

The Social Graph API uses the same algorithms that Google's search engine works with to discover how people are connected online.

CONCERNS

Facebook chief privacy officer Chris Kelly said achieving data portability is more complex than some advocates suggest. "We don't have a problem with data portability," he explained, "but there are all sorts of privacy and security worries. There are many people who would gladly attempt to exploit somebody else's personal information [if that would give them] one point of entry into a network."

"Data portability will require that the community think long and hard about ways to ensure security," said Google's Marks.

Other issues include whether making personal information easily available will threaten privacy.

"We want to make sure there are rules and controls that minimize these problems," Kelly said. "That is a critical part of these discussions, but it's something that, in a rush to call for data portability, most proponents haven't effectively considered. We joined the Data Portability Workgroup because we want to show that we're serious about having that conversation. But to just say that you can have a completely open system ignores that there are serious privacy and security challenges."

Some people say the push for data portability doesn't go far enough.

"One of the limitations of data portability is that it's not interoperability," explained Marc Canter, CEO of Broadband Mechanics, a vendor of social-networking platforms. He said being able to move data between systems doesn't necessarily mean that the systems can easily talk to one another.

Canter also expressed concern that some of the major Internet companies might see data portability as a way to get at other sites' data without necessarily sharing their own.

Another issue is whether major companies that have joined data-portability groups will take a leadership role and participate enough to make portability happen.

The deepest skepticism voiced by many about data portability isn't the idea but rather the implementation. "Portability of data is being defined on the fly and without much certainty as to what the final definition or destination will be," said Bill Washburn, the OpenID Foundation's executive director.

Data portability's many young proponents don't always realize how technology development occurs and might not understand that some things they want to do won't work, he explained.

The process of developing data portability standards is just beginning, so it is unclear how the technology will look if and when it's finalized. Nonetheless, proponents say data portability is nothing less than the Web's future. "It is central to the next phase, which is the emergence of the social Web," said Plaxo's McCrea.

In the long run, data portability will help social-networking companies, predicted John Breslin, SIOC Project founder and leader of the Social Software Group at the National University of Ireland's Digital Enterprise Research Institute.

"I think companies are realizing that providing mechanisms for data portability doesn't necessarily mean that users will leave your site en masse," he explained. "By providing open methods to access data on sites, the big players are allowing other people to build new and interesting applications on top of their sites, which encourages users to stay on board."

"And," he added, "users feel happy in knowing that they have access to their data if they need it, building loyalty as opposed to anger against restrictive [policies]. Lastly, these companies can open up avenues for an influx of new users who can easily bring their data over from other sites via data-portability mechanisms."

People want the freedom to use the Internet with anyone, anywhere, said Washburn. "It is only a matter of time until some form of portability will overwhelm the desire of even the biggest walled Web sites to own and control the identity-related data of their users."

Karen Heyman is a freelance technology writer based in Santa Monica, California. Contact her at klhscience@yahoo.com.

Editor: Lee Garber, *Computer*: l.garber@computer

TECHNOLOGY NEWS

Proponents Try to Rehabilitate Peer-to-Peer Technology

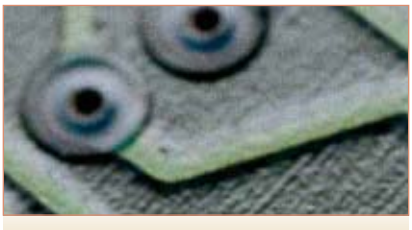
Sixto Ortiz Jr.

Many people associate peer-to-peer technology with file-sharing applications such as BitTorrent, Gnutella, Kazaa, and Napster and with concerns about the unauthorized, free distribution of video, audio, and other copyrighted content. These concerns have led the entertainment industry to crack down on P2P systems. For example, the US Recording Industry Association of America (RIAA) has taken numerous file sharers to court to recover copyright-infringement damages.

"P2P was adopted first by folks looking to share unauthorized content and is still widely used for that purpose," said analyst Cynthia Brumfield at Emerging Media Dynamics, a technology advisory firm. "As a consequence, P2P has gotten a bad rap and is perceived as a threat."

In addition, ISPs have complained that high volumes of large video files and other P2P traffic traveling between multiple peers have hurt their networks' performance. This has also inhibited P2P usage in some cases and cast doubts upon its future growth potential.

In response, proponents have undertaken a major effort to change P2P technology, rehabilitate the approach's reputation, and encourage its use for fast, efficient content distribution and improved Internet-based communications services, including telephony.



Vendors are improving the technology's efficiency while ISPs are exploring content filtering and other measures to reduce the chances that people can use P2P networks for the unauthorized distribution of copyrighted material.

These improvements occur as the increasing use of high-bandwidth applications on the Internet—such as video, file sharing, conferencing, and telephony—make P2P's ability to efficiently distribute content particularly desirable.

The technology would eliminate the need for content providers to operate complex, centralized distribution systems that require large amounts of storage and network bandwidth, said Matt Zelesko, senior vice president of engineering for Joost, an online video provider that uses P2P.

However, the approach might have to clear several technical and marketplace hurdles before it can achieve mainstream adoption.

P2P CONTROVERSY

Traditionally, peer-to-peer technology has been used for the online sharing of media and other files. This has caused the enormous controversy that has surrounded P2P during the past nine years, beginning with complaints about Napster enabling the unauthorized distribution of copyrighted material.

The development and implementation of digital rights management technology designed to prevent or limit such distribution have lowered the intensity of, although not eliminated, the issue. There have also been campaigns against the unauthorized use of copyrighted material, as well as lawsuits by content owners and their representatives, such as the RIAA.

Service providers are cooperating. For example, AT&T is working on technology to keep pirated content from even entering its network by blocking access to sites that deliver pirated material.

Inside today's P2P

Today's P2P is different than its initial incarnation, said Yale University associate professor Y. Richard Yang.

In the past, P2P either let peers communicate via a server or let a peer act as a server for sending an entire file to a requester. The technology now typically uses techniques like *swarming*, which breaks files into smaller parts—typically 256 Kbytes each—and lets peers participating in the process exchange pieces until requesters obtain a complete file, he explained.

When clients want to obtain a large file, they first download a small file from a P2P server or website that contains information about the material they want. The download generates a request that identifies the file; the requester's user identification, IP address, and port; peers that have the file; and the maximum number of peers to which the client wants to connect. The P2P applica-

Published by the IEEE Computer Society

tion then selects the peers to use and has them provide parts of the file to the requester.

This distributed system is more efficient and cost-effective than the traditional model because it doesn't require the large amounts of storage or bandwidth necessary for a server or a peer acting as a server to keep and then transmit large amounts of data.

This minimizes how often media companies and video-distribution services must expand their server farms or network infrastructure. According to Joost's Zelesko, this is why his company uses P2P.

The technology also lets individuals and smaller companies, not just big players, distribute large files.

Because of its efficiency, today's P2P is commonly deployed for uses such as video-file and -feed distribution, distance learning, telemedicine, Internet telephony, and scientific applications that use distributed computing to tackle problems like identifying new drugs.

P2P now reduces data corruption and thus enables more reliable data delivery. The technology also implements techniques that let file-sharing applications work through firewalls and network-address-translation devices, which makes more peers available to participate. And P2P works with decentralized—rather than the traditional centralized—lookup tables, which makes data more easily accessible.

Traffic concerns

The large amount of data in video and other files that P2P networks carry during file sharing and media streaming has caused traffic congestion for ISPs, said Time Warner Cable spokesperson Alex Dudley.

Some sources say P2P communications represent up to 70 percent of all Internet traffic.

P2P is a problem for service providers because they didn't design their networks to provide high bandwidth on both the uplink and downlink. For example, Comcast offers high-speed Internet service with downlink speeds of 6 Mbits per second but uplink speeds of only 384 Kbps.

Users download much more data than they send out, according to Comcast director of corporate communications Charlie Douglas. Thus, ISPs allocate more bandwidth to the downlink.

However, P2P's distributed model requires users to both send and receive large files and, therefore, have high-speed uplinks and downlinks.

Unauthorized file sharing and high bandwidth consumption hurt P2P's reputation.

ISPs have taken steps to mitigate P2P's effects. For example, when congestion occurs, Douglas said, Comcast uses several network-management technologies that slow some P2P traffic, to keep service levels high for all customers. For example, they can identify peer-to-peer traffic via packet inspection, which reveals whether a P2P protocol is being used, and then reduce the transmission's bandwidth allocation.

Some customers have claimed that Comcast is blocking, not delaying traffic, an allegation the ISP denies. The customers say this violates the concept of Net neutrality, which says all content transmitted on a broadband network should be treated the same, without regard for the application that delivers it. The US Federal Communications Commission has held a hearing on the issue.

Time Warner's Dudley said the company is experimenting with a new business model in which heavy users would pay for the bandwidth they consume. Having to pay for a disproportionate amount of bandwidth might encourage less usage in some cases.

A NEW DAY

Several important technical developments—such as faster processors and networks with more bandwidth—are helping to make P2P more desirable.

For example, Joost's Zelesko noted, the increased computing power in standard PCs has made it more practical to employ strong encryption to protect distributed content.

To optimize utilization, said Yale's Yang, many peer-to-peer applications now let users manage and control their participation, such as the amount of upload bandwidth they want to contribute or the specific files or folders they want to make visible to the P2P network.

Improving P2P with P4P

Verizon Communications, peer-to-peer software provider Pando Networks, and Yale University researchers, among others, have formed the P4P (Proactive Network Provider Participation for P2P) Working Group within the Distributed Computing Industry Association (DCIA; www.dcia.info).

The P4PWG includes organizations such as AT&T, BitTorrent, Cisco Systems, Telefónica Group, Joost, LimeWire, VeriSign, and Washington University. The Motion Picture Association of America and content providers such as Cablevision, Comcast, Cox Communications, NBC Universal, and Time Warner have joined as observers.

The group is developing a framework within which ISPs can communicate with content-delivery networks to let P2P networks obtain better connectivity and to let ISPs effectively manage network traffic.

Currently, P2P services don't generally calculate the best route to use to transmit data and thus sometimes use peers that are far from one another, increasing latency and bandwidth consumption.

As Figure 1 shows, P4P seeks to establish relationships between P2P applications and ISPs in which the service providers would share topology and peering-cost information, enabling the establishment of traffic routes that benefit both parties.

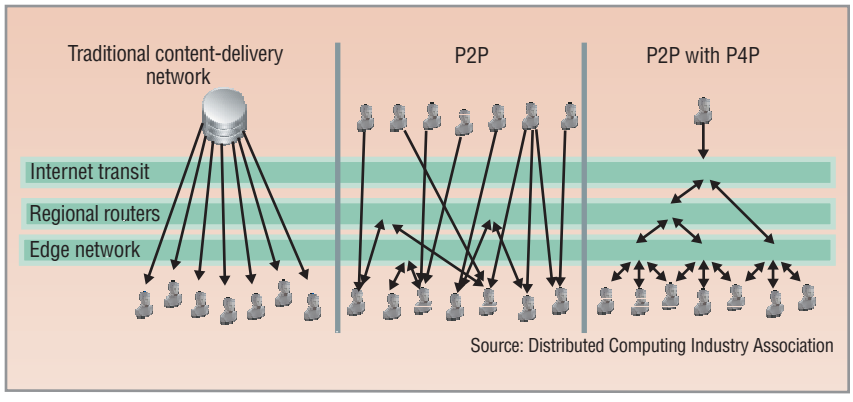


Figure 1. With traditional content-delivery networks, a server sends the same file to each of various requesters, potentially requiring large amounts of storage and network bandwidth. Standard P2P technology addresses this by breaking files into small parts and letting participating peers exchange pieces until requesters obtain a complete file. However, these services don't generally calculate the best route to use to transmit data and thus sometimes use peers that are far from one another, increasing latency and bandwidth consumption. With P4P technology, P2P applications and ISPs share topology and peering-cost information, enabling the establishment of efficient routes for traffic.

Peering costs reflect the expense that ISPs associate with transmitting data among peers.

Topology information can reveal the shortest routes for P2P traffic, while peering costs can show the least expensive routes, which may not be the same as the shortest because of factors such as network demand and congestion.

P4P software uses this information to let P2P applications make more intelligent decisions regarding the selection of peers to use for transmitting data, noted DCIA CEO Marty Lafferty. The software runs algorithms that automatically calculate and update the peering costs associated with an ISP's P2P transactions.

Thus, Yang said, P4P would let networks use peers that are close to one another and also avoid congested links. This would enable faster P2P downloads and reduce bandwidth consumption.

Lafferty said P4P's benefits depend on many factors such as an ISP's network topology, the provider's bandwidth-usage policies, network capacity, and the nature of the distributed content. In general, though, he said, P4PWG studies indicate their approach cuts P2P bandwidth consumption by about 50 percent and peering costs by about 60 percent, and increases transmission speeds by 20 to 50 percent.

A potential problem is that P4P requires ISPs and P2P content distributors to work together and share information. Historically, they have not done so and have even been adversaries at times, said Pando chief technology officer Laird Popkin. Conflicts have arisen over the unauthorized distribution of copyrighted material and P2P traffic loads.

Efforts to protect copyrighted material could help reduce this conflict, according to Lafferty. In addition, Popkin said, both groups stand to increase revenue by cooperating.

The P4PWG is conducting field tests of its technology. Lafferty said he can't predict when it will be ready for commercial use.

Content filtering

Participants are considering content filtering as a way to reduce the unauthorized transmission of copyrighted material via P2P networks. Content filtering tries to identify copyrighted material based on statistical analyses of files and the identification of various factors—including information in headers—and then blocks the content.

The French government is moving ahead with a plan to require ISPs to use content filtering to monitor their networks for illegal traffic and cut off Internet access for those caught sharing copyrighted files via P2P without authorization.

AT&T is testing filtering technology from Vobile, among others, to help it identify pirated video content and eliminate the material from its network.

Vobile chief technology officer Jian Lu said the company's VideoDNA technology—deployed in either hardware or software—examines a file's video characteristics and generates a profile that is a compact, unique, numerical representation of distinct spatial and temporal features of the original content.

The system forwards the profile to a VideoDNA server, which compares it to profiles in a back-end database to determine whether the file contains copyrighted material. Properly acquired and transmitted files generally have different profiles than those being distributed without authorization.

The Electronic Frontier Foundation, an electronic-media free-speech advocacy organization, has raised concerns about AT&T's filtering plans, saying the company may use it to discriminate against P2P applications or look at customer data.

Other concerns include content filtering's scalability as the volume of material passing through a network increases. However, said Lu, advances in video-identification algorithms and increases in computing power should help with this.

Other issues

P2P's bad reputation for unauthorized file sharing and bandwidth

consumption could hold back the technology's future adoption.

P2P creates potential security concerns, including bugs or malware—possibly disguised via encryption or compression—in P2P software or shared files. In addition, employees could use file sharing to send out sensitive corporate or government information. And systems could have problems determining whether peers are who they say they are. In addition, P2P gives computers access to other machines' hard drives, which enables hackers to plant malware or software that can steal information.

For P2P to succeed commercially, vendors will have to find reliable business models. BitTorrent has developed BitTorrent DNA, an enterprise version of its software for streaming high-quality video, and sells it to companies. Other business models could include charging for P2P services.

The growing number of high-bandwidth applications such as videoconferencing, telemedicine, and video distribution will encourage the increased use of P2P, because of the technology's efficiency. Thus, ISPs and content owners will have to find ways to work with it.

"Users don't care where they get their files," said Joost's Zelesko. "If they want to watch a video or access a file and it's not available on a legitimate service, they'll get it from an illegitimate one."

Now, though, P2P is gaining credibility as an accepted way to efficiently and cost-effectively distribute content, and it thus could be a disruptive technology, said DCIA's Lafferty.

According to Emerging Media Dynamics' Brumfield, "P2P technology is an extremely versatile and efficient way to transmit video. I expect that the technology and its various refinements will play a major role in the development of online video."

Zelesko said new P2P applications will rely more on ISPs and content distributors cooperating in the delivery of material across the Internet than on technical improvements. Also, he added, the industry will have to improve P2P's image by educating content owners and ISPs about attempts to both improve the technology and keep copyrighted content from being transported without authorization.

Vobile's Lu said, "The technology needs to be tamed to realize its great potential, which requires pragmatic attitude, creative thinking, as well as technological innovations."

Sixto Ortiz Jr. is a freelance technology writer based in Spring, Texas. Contact him at sortiz1965@gmail.com.

Editor: Lee Garber, *Computer*; l.garber@computer.org

NEWS BRIEFS

E-Paper Soon To Be in Living Color

Researchers have begun developing a color version of electronic paper, just when the monochromatic version has begun to take off in consumer applications. E Ink is one company working on the concept and has already released a demonstration version.

Color e-paper could be used in numerous applications, including e-books, e-magazines, tablet PCs, cellular telephones, and other handheld devices, said E Ink vice president of marketing Sriram K. Peruvemba.

It could also be used with billboards and other forms of outdoor advertising, said Lawrence Gasman, principal analyst at NanoMarkets, an industry-analysis firm.

E-paper technology, around since the 1970s, uses millions of tiny microcapsules. In the monochromatic version, each microcapsule contains positively charged white and negatively charged black particles suspended in a clear fluid. If a negative electrical field is applied to a part of the area, the white particles move to the top of the display, and vice versa. This creates the desired display.

"There are many approaches to achieving color, and we plan to have a color product in 2010, so I don't want to divulge the approach we are taking at this point," said Peruvemba. "The demo units we have shown have a color filter on top of the display that is similar to what is used in LCD technology. Reflected light from monochrome e-paper passes through the color filter as needed to create the desired color images."

"And," he said, "we have an improved substrate material that increases contrast, renders the display brighter, and improves the speed of response [to commands to change images]." The company declined to identify the new material.

E Ink plans to use a flexible substrate for its thin-film-transistor displays so that users can roll them up and store or transport them more easily. Many companies—both manufacturers and device makers—have expressed interest in roll-up display applications. Currently, e-paper uses a film laminated on a glass substrate, which is rigid.

E Ink is not the only company working on color e-paper, said Gasman.

In general, he explained, the technology is not good enough for widespread commercial adoption yet. Moreover, he added, users have only now just begun adopting monochrome displays, although they eventually will demand color.

When products do come to market, he said, they will have to be high quality. The e-paper market cannot afford to bring a color product to market prematurely just to compete with other types of displays, he explained.

"It's going to be three or four years before you really see color good enough to go to market," he predicted.

E Ink has developed a demonstration version of color electronic paper, which could be used in applications like e-books, e-magazines, tablet PCs, cellular telephones, other handheld devices, and even outdoor advertising such as billboards. Several companies are working on color e-paper, just as consumers are beginning to adopt the monochromatic version.

News Briefs written by ***Linda Dailey Paulson****, a freelance technology writer based in Ventura, California. Contact her at ldpaulson@yahoo.com.*

Editor: Lee Garber, ***Computer*****, l.garber@computer.org**

IBM Develops a New Type of DNA Computing

Researchers are exploring new roles that DNA might play in computing. IBM scientists, along with California Institute of Technology senior research fellow Paul Rothemund, are assessing the feasibility of using the genetic material in self-assembly techniques for making semiconductors and other nanoscale devices.

The DNA would act as scaffolding for carefully arranging carbon nanotubes—strands of carbon atoms that conduct electricity—into arrays that could serve as chips for performing calculations or storing data. This differs from traditional DNA computing, in which molecules of the material use biological processes to perform computing-related tasks.

In the new approach, electron-beam lithography creates a pattern on the substrate with sites that bind DNA molecules in predetermined locations, said Greg Wallraff, an IBM Almaden Research Center research staff member.

The substrate is generally silicon coated with any of a number of films, said Jennifer Cha, IBM Almaden Research Center research staff member.

"The surface is covered by a water solution containing the DNA structures, which then self-assemble on the patterned features, one DNA molecule per site," Wallraff explained. The researchers then use the relatively large DNA molecules as templates for even smaller components such as carbon nanotubes or silicon nanowires," he said. The nanotubes or nanowires attach to the ends of the DNA strands in the pattern necessary to perform the desired tasks.

To be useful, self-assembling material must behave precisely and predictably, said Cha. Scientists are very familiar with DNA, which consists of specific chemical bases—such as guanine and cytosine—that behave dependably.

Because the technique works at nanoscale, IBM says, it could permit the manufacture of devices with circuit widths of one-eighth to one-tenth the 45 nanometers that traditional lithographic techniques can currently fabricate.

Smaller circuitry would let chip makers either pack more transistors onto processors, thereby making them more powerful, or make smaller chips with performance equal to today's larger versions.

The DNA-based process could occur in laboratories and thus could make chip manufacturing less expensive than current approaches, which require multibillion-dollar fabrication plants and expensive lithographic processing.

According to Cha, researchers are also working on ways to make

Introducing the New Sport of Speedcabling

A California artist and university faculty member has developed a new form of competition for technophiles called *speedcabling*, in which competitors see who can untie tangled Ethernet cables the fastest.

Steven Schkolne, who teaches at the California Institute of the Arts, developed speedcabling (www.speedcabling.org), which entails the untangling of Category 5 Ethernet cables that have been put in a clothes dryer for three minutes. He said the competition uses three cable lengths: 7, 14, and 21 feet.

"In 2-2-2 competition, two cables of each length are used. In 4-4-4 competition, there are four of each length," he explained. "As the sport grows, I hope to have 12-4-1 and 0-0-8 competitions."

To start a game, he said, a referee places a bundle of tangled cables on a table in front of competitors. When the referee signals, the contestants begin the untangling process. Players must separate each wire, hold them overhead as separation occurs, and place them on the floor. If a wire touches the floor before it has been pulled free, the competitor can continue but receives a 10-second penalty. The first contestant to successfully separate the entire bundle of wires wins.

There has been one competition so far, held with 20 participants recently at a Los Angeles art gallery. Musician and Web developer Matty Howell won and received a $50 gift certificate to a local restaurant.

Because the players competed against each other, no timekeeper was necessary. However, Schkolne estimated, the fastest time for the 2-2-2 competitions in the preliminary rounds was just under one minute and the fastest 4-4-4 time in the finals was about 1 minute, 50 seconds.

According to Schkolne, he plans at least two more speedcabling competitions at art galleries, one this month in Los Angeles and another in September in New York City. He said several people have contacted him about holding their own events. ■

the nanoscale wires that the new chips would need. In addition, she said, they want to improve yields and performance, as well as use DNA to separate and eliminate nanotubes that don't conduct electricity.

Moreover, scientists must improve the basic chip-making process to ensure that the nanotubes adhere to the DNA properly and in the correct orientation.

The technique may need 10 to 20 years of work before it can be used commercially, said Wallraff.

Researchers Use Software to Find Chip Flaws

Scientists have developed software that identifies problems in chips and recommends the best way to fix them. Developed at the University of Michigan, the software would help manufacturers cope with the growing number of bugs that take an increasing amount of time to fix as chips become more complex.

Processors now house a large number of transistors and perform a growing number of functions. This leads to more bugs and makes it difficult and time-consuming to identify them before vendors ship the chips, explained University of Michigan assistant professor Valeria Bertacco, who is working on the bug-finding software.

"As the complexity increases, so does the number of lines of code, which causes an explosion of bugs," said Gary Smith, chief analyst at the Gary Smith EDA consultancy.

This can delay the commercial release of chips and increase costs for manufacturers. Problems that occur after a processor ships can be particularly time-consuming and expensive to fix, Bertacco noted. Currently, though, fully debugging prototype chips can take up to a year, she said.

Because this would affect their budgets and market-related deadlines, chip makers usually fix only as many bugs as they can within a few months and then either ship their processors or cancel the project, according to Smith.

Manufacturers debug chips both before and after they are made, generally by designing and manufacturing special test boards and applying electric currents via the pins or internal nodes. Because of the analysis required to isolate and correct problems within the complex circuitry, the process relies on expensive logic analyzers to observe the internals.

The University of Michigan's FogClear software, run on an engineering workstation, could shorten the debugging process and reduce the number of prototypes and testing cycles vendors must conduct. It perhaps could even increase chips' reliability.

According to Bertacco, at any point in the chip-design flow, the software can examine two types of design-related errors: functional and electrical.

"A functional error is a bug in which the logic used to implement the design is incorrect in one or more circuit blocks," he explained. "An electrical error is one in which the circuit is functional but fails at the clock speed, voltage, and temperature intended for correct operation, typically because it doesn't finish evaluating within a clock cycle." Electrical errors tend to get worse as circuitry shrinks in size and performance demands grow.

The University of Michigan software uses mathematical techniques to examine the differences between the correct design and the actual circuit and find the precise location of bugs, said Bertacco. "Thus, the designer does not have to spend the effort required to wade through thousands of electrical inputs and millions of transistors to locate them."

"Our software can also narrow down the possible causes and develop fixes," he explained. "If we do not know that a chip has a bug, FogClear would not detect it by itself. However, when a bug has been observed, our software can locate and identify it."

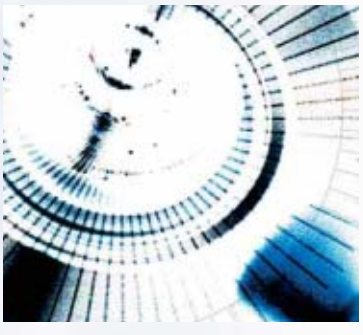
The software conducts simulations of possible solutions to find the most cost-effective design variation that will fix the bugs, sometimes in ways that may be counterintuitive or not obvious to engineers doing the work in traditional ways.

In case studies, the researchers automatically repaired about 70 percent of major problems and reduced the debugging time from weeks to days, according to University of Michigan associate professor Igor Markov.

COMPUTING PRACTICES

Using String Matching for Deep Packet Inspection

String matching has sparked renewed research interest due to its usefulness for deep packet inspection in applications such as intrusion detection, virus scanning, and Internet content filtering. Matching expressive pattern specifications with a scalable and efficient design, accelerating the entire packet flow, and string matching with high-level semantics are promising topics for further study.



Po-Ching Lin, Ying-Dar Lin, and Tsern-Huei Lee
National Chiao Tung University

Yuan-Cheng Lai
National Taiwan University of Science and Technology

A classical algorithm for decades, string matching has recently proven useful for deep packet inspection (DPI) to detect intrusions, scan for viruses, and filter Internet content. However, the algorithm must still overcome some hurdles, including becoming efficient at multigigabit processing speeds and scaling to handle large volumes of signatures.

Before 2001, researchers in packet processing were most interested in *longest-prefix matching* in the routing table on Internet routers and *multifield packet classification* in the packet header for firewalls and quality-of-service applications.[1] However, DPI for various signatures is now of greater interest.

Intrusion detection, virus scanning, content filtering, instant-messenger management, and peer-to-peer identification all can use string matching for inspection. Much work has been done in both algorithm design and hardware implementation to accelerate the inspection, reduce pattern storage space, and efficiently handle regular expressions.

According to our survey of recent publications about string matching from IEEE Xplore (http://ieeexplore.ieee.org) and the ACM digital library (http://portal.acm.org/dl.cfm), researchers formerly were more interested in pure algorithms for either theoretical interest or general applications, while algorithms for DPI have attracted more attention lately. Likewise, to meet the demand for higher processing speeds, researchers are focusing on hardware implementation in application-specific integrated circuits and field-programmable gate arrays, as well as parallel multiple processors. Since 2004, ACM and IEEE publications have featured 34 articles on ASICs and FPGAs compared to nine in the 1990s and nine again between 2000 and 2003. ACM and IEEE publications have published 10 articles on multiple processors since 2004, with 10 published during the 1990s, and three between 2000 and 2003.

0018-9162/08/$25.00 © 2008 IEEE

Published by the IEEE Computer Society

Characteristics of String-Matching Algorithms

Researchers can evaluate string-matching algorithms based on the following characteristics:

- *Number of searches*. Some applications, such as search engines, search the same text many times for different querying strings. Building an indexing data structure from the text in advance is therefore worthwhile to perform with the time complexity as low as O(*m*). In contrast, the applications in networking and biological sequences search throughout online text only once without the indexing structure, and the time complexity is linear in *n*.
- *Text compression*. Some algorithms can directly search the compressed text with minimum (or no) decompression, while others scan over the plaintext.
- *Matching criteria*. A match can be exact or approximate. An exact match demands that the pattern and matched text be identical, while an approximate match allows a limited number of differences between them.
- *Time complexity*. Some algorithms have deterministic linear time complexity, while others can have sublinear time complexity by skipping characters not in a match. The latter might be faster on average, but not in the worst case.
- *Number of patterns*. An algorithm can scan one pattern or multiple patterns simultaneously.
- *Expressiveness in pattern specifications*. Pattern specifications range from fixed strings to regular expressions in various syntax options. In addition to primitive notations of alternation, catenation, and Kleene closure, extensions in the syntax of regular expressions include the Unix representations, the extended forms in Posix 1003.2, and Perl Compatible Regular Expression.[1] An increasing number of signatures is specified in regular expressions for their expressiveness.

Reference

1. J. Friedl, *Mastering Regular Expressions*, 3rd ed., O'Reilly, 2006.

DEVELOPMENT OF STRING-MATCHING ALGORITHMS

The "Characteristics of String-Matching Algorithms" sidebar summarizes the characterization and classification of these algorithms. In DPI, *automaton*, *heuristic*, or *filtering* approaches are common. Bit parallelism techniques are often used in computational biology, but rarely in networking. We assume the text length to be *n* characters and the pattern length (or the shortest length in the case of multiple patterns) to be *m* characters.

Automaton-based approach

An automaton-based approach tracks partially matched patterns in the text by state transition in either a *deterministic finite automaton* or a *nondeterministic finite automaton* implementation that accepts the strings in the pattern set. A DFA implementation generally has lower time complexity but demands more space for pattern storage, while an NFA implementation is the opposite.[2] The automaton-based approach is popular in DPI for two reasons:

- The deterministic execution time guarantees the worst-case performance even when algorithmic attacks deliberately generate text to exploit an algorithm's worst-case scenario.
- Building an automaton to accept regular expressions is systematic and well-studied.

Given the wide data bus of 32 or 64 bits in modern computer architectures, tracking the automaton with one input character at a time poorly utilizes the bus width and degrades throughput. Extending the transition table to store transitions for two or more characters is plausible, but it's impractical without proper table compression. Storing a large pattern set is also memory-consuming due to the large number of states. Recent research therefore tries to reduce data-structure space and simultaneously inspect multiple characters. A compact data structure in a software implementation also increases performance due to the good cache locality.

Reducing sparse transition tables. A transition table is generally sparse because most states, particularly those away from the root state, have only a few valid next states. We can compress the table by storing only links to valid next states after one or more input characters and failure links of each state. We also can store the state transition table, the failure links, and the lists of matched patterns in the final states separately in a software implementation to improve the cache locality during tracking.

Snort (www.snort.org), a popular open source intrusion-detection package, has carefully tuned the data structure in this way to improve cache performance. The latest revision uses a basic NFA construction as the default search method (src/sfutil/bnfa_search.c in the source tree of Snort 2.6.1).

Reducing transitions. With the extended ASCII alphabet, an automaton has a maximum of 256 transitions from a state. Splitting an automaton into several smaller ones at the bit level can reduce the number of transitions. For example, suppose the automaton is split into eight, and then one

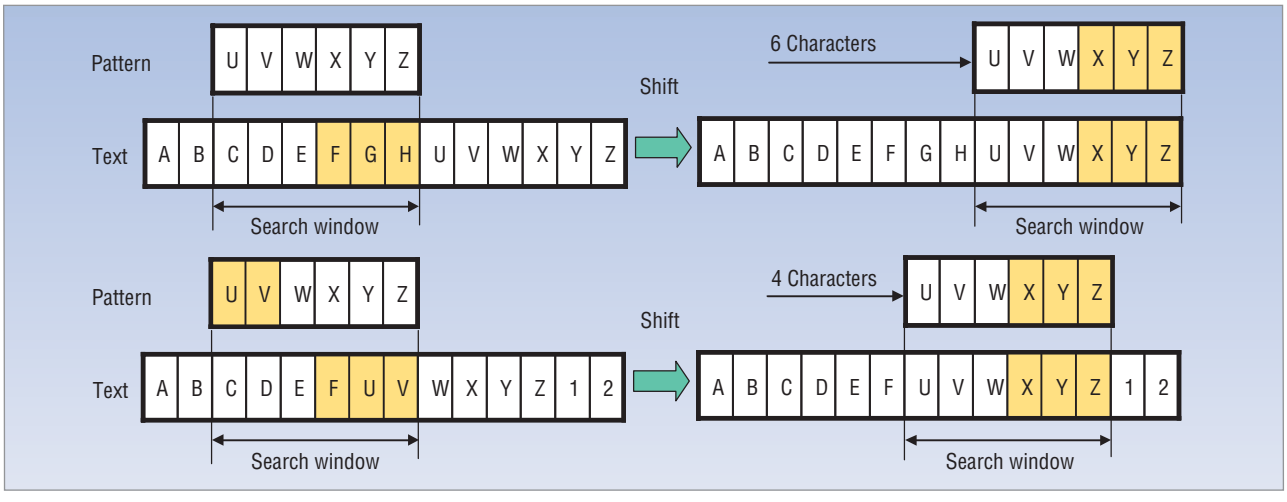


Figure 1. A simple heuristic demonstrates one pattern to visualize why skipping is efficient.

automaton is fed with b7, one is fed with b6, and so on, where b7b6 … b0 denotes the eight bits of the input characters.

This method is implemented in hardware to efficiently track these automata in parallel. These automata are compact because each state has at most two valid transitions for input bits of 0 and 1. Expanding the automata to read multiple characters at a time is also facilitated due to the significantly reduced fanout—in this example, perhaps only 16 valid transitions from a state for four input characters at once.

Because groups of states in an automaton generally have common outgoing transitions that lead to the same set of states for the same input characters, the delayed input DFA (D^2FA) method can effectively reduce these common transitions. A state in a group can maintain only its unique transitions and make a default transition to the state in the group responsible for the common transitions. This method claims to reduce more than 95 percent of transitions for regular expressions on practical products and tools.

Hash tables. A hash table can store the transitions from the states in an automaton to their corresponding valid next states (or failure links) after several input characters. Tracking multiple characters at a time becomes a table lookup. Because only a few input characters can lead to valid next states, the hash table size is still manageable. A filtering approach can weed out unsuccessful searches in the hash table to further accelerate this method. Ternary content addressable memory is an alternative for a table lookup.

Rewriting and grouping. Some combinations of wildcards and repetitions in regular expressions will generate a complex automaton that grows exponentially.[2] It's possible to rewrite the regular expressions to simplify the automaton because we don't have to find every match in the text in some networking applications. Finding an appearance of certain signatures suffices. For example, every string *s* identified by "ab+" (+ denotes one or more) can be identified by "ab" as s itself or a prefix of *s*, so reporting a match against "ab" is sufficient to report an appearance of "ab+".

Furthermore, compiling all the regular expressions in a single automaton can result in a complex automaton. In a multiprocessing environment, we can group regular expressions in separate automata according to the interaction between them. For example, grouping regular expressions sharing the same prefix can merge common states of the prefix and save the storage. An individual processing unit then processes each automaton.

Hardwiring regular expressions. Some designs use building blocks on the FPGA to match patterns from fixed strings to regular expressions. The implementation typically prefers an NFA to a DFA because an NFA has fewer states, and the inherent concurrency of hardware can easily track multiple active states.

A few techniques can reduce the area cost of building blocks. For example, identical substrings from different patterns can share common blocks. Specific hardware logics can directly handle notations in regular expressions such as class of characters, repetitions, wildcard characters, and so on.

Heuristic-based approach

A heuristic-based approach can skip characters not in a match to accelerate the search according to certain heuristics. During the search, a search window of *m* characters covers the text under inspection and slides throughout the text. A heuristic can check a block of characters in the window suffix for its appearance in the patterns. It determines whether a suspicious match occurs and moves to the next window position if not.

Shift values. Because the positions or shift values corresponding to possible blocks are computed and stored in a table beforehand, a table lookup drives shifting the search window in the search stage. Figure 1 illustrates a simple but generic heuristic for only one pattern to visu-

alize why skipping is efficient. In the upper part, because "FGH" is not a substring of the pattern and its suffix is not a prefix of the pattern, shifting the search window by $m = 6$ characters without examining the remaining characters in the window won't miss a match. After the shift, "XYZ" becomes the suffix of both the pattern and the window, meaning a suspicious match occurs. The entire window is then verified, and a match is found.

However, if a suffix of the block is the prefix of some pattern, the shift value should be less than m because the suffix might be the prefix of that pattern after the shift. Figure 1 illustrates this case. We can easily extend this heuristic to handle patterns shorter than the block size. If a short pattern is a substring of the block, looking up the block can claim a match. In addition to the heuristic for matching fixed strings, Gonzalo Navarro and Mathieu Raffinot presented a heuristic to skip text characters for regular-expression matching.[3]

Ideally, most shift values are equal or close to the pattern length m, so the time complexity is sublinear: $O(n/m)$. However, the time complexity could be $O(nm)$ in the worst case, in which the system examines each entire search window after a shift of one character. Although methods exist to guarantee the linear worst-case time complexity for a single fixed string or regular expression,[3,4] they're rarely adopted in DPI, which looks for multiple patterns. Finding an inexpensive solution to achieve sublinear time while ensuring the performance in the worst case would be an interesting challenge to the research community.

Due to their vulnerability to algorithmic attacks, heuristic-based algorithms usually are not preferable for network-security applications.

Due to their vulnerability to algorithmic attacks, heuristic-based algorithms usually are not preferable for network-security applications because an attacker might manipulate the text to degrade performance. Because applications such as Snort have short patterns of only one or two characters, the small value of m makes the advantage of skipping marginal. Nevertheless, for applications with long patterns such as the signatures of nonpolymorphic viruses in the ClamAV antivirus package (www.clamav.net), skipping over the text is still helpful.

Implementation details. Block size, mapping from the blocks to derive shift values, and other implementation details can significantly affect practical performance. When choosing proper parameter values, considerations include the size of the pattern set, block distribution, cache locality, and verification frequency. For example, a large block has fewer chances to appear in the patterns, resulting in less frequent verification.

However, a large block also generally implies a large table that stores shift values mapped from a large number of possible blocks, resulting in reduced cache locality. Careful experimenting should properly tune these parameters. When suspicious matches frequently appear, implementing an efficient method to identify the matched pattern is also important.

Because the block distribution might be nonuniform in practice, some blocks might appear more often than expected, shortening the shift distance and increasing the verification frequency. Checking the matches in additional blocks within the search window can reduce the frequency of verification.

Using a heuristic similar to that in Figure 1 to look for the longest suffix of the search window that's also a substring of some pattern might result in long shift distance even with nonuniform block distribution. However, longer shift distance doesn't always imply better performance. The overhead due to the extra examination must be carefully evaluated.

Filtering-based approach

A filtering-based approach searches text for necessary pattern features and quickly excludes the content not containing those features. For example, if a packet misses any two-character substrings of a pattern, the packet must not have that pattern. Because the efficiency relies on assuming that the signatures rarely appear in normal packets, this approach might suffer from algorithmic attacks if the attacker carefully manipulates the text.

Text filtering. A common method of text filtering is the Bloom filter, characterized by a bit vector and a set of k hash functions $h_1, h_2, \ldots, h_k$ mapped to that vector. When multiple patterns are present, the patterns of a specific length are stored in a separate Bloom filter by setting to 1 the bits the patterns' hash values address. The search queries the set of Bloom filters by mapping the substrings in the text under inspection to them with the same set of hash functions. Specifically, a substring x under inspection is mapped to the Bloom filter storing the patterns of length $|x|$.

If one of the bits in $h_1(x), h_2(x), \ldots, h_k(x)$ isn't set to 1, x certainly isn't in the pattern set; otherwise, x might be in the pattern set, and we must further verify the match. The uncertainty comes from different patterns setting checked bits. The false-positive rate is a function of the bit-vector size, the number of patterns, and the number of hash functions. Properly controlling these parameters can reduce the false-positive rate.

Parallel queries. Parallel queries to the Bloom filters generally are implemented in hardware for efficiency, but efficient software implementation of sequential queries is also possible. For example, the implementation can sequentially query with a set of hash functions, from simple to complex ones, to look for pattern prefixes of a certain length and verify a match if a prefix is found. The simple hash functions are designed to be

rapidly computed and can filter most of the text, so the search is still fast.

If there is a wide range of pattern lengths, there might be many Bloom filters because each length requires one. One solution is to limit the maximum pattern length allowed and break a long pattern into short ones. If all substrings of a long pattern appear contiguously and in order, that pattern is present.

The filtering-based approach doesn't directly support some notations in regular expressions such as wildcards and repetitions. An indirect solution is to extract the necessary substrings from the regular expressions, searching for them and verifying the match if these substrings appear. For example, ClamAV divides the signatures of polymorphic viruses into ordered parts (substrings of the signatures) and tracks the orders and positions of these parts (with a variant of the Aho-Corasick algorithm) in the text to determine whether a signature occurs. Table 1 summarizes the key methods as well as the pros and cons of each.

Table 1. Summary of approaches to string matching for DPI.

Automaton-based
Pros: Deterministic linear execution time, direct support of regular expressions
Cons: Might consume much memory without compressing data structure
1. Rewrite and group regular expressions
2. Reduce number of transitions (D^2FA)
3. Hardwire regular expressions on FPGA
4. Track a DFA that accepts the patterns (Aho-Corasick)
5. Reduce sparse transition table (Bitmap-AC, BNFA in Snort)
6. Reduce fanout from the states (split automata)
7. Track multiple characters at a time in an NFA (JACK-NFA)
Heuristic-based
Pros: Can skip characters not in a match, sublinear execution time on average
Cons: Might suffer from algorithmic attacks in the worst case
1. Get shift distance using heuristics based on the automaton that recognizes the reverse prefixes of a regular expression (RegularBNDM)
2. Get shift distance from fixed block in suffix of search window (Wu-Manber)
3. Get shift distance from the longest suffix of search window (BG)
Filtering-based
Pros: Memory efficient in the bit vectors
Cons: Might suffer from algorithmic attacks in the worst case
1. Extract substrings from regular expressions, filter text with them (MultiFactRE)
2. Filter with a set of Bloom filters for different pattern lengths
3. Filter with a set of hash functions sequentially (Hash-AV)

CURRENT TRENDS IN DPI

Matching expressive pattern specifications with a scalable and efficient design, accelerating the entire packet flow, and string matching with high-level semantics are promising topics for further study.

Matching expressive pattern specifications

Expressive pattern specifications, such as regular expressions, can accurately define the signatures. Efficient solutions to matching regular expressions in DPI are therefore attracting considerable interest. Joao Bispo and his colleagues compared several designs for regular expression matching.[5] Most of these designs can perform regular expression matching on the order of several gigabits per second.

Commercial products, including the Cavium Octeon MIPS64 processor family (www.cavium.com/OCTEON_MIPS64.html), SafeNet Xcel 4850 (http://cn.safenet-inc.com/products/safenetchips/index.asp), and Tarari RegEx5 content processor (www.lsi.com/documentation/networking/tarari_content_processors/Tarari_RegEx_Whitepaper.pdf) all claim to support regular expression matching at gigabit rates. String matching, a problem once believed to be a bottleneck, has become less critical given the latest advances.

Most existing research aims at intrusion-detection applications, especially Snort, which has thousands of signatures, but antivirus applications such as ClamAV claim a signature set of more than 180,000 patterns to date. We believe a more scalable and efficient design for matching a huge set of expressive patterns deserves further study. Moreover, some patterns might belong to only a specific protocol or file type, and some are significant only when they appear in specified positions of the text.

Rather than assuming a simple model of searching for the whole pattern set throughout the entire text, a design can optimize the performance of additional information. An efficient software implementation for these cases is also desirable, since hardware accelerators aren't always affordable in practical applications.

Accelerating packet content processing

Although numerous research efforts have been dedicated to string matching, packet processing in DPI involves even more effort. Vern Paxson and colleagues described the insufficiency of string matching in intrusion detection due to its stateless nature[6] and envisioned a framework of architecture that attempts to exploit the parallelism in network analysis and intrusion detection for acceleration.

Similarly, virus-scanning applications might reassemble packets, unpack and decompress file archives, and handle character encoding before scanning a transferred file. Accelerating only one stage is insufficient due to Amdahl's law. Meeting the high-speed demand in

networking applications requires an integrated architecture with hardware-supported functions.

Commercial products are on this track. For example, the Cavium Octeon MIPS64 processor family includes a TCP unit, a compression/decompression engine, and 16 regular expression engines on a single chip, and claims performance of up to 5 Gbps for regular expression matching plus compression/decompression.

Parsing content in high-level semantics

String matching in network applications might refer to contextual information parsed from high-level semantics.[7] For example, some patterns are significant only within the uniform resource indicators. Spam and Web filtering also demand high-level semantics to analyze the content, as does XML processing.[8] String matching with high-level semantic extraction and analysis from the text is therefore beneficial.

For example, because the Tarari random access XML content processor (www.lsi.com/documentation/networking/tarari_content_processors/Tarari_RAX_Whitepaper.pdf)can help applications directly access information inside XML documents without parsing, it accelerates XML applications significantly. The acceleration of semantic extraction from the text (perhaps with hardware support) and matching patterns with the semantic contextual information is worth studying, and will be helpful for numerous network applications.

Despite existing research, the study of string matching for DPI still has a way to go in the near future. In addition to the growing set of increasingly expressive patterns that makes scalability a challenge, matching with semantically contextual information also complicates the traditional model of string matching that looks for patterns in the text. Dealing with this complication is particularly significant because many existing efforts still use the traditional model to develop their solutions. After all, DPI applications rely on the packet content semantics to make an effective decision. These complexities require expending more effort to develop a scalable, efficient, and effective string-matching solution for DPI applications.

References

1. P. Gupta and N. McKeown, "Algorithms for Packet Classification," *IEEE Network*, vol. 15, no. 2, Mar./Apr. 2001, pp. 529-551.
2. F. Yu et al., "Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection," *Proc. Symp. Architectures Networking and Comm. Systems* (ANCS 06), ACM Press, 2006, pp. 93-102.
3. G. Navarro and M. Raffinot, "New Techniques for Regular Expression Searching," *Algorithmica*, Springer-Verlag, vol. 41, no. 2, 2004, pp. 89-116.
4. Z. Galil, "On Improving the Worst-Case Running Time of the Boyer-Moore String Searching Algorithm," *Comm. ACM*, vol. 22, no. 9, 1979, pp. 505-508.
5. J. Bispo et al., "Regular Expression Matching for Reconfigurable Packet Inspection," *Proc. IEEE Int'l Conf. Field-Programmable Technology* (FPT 06), IEEE Press, 2006, pp. 119-126.
6. V. Paxson et al., "Rethinking Hardware Support for Network Analysis and Intrusion Prevention," *Proc. Usenix Workshop Hot Topics in Security*, Usenix, 2006, pp. 63-68; http://imawhiner.com/csl/usenix/06hotsec/tech/paxson.html.
7. R. Sommer and V. Paxson, "Enhancing Byte-Level Network Intrusion Detection Signatures with Context," *Proc. ACM Computer and Comm. Security* (CCS 03), ACM Press, 2003, pp. 262-271.
8. T.J. Green et al., "Processing XML Streams with Deterministic Automata and Stream Indexes," *ACM Trans. Database Systems*, Dec. 2004, pp. 752-788.

Po-Ching Lin *is a PhD candidate in the Department of Computer Science at the National Chiao Tung University in Hsinchu, Taiwan. His research interests include network security, string-matching algorithms, hardware-software codesign, content networking, and performance evaluation. Lin received an MS in computer science from the National Chiao Tung University. He is a student member of the IEEE. Contact him at pclin@cis.nctu.edu.tw.*

Ying-Dar Lin *is a professor in the Department of Computer Science at National Chiao Tung University. His research interests include design, analysis, implementation, and benchmarking of network protocols and algorithms; wire-speed switching and routing; and embedded hardware-software codesign. Lin received a PhD in computer science from the University of California, Los Angeles. He is a senior member of the IEEE and a member of the ACM. Contact him at ydlin@cs.nctu.edu.tw.*

Yuan-Cheng Lai *is an associate professor in the Department of Information Management at National Taiwan University of Science and Technology in Taipei, Taiwan. His research interests include high-speed networking, wireless network and network performance evaluation, Internet applications, and content networking. Lai received a PhD in computer science from the National Chiao Tung University. Contact him at laiyc@cs.ntust.edu.tw.*

Tsern-Huei Lee *is a professor in the Department of Communications Engineering at the National Chiao Tung University. His research interests include high-speed networking, broadband switch systems, network flow control, data communications, and string-matching algorithms. Lee received a PhD in electrical engineering from the University of Southern California. Contact him at thlee@banyan.cm.nctu.edu.tw.*

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE: www.computer.org

OMBUDSMAN: To check membership status or report a change of address, call the IEEE Member Services toll-free number, +1 800 678 4333 (US) or +1 732 981 0060 (international). Direct all other Computer Society-related questions—magazine delivery or unresolved complaints—to help@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION: To obtain more information on any of the following, contact Customer Service at +1 714 821 8380 or +1 800 272 6657:

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

Computer. The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals. The society publishes 14 magazines, 12 transactions, and one letters. Refer to membership application or request information as noted above.

Conference Proceedings & Books. Conference Publishing Services publishes more than 175 titles every year. CS Press publishes books in partnership with John Wiley & Sons.

Standards Working Groups. More than 150 groups produce IEEE standards used throughout the world.

Technical Committees. TCs provide professional interaction in over 45 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education. The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications. The society offers two software developer credentials. For more information, visit www.computer.org/certification.

IEEE

EXECUTIVE COMMITTEE

President: Rangachar Kasturi*
President-Elect: Susan K. (Kathy) Land, CSDP*
Past President: Michael R. Williams*
VP, Electronic Products & Services: George V. Cybenko (1st VP)*
Secretary: Michel Israel (2nd VP)*
VP, Chapters Activities: Antonio Doria†
VP, Educational Activities: Stephen B. Seidman†
VP, Publications: Sorel Reisman†
VP, Standards Activities: John W. Walz†
VP, Technical & Conference Activities: Joseph R. Bumblis†
Treasurer: Donald F. Shafer*
2008–2009 IEEE Division V Director: Deborah M. Cooper†
2007–2008 IEEE Division VIII Director: Thomas W. Williams†
2008 IEEE Division VIII Director-Elect: Stephen L. Diamond†
***Computer* Editor in Chief:** Carl K. Chang†

** voting member of the Board of Governors* *† nonvoting member of the Board of Governors*

BOARD OF GOVERNORS

Term Expiring 2008: Richard H. Eckhouse; James D. Isaak; James Moore, CSDP; Gary McGraw; Robert H. Sloan; Makoto Takizawa; Stephanie M. White
Term Expiring 2009: Van L. Eden; Robert Dupuis; Frank E. Ferrante; Roger U. Fujii; Ann Q. Gates, CSDP; Juan E. Gilbert; Don F. Shafer
Term Expiring 2010: André Ivanov; Phillip A. Laplante; Itaru Mimura; Jon G. Rokne; Christina M. Schober; Ann E.K. Sobel; Jeffrey M. Voas

EXECUTIVE STAFF

Executive Director: Angela R. Burgess
Director, Governance, & Associate Executive Director: Anne Marie Kelly
Associate Publisher: Dick Price
Director, Membership Development: Violet S. Doan
Director, Finance & Accounting: John Miller
Director, Information Technology & Services: Neal Linson

COMPUTER SOCIETY OFFICES

Washington Office. 1828 L St. N.W., Suite 1202, Washington, D.C. 20036-5104
Phone: +1 202 371 0101 • Fax: +1 202 728 9614
Email: hq.ofc@computer.org

Los Alamitos Office. 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314
Phone: +1 714 821 8380
Email: help@computer.org
Membership & Publication Orders:
Phone: +1 800 272 6657 • Fax: +1 714 821 4641
Email: help@computer.org

Asia/Pacific Office. Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan
Phone: +81 3 3408 3118 • Fax: +81 3 3408 3553
Email: tokyo.ofc@computer.org

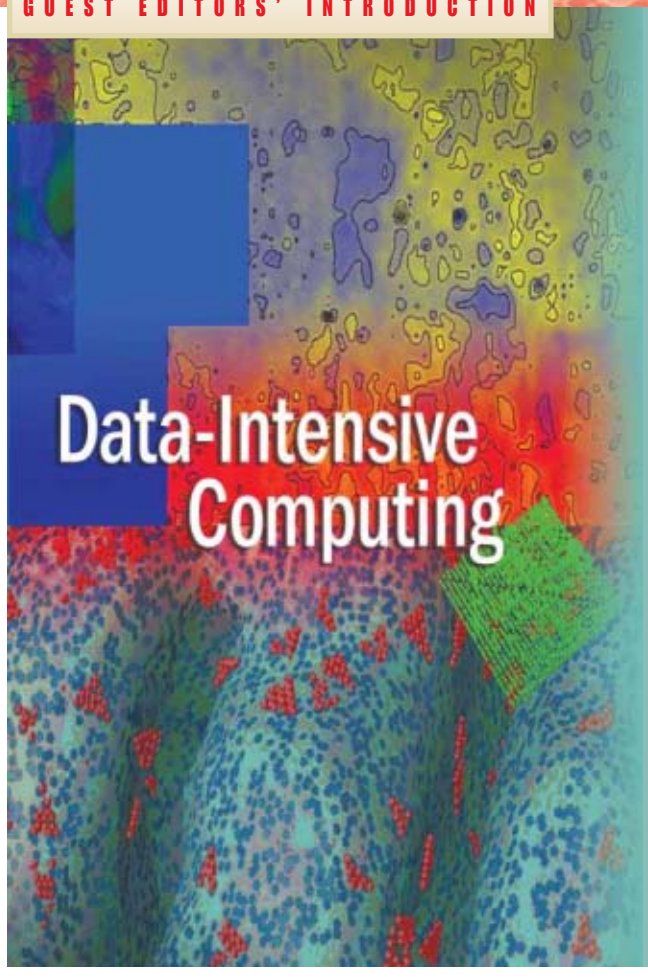
IEEE OFFICERS

President: Lewis M. Terman
President-Elect: John R. Vig
Past President: Leah H. Jamieson
Executive Director & COO: Jeffry W. Raynes
Secretary: Barry L. Shoop
Treasurer: David G. Green
VP, Educational Activities: Evangelia Micheli-Tzanakou
VP, Publication Services & Products: John Baillieul
VP, Membership & Geographic Activities: Joseph V. Lillie
VP, Standards Association Board of Governors: George W. Arnold
VP, Technical Activities: J. Roberto B. deMarca
IEEE Division V Director: Deboarh M. Cooper
IEEE Division VIII Director: Thomas W. Williams
President, IEEE-USA: Russell J. Lefevre

Next Board Meeting: 16 May 2008, Las Vegas, NV, USA

revised 3 Mar. 2008

GUEST EDITORS' INTRODUCTION



Data-Intensive Computing in the 21st Century

Ian Gorton, Pacific Northwest National Laboratory
Paul Greenfield, CSIRO
Alex Szalay, Johns Hopkins University
Roy Williams, Caltech

The deluge of data that future applications must process—in domains ranging from science to business informatics—creates a compelling argument for substantially increased R&D targeted at discovering scalable hardware and software solutions for data-intensive problems.

In 1998, William Johnston delivered a paper at the 7th IEEE Symposium on High-Performance Distributed Computing[1] that described the evolution of data-intensive computing over the previous decade. While state of the art at the time, the achievements described in that paper seem modest in comparison to the scale of the problems researchers now routinely tackle in present-day data-intensive computing applications.

More recently, others, including Tony Hey and Anne Trefethen,[2] Gordon Bell and colleagues,[3] and Harvey Newman and colleagues[4] have described the magnitude of the data-intensive problems that the e-science community faces today and in the near future. Their descriptions of the data deluge that future applications must process, in domains ranging from science to business informatics, create a compelling argument for R&D to be targeted at discovering scalable hardware and software solutions for data-intensive problems. While petabyte datasets and gigabit data streams are today's frontiers for data-intensive applications, no doubt 10 years from now we'll fondly reminisce about problems of this scale and be worrying about the difficulties that looming exascale applications are posing.

Fundamentally, data-intensive applications face two major challenges:

- managing and processing exponentially growing data volumes, often arriving in time-sensitive streams from arrays of sensors and instruments, or as the outputs from simulations; and
- significantly reducing data analysis cycles so that researchers can make timely decisions.

There is undoubtedly an overlap between data- and compute-intensive problems. Figure 1 shows a simple diagram that can be used to classify the application space between these problems.

Purely data-intensive applications process multiterabyte- to petabyte-sized datasets. This data commonly comes in several different formats and is often distributed across multiple locations. Processing these datasets

Published by the IEEE Computer Society

typically takes place in multistep analytical pipelines that include transformation and fusion stages. Processing requirements typically scale near-linearly with data size and are often amenable to straightforward parallelization. Key research issues involve data management, filtering and fusion techniques, and efficient querying and distribution.

Data/compute-intensive problems combine the need to process very large datasets with increased computational complexity. Processing requirements typically scale superlinearly with data size and require complex searches and fusion to produce key insights from the data. Application requirements may also place time bounds on producing useful results. Key research issues include new algorithms, signature generation, and specialized processing platforms such as hardware accelerators.

We view data-intensive computing research as encompassing the problems in the upper two quadrants in Figure 1. The following are some applications that exhibit these characteristics.

Astronomy. The Large Synoptic Survey Telescope (LSST; www.lsst.org) will generate several petabytes of new image and catalog data every year. The Square Kilometer Array (SKA; www.skatelescope.org) will generate about 200 Gbytes of raw data per second that will require petaflops (or possibly exaflops) of processing to produce detailed radio maps of the sky. Processing this volume of data and making it available in a useful form to the scientific community poses highly challenging problems.

Cybersecurity. Anticipating, detecting, and responding to cyberattacks requires intrusion-detection systems to process network packets at gigabit speeds. Ideally, such systems should provide actionable results in seconds to minutes, rather than hours, so that operators can defend against attacks as they occur.

Social computing. Sites such as the Internet Archive (www.archive.org) and MySpace (www.myspace.com) store vast amounts of content that must be managed, searched, and delivered to users over the Internet in a matter of seconds. The infrastructure and algorithms required for websites of this scale are challenging, ongoing research problems.

DATA-INTENSIVE COMPUTING CHALLENGES

The breakthrough technologies needed to address many of the critical problems in data-intensive computing will come from collaborative efforts involving several disciplines, including computer science, engineering, and mathematics. The following list shows some of the advances that will be needed to solve the problems faced by data-intensive computing applications:

- new algorithms that can scale to search and process massive datasets;
- new metadata management technologies that can scale to handle complex, heterogeneous, and distributed data sources;
- advances in high-performance computing platforms to provide uniform high-speed memory access to multiterabyte data structures;
- specialized hybrid interconnect architectures to process and filter multigigabyte data streams coming from high-speed networks and scientific instruments and simulations;
- high-performance, high-reliability, petascale distributed file systems;
- new approaches to software mobility, so that algorithms can execute on nodes where the data resides when it is too expensive to move the raw data to another processing site;
- flexible and high-performance software integration technologies that facilitate the plug-and-play integration of software components running on diverse computing platforms to quickly form analytical pipelines; and
- data signature generation techniques for data reduction and rapid processing.

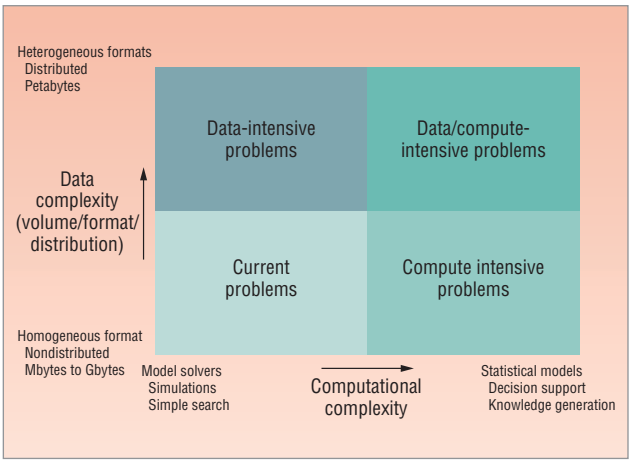


Figure 1. Research issues. Data-intensive computing research encompasses the problems in the upper two quadrants.

IN THIS ISSUE

This special issue on data-intensive computing presents five articles that address some of these challenges.

In "Quantitative Retrieval of Geophysical Parameters Using Satellite Data," Yong Xue and colleagues discuss the remote sensing information service grid node, a tool for processing satellite imagery to deal with climate change.

In "Accelerating Real-Time String Searching with Multicore Processors," Oreste Villa, Daniele Paolo Scarpazza, and Fabrizio Petrini present an optimization strategy for a popular algorithm that performs exact string matching against large dictionaries and offers solutions to alleviate memory congestion.

"Analysis and Semantic Querying in Large Biomedical Image Datasets" by Joel Saltz and colleagues describes a set of techniques for using semantic and spatial information to analyze, process, and query large image datasets.

"Hardware Technologies for High-Performance Data-Intensive Computing" by Maya Gokhale and colleagues offers an investigation into hardware platforms suitable for data-intensive systems.

In "ProDA: An End-to-End Wavelet-Based OLAP System for Massive Datasets," Cyrus Shahabi, Mehrdad Jahangiri, and Farnoush Banaei-Kashani describe a system that employs wavelets to support exact, approximate, and progressive OLAP queries on large multidimensional datasets, while keeping update costs relatively low.

We hope you will enjoy reading these articles and that this issue will become a catalyst for drawing together the multidisciplinary research teams needed to address our data-intensive future.

References

1. W. Johnston, "High-Speed, Wide Area, Data-Intensive Computing: A Ten-Year Retrospective," *Proc. 7th IEEE Symp. High-Performance Distributed Computing*, IEEE Press, 1998, pp. 280-291.
2. T. Hey and A. Trefethen, "The Data Deluge: An e-Science Perspective;" www.rcuk.ac.uk/cmsweb/downloads/rcuk/research/esci/datadeluge.pdf.
3. G. Bell, J. Gray, and A. Szalay, "Petascale Computational Systems," *Computer*, Jan. 2006, pp. 110-112.
4. H.B. Newman, M.H. Ellisman, and J.A. Orcutt, "Data-Intensive E-Science Frontier Research," *Comm. ACM*, Nov. 2003, pp. 68-77.

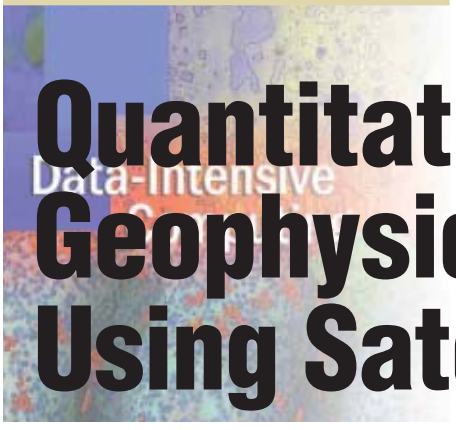
Ian Gorton *is the chief architect for Pacific Northwest National Laboratory's Data-Intensive Computing Initiative. His research interests include software architectures and middleware technologies. He received a PhD in computer science from Sheffield Hallam University. Gorton is a member of the IEEE Computer Society. Contact him at ian.gorton@pnl.gov.*

Paul Greenfield *is a research scientist in Australia's Commonwealth Scientific and Industrial Research Organisation. His research interests are distributed applications, the analysis of genetic sequence data, and computer system performance. He received an MSc in computer science from the University of Sydney. Greenfield is a member of the IEEE and the ACM. Contact him at paul.greenfield@csiro.au.*

Alex Szalay *is the Alumni Centennial Professor in the Department of Physics and Astronomy, also in the Department of Computer Science at the Johns Hopkins University. His research interests include large spatial databases, pattern recognition and classification problems, theoretical astrophysics, and galaxy evolution. He received a PhD in physics from the Eötvös University in Hungary. Contact him at szalay@jhu.edu.*

Roy Williams *is a senior scientist at the Center for Advanced Computing at Caltech. His research interests are astronomical transients, virtual observatory, and big data. He received a PhD in physics from Caltech. He is a member of the IEEE and the AAS. Contact him at cacr.caltech.edu.*

COVER FEATURE



Quantitative Retrieval of Geophysical Parameters Using Satellite Data

Yong Xue, Wei Wan, Yingjie Li, Jie Guang, Linyian Bai, Ying Wang, and Jianwen Ai
State Key Laboratory of Remote Sensing Science

The remote sensing information service grid node (RSIN) is a tool for dealing with climate change and quantitative environmental monitoring. Based on the high-throughput computing grid, RSIN enables a workflow management system for data placement. The accompanying unified data-and-computation-schedule algorithm helps load balancing between and within workflow steps.

Because it provides information that is both timely and global, satellite remote sensing is an efficient way to monitor aerosol properties, which are believed to be important for understanding the impact of aerosol radiative forcing on climate change. A reliable atmospheric remote sensing monitor relies on physical or statistical models having parameters that must be retrieved quantitatively. However, quantitative retrieval is a data-intensive application. High-resolution, wide-range, and long-duration observations produce several terabytes of data each day. Processing such massive volumes of multisensor and multitemporal data into scientific aerosol products involves addressing several computational problems. As a typical example, we consider the *moderate resolution imaging spectroradiometer* (MODIS) that the Earth Observing System sensors use to gauge computational and storage requirements, since researchers have widely used data from the instrument to study the atmosphere globally.

Processing Level 1B data for Level 2 aerosol products[1] for a single day requires 30 MODIS image granules of 13 Gbytes total to achieve full coverage of China's main land surface. The total storage requirements for such data for 30 days would be more than 4 terabytes.

This example shows that we need new processing powers to retrieve routine and timely information from satellite imagery. In addition, remote-sensing quantitative retrieval applications usually involve data processing and inversion computation. The tasks in the task stream might have vastly different complexities, data transfer demands, and execution times, imposing a mixed workload that interleaves massively parallel tasks that require large throughput with short tasks that require fast response.

GRIDLOCK

Although computational grid[2] use seems to offer the potential for enhanced remote-sensing applications, the scope of this potential remains nearly unexplored. Data independence poses one limitation to the current grid systems' handling of data-intensive computing. Existing systems closely couple data movement and computation, executing with stage-in, computation, and stage-out routines.

That data movement is usually unscheduled poses another limitation. Currently, developers perform data placement activities in the grid either manually or by simple command scripts. Thus, many applications restrict the number of concurrent jobs in a certain stage because those jobs can open only so many database connections without overwhelming the server or confronting space limitations in the shared fileserver.

Performance bottleneck

Users can avoid these pitfalls by introducing artificial dependencies between jobs.[3] However, in this case, a failure of one pipeline might prevent a set of other

0018-9162/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

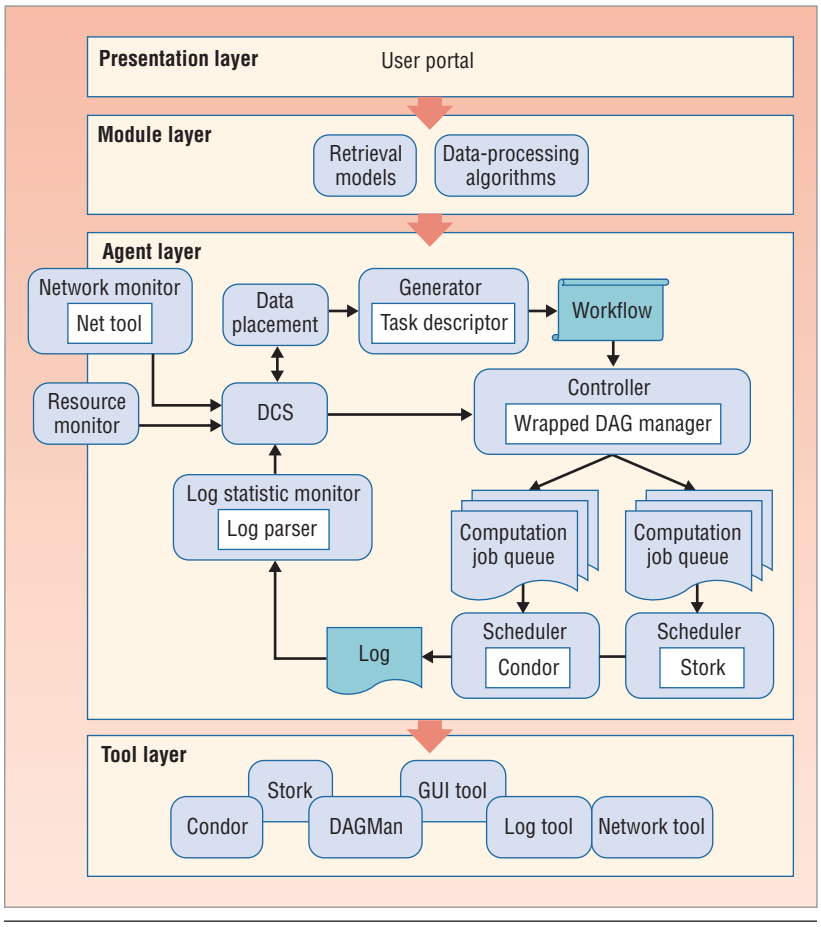


Figure 1. RSIN framework layered architecture. By adapting grid computing, the framework supports application execution for remote-sensing quantitative retrieval. A layered design in which each layer adds functionality leads to a reliable service that achieves this goal.

independent pipelines from executing because of the artificial dependency, resulting in suboptimal throughput.

Second, this limitation can result in increased execution time for a large mixed-load workflow, where the data transfer time to computation time ratio is often high because many short tasks requiring large input data exist. Without scheduled data movement, when the system allocates each task for execution on a computational node, current systems fail to prepare input files to be available to that node before computation can begin and stage its output files efficiently. Thus the cost of data transfer becomes a performance bottleneck.

The third limitation arises from the inability to dynamically balance load-sharing across heterogeneous resources. For heterogeneous grids, data transfer ability and pipeline type can differ vastly from each other. Current systems have difficulty choosing the appropriate pipeline dynamically across domains. In addition, they do not schedule data-transfer and computation tasks in a unified framework. When systems choose a compute resource for a task, the scheduling policy usually only considers computation requirements such as available CPUs, memory, and hard-disk space.

GEON

Several groups have addressed issues related to geoscience research in grid environments. The GEOsciences Network (GEON; www.geongrid.org) is a domain-specific grid of clusters for Earth sciences. GEON provides a standardized base software stack across all sites to ensure interoperability while providing structures that allow local customization.

The TeraGrid Geographic Information Science Gateway (GISolve; https://gisolve-portal.ncsa.uiuc.edu/gridsphere/gridsphere) is packaged as a webportal, with users in front and TeraGrid services in back. The gateway exposes specific sets of community codes for geography and regional science and provides user-friendly capabilities for performing geographic information analysis using TeraGrid.

Pegasus

Kavitha Ranganathan and Ian Foster conducted extensive simulation studies that examined the relationship between asynchronous data placement and replication and job scheduling.[4] They also examined a variety of job and data scheduling combinations. The data scheduling policies keep track of dataset usage and replicate popular datasets. The authors concluded that scheduling jobs where datasets are present creates the best algorithm, while actively replicating popular datasets also significantly improves execution time.

Developed at the University of Southern California, the Pegasus system[5] uses artificial-intelligence-based planning techniques to approach the workflow scheduling problem. Workflows are based on the Condor DAGMan model and, therefore, restricted to directed acyclic graphs (DAGs). Pegasus schedules workflow activities to be sent randomly to the grid sites that host the virtual data.

The study focuses on adapting the grid to remote-sensing applications. We emphasize a data-intensive computing framework that schedules data and computation tasks in the same frame.

FRAMEWORK

Developed at the Institute of Remote Sensing Applications (IRSA), Chinese Academy of Sciences (CAS), the *remote sensing information service grid node* (RSIN) is a framework that supports execution of applications for

remote-sensing quantitative retrieval by adapting grid computing. As Figure 1 shows, we achieved this goal by using a layered design in which each layer adds functionality, leading to a reliable service with low operational effort. The framework comprises a set of agents.

Every application runs a suite of agents dealing with functions such as resource monitoring and management, statistic collection, and task scheduling. The generator takes user specifications and generates job descriptions for the workflow to move and process data. The monitor acts as a resource broker that discovers and monitors resources via the Condor ClassAds mechanism.[6] The Tuner refers to the procedure that rewrites the parameters for analytic modeling according to runtime statistics. The controller provides the workflow manager's wrapper program.

Data placement failures do not result in computation failures and vice versa.

RSIN stores algorithm codes as modules. The data-process module handles the preliminary processing of raw satellite data, while the model module includes codes for remote-sensing retrieval models. A GUI front end provides the presentation layer, administration, and an easy user interface. RSIN's outside layer consists of an independent-tool-package set, usable by developers and the system. This framework includes several tools.

The Condor[7] scheduler performs the computation in our framework. Stork[8] provides the data-transfer scheduler. The network monitor tool[9] generates the network interface statistics, which the system uses to estimate throughput and identify bottlenecks. The log collection tool parses job log files and estimates computation job performance. A GUI tool helps edit the workflow graph, and the Condor Watcher viewing tool monitors all available resources.

Framework functions

Our framework views data placement tasks and computational jobs as equally important, not just a computational prestage or poststage. A metascheduler executes data placement, distributing data so that it stays separate and asynchronous from computational execution. Thus, data placement failures do not result in computation failures and vice versa. Data can be incorporated into computations efficiently by placing datasets into grids beforehand, moving data off computational resources partly or quickly when computation completes.

A data task descriptor handles the failure tolerance issue. Individual data task descriptors include

- the pipeline identifier string that specifies the dataset's source directory, the name of the files to be accessed, the result's destination directory, and the transfer protocol to use; and
- the time span for the data transition's time-out feature.

When users want to run a data task, they call the generator to fill a descriptor that specifies a data movement task, then submit it like an ordinary task and leave the execution work for the scheduler. The user can change these priorities.

We developed a failure-management strategy using a throughput estimated model for data nodes. This strategy can classify failures as transient or permanent and handle each failure appropriately. If the transfer failed because of a server crash, for example, the controller would suggest resubmitting the data task after switching to an alternate pipeline.

To handle concurrency issues, we rely on the multidomain scheduler that determines the available resources in each domain and uses that result to dynamically assign tasks to domains in task-priority order. The scheduler can restrict the number of concurrent data transfers for a resource.

DATA AND COMPUTATION SCHEDULE

The combined data placement and computation schedule algorithm has two complementary steps:

- planning tasks between workflow steps, and
- scheduling tasks at the same workflow step using a grid hierarchy.

Throughput estimation model

In our throughput model, both the data task and compute task are considered in the same frame, where computation is viewed as a form of data transition network. We bracket a computation task and all data tasks associated with it. This network can contain several data tasks if the computation takes multiple inputs, and perhaps no data task connecting to it if computation takes no input.

A task's throughput requirement is the maximum of the throughputs among the three stages. If the sum of data throughput is greater than that of computation throughput, the combination is set as data-prior. Otherwise, it is set as computation-prior.

Scheduling task between workflow steps

Figure 2 shows the structure of a small remote sensing retrieval workflow. In Figure 2a, the graph of the abstract workflow contains data transfer tasks assigned with D and computation tasks assigned with C. The workflow arranges data placement before execution.

Six workflow steps correspond to processing steps in the remote-sensing procedure. Each step can be further

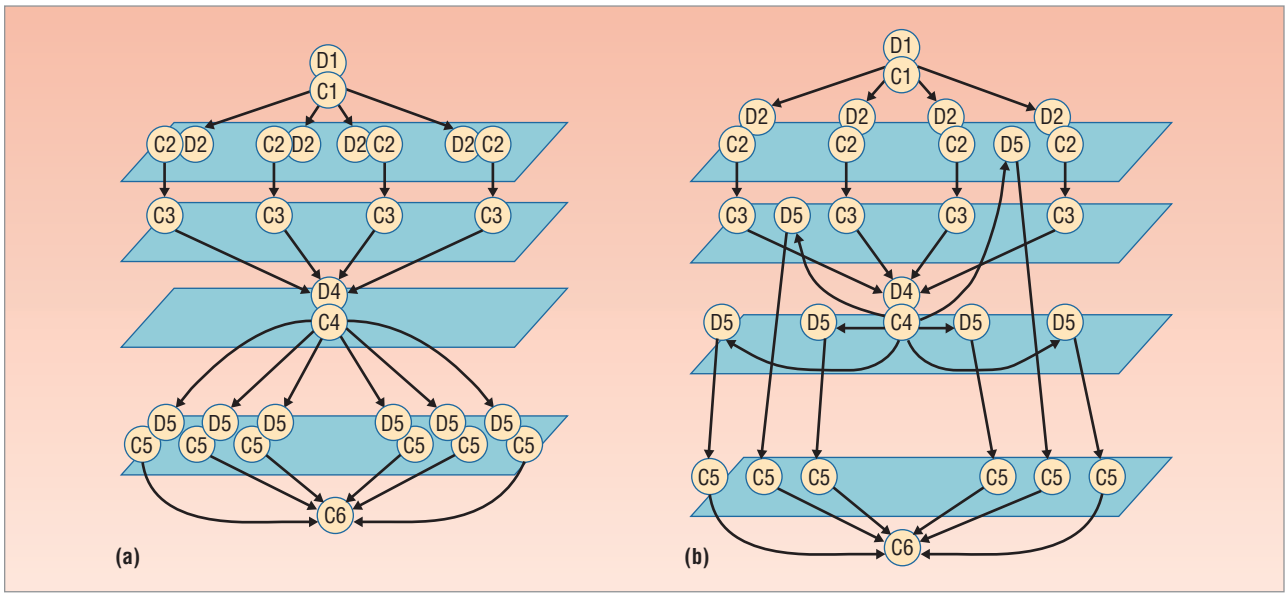


Figure 2. (a) Abstract workflow. The workflow contains data transfer tasks assigned with D and computation tasks assigned with C. (b) Workflow placement. The system performs its data placement strategy based on explicit knowledge of which files will be used during workflow execution.

decomposed into stages such as task decomposing, parallel job running, and result collections. The workflow management specifies two task types: computation and data. Data movement takes place asynchronously with respect to the computation's execution. The system performs the data placement strategy based on an explicit knowledge of which files will be used during workflow execution.[4] Users specify these files as they edit the workflow.

RSIN moves files to a storage system associated with the cluster where workflow execution will take place. Thus, the RSIN workflow avoids explicitly staging the data onto computational resources at runtime. The system obtains data from database resources at step 1. At the next step, the system assigns compute nodes as temporary storage nodes and locally stores result data to await processing at step 3. This approach reduces redundant data transfer.

To balance the workload between workflow steps, our framework uses both data-movement tasks and computation tasks to fill in an underloaded level if estimates show them to be short tasks. The system places some data in grid nodes ahead of time, while it holds other data in reserve on a stage node or file replica server until the computation jobs execute. Therefore, workloads are balanced between steps in the workflow, while the system fully utilizes resources at each step.

Scheduling tasks

To classify available grid sets according to their performance characteristics, we sought to build a grid hierarchy inspired by the grid execution queue hierarchy[10] and multiqueue scheduling. The mixed-load workflow, especially as planned by the DCS algorithm, includes many short tasks. We allow any task to find a level in the grid hierarchy that best matches its needs in terms of response time and available resources. This scheduling ensures that any submitted task will eventually reach the hierarchy level that provides enough resources for longer tasks while providing fast response time for shorter ones.

The algorithm assigns executions or data tasks to domains in proportion to their estimated throughput by the ClassAds match-up mechanism. In an ideal state, it can allocate optimal resources for both computation and data transfer because our task throughput estimation model easily identifies performance bottlenecks. All computation tasks inherently associated with the data task will be sent to the same grid node. Even though the computation tasks take more time, the overall throughput remains optimized.

However, the proper execution level for a given task is difficult to determine if estimating task running time is in itself a demanding computational problem. We used analytical models to solve this. These models yield accurate results and improve precision the longer they execute.

First, we select the candidate grids. We use the method based on a general-purpose resource selection framework that provides necessary information about the heterogeneous system. A set-extended ClassAds language is used to express resource requests, and set-matching between resource requirement and job requirement is used to identify suitable resources.

Second, we use our analytical models to compute grid throughput and estimate a certain task's throughput requirement. We use the throughput to measure system performance, as throughput is proportional to a system's processing ability.

Next, to build the grid hierarchy we sort the grids already filtered by our selection framework according to their estimated throughput. The grid hierarchy is virtually organized by multigrids, so it can take on different forms and be reconstructed dynamically. The number of levels in the grid hierarchy largely depends on the expected complexities in the incoming workflow task stream. The user can also specify which grid belongs to a level. For example, if a workflow consists of two distinctive task classes, such as a small processing task and a massively parallel computing task, we can safely construct a two-level grid hierarchy for these two classes. Grids are classified into the two levels of the hierarchy according to their size and abilities.

We use the following implementation to connect each level of the execution hierarchy to one queue: In a configuration that places multiple grids at a flat form, they can be organized in many ways, including using the available metaschedulers, or *flocking*.

The scheduler maintains multiple virtual task queues, based on a requirements match-up mechanism between queues and grids. Each queue periodically samples the availability of resources in all grids at its level of the execution hierarchy and then labels the tasks' requirements according to the hierarchy level. This information, combined with the data on the total workload complexity in each queue, lets us estimate the expected completion time of tasks. So even if a task can be sent to a certain grid outside its hierarchy, the grid will deny the task's execution request because of conflicting task and grid resource requirements. In the end, each queue connects to one or more grids at the corresponding hierarchy level. The queue configured to serve the shortest tasks connects to the highest level of grid hierarchy, and the queue for somewhat longer tasks connects to the next level.

CASE STUDY

The following case study explores the challenge of retrieving data for an aerosol optical thickness (AOT) application.

Testbed and experimental data

Figure 3 shows the RSIN hierarchy. Level 1 connects to the dedicated CPU server acting as a database and invokes tasks directly without parallelization. Levels 2, 3, and 4 reside on submission machines connected to Condor pools of varying size. Commodity PCs provide the desktop stations. We increased the throughput at Level 5 by connecting to 32-node workstations; at Level 6 we moved to a larger-scale parallel platform.

Table 1 shows the experiment data. Each entry is approximately 200 Mbytes. One mosaic image consists of 14 image scene granules. One day's input image data

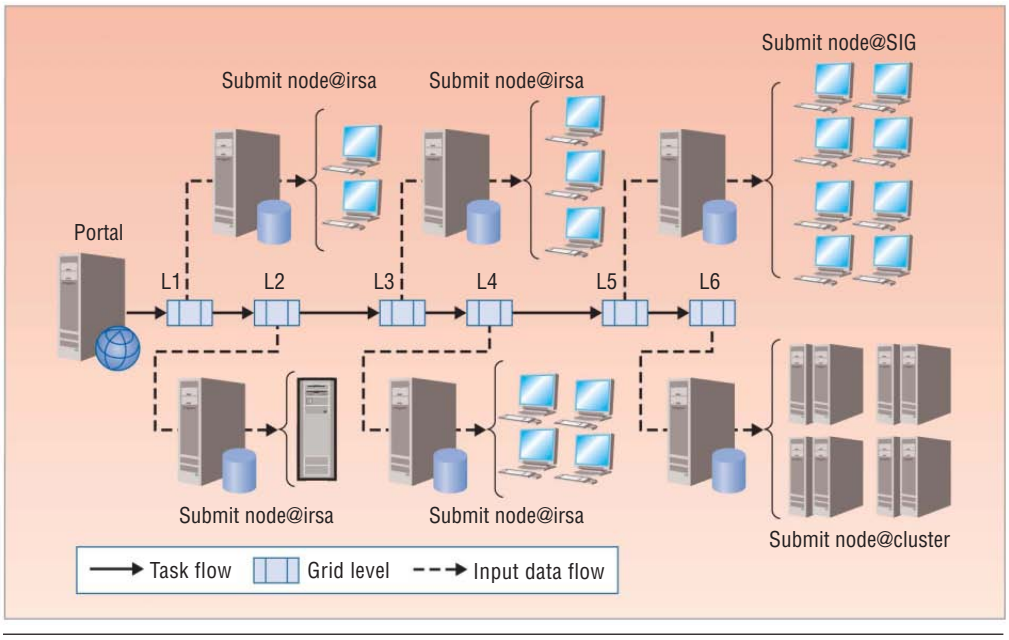


Figure 3. Testbed consisting of six grid levels. Level 1 connects to the dedicated database CPU; levels 2, 3, and 4 reside on submission machines connected to Condor pools of varying size; 32-node workstations provide the throughput at Level 5; and a larger-scale parallel platform provides the computing power at Level 6.

Table 1. Earth science data used for aerosol optical thickness retrieval.

Dataset	Description	Type	Size
NASA satellite imagery	Several hundred scenes	HDF	5.6 Gbytes
MODIS cloud product data	Remote-sensing auxiliary data	GeoTIFF	1.8 Gbytes
China geology	Geologic country map at 1:1,000,000 scale	Shapefile	300 Mbytes
Atmospheric data	China's precipitation, ozone, and CO_2 data for June 2007	ASCII file	500 Mbytes
AERONET data	AERONET stations in China, 2007	ASCII file	60 Mbytes

Table 2. Size of input files in remote-sensing workflow.

Stage of workflow process	Regional granule (Mbytes)	National data (Gbytes)
1 Parameter data process	168	1.52
2 Radiation calibration	680	4.07
3 Geometric correction	680	4.07
4 Image reference	597	3.94
5 Mosaic	0	6.58
6 Cloud detection and mask	802	8.1
7 Retrieval calculation	1.2	10.7
8 Covert map projection	300	1.8

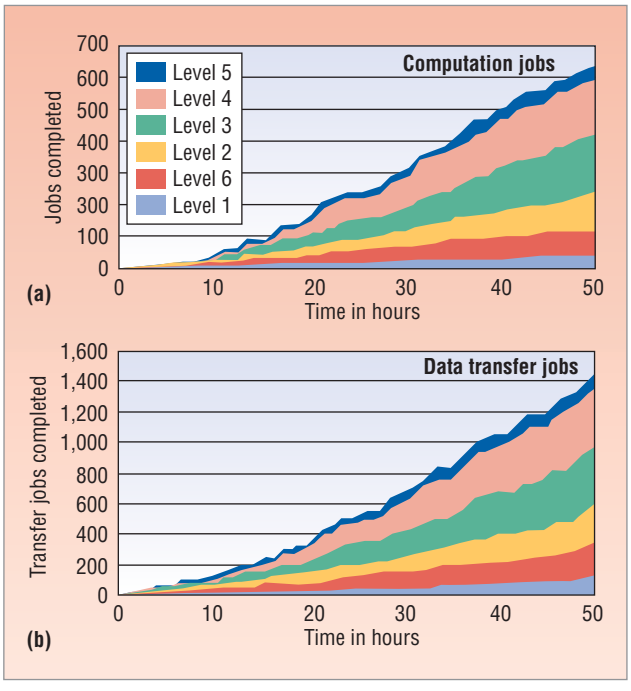


Figure 4. System throughput. (a) Completed computation jobs in 50 hours; (b) completed data transfer jobs in 50 hours.

consists of 28 image scenes from two satellites providing data with a total size of 5.6 Gbytes. The auxiliary input requires another 3 Gbytes.

Table 2 shows the total number of files used as input to each workflow execution and the total input size for each step of the AOT workflow.

Experimental results and performance evaluation

Table 2 shows a step-by-step approach to the retrieval workflow. We completed 600 computation jobs and successfully processed 24 Gbytes of aerosol-property products retrieved from satellite image data in a 50-hour period, as Figure 4 shows. The 1,400 data transfer jobs, including stage-in and stage-out, amount to almost twice as many data as computation executions.

As the statistics reflect, the IRSA Condor pools run dominating tasks. Condor pools for Levels 1 through 4 had a total throughput of eight jobs per hour. Level 5 had higher throughput but more fluctuation as well. At Level 6, the network constraints on the staging node created a bottleneck and slowed the stage-in to that node, consequently influencing overall throughput.

We evaluated our AOT workflow to study the impact of both using and not using the DCS scheduling algorithm, and how DCS performs as input data size increases. We compared the average execution time it takes to schedule tasks with DCS against letting the workflow management system work with the default configuration. As Figure 5 shows, image processing time varies from 15 minutes to three hours. Timing results show that it takes 38 hours to retrieve a mosaic scene of the grids with data. The transition time of workflow steps 5 and 6 dropped to zero because data was placed beforehand and didn't need to be staged in at runtime.

Although computation time increased drastically as the input data increased, the data transfer runtime remained almost constant as t for the input size of original data. This occurred because the DCS data placement strategy offset the time spent processing additional input files. Thus, in these steps, total execution time increased only slightly. The results indicate that the RSIN workflow can achieve a significant reduction in transition time.

Figure 5a shows performance with the small input size of the image granule (168 Mbytes input, 2 Gbytes in total) for the AOT workflow. The graph shows a large advantage for scheduling small tasks. At step 5, which consists of many short-lived tasks, the DCS workflow execution time lasts almost as long as serial execution. In contrast, the default workflow execution time increases by 10 times. Similarly, at step 6, using DCS reduces the total time by approximately 23 minutes when compared to the default case. With workflow step 7, the total input data size is about 2.0 Gbytes, approximately 12 times the size of step 5.

The computation-to-data-transfer ratio rose at step 7, where the most time-consuming computation runs. Figure 5a shows that the workflow execution time increased by more than 2 percent when we used DCR scheduling. This occurred because of the overheads the scheduling caused when using the grid hierarchy. The relative performance improvement is not achieved here in terms of time-reduction percentage.

To facilitate comparisons, we modified our experiments' input data. Figure 5b shows the execution time of each step in the AOT workflow with large-size input data, as we input the image-scenes mosaic that covers China (1.6 Gbytes input, 6.8 Gbytes total). The total

input data size reached approximately 12 times the size of the last experiment.

The graph shows that this workflow offers a great advantage, with the increased data throughput having no effect on workflow performance. On the contrary, workflow execution time dropped by 17.7 percent when we used DCR scheduling and partly preplaced staged datasets before execution began. Our hypothesis asserted that DCR scheduling that prioritizes short-lived tasks and achieves load balance between grids would be more advantageous for data-intensive workflows.

As expected, for the most data-intensive workloads we measured, as the data-transfer task increases, the amount of reduced processing time becomes greater. These results demonstrate the potential scalability of task scheduling for data-intensive scientific applications with mixed workloads. Workflow execution with DCS improves execution time over the performance of default workflow execution as input data increases.

The results of our work have demonstrated that the techniques we've described can improve performance and scalability as data size increases. This will likely introduce new perspectives to researchers for exploiting research based on grid computing with remotely sensed Earth and planetary data.

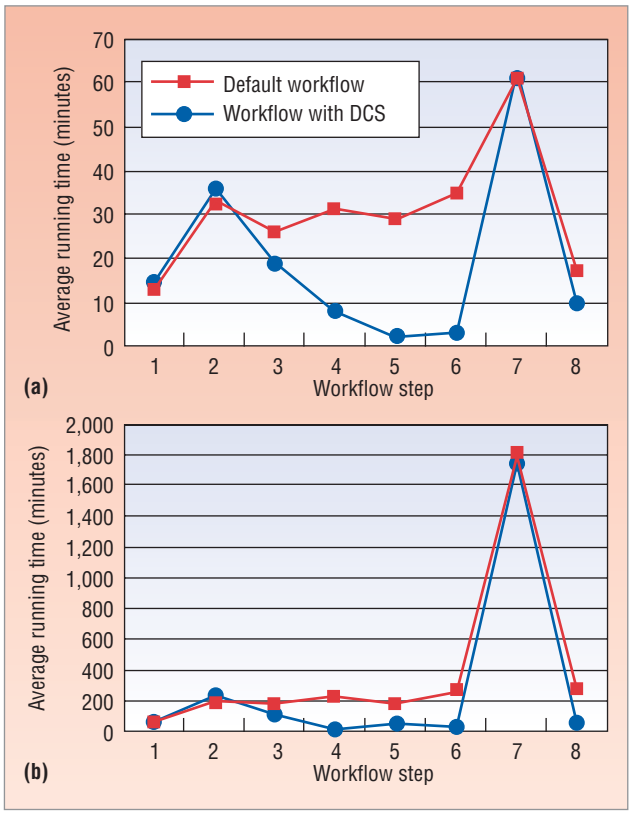


Figure 5. Execution time behavior with respect to different workflow configurations. (a) Average execution time with respect to the workflow step using input data of small size (image granule, 168 Mbytes); (b) average execution time with respect to the workflow step, using input data of large size (image scene mosaic, 1.6 Gbytes).

Acknowledgments

This work was supported in part by the NSFC under grants no. 40671142 and 40471091, by the CAS under grant no. KZCX2-YW-313, and by MOST China under grant no. 2007CB714407.

References

1. C.O. Justice et al., "An Overview of MODIS Land Data Processing and Product Status," *Remote Sensing of Environment*, vol. 83, 2002, pp. 3-15.
2. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputing Applications*, vol. 15, 2001, pp. 1-10.
3. G. Kola et al., "DISC: A System for Distributed Data Intensive Scientific Computing," *Proc. 1st Workshop Real, Large Distributed Systems* (WORLDS 04), 2004; http://www.usenix.org/events/worlds04/tech/full_papers/kola/kola.pdf.
4. K. Ranganathan and I. Foster, "Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications," *Proc. 11th IEEE Int'l Symp. High-Performance Distributed Computing* (HPDC 02), IEEE Press, 2002, p. 352.
5. E. Deelman et al., "Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems," *Scientific Programming J.*, vol. 13, 2005, pp. 219-237.
6. R. Raman, M. Livny, and M. Solomon, "Matchmaking: Distributed Resource Management for High-Throughput Computing," *Proc. 7th IEEE Int'l Symp. High-Performance Distributed Computing*, IEEE Press, 1998, pp. 140-146.
7. J. Frey and T. Tannenbaum, "Condor-G: A Computation Management Agent for Multi-Institutional Grid," *Cluster Computing*, vol. 5, 2001, pp. 237-246.
8. T. Kosar and M. Livny, "Stork: Making Data Placement a First-Class Citizen in the Grid," *Proc. 24th IEEE Int'l Conf. Distributed Computing Systems* (ICDCS 04), IEEE Press, 2004, pp. 342-349.
9. M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with Tcp Throughput," *IEEE/ACM Trans. Networking*, vol. 11, 2003, pp. 537-549.
10. M. Silberstein et al., "Scheduling Mixed Workloads in Multi-Grids: The Grid Execution Hierarchy," *Proc. 15th IEEE Symp. High-Performance Distributed Computing* (HPDC 06), IEEE Press, 2006, pp. 291-302.

Yong Xue *is a professor at the State Key Laboratory of Remote Sensing Science, jointly sponsored by the Institute of Remote Sensing Applications (IRSA) of the Chinese Academy of Sciences (CAS) and Beijing Normal University. His research interests include geocomputation,*

aerosol optical depth retrieval from remotely sensed data, thermal inertia modeling, and heat exchange calculation for the boundary layer. Xue received a PhD in remote sensing from the University of Dundee. He is a senior member of the IEEE. Contact him at y.xue@londonmet.ac.uk.

***Wei Wan** is a research PhD student at IRSA, CAS. His research interests include geocomputation, remote-sensing image processing, and aerosol optical depth retrieval. Wan received an MSc in electronic engineering from Central China Normal University. Contact him at eric_104@163.com.*

***Yingjie Li** is a research student at the Graduated University of the Chinese Academy of Sciences. His research interests include quantitative remote sensing, aerosol science, and signal processing. Li received a BS in remote sensing science and technology from Nanjing University of Information Science and Technology. Contact him at liyingjie07@mails.gucas.ac.cn.*

***Jie Guang** is a research PhD student at IRSA, CAS. Her research interests include aerosol optical depth retrieval and urban BRDF modeling. Guang received an MSc in geographic information systems from Nanjing Normal University. Contact her at guangj82@hotmail.com.*

***Linyian Bai** is a research PhD student at IRSA, CAS. Her research interests include quantitative remote-sensing inversion, remote-sensing monitoring of atmospheric environment, grid computing and applications in geosciences. Bai received an MSc in geographic information systems from China University of Mining & Technology, Beijing. Contact her at bai_linyan@163.com.*

***Ying Wang** is a research PhD student at IRSA, CAS. Her research interests include remote-sensing image processing and aerosol optical depth retrieval. Wang received a BS in geographic information systems from China University of Geosciences. Contact her at wangy86@sina.com.*

***Jianwen Ai** is a research PhD student at IRSA, CAS. His research interests include geocomputation and remote-sensing image processing. Ai received an MSc in geographic information systems from China University of Mining and Technology, Beijing. Contact him at neau_ajw@hotmail.com.*

COVER FEATURE

Accelerating Real-Time String Searching with Multicore Processors

Oreste Villa, Politecnico di Milano/Pacific Northwest National Laboratory
Daniele Paolo Scarpazza and Fabrizio Petrini, IBM T.J. Watson Research Center

String searching is at the core of tools used to search, filter, and protect data, but this has become increasingly difficult to do in real time as communication speed grows. The authors present an optimization strategy for a popular algorithm that fully exploits the IBM Cell Broadband Engine architecture to perform exact string matching against large dictionaries and also offer various solutions to alleviate memory congestion.

The amount of digital data produced and exchanged throughout the world is exploding.[1] Accessing and searching this expanding digital universe and protecting it against viruses, malware, spyware, spam, and deliberate intrusion attempts are becoming key challenges.

Network intrusion detection systems (IDSs) no longer can filter undesired traffic using mere header information because threats often target the application layer; consequently, an IDS must employ *deep packet inspection* (DPI) and check the incoming payload against a database of threat signatures. However, given the growing number of threats and increasing link speeds—especially 10-Gbps Ethernet and beyond—DPI is becoming harder to perform in real time without impacting the bandwidth and latency of the communications under scrutiny.

The algorithms outlined in the "String-Searching Algorithms" sidebar address this need to search and filter information and are at the core of search engines, IDSs, virus scanners, spam filters, and content-monitoring systems. Fast string-searching implementations traditionally have been based on specialized hardware like field-programmable gate arrays (FPGAs) and application-specific instruction-set processors, but the advent of multicore architectures such as the IBM Cell Broadband Engine (CBE) is adding important new players to the game.

Previous DPI research[2-10] has mainly focused on speed: achieving the highest processing rate, typically with a small dictionary, on the order of a few thousand keywords. More recently, a second dimension is gaining importance: dictionary size. Implementations with dictionaries of hundreds of thousands of patterns are beyond the state of the art. A major challenge for the scientific community is to seamlessly extend the scanning speed already obtained with small dictionaries to much larger datasets.

In 2007, we presented a parallel algorithm for the CBE based on a deterministic finite-state machine that provides search throughput in excess of 5 Gbps per synergistic processing element (SPE).[6] All eight SPEs in a CBE jointly deliver 40 Gbps with a dictionary of about 200 patterns, or a throughput of 5 Gbps with a larger dictionary of 1,500 patterns. While this solution's throughput is very high, it can only handle a small dictionary, making it unsuitable for many applications.

More recently, we've expanded our work to include large dictionaries by using the entire available main memory—1 Gbyte in our experimental setup—to store the dictionary. Because accessing main memory requires higher latency than each SPE's local store, we've focused on orchestrating the memory traffic to increase throughput and lower response time.

Toward that end, we've developed a parallelization strategy for the Aho-Corasick string-searching algo-

Published by the IEEE Computer Society

String-Searching Algorithms

Researchers have developed numerous string-searching techniques that can be roughly divided into two classes: Bloom filters and exact string-searching algorithms.

Bloom filters[1] are space-efficient probabilistic data structures that can be used for generic membership tests. A Bloom filter can store a compact representation of a dictionary. We can query the filter to determine whether a string belongs in the dictionary. The price for this reduced footprint is a predictable rate of false positives and a relatively high query cost. Designers can choose parameters to negotiate the false-positive rate and employ an exact string-searching algorithm to perform the final decision.

Bloom filters exhibit considerable data-level parallelism that specialized hardware like field-programmable gate arrays can exploit. On the other hand, implementing Bloom filters efficiently on general-purpose processors is difficult, and it's even harder on the Cell Broadband Engine—for example, a dictionary of strings with variable length requires a bank of Bloom filters working on input windows of different length.

Exact string-searching algorithms are based on finite-state machines or precomputed tables. Knuth-Morris-Pratt[2] searches all the occurrences of a single pattern in a text, and it skips as many input symbols as possible when a mismatch occurs. The Boyer-Moore[3] single-pattern searching algorithm speeds up the search via precomputed tables. It starts matching from the last character in the pattern so that, upon a mismatch, it skips an entire span of the input. Commentz-Walter[4] is a multiple-pattern extension of the Boyer-Moore algorithm. Aho-Corasick,[5] which we employed, also belongs in this class.

References

1. B.H. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," *Comm. ACM*, vol. 13, no. 7, 1970, pp. 422-426.
2. D.E. Knuth, J.H. Morris Jr., and V.R. Pratt, "Fast Pattern Matching in Strings," *SIAM J. Computing*, vol. 6, no. 2, 1977, pp. 323-350.
3. R.S. Boyer and J.S. Moore, "A Fast String Searching Algorithm," *Comm. ACM*, vol. 20, no. 10, 1977, pp. 62-72.
4. B. Commentz-Walter, "A String Matching Algorithm Fast on the Average," *Proc. 6th Int'l Colloq. Automata, Languages and Programming*, H.A. Maurer, ed., LNCS 71, Springer, 1979, pp. 118-131.
5. A.V. Aho and M.J. Corasick, "Efficient String Matching: An Aid to Bibliographic Search," *Comm. ACM*, vol. 18, no. 6, 1975, pp. 333-340.

rithm that achieves performance comparable to other results in the literature with small data dictionaries but exploits the CBE's sophisticated memory subsystem to effectively handle large dictionaries, clearly demonstrating that multicore processors are a viable alternative to special-purpose solutions or FPGAs. We believe that this technique can be generalized to other algorithms and application domains to handle large datasets.

We've also conducted an in-depth analysis of the algorithm's performance bottleneck: memory congestion. Synthetic memory traffic provides almost optimal performance, but actual implementations of the algorithm experience significant slowdown. We identified three aspects of memory congestion—memory pressure, memory layout issues, and hot spots—and alleviated these phenomena algorithmically, achieving quasi-optimal performance.

PARALLELIZATION STRATEGY

Our solution is based on AC-opt, a variant of the Aho-Corasick algorithm that uses deterministic finite automata (DFA)[11] to process separate chunks of input text—in this case, against a dictionary that is too large to fit in the SPE's local store and must therefore be kept in main memory. An AC-opt automaton is a two-step loop that consists of

- determining the address of the next state in the state transition table (STT) depending on the input and the current state, and
- fetching the next state from main memory.

The first step takes a few nanoseconds, an interval we call *transition time*, but the second can involve a memory latency of several hundred ns. A single automaton thus exhibits abysmal hardware utilization because it spends most of its time waiting.

Our parallelization technique ameliorates this problem by mapping multiple automata on the same SPE and overlapping data transfer and computation. This is possible because each SPE includes a memory flow controller (MFC) that carries out data transfers while the program execution continues. Each automaton operates on distinct chunks of the input text. As Figure 1a shows, our strategy splits the input text a first time among the different SPEs and a second time, within each SPE, among the automata that run on it. The chunks overlap partially to allow matching of those patterns that cross a boundary. The neces-

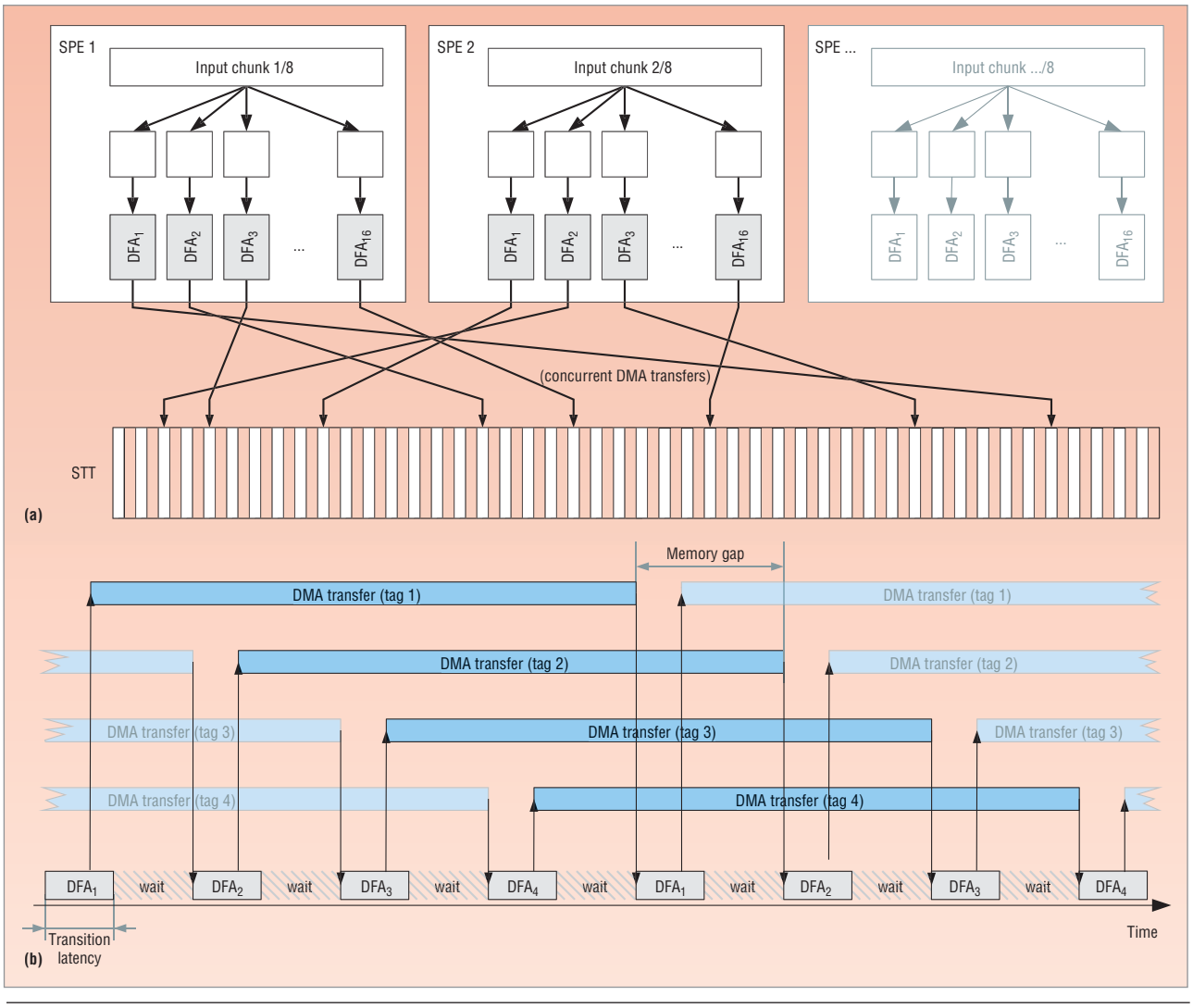


Figure 1. Parallelization strategy. (a) Each SPE runs multiple deterministic finite automata (DFA_1, DFA_2, ...). Each automaton processes a separate chunk of the input text, but all the automata access the same state transition table (STT) in main memory. (b) A sample schedule of multiple automata ($DFA_1 \ldots DFA_4$) that overlaps computation and data transfers. Gray rectangles represent transition times; upward arrows indicate direct memory access (DMA) issue events; downward arrows indicate DMA completion events.

sary overlapping is equal to the length of the longest pattern in the dictionary minus 1 symbol. As Figure 1b shows, a schedule that overlaps computation and data transfer can fully utilize the memory subsystem. The figure also illustrates the concept of *memory gap*: the minimum sustainable time interval between the completion of two consecutive memory accesses. As long as the transition time is shorter than the memory gap (a condition always true in our experiments), the memory gap determines the rate at which the DFAs can process input symbols and, ultimately, overall algorithm performance.

Our algorithm can exploit memory bandwidth by issuing large transfers and using all the transferred data. This isn't possible with AC-opt because of the small amount of data actually needed at each read: A pointer to the next state is only 4 bytes in size. Moreover, there's no trivial way to exploit locality and group multiple reads in a single block read.

Our benchmarks reveal that the memory subsystem is best utilized when the blocks are smaller than 64 bytes and 16 or more direct memory access (DMA) requests are outstanding at all times, as Figure 2 shows. This suggests mapping 16 automata per SPE and accessing the STT in 64-byte blocks. A block size of 32 or 64 bytes makes the memory gap practically independent from the number of SPEs, guaranteeing almost ideal scaling up to eight SPEs.

Table 1 shows the performance bounds for a system composed of one and two CBE processors. The reciprocal of the memory gap is the maximum frequency at which an SPE can accept input symbols (one symbol

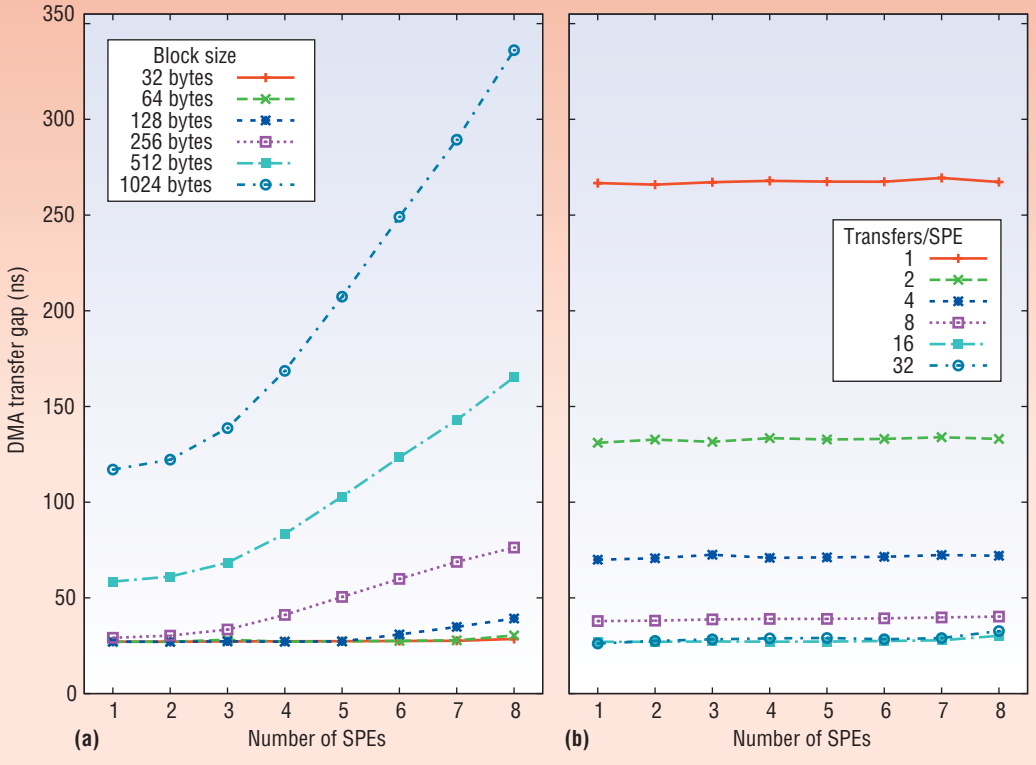


Figure 2. Memory gap. The memory gap is a function of (a) transfer size S and (b) number of concurrent direct memory access (DMA) requests N. To optimize the memory gap, each SPE can use N = 16 and S ≤ 64 bytes. In these conditions, scalability is ideal. In both cases, transfers hit random locations in an 864-Mbyte-wide contiguous area. In the left plot, N = 16; in the right plot, S = 64 bytes.

Table 1. Performance bounds for system containing one and two Cell Broadband Engine processors.

CBE system	Memory gap (per SPE)	Symbol frequency (per SPE)	Throughput (per SPE)	Throughput (aggregate)
One CBE processor (8 SPEs)	29.34 ns	34.07 MHz	272.56 Mbps	2.18 Gbps
Two CBE processors (16 SPEs)	40.68 ns	24.58 MHz	196.64 Mbps	3.15 Gbps

= 1 byte; the impact of chunk overlapping is ignored). Aggregate performance values are in Gbps to simplify comparison with network wire speeds. These values represent the upper bounds for a main-memory-based AC-opt implementation. A real implementation should approximate these boundaries as closely as possible.

PERFORMANCE ANALYSIS

We profiled our AC-opt implementation in four representative experimental scenarios. In the first, a full-text search system searched the King James Bible against a dictionary containing the 20,000 most used words in the English language, whose average length is 7.59 characters. In scenario 2, a network content monitor searched a tcpdump stream obtained while a user browsed the popular news website Slashdot against the same English dictionary. In the third scenario, a network IDS searched the same stream against a dictionary of 8,400 randomly generated binary patterns, whose length is uniformly distributed between 4 and 10 characters. In scenario 4, an antivirus scanner searched a randomly generated binary file against a dictionary with the same properties as the third scenario.

Figure 3 shows the performance results. For uniformity, all inputs are truncated to the same length as scenario 1 (4.43 Mbytes), and the dictionaries yield similar-sized STTs (49,849 and 50,126 states). To our surprise, performance is poor: Throughput is significantly below the theoretical boundary and doesn't scale.

Because the transition time is identical in all scenarios, the degradation must be due to memory access. Binary,

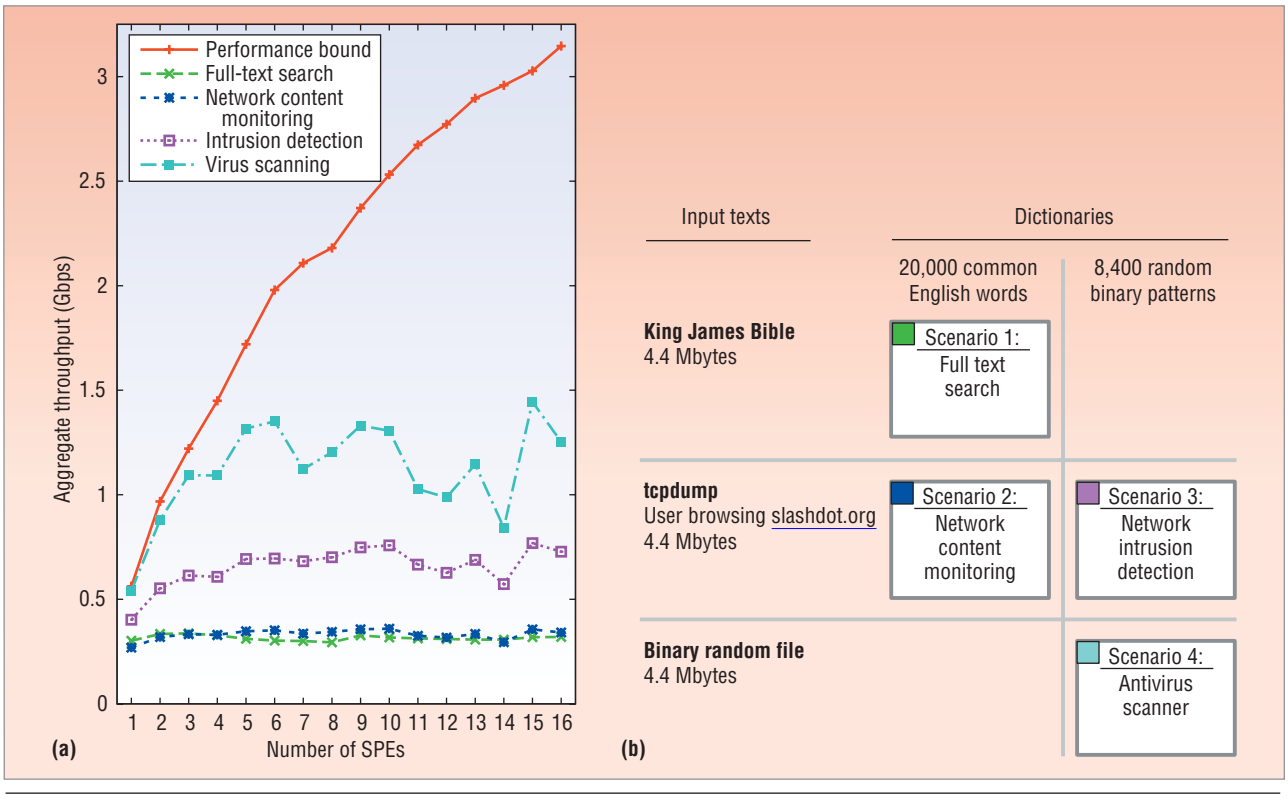


Figure 3. Four representative experimental scenarios. (a) Throughput is significantly below the theoretical boundary and doesn't scale. (b) Performance and scalability fall largely below the expected values.

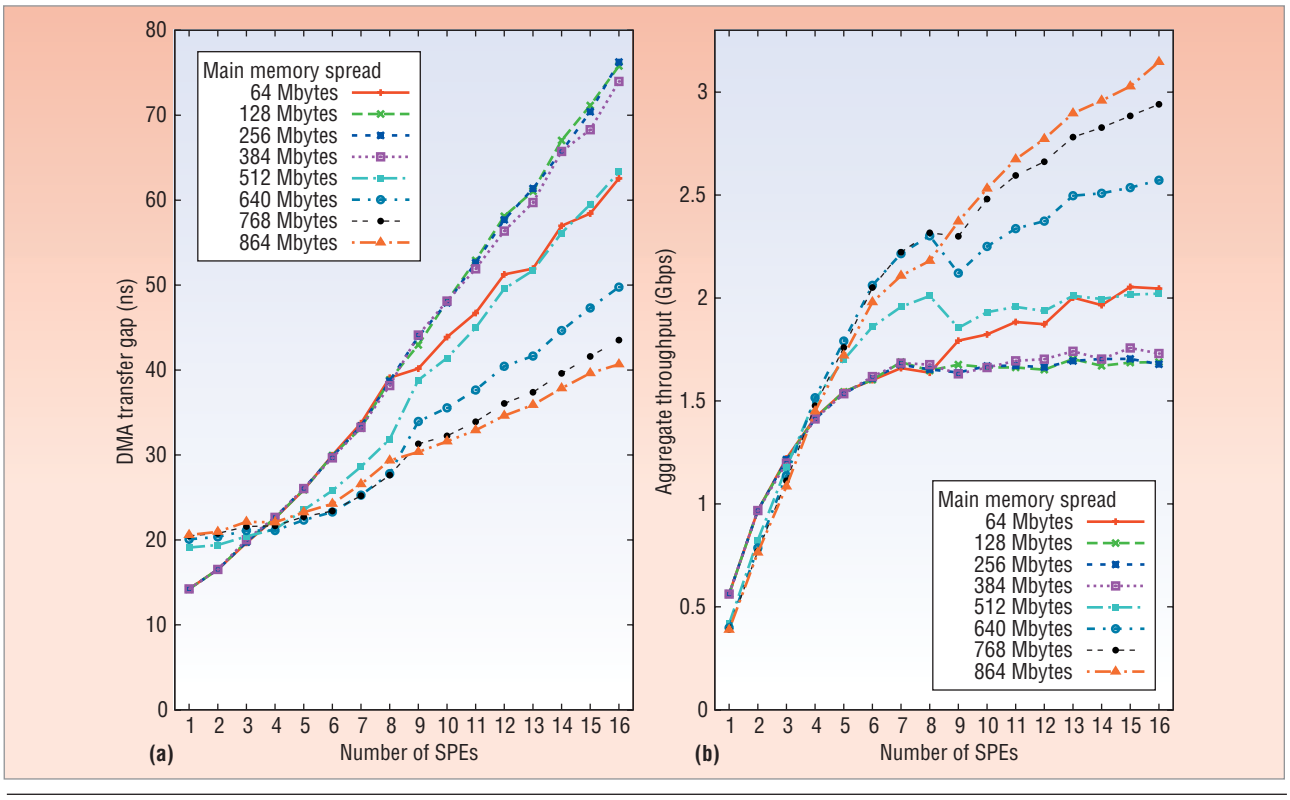


Figure 4. Memory pressure. When memory pressure increases, (a) the memory gap increases and, consequently, (b) throughput degrades. Spreading the STT across a larger area alleviates pressure and improves performance. Each SPE generates 16 concurrent 64-byte transfers.

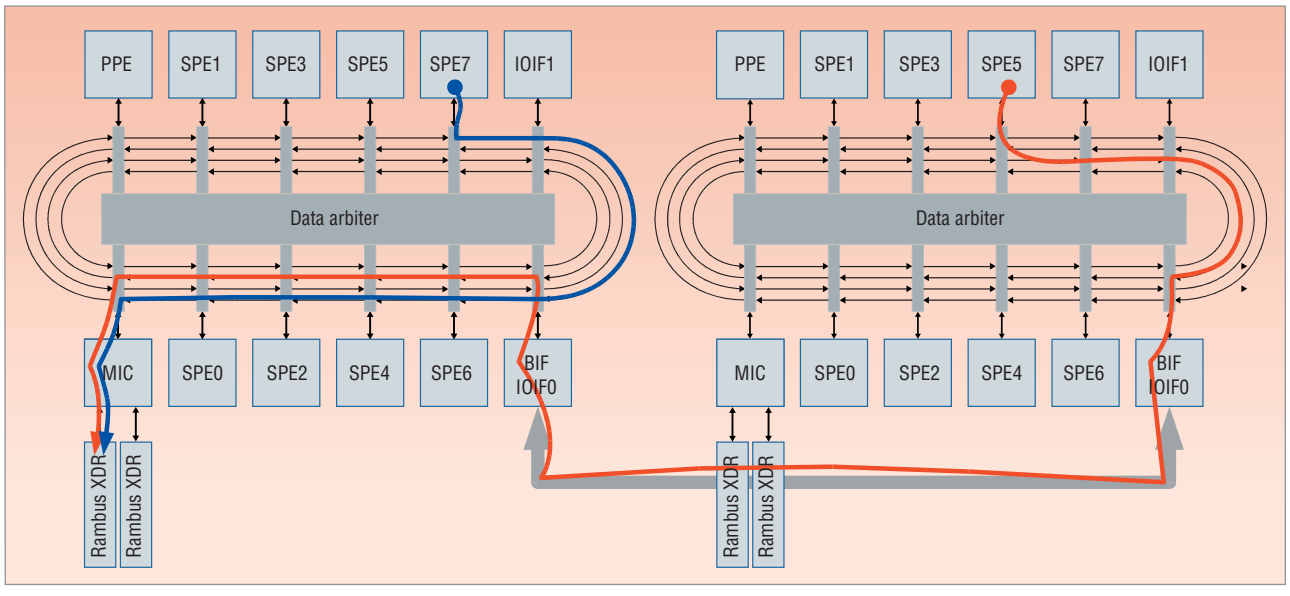


Figure 5. Memory layout. If the STT resides entirely in memory connected to the memory interface controller of the first of two coupled CBE processors, all the SPEs will be congesting that MIC while the other MIC is unused.

randomly generated dictionaries outperform natural, alphabetic ones. The antivirus scanner, which searches random inputs against random dictionaries, achieves the best performance. This scenario has the highest and most uniform fanout from each state, suggesting that uniformly distributed usage patterns cause less congestion.

MEMORY CONGESTION

Several experiments we conducted show that memory congestion has three major components: *memory pressure*, *memory layout issues*, and *hot spots*. The results suggest changing the algorithm to make accesses as uniform as possible across all available memory.

Memory pressure is the number of accesses that hit each block of memory of a given, fixed size in a unit of time. It's higher when we employ many SPEs and automata and allocate the STT in a smaller area; it's lower when we use fewer SPEs and spread the STT across a large memory area.

To assess its impact, we considered concurrent accesses to an STT stored in a contiguous area of main memory varying in size from 64 Mbytes to 864 Mbytes, the maximum available space on our system. As Figure 4 shows, memory pressure degrades performance. Spreading STT entries across a larger area relieves memory pressure and improves performance.

Memory layout also influences the curves in Figure 4, as well as the difference in scalability when we use more than eight SPEs. Consider the coupled CBE processors in Figure 5, in which half of the memory is connected to the first processor's memory interface controller (MIC) and the other half to the second processor's MIC. The implementation of our algorithm allocates heap memory sequentially starting from the first half, thus a relatively small STT will reside entirely in the memory connected to the first processor. The SPEs in the first processor will access the STT directly (blue line), but an SPE in the second processor will access the STT through the first one's MIC (red line). This access incurs a longer latency due to the more numerous transactions and loads the first processor's MIC. Larger STTs allocated across the boundary between the two memory areas can still be unevenly distributed. Hot spots are memory regions frequently accessed by multiple transfers at the same time, typically because they contain states that are hit frequently. In general, multiple automata are likely to access frequently hit states at the same time, causing hot spots. Our experiments show that a few states, typically in levels 0 and 1 of the DFA's state transition graphs, are responsible for the vast majority of the hits while the remaining ones are almost never accessed.

PERFORMANCE OPTIMIZATION

To counter memory congestion, we transform the STT using a combination of four techniques: state caching, state shuffling, alphabet shuffling, and state replication.

Optimization techniques

State caching involves storing a copy of the states closest to the initial state in each SPE's local store. Given the little space available, the local store can statically cache only 180 STT entries. This technique relieves the entire traffic generated by the cached states from the memory subsystem.

The *state shuffling* technique randomly renumbers all the states except the initial one and rewrites the STT accordingly. By using the largest state space that the

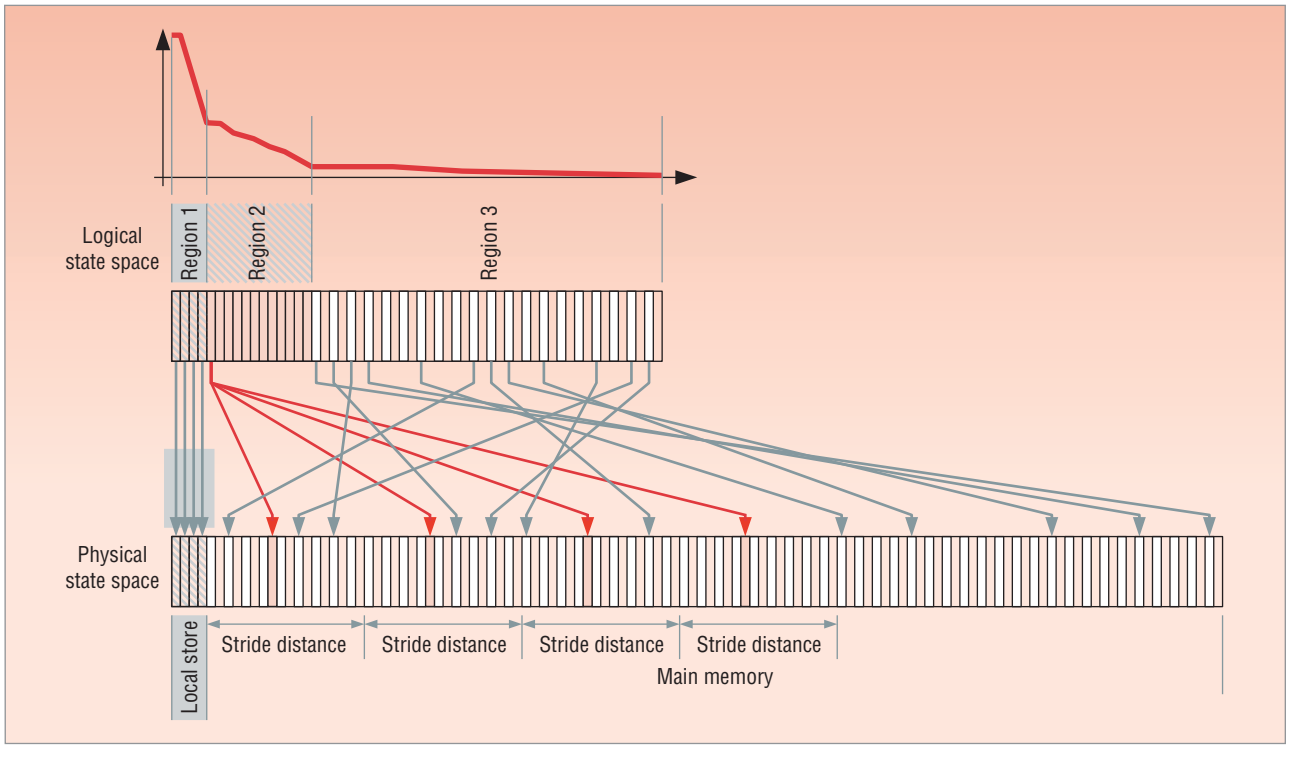


Figure 6. Performance optimization. To reduce memory congestion, we transform the logical state space into a physical state space through state caching (region 1), state shuffling (regions 2 and 3), state replication (region 2), and alphabet shuffling (all three regions).

available memory allows, it spreads memory pressure along all the available blocks at the expense of space efficiency—the resulting STT includes unused entries.

Concurrent accesses to close offsets within separate STT entries also cause contention. To counter this phenomenon, *alphabet shuffling* changes the order of the input symbols (from 0 to 255) within each entry and stores next-state values accordingly. A simple function like "symbol′ = (symbol + state) mod A", where A is the alphabet size′, is inexpensive (the modulo is reduced to a bitwise AND), lets different states enjoy different shuffles, and is just as effective as other, more complex functions.

To mitigate hot spots, we replicate frequently used states so that different automata access different replicas. Replicas are allocated at regular intervals so that the SPEs can compute their addresses with inexpensive arithmetics. *State replication* is very effective: Replicating a state four times reduces memory pressure by 75 percent.

We assume that states in an AC-opt automaton are numbered by increasing distance from the initial state. Our optimization techniques map this *logical state space* into a *physical state space* in main memory. As Figure 6 shows, we partition the logical state space in three regions and apply state caching to region 1, state shuffling and replication to region 2, state shuffling to region 3, and alphabet shuffling to all three regions. We determine the border between regions 2 and 3, the replication factor, using some ad hoc heuristics.

Experimental results

To evaluate the effectiveness of these optimization techniques, we compared their performance against the bounds listed in Table 1. In each of the graphs shown in Figure 7, a solid red line indicates the technique's performance corresponding to this bound.

All the experiments used AC-opt implemented in the C language with CBE intrinsics, compiled under the IBM CBE Software Development Kit v2.1, and run on an IBM DD3 blade with two processors running at 3.2 GHz, with 1 Gbyte of RAM and Linux kernel v2.6.16. The experiments considered the same four scenarios shown in Figure 3.

Figure 7a shows the effect of state shuffling: a significant improvement over Figure 3, especially in the two scenarios that employ a binary dictionary. On the other hand, congestion still compromises scalability. Figure 7b illustrates the combined impact of state shuffling and replication. Region 2 contains 10,000 states—approximately 20 percent of the states in all four scenarios. Figure 7c shows the additional effect of alphabet shuffling. Values are almost optimal up to seven to eight SPEs, then scalability decreases due to memory layout issues.

Figure 7d shows the performance when we use all four techniques, including state caching (180 states). The longer transition time necessary to determine whether or not each state is cached causes a small

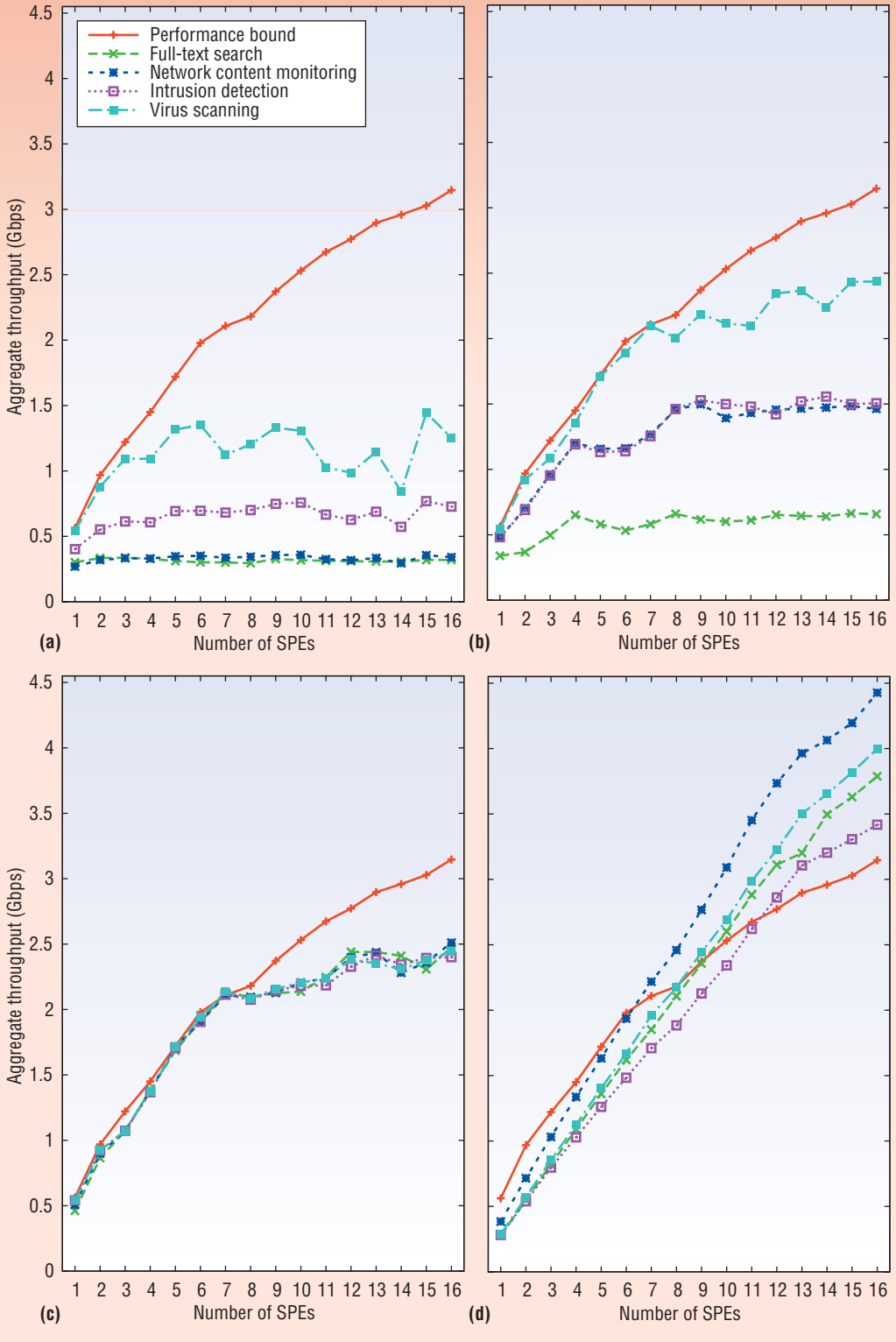


Figure 7. Optimization technique performance. (a) State shuffling. (b) State shuffling and replication. (c) State shuffling, state replication, and alphabet shuffling. (d) State shuffling, state replication, alphabet shuffling, and state caching.

throughput degradation when we employ a few SPEs, which disappears when we use more SPEs. Because state caching is the only technique that also involves the local store, it achieves higher performance than the theoretical bound, which assumes use of the main memory only.

The Cell Broadband Engine can be successfully applied to high-performance keyword scanning, traditionally the domain of specialized processors or FPGAs. Our prototype implementation of AC-opt achieved a remarkable throughput of 2.5 Gbps in a representative set of input texts and dictionaries. This result relies on the CBE's unique capability to support irregular memory communication using explicit DMA primitives. By properly orchestrating and pipelining memory requests, we've created, at the user level, a virtual layer in the memory hierarchy with a latency of only 30 ns that is as large as the entire main memory.

We can achieve this level of performance only under ideal traffic conditions, but we've developed multiple techniques to approximate these conditions: state caching, state shuffling, alphabet shuffling, and state replication. We believe that the proposed solutions can be successfully applied to other data-intensive applications displaying irregular access patterns across large datasets.

Acknowledgments

This research was conducted under the Laboratory Directed Research and Development Program for the Data Intensive Computing Initiative at Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the US Department of Energy under contract DEAC0576RL01830.

References

1. "The Expanding Digital Universe: A Forecast of Worldwide Information Growth through 2010," white paper, IDC Corp., Mar. 2007.
2. D.C. Suresh et al., "Automatic Compilation Framework for Bloom Filter Based Intrusion Detection," *Reconfigurable Computing: Architectures and Applications*, K. Bertels, J.M.P. Cardoso, and S. Vassiliadis, eds., LNCS 3985, Springer, 2006, pp. 413-418.
3. L. Bu and J.A. Chandy, "A CAM-Based Keyword Match Processor Architecture," *Microelectronics J.*, vol. 37, no. 8, 2006, pp. 828-836.
4. I. Sourdis and D. Pnevmatikatos, "Pre-Decoded CAMs for Efficient and High-Speed NIDS Pattern Matching," *Proc. 12th Ann. IEEE Symp. Field-Programmable Custom Computing Machines* (FCCM 04), IEEE CS Press, 2004, pp. 258-267.
5. S. Antonatos et al., "Performance Analysis of Content Matching Intrusion Detection Systems," *Proc. 2004 Symp. Applications and the Internet* (SAINT 04), IEEE CS Press, 2004, pp. 208-218.
6. D.P. Scarpazza, O. Villa, and F. Petrini, "Peak-Performance DFA-Based String Matching on the Cell Processor," *Proc. 21st Int'l Parallel and Distributed Processing Symp.* (IPDPS 07), IEEE Press, 2007.
7. C. Chang and R. Paige, "From Regular Expressions to DFA's Using Compressed NFA's," *Proc. 3rd Ann. Symp. Combinatorial Pattern Matching* (CPM 92), A. Apostolico et al., eds., LNCS 644, Springer, 1992, pp. 88-108.
8. R. Sidhu and V.K. Prasanna, "Fast Regular Expression Matching Using FPGAs," *Proc. 9th Ann. IEEE Symp. Field-Programmable Custom Computing Machines* (FCCM 01), IEEE CS Press, 2001, pp. 227-238.
9. B.L. Hutchings, R. Franklin, and D. Carver, "Assisting Network Intrusion Detection with Reconfigurable Hardware," *Proc. 10th Ann. IEEE Symp. Field-Programmable Custom Computing Machines* (FCCM 02), IEEE CS Press, 2002, pp. 111-120.
10. H-J. Jung, Z.K. Baker, and V.K. Prasanna, "Performance of FPGA Implementation of Bit-Split Architecture for Intrusion Detection Systems," *Proc. 20th Int'l Symp. Parallel and Distributed Processing Symp.* (IPDPS 06), IEEE Press.
11. B.W. Watson, *The Performance of Single-Keyword and Multiple-Keyword Pattern Matching Algorithms*, tech. report 94/19, Faculty of Mathematics and Computing Science, Eindhoven Univ. of Technology, 1994.

Oreste Villa *is a PhD student in the Dipartimento di Elettronica e Informazione at Politecnico di Milano, Milano, Italy, and a PhD intern in the Applied Computer Science Group of the Pacific Northwest National Laboratory (PNNL), Richland, Washington. His research interests include computer architectures, multiprocessor systems, synchronization techniques, and programming techniques for multicore architectures. Villa received an ME in embedded systems design from the Advanced Learning and Research Institute, University of Lugano, Switzerland, and an MS in electrical engineering from the Università degli Studi di Cagliari, Cagliari, Italy. Contact him at ovilla@elet.polimi.it.*

Daniele Paolo Scarpazza *is a postdoctoral research fellow in the Cell Solutions Department at the IBM T.J. Watson Research Center, Yorktown Heights, New York. His research focuses on parallel algorithms and programming abstractions for the CBE processor, with emphasis on pattern matching, text searching, information retrieval, and large data structure exploration. Scarpazza received a PhD in information engineering from Politecnico di Milano. Contact him at dpscarpazza@us.ibm.com.*

Fabrizio Petrini *is a senior researcher in the Cell Solutions Department at the IBM T.J. Watson Research Center. His research interests include high-performance interconnection networks, parallel computer architectures, performance evaluation, scheduling algorithms for parallel architectures, and cluster computing. Petrini received a PhD in computer science from Università di Pisa, Pisa, Italy. He is a member of the IEEE. Contact him at fpetrin@us.ibm.com.*

COVER FEATURE

Analysis and Semantic Querying in Large Biomedical Image Datasets

Vijay S. Kumar, Sivaramakrishnan Narayanan, Tahsin Kurc, Jun Kong, Metin N. Gurcan, and Joel H. Saltz
The Ohio State University

Biomedical image analysis plays an important role in diagnosing, prognosing, and treating complex diseases. The authors describe a set of techniques for analyzing, processing, and querying large image datasets using semantic and spatial information.

Digital microscopy opens up new opportunities to study the tissue characteristics of disease at the cellular scale. Traditionally, human experts visually examine tissue images, classify the images and image regions, and make a diagnosis. This process is time-consuming and subject to high inter- and intraobserver variation. Such weaknesses associated with visual evaluation processes have motivated the development of computer-aided prognosis solutions. The sheer size of image datasets makes gleaning information from digital microscopy slides a data-intensive process requiring efficient system support.

Consider, for example, computer-aided classification of neuroblastoma, a type of peripheral neuroblastic tumor and the most common extracranial, solid, malignant tumor affecting children's health. This most common cancer in infants and the most common extracranial childhood cancer prompted research efforts to identify its pathological characteristics and develop computerized classification algorithms for using digitized microscopy images to assist in characterizing and classifying tissues.

A high-power microscope can rapidly obtain a 10-30 Gbyte image from a tissue sample. Clinicians usually obtain multiple images from a subject to conduct longitudinal studies or to study changes across multiple layers of tissue. The size of such image datasets reaches hundreds of gigabytes or terabytes.

The process of neuroblastoma prognosis using digitized slides is essentially a workflow involving a sequence of steps: image segmentation, feature construction, feature selection, feature extraction, classification of features, and annotation of images and image regions.[1] Executing this workflow on large image datasets can take several hours (or days in some cases), hindering the effectiveness of computer-aided prognosis in imaging studies.

Image analysis can provide semantic information through annotations and classifications. Neuroblastoma cases, for example, can be classified by tissue type into different histologies based on features such as neuroblastic differentiation (undifferentiated, poorly differentiated, and differentiated subtypes) and the presence or absence of Schwannian stroma development (stroma-poor and stroma-rich tissue).

A classification scheme can be organized hierarchically. Many projects seek to build a semantic database using analysis results to support more effective organization and aggregation of imaging information and integration of this information with other types of data, such as clinical data and molecular data, for further analysis. Indeed, large-scale efforts such as the cancer Biomedical Informatics Grid (caBIG, https://cabig.nci.nih.gov), the Cardiovascular Research Grid (CVRG, http://cvrgrid.org), the Biomedical Informatics Research Network (BIRN, www.nbirn.net), and myGrid (www.

Published by the IEEE Computer Society

mygrid.org.uk) are working on systems for semantic integration of heterogeneous data types in the grid.

We address problems in two areas: processing of large digitized slides for analysis and semantic query of annotated images and image regions in a large image dataset. We present these solutions in the context of an integrated framework, as Figure 1 shows.

Our earlier work investigated the design and implementation of adaptive algorithms to efficiently process large image datasets.[1,2] Here, we investigate the influence of image content on these algorithms' behavior. We also introduce the framework's semantic knowledge-base component.

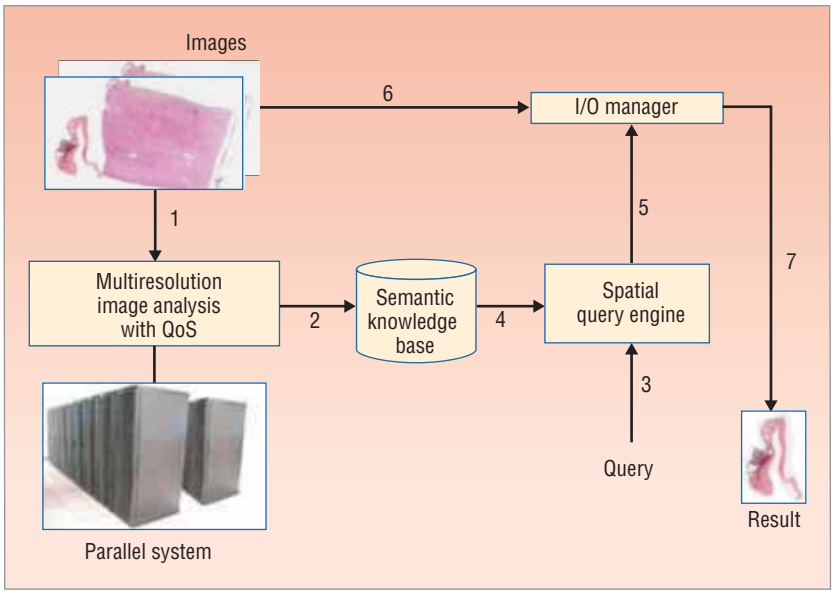


Figure 1. Image analysis and query framework. The framework consists of a component that implements adaptive runtime techniques for processing of images under user-provided QoS requirements and a component to support a knowledge base of annotated image data.

A MULTIRESOLUTION APPROACH TO IMAGE ANALYSIS

Digital scanners can generate images that are too large to fit in a machine's main memory and require long processing times for analysis. To alleviate these problems, we partition images into disjoint tiles, which serve as input to the data analysis system. Nevertheless, processing all tiles for a large image (for example, a 20-Gbyte image) at the highest resolution can still take several hours.

Our multiresolution processing approach reduces the overall processing time. In this approach, we process each tile at multiple resolutions. At each resolution, we apply a sequence of image-processing steps on the tile and compute a confidence value.

We establish the confidence level to quantitatively measure the likelihood that we'll correctly classify the image tile under investigation. Statistically, we measure it by the Mahalanobis distance from the feature point associated with the current image tile to the class centroid. Therefore, the higher the confidence level, the more likely we'll get a correct classification result. We consider a tile finalized if its assigned confidence value is greater than a user-defined threshold. Our goal is to classify image regions with an acceptable degree of confidence.

In general, the time required to process a tile increases with resolution because the tiles carry greater feature detail at higher resolutions. A tile's processing starts at the lowest resolution. If the confidence value at a given resolution lies below the user-defined threshold for that resolution, we process the tile at the next higher resolution. Otherwise, we consider the tile to be finalized at that resolution.

Under resource and time constraints, finalizing all tiles in an image might not be possible. Our objective is to process tiles in a way that satisfies constraints under user-imposed QoS requirements. For example,

- maximize the tiles' average accuracy (average confidence level) or maximize the number of finalized tiles within t time units; and
- given a minimum average accuracy or minimum number of finalized tiles, minimize the execution time.

The problem of meeting constraints with QoS requirements maps to a scheduling problem, in which we must determine the optimal order of tiles for processing. Under ideal conditions, we have full a priori knowledge of each tile's performance-versus-confidence value (PvC) behavior. That is, we can predict the confidence value of a tile's classification at different resolutions. In most practical cases, however, exactly estimating tiles' PvC characteristics is impossible. We propose heuristics to order tiles to achieve a good approximation of the execution order under ideal conditions.

Basic strategy

The basic approach (which we call *Rand*, for random) is to randomly choose one tile at a time and process it iteratively, starting from the lowest resolution and continuing until the tile is finalized. Once finalized, we randomly choose the next tile. Using this strategy, we can avoid starting from a region of the image where we must process all tiles at high resolutions to finalize them or where the confidence value per tile increases very slowly as resolution increases.

Related Work in Resource Management in Adaptive Applications

Researchers have studied dynamic quality-of-service-driven resource management in various adaptive applications, including visualization, e-commerce, object tracking, and multimedia processing. Prior efforts in this area have developed language and compiler support to meet the QoS demands.[1,2] Epiq,[3] ERDoS,[4] ActiveHarmony,[5] and Agilos[6] are examples of middleware-based runtime efforts to support adaptive applications and dynamic QoS-driven resource management.

Commonalities between earlier approaches and ours include the use of adaptation parameters that can be adjusted so that performance meets certain requirements. Our work differs from previous work in that we support QoS requirements not only with respect to time but also properties of the data and the output. Additionally, we modify the sequence in which data is processed to meet user requirements, thereby contributing a new dimension to application adaptation.

Much of the work in semantic information management is inspired by work in the description logics community, where the focus is on handling expressive logics and developing sound and complete reasoning algorithms. In recent years, there has been interest in storage, querying, and reasoning on assertions (also known as ABox reasoning). Systems such as Jena[7] and OWLIM[8] provide support for assertions represented in the Resource Description Framework (RDF, www.w3.org/RDF) or OWL (www.w3.org/TR/owl-ref). We use these engines as subsystems in our framework and support semantic queries with spatial predicates on large image datasets stored on disk.

References

1. V.S. Adve, V.V. Lam, and B. Ensink, "Language and Compiler Support for Adaptive Distributed Applications," *Proc. Workshop on Languages, Compilers, and Tools for Embedded Systems/Optimization of Middleware and Distributed Systems* (LCTES/OMDS 01), ACM Press, 2001, pp. 238-246.
2. W. Du and G. Agrawal, "Language and Compiler Support for Adaptive Applications," *Proc. 2004 ACM/IEEE Conf. Supercomputing (*SC 2004), IEEE CS Press, 2004.
3. J. Liu et al., "Epiq QoS Characterization," draft, research report, Univ. of Ill., Urbana-Champaign, Dept. of Computer Science, July 1997; http://citeseer.ist.psu.edu/liu97epiq.html.
4. S. Chatterjee, "Dynamic Application Structuring on Heterogeneous, Distributed Systems," *Proc. 13th Int'l Parallel Processing Symp. and 10th Symp. Parallel and Distributed Processing* (IPPS/SPDP 99), LNCS 1586, Springer, 1999, pp. 442-453.
5. C. Tapus, I.-H. Chung, and J.K. Hollingsworth, "Active Harmony: Towards Automated Performance Tuning," *Proc. 2002 ACM/IEEE Conf. Supercomputing* (SC 02), IEEE CS Press, 2002, pp. 1-11.
6. B. Li, W. Kalter, and K. Nahrstedt, "A Hierarchical Quality of Service Control Architecture for Configurable Multimedia Applications," *J. High-Speed Networks*, Dec. 2000, pp. 153-174.
7. K. Wilkinson et al., "Efficient RDF Storage and Retrieval in Jena2," *Proc. Very Large Databases (VLDB) Workshop Semantic Web and Databases*, 2003, pp. 131-150; www.cs.uic.edu/~ifc/SWDB/papers/Wilkinson_etal.pdf.
8. A. Kiryakov, D. Ognyanov, and D. Manov, "OWLIM—A Pragmatic Semantic Repository for OWL," *Proc. WISE Workshops*, LNCS 3807, Springer, 2005, pp. 182-192.

Region-based algorithm

Our region-based algorithm (*Hier*, for hierarchical) seeks to identify, as early as possible, the best candidate tiles for a given QoS requirement. By processing tiles in a specific order, as opposed to randomly selecting tiles, we might better satisfy the QoS requirements given the constraints. A priority queue helps us determine the best tiles for a given requirement. However, we observed that a nonlinear relationship exists between confidence value and resolution—higher resolutions don't always imply higher confidence values. This observation, coupled with the absence of a priori information about tiles' PvC behavior, means an algorithm operating at a tile granularity might not perform much better than Rand.[2]

To overcome this drawback, Hier leverages correlations between image tiles that share spatial proximity. This assumption follows from the expectation that spatially close tiles will have similar features and thus will exhibit similar PvC behavior. If we determine a good candidate tile for a given requirement, its neighbor tiles will likely be good candidates too. Hier quickly identifies and converges on such localized neighborhoods of tiles and processes them as early as possible.

Hier partitions images into M disjoint rectangular regions. Each region contains a set of tiles that are spatially adjacent to each other. Each such region has a corresponding entry in the priority queue. By suitably modifying the queue-insertion operation, we use the same queue structure and algorithm for different QoS requirements.[2] Initially, we randomly select a representative tile from each region and process it iteratively, starting from the lowest resolution, until it's finalized. We insert an entry into the queue for every region based on the confidence value for its representative tile.

In essence, we sample the image space to determine which regions contain the best candidate tiles. We randomly select a new tile from the region corresponding to the top queue entry. We then divide the region into two subregions using the median line between the newly selected tile and the old representative tile along the *X* or *Y* dimension. In this way, one subregion contains the newly selected tile and the other subregion has the old representative tile. We assign the other tiles in the region to subregions based on whether they're to the left (below) or right (above) of the median line. We create queue entries for each subregion and insert them into the queue using suitable insertion schemes.[2]

When all tiles in a region are finalized, we delete the entry from the queue. This process repeats until the queue is empty or the constraint is reached.

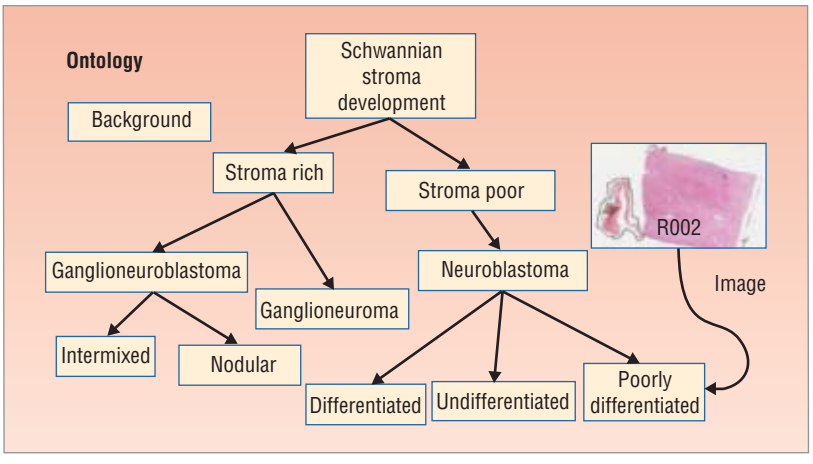


Figure 2. Image annotation using ontology. This ontology, based on the International Neuroblastoma Classification System, describes the morphological characterization of tissue used for prognosis.

Execution on parallel machines

Rand and Hier can benefit from a master-slave parallelization on distributed memory parallel machines, in which each node has local disks and is connected to the other nodes via a fast network. DataCutter,[3] a component-based middleware framework, provides the runtime support for parallel execution. DataCutter implements the application-processing structure as a set of components, or *filters*, that exchange data through a stream abstraction.

Our version of DataCutter uses a message-passing interface as the message-passing substrate. To perform distributed image analysis, we use a master console process and several worker processes implemented as filters. The console filter runs on a storage node that houses the image data. A user can submit workflows such as the neuroblastoma pipeline to the console, which launches workers on each node accordingly. The console distributes image tiles among the workers in a demand-driven fashion to dynamically adapt to changes in worker load. Worker processes on each node apply all steps of the workflow on the tiles.

The confidence value computed for every processed tile is fed back to the console. The console decides the order in which the workers process the tiles, so it controls the output quality. Our earlier work gives a detailed explanation of the system implementation.[2]

SEMANTIC QUERY OF IMAGE DATASETS

When the execution of the data analysis methods ends, different portions of the image will have been annotated, depending on the individual tiles' classifications. We organize these annotations into a knowledge base, which we use for semantic query of the image data for further analysis, data integration, and data mining of images.

Knowledge base

The knowledge base consists of images (pixel data), uniquely identified regions with a spatial extent, and annotations on these regions. Annotations can draw from terms defined in an ontology expressed in description logic, which defines classes, properties, and the relationships among them. The knowledge base also stores the application ontology. Figure 2 shows part of the ontology for the neuroblastoma classification application.

The knowledge base also has a spatial ontology that captures important spatial relationships between regions. The spatial ontology has a *region* class that represents spatial entities. Every region instance is associated with a polygon literal by a `HasPolygon` property in the annotations. The main properties are `Intersects`, `Contains`, and `Disjoint`.

The spatial ontology also specifies that `Intersects` and `Disjoint` are commutative properties and `Contains` is a transitive property. These properties don't explicitly occur in annotation data because there would be $O(n^2)$ spatial relationships between n regions, leading to an explosion of annotation information. However, queries should be able to use these spatial properties as if they were any other property-relating regions.

Figure 2 also shows example annotation data that can be expressed as:

- (R002, `HasPolygon`, Polygon (..))
- (R002, `OfImage`, $Image_0$)
- (R002, `type`, *Poorly Diff Neuroblastoma*)

Query model

To provide access to the knowledge base, we support a subset of SPARQL (www.w3.org/TR/rdf-sparql-query), a popular standard used to access Resource Description Framework (RDF) data. SPARQL has well-established semantics for expressing terminological constraints. To

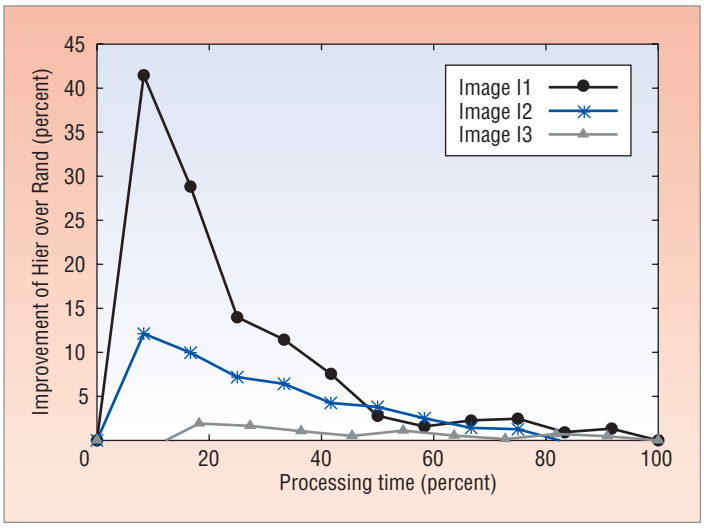


Figure 3. Percentage increase in number of finalized tiles for Hier as a function of processing time. We tested the algorithm using 32 nodes and a threshold of 0.7.

enable predicates on spatial information, we extend the SPARQL semantics with two expression constructs:

- (*x*, `intersects`, Polygon (..)) states that the polygon associated with the image region *x* should intersect the polygon literal; and
- (*x*, `intersects`, *y*) aims to restrict bindings in the query result to image regions *x* and *y* so that their polygons intersect.

Similarly, a query can contain expressions involving `contains` and `disjoint`. For example, the following query would retrieve pairs of regions that intersect and are in the same image, with one region of type *Differentiating* and the other of type *Stroma Poor*.

```
SELECT ?x ?y WHERE { ?x Intersects ?y.
?x type Stroma Poor . ?y type
Differentiating
?x OfImage ?z . ?y OfImage ?z }
```

Query execution

The knowledge-base system aims to support querying for regions of images that satisfy constraints. It has three main components.

The semantic *RDF store* serves assertions on elements (images and image regions) in the dataset. Our current implementation provides wrappers for Jena and OWL-Lite In-Memory engines.

The *I/O manager* provides the storage environment for image data. It oversees the image tiles, which it stores on disk in files. The I/O manager also stores metadata such as the mapping from regions to image tiles and tile location information (filename, offset, and size). The I/O manager uses this information during query execution to retrieve the pixel-level information.

The *spatial engine* implements the data structures and runtime support for query plan generation and query execution. It executes queries in two stages. In the first stage, it finds image regions that satisfy the query constraints. In the second stage, it retrieves pixel-level data corresponding to these regions.

Upon receiving a client query, the spatial engine generates a query plan. We've implemented many query-plan-generation algorithms, but we describe the simplest one for brevity.

The algorithm separates the query's spatial and nonspatial constraints, because the RDF store can't handle queries involving spatial constraints. It composes the nonspatial constraints as a SPARQL query and issues it to the underlying RDF store. The algorithm pipes and filters the results of the nonspatial query through a network of spatial operators. We also investigated optimizations such as vertical partitioning of queries. This stage resulted in a set of identifiers corresponding to regions that satisfy the query.

In the next stage of query execution, the spatial engine sends the region IDs to the I/O manager. For each region in the result set, the I/O manager finds the list of tiles that intersect or are contained in the region. A tile can intersect multiple regions. A `DISTINCT` operation removes duplicate tiles. The I/O manager then reads the tiles from disk and returns them to the client.

EXPERIMENTAL RESULTS

We performed a series of experiments to evaluate our techniques.

Accuracy-performance tradeoff

We performed our experiments using a cluster of 32 AMD Dual 250 Opteron nodes equipped with 8 Gbytes of memory, interconnected by an Infiniband and a 1-Gbyte-per-second Ethernet network. Each node has 2 × 250-Gbyte Serial Advanced Technology Attachment (SATA) disks installed locally, joined into a 437-Gbyte redundant array of independent disks 0 (RAID 0) volume. The maximum disk bandwidth per node was around 35 Mbytes per second for sequential reads and 55 Mbps for sequential writes.

Our dataset consisted of three images (*I*1, *I*2, and *I*3) ranging in size from 8 to 22 Gbytes. We partitioned the images into tiles of 512 × 512 pixels each and maintained tile data at four resolutions ($r1 < r2 < r3 < r4$). We characterized the images by differences in their feature content.

We tested our hypothesis that our region-based algorithm would yield better responses to user QoS requirements than the basic algorithm.

User requirement: Maximize number of finalized tiles within *t* units of time. Figure 3 shows the percentage increase in the number of classified tiles that our region-based algorithm (Hier) achieved over the basic algorithm (Rand) as a function of processing time. (The threshold was 0.7 for all resolutions.)

For images *I*1 and *I*2, Hier finalized significantly more tiles than Rand during the first half of the processing. This is because, given a time constraint and set of worker nodes, Hier rapidly converges on regions of tiles more likely to be finalized at lower resolutions. By choosing these tiles early in the analysis phase, Hier provides more information within the same time, letting the expert make a better diagnosis.

Hier processes the remaining tiles later because it places them at the bottom of the queue. This explains the drop in improvement toward the later processing stages. For *I*3, Hier performed no better than Rand at this threshold. We attribute such differences in algorithm behavior across images to image content.

Table 1 presents our observations on tile finalization per image. The table lists the percentage of tiles for each image that are finalized at a given resolution.

For *I*1, the number of tiles finalized at lower and higher resolutions is roughly equal. However, for *I*3, 75 percent of the tiles are finalized only at *r*4. In the extreme case, in which all tiles are finalized only at the highest resolution, Hier performs no better than Rand. Processing time for each tile is roughly the same. Hence, the tile processing order won't influence performance.

For *I*3, Hier provides less improvement because of the limited set (25 percent) of candidate tiles that can be assigned high priority. Therefore, images exhibit marked differences in the resolutions at which their tiles are finalized. This finding makes a strong case for incorporating factors such as image content into the heuristic. We'll investigate this idea in future work.

User requirement: Maximize average confidence value within *t* units of time. The results for this requirement mirror the trends reflected in Figure 3.

For *I*1, Hier achieved up to a 40-percent improvement in average confidence value over Rand. This is because we modified Hier's queue-insertion scheme so regions of tiles that achieved highest confidence gains at lower resolutions were always at or near the top of the queue and hence were processed before other tiles.

We use confidence values as a proxy for analysis result accuracy. In general, a researcher will want to obtain the maximum accuracy possible. Sometimes, a moderate increase in confidence value will lead to much better classification. Other times, an increase will be less effective. Future work will investigate the overall impact of confidence values on the end results—for example, more accurate tissue classification.

We again observed diminished improvement for *I*3.

Table 1. Influence of image content at threshold 0.7.

Image	Percent of tiles finalized at resolution			
	*r*1	*r*2	*r*3	*r*4
*I*1	25.9	19.1	11.1	43.8
*I*2	18.1	11.6	10.9	59.3
*I*3	10.5	7.7	6.6	75.2

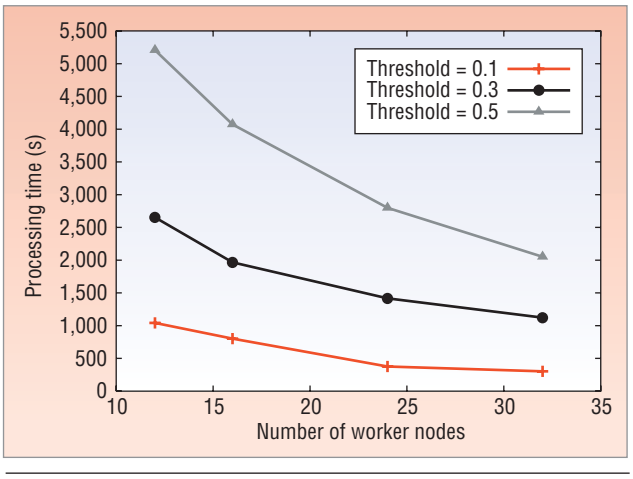


Figure 4. Hier algorithm linear scalability. Increasing the number of worker nodes reduces the amount of time needed to reach an average confidence threshold.

We can attribute the different algorithm behavior across images to image content.

Scalability tests. We tested our Hier algorithm's scalability by varying the resource pool used for processing *I*1, using from 12 to 32 worker nodes.

Figure 4 shows the linear scalability. As we double the number of nodes, Hier achieves a given average confidence threshold in roughly half the time.

We also tested scalability with respect to image size. Given two images with similar content (tile finalization characteristics) but different sizes (22 Gbytes versus about 11 Gbytes), and using the same set of resources, processing the larger image for a given QoS requirement took about twice as long as the smaller image.

These linear scalability results justify extending our experimental methodologies to handle terabyte-size images in the future.

Semantic query results

For our semantic query experiments, we used an AMD Dual 250 Opteron node with 8 Gbytes of DDR400 RAM and a 3.5-Tbyte SATA array (RAID 5). We stored both the images and annotations on the same disk. The disk had a sequential read bandwidth of about 295 Mbps. We used between 4 and 31 images in these experiments, each 10–20 Gbytes in size.

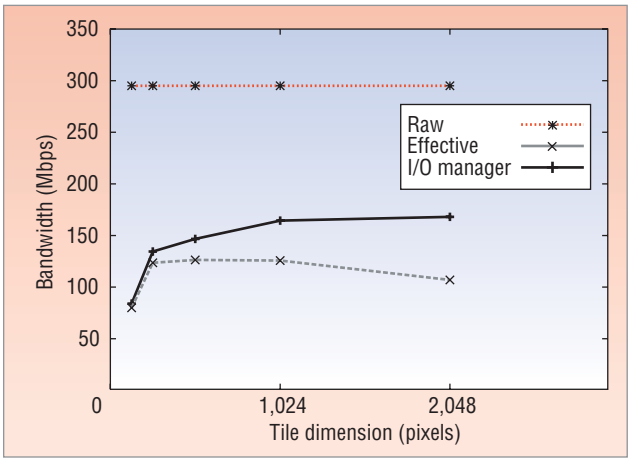


Figure 5. End-to-end bandwidth performance for different tile sizes. The system achieves about 41.7 percent of the peak disk bandwidth.

The spatial query engine identifies regions satisfying a query. The I/O manager proceeds to identify the tiles to be retrieved and then issues read requests to the underlying file system.

We first evaluated the spatial query engine's scalability using annotations on 6, 15, and 31 images. These datasets contained 317, 1,047, and 2,247 regions. The average execution times of three representative queries over many runs on these datasets were 637 ms, 5,639 ms, and 26,101 ms, respectively. This quadratic increase occurred because, for k distinct regions returned by the nonspatial part of the query, the spatial engine performs (k^2) pairwise checks for intersection, containment, and disjointness, as specified in the nonspatial part of the query.

We also evaluated the performance of the second stage of query execution—that is, retrieval of tiles corresponding to the query. Figure 5 summarizes the results.

We store an image as 2D tiles on disk using Hilbert-curve declustering. Larger tile sizes result in fewer tiles per image (and per region), which reduces the amount of metadata that the I/O manager maintains. It also reduces the number of I/O requests to the underlying disk for a query. However, larger tiles result in the retrieval of wasteful information from disk that must be filtered out. Smaller tile sizes can cause underutilization of the disk but also result in finer retrieval.

We calculate the I/O manager bandwidth and the effective bandwidth as:

$$\text{I/O manager bandwidth} = \frac{TimeSize * N_{tiles}}{Time}$$

$$\text{Effective bandwidth} = \frac{\sum_{Region\ R} Size_R}{Time}$$

Figure 5 shows the trend for these measures over three queries in a four-image dataset. Increasing tile size helps the I/O manager bandwidth but only up to a point. Increasing tile size beyond this point affects the effective bandwidth because we must filter excess pixels along the boundaries. The figure shows that the system achieves about 41.7 percent of the peak disk bandwidth.

Our experiments showed that, on average, the first stage of query execution takes less than 2 percent of the end-to-end query-execution time. However, this characteristic is application and dataset specific. Having numerous interesting regions per image would increase processing time in the RDF store. It would also cause a quadratic increase in the spatial query engine's execution time. A very complex underlying application ontology would likewise increase execution times in the RDF store. Thus, dataset-specific factors can affect the total execution time and the percentage contribution to execution time of the two stages.

In our ongoing work, we're investigating techniques to implement the knowledge base on a cluster system to achieve better query performance and scale to larger datasets. We also plan to optimize the detection algorithms for stroma and grade of differentiation subcomponents of our computer-aided prognosis system. This work will entail investigating the impact of these optimizations along with different types of quality-of-service requirements on system efficiency.

Although this work focuses on classifying neuroblastoma, our system has applicability in other biomedical imaging studies. Multiscale analysis of tumor microenvironments is one example. In these studies, clinicians integrate phenotypic characteristics (shape, location, volume, and so on), obtained from digitized microscopy images, with genomic information to investigate biomarkers at different scales. Such studies can benefit from efficient processing of large image data and support for querying image and genomic information via spatial annotations and genome ontologies.

Acknowledgments

The US National Science Foundation partly supported this work under grants CNS-0403342, CNS-0426241, ANI-0330612, CNS-0203846, ACI-0130437, CCF-0342615, CNS-0615155, and CNS-0406386. The Ohio Board of Regents, through grants BRTTC BRTT02-0003 and AGMT TECH-04049; and the US National Institutes of Health, through grants R01 LM009239, 79077CBS10, and P20-EB000591, also supported this work. We thank Dr. Hiroyuki Shimada and the Columbus Nationwide Children's Hospital for providing the slides and images used in our work.

References

1. J. Kong et al., "Computer-Aided Grading of Neuroblastic Differentiation: Multiresolution and Multi-Classifier Approach," *Proc. IEEE Int'l Conf. Image Processing* (ICIP 07), IEEE CS Press, 2007, pp. 525-528.
2. V. Kumar et al., "Performance vs. Accuracy Tradeoffs for Large-Scale Image Analysis Applications," *Proc. IEEE Int'l Conf. Cluster Computing* (Cluster 07), IEEE Press, 2007.
3. M.D. Beynon et al., "Distributed Processing of Very Large Datasets with DataCutter," *Parallel Computing*, Oct. 2001, pp. 1457-1478.

Vijay S. Kumar *is a PhD candidate in computer science at the Ohio State University, where he is a research assistant in the Department of Biomedical Informatics. His research interests include algorithms for high-performance computing and middleware development and runtime performance optimizations for workflow-based analysis of very large image data using parallel and distributed computing resources. Kumar received an MSc in chemistry from the Birla Institute of Technology and Science, Pilani, India. Contact him at vijayskumar@bmi.osu.edu.*

Sivaramakrishnan Narayanan *is a PhD candidate in computer science at the Ohio State University, where he is a research assistant in the Department of Biomedical Informatics. His research interests include optimizing access to large-scale scientific data on the grid and data-intensive computing and scheduling in parallel and distributed environments. Narayanan received a BE in computer science from the Birla Institute of Technology and Science, Pilani, India. Contact him at krishnan@bmi.osu.edu.*

Tahsin Kurc *is a research assistant professor in the Department of Biomedical Informatics at the Ohio State University. His research interests include high-performance data-intensive computing, runtime systems for efficient storage and processing of very large scientific datasets, and scientific visualization on high-performance machines. Kurc received a PhD in computer science from Bilkent University, Turkey. Contact him at kurc@bmi.osu.edu.*

Jun Kong *is a PhD student in the Department of Electrical and Computer Engineering at the Ohio State University. His research interests include computer vision, machine learning, and pathological image analysis. Kong received an MS in electrical engineering from Shanghai Jiao Tong University, Shanghai. Contact him at kongj@bmi.osu.edu.*

Metin N. Gurcan *is an assistant professor in the Department of Biomedical Informatics at the Ohio State University. His research interests include image analysis and understanding and computer vision with applications to biology and medicine. Gurcan received a PhD in electrical and electronics engineering from Bilkent University, Turkey. He is a member of RSNA, SPIE, and a senior member of the IEEE. Contact him at gurcan@bmi.osu.edu.*

Joel H. Saltz *is a professor and chair of the Department of Biomedical Informatics, a professor in the Department of Computer Science and Engineering, the Davis Endowed Chair of Cancer at the Ohio State University, and a senior fellow of the Ohio Supercomputer Center. His research interests are in the development of systems software; databases and compilers for the management, processing, and exploration of very large datasets; medical informatics systems; and grid and high-performance computing in biomedicine. Saltz received an MD and a PhD in computer science from Duke University. Contact him at joel.saltz@osumc.edu.*

COVER FEATURE

Hardware Technologies for High-Performance Data-Intensive Computing

Maya Gokhale, Jonathan Cohen, Andy Yoo, and W. Marcus Miller
Lawrence Livermore National Laboratory
Arpith Jacob, Washington University in St. Louis
Craig Ulmer, Sandia National Laboratories
Roger Pearce, Texas A&M University

Data-intensive problems challenge conventional computing architectures with demanding CPU, memory, and I/O requirements. Experiments with three benchmarks suggest that emerging hardware technologies can significantly boost performance of a wide range of applications by increasing compute cycles and bandwidth and reducing latency.

As the amount of scientific and social data continues to grow, researchers in a multitude of domains face challenges associated with storing, indexing, retrieving, assimilating, and synthesizing raw data into actionable information. Combining techniques from computer science, statistics, and applied math, *data-intensive computing* involves developing and optimizing algorithms and systems that interact closely with large volumes of data.

Scientific applications that read and write large data sets often perform poorly and don't scale well on present-day computing systems. Many data-intensive applications are data-path-oriented, making little use of branch prediction and speculation hardware in the CPU. These applications are well suited to streaming data access and can't effectively use the sophisticated on-chip cache hierarchy. Their ability to process large data sets is hampered by orders-of-magnitude mismatches between disk, memory, and CPU bandwidths.

Emerging technologies can improve data-intensive algorithms' performance, at reasonable cost in development time, by an order of magnitude over the state of the art. Coprocessors such as graphics processor units (GPUs) and field-programmable gate arrays (FPGAs) can significantly speed up some application classes in which data-path-oriented computing is dominant. Additionally, these coprocessors interact with application-controlled on-chip memory rather than a traditional cache.

To alleviate the 10-to-100 factor mismatch in bandwidth between disk and memory, we investigated an I/O system built from a large, parallel array of solid-state storage devices. While containing the same NAND flash chips as USB drives, such I/O arrays achieve significantly higher bandwidth and lower latency than USB drives through parallel access to an array of devices.

To quantify these technologies' merits, we've created a small collection of data-intensive benchmarks selected from applications in data analysis and science. These benchmarks draw from three data types: scientific imagery, unstructured text, and semantic graphs representing networks of relationships. Our results demonstrate that augmenting commodity processors to exploit these technologies can improve performance 2 to 17 times.

COPROCESSORS

Coprocessors designed for data-oriented computing can deliver orders-of-magnitude better performance than general-purpose microprocessors on data-path-centric compute kernels. We evaluated the benefits of two coprocessor architectures: graphics processors and reconfigurable hardware.

Published by the IEEE Computer Society

Graphics processors

The GPU, a commodity product that accelerates the rendering of images to the display, has been highly optimized for the computer-game industry to offer realistic 3D rendering of fast-moving scenes. It offers a high degree of data-parallel operation on floating-point data, up to hundreds of Gflops with a single PCI-E board. Responding to the need for general-purpose use of graphics hardware, programming interfaces such as Nvidia's Compute Unified Device Architecture (www.nvidia.com/object/cuda_home.html) expose the processing cores to parallel GPU algorithms.

In this work, we used the Nvidia GeForce 8800 GTX GPU. In contrast to previous-generation fixed-function graphics pipelines, the 8800 has an array of 128 IEEE 754-compliant scalar floating-point units clocked at 1.35 GHz and grouped into clusters of 16. It has 768 Mbytes of RAM with a 384-bit memory interface and 86.4-GBps memory bandwidth. Our GPU benchmark is written in Cg, which compiles into an OpenGL-supported assembly code and is vendor-neutral. The Nvidia card is attached to a 3.0-GHz dual-core Xeon processor via a PCI-E 16x slot.

Reconfigurable hardware

FPGA-based reconfigurable hardware provides direct execution of an algorithm. Unlike fixed-function hardware, reconfigurable logic can be reprogrammed an unlimited number of times, allowing different algorithms to execute on the same device.

The FPGA is a complex system-on-a-chip that combines processors, on-chip RAM, specialized arithmetic units, and reconfigurable logic. When an algorithm kernel is mapped onto its hardware resources, an FPGA can achieve a 10-to-100 times speedup over equivalent software. Another advantage of FPGAs is that, because the device is often used to communicate to the data source, application-specific logic can be inserted into a pipelined data stream. FPGAs are already available in the marketplace for data-intensive computing tasks such as bioinformatics, text processing, and relational databases.

In these experiments, we used the XtremeData XD1000 system, shown in Figure 1, which features a dual-core 2.2-GHz Opteron CPU and an Altera Stratix EP2S180F1508-C3 FPGA. Each processor has 4 Gbytes of dynamic RAM (DRAM); the FPGA additionally has 4 Mbytes of static RAM (SRAM). The Opteron and FPGA are on a dual-socket motherboard and communicate via a noncoherent HyperTransport (HT) link, with a bidirectional peak bandwidth of 1.6 GBps. The actual bandwidth achieved depends on the FPGA clock speed. Bandwidth measurements of HT communication between a test FPGA design and the Opteron showed a rate of roughly 500 MBps. Our FPGA application is written in VHDL and compiled with the Altera tool chain.

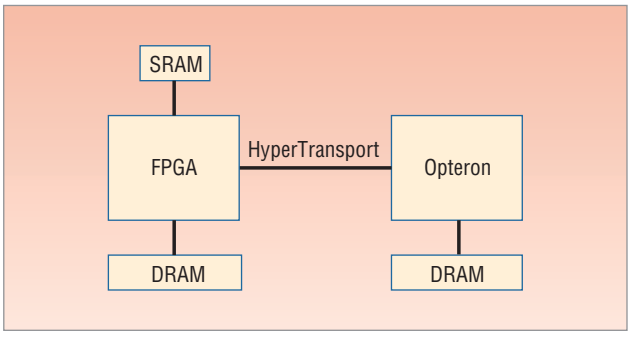


Figure 1. XtremeData XD1000 architecture. The system features a dual-core 2.2-GHz Opteron CPU and an Altera Stratix EP2S180F1508-C3 FPGA that communicate via a noncoherent HT link. Each processor has 4 Gbytes of DRAM; the FPGA additionally has 4 Mbytes of SRAM.

SCIENTIFIC IMAGERY ANALYSIS

The Large Synoptic Survey Telescope (www.lsst.org) will be a ground-based 8.4-meter, 10^2-degree-field device sited on a mountain in Chile, and is expected to start producing astronomical data in 2012. Processing LSST data will be extremely challenging. The raw data from the 3-Gpixel charge-coupled device camera is collated at a rate of 500 MBps and must be preprocessed in real time. Lawrence Livermore National Laboratory (LLNL) is a member of the LSST Corporation and contributes to the project's camera design and data management.[1]

Lanczos resampling filter

Our image-processing benchmark, the Lanczos resampling filter, is derived from SWarp,[2] an application used in parts of the LSST data-processing pipeline. SWarp transforms images from the telescope to the sky template, making it possible to compare a newly acquired image with the associated sky template section and discover anomalies such as supernova explosions and gamma ray bursts.

The benchmark factors out a computationally expensive and fundamental piece of the SWarp functionality for implementation: gray-scale image resampling. The input data is a gray-scale raster image (row-major order) with 8 or 16 bits per pixel. The output is a gray-scale image with 8, 16, or 32 bits per pixel. The typical SWarp execution for LSST includes resampling a 16-bit input into a 32-bit, floating-point output.

It took us approximately one month to develop the Lanczos filter, including profiling SWarp to select the benchmark, designing, writing, debugging, and performance tuning.

Computational kernels

For each output pixel, SWarp applies a filter kernel that takes a weighted combination of the input pixels, specifically a Lanczos filter, to combine either 16 (4 × 4), 36 (6 × 6), or 64 (8 × 8) input pixels to generate each

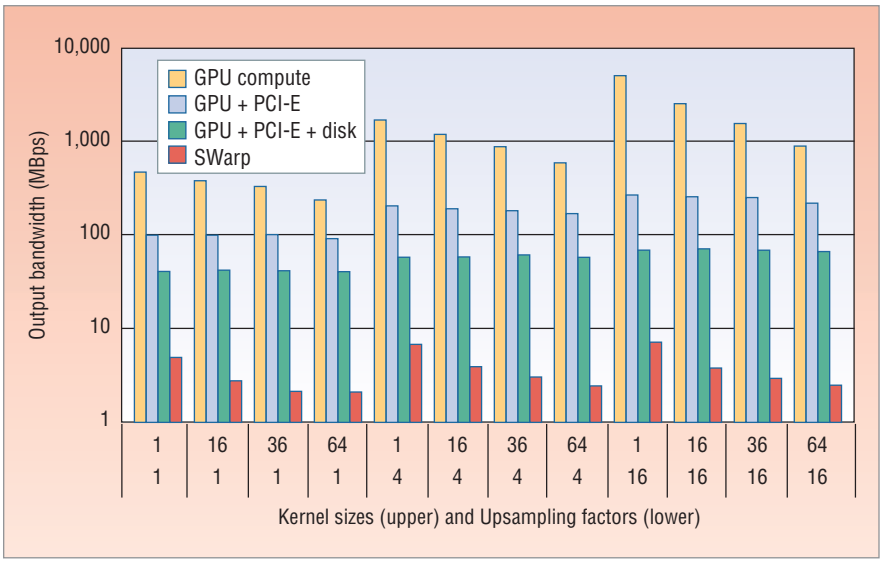


Figure 2. Image resampling bandwidths. The graph shows the rate of generated output pixels in logarithmic scale as a function of upsampling scale factor and kernel size.

output pixel. The Lanczos filter kernel is convolved with the input pixels to generate the output pixels. Each input pixel has a location (x, y) relative to the projection of an output pixel into the input image. The weight used for that input pixel's contribution to the output pixel is

$$L_k(x,y) = \frac{k\sin(\pi x)\sin(\frac{\pi}{k}x)}{\pi^2 x^2} \times \frac{k\sin(\pi y)\sin(\frac{\pi}{k}y)}{\pi^2 y^2}, \quad |x| < k, |y| < k$$

where k can be 2, 3, or 4 (the so-called Lanczos2, Lanczos3, and Lanczos4 kernels), using 16, 36, or 64 input pixels per output pixel, respectively. We've written four resampling codes, one for each of these Lanczos convolution kernels, plus a simple nearest-neighbor filter, using NVIDIA's Cg programming language.

The Cg compiler compiles the code down to ARB Fragment Program code, an OpenGL-supported assembly code for the fragment programs on current GPUs that is vendor neutral. Because we're benchmarking the recent NVIDIA 8800 GTX, which is a scalar GPU, it's not necessary to optimize the computation for the more traditional four-way vectorized GPU. The philosophy behind the NVIDIA 8XXX series is to provide more scalar cores (128) rather than fewer vector cores, facilitating full utilization.

I/O methods

The SWarp implementation, which employs memory-mapped I/O for large files, processes output pixels in raster order, matching the order of the data in the disk file; consequently, output is purely streaming, whereas input requires more random access. However, to better leverage the GPU's significant processing power, memory bandwidth, and data-caching capability, we programmed the Lanczos filter to produce the output data in 2D *tiles*. The benchmark generates each output pixel in the tile in a separate execution thread on the GPU.

Data flows from disk to main memory, main memory to GPU memory, GPU memory back to main memory, and main memory back to disk. In each case, the program specifies the data to be moved from source to target location as a 2D rectangle. If this rectangle's width is the same as that of the source and target, the data copy can occur as a single, contiguous stream. However, if the widths differ, the data copy must occur in a *strided* fashion, contiguous only at the level of individual rows.

Strided copies incur additional overhead, the largest of which are disk seeks if the striding occurs on the disk-file end of a copy between main memory and disk. Striding can also incur minor overhead when it occurs between main memory and GPU memory. Note also that, whereas support for strided copies is generally already built into GPU drivers, it's not built into the operating system's file I/O interface and therefore must be explicitly coded into the benchmark using some combination of reads, seeks, and buffering.

The computation proceeds one output tile at a time, in row-major order. For each output tile, the program determines the input data required and loads that rectangle of data from main memory to the GPU over the PCI-E bus. The filter is then applied and the program reads the data back from the GPU to main memory over the PCI-E bus.

Performance evaluation

We used the Lanczos filter to assess the potential for accelerating LSST image-processing tasks using GPUs. Our investigation comprised numerous executions of the benchmark and the original SWarp application. Figure 2 plots performance, as measured by the rate of data output in MBps. For each execution, we chose a scale factor, which is the ratio of output pixels to input pixels, and a kernel size, which is the number of input pixels combined to generate each output pixel.

GPU compute. The first column indicates the pure computational rate of our GPU-based kernel implementation. In this execution, we kept the image resident in memory and disabled data transfers to and from the GPU. In lieu of data download from the GPU, we flushed

the pipeline to allow accurate measurement of the total compute time.

Predictably, this rate decreases with increasing kernel size. What might be more surprising is the dramatic increase in compute performance as the scaling factor increases. This is attributable to the highly effective 2D data (texture) cache on the GPU. As the scale factor increases, the filter repeatedly uses the same neighborhood of input pixels to generate a neighborhood of output pixels.

The GPU compute column most closely relates to peak performance of the graphics hardware, analogous to peak floating-point performance from a general-purpose CPU. It exceeds 1 GBps for some kernel-size and scaling-factor combinations.

GPU + PCI-E. The second column reports performance when taking into account data transfers to and from the GPU over the PCI-E bus. The time to read data back from the GPU to the host generally dominates because

- there's more data to read back from the GPU than to send, due to the upsampling factor and the doubling in bits (from 16 to 32) per pixel, and
- raw download rates are typically slower than upload rates.

Using a commodity-parallel architecture such as the GPU can provide substantial speedup over a single-CPU implementation.

Performance exceeds 100 MBps for all combinations of kernel size and scale factor, with the best performance at 259 MBps for a kernel of one and scale factor of 16.

GPU + PCI-E + disk. The third column reports the benchmark's overall performance on out-of-core data, using a 2D strip representation to load and store the data. The maximum output achieved is 73 MBps. For the lowest scale factor of 1, this significantly decreases to around 40 MBps, presumably because the filter requires significant input data as well as output data bandwidth, and the input and output files are sharing the same two-disk redundant array of independent disks (RAID). Experimentally disabling reading of input data from disk increased performance to around 70 MBps.

SWarp. To measure real SWarp performance, shown in the fourth column, we eliminated as many computations as possible that are extraneous to the benchmark computation. In particular, we configured the resampling to perform in the application's PIXEL coordinate system, thus avoiding use of any of the dozens of more complex astrometric coordinate transformations possible. The range of speedups of the GPU implementation compared to SWarp ranged from 9 to 30 times.

The need for oversampling and high-quality kernels to produce excellent results argues that the tests with larger scale factors and speedups are most applicable in practice. SWarp can also be executed in parallel on multiprocessor and multicore machines, and it exhibited a nearly two-times speedup over the listed results when run on the test machine's two processors.

Conclusions. The Lanczos filter within the original SWarp implementation is purely CPU-bound; using the GPU completely eliminated the CPU bottleneck. For images that can fit into main memory such as small tiles of LSST imagery, the DRAM ↔ PCI-E communications bandwidth dominates performance. For out-of-core imagery such as large tiles or the complete sky image, the Lanczos filter is I/O-bound. The benchmark thus demonstrates that using a commodity-parallel architecture such as the GPU can provide substantial speedup over a single-CPU implementation, up to the point where disk I/O becomes the primary bottleneck. For the current test system, the I/O limited the speedup to 30 times.

UNSTRUCTURED TEXT PROCESSING

Language classification is an increasingly important task on the World Wide Web, where a growing number of documents are in a language other than English. It finds uses in search-engine indexing, spam-filtering heuristics, information retrieval, text mining, and other applications that apply language-specific algorithms. Language classification is a key step in processing large document streams and is data-intensive.

While many solutions run on general-purpose processors, the growth of document sets has far surpassed microprocessor improvements afforded by Moore's law. FPGA-based systems offer an alternative platform that enables the design of highly parallel architectures to exploit data parallelism available in specific algorithms.

Language classification using n-grams

A well-known technique to classify a text stream's language is to create an *n-gram profile* for the language.[3] An n-gram is a sequence of characters of length exactly *n*. N-grams are extracted from a string or a document by a sliding window that shifts one character at a time. An n-gram profile of a set of documents is the *t* most frequently occurring n-grams in the set. The probability that an input document is in a particular language is determined by the closeness of the document's n-gram profile to the language profile.

In this work, we used Bloom filters[4] to improve an existing FPGA-based n-gram text categorizer, the HAIL (hardware-accelerated identification of languages) architecture,[5] to build a highly scalable design. We implemented our design in VHDL on the XtremeData XD1000 development system. The time to select, implement, debug, and performance-tune the benchmark was approximately six weeks.

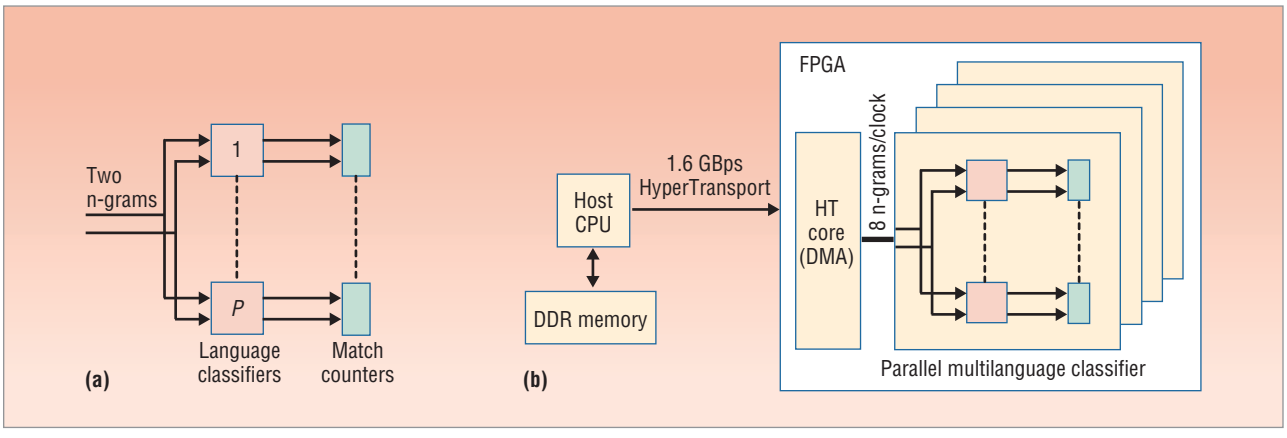


Figure 3. Parallel Bloom-filter-based n-gram classifier hardware. (a) Multilanguage classifier. (b) Parallel multilanguage classifier on the XtremeData system.

An n-gram-based classifier builds a document's n-gram stream by taking a sliding n-character window across the text. It generates a language's n-gram profile by taking the top t n-grams—we selected 5,000 for our benchmark—from a representative sample of documents in that language. The classifier selects the language of an unknown document by comparing its n-gram profile to all the language profiles and selecting the language with the highest match count.

Our design accomplishes n-gram tabulation using parallel Bloom filters as a probabilistic test for set membership. It streams the n-gram to k parallel hash functions, whose outputs address k separate $1 \times m$ bit memories. To add a new n-gram, the design sets each addressed bit to 1. To test membership, the design reads all addressed bits, and finds an n-gram present in the language profile if all locations have 1 at that address. This technique can give false positives due to the hash function but won't generate false negatives. Our parallel Bloom filter implementation uses 64 Kbits (k = 4 hash functions, m = 16-Kbit memories). The average accuracy rate for this configuration is 99.45 percent.

Our design exploits parallelism at multiple levels. First, there is a separate memory for each hash function, so that all the hash functions can be applied in parallel. Second, the memories are dual-ported, so that two different n-grams can be tested in a single clock cycle, as Figure 3a shows. Finally, our design duplicates p classifiers four times, enabling eight n-grams to be processed every clock cycle, as Figure 3b shows.

Performance evaluation

We measured our implementation's performance using the JRC-Acquis Multilingual Parallel Corpus, v3.0 (http://wt.jrc.it/lt/Acquis).[6] This corpus is the body of European Union law applicable to the EU member states available in 22 European languages. We used 10 languages: Czech, Slovak, Danish, Swedish, Spanish, Portuguese, Finnish, Estonian, French, and English. For our tests we parsed a subset of the corpus containing only the documents' text bodies.

There was an average of 5,700 documents for each language, with an average of 1,300 words per document. The average size of a single language corpus was 48 Mbytes, and an individual document averaged 10 Kbytes. We used 10 percent of the corpus as the training set for each language and tested the classifier on the remaining documents. To measure the system's throughput, we used the configuration with k = 4, m = 16 Kbits accepting eight n-grams per clock, and running at a clock speed of 194 MHz.

We measured the wall clock time for the transfer of documents and receipt of results in memory. The measured time didn't include the Bloom filter programming time, which is a setup cost that can be amortized over large runs. Also, we didn't include the preprocessing step to generate the n-gram profiles in the timing because it's a one-time cost prior to classification.

Figure 4 compares various throughput rates for our FPGA-based design as well as software only.

FPGA only. The theoretical rate at which our design can accept document n-grams is 194 MHz × 8 = 1,552 million n-grams per second. Because each n-gram corresponds to a byte in the input stream, our design can perform language classification at a peak rate of 1.4 GBps. This is well within the 1.6-GBps bandwidth provided by the HT bus. However, the HT core of the XtremeData machine we used achieves only a maximum of 500 MBps and so limits the practical performance that our design realizes.

FPGA + HT and FPGA + HT + I/O. By using asynchronous direct memory access (DMA) in a multithreaded communications interface to transfer data and control between the Opteron CPU and FPGA, we achieved a throughput rate of 470 MBps. When we included the time to read documents from storage in the measurements, the throughput was 93 MBps on a NAND flash drive and 55 MBps on a local Serial ATA (SATA) disk.

Compared to HAIL, our equivalent FPGA + HT hardware runs 1.45 times faster on 10 languages. While HAIL can classify up to 255 languages at this rate, our hardware is limited to between 10 and 30 languages by the number of on-chip embedded RAMs. The advantage of our design is that it's both flexible and scalable, allowing the designer to trade off the number of hash functions with memory size. Also, in contrast to HAIL, it uses only on-chip memory, eliminating the need for specific external memory configurations.

Software. To compare the performance of our Bloom-filter-based classifier to that of software, we measured the system throughput of mguesser (www.mnogosearch.org/guesser), an optimized version of the n-gram-based text categorization algorithm. We ran mguesser on a 2.4-GHz AMD Opteron processor with 16 Gbytes of memory, using 10 languages for identification and 81-Mbyte-size documents.

Mguesser's average throughput was 5.5 MBps, which doesn't include the time to read the documents from disk; these were cached in memory before a timing run. In comparison, our FPGA + HT implementation is 85 times faster, and our FPGA + HT + I/O implementation is 17 times faster, than mguesser's compute-only time.

Conclusion. As with the image-resampling benchmark, this experiment showed that coprocessor technology can eliminate the CPU bottleneck of compute-intensive data analysis applications at a reasonable cost in development time.

FLASH MEMORY I/O DRIVE

To effectively use coprocessors, the memory and I/O systems must deliver and store data at the coprocessor's rate. Many data-intensive algorithms interact with large data sets that can't be stored cost-effectively in main memory. While coprocessors with application-specific caches can accelerate in-memory computation, the bandwidth gap between volatile and persistent memory can greatly diminish the coprocessor advantage, causing I/O operations to dominate many such algorithms' runtime.

Disk storage has dramatically increased in capacity during the past decade, achieving a 60 to 100 percent compound annual growth rate between 1996 and 2004. However, latency and bandwidth have lagged, and access times for rotating media are likely to stay flat for the foreseeable future. In addition, large collections of disk drives incur costs in reliability and power usage.

Flash architectures

Strong consumer demand for music products with embedded mass storage has driven memory manufacturers to significantly lower the price and increase the capacity of flash memory chips. Flash memory is a form of nonvolatile storage in which charge trapped on a floating-gate transistor represents a data value.

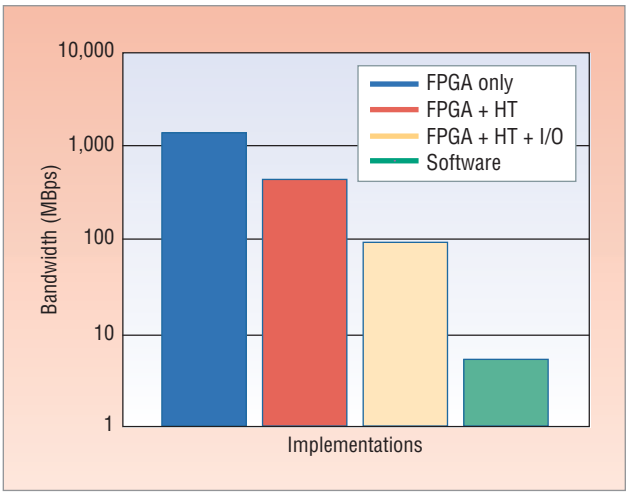


Figure 4. Language classification bandwidths. The graph compares various rates of n-gram processing in logarithmic scale of our FPGA-based design as well as software.

Charge is initially deposited on each transistor in a section of the chip when an erase operation is performed, then removed when a write operation of a logical "0" occurs. Once a gate's charge is removed, it can't be replenished until an erase operation takes place. Given that erase operations can only be performed on large regions (128 Kbytes) of a chip at a time, a flash device can be viewed as an erasable form of write-once, read-many (WORM) storage.

Memory vendors arrange floating-gate transistors in large arrays of either NOR or NAND gate structures. While NOR flash chips provide random byte access to the user, NAND flash chips feature higher capacities and better programming times.

The drawback of NAND flash chips is that they operate on data in page-sized quantities (2 Kbytes). Each time a hardware controller issues a read request to a flash memory chip, the flash memory chip must locate the corresponding page in its storage array and transfer the entire page to a special buffer before the data can be moved off chip. This internal data transfer can take as long as 25 µs. Once the transfer completes, the data can be streamed out of the flash chip sequentially at a rate of 40 MBps—that is, 50 µs for a full page.

To increase yields, memory vendors typically pack multiple flash dies on a single chip. For example, a 16-Gbit flash chip from Micron Technologies consists of a stack of four 4-Gbit die. To increase capacity for a given printed circuit board (PCB) footprint, multiple flash chips can be stacked on top of each other using simple spacer devices. Consequently, a single socket on a PCB can house a stack of 16 or more flash memory dies. This vertical parallelism provides an opportunity to hide access time, as each die can be issued its own memory transaction, provided that no two dies simultaneously assert data on the flash pins.

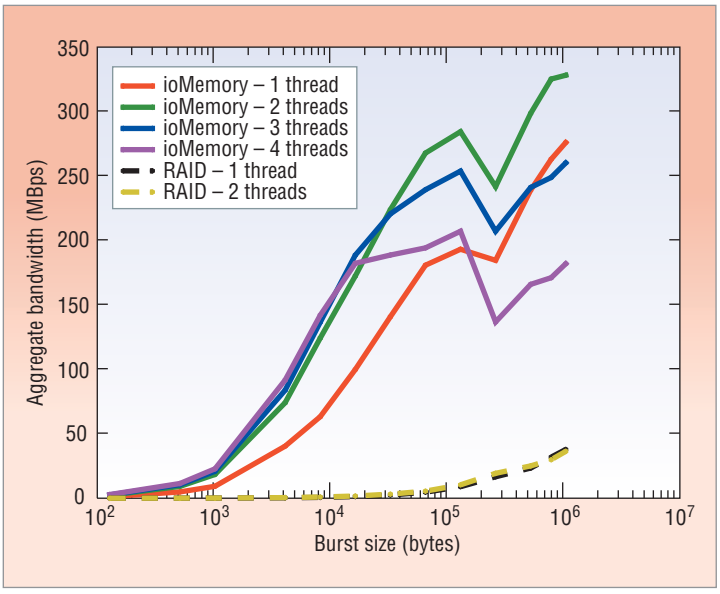


Figure 5. Random burst read performance of the ioMemory benchmark using multiple threads. Moving from one thread to two boosted performance by 19 to 99 percent; due to limitations of the beta version's DMA controller, performance decreased when more than two threads were employed.

Manufacturers can likewise use horizontal parallelism to scale bandwidth by using multiple flash memory chip stacks on a PCB and striping data across the stacks. The fact that flash memory chips employ a low pin count—15 pins for the user interface—makes it possible to utilize a large number of stacks when flash-controller logic is implemented with an FPGA or application-specific integrated circuit.

ioMemory benchmark

Given the bandwidth and latency potentials of flash memory, multiple vendors are building hard-drive replacement products out of NAND flash parts. In 2007, Fusion-io, which is developing ioMemory, a PCI-E-based product, provided us with access to a beta prototype featuring 32 Gbytes of NAND flash storage built on 16 single-chip stacks of 2-Gbyte flash chips. A midsized Xilinx FPGA holds the low-level flash controller hardware and the PCI-E interface to the host. Fusion-io also provides host device drivers to make the ioMemory appear to the Linux kernel as a standard block device.

We performed multiple benchmarks to observe the ioMemory's low-level performance details. When sequentially streaming through large files, it yielded 446 MBps in read tests and 378 MBps in write tests. These speeds were roughly four times faster than a pair of SATA hard drives arranged in a software RAID 0. However, a random read test demonstrated the true potential of flash storage. In this test, multiple threads issued block read requests to random locations within a 16-Gbyte file. While the random access patterns limited the SATA hard-drive RAID to 30 MBps, the ioMemory achieved 328 MBps.

Figure 5 presents the performance results for different burst sizes and numbers of threads. As these numbers indicate, moving from one to two threads with the ioMemory gives a sizable performance gain. This gain can be attributed to the fact that the ioMemory can process a small number of transactions concurrently, thereby overlapping access times. Performance decreased when more than two threads were employed. Fusion-io attributed this drop to limitations in the beta hardware's DMA controller that will be fixed in the final product. Performance also dropped when bursts exceeded 128 Kbytes. This drop can be attributed to fragmentation, as the card internally stores up to 128 Kbytes of data contiguously. More recent versions of ioMemory promise even better performance, as the hardware has been scaled from 16 flash stacks to 20. This geometry scales theoretical read performance from 640 to 800 MBps.

I/O-intensive sparse graph analysis

To investigate ioMemory performance on I/O-dominated applications, we developed a graph analysis benchmark motivated by real-world applications of semantic graph analysis, which is used to discover relationships in large data sets. Graphs that represent interaction in semantic networks can become extremely large, requiring hundreds of millions of nodes. In practice, the current size limit for graph analysis is 10^8 nodes, while the projected need is 10^{12}.

LLNL first demonstrated breadth-first search of a 3×10^9 node graph on the IBM BlueGene/L, the world's fastest supercomputer.[7] A random graph of this size is the largest that can fit in the machine's 32,768-node memory. Subsequently, LLNL processed a 10^{10}-node scale-free graph using a very different approach and architecture. The breadth-first search was written as SQL queries into a relational database that stored edges in a table, and it used a 648-node Netezza server.

Our benchmark graph algorithm performs out-of-core level-set expansion, a variant of breadth-first search. It runs on a standard Linux machine and uses the ext3 file system. While graph traversal can result in random reads to the file system, the out-of-core algorithm uses an optimized file-based graph layout that attempts to place adjacent vertices in the same disk block.

The algorithm's ingest phase builds the graph. It reads edges from external storage and places them in an in-memory edge buffer, from which it forms adjacency lists. It then places these lists in an in-memory adjacency buffer, and when the buffers become full, merges them into two files, a partition index file (PIF) and a partition file

(PF). The PIF holds vertices and pointers to their adjacency lists in the PF. In the search phase, the level-set expansion benchmark reads the PF and PIF to derive level set $n + 1$ from level set n.

Our experiment used two real graphs, the Internet Movie Database (IMDB) graph (www.iruimte.nl/graph/imdb.txt) with 3.5 million vertices and the Computer Science Bibliograph (DBLP; http://dblp.uni-trier.de/xml) with 1.2 million vertices and a synthetic scale-free graph with 1 million vertices and an average degree of 5. We measured runtime using six different block sizes for the graph data file (256, 512, 1,024, 2,048, 4,096, and 8,192 bytes) and two different locations for the input data sets and temporary files—a local SATA disk and ioMemory. The local disk was a Seagate Barracuda 7,200 rpm (ST380815AS), with 3 Gbps SATA volume, capable of streaming 60 MBps to Linux applications.

The ingest phase, which reads in the raw graph and writes out the optimized graph layout, doesn't benefit from the NAND flash drive. However, the read-dominated search phase showed up to a factor of two improvement in runtime when the data set and graph files were accessed from the ioMemory. Figure 6 compares the runtime of the graph benchmark's search portion for the three graphs. Runtime is lower for all three graphs, with the DBLP graph showing the greatest benefit—an average speedup of a factor of two. Although the ioMemory bandwidth tests showed an order of magnitude improvement between disk and ioMemory random reads, the measured speedup reflects the fact that the graph algorithm can exploit vertex locality in memory and therefore doesn't need to access the drive continuously.

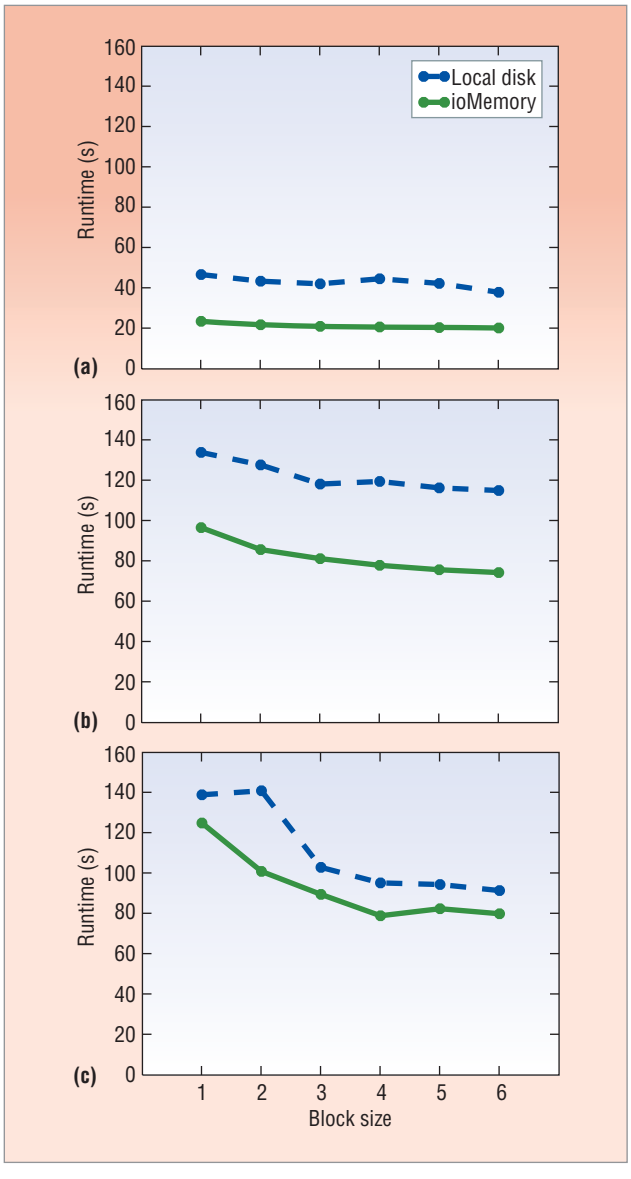


Figure 6. Graph benchmark performance: local disk and ioMemory. (a) DBLP graph, (b) IMDB graph, (c) synthetic graph.

Data-intensive problems challenge conventional computing architectures with demanding CPU, memory, and I/O requirements. Our experiments to date suggest that emerging hardware technologies to augment traditional microprocessor-based computing systems can deliver 2 to 17 times the performance of general-purpose computers on a wide range of data-intensive applications by increasing compute cycles and bandwidth and reducing latency.

GPU and FPGA coprocessors can deliver one to two orders of magnitude increase in compute cycles through massive parallelism and application-specific caches, while high-performance I/O systems based on solid-state nonvolatile memory offer one to two orders of magnitude improvement in latency over enterprise-class hard-disk drives.

Our experiments demonstrate the advantages of using a coprocessor and NAND flash separately. In addition, the language classification benchmark further shows that combining the two technologies offers a substantial benefit—a 1.75 speed increase by using the ioMemory rather than local disk to stream data to the coprocessor. Speedup was limited by having to stage the data in the CPU's memory before forwarding it to the coprocessor. Our future work will focus on methods to bypass the CPU memory and pass data directly from the flash device to the coprocessor, thereby letting the coprocessor access the data at closer to the raw NAND array rate.

Acknowledgments

We thank John Grosh, John Johnson, John May, David Hysom, Don Dossa, Scott Kohn, Eric Greenwade, and Lisa Corsetti for their contributions to the Storage Intensive Supercomputing project at Lawrence Livermore

National Laboratory. This work was performed under the auspices of the US Department of Energy by LLNL under contract DE-AC52-07NA27344.

Dedication

We dedicate this article to the memory of W. Marcus Miller (19 Sept. 1957-19 Feb. 2008), our coauthor and respected colleague at LLNL, without whose efforts this work wouldn't have been possible.

References

1. R. Kolb, "The Large Synoptic Survey Telescope (LSST)," white paper, LSST Corp., 2005; www.lsst.org/Science/docs/LSST_DETF_Whitepaper.pdf.
2. E. Bertin, "SWarp v2.17.0, User's Guide," Institut d'Astrophysique & Observatoire de Paris, 7 Jan. 2008; http://terapix.iap.fr/IMG/pdf/swarp.pdf.
3. W.B. Cavnar and J.M. Trenkle, "N-Gram-Based Text Categorization," *Proc. 3rd Ann. Symp. Document Analysis and Information Retrieval*, Univ. of Nevada, 1994, pp. 161-175.
4. B.H. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," *Comm. ACM*, vol. 13, no. 7, 1970, pp. 422-426.
5. C.M. Kastner et al., "HAIL: A Hardware-Accelerated Algorithm for Language Identification," *Proc. 2005 Int'l Conf. Field Programmable Logic and Applications*, IEEE Press, 2005, pp. 499-504.
6. R. Steinberger et al., "The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages," *Proc. 5th Int'l Conf. Language Resources and Evaluation*, ELRA, 2006; http://langtech.jrc.it/Documents/0605_LREC_JRC-Acquis_Steinberger-et-al.pdf.
7. A. Yoo et al., "A Scalable Distributed Parallel Breadth-First Search Algorithm on BlueGene/L," *Proc. 2005 ACM/IEEE Conf. Supercomputing*, IEEE CS Press, 2005, pp. 25-35.

Maya Gokhale *is a computer scientist at Lawrence Livermore National Laboratory (LLNL). Her research interests include storage-intensive computing systems, reconfigurable computing, and parallel architectures. Gokhale received a PhD in computer science from the University of Pennsylvania. She is a Fellow of the IEEE. Contact her at maya@llnl.gov.*

Jonathan Cohen *is a computer scientist at LLNL. His research interests include computer graphics and visualization, geometric algorithms, and parallel graphics architectures. Cohen received a PhD in computer science from the University of North Carolina at Chapel Hill. He is a member of the ACM. Contact him at jcohen@llnl.gov.*

Andy Yoo *is a computer scientist at LLNL. His research interests include graph mining, scalable graph algorithms, and data management systems for large-scale graphs. Yoo received a PhD in computer science and engineering from the Pennsylvania State University. He is a member of the IEEE Computer Society and the ACM. Contact him at ayoo@llnl.gov.*

W. Marcus *Miller (deceased) was a computer scientist at LLNL. He received a PhD in computer science from Colorado State University.*

Arpith Jacob *is a PhD student in the Computer Science and Engineering Department at Washington University in St. Louis. His research interests include the design of systolic arrays and heuristic architectures for sequence analysis algorithms in computational biology. Jacob received an MS in computer science from Washington University. He is a student member of the IEEE. Contact him at jarpith@cse.wustl.edu.*

Craig Ulmer *is a senior member of the technical staff at Sandia National Laboratories. His research interests include reconfigurable computing, novel storage technologies, and network interface processors. Ulmer received a PhD in electrical and computer engineering from the Georgia Institute of Technology. Contact him at cdulmer@sandia.gov.*

Roger Pearce *is a graduate student in the Computer Science Department at Texas A&M University. His research interests include graph algorithms and robotics applications. Pearce is a student member of the IEEE Computer Society. Contact him at rpearce@tamu.edu.*

COVER FEATURE

ProDA: An End-to-End Wavelet-Based OLAP System for Massive Datasets

Cyrus Shahabi, Mehrdad Jahangiri, and Farnoush Banaei-Kashani
University of Southern California

ProDA employs wavelets to support exact, approximate, and progressive OLAP queries on large multidimensional datasets, while keeping update costs relatively low. ProDA not only supports online execution of ad hoc analytical queries on massive datasets, but also extends the set of supported analytical queries to include the entire family of polynomial aggregate queries as well as the new class of plot queries.

Recent advancements in sensing and data acquisition technologies have enabled collection of massive datasets that represent complex real-world events and entities in fine detail. In light of access to such datasets, scientists and system analysts are no longer restricted to modeling and simulation when analyzing real-world events. Instead, the preferred viable approach derives observations and verifies hypotheses by analytical exploration of representative real datasets that capture the corresponding event. This approach demands intelligent data storage, access, and analytical querying solutions and tools that facilitate convenient, efficient, and effective exploration of these massive datasets. This poses both an opportunity and a Grand Challenge for the database community.[1]

By design, developers optimize traditional databases for transactional rather than analytical query processing. These databases support only a few basic analytical queries with nonoptimal performance and, therefore, provide inappropriate tools for analyzing massive datasets.

Instead, current practice exploits the extensive analytical query processing capabilities of spreadsheet applications such as Microsoft Excel and Lotus 1-2-3 to explore the data. However, in this case the limitation lies in the spreadsheet applications' capability to handle large datasets. With this approach, while the original datasets still reside in a database server, smaller subsets of the data will be selected—by sampling, aggregation, or categorization—and retrieved as new data products for further local processing with the spreadsheet application at the client side.

This inconvenient and time-consuming process of generating a secondhand dataset might unavoidably result in loss of relevant or detailed information. This can bias the analysis and encourage analysts to justify their own assumptions rather than discover surprising latent facts from the data. Further, given that most of the analysis occurs locally at the client with this approach, the data transfer overhead is increased, resource sharing at the server side does not apply, and considerable processing is required at the client side.

Online analytical processing tools have emerged to address the limitations of traditional databases and spreadsheet applications. Unlike traditional databases, OLAP tools support a range of complex analytical queries; unlike spreadsheet applications, they can also handle massive datasets. Moreover, OLAP tools can process user queries online. Online query processing is arguably

0018-9162/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

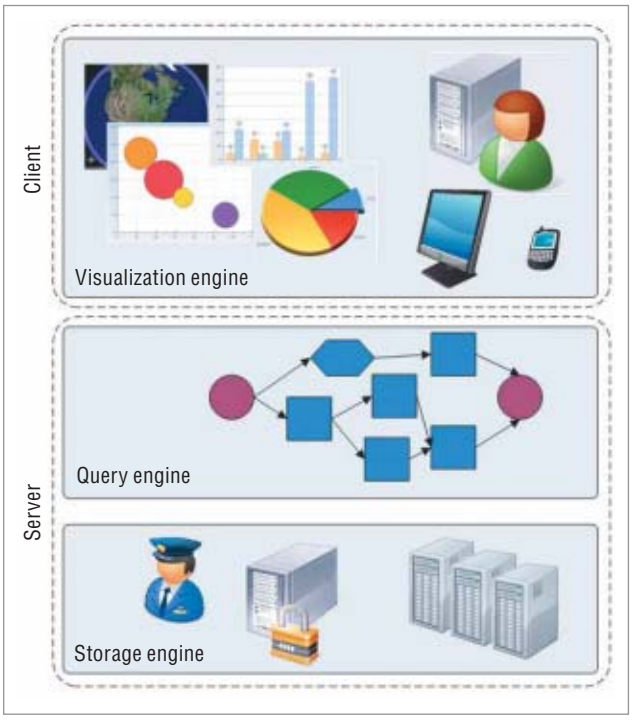


Figure 1. ProDA architecture. The system supports a wide range of analytical queries while being able to handle massive datasets. The storage tier maintains the data while the query tier executes the queries. Together, these elements comprise the ProDA server. The ProDA client, on the other hand, implements the visualization tier on top, where user queries are formulated and results are presented.

a requirement for effectively supporting exploratory data querying. However, current OLAP tools heavily rely on precalculating the query results to enable online query processing. Consequently, they can support only a limited set of predefined (rather than ad hoc) queries online.

Over the past half decade, we have designed, developed, and matured an end-to-end system, dubbed ProDA (for progressive data analysis system), that efficiently and effectively analyzes massive datasets. ProDA functions as a client-server system with the three-tier architecture shown in Figure 1: The storage tier maintains the data at the bottom while the query tier executes the queries at the midlevel; together these elements comprise the ProDA server. The ProDA client, on the other hand, implements the visualization tier on top, where user queries are formulated and query results are presented.

As an OLAP tool, ProDA supports a wide range of analytical queries while also being able to handle massive datasets. However, compared to current OLAP tools, ProDA offers the extended and enhanced online query processing capabilities made possible by leveraging our in-house wavelet-based technology.[2,3]

Specifically, ProDA supports more complex analytical queries, including the entire family of polynomial aggregate queries as well as the class of plot queries previously unsupported by OLAP tools. Moreover, unlike current OLAP tools, ProDA supports online ad hoc queries. To enable online execution of these queries, we take two measures to improve the efficiency of query execution. First, we treat analytical queries as database queries and push them down, close to the data, to be executed at the server side rather than in client-side applications. Second, we leverage the wavelet transform's excellent energy-compaction properties, which allow for accurate approximation of the query result with minimal data access. Here, we innovate by transforming the query as well as the data.

Since queries are often more patterned than data, they are also more compactable when transformed into the wavelet domain. With a highly compact yet accurate query representation, in addition to a compact data representation, we can effectively select and retrieve the high-energy data coefficients relevant to the query with exponentially less data access when compared to previous approaches that only transform data. Therefore, we can approximate the query result accurately with an exponentially improved response time. In combination, these two measures let ProDA carry out the online execution of ad hoc queries.

Further, leveraging the multiresolution properties of the wavelet transform can answer approximate queries progressively, either with fixed accuracy or, alternatively, with fixed performance such as a limited time frame. This feature is particularly useful for exploratory data analysis, which can be quite time- and resource-intensive with massive datasets.

With exploratory analysis, users often issue several back-to-back queries, each time revising and enhancing a query based on quick observation of the previous query's partial results. With progressive queries, users can save time and system resources according to the required query accuracy or available time and system resources to execute each query.

On the other hand, with ProDA—inline with the typical use of wavelet transform in databases—we can optionally drop the transformed data's low-energy coefficients to save storage space. Since the storage space is no longer the main resource constraint, we prefer lossless data storage to allow for exact query answering, if needed.

Finally, ProDA also introduces novel operators that let developers manipulate the stored data by inserting, deleting, and updating data records directly in the wavelet domain rather than in the original domain. These operators are extremely useful for maintaining the massive datasets, particularly when the data is frequently updated by, for example, incoming data streams; otherwise, the entire dataset must be transformed back to the original domain for any minor data manipulation.

CASE STUDY

We have successfully used ProDA to analyze real-world data applications. These case studies, such as the following oilfield sensor data analysis study, served as proofs of concept and as testbeds for realizing and addressing ProDA's practical limitations.

With recent advancements in sensor technology, we can now economically equip oil production wells and an oilfield's water and steam injection wells with multimodal sensor devices to monitor gas, water, steam, oil pressure, and related factors. Smart oilfield management systems (http://cisoft.usc.edu, for example) must analyze such a data feed in real time to provide decision support for the oilfield operators. The system can also extend to automatically control the oilfield when it creates a closed loop.

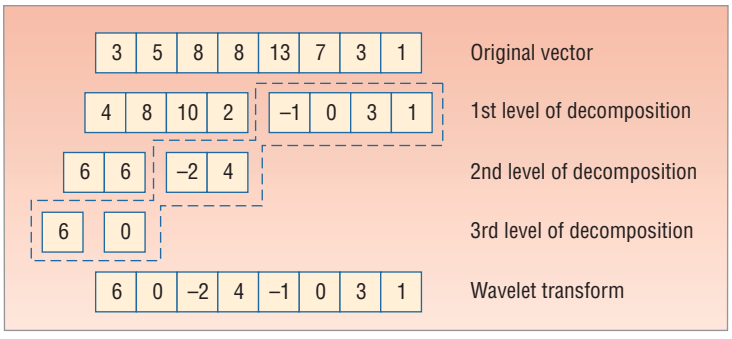


Figure 2. Wavelet transform. To perform this transform, we first compute the pairwise average and pairwise difference for all consecutive pairs of data values in the original vector. The result consists of two vectors, each of half size, including a smooth average view and a rough difference view. We repeat this process.

Analyzing oilfield sensor data is complicated not only by the data's size and many dimensions, but also because of the data's high update rate, which renders any slow analysis process useless. With this application, domain experts must execute complex ad hoc queries on the fly to understand the oilfield's dynamic behavior in real time and react accordingly.

To illustrate, imagine that the oilfield management system is continually receiving gas, water, steam, and oil pressure readings from a field of 4,000 wells, where each well has 20 sensor devices with a sampling period of 15 seconds—deployed at the well's various depths. Typically, a reservoir engineer must continuously monitor the covariance between the water and steam injection and oil production across all wells. The covariance matrix determines the injection rates required for optimal total production. As far as we know, only ProDA can compute such a complex query on the fly.

UNDERLYING TECHNOLOGY

The technology ProDA uses to manage massive data relies heavily on tools such as wavelets.

Preliminaries: Wavelet transform

Developers originally adopted the wavelet transform tool from the signal processing literature for multiscale decomposition of data signals. Wavelets can transform a data signal into a pair of rough and smooth views. The rough view captures the data's low-energy components, whereas the smooth view represents the high-energy components. The tool then iteratively applies the transformation to each view to generate more pairs of rough and smooth views at lower resolutions. Eventually, the transformed data appears as a combination of selected views at different resolutions.

Wavelet transform is reversible. Thus, the system can use the transformed data to reconstruct the original data at any resolution, including the original one. With wavelet transform, the system generates the smooth and rough views by applying low-pass and high-pass filters to the data, respectively. In the case of discrete wavelet transform (DWT), a filter simply consists of a coefficient pair. For example, in the case of the basic Haar wavelet transform, the low-pass filter is

$$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$$

and the high-pass filter is

$$\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right).$$

To apply, the filter is convoluted with the signal to generate the transformed view. The following example simplifies the demonstration by using (½, ½) and (½, –½) as filters.

Data reconstruction

Consider the vector of 8 values (3, 5, 8, 8, 13, 7, 3, 1) depicted in Figure 2. To perform wavelet transform, we first compute the pairwise average and pairwise difference for all consecutive pairs of data values in the original vector. The result consists of two half-size vectors, including a smooth average view (4, 8, 10, 2) and a rough difference view (–1, 0, 3, 1) that, together, form the data's first-level decomposition.

Next, we continue by similarly constructing the average and difference views for the average view from the first-level decomposition. The next-lower resolution average and difference views are (6, 6) and (–2, 4), respectively. Finally, by repeating the process, we find the average view, (6), and the difference view, (0), for the average view from the second-level decomposition. These views form the decomposition's third and final level. By combining the final average and the differences produced at all levels of decomposition,

we form the original vector's wavelet transform: (6, 0, –2, 4, –1, 0, 3, 1).

Wavelet-based online query processing

In the database literature, developers often use wavelet transform for data compression and not for efficient query execution as we do with ProDA. This approach transforms the data and, to save storage space, it preserves only the transformed data's high-energy coefficients. This is by nature a lossy transformation and, as an unwanted consequence, querying such data inevitably results in approximate results. Moreover, the data compression's efficacy is highly data-dependent, and compression is only effective when the data has a concise wavelet approximation.

Instead, we use wavelets to approximate the incoming queries rather than the data. If certain conditions are met, the queries are well compressible, and query execution in the wavelet domain thus functions efficiently. With ProDA, although the dataset is still wavelet transformed, it is not necessarily approximated because we can optionally maintain all coefficients of the transformed data. As a result, ProDA can execute queries accurately and efficiently, independent of the data characteristics.

Data compression is essential for real-world situations that deal with either extremely large data or limited storage capability.

On the other hand, the query's desired accuracy varies per application, user, or dataset, and therefore offers an opportunity to trade off accuracy for improved response time. ProDA lets progressive query execution exploit this opportunity. Intuitively, wavelet transform preserves the data's energy but redistributes it across all wavelet coefficients such that most of the energy is localized in a few coefficients. These high-energy wavelet coefficients are more significant than others in representing the data. By ordering and progressively retrieving the coefficients according to their significance, ProDA can quickly produce an accurate estimation of the query result, which progressively converges to the exact result.

BOTTOM TIER: STORAGE ENGINE

The bottom tier of ProDA—the storage engine—stores and manages the data. This engine consists of several different data sources: user accounts, a history of user activities, a datacube directory, and datacubes. Each datacube consists of

- general information about the cube,
- a list of user-defined polynomial queries,
- the dimension values of datacubes, and
- the wavelet-transformed measure values.

With the exception of the Wavelet DB, we have implemented all the data sources in a relational database server to facilitate the management of the data. For the Wavelet DB, we have developed our own custom-made binary file structures to store multidimensional wavelet-transformed data using our optimal disk block allocation strategy in order to achieve high efficiency.

Disk block allocation

In practice, a disk block can store more than one data coefficient. This presents the challenge of optimizing the placement of the wavelet coefficients in the secondary storage (for example, magnetic disk drives), such that the number of disk blocks retrieved to answer queries is minimized. In other words, we must lay the data out on the disk in such a way that the *principle of locality of reference* holds, that is, when a datum is accessed on a disk block, it is likely for other data on the same block to be accessed simultaneously.

We have observed that OLAP queries on wavelet data require a distinct access pattern, such that when the system retrieves a wavelet coefficient, it guarantees that all its dependent coefficients will also be retrieved. Hence, we have exploited this unique access pattern and designed a disk-placement strategy for wavelet data that yields the best possible I/O complexity for query evaluation.[4]

Query-aware data compression

Many real-world situations deal with either extremely large data or limited storage capability. For either of these, data compression is essential. With ProDA to compress the data (assuming that the storage space is limited), we can store only the most significant wavelet coefficients. The most significant coefficients are those with either the lowest frequencies or the highest energy. We define the energy of a coefficient as the power two of the coefficient value. We take this approach because high-frequency coefficients or small coefficients can be considered as noise and, thus, could be dropped.

Next, if a query requests a data coefficient that is not stored, such as one dropped previously due to data compression, we assume the coefficient equals zero and continue the process. This assumption basically implements *hard thresholding*. Using such a compression schema, we provide excellent approximate query results with ProDA while maintaining the query response time independent of the compression ratio.

MIDDLE TIER: QUERY ENGINE

The query engine of ProDA provides a rich set of Web services that consists of four groups: browsing services,

essential querying services, advanced querying services, and data mining services. Figure 3 illustrates these categories in four layers, as each is built atop another.

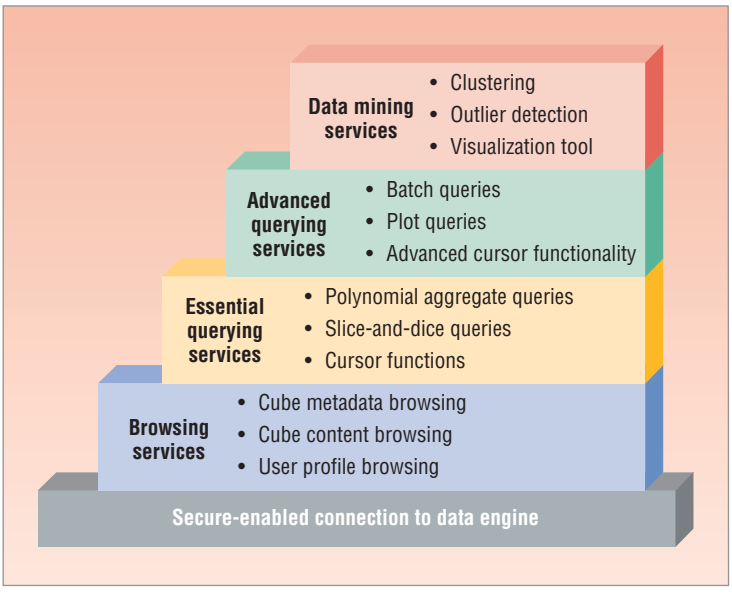


Figure 3. ProDA's query engine. The engine provides a rich set of Web services consisting of four groups of services.

Browsing services

We have designed this group of Web services to allow users to manage, explore, and modify the available datacubes.

Cube metadata browsing. These services let users explore the metadata of the available datacubes, such as the description, schema, and the wavelet filter used for datacube transformation. Users can add, modify, or drop the user-defined queries, and can browse through and reuse previously issued queries.

Cube content browsing. This family of services lets users directly access the content of the selected datacube. Using this set, we can add or drop a datacube, or modify a datacube's wavelet coefficients by issuing update queries.

User profile browsing. This family of services is used to implement the user access control. Besides, the services provide facilities for the users to add, drop, and modify the user profile information.

Essential querying services

This class of services includes polynomial aggregate queries, slice-and-dice queries, and cursor functions.

Polynomial aggregate queries. The standard statistical range-aggregate queries are implemented as predefined queries in ProDA—for example, count, sum, average, variance, covariance, and correlation. Furthermore, we have broadened the supported queries to encompass ad hoc statistical functions by providing a formal method of expressing analytical queries as polynomial functions on the data attributes. Users can define any polynomial expression on attributes and share the definition with others. For example, a high-ordered polynomial function such as *kurtosis* (the fourth moment of data divided by the square of the variance) is not predefined in ProDA; however, our new utility environment lets users define and share such complex queries on-the-fly.

We implement an arbitrary polynomial query by using two basic Web services, *PushTerm* and *PushOperator*. Given the importance of the order of the function calls, ProDA parses the equations in post order. For instance, consider implementation of the variance function using PushTerm and PushOperator. Variance is defined as follows:

$$Var(x) = \frac{\sum x_i^2}{n} - \left(\frac{\sum x_i}{n}\right)^2$$

The post order representation of this function is $\sum x_i^2, n, /, \sum x_i, n, /, ^2$, and $-$. Accordingly, we implement our polynomial query by executing the following [9] calls:

```
PushTerm(x, 2); PushTerm(x, 0);
  PushOperator('/'); PushTerm(x, 1);
  PushTerm(x, 0);
PushOperator('/'); PushOperator('**');
  PushOperator('—'); Submit();
```

Slice-and-dice queries. Consider the scenario in which we wish to extract a region of the original data from its wavelet transform. This poses the following dilemma: We can either reconstruct the entire dataset and extract the desired region (which is infeasible), or reconstruct the desired region point-by-point (which is inefficient, particularly for large regions). Instead, we translate the selected operation of the relational algebra to the wavelet domain and choose the required coefficients for reconstruction of the desired range.[5] By employing this technique, ProDA clients can access small subsets of the wavelet data instantly because the server never needs to reconstruct the entire dataset. This class of queries is used when ProDA users intend to download a small subset of data into their own machines for convenient interaction. For example, an oil production engineer usually analyzes a few production oil wells at a time, without accessing the data of the entire reservoir. This relevant data is usually small enough to be cached in the client machine. The cached data is readily updatable and enables efficient query processing. Meanwhile, ProDA allows the user to receive the exact result when the remote connection is available.

```
// Creating an instance and storing session state
ProDAWebServices pws=new ProDAWebServices();
// Selecting a cube with the login information
pws.SelectDB(dbName,userName,password);
// Defining a range and submitting a query
pws.SetRange(lowerLeft,upperRight);pws.Variance(1);
// Asking for result progressively
while(pws.HasMore())
    Console.Write("Result="+pws.Advance(5%));
```

Figure 4. Sample ProDA client for progressive querying in C#. The client uses cursor services to obtain the query result progressively.

Cursor functions. ProDA provides progressive query answering by ordering the wavelet coefficients based on the significance of the query coefficients. Hence, ProDA incrementally retrieves the data coefficients related to each query from the storage engine. Cursor functions track the progress of the query operations and the data retrieval operations. They also let users stop the query processing any time they are satisfied with the intermediate approximate result.

Advanced querying services

Utilizing the essential querying services, we efficiently implement two widely used advanced queries—batch and plot queries. Furthermore, we provide additional cursor functionality for these two query classes by prioritizing certain query regions.

Batch queries. Scientists typically submit queries in batch rather than issuing individual, unrelated queries. We have proposed a wavelet-based technique that exploits I/O sharing across a query batch for efficient and progressive evaluation of the batch queries. The challenge is that controlling the structure of errors across the query results now becomes more critical than minimizing errors per each individual query. We have defined a class of structural error-penalty functions in our framework to achieve the optimal progressiveness for a batch queries.[6]

Users can invoke the batch query services by specifying a grid over data dimensions. Thereafter, they can progressively receive the results for the entire batch.

Plot queries. Plots are among the most important and widely used tools in scientific data analysis and visualization applications. In general, each plot point is an aggregate value over one or more measure attributes for a given dimension value. The current practice for generating a plot over a multidimensional dataset involves computing the plot point-by-point, where each point is the result of computing an aggregate query. Therefore, for large plots a large number of aggregate queries are submitted to the database. This method is neither memory-efficient (on either the client or the server side) nor communication-efficient.

On the other hand, we redefine a plot as a single database query and employ a wavelet-based technique that exploits I/O sharing across the aggregate queries for all plot points to evaluate the plot efficiently. With this approach, the main idea is to decompose a plot query into two sets: a set of aggregate queries and a set of slice-and-dice queries. Subsequently, we can use our earlier results to compute both sets of queries efficiently in the wavelet domain.

Users can invoke the plot query services by specifying a range over the data dimensions and selecting the independent variable for the plot. A developer can employ the following advanced cursor functionality to generate the plot output progressively.

Advanced cursor functionality. Scientists consider batch and plot queries among the most favored statistical analysis tools. They are widely used to provide valuable insights about any dataset. For example, we can extract outliers, trends, clusters, or local maxima by quickly looking at the output of such queries. Furthermore, the entire query result often is not used at once. For instance, the result might not fit on the screen, the user might point to a specific region of the result, or the user might prioritize the subsets of the result—for example, local maxima, or the regions with high values or high gradients, to be computed. Accordingly, the advanced cursor functionality allows users to modify the structural-error-penalty functions to control the progressiveness of the query.

Data mining services

We are currently designing and implementing additional analytical query processing components in ProDA to support complex data mining functionalities. So far, we have enhanced ProDA by incorporating clustering and outlier detection techniques, and an effective visualization tool for exploratory data mining. For clustering, we use various methods—such as K-means with different distant functions—to cluster batch or plot query results. Using a similar approach with a customized penalty function, ProDA enables the progressive answering of the outlier detection queries. The visualization tool generates time-stamped KML files to be imported to Google Earth for effective spatial and temporal visualization.

TOP TIER: VISUALIZATION

By pushing the extensive and complex data processing tasks to the server side, the ProDA client can be implemented as a light and yet effective interface.

Figure 4 demonstrates a sample client that invokes ProDA Web services for query processing. First, we create an instance of ProDA services, then select a datacube with appropriate login information. Next, we specify a range and submit a variance query. Finally, we use our cursor services to obtain the query result progressively.

In addition, we have developed a stand-alone graphical C# client, ProDA client, for efficient interaction with arbitrary scientific datasets. We emphasize using the graphical interface as a more intuitive query interface. We have incorporated smart client functionalities (for example, smart data management, online/offline capability, high-fidelity UI) into ProDA client; thus, ProDA provides an adaptive, responsive, and rich interactive experience by leveraging local resources and intelligently connecting to distributed data sources.

The ProDA client consists of data and query visualization, high-fidelity UI, connectivity management, and advanced visualization. In short, ProDA lets the user select a datacube and visualize the data. It also accepts queries from a user and displays the results progressively in offline or online mode, provides resource sharing at the client machine, and utilizes advanced commercial visualization tools.

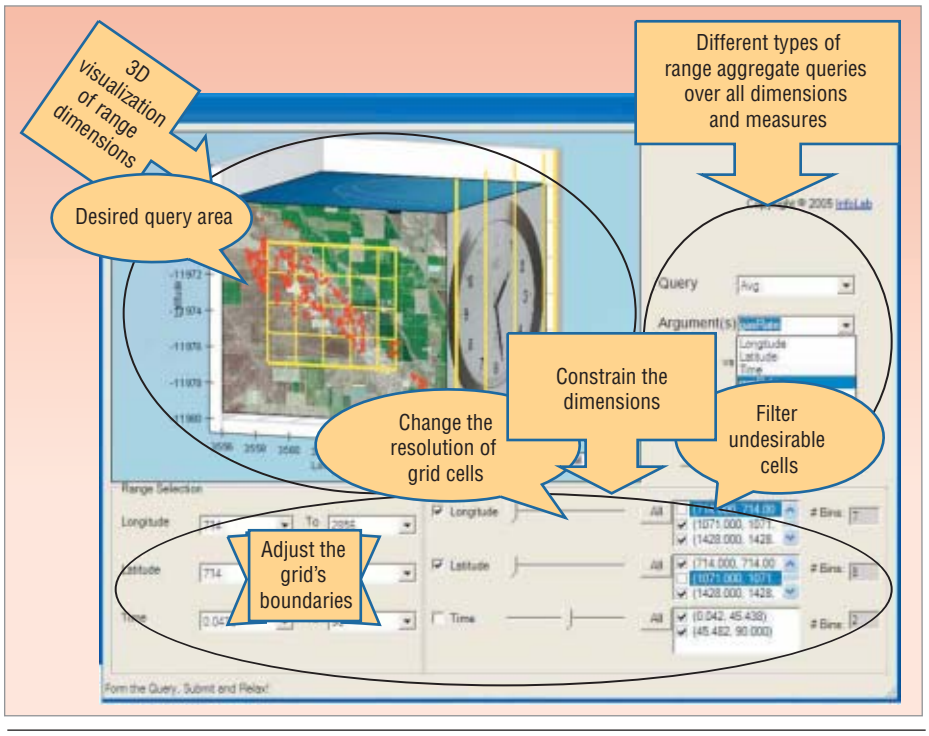


Figure 5. Data visualization. This sample data visualization shows a particular dataset with a user-specified grid for batch query processing. Data visualization modules exhibit various attribute types, including spatial, temporal, numeric, and categoric.

Data visualization

This module lets the user log in to the ProDA storage engine, browse the data sources, and select one to interact with. It then provides a visualization of the selected dataset for the user to browse, scroll, and rotate. It also facilitates definition of the desired ranges and queries. Defining a bounding box for a single range query and a grid for batch queries over all dimensions is one of many necessary functionalities the data visualization module must provide. Figure 5 demonstrates a sample oilfield sensor data visualization. The grid is specified by the user for batch query processing.

Data visualization modules exhibit various attribute types, including spatial, temporal, numeric, and categoric. In addition to presenting the hierarchy of the dimension values, ProDA displays dependent dimensions and allows the user to work with all dimensions simultaneously. For example, while exploring the oilfield sensor data, we can select a set of oil wells by identifying a certain window of interest or, alternatively, by choosing the wells based on their corresponding labels.

Once a developer selects a datacube, ProDA enables the list of available queries. This list includes common analytical queries, user-defined polynomial range-aggregate queries, plot queries, and slice-and-dice queries. In addition, the user can define a new polynomial query, add it to the list, and even share it with other users of this dataset.

Query visualization

The ProDA client displays the output of queries once they become available, then updates them frequently as new updates appear. It also visualizes the output of batch queries and plot queries using various advanced built-in chart types. As a query progresses, the user can start interacting with the result through the ProDA visualization module's operations. Zooming in and out, pivoting, and exporting are among the many possible actions.

When real-world data contains noise, the output of a plot query displays some undesired small variations. These variations not only do not carry any valuable information, but also confuse the users when analyzing the data. Leveraging from wavelet hard threshholding, ProDA supports advanced denoising of the query output and generates smoother outputs, especially when they carry white Gaussian noise.

High-fidelity UI

ProDA, as a well-designed smart client, guarantees that the utilities already installed on the client machine can access and process the data that ProDA generates. This is essential in practice because typical users are accustomed to using Microsoft Office components,

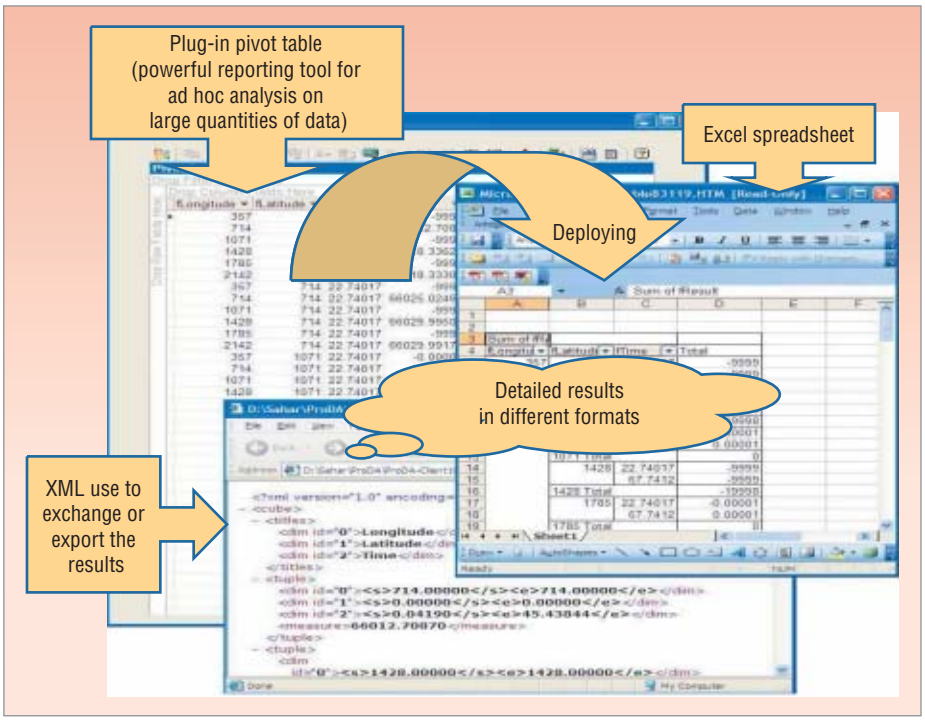


Figure 6. ProDA's export capability. With its extensive set of export functionalities, the system can be connected to almost any application. At any time, users can export its data to XML, Excel, text files, and many more formats.

especially Microsoft Excel, in addition to ProDA's built-in visualization packages.

Toward this end, the ProDA client provides a similar interface to improve the users' performance and decrease training costs. In particular, we use the Microsoft Excel pivot table components designed for ad hoc analysis of large quantities of data. A pivot table is a powerful reporting tool that features basic to complicated calculation modules independent of the spreadsheet's original data layout. With the table's drag-and-drop function, users can pivot the data and perform local computation tasks. Consequently, users will have an interactive table that automatically extracts, organizes, and summarizes the data. They can use this report to analyze the data, make comparisons, detect patterns and relationships, and discover trends.

With its extensive set of export functionalities, ProDA can be connected to almost any application. At any time, users can export the data to XML, Excel, text files, and many more formats. Figure 6 shows ProDA's export functionality.

Connectivity management

ProDA client lets a user cache a subset of a dataset while the system is online. Later, when the system is offline, by utilizing the cached data, the user can access all ProDA's functionalities as if the system were still connected to the server. Offline query processing offers an especially attractive feature for mobile users who need access to the data during a disconnected operation. Hence, we will empower ProDA clients to support offline wavelet-based query processing. With ProDA, the user can query the wavelet-transformed sketch to receive an excellent approximate answer.

Advanced visualization

We employ other widely accepted commercial products for universal spatial data representation. In particular, ProDA exports the spatial query results to Google Earth for advanced visualization. This tool allows a group of users to exchange their query results with each other while viewing other spatial data in relation to the problem.

ProDA enables exploratory analysis of massive multidimensional datasets. Standard OLAP systems that rely on query precalculation are expensive to update, whereas traditional, easily updatable databases often have poor response time with analytical queries. With ProDA, we employed wavelets to support exact, approximate, and progressive OLAP queries on large multidimensional datasets, while keeping update costs relatively low. ProDA extends the set of supported analytical queries to include the entire family of polynomial aggregate queries as well as the new class of plot queries.

Acknowledgments

This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0238560 (PECASE), unrestricted cash gifts from Google and Microsoft, and by NASA's JPL SURP program and the Center for Interactive Smart Oilfield Technologies (CiSoft). CiSoft is a joint University of Southern California/Chevron initiative.

References

1. M. Stonebraker et al., "The Lowell Database Research Self-Assessment," *Comm. ACM*, vol. 48, no. 5, 2005, pp. 111-118.
2. M. Jahangiri and C. Shahabi, *Wolap: Wavelet-Based Range Aggregate Query Processing*, tech. report, Dept. Computer Science, Univ. of Southern California, 2007.

3. R. Schmidt and C. Shahabi, "Propolyne: A Fast Wavelet-Based Technique for Progressive Evaluation of Polynomial Range-Sum Queries," *Proc. Extending Database Technology Conf.* (EDBT 02), Springer, 2002, pp. 664-681.
4. C. Shahabi and R. Schmidt, *Wavelet Disk Placement for Efficient Querying of Large Multidimensional Data Sets*, tech. report, Dept. of Computer Science, Univ. of Southern California, 2004.
5. M. Jahangiri, D. Sacharidis, and C. Shahabi, "SHIFT-SPLIT: I/O Efficient Maintenance of Wavelet-Transformed Multidimensional Data," *Proc. ACM SIGMOD*, ACM Press, 2005, pp. 275-286.
6. R. Schmidt and C. Shahabi, "How to Evaluate Multiple Range-Sum Queries Progressively," *Proc. ACM PODS*, ACM Press, 2002, pp. 133-141.

Cyrus Shahabi *is an associate professor and the director of the Information Laboratory (InfoLAB) at the University of Southern California's Computer Science Department. His research interests include geospatial and multidimensional data analysis, peer-to-peer systems, and streaming architecture. Shahabi received a PhD in computer science from the University of Southern California. Contact him at cshahabi@cs.usc.edu.*

Mehrdad Jahangiri *is a PhD candidate in the Department of Computer Science at the University of Southern California. His research interests include OLAP, data integration, and data mining. He received an MS in computer science from the University of Southern California. Contact him at jahangir@usc.edu.*

Farnoush Banaei-Kashani *is a postdoctoral research associate in the Department of Computer Science at the University of Southern California. His research interests include data-stream systems, networked databases, and group data management. He received a PhD in computer science from the University of Southern California. Contact him at banaeika@usc.edu.*

2008 RATES for IEEE COMPUTER SOCIETY

Membership Dues and Subscriptions

Membership and periodical subscriptions are annualized to and expire on 31 December 2008.
Pay full or half-year rate depending upon the date of receipt by the IEEE Computer Society as noted below.

Membership Options*

All prices are quoted in U.S. dollars.

	FULL YEAR Applications received 17 Aug 07 – 29 Feb 08	HALF YEAR Applications received 1 Mar 08– 15 Aug 08
I do not belong to the IEEE and I want to join only the Computer Society:	☐ $113.00	☐ $57.00
I want to join **both** the Computer Society and the IEEE:		
I reside in the USA	☐ $220.00	☐ $110.00
I reside in Canada	☐ $195.00	☐ $98.00
I reside in Africa/Europe/Middle East	☐ $187.00	☐ $94.00
I reside in Latin America	☐ $180.00	☐ $90.00
I reside in Asia/Pacific	☐ $181.00	☐ $91.00
I already belong to the IEEE, and I want to join the Computer Society:	☐ $50.00	☐ $25.00

Are you now or were you ever a member of the IEEE?
☐ Yes ☐ No If yes, please provide member # if known: _____

Add Periodicals**

	ISSUES PER YEAR	FULL YEAR Applications received 16 Aug 07 – 29 Feb 08 PRINT + ONLINE	HALF YEAR Applications received 1 Mar 08 – 15 Aug 08 PRINT + ONLINE
BEST VALUE!			
IEEE Computer Society Digital Library (online only)	n/a	☐ $121	☐ $61
ARTIFICIAL INTELLIGENCE			
IEEE Intelligent Systems	6	☐ $43	☐ $22
IEEE Transactions on Learning Technologies†	4	☐ $25	☐ $13
IEEE Transactions on Pattern Analysis and Machine Intelligence	12	☐ $52	☐ $26
BIOTECHNOLOGY			
IEEE/ACM Transactions on Computational Biology and Bioinformatics	4	☐ $36	☐ $18
COMPUTATION			
Computing in Science & Engineering	6	☐ $45	☐ $23
COMPUTER HARDWARE			
IEEE Computer Architecture Letters	2	☐ $29	☐ $15
IEEE Micro	6	☐ $41	☐ $21
IEEE Design & Test of Computers	6	☐ $40	☐ $20
IEEE Transactions on Computers	12	☐ $47	☐ $24
GRAPHICS & MULTIMEDIA			
IEEE Computer Graphics and Applications	6	☐ $43	☐ $22
IEEE MultiMedia	4	☐ $38	☐ $19
IEEE Transactions on Haptics	2	☐ $31	☐ $16
IEEE Transactions on Visualization and Computer Graphics	6	☐ $43	☐ $22
HISTORY OF COMPUTING			
IEEE Annals of the History of Computing	4	☐ $34	☐ $17
INTERNET & DATA TECHNOLOGIES			
IEEE Internet Computing	6	☐ $43	☐ $22
IEEE Transactions on Knowledge and Data Engineering	12	☐ $49	☐ $25
IEEE Transactions on Services Computing†	4	☐ $25	☐ $13
IT & SECURITY			
IT Professional	6	☐ $42	☐ $21
IEEE Security & Privacy	6	☐ $24	☐ $12
IEEE Transactions on Dependable and Secure Computing	4	☐ $33	☐ $17
MOBILE COMPUTING			
IEEE Pervasive Computing	4	☐ $43	☐ $21
IEEE Transactions on Mobile Computing	12	☐ $43	☐ $22
NETWORKING			
IEEE Transactions on Parallel and Distributed Systems	12	☐ $47	☐ $24
SOFTWARE			
IEEE Software	6	☐ $49	☐ $25
IEEE Transactions on Software Engineering	6	☐ $38	☐ $19

* Member dues include $17 for a 12-month subscription to *Computer* magazine.
** Periodicals purchased at member prices are for the member's personal use only.
† Online issues only .

Payment Information

Payment required with application

Membership fee
$ _____

Periodicals total
$ _____

Applicable sales tax***
$ _____

TOTAL:
$ _____

Enclosed:
☐ Check/Money Order****

Charge my:
☐ MasterCard
☐ VISA
☐ American Express
☐ Diner's Club

_____
Card Number

_____
Exp Date (month/year)

_____
Signature

USA Only include
5-digit billing zip code

☐☐☐☐☐

* Member dues include $17 for a 12-month subscription to *Computer*.

** Periodicals purchased at member prices are for the member's personal use only.

*** Canadian residents add 14% HST or 6% GST to total. AL, AZ, CO, DC, GA, IN, KY, MD, MO, NM, and WV add sales tax to periodical subscriptions. European Union residents add VAT tax to IEEE Computer Society Digital Library subscription.

**** Payable to the IEEE in U.S. dollars drawn on a U.S. bank account. Please include member name and number (if known) on your check.

Allow up to 8 weeks for application processing. Allow a minimum of 6 to 10 weeks for delivery of print periodicals.

Please complete both sides of this form.

For fastest service, apply online at www.computer.org/join

Personal Information

Enter your name as you want it to appear on correspondence. As a key identifier in our database, circle your last/surname.

☐ Male ☐ Female

Date of birth (Day/Month/Year)

Title | First name | Middle

Last/Surname

Home address

City | State/Province

Postal code | Country

Home telephone

Home facsimile

Preferred e-mail

Send mail to: ☐ Home address ☐ Business address

Educational Information

First professional degree completed | Month/Year degree received

Program major/course of study

College/University | State/Province | Country

Highest technical degree received | Program/Course of study

Month/Year received

College/University | State/Province | Country

Business/Professional Information

Title/Position

Years in current position | Years of practice since graduation

Employer name

Department/Division

Street address

City | State/Province

Postal code | Country

Office phone

Office facsimile

I hereby make application for Computer Society and/or IEEE membership and agree to be governed by IEEE's Constitution, Bylaws, Statements of Policies and Procedures, and Code of Ethics. I authorize release of information related to this application to determine my qualifications for membership.

Signature | Date

IF8C

NOTE: In order for us to process your application, you must complete and return BOTH sides of this form to the office nearest you:

Asia/Pacific Office
IEEE Computer Society
Watanabe Bldg.
1-4-2 Minami-Aoyama
Minato-ku, Tokyo 107-0062 Japan
Phone: +81 3 3408 3118
Fax: +81 3 3408 3553
E-mail: tokyo.ofc@computer.org

Publications Office
IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314 USA
Phone: +1 800 272 6657 (USA and Canada)
Phone: +1 714 821 8380 (worldwide)
Fax: +1 714 821 4641
E-mail: help@computer.org

BPA Information

This information is used by society magazines to verify their annual circulation. Please refer to the audit codes and indicate your selections in the box provided.

A. Primary line of business

1. Computers
2. Computer peripheral equipment
3. Software
4. Office and business machines
5. Test, measurement, and instrumentation equipment
6. Communications systems and equipment
7. Navigation and guidance systems and equipment
8. Consumer electronics/appliances
9. Industrial equipment, controls, and systems
10. ICs and microprocessors
11. Semiconductors, components, sub-assemblies, materials, and supplies
12. Aircraft, missiles, space, and ground support equipment
13. Oceanography and support equipment
14. Medical electronic equipment
15. OEM incorporating electronics in their end product (not elsewhere classified)
16. Independent and university research, test and design laboratories, and consultants (not connected with a manufacturing company)
17. Government agencies and armed forces
18. Companies using and/or incorporating any electronic products in their manufacturing, processing, research, or development activities
19. Telecommunications services, and telephone (including cellular)
20. Broadcast services (TV, cable, radio)
21. Transportation services (airlines, railroads, etc.)
22. Computer and communications and data processing services
23. Power production, generation, transmission, and distribution
24. Other commercial users of electrical, electronic equipment, and services (not elsewhere classified)
25. Distributor (reseller, wholesaler, retailer)
26. University, college/other education institutions, libraries
27. Retired
28. Others (allied to this field) _____

B. Principal job function

1. General and corporate management
2. Engineering management
3. Project engineering management
4. Research and development management
5. Design engineering management — analog
6. Design engineering management — digital
7. Research and development engineering
8. Design/development engineering — analog
9. Design/development engineering — digital
10. Hardware engineering
11. Software design/development
12. Computer science
13. Science/physics/mathematics
14. Engineering (not elsewhere classified)
15. Marketing/sales/purchasing
16. Consulting
17. Education/teaching
18. Retired
19. Other _____

C. Principal responsibility

1. Engineering or scientific management
2. Management other than engineering
3. Engineering design
4. Engineering
5. Software: science/management/engineering
6. Education/teaching
7. Consulting
8. Retired
9. Other _____

D. Title

1. Chairman of the Board/President/CEO
2. Owner/Partner
3. General Manager
4. V.P. Operations
5. V.P. Engineering/Director Engineering
6. Chief Engineer/Chief Scientist
7. Engineering Manager
8. Scientific Manager
9. Member of Technical Staff
10. Design Engineering Manager
11. Design Engineer
12. Hardware Engineer
13. Software Engineer
14. Computer Scientist
15. Dean/Professor/Instructor
16. Consultant
17. Retired
18. Other Professional/Technical _____

ADVERTISER / PRODUCT INDEX

APRIL 2008

Advertisers	Page
Cambridge University Press	86
e-Science Conference 2008	10
IBM Press	86
IEEE Computer Society Awards	Cover 4
IEEE Computer Society Membership	78-80
King Abdullah University of Science and Technology	85
Milwaukee School of Engineering	84
MIT Press	86
Oak Ridge National Laboratory	82
SRM University	83
Syngress	86
University of Massachusetts Dartmouth	84
Wiley	86
Classified Advertising	82-85

Boldface denotes advertisements in this issue.

Computer
IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, California 90720-1314
USA
Phone: +1 714 821 8380
Fax: +1 714 821 4010
http://www.computer.org
advertising@computer.org

Advertising Sales Representatives

Mid Atlantic (product/recruitment)
Dawn Becker
Phone: +1 732 772 0160
Fax: +1 732 772 0164
Email: db.ieeemedia@ieee.org

New England (product)
Jody Estabrook
Phone: +1 978 244 0192
Fax: +1 978 244 0103
Email: je.ieeemedia@ieee.org

New England (recruitment)
John Restchack
Phone: +1 212 419 7578
Fax: +1 212 419 7589
Email: j.restchack@ieee.org

Northwest (product)
Lori Kehoe
Phone: +1 650-458-3051
Fax: +1 650 458 3052
Email: l.kehoe@ieee.org

Southeast (recruitment)
Thomas M. Flynn
Phone: +1 770 645 2944
Fax: +1 770 993 4423
Email: flynntom@mindspring.com

Midwest (product)
Dave Jones
Phone: +1 708 442 5633
Fax: +1 708 442 7620
Email: dj.ieeemedia@ieee.org

Will Hamilton
Phone: +1 269 381 2156
Fax: +1 269 381 2556
Email: wh.ieeemedia@ieee.org

Joe DiNardo
Phone: +1 440 248 2456
Fax: +1 440 248 2594
Email: jd.ieeemedia@ieee.org

Midwest/Southwest (recruitment)
Darcy Giovingo
Phone: +1 847 498 4520
Fax: +1 847 498 5911
Email: dg.ieeemedia@ieee.org

Southwest (product)
Steve Loerch
Phone +1 847 498 4520
Fax: +1 847 498 5911
Email: steve@didierandbroderick.com

Connecticut (product)
Stan Greenfield
Phone: +1 203 938 2418
Fax: +1 203 938 3211
Email: greenco@optonline.net

Southern CA (product)
Marshall Rubin
Phone: +1 818 888 2407
Fax: +1 818 888 4907
Email: mr.ieeemedia@ieee.org

Northwest/Southern CA (recruitment)
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Southeast (product)
Bill Holland
Phone: +1 770 435 6549
Fax: +1 770 435 0243
Email: hollandwfh@yahoo.com

Japan
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Europe (product/recruitment)
Hillary Turnbull
Phone: +44 (0) 1875 825700
Fax: +44 (0) 1875 825701
Email: impress@impressmedia.com

Advertising Personnel

Marion Delaney
IEEE Media, Advertising Director
Phone: +1 415 863 4717
Email: md.ieeemedia@ieee.org

Marian Anderson
Advertising Coordinator
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: manderson@computer.org

Sandy Brown
IEEE Computer Society,
Business Development Manager
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: sb.ieeemedia@ieee.org

CAREER OPPORTUNITIES

SYSTEMS ANALYST. Design, test and modify systems for performance, data integrity and validations. Provide technical assistance in a multi-user network environment. Assist in design of infrastructure and recommend commercially available software. Req: 4 years experience in job offered or related field. 40 hr/wk. Job/Interview Site: Canoga Park, CA 91303. Fax resume to: T&T Solutions Inc. dba Technocrat Solutions @ (818) 676-1272.

JUNIOR COMPUTER PROGRAMMER. Using Java to design and develop software programs. Conduct trial runs of programs and software applications. Produce documentation of program development. Perform revision, repair, and maintain software programs. Ensure systems response to program's instructions. Req: BS in Comp. Sci. or related field. 40 hr/wk. Job/Interview Site: Costa Mesa, CA. Send resume to: Acropoint, Inc. @ 4 Executive Circle Suite 170, Irvine CA 92614.

COMPUTER MANAGER is wanted by Importing/Distributing Light Bulbs Company in Fairfield, NJ. Must have Master's degree in Computer Science. Must speak, read and write Korean. Apply to: SK America, Inc. 80 Little Falls Road, Fairfield, NJ 07004.

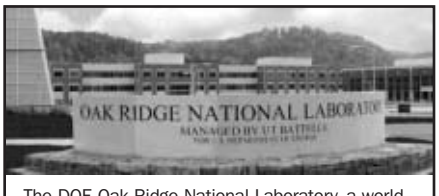
TECHNICAL PROJECT MANAGER, SR. (Woodland Hills, CA) Using AJAX, multithreading, C#, ASP.Net, CSS, SQL 2005, Windows server 2003 setup & integration, web 2.0 dsgn & prgmg. Email res to ScenarioPost.com, thomas@lumincapital.com.

SYSTEM ENGINEER, DATABASE (Mahwah, NJ). Dsgn secure, controlled-access WAN over Internet, w/firewalls & encryption, fully dsgn'd for multinational corps. Working w/ counterparts in Japan, dsgn, install, test, implmt & admin. bilingual logistics control & mgmt info d/base sys. Continually upgrade & integrate OS's & auxiliary s/ware applics into co's global d/base n/work. MA in Comp Sci, 1 yr exp & Japanese fluency req'd. Mail resume to D&M Holdings US, Inc., 100 Corporate Dr., Mahwah, NJ 07430 Attn: Mr. Dom Golio (SE01).

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Cupertino, CA. **Application Engineer**. Resp. for Java applic. devlpmt., JSP, Jboss, Weblogic, Websphere HTML; any J2EE exp. proficiency in a UNIX environment; exp. working with Oracle databases using SQL & PL/SQL; exp. with Oracle databases & database relations; exp. in a complex tech. domain relating to the support, dev. or config. of enterprise S/W; knowledge of Oracle Applic., incl. Application Object Library; troubleshooting skills; minimum of 2-5 yrs. business exp. in a high-tech environment; at least 2-3 yrs. of providing direct functional & tech. support in Project & Portfolio Mgt. Reqs. incl. Master's degree. or foreign equiv. in CS, CE or related field of study & 2 yrs. of exp. or Bachelor's degree. or foreign equiv. in CS, CE or related field of study & 5 yrs. of exp. Send resume & refer to job #CUPSGO. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Cupertino, CA. Technical Liaison. Reqs. comprehensive understanding & familiarity with the Quality Assurance field & end-users & deep understanding of specific technologies, such as SOA, Web Services, Test Automation, & Oracle products. Reqs. incl. Bachelor's degree or foreign equiv. in CS, CE or related field of study & 5 yrs. of related exp. Send resume & refer to job #CUPISH. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

NOKIA SIEMENS NETWORKS US LLC has the following exp/degree position(s) at the following locations: Denver, Colorado ***Senior RF Planning Engineer (Specialist)**: Perform RF design and



The DOE Oak Ridge National Laboratory, a world leader in critical scientific research, is seeking a:

Director of the Computer Science and Mathematics Division

The division conducts advanced computer science research, evaluates future computer technologies and develops new algorithms for the highest performing computers in the world. Successful candidate will be challenged to support the laboratory's goals in extreme-scale computing, as well as developing program funding and attracting highly qualified staff through effective management of all division functions. Strategic planning, top-level program development and execution, and aggressive managerial and technical leadership will be key responsibilities.

A PhD or equivalent education/experience in computational science or computer science, an internationally recognized record of research, and 10 years experience are required. Five years of management experience are also required, along with excellent communication, planning, and organization skills.

For a full job description and to apply, please visit **www.jobs.ornl.gov**

ORNL, a multiprogram research facility managed by UT-Battelle, LLC, for the U.S. Department of Energy, is an equal opportunity employer committed to building and maintaining a diverse work force. EOE.

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

SUBMISSION DETAILS: Rates are $299.00 per column inch ($320 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to: Marian Anderson, Classified Advertising, Computer Magazine, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; (714) 821-8380; fax (714) 821-4010. Email: manderson@computer.org.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, Computer may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...," "...1-4 years maximum experience...," "...up to 5 years experience," or "...10 years maximum experience." Computer reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. Computer assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. Computer encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

optimization with planning and optimization tools; provide technical customer support; have excellent communication and customer service skills; and must be willing to travel. ID# NSN-CO-SPES. Naperville, Illinois ***Project Manager:** Manage base station system implementation projects with competence transfer; manage spreadsheet software to ensure implementation of services; and ensure that project targets are met. ID# NSN-IL-PM. Mail resume to: Attn: 4E-3-350, NSN Recruiter, Nokia Siemens Networks, 6000 Connection Dr., Irving, TX 75039. Must reference ID #. Equal Opportunity Employer.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Cupertino, CA. **Software Designer.** Reqs. exp. in multiple prog. platforms; OS; communication protocols; design & dev. of large scale distrib. applic.; message-queuing middleware; RPC-based network prog.; database tech. & transaction processing; OO design techniques & heuristics; applic. archit. & design patterns; algorithms & data structures; S/W methodologies & notations; JFC/Swing; config. mgt. tools; & deep knowledge of: Java class libraries, Java threading model, & J2EE interfaces & market leading products Reqs. incl. Bachelor's degree or foreign equiv. in CS, Eng., or related field of study & 5 yrs. of related exp. Send resume & refer to job #CUPCDR. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

NOKIA INC. has the following exp/degree position(s) in Burlington, Massachusetts: ***IPR Patent Engineer:** Identify and evaluate inventions; support the IPR process; manage patent portfolio; and collaborate with inventors, technical experts and external patent attorneys. ID# NOK-MA-IPR. Mail resume to: Nokia Recruiter, 2301 N. Greenville Ave., Suite 175, Richardson, TX 75082. MUST REFERENCE ID#. Equal Opportunity Employer.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Palo Alto, CA. **Technology Consultant.** Reqs. exp. in at least one of these: Database, ETL and BI design and build in at least one of the following areas: business intelligence, reporting, business performance management and data warehousing. Exp. with at least one of the following tools: Ab Initio, Informatica, Datastage, Ascential, Business Objects, Cognos, or other info. mgt. tools and disciplines. Travel to various unanticipated worksites throughout the U.S. Reqs. incl. Bachelor's degree or foreign equiv. in CS, CE, Electrical Eng., Electronic Eng., or related field of study & 5 yrs. of related exp. Send resume & refer to job #KBTC3BS. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Yorktown Heights, NY. **Technical Consultant.** Reqs. exp. with Weblogic, Websphere, MSSQL, Oracle, SQL Server, IIS, Tomcat; Windows desktop & server

operating systems (NT, 2000, 2003); HTTP Technologies; UNIX administration; Sun Solaris, HP/UX, Red Hat Linux; Java, C & C++ programming. Reqs. incl. Masters degree or foreign equiv. in CS, CE, EE or related. Send resume & refer to job #YORNGO. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

CISCO SYSTEMS, INC. is accepting resumes for the following positions: **CALIFORNIA:** San Jose/Milpitas/Santa Clara, User-Centered Design Engineer (Ref# SJ65IC). **COLORADO:** Boulder, IT Project Manager (Ref# BOU1IC). Englewood, Network Consulting Engineer (Ref# ENG1IC). **NORTH CAROLINA:** Research Triangle Park, IT Analyst (Ref# RTP10IC), Advanced Services Project Manager (Ref# RTP11IC). **TEXAS:** Richardson, Business Development Manager (Ref# RIC8IC). **WASHINGTON:** Bellevue, Network Consulting Engineer (Ref# BEL1IC). Please mail resumes with job reference number to Cisco Systems, Inc., Attn: Jasbir Walsh, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE. www.cisco.com.

ERICSSON INC. has the following degree/exp. positions open in Warrendale, PA. ***Engineer-Software:** Exp. with C/C++ and Linux; networking process to include IP, TCP/IP, & router technologies; & writing tech. design documentations with UML. (ID# 08-PA-ES1); ***Hardware Engineer:** Exp. with HW design & troubleshooting within programmable logic & diagnosing problems & ASIC/programmable logic areas; & schematic capture tools, PCB lay out, & signal integrity analyst. (ID# 08-PA-HW); ***Engineer-Software:** Work with C, C++, & network protocols; developing syst. for embedded SW; designing and developing Graphical User Interfaces, & audio & video streams. (ID# 08-PA-ES2) ***Software Engineer:** Exp. with TCP/IP routing protocol develop; IP multicast testing in DSL equip; TCL/TK automation platform develop; & network modules testing. (ID#08-PA-SW) *Hardware Development Engineer: Exp. with VHDL, Verilog, & Syst. Verilog. (ID#08-PA-HDE) Send resumes to Attn: S. Bernola - HR, Ericsson Inc., 5000 Marconi Drive, Warrendale, PA 15086; reference specific Job ID# when applying. No Phone calls.

AJINOMOTO USA in Ft Lee, NJ seeks **Assistant Mgr – MIS Dept** to analyze user requirements, procedures & problems to automate processing & improve existing computer sales & distribution & other logistic applications; Req BS in Comp Sci or related field & 2 yrs exp performing similar job duties. Email resume to lamendolas@ajiusa.com.

ENTRISPHERE INC. has the following degree/exp. positions open in Santa, Clara, CA. ***Senior Systems Test Engineer:** Exp. with interoperability testing with multi-vendor CPE for Triple play voice & data serv. using DSL technologies; knowledge of Ethernet protocols & IP testing equip. (Job ID#08-ENT-STE) ***Senior Software Engineer:** Exp. with

embedded development using RTOS & C; Ethernet Layer 2 & QoS develop; carrier class telecom or datacom SW develop; knowledge in VLAN & IPTV serv. model, and/or ITU interface specifications. (Job ID#08-ENT-SWE). Send resume to D. Ehrsam-HR, Entrisphere, Inc., 2770 San Tomas Expressway, Santa Clara, CA 95051; reference specific Job ID# when applying. No phone calls.

INFORMATION SYSTEMS ANALYST: Compile computation needs for information management systems using local area networks, LAN and WAN under the supervision of the IT Manager. Assist in setting up a centralized single platform/central database. Troubleshoot, recover and maintain an operational system on 2000NT Server and networks. Maintain host connectivity multi-user environment. Install upgraded business software including Business Works, GoldMine, and SalesLogix. Req: AA in Science or Eng. or its equiv. & 2 yrs. exp. 40 hr/wk. interview/Job Site; Milpitas, CA. Send Resume to: Global Finance, 1323 Jacklin Rd., Milpitas, CA 95035.

PROGRAMMER ANALYST. Develop Web Pages using ASP/HTML/VBScript/JavaScript, create .NET applications using VB, C#, Tune SQL Scripts for website performance optimization, client interaction. Create test plans & conduct unit/integration test. Identify logical errors & modify programs. Provide production support, website configuration & release management. Perform database backup & restore procedures. Req: Masters Deg. In Comp Sci, Comp Eng or Elec Eng. 40 Hr/wk. Job/Interview Site: Northbrook, IL. Send resume with Job# DTSIEEE to: Digitron Solutions LLC by Email at hr@digitronsolutions.net.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Farmington Hills, MI. **Customer Project/Program Manager.** Reqs. project mgt exp.; exp. managing multiple complex tasks in order to meet customer business needs; & exp. managing the development of complex business solutions. Reqs. incl. Bachelor's degree or foreign equiv. in CS, EE, MIS, or related field of study & 3 yrs. of related exp. Send resume & refer to job #FAHVPE. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

KING ABDULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY (KAUST)
Faculty Openings in Computer Science and Applied Mathematics

King Abdullah University of Science and Technology (KAUST) is being established in Saudi Arabia as an international graduate-level research university dedicated to inspiring a new age of scientific achievement that will benefit the region and the world. As an independent and merit-based institution and one of the best endowed universities in the world, KAUST intends to become a major new contributor to the global network of collaborative research. It will enable researchers from around the globe to work together to solve challenging scientific and technological problems. The admission of students, the appointment, promotion and retention of faculty and staff, and all the educational, administrative and other activities of the University shall be conducted on the basis of equality, without regard to race, color, religion or gender.

KAUST is located on the Red Sea at Thuwal (80km north of Jeddah). Opening in September 2009, KAUST welcomes exceptional researchers, faculty and students from around the world. To be competitive, KAUST will offer very attractive base salaries and a wide range of benefits. Further information about KAUST can be found at http://www.kaust.edu.sa/.

KAUST invites applications for faculty positions at all ranks (Assistant, Associate, Full) in Applied Mathematics (with domain applications in the modeling of biological, physical, engineering, and financial systems) and Computer Science, including areas such as Computational Mathematics, High-Performance Scientific Computing, Optimization, Computer Systems, Software Engineering, Algorithms and Computing Theory, Artificial Intelligence, Graphics, Databases, Human-Computer Interaction, Computer Vision and Perception, Robotics, and Bio-Informatics (this list is not exhaustive). KAUST is also interested in applicants doing research at the interface of Computer Science and Applied Mathematics with other science and engineering disciplines. High priority will be given to the overall originality and promise of the candidate's work rather than the candidate's sub-area of specialization within Applied Mathematics and Computer Science.

An earned Ph.D. in Applied Mathematics, Computer Science, Computational Mathematics, Computational Science and Engineering, or a related field, evidence of the ability to pursue a program of research, and a strong commitment to graduate teaching are required. A successful candidate will be expected to teach courses at the graduate level and to build and lead a team of graduate students in Master's and Ph.D. research.

Applications should include a curriculum vita, brief statements of research and teaching interests, and the names of at least 3 references for an Assistant Professor position, 6 references for an Associate Professor position, and 9 references for a Full Professor position. Candidates are requested to ask references to send their letters directly to the search committee. Applications and letters should be sent via electronic mail to kaust-search@cs.stanford.edu. The review of applications will begin immediately, and applicants are strongly encouraged to submit applications as soon as possible; however, applications will continue to be accepted until December 2009, or all 10 available positions have been filled.

In 2008 and 2009, as part of an Academic Excellence Alliance agreement between KAUST and Stanford University, the KAUST faculty search will be conducted by a committee consisting of professors from the Computer Science Department and the Institute of Computational and Mathematical Engineering at Stanford University. This committee will select the top applicants and nominate them for faculty positions at KAUST. However, KAUST will be responsible for actual recruiting decisions, appointment offers, and explanations of employment benefits. The recruited faculty will be employed by KAUST, not by Stanford. Faculty members in Applied Mathematics and Computer Science recruited by KAUST before September 2009 will be hosted at Stanford University as Visiting Fellows until KAUST opens in September 2009. At Stanford, these Visiting Fellows will conduct research with Stanford faculty and will occasionally teach courses.

BOOKSHELF

Mechanisms: New Media and the Forensic Imagination, Matthew G. Kirschenbaum. The author examines new media and electronic writing against the textual and technological primitives that govern writing, inscription, and textual transmission in all media: erasure, variability, repeatability, and survivability.

Drawing a distinction between "forensic materiality" and "formal materiality," the author uses applied computer forensics techniques in his study of new media works. Just as the humanities discipline of textual studies examines books as physical objects and traces different variants of texts, computer forensics encourages us to perceive new media in terms of specific versions, platforms, systems, and devices.

The author demonstrates these techniques in media-specific readings of three landmark works of new media and electronic literature, all from the formative era of personal computing: the interactive fiction game *Mystery House*, Michael Joyce's *Afternoon: A Story*, and William Gibson's electronic poem Agrippa.

Drawing on newly available archival resources for these works, the author uses a hex editor and disk image of *Mystery House* to conduct a "forensic walkthrough" that explores critical reading strategies linked to technical praxis, examines the multiple versions and revisions of *Afternoon* to address the diachronic dimension of electronic textuality, and documents the volatile publication and transmission history of Agrippa as an illustration of the social aspect of transmission and preservation.

MIT Press; mitpress.mit.edu; 0-262-11311-2; 240 pp.

Introduction to Software Testing, Paul Ammann and Jeff Offutt. Extensively class tested, this text takes an innovative approach to explaining the process of software testing. It defines testing as the process of applying a few well-defined, general-purpose test criteria to a structure or model of the software.

The text's structure incorporates the latest innovations in testing, including techniques to test modern types of software such as OO, Web applications, and embedded software.

Cambridge University Press; www.cambridge.org; 978-0-521-88038-1; 344 pp.

Sarbanes-Oxley IT Compliance Using Open Source Tools, 2nd ed., Christian B. Lahti and Roderick Peterson. This book describes the many open source cost-saving opportunities that public companies can explore in their IT enterprise to meet the mandatory compliance requirements of the Sarbanes-Oxley (SOX) act. It also demonstrates by example and technical reference both the infrastructure components for open source that can be made compliant and the open source tools that can aid in the journey to compliance.

Each chapter begins with IT business and executive considerations for open source and SOX compliance. The text includes specific examinations of open source applications and tools that relate to the given subject matter. A bootable CD provides fully configured running demonstrations of open source tools as a valuable technical reference for implementing the book's concepts.

Syngress; www.syngress.com; 978-1-59749-216-4; 448 pp.

Eating the IT Elephant: Moving from Greenfield Development to Brownfield Product Page, Richard Hopkins and Kevin Jenkins. Most conventional approaches to IT development assume that engineers are building entirely new systems. But today "greenfield" development is a rarity. Nearly every project exists in the context of current complex system landscapes that are often poorly documented and poorly understood. Here, the authors—senior IBM system architects—offer a new approach fully optimized for today's "brownfield" development projects.

This books shows readers why accumulated IT complexity is the root cause of large-scale project failure—and how to overcome that complexity. The authors explain how to manage all four phases of a brownfield project, leveraging breakthrough collaboration and communication tools and techniques—including Web 2.0, semantic software engineering, model-driven development and architecture, and even virtual worlds.

IBM Press; www.ibmpressbooks.com; 0-13-713012-0; 256 pp.

Communications Engineering: Essentials for Computer Scientists and Electrical Engineers, R.C.T. Lee, Mao-Ching Chiu, and Jung-Shan Lin. The authors' observation that convergence requires computer science students to gain a better understanding of communications concepts motivated the writing of this book. The text directly addresses this gap, thoroughly delivering to computer science students the key essentials.

The authors walk the reader through the Fourier transform, analog and digital modulation techniques, multiple access communications, spread-spectrum communications, and source and channel coding. This book has been used in the classroom as an introductory text in university electrical engineering programs.

Wiley; www.wiley.com; 978-0-470-82245-6; 240 pp.

Send book announcements to newbooks@computer.org.

Computer Science Enrollments Drop

Each year, the Computing Research Association conducts its Taulbee Survey of PhD-granting departments of computer science and computer engineering in North America. The survey documents trends in student enrollment, postgraduate employment, and faculty salaries. The CRA releases preliminary results on undergraduate enrollment in March and full results in May.

FEWER COMPUTER SCIENCE UNDERGRADUATES

The number of students enrolled in computer science has fallen for several years. In fall 2007, the number of new computer science majors (7,915) was half of what it was in fall 2000 (15,958). Between 2005/2006 and 2006/2007, total enrollments declined 18 percent to 28,675. Overall, enrollments have dropped 49 percent from their peak in 2001/2002, while the median number of students enrolled in each department has fallen 53 percent since 2000/2001.

DEGREE PRODUCTION SINKS

The decline in undergraduate numbers has had a significant impact on degree production. After posting several years of increases, the total number of bachelor's degrees awarded by PhD-granting computer science departments fell 43 percent to 8,021 between 2003/2004 and 2006/2007. The median number of degrees granted per department declined 39 percent to 42. The CRA suggests that the sustained drop in total enrollments and student interest in computer science as a major will cause degree production numbers to continue their slide over the next few years.

A steep drop in degree production among computer science departments has happened before. According to the US National Science Foundation, undergraduate computer science production nearly quadrupled between 1980 and 1986 to more than 42,000 degrees. This period was followed by a swift decline, leveling off in the early 1990s, with the number of degrees granted hovering around 25,000. During the late 1990s, computer science degree production again surged, reaching more than 57,000 in 2004.

COMPUTING RESEARCH ASSOCIATION

The CRA is an association of more than 200 academic departments of computer science, computer engineering, and related fields. It includes organizations in industry, government, and academia that engage in basic computing research as well as affiliated professional societies. The Taulbee Survey is named in honor of the late Orrin E. Taulbee of the University of Pittsburgh, who from 1974 to 1984 conducted the survey for the Computer Science Board, the CRA's predecessor.

IEEE Computer Society Petition Candidate Nominations Due 6 May

In preparation for the annual election of its officers, the IEEE Computer Society welcomes the nominations of candidates for office. To add a name to the ballot, a member can submit a petition to the Society secretary via mail, fax, or e-mail indicating the desired office, the starting date of the term, and the name of the candidate. The petition must also include the signatures of voting members of the Society: at least 250 for Board term nominees and at least 1,000 for officer nominees. Petition "signatures" can simply indicate the signing member's name and member number. A voting member can sign only one Board of Governors petition and one officer petition for each other office.

For each petition nomination, the Society secretary must receive a statement signed by the nominee indicating a willingness and availability to serve if elected. Petition candidates must also submit biographical data, position statements, and 300-dpi digital images or studio-quality head-and-shoulders photographs to the Society secretary.

All petition nominee materials must be received by **6 May**. Send them to Computer Society Secretary Michel Israel at IEEE Computer Society, 1828 L. Street, NW, Suite 1202, Washington, DC 20036-5104; or m.israel@computer.org.

Society Rolls Out New Certification

In response to industry requests for a way to confirm the skill and knowledge levels of those just entering the software field, the IEEE Computer Society has created the Certified Software Development Associate certification, a new program created for those entering the software development profession.

CSDA certification takes a broad view of software development and validates knowledge of the foundations of computer science, mathematics, and engineering. Core software engineering principles covered include software construction, design, testing, requirements, and methods. The CSDA exam centers on key concepts addressed in *The Guide to the Software Engineering Body of Knowledge* (SWEBOK) and *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering* (SE2004.)

Certified Software Development Associate exam questions cover topics in each of the following areas:

I. Software Requirements (6-8% questions)
II. Software Design (7-9% questions)
III. Software Construction (8-10% questions)
IV. Software Testing (6-8% questions)
V. Software Maintenance (6-8% questions)
VI. Software Configuration Management (2-4% questions)
VII. Software Engineering Management (2-4% questions)
VIII. Software Engineering Process (4-6% questions)
IX. Software Engineering Methods (4-6% questions)
X. Software Quality (4-6% questions)
XI. Software Engineering Professional Practice (5-7% questions)
XII. Software Engineering Economics (3-5% questions)
XIII. Computing Foundations (8-10% questions)
XIV. Mathematical Foundations (8-10% questions)
XV. Engineering Foundations (8-10% questions)

Candidates can prepare for the exam by reviewing SWEBOK and selectively reading references in areas of software engineering that the exam covers. Candidates can also review applicable textbooks or university course notes. Two textbooks that cover the basics of software engineering are Ian Sommerville's *Software Engineering*, 8th edition (Addison-Wesley, 2007) and Richard H. Thayer and colleagues' two-volume *Software Engineering*, 3rd edition (John Wiley & Sons, 2005).

To learn more about the CSDA certification process, launched in beta mode early in 2008, visit www.computer.org/certification/csda.

Nominations for Cray and Fernbach Awards Due 1 July

Each fall, the IEEE Computer Society presents two of the most distinguished awards in computing. The Seymour Cray Computer Science & Engineering Award and the Sidney Fernbach Award recognize individuals for making outstanding contributions to computer science and engineering.

Supercomputing pioneer Seymour Cray was well known for discovering unconventional solutions to vexing problems. The IEEE Computer Society's Seymour Cray Computer Science & Engineering Award recognizes individuals whose contributions to high-performance computing systems best reflect Cray's innovative, creative spirit. Recipients of the Cray Award also receive a crystal memento, an illuminated certificate, and a $10,000 honorarium.

Sidney Fernbach, an early researcher in high-performance computing, made important strides in the use of high-performance computers to solve large computational problems. In 1992, the Computer Society established the Sidney Fernbach Memorial Award to recognize individuals who have made notable contributions to developing applications for high-performance computing. The Fernbach award winner receives a certificate of recognition and a $2,000 honorarium.

Recipients of both the Cray and Fernbach awards will accept their honors during a special awards ceremony at SC 2008 in Austin, Texas, this November.

Computer Society awards recognize technical achievements, contributions to engineering education, and service to the Society or the profession. Nominations for the Cray and Fernbach awards are due by **1 July**. Other Computer Society awards with 1 July deadlines include the Taylor L. Booth Education Award and the Computer Science & Engineering Undergraduate Teaching Award. To nominate a candidate for any IEEE Computer Society award, visit http://awards.computer.org/ana.

IEEE Computer Society Launches Peer-Reviewed Webinars

The IEEE Computer Society is launching a peer-reviewed webinar series under the banner of Computing Now, a new initiative designed to bring more online technical content to its members and raise awareness of the Society's 14 technical magazines.

The Computing Now Webinar Series effort debuted with a webinar on standardizing software process improvement initiatives by Computer Society president-elect Susan (Kathy) Land, CSDP, principal software and systems engineer at MITRE. Land has more than 20 years of industry experience in practical software engineering methodologies, information systems management, and software development team leadership.

The six Computer Society webinars set for 2008 offer expanded access to the Society's wide-ranging intellectual property and the expertise of its members and contributors. In contrast to many commercial offerings, the free webinars are delivered by real experts—authors, researchers, and scientists who seek to advance the profession.

To learn more about the free webinars, presented in cooperation with ON24, go to www.computer.org/webinar/standardizing.

Editor: Bob Ward, Computer, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; bnward@computer.org

CALLS FOR ARTICLES FOR IEEE CS PUBLICATIONS

IEEE Pervasive Computing seeks articles for a December 2008 issue on environmental sustainability. The creation, use, and disposal of large quantities of pervasive technologies such as sensors and mobile devices have strong implications for resource consumption and waste production. Submissions should address design for technology reuse, repurposing, or lifetime extension; sensor network applications that support the efficient use or protection of natural resources; novel systems, devices, or interfaces that support stewardship of the natural environment; and resource-efficient system design, among other topics.

Articles are due by **23 June**. Visit www.computer.org/pervasive to view detailed author instructions and the complete call for papers.

CALLS FOR PAPERS

HiPC 2008, Int'l Conf. on High-Performance Computing, 17-20 Dec., Bangalore, India; Submissions due **12 May;** www.hipc.org/hipc2008/papers.htm

Music And Multimedia 2008, The Use of Symbols to Represent Music and Multimedia Objects, 8 Oct., Lugano, Switzerland; Submissions due **31 May;** http://conferences.computer.org/icws/2008/call-for-papers.html

WI-IAT 2008, IEEE/WIC/ACM Int'l Conf. on Web Intelligence & IEEE/WIC/ACM Int'l Conf. on Intelligent Agent Technology, 9-12 Dec., Sydney; Submissions due **10 July;** http://datamining.it.uts.edu.au/wi08/html/wi/?index=cfp

CALENDAR

MAY 2008

4-8 May: VTS 2008, 28th IEEE VLSI Test Symp., San Diego; www.tttc-vts.org

5-7 May: ISORC 2008, 11th IEEE Int'l Symp. on Object/Component/Service-Oriented Real-Time Distributed Computing, Orlando, Florida; http://ise.gmu.edu/isorc08

7-9 May: EDCC 2008, 7th European Dependable Computing Conf., Kaunas, Lithuania; http://edcc.dependability.org

10-18 May: ICSE 2008, 30th Int'l Conf. on Software Eng., Leipzig, Germany; http://icse08.upb.de

12-13 May: HST 2008, 8th IEEE Int'l Conf. on Technologies for Homeland Security, Waltham, Massachusetts; www.ieeehomelandsecurityconference.org

12-16 May: AAMAS 2008, 7th Int'l Conf. on Autonomous Agents and Multiagent Systems, Estoril, Portugal; http://gaips.inesc-id.pt/aamas2008

14-16 May: ICIS 2008, 7th IEEE Int'l Conf. on Computer and Information Science, Portland, Oregon; http://acis.cps.cmich.edu:8080/ICIS2008

19-22 May: CCGrid 2008, 8th IEEE Int'l Symp. on Cluster Computing and the Grid, Lyon, France; http://ccgrid2008.ens-lyon.fr

22-24 May: ISMVL 2008, 38th Int'l Symp. on Multiple-Valued Logic, Dallas; http://engr.smu.edu/ismvl08

24 May: ULSI 2008, 17th Int'l Workshop on Post-Binary ULSI Systems (with ISMVL), Dallas; http://engr.smu.edu/ismvl08

25-28 May: GPC 2008, 3rd Int'l Conf. on Grid and Pervasive Computing, Kunming, China; http://grid.hust.edu.cn/gpc2008

25-28 May: WaGe 2008, 3rd Int'l Workshop on Workflow Management and Applications in Grid Environments (with GPC), Kunming, China; www.swinflow.org/confs/WaGe08/WaGe08.htm

25-28 May: WMCS 2008, 4th Int'l Workshop on Mobile Commerce and Services (with GPC), Kunming, China; www.engr.sjsu.edu/wmcs

25-29 May: ETS 2008, IEEE European Test Symp., Verbania, Italy; www.cad.polito.it/~ets08

JUNE 2008

2-4 June: Policy 2008, IEEE Int'l Workshop on Policies for Distributed Systems and Networks, Palisades, New York; www.policy-workshop.org/2008

Submission Instructions

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting.

Visit www.computer.org/conferences for instructions on how to submit conference or call listings as well as a more complete listing of upcoming computer-related conferences.

4-6 June: SMI 2008, IEEE Int'l Conf. on Shape Modeling and Applications, Stony Brook, New York; www.cs.sunysb.edu/smi08

10-13 June: ICPC 2008, 16th IEEE Int'l Conf. on Program Comprehension, Amsterdam; www.cs.vu.nl/icpc2008

11-13 June: SIES 2008, IEEE 3rd Symp. on Industrial Embedded Systems, La Grande Motte, France; http://www.lirmm.fr/SIES2008

11-13 June: SUTC 2008, IEEE Int'l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, Taiwan; http://sutc2008.csie.ncu.edu.tw

17-20 June: ICDCS 2008, 28th Int'l Conf. on Distributed Computing Systems, Beijing; www.ieee-icdcs.org

23-25 June: CSF 2008, 21st IEEE Computer Security Foundations Symp. (with LICS), Pittsburgh; www.cylab.cmu.edu/CSF2008

23-25 June: WETICE 2008, 17th IEEE Int'l Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, Rome; www.sel.uniroma2.it/wetice08/venue.htm

23-26 June: ICITA 2008, 5th Int'l Conf. on Information Technology and Applications, Cairns, Australia; www.icita.org

24-27 June: LICS 2008, IEEE Symp. on Logic in Computer Science, Pittsburgh; www2.informatik.hu-berlin.de/lics/lics08

JULY 2008

6-13 July: ICALP 2008, 35th Int'l Colloquium on Automata, Languages, and Programming, Reykjavik, Iceland; www.ru.is/icalp08/workshops.html

7-11 July: Services 2008, IEEE Congress on Services, Hawai'i; http://conferences.computer.org/services/2008

8 July: WS-Testing 2008, IEEE Int'l Conf. on Computer and Information Technology (with SCC), Hawai'i; http://conferences.computer.org/services/2008/WS-Testing-2008.htm

8-11 July: CIT 2008, IEEE Int'l Conf. on Computer and Information Technology, Sydney, Australia; http://attend.it.uts.edu.au/cit2008

8-11 July: SCC 2008, IEEE Int'l Conf. on Services Computing, Honolulu; http://conferences.computer.org/scc/2008

8-11 July: SOPOSE 2008, 3rd Int'l Workshop on Service- and Process-Oriented Software Eng. (with SCC), Honolulu; www.dsl.uow.edu.au/sopose/index.php?l1=sopose08

10-13 July: ICPC 2008, 16th IEEE Int'l Conf. on Program Comprehension, Amsterdam; www.cs.vu.nl/icpc2008

11-12 July: NCA 2008, 7th IEEE Int'l Symp. on Network Computing and Applications, Cambridge, Massachusetts; www.ieee-nca.org

Call for Articles for *Computer*

Computer seeks articles for a December 2008 special issue on trust management in Web service environments.

A key challenge for the envisioned service Web, where services are considered as first-class objects, is to provide a trusted framework for enabling composition and selection of Web services in a large, highly volatile, and dynamic environment. The intentional lack of any global monitoring system, while undeniably desirable, has also exacerbated the problem of trust management in Web service environments. The inherently open and unpredictable nature of Web service environments means that traditional approaches are of little help in providing a framework for interactions. This special issue of *Computer* will cover the core challenges and solutions to enabling trust in Web service environments.

Topics of interest include trust models and metrics, bootstrapping trust among Web services, trust tampering detection and prevention, trust in Web service standards, agent- and reputation-based trust management, and case studies for trust management among Web services.

Authors should present a proof of concept for any novel technique and discuss the significance and applicability of their proposed architecture or system. All papers are subject to expert peer review.

Direct inquiries to the guest editors, Elisa Bertino, Purdue University, bertino@cs.purdue.edu, or Athman Bouguettaya, Commonwealth Scientific and Industrial Research Organization, athman.bouguettaya@csiro.au. Paper submissions are due by **1 June**. Complete submission instructions are available at www.computer.org/portal/pages/computer/content/author.html.

Events in 2008

MAY

4-8 . . . . . VTS 2008
5-7 . . . . . .ISORC 2008
7-9 . . . . . EDCC 2008
10-18 . . . . . ICSE 2008
12-13 . . . . . HST 2008
12-16 . . . . . .AAMAS 2008
14-16 . . . . . ICIS 2008
19-22 . . . . . .CCGrid 2008
22-24 . . . . . .ISMVL 2008
24 . . . . . ULSI 2008
25-28 . . . . . GPC 2008
25-28 . . . . . .WaGe 2008
25-28 . . . . . WMCS 2008
25-29 . . . . . .ETS 2008

JUNE

2-4 . . . . . Policy 2008
4-6 . . . . . SMI 2008
10-13 . . . . . .ICPC 2008
11-13 . . . . . SIES 2008
11-13 . . . . . SUTC 2008
17-20 . . . . . ICDCS 2008
23-25 . . . . . CSF 2008
23-25 . . . . . WETICE 2008
23-26 . . . . . ICITA 2008
24-27 . . . . . LICS 2008

JULY

6-13 . . . . . .ICALP 2008
7-11 . . . . . Services 2008
8 . . . . . .WS-Testing 2008
8-11 . . . . . .CIT 2008
8-11 . . . . . SCC 2008
8-11 . . . . . SOPOSE 2008
10-13 . . . . . .ICPC 2008
11-12 . . . . . .NCA 2008
28 July-1 Aug . . . . . COMPSAC 2008
28 July-1 Aug . . . . . .ESAS 2008
28 July-1 Aug . . . . . .SAINT 2008

28 July-1 Aug: COMPSAC 2008, 32nd IEEE Int'l Computer Software and Applications Conf. (with SAINT), Turku, Finland; www.compsac.org

28 July-1 Aug: ESAS 2008, 3rd IEEE Int'l Workshop on Eng. Semantic Agent Systems (with COMPSAC), Turku, Finland; http://conferences.computer.org/compsac/2008/workshops/ESAS2008.html

28 July-1 Aug: SAINT 2008, IEEE/IPSJ Symp. on Applications and the Internet (with COMPSAC), Turku, Finland; www.saintconference.org

AUGUST 2008

3-5 Aug: ISECS 2008, Int'l Symp. on Electronic Commerce and Security, Guangzhou, China; www.iita-conference.org/isecs

4-6 Aug: ACSAC 2008, 13th IEEE Asia-Pacific Computer Systems Architecture Conf., Hsinchu, Taiwan; www.ccrc.nthu.edu.tw/acsac2008

4-7 Aug: ICSC 2008, 2nd IEEE Int'l Conf. on Semantic Computing, Santa Clara, California; http://icsc.eecs.uci.edu

17-20 Aug: ICGSE 2008, Int'l Conf. on Global Software Eng., Bangalore, India; www.icgse.org

SEPTEMBER 2008

1-3 Sept: AVSS 2008, 5th IEEE Int'l Conf. on Advanced Video and Signal-Based Surveillance, Santa Fe, New Mexico; www.cpl.uh.edu/avss2008

23-26 Sept: ICWS 2008, IEEE Int'l Conf. on Web Services, Beijing; http://conferences.computer.org/icws/2008

28-29 Sept: SCAM 2008, 8th IEEE Int'l Working Conf. on Source Code Analysis and Manipulation (with ICSM), Beijing; www2008.ieee-scam.org

IEEE LCN 2008

The IEEE Conference on Local Computer Networks, sponsored by the IEEE Computer Society, is one of the networking industry's longest-running conferences.

LCN 2008 focuses on practical, leading-edge applications and research in the area of computer networks. An informal, workshop-style atmosphere offers opportunities for speakers, panelists, and attendees to spend unstructured time together. Attendees come from around the world, including North America, Europe, and the Pacific Rim.

Topics of paper sessions include high-speed networking; high-performance protocols; local, metropolitan, and wide area networks; internetworking; network and protocol design; wireless networks; routing and switching; multimedia networks; distributed systems; and real-time networks.

IEEE LCN 2008 takes place **20-23 October** in Montreal. For further details, visit the IEEE LCN website at www.ieeelcn.org.

SOFTWARE TECHNOLOGIES

Dynamic Software Product Lines

Svein Hallsteinsen, SINTEF ICT
Mike Hinchey, Lero—The Irish Software Engineering Research Centre
Sooyong Park, Sogang University
Klaus Schmid, University of Hildesheim

DSPLs produce software capable of adapting to changes in user needs and resource constraints.

> Any customer can have a car painted any colour that he wants so long as it is black.
>
> —Henry Ford, *My Life and Work*, 1922

Henry Ford, founder of the car company that bears his name, is widely regarded as the father of assembly-line automation, which he introduced and expanded in his factories producing Model Ts between 1908 and 1913.

What's less known is that Ford achieved this innovation through the use of interchangeable parts, based on earlier work by Honoré Blanc and Eli Whitney. This significantly streamlined the production process over earlier efforts in which parts were often incompatible and one difference in a product meant restarting the entire process.

The result was economies of scale and a line of motor cars that were affordable, built quickly, and of high quality, even if certain choices—paint color, for example—were extremely limited.

PRODUCT LINE ENGINEERING

Ford's ideas influenced the development of product line engineering (PLE), which seeks to achieve something conceptually similar to economies of scale: *economies of scope*. As Jack Greenfield and colleagues explain in *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* (Wiley, 2004), "Economies of scale arise when multiple identical instances of a single design are produced collectively, rather than individually. Economies of scope arise when multiple similar but distinct designs and prototypes are produced collectively, rather than individually."

Economies of scope imply mass customization, which can be defined as "producing goods and services to meet individual customers' needs with near mass production efficiency" (M.M. Tseng and J. Jiao, "Mass Customization," G. Salvendy, ed., *Handbook of Industrial Engineering: Technology and Operations Management*, John Wiley & Sons, 2001, pp. 684-709).

PLE provides a means of customizing variants of mass-produced products. Its key aim is to create an underlying architecture for an organization's product platform in which core assets can be reused to engineer new products from the basic family, thereby increasing variability and choice while simultaneously decreasing development cost and lead time.

The software development community has caught on to the usefulness of this approach with the idea of *software product lines*.

SOFTWARE PRODUCT LINES

The Software Engineering Institute (SEI) defines an SPL as "a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" (www.sei.cmu.edu/productlines).

Developers have successfully applied SPLs in many different domains—including avionics, medical devices, and information systems—in a wide variety of organizations ranging in size from five developers to more than a thousand (www.sei.cmu.edu/productlines/plphof.html). Using this approach has consistently achieved improvements in time to market, cost reduction, and quality (F.J. van der Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*, Springer, 2007).

A fundamental principle of SPLs is *variability management*, which involves separating the product line into three parts—common components, parts common to some but not all products, and individual products with their own specific requirements—and managing these throughout development. Using

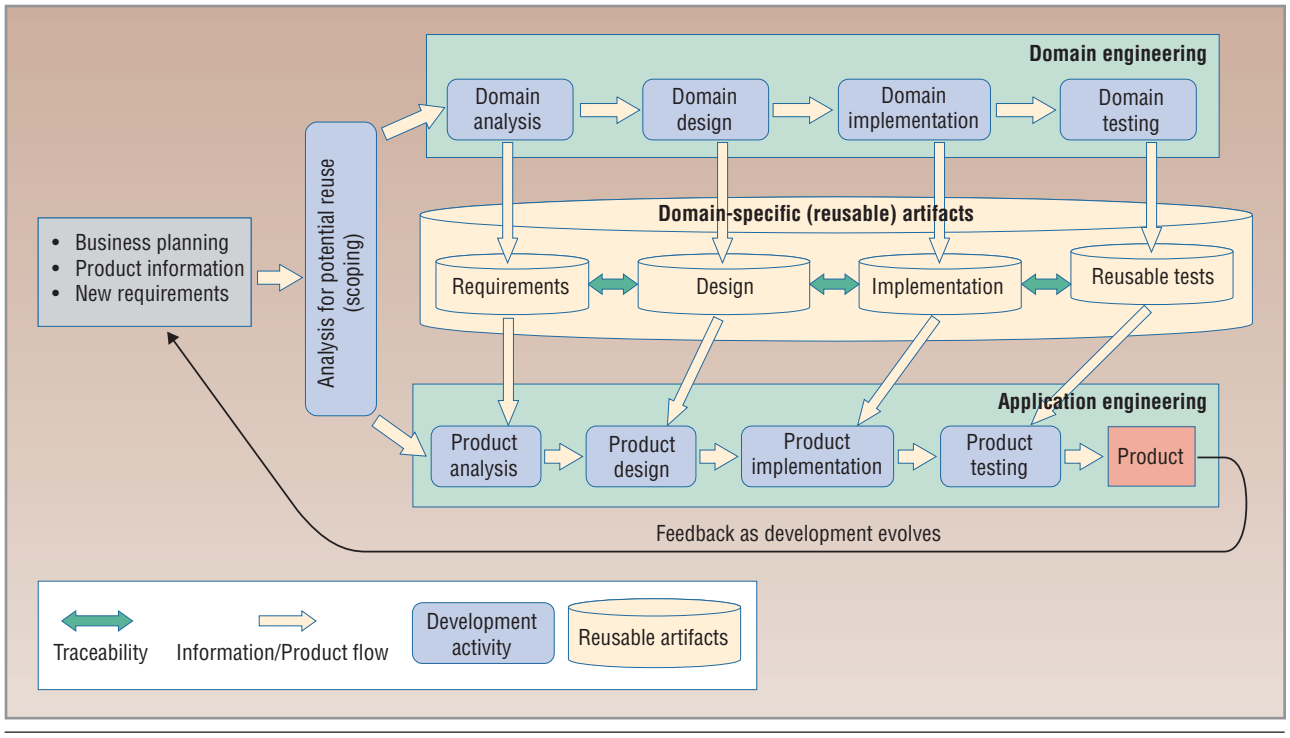


Figure 1. Software product lines. SPLs use a two-life-cycle approach that separates domain and application engineering.

SPLs seeks to maximize reusable variation and eliminate wasteful generic development of components used only once.

As Figure 1 shows, SPLs employ a two-life-cycle approach that separates domain and application engineering. *Domain engineering* involves analyzing the product line as a whole and producing any common (and reusable) variable parts. *Application engineering* involves creating product-specific parts and integrating all aspects of individual products. Both life cycles can rely on fundamentally different processes—for example, agile application engineering combined with plan-driven domain engineering.

DYNAMIC SPLS

In emerging domains such as ubiquitous computing, service robotics, unmanned space and water exploration, and medical and life-support devices, software is becoming increasingly complex with extensive variation in both requirements and resource constraints. Developers face growing pressure to deliver high-quality software with additional functionality, on tight deadlines, and more economically.

In addition, modern computing and network environments demand a higher degree of adaptability from their software systems. Computing environments, user requirements, and interface mechanisms between software and hardware devices such as sensors can change dynamically during runtime.

Because it's impossible to foresee all the functionality or variability an SPL requires, there's a need for *dynamic* SPLs that produce software capable of adapting to fluctuations in user needs and evolving resource constraints. DSPLs bind variation points at runtime, initially when software is launched to adapt to the current environment, as well as during operation to adapt to changes in the environment.

Although traditional SPL engineering recognizes that variation points are bound at different stages of development, and possibly also at runtime, it typically binds variation points before delivery of the software. In contrast, DSPL engineers typically aren't concerned with pre-runtime variation points. However, they recognize that in practice mixed approaches might be viable, where some variation points related to the environment's static properties are bound before runtime and others related to the dynamic properties are bound at runtime.

In DSPLs, monitoring the current situation and controlling the adaptation are thus central tasks. The user, the application, or generic middleware can perform these tasks manually or automatically.

Although dynamic software product lines build on the central ideas of SPLs, there are also differences. For example, the focus on understanding the market and letting the SPL drive variability analysis is less relevant to DSPLs, whose primary goal is to adapt to variations in individual needs and situations rather than market forces.

In summary, a DSPL has many, if not all, of the following properties:

- dynamic variability: configuration and binding at runtime,
- changes binding several times during its lifetime,

- variation points change during runtime: variation point addition (by extending one variation point),
- deals with unexpected changes (in some limited way),
- deals with changes by users, such as functional or quality requirements,
- context awareness (optional) and situation awareness,
- autonomic or self-adaptive properties (optional),
- automatic decision making (optional), and
- individual environment/context situation instead of a "market."

Given these characteristics, DSPLs would benefit from research in several related areas. For example, situation monitoring and adaptive decision making are also characteristics of autonomic computing, and DSPL can be seen as one among several approaches to building self-adapting/managing/healing systems.

In addition, dynamically reconfigurable architectures provide mechanisms to rebind variation points at runtime, while multiagent systems, which focus on the use of agents and communities of agents, are particularly useful for evolving systems such as DSPLs.

Interest in DSPLs is growing as more developers apply the SPL approach to dynamic systems. The first workshop on DSPLs was held at the 11th International Software Product Line Conference in Kyoto in 2007. A follow-up workshop will be held at SPLC 2008 in Limerick, Ireland, this September (www.lero.ie/splc2008).

Svein Hallsteinsen *is senior scientist at SINTEF ICT, Trondheim, Norway. Contact him at svein.hallsteinsen@sintef.no.*

Mike Hinchey *is a professor of computer science at Loyola College in Maryland and codirector designate of Lero—The Irish Software Engineering Research Centre. Contact him at mike.hinchey@lero.ie.*

Sooyong Park *is a professor of computer science at Sogang University, Seoul. Contact him at sypark@sogang.ac.kr.*

Klaus Schmid *is a professor of computer science at the University of Hildesheim, Germany. Contact him at schmid@sse.uni-hildesheim.de.*

Editor: Mike Hinchey, Lero—The Irish Software Engineering Research Centre; mike.hinchey@lero.ie

HOW THINGS WORK

Remote Medical Monitoring

Andrew D. Jurik and Alfred C. Weaver
University of Virginia

Body network sensors have become lightweight healthcare assistants—they acquire data, run diagnostics, report adverse events, and even warn of unsafe situations.

The commoditization of computer hardware and software has enabled a new computing paradigm whereby computers will sense, calculate, and act on our behalf, either with or without human interaction as best fits the circumstances. Further, this will occur in an everyday environment, not just when a person is working at a desk.

This paradigm shift was made possible by the inexorable increase in computing capabilities as we moved from mainframes (one computer, many people) to the personal computer (one computer, one person) to ubiquitous computing (many computers, one person). It is not uncommon to find a single person managing a desktop PC, laptop, cell phone, PDA, and portable media player. Today, these devices are discrete and managed individually. But as ubiquitous computing evolves, the computers will become both more numerous and less visible; they will be integrated into everyday life in a way that does not call attention to their presence.

In the context of medicine, ubiquitous computing presents an exciting challenge and a phenomenal opportunity. *Proactive computing* is a form of ubiquitous computing in which computers anticipate the needs of people around them. Wearable computing results from placing computers and sensors on the body to create a *body area network* (BAN) that can sense, process, and report on some set of the wearer's attributes. Proactive computing and wearable computing working in tandem let computers fade into the woodwork, enriching quality of life and engendering independence.

TELEHEALTH

The availability of communication systems such as the public switched telephone network, the Global System for Mobile communication (GSM) network, the Internet, and proprietary wide area networks have enabled some parts of the healthcare industry to transition from in-person visits to remote consultation. Properly configured, telemedicine (the use of communications and information technology to deliver clinical care) can be more cost-effective and convenient for patients, and it is especially attractive for healthcare delivery to remote or underserved populations. Bidirectional videoconferencing is often used as one component of telemedicine to mimic the dynamics of a traditional in-person visit.

Remote medical monitoring—also known as remote patient or healthcare monitoring—expands the usefulness of telemedicine by, at first, treating patients with chronic conditions and diseases by monitoring day-to-day health so that preventive and emergency care can be delivered as needed. As this technology matures and gains acceptance, remote medical monitoring will become the standard procedure for managing certain conditions, including heart disease and diabetes.

SYSTEM ARCHITECTURE

As Figure 1 shows, a typical remote medical monitoring system is a three-tier architecture, each tier distinguished by its locality and functionality within the broader system.

The first tier is the set of sensors that discern signals of interest, then relay information to each other and the data hub.

Tier two, the data hub, is a device that provides more computational capacity, allowing data to be stored or further processed before transmission to some outside medical network via the Internet, GSM, or some other means.

The third tier is the medical network, which is operated by a healthcare provider such as a hospital or telemedicine center where the staff can handle emergency situations. We can imagine the remote medical monitoring system as being a hybrid of a broadcast service and a 9-1-1 service. In broadcast mode, it can provide periodic data updates to physicians regarding a patient's health; in 9-1-1 mode, it can autonomously raise a red flag whenever it detects a dangerous anomaly in monitored data.

Sensors

Applications for sensors, and more specifically wireless sensors, in tier one are growing, and the medical domain represents only one area of that growth. Sensors come in all shapes and sizes, offering different

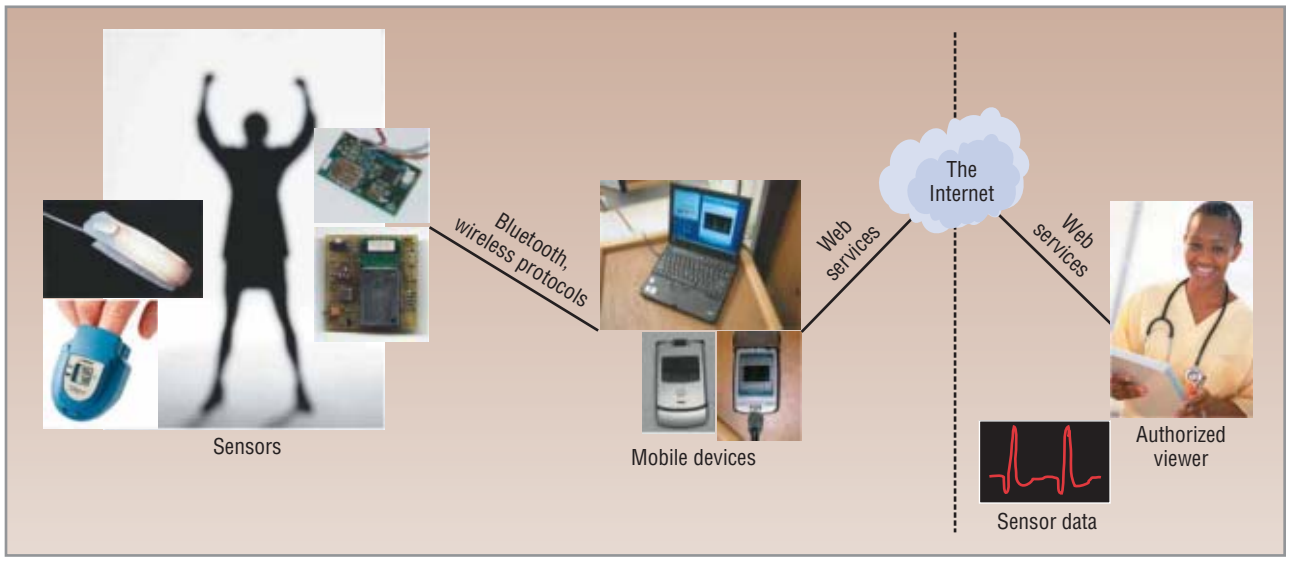


Figure 1. Remote medical monitoring system consists of three tiers: one or more sensors that capture information about the patient, a data hub (such as a PDA, laptop, or cell phone) for local data processing and display, and a medical network that records and analyzes information to detect anomalies.

functionality and accommodating different constraints. Typical medical applications for sensors include monitoring pulse, temperature, motion/acceleration, blood pressure, and pulse oximetry. Physicians use these sensor readings to gain a broader assessment of a patient's medical status.

Wireless sensors communicate via a host of protocols depending on requirements such as power, range, and interoperability with other devices. Given sensors' several limiting characteristics, including battery life and memory capacity, low-power wireless communication is a necessity for any medical monitoring system to be practical. Bluetooth, ZigBee, and the new Wibree standard (intended to be interoperable with Bluetooth) provide promising avenues for leveraging the technologies that commercial mobile devices provide.

Data hub

The tier-two data hub, often a workstation or mobile device such as a cell phone or PDA, fills the roles of data repository and communication mediator. Sensor data is stored on the hub for immediate or later transmission to the external medical network, which can be accomplished automatically or manually. The ability to upload the data manually provides the patient with more control over when and what data is used. Automatic transmission has the advantage that the patient need not remember to upload the data, but it has the potentially detrimental effect that the patient doesn't necessarily know when or even how the data is transmitted.

Medical network

Perhaps the most crucial part of the system is the back-end tier-three medical network that receives the information. This is where the patient places trust in the network to properly guard and protect personal data. Such a network must scale to handle multiple users and a variety of information formats to enable caregivers to make well-informed decisions.

Processing and data visualization assist caregivers by picking out the essential elements in a flood of sensor data from multiple patients. Such a medical portal must also be robust against potential misuses of the system, including false users and misleading data.

REMOTE MEDICAL MONITORING SYSTEMS

Several research groups and commercial vendors have started developing remote medical monitoring systems. All have common threads in their architectures, but they differ in the specific purposes they serve. For example, some systems focus on general frameworks such as assisted-living environments, while others focus on the creation and integration of new sensors or new applications that use clothing embedded with sensors. All systems are built to empower individuals by providing them with the ability to monitor themselves, while at the same time availing themselves of a watchful "guardian angel" that monitors and advises from a distance.

Secure Mobile Computing

The Secure Mobile Computing project (www.cs.virginia.edu/~acw/SecureMobileComputing) at the University of Virginia is representative of remote medical monitoring systems in terms of both functionality and scope. An individual wears a small biometric patch with the form-factor of an adhesive bandage; as Figure 2 shows, the "patch" consists of a biometric sensor, a microcontroller, and

Figure 2. Prototype for the Secure Medical Computing project. This device contains a biometric sensor (ECG), a microcontroller, and a radio (Bluetooth). By summer 2008, these components will be encapsulated into a custom integrated circuit. In the future, subthreshold logic design will permit the chip to be powered by energy harvesting from the human body.

Figure 3. PDA receives the ECG signal over the Bluetooth channel and plots the resulting waveform.

a radio. The patch's initial biosensor detects an electrocardiogram (ECG) signal from which heart rate information is extracted. A chest strap is envisioned to host an array of additional sensors.

Due to the wireless nature of the patch, energy is a primary concern. Initially, the patch is powered by a battery, but in the future researchers will exploit subthreshold digital circuit design to make it ultra-low-power. Energy harvesting from the body's natural motions or from the skin/air temperature gradient, coupled with very-low-power circuitry, could power the patch indefinitely.

The burgeoning handheld device industry is heading toward further integration of disparate components; therefore, the patient's ability to monitor personal data on such a device is an important design aspect. The ECG sensor interfaces with the handheld device via the Bluetooth wireless standard for cable replacement. As Figure 3 shows, once a connection has been established, the mobile device receives the current heart rate data, analyzes it, and optionally plots a real-time ECG. The device can log the data and transmit it over the Internet via Web services so that an authorized user can view the real-time data anytime, anywhere.

The Secure Mobile Computing system subscribes to a service-oriented architecture to maximize interoperability with other systems. The sensor data is encrypted and sent to a server from which the data can be viewed. The entire architecture from tier one to tier three is modular in that new sensors, wireless protocols, and data processing algorithms can be readily incorporated. In the future, when a patient feels ill and has this system at his disposal, he can put on the patch, place a call to his healthcare provider, and transmit relevant health information (either recorded or real-time) to provide a more holistic picture of his current state.

CodeBlue

A self-described "information plane" in which a diverse set of sensors can discover and communicate with one another in an ad hoc fashion, CodeBlue (www.eecs.harvard.edu/~mdw/proj/codeblue/) is a wireless sensor network developed at Harvard University and intended to assist the triage process for monitoring victims in emergency and disaster scenarios.

The software framework offers service discovery protocols, publish/subscribe multihop routing, and a query interface for caregivers to request data. CodeBlue also provides for traffic prioritization, robust routing, authentication and encryption, and in-network filtering and data aggregation. Data obtained from sensors can be used for "decision support" to guide trauma care and realize a holistic view of the scene.

AMON

Sponsored by the European Union Information Society Technologies, AMON (alert portable telemedical monitor; ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1362650), encapsulates many sensors (blood pressure, pulse oximetry, ECG, accelerometer, and skin temperature) into one wrist-worn device that is connected directly to a telemedicine center via a GSM network, allow-

ing direct contact with the patient if necessary.

The unit promises to be an important early prototypical remote medical monitoring system, but the results of testing in a medical study called for more research because most sensor outputs couldn't be used in a clinical setting—especially the ECG, which couldn't be detected reliably at the wrist.

IBM Personal Care Connect

IBM's PCC (www.zurich.ibm.com/pcc/) is a standards-based platform for interfacing to biomedical devices and sensors while collecting, storing, and making available the data received from them. PCC is meant to be open and extensible so that new technologies can leverage the PCC architecture.

A device manager maintains information on hubs, devices, and their relationship to patients. A kit wraps the notion of one patient, one hub, and a customized set of biomedical sensors relevant to that particular patient.

Smart Medical Home

The University of Rochester hosts the Smart Medical Home (www.rochester.edu/pr/Review/V64N3/feature2.html), a controlled environment for medical-monitoring research. The overarching goal is to provide seamless integration of all monitoring technologies.

The house has five rooms with computers, infrared sensors, biomedical sensors, and video cameras. The Smart Medical Home takes the first steps toward an automated at-home doctor by developing a virtual "personal medical advisor" that interacts with individuals in the comfort of their own homes to discuss medical issues and give advice.

AlarmNet

A prototype wireless medical sensor network developed at the University of Virginia, AlarmNet (www.cs.virginia.edu/wsn/medical) continuously monitors assisted-living and independent-living residents. Tailored to the stable operating environment of a home, the system integrates information from sensors in the living areas as well as body sensors. Context-aware protocols informed by an individual's patterns of activity enable customized power management and alert policies. AlarmNet also features a query protocol for streaming online sensor data to user interfaces, integrated with privacy, security, and power management.

SYSTEM ISSUES

A remote medical monitoring system's reliability depends on avoiding faults so that it can provide continuous, correct service. Equally important are various aspects of system security, especially privacy and data integrity. Any successful monitoring system must have robust procedures in place to guarantee whose data is being recorded and who will have access to that data.

Authentication (the process of verifying a person's identity) is essential. Sensors must "know" who they are sensing so that the information collected is attributed to the correct person. For a wireless sensor network positioned on the body, it is difficult for remote computers to tell if the sensors are on the right person without explicit human guidance. In a smart home, the problem becomes more significant because the process of associating information with a particular individual is nontrivial, particularly when using passive sensors—for example, motion detection—and there is more than one person in the home.

Remote medical monitoring is currently in its infancy, but its future is bright. Patients with chronic diseases will be outfitted with appropriate sensors from which data will be transported to hubs for local processing; the data ultimately will be forwarded to secure medical networks for visualization and analysis by physicians, aided by software agents that continuously monitor the data stream. Self-sufficient patients will use this technology to stay independent longer, and patients in assisted-living environments will benefit from continuous monitoring and a faster, better-informed medical response to adverse events. The push to take healthcare home in an inconspicuous and minimally invasive fashion equips individuals to be attuned to their health, encouraging a healthier and more well-informed society.

Andrew D. Jurik *is a graduate student in the Department of Computer Science at the University of Virginia who is working on the Secure Mobile Computing project. Contact him at adj3t@virginia.edu.*

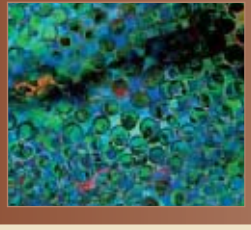
Alfred C. Weaver *is a professor of computer science at the University of Virginia. He is working with faculty research partners Ben Calhoun and Travis Blalock in UVa's ECE department to build a low-power, modular system to acquire, analyze, display, and publish biotelemetric data. Contact him at weaver@virginia.edu.*

Computer welcomes your submissions to this bimonthly column. For additional information, or to suggest topics that you would like to see explained, contact column editor Alf Weaver at weaver@cs.virginia.edu.

ENTERTAINMENT COMPUTING

Massive Media Shift

Michael van Lent, Soar Technology



The days of physical entertainment media may well be numbered.

Occasioned by a new job opportunity, I recently completed the challenging exercise of packing up the family and moving 2,000 miles to a new house in a new town. This required simultaneously transferring all my material stuff to a new house and transferring all my digital stuff to a new laptop. The parallels and differences between these two activities got me thinking about the library of music, video, pictures, and books that, to differing degrees, exist in both physical and digital form.

Why must I still lug around a big box of CDs when I also have those songs in electronic form? I paid to move boxes and boxes of books, at $25 per 100 pounds, but have very few books in electronic form. The digital video recorder (DVR) had several TV programs recorded on it that I never found time to watch before the move forced me to return it to the cable company.

Clearly I, like many of you, am in the midst of a transition from a physical entertainment media library to a digital one. Trust that recent developments in the continuing migration from physical to electronic entertainment media will have far-reaching implications.

THE GREAT MIGRATION

The first challenge, migrating from physical to electronic media, will demand that we learn how to transfer an existing library of media in physical form to an electronic format.

Music and pictures are two forms of entertainment media that have made the most significant progress away from physical media. With the advent of digital music players and online music distribution providers, building an electronic music library becomes ever easier. A variety of free and easy to use applications can "rip" songs from CDs into electronic form, although the legality of ripping songs from physical media has recently been called into question.

SHARPER IMAGES

Scanning physical photographs and transferring video, especially VHS and other videotape formats, is fairly straightforward but requires hardware that varies greatly in quality and expense. Several companies have sprouted that will perform the conversion for a reasonable fee, using high-quality hardware.

In the case of photographs, the larger market seems to appeal to those converting in the other direction, from digital photographs to physical ones. With the rapidly increasing market penetration of digital cameras and digital music players, photographs and music represent the first forms of entertainment media that people add directly to their library in electronic form, without a physical counterpart.

As the multiple services available for printing physical photographs from digital photographs demonstrate, people want to have at least some of these digital images in physical form. This trend highlights two aspects of the physical-to-digital migration.

First, for some forms of entertainment media, the physical form holds special status. This could be due to sentimental value (my wife had old family pictures scanned, but wouldn't dream of discarding the physical copies) or because it fills some niche more effectively. Digital picture frames are available, but it costs less to display a physical print in a traditional frame.

Second, people seem to be comfortable acquiring media in one form even if the primary use of that media will be in a different form. This works both ways. People take digital pictures to display in physical form and purchase physical CDs to rip and listen to in electronic form.

At least one form of physical-to-digital conversion trails behind this trend: consumer-level scanning of books. Several large-scale efforts to convert entire libraries into electronic form have been undertaken, including Project Gutenberg's focus on Western literature and Google Book Search's focus on leading universities' libraries around the world.

Unlike traditional scanners, book scanners generally use high-megapixel digital cameras

to capture the pages, as Figure 1 shows. However, book scanners are currently much too expensive for everyday consumers. The first book scanner able to automatically turn pages by itself was released in 2006 at a cost of approximately $35,000 (http://en.wikipedia.org/wiki/Book_scanning#_note-1).

Book scanners might be so expensive because of low consumer demand. Like pictures, books are a media type for which the physical form holds special status. I only own a handful of books in electronic form, and most of those are audiobooks purchased for long car rides. In general, I much prefer reading a book in physical form to reading an electronic book on a computer display.

Manufacturers have made several e-book readers available, all of which, to a greater or lesser degree, attempt to recreate the traditional form factor of a physical book. Most e-book readers use a display called e-paper, which is a form of electrophoretic display designed to look as much as possible like ink printed on a paper page (http://en.wikipedia.org/wiki/Electrophoretic_display).

Amazon's new e-book reader, Kindle, turns pages via a long switch that takes up most of the device's right edge, mimicking a person's grasping of a page's right edge to turn it to the left. Even with these efforts to make e-book readers and physical books alike, consumers have been slow to move away from good old paperbacks and hardcovers.

Once a substantial library of electronic media has been built up, either via migration from physical form or acquired directly in electronic form, that library continues to grow independently of the specific devices that hold the stored library. I've listened to many songs in my library on a long string of laptops, MP3 players, and now an iPod. However, some forms of digital rights management—although the Free Software Foundation suggests the term should be *Digital Restrictions Management*—make it difficult to transfer songs to new devices. Songs purchased from Napster are only compatible with players that support Microsoft's PlaysForSure, which doesn't include either iPods or Microsoft's own Zune device.

Digital video presents a special challenge because the size of these files is significantly larger than the size of digital images, songs, or e-books. One hour of high-definition digital video occupies approximately 35 to 50 gigabytes of disk space, depending on format. While current hard drives, which typically hold 500 gigabytes or less, can store huge photograph and music libraries, this same capacity can accommodate only a few hours of HD video. These recorded programs are kept until watched, then deleted.

Apple recently started to offer online video rental through iTunes, but, as a rental service, the large video files will only be kept temporarily. While extensive libraries of electronic music and photographs are becoming the norm, libraries of digital video must wait until hard drives get a lot bigger.

What does the future hold for digital music, photographs, e-books, or those large digital videos? Amazon's Kindle points to one possibility with its EVDO-based wireless network capability. With this feature, consumers can buy and download new e-books anytime, anywhere.

As wide-area wireless network connections become pervasive, devices will no longer be limited to the content loaded during the last synch with a computer. Instead, the library will be streamed to the device on demand from a central server.

***Michael van Lent**, the Entertainment Computing column editor, is the Chief Scientist of Soar Technology. Contact him at vanlent@soartech.com.*

Figure 1. Going electronic. Scanners that use high-megapixel digital cameras to capture book pages are still too expensive for consumer use (Atiz—http://booksnap.atiz.com/gallery), but Amazon's Kindle (inset) provides a more portable alternative.

INVISIBLE COMPUTING

Activity Recognition for the Digital Home

Jeonghwa Yang, Georgia Tech
Bill N. Schilit, Google Research
David W. McDonald, University of Washington

Patterns mined from home networks can support smarter applications.

Home networked devices enable a wide range of daily activities including multiplayer gaming, movie downloading, and music streaming as well as modern conveniences such as home automation, wireless networking, and Internet access. Nevertheless, for most of us the futuristic digital home we see in movies isn't a reality.

We spend lots of time figuring out how to get our devices to do what we want and to keep them properly working and tuned. Ideally, these devices should support as well as enable the activities we like to do. For example, a user should be able to request "my favorite radio station" rather than have to input something like "192.168.1.100."

One step toward realizing a smarter digital home is efficiently modeling and recognizing human activities. We've developed a prototype system to detect various digital media and information access activities using the consumer electronic devices people already have. Cameras or other hardware sensors aren't necessary; instead, the network itself is the sensor, and the system monitors associated data flows among the devices and uses a template to match the data against generic classes of activities.

HOME ACTIVITY RECOGNITION

Activity recognition is a key feature of many ubiquitous computing applications ranging from just-in-time information for office workers to home healthcare. In general, activity recognition systems unobtrusively observe the behavior of people and characteristics of their environments and, when necessary, take actions in response—ideally with little explicit user direction.

In the home environment, such systems can, for example, remind users to perform missed activities or complete actions (like taking medicine), help them recall information, or encourage them to act more safely.

Our digital home research has focused on recognizing common activities, including

- browsing the Web,
- reading an online newspaper,
- watching movies and Internet TV/videos,
- listening to music and Internet radio, and
- playing networked console games.

These activities might not be as important as monitoring medication usage, but they represent areas where people struggle with technology.

One application of home activity recognition is to observe users' interests based on their interaction with digital media and provide reasonable suggestions for future activity. For example, an application could inform you that a presidential debate is on TV based on your previous viewing of such debates.

Patterns mined from home activities can be used to support a wide range of similar over-the-shoulder applications. For example, such information could help define defaults such as which rooms to play music in or what stations to "preprogram" on Internet radio; it could likewise help users refind memorable videos on their media players, much as the history feature on Web browsers makes it easier to refind interesting websites.

Home activity recognition and data traffic monitoring can also help people understand complex network behavior—say, why they can't get an Xbox Live connection or why Internet access is slow (perhaps someone is watching streaming video in the living room). Information about household activities can even be used to recommend changes in behavior—for example, to reduce TV viewing and spend more time playing aerobic games on the Wii.

These examples only hint at what's possible. Several researchers are exploring other aspects of this problem space. For example, Monika Henziger and colleagues describe how digital TV closed captioning could be used to generate Web queries ("Query-Free News

Search," *World Wide Web,* vol. 8, no. 2, 2005, pp. 101-126).

DIGITAL HOME NETWORKS

During the past several years, the computer and consumer electronics industries have introduced numerous home-networked communication and entertainment devices. Network-attached disks store and stream audio and video to PCs and network stereos, phones and game consoles connect through home routers to the Internet, and networked printers and picture frames print and display digital photos from laptops and Wi-Fi cameras.

All these devices support networking capability using wired (IEEE 802.3) or wireless (IEEE 802.11 a/b/g) technologies and IPv4. Many also use HTTP or the Real-Time Transfer Protocol for media transport and higher-level protocols such as universal plug and play for device discovery and UPnP AV (audio and video) for media control and management.

An important feature of UPnP is that it provides a standard way of describing devices on the network. Each UPnP device publishes an XML description that includes the name, manufacturer, model, and serial number as well as a list of embedded services—such as "rendering" image or audio files—available through URLs. Because each of the service classes is standardized, it's possible to determine which devices can work together for various uses even if they have different manufacturers.

HOME NETWORK MONITORING

In our research, we use the UPnP discovery protocol to detect UPnP-enabled devices on a home network and then create a UPnP *instance* table that contains a description of each device and its capabilities along with its physical (MAC) address.

We augment this static description of the network with a UPnP *operational status* table that lists which devices are sending packets to each other, the devices' streaming status (playing, paused, stopped), active media file names (like "lovesong.mp3"), and other information. The table contents change as devices connect to each other and play music, movies, and so on. To populate this table, we use both a low-level traffic flow monitor and a high-level UPnP event monitor.

A few home routers now can report data traffic using RFlow, a protocol based on Cisco Systems' NetFlow product for enterprises. RFlow records the MAC addresses for source and destination devices, along with a packet byte count, for a given period of time. Our experimental home network's RFlow-capable router sends samples to an SQL database server at uniform time intervals, indicating which devices have active data flows—for example, music streaming from a PC to a network stereo.

Home activity recognition and data traffic monitoring can help people understand complex network behavior.

Although RFlow provides information about data traffic between devices, it doesn't say much about that data's content. To obtain eventing information, we exploited UPnP event notification technology, which uses a publish-subscribe model to send status and control-variable changes between a media server or renderer to registered parties—this is useful for a remote-control point, for example.

We used Tools for UPnP Technologies, developed by Intel's Digital Home Group, as well as our own event subscriber application to capture media events for the operational status table. These events include the media content's format and uniform resource identifier, which can be used to obtain further information such as ID3 tags.

ACTIVITY TEMPLATES

The UPnP instance and operational status tables provide a medium-level description of home network transactions and capabilities. To make this information meaningful to people, we created high-level activity descriptions such as "read [online] newspaper," "watch Internet TV," "listen to Internet radio," "listen to music," "watch movie," "look at photo," and "play Xbox game." We define activities by media type, but other criteria such as device type could also be used.

For each activity, we developed an *activity template* that lists attribute names and values including the source device, destination device, and media of data transferred along with numerous other parameters that characterize the activity duration and type.

We manually created our activity categories and templates, but it's possible to automatically generate activity tuples—for example, using potential device pairings based on client-server protocol information. However, user-meaningful activity labels would still need to be added manually. A privacy filter could also obscure some or all details of activities.

RECOGNITION ENGINE

We also developed a *recognition engine* that infers activities based on network traffic data and UPnP device information matched with generic classes of activities defined in the activity templates. When the engine observes a new traffic flow, it analyzes the source and destination devices involved and the time stamp, then refers to the UPnP instance and status tables to determine which devices are a media server and renderer.

The recognition engine also examines application-level events involving those two devices around the time of the network flow and matches this data against the activity templates. It records matching templates at each sampling interval and combines templates from one interval to the next

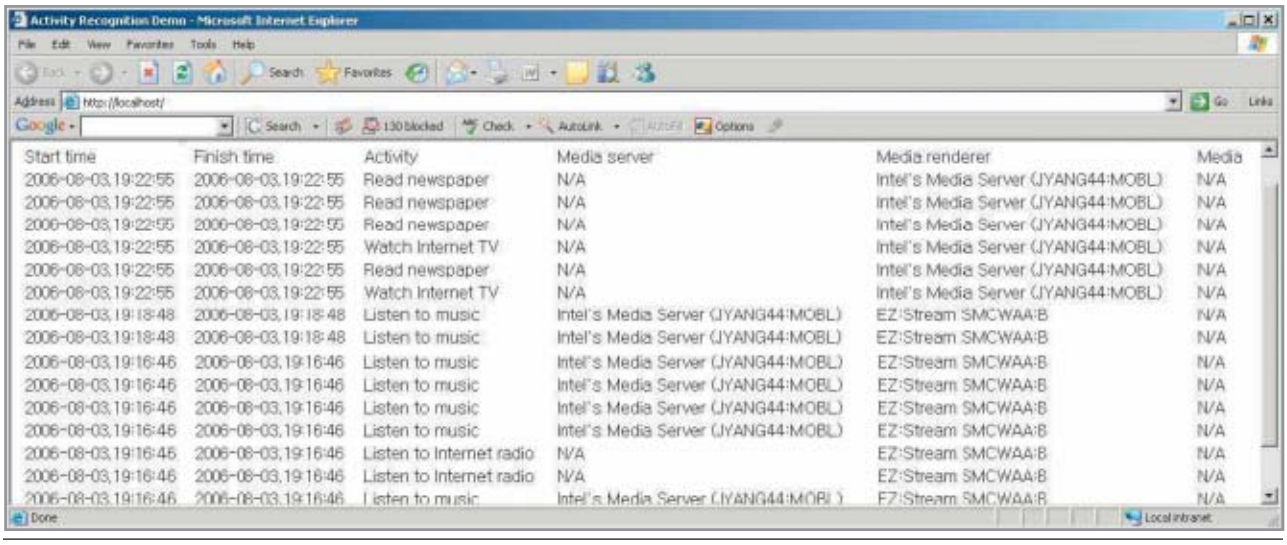


Figure 1. Web-based network activity visualizer. An activity history log combines multiple short, similar activities into a single human-understandable session.

so long as the source and destination devices are identical.

When a template no longer matches the network monitoring data, the engine maintains it for a "maximum pause time" before writing it to an activity history log. We do this to combine multiple short, similar activities into a single human-understandable session. Similarly, a template must match for a "minimum duration" to be written to the log.

The activity history log can take various forms—for example, it can be displayed on a computer or TV much like a program schedule. For our work, we developed the real-time Web-based activity visualizer shown in Figure 1 using JavaScript and PHP.

Modeling and detecting human activities using home network data traffic patterns is a step toward creating a smarter digital home that provides useful information and services in a more user-friendly way.

Primarily using UPnP technology, we created a network map of devices and their characteristics and recorded dynamic data flows along with high-level events. Based on these static and operational values, we manually defined and automatically matched human-level activity templates. We believe that a learn-by-demonstration method or simply sharing activity templates among a large user community could improve the manual steps of our process.

Jeonghwa Yang *is a PhD student in the School of Computer Science at the College of Computing, Georgia Institute of Technology. Contact her at jeonghwa@cc.gatech.edu.*

Bill N. Schilit *is at Google Research. Contact him at schilit@computer.org.*

David W. McDonald *is an assistant professor in the Information School at the University of Washington. Contact him at dwmc@u.washington.edu.*

Part of the work described in this article was done while the authors were with Intel Research.

Continued from page 108

system should have given an error message in this case, instead of ignoring all typed digits after the first 11. She has acknowledged she would have no case if only 11 digits had been typed. The bank argues that she cannot prove by any measure of probability that she keyed 12 digits. They further state that there cannot be different rules of responsibility depending on the number of digits given. Finally, they stress that she confirmed the $100,000 transaction.

At this point, I was called in as an expert witness for Fossbakk. In my opinion, and I should expect that of most other computing professionals, a system should give an error message when the customer types a too-long number. Clearly, such a test can be inserted with a minimum of effort. In fact, the Financial Supervisory Authority of Norway has required all banks to implement this functionality based on the Fossbakk case.

Reasonably, we could argue that the bank showed negligence when developing the user interface in question. However, can we prove, beyond doubt, that Fossbakk keyed 12 digits? Since any digits beyond 11 were stripped from the HTML form, no information log exists that can tell us what happened.

BANK SIMULATOR

The answer to this case cannot be found in the literature. It seems that researchers lost interest in studying keying errors when keypunches disappeared. We therefore decided to get our own data by implementing an "Internet bank simulator," a simple interface that works similarly to the system Fossbakk used.

Our system consists of two forms. In the first form we entered the data, date, customer identification number, message text, amount, and account number. After hitting the "pay" button on this form, the data appeared in a new form for confirmation, allowing the user to either confirm or edit the displayed information. Students from a college and some high schools, 69 testers altogether, engaged in entered 30 transactions each from a predetermined test set. After removing some outliers, this gave data on 1,778 transactions.

Results

Our student testers got 124 account numbers wrong, 7 percent of the transactions. This error rate is higher than we would expect in a real system. First, since analyzing faults is our initial task, the simulator does not offer any error messages. However, the user must confirm the transaction, just as in the real system, and the count will not include any errors corrected before confirmation.

It seems nonchalant to code software that lacks a detect-too-long number.

Second, the testers enter a large set of transactions. In some cases, the account number for the preceding or following transaction has been used instead. This could happen in real life, but will be much more frequent here since testers enter the transactions from a list. Third, the test situation does not involve any real money. We suspect that users would verify transactions more carefully when using a live system.

While the overall error rate might be higher, there seems to be no reason why the distribution of different error types should be any different from what we would find in a real system—an exception being the case in which an account number is replaced by another from the data set.

In 29 percent of the cases with a wrong account number, the number ran too long. In half the cases where this happened, students made the same error as Fossbakk, inserting an extra digit in a sequence of two or more identical digits. The bank interface's strategy of skipping digits beyond 11 would have given the correct number in 64 percent of the cases. Of the remaining abbreviated numbers, the modulo 11 test captured all but three. That is, of the nearly 1,800 transactions, three (0.2 percent) would have passed the banking interface's error-detection routines. Multiply this by the nearly 200 million Internet transactions performed each year in Norway, and we see that this small percentage hides a massive problem.

In an improved interface, with a "too long" check, along with the modulo 11 routine, all errors made in our test would have been captured—except in cases where someone entered an account number from another transaction in the set.

Analysis of customer identification numbers, also a part of the transaction, showed the same result. Adding an extra digit in a sequence is a normal mistake. Missing a digit in a sequence is also easy. This test found these errors most commonly. If we ignore the error of typing another account number from the set, "too long" errors occur in 41 percent of the cases, "too short" errors in 35 percent, and wrong 11-digit numbers in 24 percent.

Given these statistics, it seems nonchalant to code software that lacks a detect-too-long number. Since none of the people who entered an extra digit or missed one managed to finish with an 11-digit number by making yet another error, we can state with high probability that Fossbakk entered a 12-digit account number.

Confirmation

This leaves us with the argument that she confirmed a $100,000 transfer to the wrong account—as did the students in 124 of our test cases. In addition, for every tenth transaction, the simulator replaced the typed number with a similar-looking number before confirmation. For example, it replaced the number 70581555022 with 70581555502. In only five out of the 178 cases, 2.7 percent, where this was done did the users recognize the error and correct the number.

It appears that most people perform the inspection while keying,

not when the whole number is displayed onscreen. In many ways, this is efficient. While keying, we can concentrate on one digit at a time, and after keying we have a large number. If this *seems* correct, we hit the "confirm" button.

Psychologist Donald A. Norman explains this behavior in his book, *Psychology of Everyday Things* (Basic Books, 1988). Here, a user confirmed deletion of his "most important work." According to Norman, the user confirms the action, not the file name. Thus, the "confirm" part of the transaction, while having some legal implications, has a minimal effect on detecting errors.

ACCOUNTABILITY

Like many new IT applications, Internet banking is effective. As users, we enjoy reduced costs and 24/7 availability. However, transferring real money based on instructions from possibly inexperienced humans who might slip up means we must look to the system for help. Developers must require that it intercept as many errors as possible. If Fossbakk had used the manual system instead—by writing a letter to her bank requesting the transaction—no responsible employee would have removed the 12th digit of the account number in hopes this would correct the error.

We should expect more. The banking system could offer the account owner's name as confirmation when an account number is entered. In cases where this conflicts with privacy issues, a first name or alias could be used. The system could give a warning message whenever a previous pattern is violated. For example, if we pay a utility bill of $100 to $300 each month, we should get a warning if the reported amount is way off in either direction. Further, e-invoices and other automatic procedures can limit the number of transactions that must be keyed in, reducing the overall error rate.

In Fossbakk's case, the banking system erred. Next time, it might be a weapons system or medical information system that fails. Examples from these areas have already revealed misinterpretations between systems and users that caused serious consequences. For all systems, we must, as computer professionals, protect users from their own errors, intercept all detectable errors, and give informative warnings when we believe the user might have made an error. The "she made an error and must take responsibility" defense is too simple. We need systems that work in collaboration with the user such that the overall error rate drops to a minimum. Yes, we need responsible users, but a good system can handle most slips and typos they make, as this case shows.

Kai A. Olsen is a professor at Molde College and the University of Bergen, and an adjunct professor at the University of Pittsburgh. Contact him at Kai.Olsen@hiMolde.no.

Editor: Neville Holmes, School of Computing and Information Systems, University of Tasmania; neville.holmes@utas.edu.au.

THE PROFESSION

The $100,000 Keying Error

Kai A. Olsen, Molde University College and University of Bergen

Losing $100K hurts, but other input mistakes can cost much more.

Although it sounds disastrous, making a $100,000 mistake seems relatively minor when stockbrokers have lost millions by hitting the wrong key. The following case proved to be different, however.

An ordinary bank customer, Grete Fossbakk, used Internet banking to transfer a large amount to her daughter. She keyed one digit too many into the account number field, however, inadvertently sending the money to an unknown person. This individual managed to gamble away much of the sum before police confiscated the remainder.

Subsequently, the case received extensive media coverage in Norway. The Minister of Finance criticized the bank's user interface and requested new and improved Internet banking regulations. Suddenly, the risk to Internet banking had become apparent to both the government and ordinary citizens.

Clearly, the user made a slip. She also had the chance to correct the typo before she hit the confirm button. However, as we shall see, the system also had every opportunity to catch her mistake. Yet this did not happen. The system's developers had neglected to build in a simple check that would detect if the correct input were missing.

This case raises questions about what the minimum validation procedures from a banking system developed for ordinary users should be. It also challenges us, as system designers, to help users avoid such errors.

Today's users operate alone in front of a computer, with intermediates and colleagues replaced by computer systems. This new reality makes it important to have interfaces that can offer as good as—or even better—error detection than that found in previous manual systems.

FOSSBAKK CASE

The Fossbakk case provides an illuminating example. The Internet system she employed when making her fatal mistake was one common to a large group of Norwegian banks. Reviewing this case provides insight into the types of typos that users make, the psychology behind "confirmation," and the pitfalls inherent in many Web-based transactions systems.

Fossbakk's daughter's account number was 71581555022, but she inserted an extra 5 and keyed in 715815555022. The user interface accepted only 11 digits in this field (the standard length of a Norwegian account number), thus truncating the number to 71581555502. The last digit is a checksum based on a modulo-11 formula. This will detect all single keying errors and errors where two consecutive digits are interchanged. Inserting an extra 5 changed both the ninth and tenth digits.

The average checksum control will catch only 93 percent of the cases in which such errors occur. For Fossbakk, the final eleven-digit number was a legal account number. However, only a small fraction of all legal account numbers are in use. Further, the chance of mistyping the account number so that it benefits a dishonest person without income or assets is overwhelmingly low in a homogeneous country such as Norway. Our user was thus extremely unlucky. The person who received her $100,000 transaction and kept the proceeds has been sentenced to prison, but this does little to help Fossbakk get her money back.

LITIGATION

Fossbakk took the case to the Norwegian Complaints Board for Consumers in Banking. This board deals with disputes between consumers and banks. The board has two representatives for the consumers and two from the banks, with a law professor as chair. In a three-to-two vote, Fossbakk lost. The chair voted for the bank, arguing that "she made an error and has to take responsibility." He also regretted that Norwegian regulations set no limit for a consumer's loss in these cases, as there would have been if Fossbakk had lost her debit card.

Fossbakk is now taking the case to court, backed by the Norwegian Consumer Council. She argues that she typed 12 digits and that the bank

Continued on page 106

IEEE COMPUTER SOCIETY
EDUCATIONAL AWARDS

Nominations are solicited for the

Computer Science & Engineering Undergraduate Teaching Award

and the

Taylor L. Booth Education Award

Computer Science & Engineering Undergraduate Teaching Award

A plaque, certificate, and $2,000 honorarium are awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and make a statement about the importance with which we view undergraduate education.

Taylor L. Booth Education Award

A bronze medal and $5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown
- Writing an influential text
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field
- Inspiring others to a career in computer science and engineering education

IEEE

IEEE computer society

Nomination form and submission:
http://awards.computer.org/ana

Deadline for both awards:
1 July 2008