Innovative Technology for Computer Professionals

Computer

April 2005

Points of Highest Leverage, p. 12

Game Engine Virtual Reality, p. 79

Social-Mobile Applications, p. 84

IEEE

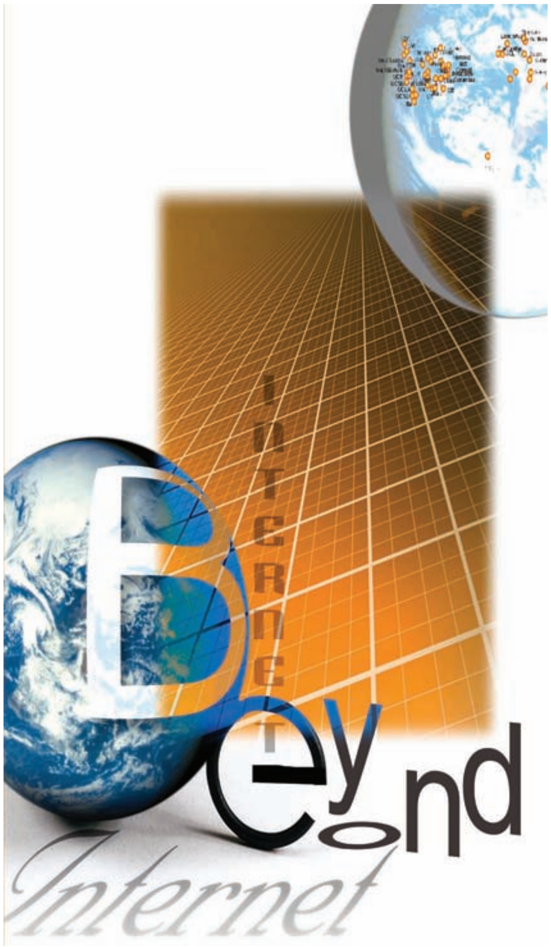
IEEE COMPUTER SOCIETY

http://www.computer.org

Innovative Technology for Computer Professionals

Computer

April 2005, Volume 38, Number 4



Cover design and artwork by Dirk Hagner

ABOUT THIS ISSUE

As we look to ever more advanced applications for the worldwide system of interconnected computer networks that has increasingly become an integral part of our critical infrastructure, what technology breakthroughs will be needed to support the next-generation Internet? In this issue, we look at PlanetLab, a collaborative effort that fosters applied research by offering easy access to virtual testbeds that support multiple simultaneous architectures. We also look at the competing frameworks and specifications proposed for creating the services and associated applications that make it possible for individual projects to interact with one another in the larger Grid environment.

PERSPECTIVES

COVER FEATURES

RESEARCH FEATURES

Computer

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

OPINION

NEWS

MEMBERSHIP NEWS

NEXT MONTH:
Virtualization

COLUMNS

DEPARTMENTS

Membership Magazine of the IEEE Computer Society

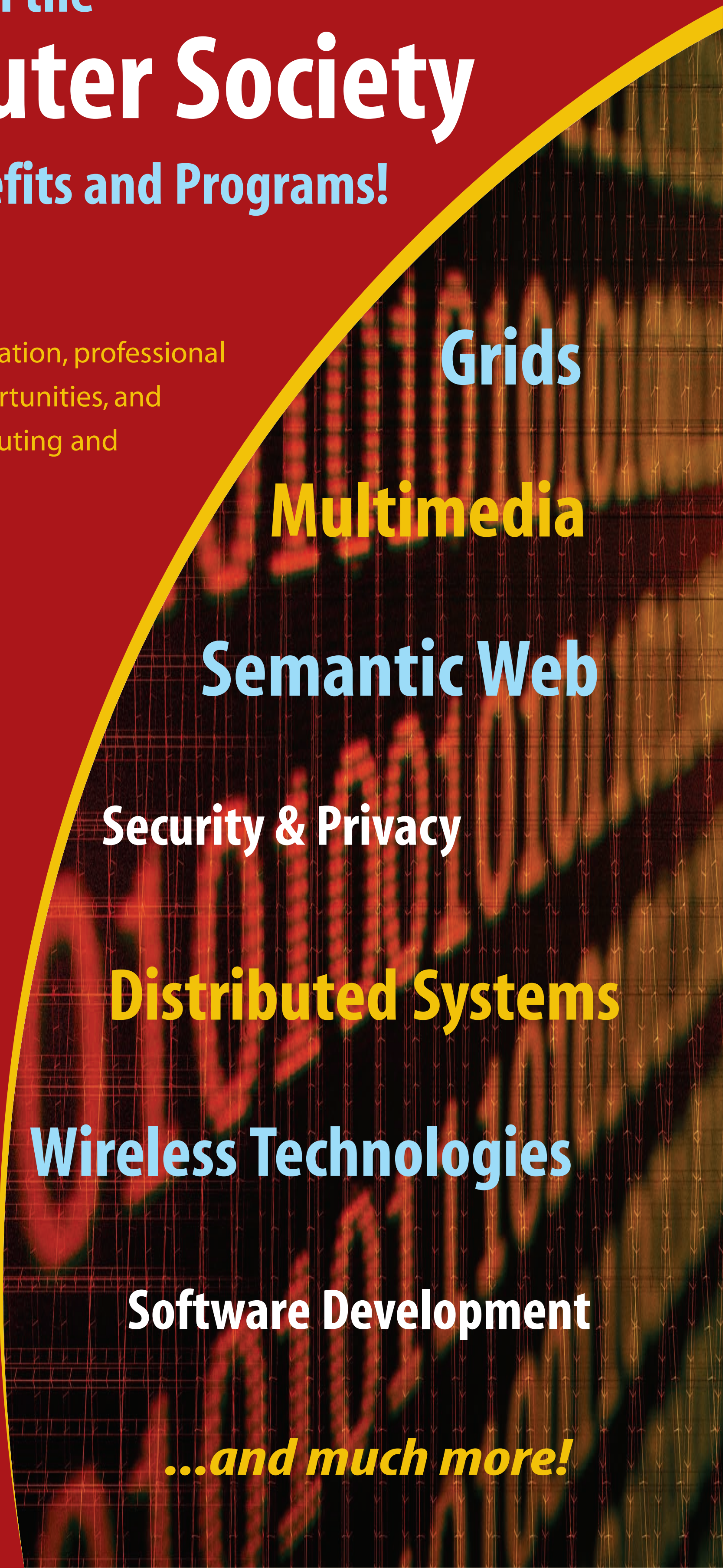
Innovative Technology for Computer Professionals

Computer

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 $94.00.

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.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.


Grids
Multimedia
Semantic Web
Security & Privacy
Distributed Systems
Wireless Technologies
Software Development
...and much more!

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

APPLICATION MUST BE SIGNED

NOTE: In order for us to process your application, you must complete and return both sides of this form.

IEEE

IEEE COMPUTER SOCIETY

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, 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 ______________

The Future Interconnection Environment

pp. 27-33

Hai Zhuge

Nearly half a century after Marvin Minsky predicted that computers would be as smart as humans, computing systems still cannot pass the Turing test. Despite impressive achievements in robotics, mathematical theorem proving, scientific classification, and advanced user interfaces, artificial intelligence remains elusive.

Scientists and engineers have nearly realized Vannevar Bush's dream of a universal multimedia data-processing machine with the Internet and the World Wide Web. It is now possible to foresee the development of highly secure, highly available, self-programming, self-managing, and self-replicating computer networks. However, creating intelligent networks that can program, manage, and replicate themselves remains a major challenge.

The China Knowledge Grid Research Group, established in 2001, is exploring the operating principles of this future interconnection environment.

Overcoming the Internet Impasse through Virtualization

pp. 34-41

Thomas Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner

The prospects for significant change in the Internet's existing architecture appear slim. In addition to requiring changes in routers and host software, the Internet's multiprovider nature requires that ISPs jointly agree on any architectural change.

The canonical story about architectural research's potential impact has long maintained that if testbed experiments show an architecture to be promising, ISPs and router vendors might adopt it. This story might have been realistic in the Internet's early days, but not now: Not only is reaching agreement among the many providers difficult to achieve, attempting to do so also removes any competitive advantage from architectural innovation.

By providing easy access to virtual testbeds, the authors hope to foster a renaissance in applied architectural research that extends beyond incrementally deployable designs. Moreover, by replacing a discredited deployment story with a plausible one closely linked to the experimental methodology, they hope to raise the research community's sights.

Emerging Grid Standards

pp. 43-50

Mark Baker, Amy Apon, Clayton Ferner, and Jeff Brown

The Grid has evolved from a carefully configured infrastructure that supported limited Grand Challenge applications to a seamless and dynamic virtual environment being driven by international development and take-up. Commercial participation has accelerated development of software that supports Grid environments outside academic laboratories. This in turn has impacted both the Grid's architecture and the associated protocols and standards.

The recent adoption of Web services has produced a somewhat fragmented landscape for application developers. Developers currently face the dilemma of deciding which of the many frameworks and specifications to follow.

The Open Grid Services Architecture and the Web Services Resource Platform represent significant cooperation among researchers in academia, government, and industry. These joint efforts point to a promising future for the Grid regardless of the problems developers currently face.

Scaling Network Services Using Programmable Network Devices

pp. 52-60

Christoph L. Schuba, Jason Goldschmidt, Michael F. Speer, and Mohamed Hefeeda

Over the past several years, one successful solution for managing huge amounts of data on the Internet concentrates critical computing resources in Internet data centers. An IDC collects computing resources and typically houses them in one physical location: a room, a building floor, or an entire building. Large enterprises that rely heavily on the Internet and e-commerce applications typically operate their own IDCs, while smaller companies may lease computing resources within an IDC owned and operated by a service provider.

The NEon architecture, a novel approach for implementing the network services that IDCs provide, is a paradigm shift away from special-purpose network devices. By employing new flow-handling mechanisms to merge heterogeneous network services into one system, NEon offers an integrated approach to architecting, operating, and managing network services.

Leveraging Social Networks to Fight Spam

pp. 61-68

P. Oscar Boykin and Vwani P. Roychowdhury

The amount of unsolicited commercial e-mail—spam—has increased dramatically in the past few years. A recent study showed that 52 percent of e-mail users say spam has made them less trusting of e-mail, and 25 percent say that the volume of spam has reduced their e-mail use.

This crisis has prompted proposals for a broad spectrum of potential solutions. The objective of the various proposed legal and technical solutions is the same: to make sending spam unprofitable and thereby destroy the spammers' underlying business model.

Achieving these goals requires widespread deployment and use of antispam techniques. To gain user confidence, a prerequisite for wide deployment, the tool must be accurate, user friendly, and computationally efficient. The authors describe a technique, predicated on recognizing the unique characteristics inherent to social networks, that simultaneously achieves all these requirements.

GLOBAL WARMING

I always enjoy reading The Profession column, but the February 2005 essay (N. Holmes, “The Profession and the Big Picture,” pp. 104, 102-103) really stands out.

Having written about professional responsibilities in my small corner of the engineering world, I sometimes receive unkind comments in return. That comes with the territory, so I accept it. But I’ve never attempted to go above the radar with comments that remind engineers of their social accountabilites in a journal like *Computer*, with its circulation and potential audience.

If you receive any unkind responses to this essay, please know that Holmes has been successful in illuminating something that struck a nerve—that is, something that is very important—and also that there are many of us who agree with him.

Rick Schrenker
Boston, Mass.
raschrenker@partners.org

I enjoy The Profession column, which is usually the first thing I read in *Computer*. However, the nature of being engaged enough to send an e-mail message ensures that it is most likely to be opposed to the author’s comments. My message about the February 2005 essay is also, but perhaps not in the expected way.

I will not try to dispute whether global warming is happening or that humans are causing it. Scientists are human, though, and they often make erroneous assumptions. The projections and warnings about global warming have this flaw. The assumption is that we can actually continue to pump large amounts of CO_2 into the environment if we choose to do so. That assumption depends on believing that there is enough fossil fuel to continue using it for the next 20, 40, or 100 years like we are today.

Holmes did not mention that fossil fuels are a nonrenewable resource. Just as he searched the Web and found many resources providing information about global warming, it is also possible to find information about fossil fuel reserves. A good topic to search on is “peak oil,” a subject that, in my opinion, will be the biggest news story in the next couple of years, maybe even this year.

Global warming might be a nonissue simply because the way we live right now is unsustainable, even for the near future. Most people are incredulous that fossil fuel reserves might actually be a near-future concern, but a little studying can pique interest in the possibility.

Given the fact that our entire way of life as we know it today depends on cheap, abundant energy, it seems that this issue would be of desperate importance. Indeed, I believe it will easily and quickly eclipse any concern over global warming in the next few years. Ironically, it might also “solve” global warming, although not in any comfortable way.

Jonathan Cook
Las Cruces, N.M.
jcook@cs.nmsu.edu

Neville Holmes responds:

What worries me is that there might be enough liquid oil in reserves to tip us into crisis before it’s used up, even if the Earth doesn’t become saturated and stop soaking up huge amounts of CO_2. In any case, the energy companies are moving more to “natural” gas, shale oil, and coal. Coal is already in heavy use, and there are huge reserves of it. And it’s not just that CO_2 is being put into the atmosphere—it’s that oxygen is being taken out.

What I hear the climate forecasters saying is that waiting for fossil fuel to run out is not an option. They’re worried, too.

Neville Holmes states that meteorological records “build a picture” of climate change caused by humans. Perhaps it’s just a poor choice of words, but building a picture doesn’t seem as solid as demonstrating, showing, or proving.

The picture is then “validated by scientific modeling.” Really? Does the model validate the data, or does the data validate the model? More importantly, does data—for example, clouds and solar output—that is not used to create the model also validate the model?

Holmes further asserts, “as professionals avowing rationality” we must not only accept his facts but “become familiar” with them and “be ready to promptly counter” any denials. Maybe if he published them in a little red book—am I taking this too far, or does Holmes see us as a bunch of Quincy, M.E. types, who after a few scary “facts” from a blonde social worker leave our corpses to Sam Fujiyama and dash off in self-assured pursuit of justice?

As someone practicing the profession in question, I simply don’t have time to become expert in climate change science. Call it rational ignorance.

However, I am interested in the intersection between computer science and climate change science. So how will the profession contribute to climate change modeling? Holmes tells us what he “suspects,” not what he knows. How will the profession contribute to climate change mitigation? By teaching math proficiency to schoolchildren, of course. By now I’m beginning to suspect the remainder of the essay will not add much light to all its heat.

In fairness to Mr. Holmes, fear as motivation is difficult to accomplish on the printed page, especially when it involves all of humanity centuries from now and requires readers to remember to turn to a lower numbered page to continue reading.

I would humbly suggest that Holmes's tactic is unwarranted; computing professionals don't need to be panicked (or ideologically pure) about climate change to be interested or involved. We, and perhaps Mr. Holmes most of all, should be prepared for the possibility that computer science can aid the skeptics, too.

David Nadle
New York, N.Y.
david@nadle.com

Neville Holmes responds:

A picture is a summary impression, not a fact except per se. The usual scientific procedure validates the modeling from which the picture derives: build a model from existing data, make a prediction, compare it with reality, refine the model, and so on.

The modeling involved is very complex, encompassing a host of aspects beyond cloud cover and insolation, and the data gathering is far from sufficient. But practicing climate scientists do not dispute the general picture of anthropogenic climate change. Although their collections of data are factual observations, they are prominently and irrationally denied, which is an interesting phenomenon in itself (www.theecologist.org/archive_article.html?article=282).

Computing professionals should make themselves familiar with the facts that the climate scientists provide and they should counter denials, such as Michael Crichton's. I would not suggest that they should ignore debateable details.

The climate scientists need the help of all professionals to bring the facts and their implications to the public's attention. Among the many specific suggestions I made was for computing professionals to use the Web and media as conduits for informing the public of the relevant facts and projections.

My reading on the subject tells me not only that our civilization's effect on the Earth might already have made life "centuries from now" extremely difficult if not impossible, but also that side effects such as methane burps or changes in oceanic currents could do this within decades (www.tai.org.au/WhatsNew_Files/WhatsNew/pentagon_climate_change1.pdf).

The last thing that's needed is panic. What's needed most is a focus on and investment in scientific evaluation of possible abrupt climate change. Digital technology's role in this, and thus the role of computing professionals, is clearly of vast importance.

I always look forward to reading The Profession column, and I often agree with Neville Holmes's point of view. However, his essay in the February 2005 issue misses the mark in several important areas.

Predicting the weather is difficult, and predicting it over a long time horizon is all but impossible. "Simple projection" is not good enough. The computational aspects of the problem, although important, are the least significant part of the entire task.

Also, agreeing with the "relevant facts," which I am more than prepared to do, does not imply that something can or should be done. The stock market, for which all of the relevant facts are known—at least in theory—cannot be predicted with even a moderate level of confidence over a weekly much less a yearly horizon.

In a 50- to 100-year projection, many things can and will happen that cannot be included in the model. The history of the Earth is rife with unplanned and often catastrophic events.

I do not consider global warming an issue that anyone concerned with our profession needs to get overly heated about. There are many other current issues that can certainly profit from our attention, including education, which Holmes did mention.

Charles R. Guarino
Gaithersburg, Md.
chas.r.guarino@lmco.com

Neville Holmes responds:

There is an essential distinction between weather forecasting, which is short term, and climate prediction, which is long term. Sadly, my amateurish outline of climate modeling somewhat confused the two.

The problems with climate prediction are that it isn't local so it can too easily be dismissed as irrelevant; it's probabilistic so it can be dismissed as uncertain, and catastrophes aside, it depends more on what decisions human society makes than on anything else. Nevertheless, climate prediction has been shown to work.

There is no doubt in the minds of the climate (and environmental) scientists that if things go on as they are, the world is in for severe trouble. However, there is a problem in working out just how much trouble and in moving society to lessen the likelihood of that trouble.

But the biggest and most worrying issue with regard to catastrophes is that we do not know the facts yet. Many relevant professionals are concerned that changing conditions could trigger an event, such as the release of gigantic amounts of sequestered methane, that will change the climate fatally within a mere few years.

Because climatic changes could be bringing about new, dangerous, and possibly irreversible effects, the relevant scientific professions need our political and professional support to find out as quickly as possible what will happen to the global climate under various contingencies.

The argument that there are more important things that need our attention is that of Bjorn Lomborg, which has been thoroughly refuted (www.tai.org.au/WhatsNew_Files/WhatsNew/lomborg.pdf; www.guardian.co.uk/print/0,3858,5043956110970,00.html).

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

APRIL 1973

NETWORK SERVICE (p. 7). "Packet Communications Inc. (PCI) has submitted an application for Federal Communications Commission authorization to offer a nationwide packet-switched data communication network service. Their proposed network will be similar in concept to the Advanced Research Projects Agency's (ARPA) Network, which currently serves over 30 computers in universities, research centers and Government installations. (The ARPA Network was developed by the U.S. Department of Defense and is not available for commercial use.)"

"PCI's application notes prominently that PCI does not intend to construct or acquire any communication lines. Rather, PCI intends to utilize the lines and facilities offerings of the existing communications carriers, and in addition to these offerings to add equipment and ancillary services in such a way that the resulting network will solve the special problems of teleprocessing users."

TECHNOLOGY'S SOCIAL IMPLICATIONS (pp. 8-9). "Engineers are becoming increasingly concerned with the effects of technology on our society. The uses of technology, the priorities assigned in developing new technology, and the effects on our physical and social environment are of vital importance to our future. The present generation is the first in history to face the prospect of a planet limited in its ability to support an exponentially growing and polluting human race. The public has become increasingly intolerant of what it sees as the nuisances or hazards resulting from technology.

"In response to these concerns the Executive Committee of the Institute of Electrical and Electronics Engineers (IEEE) has formed an Ad Hoc Committee on Social Implications of Technology (C-SIT). Its areas of concern include: professionalism and social responsibility in engineering; understanding the interaction between technology and society; predicting and evaluating the impact of technology on society; and fostering study, discussion and appropriate action in these areas."

COMPUTING NETWORKS (Ruth M. Davis, p. 14). "The $64 question today is 'How do we make sure that computer networks are used to our best advantage?' I submit that we don't yet have a satisfactory answer. Further, I am convinced that we have not yet asked the right questions that let us present, to a listening audience, a realistic appraisal of what is the real power of computer networks.

"I do assert, however, that time has run out for those of us who still wish to equivocate about computer networks and their place in today's world. Experience to date with computer networks makes cowards of those of us who still vacillate in speaking of the roles for computer networks in society today.

"The same technical cowardice is exhibited by those who try to brush aside the significance of minicomputers and their steadily increasing numbers in our national inventory of computers. The tremendous power for good possessed by the coupling of minicomputers and their gigantic counterparts—the maxicomputers—by computer networks should be a cause célèbre of the computer industry and not another false reason for self-flagellation by computer professionals."

VIDEO GENERATOR (p. 38). "One of the interesting aspects of the graphic video generator presented here is that it represents an all digital, all electronic solution to a problem that heretofore required analog or mechanical devices for its solution. Thus we have been able to achieve both performance improvement and cost reduction over analog devices to obtain a device that is well-suited for minicomputer applications. It is rather interesting to speculate about other situations in which we can take advantage of advances in integrated circuit technology to arrive at totally different design approaches. It is safe to say that even the video generator described here will undergo drastic changes in coming years as newer semiconductor memories become available."

ARITHMETIC PROCESSOR (p. 45). "A new 10-digit binary-coded decimal arithmetic processor in a single integrated circuit has been announced by Texas Instruments. Designated the TMS0117, the IC is designed to process numerical data in serial BCD format. Numbers of up to 10 digits can be processed in under 100 milliseconds main operation time. The four basic operations—add, subtract, multiply, and divide—are provided; others include increment, decrement, shift left, shift right, exchange operands, add to overflow, and subtract to zero."

FIRE FIGHTING (p. 46). "Glasgow, Scotland plans to link its fire engines with a computer to fight blazes more efficiently.

"Small facsimile printers installed in the cabs of 40 fire engines will receive by radio and print out detailed information on floor plans of the burning building and its known fire hazards while the firemen are on their way to battle the blaze."

"The system, based on two Honeywell 316 computers due to be installed in June or July, will ultimately contain data on 10,000 properties. The information, to be updated daily, would include building plans and layouts, known hazardous materials in the building, and a special file of 1,000 hazardous substances and how to handle them in the case of fire."

AIRPORT SECURITY (p. 48). "Friendship International Airport is starting installation of a computer-based security system as part of its total airport security concept, Robert J. Aaronson, State Aviation Administrator, announced today. The $200,000 security system goes into operation this spring.

"The Baltimore-Washington, D.C. airport will be the first in the nation to use IBM's new controlled access system, which will link 60 security stations throughout the airport to a central computer in the airport security office.

APRIL 1989

HARDWARE TESTING (p. 12). "Why an issue of *Computer* on [software tools for hardware] testing? Because the automatic generation of test vectors and their evaluation through fault simulation are extremely complex and time-consuming operations, consuming hours and days of computer time. The complexity of the circuits on which these tools are used is growing faster than the speed of the computers on which they run. New algorithms and techniques are required for both the circuits of today and tomorrow. The techniques for solving complex problems in the physical sciences, such as vectorization and the exploitation of parallelism, do not lend themselves to solving testing problems. The purpose of this issue is to expose these problems to a wider audience and, perhaps, stimulate research that will find solutions."

TEST GENERATION (p. 16). "VLSI test generation is very complex. The test generation problem is NP-complete when defined in terms of the most common (low-level) circuit and fault models, which represent the circuit using Boolean logic elements and binary signals. Specialized design-for-testability techniques and high-performance computer-aided design workstations have held this intractability in check, but the design techniques are not without their costs and might not always apply. As a result, considerable recent research has focused on test generation techniques that give good results on wide classes of circuits and design styles. Much of this effort focuses on what we call *high-level approaches*, which view the circuit with less structural detail, that is, from a more abstract viewpoint and often hierarchically."

SYSTEM TESTABILITY (p. 59). "System testability and diagnosability depend on the design of the system and on the test sets used to perform testing and diagnosis. It is important to emphasize, however, that irrespective of the resources (for example, computer time, test time, automatic test equipment capabilities) one can afford to allocate for test set development, the system design defines an upper limit on the degree of testability and diagnosability that can be achieved in a given system. Therefore, the designer can directly affect a system's degree of testability and diagnosability by considering its test and diagnosis requirements as design requirements, not as test requirements decoupled from the design process, as designers often do today."

COMPUTER LITERACY (p. 80). "Computer Literacy Month has become a year-round campaign to promote computer literacy in North America with the establishment of the Computer Learning Foundation. Supported by major software publishing companies, as well as Apple, IBM, Tandy, and Commodore in 1988, CLF expects to receive up to $1 million in funding this year.

"CLF's announcement coincided with predictions of national technological decline touched off by an Educational Testing Service study that showed 13-year-old US students scoring the lowest in an international comparison of mathematics and science skills. Earlier, a National Research Council study reported that American students were being 'left behind' due to a mathematics teaching system that set its expectations too low."

PRINTERS (p. 83). "Microcomputer printers have come a long way since the high-decibel, low-resolution, dot matrix boxes and the daisywheel dinosaurs of yesteryear. Today, microcomputer owners can choose from a bewildering variety of fast, sophisticated, 9- and 24-pin dot matrix printers at the low end of the price range and from a plethora of whisper-quiet laser printers at the high end that rival typesetters in print quality."

A MILLION TRANSISTORS (p. 95). "The 64-bit i860 RISC microprocessor from Intel contains more than 1 million transistors and performs up to 80 million calculations per second, according to the company. The chip reportedly targets multiprocessing systems, 3D workstations, and graphic subsystems.

"The i860 contains integer and floating-point graphics units, a memory management unit, and instruction and data caches. It is manufactured using the company's CHMOS IV one-micron process."

MULTIPROCESSOR UNIX (p. 97). "HCL America has announced the M3000 series of Unix-based multiprocessor minicomputers, built around Motorola's 25-MHz 68030 processor and optional 68882 math coprocessor. The systems come with from one to six CPUs with performance ranging from 4 to more than 15 million instructions per second.

"The architecture uses a single global shared memory with two-way interleaving. The proprietary HMP (High-speed Multi Processor) bus handles transfers between CPUs and memory, while the VMEbus handles peripheral I/O."

WAFER SCALE INTEGRATION (p. 104). "Tadashi Sasaki of Sharp Corp. presented compelling arguments for the economic importance of wafer scale integration for future complex microelectronic systems … ."

"Sasaki described Japan's long-range plan to bring its WSI technology into place in the year 2000. He commented that progress is well ahead of schedule, so we should see this technology used in the 1990s."

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

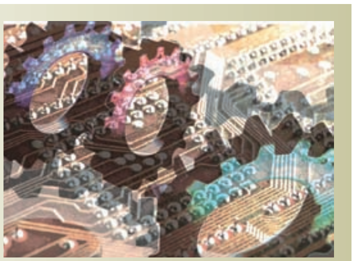
Point of Highest Leverage

Bob Colwell

At a recent workshop, John Knowles, a well-known Nashville guitarist, commented on the tendency for experts to make difficult things look easy. Knowles, a contemporary of the legendary Chet Atkins, said he once witnessed Atkins tossing off some licks that made jaws drop. Someone asked Atkins if it really was as easy for him as it appeared to be. "Didn't used to be," Atkins replied. It had only become easy for him because he worked so hard to make it that way.

Atkins developed a practice regimen that helped him maintain proficiency on his existing vast repertoire while he also learned new techniques and new music in an efficient way. He had to—he developed his career back in the days of live radio. The listeners heard what you played, no retakes, no overdubbing, no second chances, and the show must go on.

Many amateur musicians never develop a practice discipline. We all have innate strengths and weaknesses, and it's a lot more fun to play what comes naturally than to grind out the parts that don't. Unless we fight the temptation, this leads to practice sessions where we spend 90 percent of our time cruising over the parts of the music that sound good—because we can already play those proficiently, and we feel like we're pretty hot stuff—and only 10 percent on the parts we struggle with.



There's just something that feels right about grasping a basic insight and then applying it to other situations.

Especially with younger musicians, this 90/10 disparity also reflects itself in tempo—they play the "easy" parts as fast as they can and slow down drastically for the harder stuff. What are the chances this tactic will accurately express the composer's intent?

Although playing fast isn't the point of achieving proficiency on an instrument—those who think it is require an immediate infusion of Madeline Bruser's *The Art of Practicing* (Bell Tower, 1997)—many young players believe it is, and it's at least one measure of fluency.

One of the wisest pieces of advice I've heard on this topic is this: If you want to play fast, first play slowly. Only speed up the tempo when you can play the most difficult parts perfectly at the higher speed. You can tell this is good advice because it's easy to say but hard to do.

This plan naturally focuses your efforts on precisely the activity that will most benefit you: working on the trouble spots, not the parts you can already play well. On the path to improving your playing, those trouble spots are your points of highest leverage.

A UNIVERSAL PRINCIPLE

I love universal principles like the point of highest leverage. There's just something that feels right about grasping a basic insight and then applying it to other situations.

As Jeff Hawkins points out in *On Intelligence* (Henry Holt & Co., 2004), our brains seem to be wired to naturally store abstract representations of ideas or sensory patterns. This proclivity facilitates making odd connections between seemingly unrelated things. Stand-up comics earn their living by making us laugh when we realize that the allusion they have just drawn between two random things also resonates with us and, further, that the comic knew it would do that. At some deep level, we have just confirmed that we are alike in some profound way, and we like it.

If you have written computer programs, you have probably wrestled with computer performance analysis. Naïve programmers may just link dozens of off-the-shelf data structures and algorithms together, while more experienced coders design their program with an eye toward the resulting speed. But either way, you end up running the program and wishing it was faster.

The first thing you do is to get a runtime histogram of your code, which reveals that of the top 25 sections, one of them accounts for 72 percent of the overall runtime, while the rest are in single digits. Musicians who have learned to play fast know where this

Published by the IEEE Computer Society

leads: Do you a) notice that one of the single-digit routines is something you'd previously worried about and set out to rewrite it, or b) put everything else aside and figure out what to do about that 72 percent routine?

If you picked a), you are the person who is the subject of the joke about looking for a contact lens where the light is better instead of where it was actually lost. If speeding up your code is your goal, tackling the 72 percent routine first is your point of highest leverage. In this example, the code in the single digits constitutes a point of almost no leverage.

DESIGNING BY INTUITION

At Multiflow in the 1980s, we designed the first version of the TRACE VLIW minisupercomputer largely by intuition, with few simulations to guide us. Schematics of logic gates were our design entry language.

When we began designing our second-generation machine, we resolved to do better, and we bought a Hardware Description Language compiler/simulator from a small CAD company. We dutifully entered our new design into this new source language, got it to compile, and started up the simulator. Well over an hour later, the first simulated clock cycles began to appear on the screen.

Because we had no collective experience with an HDL environment, we didn't know if the one-hour startup lag was necessary or reasonable, but we did know we didn't much like it. While some of us thought our point of highest leverage was to mutter dire imprecations about the software we had purchased, Dave Papworth—being the brilliant, indefatigable engineer that he is—strode fearlessly into the lion's den and found the code module that was using up that first hour of simulation.

We didn't have source code, but we could disassemble the manageably short loop that comprised the module in question. We quickly determined that it appeared to be repeatedly traversing some kind of data structure, finding a particular data element, and setting that element to zero.

It soon dawned on us that this simulator was spending the first hour doing nothing but setting all of the internal variables to logical zero. And it was doing it in the stupidest possible way—by finding each variable independently, traversing dozens of linked-list nodes over and over, starting from the root each time.

If you've accumulated enough experience, you've also learned never to take anything for granted.

We could see that these variables were all being held in contiguous memory, regardless of how pointers were referencing them. So Dave wrote a quick one-index loop that walked over that memory and zeroed everything it found. Then he patched the object code to substitute his initializer for the one-hour version. Presto, the simulator now booted in 10 seconds.

To prove that no good deed goes unpunished, we helpfully told the software vendor about the problem and offered them our solution. A few weeks later, we received a letter from their attorney threatening to sue us for having reverse-engineered their object code. Given that we were a funds-challenged startup, that was certainly not their point of highest leverage on anything.

We dutifully promised never to help them again.

THINK LIKE LEWIS AND CLARK

Beginning the design of any major undertaking is like the Lewis and Clark expedition across the American west in the early 1800s. They knew enough to know that they didn't know enough. So they packed accordingly, to give themselves options when things inevitably went awry. They based their planning decisions on a mixture of data and intuition born of experience.

Designing a microprocessor is much the same, except for the boats, horses, and guns. There are things you know and things you don't know. In planning the project, your job is to set things up so that you can later steer the project safely through whatever contingencies may arise.

How do you do that? Start with the givens. Experience gives you a set of things you can take for granted: techniques, know-how, who is good at what, tools that have proven themselves, validation plans and repositories, how to work within corporate planning processes. If you've accumulated enough experience, you've learned never to take anything for granted, but some things don't need to appear at the top of your worry list.

It is, however, crucial to identify exactly what *should* be at the top of your worry list. Important changes (read: risks) such as new process technologies automatically go on that list because if trouble arises there, you have few viable alternatives. If you're contemplating a new microarchitecture, that goes at the top of the list. After all, your team hasn't conjured up the new microarchitecture yet—you're only asserting that you need one. The gap between those two facts may turn out to be insurmountable.

Perhaps you need new compiler techniques for your design to meet its targets. Or maybe you need a new on-the-fly runtime compilation scheme that has never before been attempted, as was the case for Transmeta a few years ago.

Whatever you perceive to be your project's biggest risks are also highly likely to be your points of highest leverage—the places where your immediate actions are likely to have the highest payoffs. If you're not sure some crucial new idea is going to work, tackle that first. Simulate it, analyze it, fake it in a system, discuss it, debate it, establish prizes for people who make progress on it—whatever it takes to run the idea to ground.

If your initial attempts don't increase your confidence that the project is doable, you should also launch serious attempts to find contingency plans in case it ultimately turns out to be unworkable. And always remember Swope's dictum regarding backup plans: If you're going to have one, then take it just as seriously as your main plan. Otherwise, you're worse off than if you had none. Getting this wrong was a key reason for the 1986 *Challenger* shuttle disaster.

Obsessing over your top worries might not be fun, but it is your point of highest leverage over the final quality of the product.

DO YOUR BOSS'S JOB

During the 1970s, I had a boss at Bell Labs who told me that the key to getting promoted was to "do your boss's job." What he meant was that at every level in a well-run company, the person in a particular job is being held accountable by someone else.

If you are an engineer at a "leaf-node" in the corporate tree, you have a set of technical tasks to do on a given schedule that are subject to various constraints. Your first-line supervisor has a set of deliverables, of which yours are a part. Your supervisor, in turn, answers to a second-line manager, who needs what your boss is accomplishing to satisfy whoever that manager reports to.

My Bell Labs boss was reminding me that doing the work that is assigned to you is a requirement, but, ultimately, the people who contribute the most value overall see the bigger picture—the entire management chain and its guiding vision—and find ways to contribute to that larger reality. They not only accomplish their required tasks in an exemplary fashion, they also routinely look for and find ways to make contributions they were *not* expected to make.

Better evidence of leadership capability can't be found. And it addresses one of the perennial worries about promotions: Can the person operate successfully at the next level up?

This whole-project thinking is surprising in two ways. It's surprisingly effective—responsibilities at Bell Labs were allocated down a management chain by very senior, experienced managers, probably in a logical, intelligent manner. But everyone can make mistakes, and engineers at the "leaf-nodes" often know things about the technology that the senior people have missed. That's the other surprise—how few engineers take the initiative to occasionally step outside their box to see if the overall project picture looks right.

Whole-project thinking is surprisingly effective.

DON'T ABUSE IT

Of course, just as a good idea like Intel's Constructive Confrontation can be misused (sometimes turning into "I Just Want to Say I Hate You and the Horse You Rode In On"), so can this idea of contributing outside your assigned zone. Sometimes a corporate culture becomes so ossified that people only listen in on officially designated communications channels. It is impossible, in my opinion, to design world-class products in such an environment, but I know for a fact that such places exist. In a workplace like this, the issue isn't finding the point of highest leverage, it's finding any point of leverage whatsoever.

Other times, people abuse this idea by making a lot of noise elsewhere to cover their inability to get an assigned job done. Or maybe they think stepping out of line is a way to get noticed and jump onto the career fasttrack.

A particularly dangerous ploy is to bring an out-of-the-box idea to a supervisor (which is the correct first step) but then threaten to take it higher in the management chain if that supervisor doesn't salute fast enough. Such a situation requires good judgment on the part of all concerned. Sometimes your boss just misses the point, and you are right not to let an idea die. But other times, she is weighing implications of what you are proposing that you may not even be aware of, and her counsel to you might be wise even if unwelcome.

If you have really thought about something, and you have sanity-checked it with a few of your peers, but your boss still wants to kill it and you can't understand why, then you should be willing to take it "over her head." Just don't get carried away with this process, going around her on something every other week, or multiple levels of corporate management will learn to cringe when they hear your name, which is almost never good for your career.

Archimedes once said that if given a lever long enough and a place to stand, one could move the world. Evidently, Archimedes thought that was an inferior insight to his realization that buoyancy is related to the weight of a volume of water because history did not record his running naked through the streets shouting "Eureka!" about levers.

Archimedes would definitely have understood the idea of a point of highest leverage. But could he have played "Brown Bomber" as fast as Pete Huttlinger? ■

Bob Colwell was Intel's chief IA32 architect through the Pentium II, III, and 4 microprocessors. He is now an independent consultant. Contact him at bob.colwell@comcast.net.

FREE Access to 300 Online Books for Computing and Information Technology!

The *IEEE Computer Society Online Bookshelf* resource lets you...

✓ Keep up to date on current and developing technologies
✓ Learn about a new topic
✓ Search by keywords, author, title, ISBN, and publisher
✓ Use a personalized bookshelf for quick retrieval of your favorite books
✓ Use bookmarks to easily return to a specific chapter or page

Powered by:
books24x7®
Referenceware℠
For Professionals

Members get unlimited access to the *IEEE Computer Society Online Bookshelf*, a rotating collection of 300 unabridged books on a variety of technology topics, including…

Business and Culture
- Management
- E-Commerce
- Research & Development

Certification and Compliance
- Cisco: CCNA/CCNP
- Java
- Microsoft: MCAD/MCSD

Databases
- Data Mining
- Oracle
- Microsoft SQL

Desktop & Office Applications
- Integrated Software
- Microsoft Office
- Productivity Tools

Enterprise Computing
- Implementation & Design
- GroupWare
- Intranets & Extranets

Graphic Design & Multimedia
- 3-D Graphics
- Image Editing
- Multimedia

Hardware
- Power Supplies & Electronics
- Printers & Peripherals
- Routers & Switches

Networks & Protocols
- Network Management
- Protocols & Standards
- Remote Access

Operating Systems
- Linux OS
- UNIX
- Windows

Programming Languages
- .NET Framework
- C / C++
- J2EE

Security
- Intrusion Detection & Prevention
- Disaster Recovery
- Information Security

Software Engineering
- Architecture
- Design
- Software Testing

Telecommunications
- Wireless
- Security
- Telephony

Web Programming & Development
- Web Services
- Scripting and Programming
- Site Development & Maintenance

Note: Topics are subject to change as titles are added or removed to ensure currency and variety of coverage.

A sample of popular titles…

The Art of Software Architecture: Design Methods and Techniques *—John Wiley & Sons*
A Guide to the Project Management Body of Knowledge *—Project Management Institute*
Cisco Routers and Switches *—SkillSoft Press*
ASP.NET Weekend Crash Course *—John Wiley & Sons*
Linux Applications Development for the Enterprise *—Charles River Media*
Software Testing Fundamentals: Methods and Metrics *—John Wiley & Sons*
.NET Development for Java Programmers *—Apress*
Fundamentals of Network Security *—Artech House*
Complex IT Project Management: 16 Steps to Success *—Auerbach Publications*
Bulletproofing Web Applications *—John Wiley & Sons*
Wireless Communication Circuits and Systems *—IEEE*
C# for Java Developers *—Microsoft Press*
Mobile Commerce: Technology, Theory and Applications *—Idea Group Publishing*
Hack Attacks Testing: How to Conduct Your Own Security Audit *—John Wiley & Sons*

IEEE

IEEE
COMPUTER
SOCIETY

Take advantage of the *IEEE Computer Society Online Bookshelf* today!

Visit www.computer.org/bookshelf

Will Binary XML Speed Network Traffic?

David Geer

With its ability to enable data interoperability between applications on different platforms, XML has become integral to many critical enterprise technologies. For example, XML enhances e-commerce, communication between businesses, and companies' internal integration of data from multiple sources, noted analyst Randy Heffner with Forrester Research, a market-analysis firm.

XML use is thus increasing rapidly. Analyst Ron Schmelzer with market-research firm ZapThink predicted XML will rise from 3 percent of global network traffic in 2003 to 24 percent by 2006, as Figure 1 shows, and to at least 40 percent by 2008.

However, XML's growing implementation raises a key concern: Because it provides considerable metadata about each element of a document's content, XML files can include a great deal of data. They can thus be inefficient to process and can burden a company's network, processor, and storage infrastructures, explained IBM Distinguished Engineer Jerry Cuomo.

"XML is extremely wasteful in how much space it needs to use for the amount of true data that it is sending," said Jeff Lamb, chief technology officer of Leader Technologies, which uses XML in teleconferencing applications.

Nonetheless, said Heffner, "XML adds intelligence on top of data in motion to make that data more manageable across vast technical boundaries. XML is so important that the industry is looking for ways to make its data load more manageable."

Proponents say a thinner binary XML will help. XML currently uses only a plain-text format.

The World Wide Web Consortium (W3C), which oversees and manages XML's development as a standard, and Sun Microsystems are working on binary XML formats.

Some industry observers have expressed concern that multiple formats or proprietary implementations of binary XML could lead to incompatible versions, which would reduce the openness that makes the technology valuable.

XML'S PROBLEMS

The W3C started work on XML in 1996 as a way to enable data interoperability over the Internet. The consortium approved the standard's first version in 1998.

A key factor driving the standard's development was increased Internet and network usage requiring companies on different platforms to be able to communicate. Many businesses also wanted to make legacy data available to new Web-based applications.

How XML works

XML is a markup metalanguage that can define a set of languages for use with structured data in online documents. Any organization can develop its own XML-based language with its own set of markup tags. For example, a group of retailers could agree to use the same set of tags for categories of data—such as "customer name" or "price per unit"—on a product order form.

A typical XML file also includes information about a document unrelated to content, such as the encryption used and the programs that must be executed as a result of or as part of processing the file.

The XML document type definition describes a document's metadata rules—identifying markups, stating which elements can appear, and noting how they can be structured—to the applications that must work with it. XML documents are written and stored as text, and documents are read via either text editors or XML parsers.

By enabling cross-platform communications, XML eliminates the need to write multiple versions of documents or to use costly and complex middleware. However, the files contain considerably more information than just the content they are communicating.

XML is the basis for important technologies such as Web services and important standards such as the Simple Object Access Protocol, a way for a program running in one operating system to communicate with a program running in another by using HTTP and XML as the information-exchange mechanisms.

Performance hit

Standard XML is bigger and, more importantly, less efficient to process than a binary version would be, thereby slowing the performance of databases and other systems that handle XML documents.

For example, IBM's Cuomo said, "You have information in a database that is SQL compatible. You get result sets out of the database and, in our case, you put it into Java Object format, convert it to XML and then to HTML before you send it to the end user." The process must be reversed when the user sends back material, Cuomo explained. "This consumes MIPS," he noted.

Using XML also causes Web services, which are becoming increasingly popular, to generate considerable traffic.

In addition, said Glenn Reid, CEO of Five Across, a Web development firm that works with XML, "You can't really start to process an XML file until you've received the entire thing." Because of the syntax, systems must read to the end of an XML document before determining the data structure. On the other hand, systems can process some file types as they receive them.

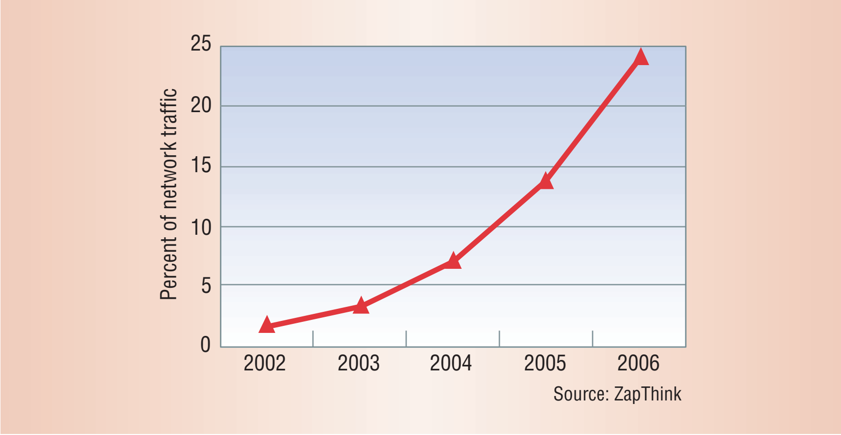
SOLVING THE PROBLEM

One approach to solving XML-related problems is using appliances dedicated to making the documents more manageable. These products—sold by vendors such as DataPower, F5 Networks, Intel, and Sarvega—can preprocess an XML document by applying XSL (Extensible Stylesheet Language) transformations to reorganize its structure so that the host system doesn't have to do all the work.

The appliances can also compress XML files or streamline them by eliminating material—such as spaces or tabs—present only to keep the material in textual, human-readable form.

However, noted Leader Technologies' Lamb, "These appliances are expensive." It would be preferable to make XML itself easier to work with, he said, to reduce costs and enable more complex and rich XML-based applications.

Thus, the leading proposal to alleviate XML's performance hit is binary XML, a format that optimizes documents for faster handling.



Source: ZapThink

Figure 1. XML usage, as represented by XML's percentage of all network traffic, has grown rapidly during the past few years and is predicted to continue doing so.

W3C specifications

The W3C has formed the Binary Characterization Working Group (www.w3.org/XML/Binary/) to study binary XML. The working group has issued three recommendations—backed by software vendors such as BEA Systems, IBM, and Microsoft—designed to make handling XML files more efficient.

"All three of these specifications have reached the final stage of the W3C recommendation track process," said Yves Lafon, a W3C XML protocol activity leader who also participates in the working group.

XML Binary Optimized Packaging. XOP makes XML files smaller by extracting binary parts such as images, sending them as a separate package with the document, and providing a uniform resource identifier as a link that recipient systems can use to access the extracted material, explained Lafon.

Currently, images and other binary data in a standard XML document must be encoded in base64 to be processed with the rest of the file. Base64 encodes binary data as ASCII text. The process divides three bytes of the original data into four bytes of ASCII text, making the file one-third bigger.

Using XOP eliminates the need for larger files, as well as the time and effort necessary to conduct base64 conversions.

Message Transmission Optimization Mechanism. The W3C has incorporated XOP's method for representing binary data into the MTOM communications protocol. In essence, MTOM implements XOP for SOAP messages. MTOM uses MIME (multipurpose Internet mail extensions) multipart to package the message, after XOP processing, with the extracted binary parts, Lafon explained.

Resource Representation SOAP Header Block. RRSHB provides a way for an application receiving an XML message—from which binary parts have been extracted via XOP and packaged with the main file via MTOM—to retrieve the binary parts. In the message's SOAP header, RRSHB references where the binary parts are and how the application receiving the message should access them.

Sun's Fast Infoset Project

Sun has started the Fast Infoset Project (https://fi.dev.java.net), an open source implementation of the International Organization for Standardization's and the International Telecommunication Union's Fast Infoset Standard for Binary XML, used for turning standard XML into binary XML (http://asn1.elibel.tm.fr/xml/finf.htm).

According to Sun Distinguished Engineer Eduardo Pelegri-Llopart, the technology encodes an XML document's information set (infoset) as a binary stream and then substitutes number codes for all of the metatags, thereby reducing a file's size. Included in the stream is a table that defines which metatag each number code stands for.

The overall document is generally smaller than a comparable textual XML file, and recipient systems can parse and serialize it more quickly.

In early tests, Sun says, XML applications perform two or three times faster when using software based on its technology.

CONCERNS OVER INCOMPATIBILITY

According to Leader Technologies' Lamb, XML is currently standardized and interoperable largely because it uses a plain-text format. Moving to binary XML without maintaining standardization, he said, would cost much of the interoperability for which XML was created.

Five Across' Reid expressed concern that the binary XML efforts might lead to incompatible versions of the technology. In addition, he said, different companies could create incompatible binary formats, including some for specific applications such as mobile phones, which have severe processing and memory constraints.

Some industry observers say that future increases in network and processor performance could improve systems' ability to handle standard XML and thereby eliminate the need for binary XML.

However, stated Sun's Pelegri-Llopart, binary XML would offer a badly needed solution sooner than waiting for adequate network and processor improvements to occur.

And, according to IBM's Cuomo, faster networking won't work or isn't available in many situations, such as in small towns or developing countries in which broadband networking isn't readily accessible or affordable.

Because binary XML is suitable when network efficiency is important, ZapThink's Schmelzer said, users might decide to work with it only for high-volume applications that demand the best performance, like those in financial transactions, telecommunications, and multimedia.

Even if a single approach is standardized, there will still be applications and systems that can't work with binary XML. In some cases, standard textual XML will be preferable because it is easy to code by hand and is universally understandable.

There is some concern about how well binary XML would work with Web services even if it is standardized. Many Web services models allow intermediate entities—such as an XML security gateway or a policy-enforcement tool—to act on a message during transmission. The overhead involved if intermediaries must code and decode messages could reduce or eliminate binary XML's efficiency.

Nonetheless, Cuomo said, the urgent need for a faster XML that would reduce the burden on CPUs, memory, and the network infrastructure will help ensure its future success. ■

David Geer is a freelance technology writer based in Ashtabula, Ohio. Contact him at david@geercom.com.

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

Mobile Phones: The Next Frontier for Hackers?

Neal Leavitt

Security experts are finding a growing number of viruses, worms, and Trojan horses that target cellular phones. Although none of the new attacks has done extensive damage in the wild, it's only a matter of time before this occurs, noted Aaron Davidson, CEO of SimWorks International, a New Zealand-based antivirus company.

Security researchers' attack simulations have shown that before long, hackers could infect mobile phones with malicious software that deletes personal data or runs up a victim's phone bill by making toll calls. The attacks could also degrade or overload mobile networks, eventually causing them to crash. And they could be even more insidious in the future by stealing financial data, said Davidson.

Smart phones represent a particular risk. They offer Internet connectivity, function like minicomputers, and can download applications or files, some of which could carry malicious code.

Market research firm IDC predicts that by 2008, vendors will sell more than 130 million smart phones, representing 15 percent of all mobile phones. ARC Group, another market research firm, said 27 million smart phones were sold worldwide in 2004, accounting for about 3 percent of the total global handset market.

Mobile-device technology is still relatively new, and vendors have not developed mature security approaches, according to Matias Impivaara, director of mobile security services for antivirus-software vendor F-Secure. "The most worrying scenarios are not coming from stereotypical virus writers such as teenagers but from more organized [criminal groups]."

To counter the growing threat, antivirus companies have stepped up their research and development. In addition, vendors of phones and mobile operating systems are looking for ways to improve security.

DRIVING THE MOBILE ATTACK

Financial gain is perhaps the principal driving force behind mobile malicious code, said Joshua Wright, deputy director of training for the SANS Institute, a research and education organization that operates the Internet Storm Center early-warning system.

Viruses can let intruders access passwords or corporate data stored on a cell phone. Also, attackers can manipulate a victim's phone to make calls or send messages, a crime called *theft of service*.

Users are just beginning to make purchases and conduct financial transactions over mobile devices, particularly in Europe and Japan. Many industry observers expect such activity to increase dramatically during the next few years. Even now, some mobile-phone users store their credit card numbers and other financial information in electronic wallet software.

Cell phones are becoming targets largely because of their widespread use, providing millions of potential targets. They also have numerous vulnerabilities. For example, they generally don't come with antivirus software.

In addition, mobile devices are much more connected to the outside world than PCs. "Phones are primarily used to communicate. They are built to make communication as easy as possible," noted SimWorks' Davidson. "Phone users want to communicate, and viruses want to be communicated."

Some hackers may be discouraged from targeting wireless devices because, to reach a large number of victims, they would have to design separate sets of malicious code for each mobile operating system and each processor platform, said Vanja Svajcer, principal virus researcher for SophosLabs, a global network of virus and spam analysis centers overseen by antivirus company Sophos.

Cell phones use a variety of processor platforms, including those from ARM, Motorola, and Texas Instruments.

The three dominant mobile-device OSs are Symbian, Palm, and two Windows CE versions: Pocket PC Phone Edition and Smartphone Edition. According to Canalys, an industry-analysis research firm, Symbian's market-leading share rose to 53 percent in 2004 from 38 percent in 2003. Thus, Symbian phones have become malware writers' favorite target.

"If a generic language such as Java is used for creating the malicious code,

Published by the IEEE Computer Society

it could affect devices that support Java," noted Impivaara.

NEW MOBILE MALICIOUS CODE

Because mobile malware is relatively new, virus writers have released it primarily as proof-of-concept code so far, according to Wright.

F-Secure found the first mobile virus—designed for Palm devices—in 2000. The company estimates hackers released about a dozen mobile viruses between 2001 and 2003. In 2004, security researchers discovered 21. And F-Secure already identified 10 in the first two months this year.

Several recent mobile viruses have been particularly noteworthy.

Cabir

The well-known 29A Eastern European hacker group, which specializes in creating proof-of-concept viruses, sent the first version of the Cabir worm, known as Cabir.A, to a number of antivirus firms.

Cabir runs on smart phones from vendors such as Motorola, Nokia, Panasonic, and Sony Ericsson that support the Nokia-licensed Symbian Series 60 platform.

Cabir can be acquired via a shared infected application or it can replicate via Bluetooth, a short-range, radio-based, wireless connectivity technology. The worm arrives on victims' phones as an .SIS (Symbian installation system) application-installation file.

Target devices display a message asking users if they want to receive a message via Bluetooth and then ask for further confirmation if the application is not digitally signed by an authorized Symbian authority. If the user chooses to receive the file, it installs and then sends itself to other Bluetooth-enabled devices within the technology's 10-meter range.

After infecting a phone, Cabir.A displays the text "Caribe VZ/20a"and Cabir.B displays "Caribe" on the victim's screen. The worm also interferes with a host device's normal Bluetooth system by forcing it to constantly scan for other enabled devices. This reduces a device's battery life and either makes Bluetooth unavailable to legitimate applications or degrades Bluetooth performance, explained Davidson.

A few users of sites that distribute *warez*—software stripped of copy protection and placed on the Internet for downloading, generally illegally—have reported accessing Cabir-infected applications.

"We recently reported its arrival in Australia and in other countries including China, the Philippines, Singapore, and the United Arab Emirates," Davidson said.

Security experts are finding more malicious code that targets mobile devices.

Sophos advises users to protect themselves against Cabir and other Bluetooth-based threats by simply turning off the Bluetooth settings in their phones that let other devices recognize and contact them via the technology.

There have been several Cabir variants. Cabir.H, for example, attaches itself to applications' installation files on a phone. Victims who download and install the application can unknowingly infect their devices with Cabir.

Skulls

Skulls is a Trojan horse and thus masquerades as a useful application to convince users to install it. Its authors wrote Skulls to appear to be an application that lets users preview, select, and remove design themes for their phone screens.

Hackers deliberately—and file sharers inadvertently—uploaded Skulls to several shareware sites, from which unsuspecting users have downloaded the application.

Skulls targets the Nokia 7610 phone, although some other Symbian Series 60 phones can also install it.

According to SophosLabs' Svajcer, Skulls makes the original Symbian binaries for everyday functions—such as file management, Bluetooth control, messaging, Web browsing, and application installation and removal—useless by replacing them with nonfunctional binaries. The phones can then only make and receive calls.

Because Skulls disables Symbian applications, only phones with third-party file managers can remove the Trojan. Those using Symbian's file manager must perform a hard reset, thereby erasing all stored data. Skulls also replaces each application icon with a skull and crossbones.

Each of several Skulls variants and hybrids has a slightly different effect. For example, Skulls.D—posted to several Web discussion forums and warez sites—pretends to be a Macromedia Flash player for Symbian Series 60 devices. The variant replaces system binaries related to application uninstall and Bluetooth control with nonfunctional binaries, installs the Cabir.M worm, and disables antivirus programs and third-party file managers.

Mquito

Mquito is a version of the popular Mosquito game whose copy protection crackers have broken. Once the game is installed on a Symbian Series 60 device, it surreptitiously sends unauthorized SMS text messages to high-cost toll phone numbers in Germany, Holland, Switzerland, and the UK.

Reportedly, said Vincent Weafer, senior director of Symantec Security Response, game-maker Ojom deliberately added Mosquito's hidden SMS functionality as a copy-protection technique. He said that Ojom, which declined to comment for this article, wanted the program to send an SMS message alerting the company if someone was using an unlicensed copy.

"The Symbian OS provides the functionality required for any application to send and receive SMS messages with or without user intervention," said

Symbian spokesperson Peter Bancroft.

Current versions of the game no longer have the hidden SMS functionality, but cracked versions with the capability are still available online for downloading.

Potential Future Attack Approaches

In the future, mobile viruses will likely try to spread by using the Short Message Service or Multimedia Messaging Service, according to Joshua Wright, deputy director of training for the SANS Institute, an information-security research and education organization.

A fast-spreading SMS or MMS mobile virus could send huge numbers of messages and inundate a carrier's service center or mobile infrastructure, noted John Girard, vice president and research director of security for Gartner, a market research company.

Security vendor SimWorks International recently identified the first Symbian virus capable of spreading via MMS messages. The CommWarrior.a virus scans an infected phone's address book. Using the addresses, it sends itself via MMS to Symbian Series 60 cell phones anywhere in the world, not just within the 10-meter range of Bluetooth, a wireless technology used by some mobile viruses.

SMS and MMS

SMS—a paging-like service for cell phones that use the Global System for Mobile and Code-Division Multiple-Access technologies—is used to send brief text messages to mobile phones. "At 168 characters, the data capacity is very small. It [thus] may not be a useful mechanism for spreading mobile viruses but could let a virus cause harm by generating enormous quantities of SMS traffic," said Aaron Davidson, CEO of SimWorks International.

MMS—an advanced type of SMS for phones that are based on General Packet Radio Service technology—carries up to 50 Kbits of data, large enough for many viruses.

Other approaches

Many cell phones run e-mail applications. However, a virus author probably would not write mobile malware that uses e-mail attachments to transmit itself to wireless devices, as occurs with PCs, according to Wright.

The damage would not be sufficiently great because, unlike SMS and MMS, not many people use cell phones exclusively to read e-mail, explained Vanja Svajcer, principal virus researcher for SophosLabs, a global network of virus and spam analysis centers overseen by antivirus company Sophos. Virus writers would prefer to send malicious code via approaches used primarily by cell phones, he said.

"As mobile instant messaging's popularity grows, the same sorts of attacks seen on PCs are likely to appear, such as hijacking lists of IM names and sending links to recipients to direct them to malicious sites," said Girard. Mobile viruses could also send out IM messages with the malicious code attached, he noted.

The community that develops *warez*—software stripped of copy protection and placed on the Internet for downloading, generally illegally—could make infected mobile games available online to unsuspecting users, added Matias Impivaara, director of mobile security services for antivirus-software vendor F-Secure.

Windows CE virus

The 29A hacker group has written the first proof-of-concept virus for Microsoft's mobile operating system.

Razcan Stoica, spokesperson for BitDefender, a Romanian security company, said the WinCE.Duts.A virus sends recipients a message asking for permission to download.

When granted permission, the virus tries to infect all executable files bigger than 4,096 bytes. During the infection process, the virus appends itself to a file. If a victim tries to run an infected file, the virus will function but the application won't. The virus then attempts to spread, looking for new files to infect.

"When files are exchanged between devices, the virus spreads along with them," said Stoica. "Being a proof-of-concept virus, it has no payload. However, it could be easily adapted."

Metal Gear

Metal Gear is a Trojan camouflaged as a mobile version of the *Metal Gear* Solid video game. To get infected with the Trojan, users must open and install the fake Metal Gear game.

According to SimWorks' Davidson, designers often port PC games to mobile platforms, so *Metal Gear* fans might believe the Trojan actually is a mobile version of the game.

The *Metal Gear* Trojan disables antivirus programs and installs the Cabir.G worm, which tries to spread a second Trojan program, SEXXXY, to nearby phones via Bluetooth.

"Users will have difficulty repairing their phones because the *Metal Gear* Trojan effectively disables all tools on the phone necessary to undo the damage," said Davidson.

Lasco

Lasco.A, a proof-of-concept program, uses Bluetooth to infect mobile phones running on the Symbian Series 60 platform. Lasco can create its own .SIS installer file, which lets the application load itself onto other Bluetooth-enabled devices within range. It can also insert itself into other .SIS files and thereby spread during file sharing. According to the SANS Institute's Wright, Lasco is the first mobile malware that can use both methods to infect devices, thereby increasing its ability to spread.

Once installed, Lasco changes a phone's file directory to include the appended file. It also sets up the .SIS file to tell the target phone's application manager to run Lasco during installation.

The file arrives in the phone's messaging inbox and asks, "Install Velasco?" If the user gives permission, the worm activates and looks for new devices to infect.

Gavno

Gavno, a Trojan reported to SimWorks but not yet found in the wild, contains an application file that hackers have deliberately rendered invalid by, for example, removing critical data. When the Symbian OS tries to use it as the type of file it is supposed to be, problems arise that cause a series of cascading errors in Nokia 6600 and 6630 phones.

The errors cause the OS to become unstable, limiting infected phones to receiving calls. Gavno then makes the phone reboot, which produces similar errors.

One of two variants, Gavno.B, includes a Cabir version.

SimWorks' Davidson predicted that mobile malware will become more sophisticated as virus writers gain more experience and hackers publish the source code for various viruses, worms, and Trojans. The "Potential Future Attack Approaches" sidebar provides more information.

However, Wright said, device vendors and service providers will also increasingly provide better antivirus and other security applications for cell phones, as the "Response and Prevention" sidebar explains.

John Girard, vice president and research director of security for Gartner , a market research company, said, "Antispyware and antivirus functionality will help mobile users be more resistant, but like in the PC world, there will always be hackers who want to rise to the challenge. Mobile device users will have to learn to be more vigilant to ensure that their data and communications stay secure." ■

Response and Prevention

While users now often protect their PCs with antivirus software, such measures are not so widespread in cellular phones. Most users aren't aware of potential mobile malicious code problems and thus aren't vigilant in preventing or avoiding attacks on their phones, said Vanja Svajcer, principal virus researcher for SophosLabs, a global network of virus and spam analysis centers overseen by antivirus company Sophos.

Also, few mobile phones currently have antivirus software, although companies are starting to install it. For example, Japan's NTT DoCoMo now provides buyers of its new Symbian-based FOMA 901i phones with McAfee's VirusScan technology.

Nokia has introduced two phones with Symantec Client Security software, which is preloaded on the memory card and can be updated wirelessly through Symantec LiveUpdate.

Antivirus-software vendor Trend Micro recently rolled out Trend Micro Mobile Security, which provides antivirus and antispam protection for mobile devices' SMS applications.

Mobile antivirus programs are similar to those used for PCs in that they scan files for code strings associated with viruses or watch for potentially harmful activities like those that viruses frequently undertake. Although they must be simpler than PC antivirus programs because mobile devices offer less memory and performance, the OSs and viruses they deal with are also simpler, explained Razcan Stoica, spokesperson for BitDefender, a Romanian security company.

Meanwhile, Symbian's latest version, OSv9, works with Symbian Signed. In this industry-supported program, application developers sign their programs with a tamper-proof digital certificate to verify their identity.

Any Symbian Signed-compliant application will install on a Symbian phone without requiring warning boxes, noted company spokesperson Peter Bancroft. Users could refuse to accept unsigned applications.

"This digital certification will prevent applications from being tampered with, such as by including malware," Bancroft said.

Neal Leavitt *is president of Leavitt Communications, an international marketing communications company based in Fallbrook, California. He writes frequently on technology-related topics. Contact him at neal@leavcom*

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

Proposed Standard Promises to Enhance Network Operations

Two major standards organizations are working on a proposed specification that promises to enhance current and next-generation converged, broadband, and IP networks and services. Agere Systems' application service resiliency(ASR) technology would help telecommunications carriers provide new prioritized, revenue-generating services for select traffic flows while reducing operational costs.

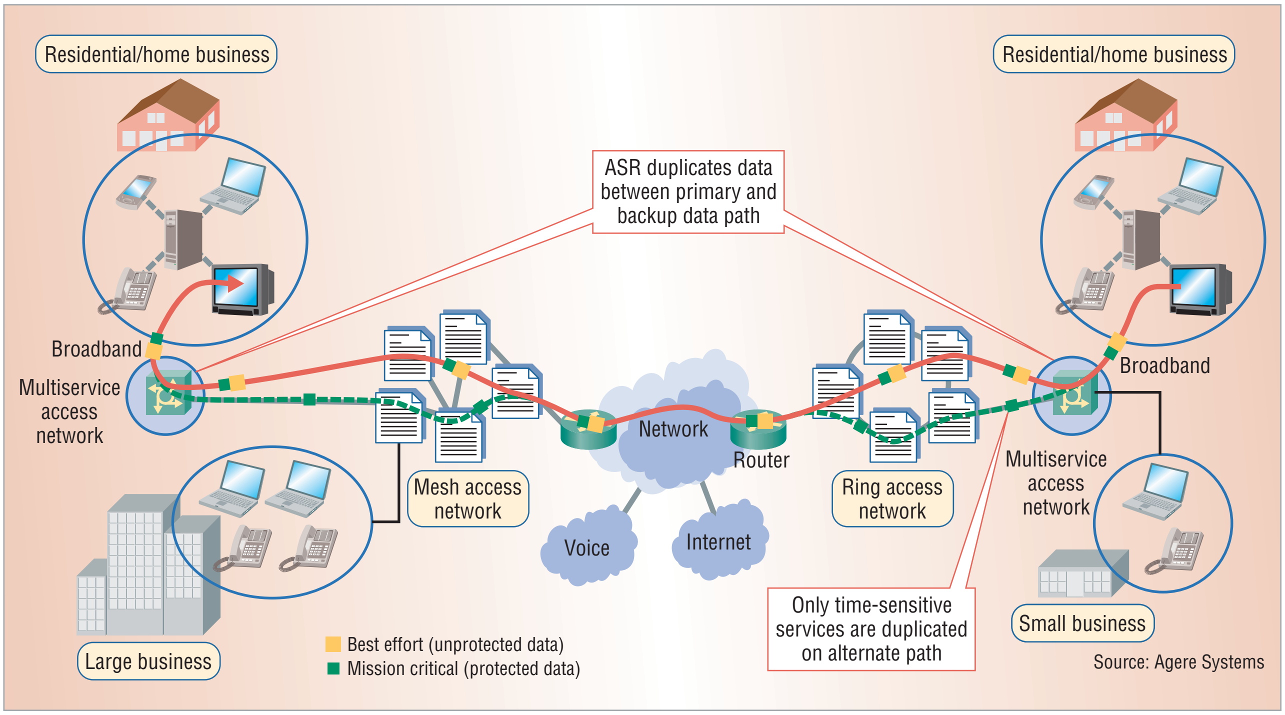
The International Telecommunication Union's Next Generation Networks Focus Group and the American National Standards Institute's T1S1 and T1A1 groups are working on standardizing ASR for converged voice, video, and data networks.

ASR is designed to maintain application-service continuity regardless of failures in the communications network such as fiber cuts, unplugged equipment line cards, and remote-network failures, according to Chris Hamilton, Agere's director of strategic marketing.

Carriers could offer such reliability levels on specific traffic flows designated by customers, rather than having to provide them for all traffic on a network as has been the case in the past. Providers typically have had to spend money to run two networks: one that provides reliability for high-priority traffic and another for other traffic.

ASR would eliminate this expense, Hamilton explained. The technology would also let carriers make money by charging for service guarantees on important transmissions. High reliability is important for mission-critical and delay-sensitive services such as Internet telephony, streaming video, and e-commerce applications.

Network processors running Agere's unbreakable access algorithms—developed with firms such as British


Residential/home business
Residential/home business
ASR duplicates data between primary and backup data path
Broadband
Multiservice access network
Broadband
Network
Router
Mesh access network
Ring access network
Multiservice access network
Voice
Internet
Large business
Small business
🟨 Best effort (unprotected data)
🟩 Mission critical (protected data)
Only time-sensitive services are duplicated on alternate path

Source: Agere Systems

Two organizations are standardizing Agere Systems' application service resiliency technology, designed to maintain application-service continuity regardless of failures in current and next-generation converged, broadband, and IP networks. ASR would help mission-critical and delay-sensitive services and enable telecommunication carriers to offer prioritized, revenue-generating services for select traffic flows. Customers would tag high-priority traffic, and the ASR algorithm would then multicast the data along two paths. Because there are two flows, a system could use one if network problems interrupt transmission of the other.

Telecommunications, Fujitsu, and Marconi—would implement ASR.

Customers would tag traffic—for example, within an IPv4 or IPv6 header—for service reliability. The ASR algorithm would then multicast the same set of data along two paths, telling routers, switches, and other networked devices along the way that the traffic is high priority, Hamilton explained. Because there are two flows, a system could use one if network problems interrupt transmission of the other.

The processor running the algorithm includes related vital functions such as traffic classification and management for IP applications and services.

In the past, IP networks used rerouting mechanisms to deal with network failures. However, this approach requires using multiple routers to recalculate proper traffic paths, a process that takes several seconds, which is too long for real-time, mission-critical services.

Mark Seery, IP infrastructure program director with RHK, a market research firm, said ASR is an example of using packet-based technology to improve network services. However, he added, implementing this new approach could be expensive at first, and "it's not clear to me whether this is a burning problem that needs to be solved today."

Agere is working with AT&T, Cisco Systems, and Nortel Networks to refine ASR for standardization. The specification is scheduled to go to ITU study groups this summer and, after standardization—slated for later this year—to the ANSI, Hamilton explained. ■

Casinos Bet on RFID Technology

In an effort to improve marketing and stem fraud, casinos around the world are starting to use a high-tech innovation in a low-tech piece of equipment: betting chips that contain radio-frequency identification technology.

RFID uses electromagnetic or electrostatic coupling in radio frequencies to send signals that identify a host object to another system. An RFID system has an antenna, a transceiver, and a transponder that contains the radio circuitry and the host system's identifying information. The transponder sends the antenna the identifying signals for transmission to a processing device.

RFID has been touted as a replacement for the bar code technology used to identify store merchandise. Casinos, though, are using RFID to identify chips and monitor gambling activity. For example, casinos can give RFID chips to gamblers and then keep track of how much they bet. The casino could then reward large-scale gamblers with freebies to keep them coming back, noted Russel McMeekin, president and CEO of RFID chip vendor Progressive Gaming International.

RFID systems could also gather data, analyze game activity, and use statistical models to alert management of a player's winning streak that could be due to cheating or card counting, a legal practice that casinos don't permit because, they say, it gives gamblers an unfair advantage.

The Wynn Las Vegas hotel, scheduled to open in the near future, will use RFID equipment to tell the casino's computer systems if someone is using counterfeit chips or tries to alter a chip's redemption value, explained David Sisk, the casino's senior vice president and chief financial officer.

The Hard Rock Hotel and Casino in Las Vegas is installing RFID readers and computers at game tables, with antennas located at each gaming seat, said Bart Pestrichello, the facility's vice president of casino operations.

The technology could register the wagers when the dealer closes all betting, thereby catching players who try to surreptitiously add chips if they have a good hand or remove chips if they have a bad hand.

The system could also record activity at a table for bookkeeping purposes and to detect whether dealers consistently can't reconcile the chips they have with what should be on hand, a possible sign of theft.

In addition to chips, casinos typically also must buy equipment such as RFID readers, computers, and networking gear. The Wynn says it is spending about $750,000 on the technology.

Manufacturers of RFID equipment for casinos are trying to make the technology faster. Currently, an RFID-equipped game table requires seven seconds to read 100 chips, which is too slow for fast-moving games such as baccarat, pai-gow poker, or roulette.

This fall, McMeekin noted, Progressive Gaming expects to have equipment that can read chips within 0.7 seconds. "That will satisfy any application," he said. ■

US5895321 PROGRESSIVE FR2739587
SAFECHIP BY BOURGOGNE ET GRASSET®

Casinos around the world are beginning to use radio-frequency identification technology within betting chips to improve marketing and prevent fraud. The RFID chips could track whether individual gamblers are betting enough money to qualify for perquisites, gather data for analysis to determine whether players are cheating or violating card-counting policies, or spot possible employee improprieties.

Using Technology to Save Drowning Victims

A pair of companies has developed a computer-vision-based drowning-detection system that has already saved five lives.

Vision IQ and Poseidon Technologies have developed the Poseidon system. "This is not designed to replace lifeguards but rather to assist them in recognizing a person in trouble," explained Joshua L. Brener, principal and founder of the Water Solutions marketing firm, which is in charge of US marketing for Poseidon Technologies.

According to Brener, lifeguards have difficulty seeing everything that happens in a large pool, particularly under water. According to Brener, 500 people drown in lifeguard-protected pools every year in the US alone.

Poseidon uses computer vision to recognize texture, volume, and movement within a pool. The system consists of a network of cameras that are under water in areas at least seven feet deep and overhead in shallower sections. The system does not work in water less than two feet deep. A typical pool could have four to six overhead cameras and three or four digital cameras under water.

The cameras survey the pool and gather information that they feed into a PC, to which they have been hard-wired. Two algorithms examine the camera output. One recognizes volume and can distinguish the difference between, for example, a shadow and a body. The other distinguishes textures and can thus recognize the difference between a towel and a body.

If the system recognizes something with the volume and texture of a person at the pool's bottom moving at less than two-tenths of a meter per 10 seconds—the maximum speed of a typical drowning victim, according to Vision IQ—it transmits a visible and audible alarm to lifeguards.

This enables a fast response, explained Brener, who said this is important because "a person in a moderately warm pool could live for only a few minutes before oxygen deprivation to the brain causes damage."

Poseidon has an added function that alerts lifeguards if the pool needs to be cleaned so that the visualization system can "see" properly.

The system can also record rescues, as well as activities such as fights and assaults. Said Brener, "The video can see a person go to the bottom, see the alarm, see how long the rescue took, and watch the lifeguard's technique. The video is time-tagged so there is a record of what happened."

Poseidon is installing 50 systems in the US and 90 in Europe, mostly in indoor swimming pools. ■

Cell Phone Technology Detects Gases, Odors, Even Bad Breath

A European research team is developing sensor technology so small, it could fit inside a cell phone and detect a variety of gases, including those that indicate the presence of hazardous substances or even bad breath.

Siemens' Corporate Technology Department is working on two principal gas-sensing technologies, according to Maximilian Fleischer, the department's senior principal engineer and project manager for gas sensors.

"One is the use of very small ceramic chips," he explained. "They are suspended in the surrounding air by thin wires and heated—like the glowing wire of a small bulb—to several hundred degrees Celsius." The chips carry semiconducting metal oxides that, when heated, interact with the surrounding air and indicate the presence of a target gas by measuring a change in electrical resistance.

The other gas-sensing approach uses a chip with field-effect-transistor transducers with receptor materials sensitive to gases. When the materials react to the presence of a gas, they create a small electrical voltage that activates a warning to the user. "By using a multitude of sensitive spots equipped with different sensing materials, these chips can detect and distinguish several different gases at once," Fleischer noted. He said the technologies could have a variety of applications, such as detecting natural gas leaks or ozone levels too high for safe jogging.

Fleischer said Siemens is still researching both gas-sensing approaches. "Initial tests have shown that various types of sensors are suitable for employment in mobile phones. The decision as to whether to sell mobile phones with gas sensors will not be made until there has been further progress in the research."

This would be part of a growing trend in which cell phones are used for functions in addition to making and receiving calls. ■

*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

The Future Interconnection Environment

Hai Zhuge
Chinese Academy of Sciences

Networks pervade nature, society, and virtual worlds, giving structure and function to a variety of resources and behaviors. Discovering the rules that govern the future interconnection environment is a major challenge.

In 1960, Marvin Minsky predicted that computers would be as smart as humans within three to eight years. Nearly half a century later, however, computing systems still cannot pass the Turing test. Despite impressive achievements in robotics, mathematical theorem proving, scientific classification, and advanced user interfaces,[1] artificial intelligence remains elusive.

Scientists and engineers have nearly realized Vannevar Bush's dream of a universal multimedia data-processing machine with the Internet and the World Wide Web. Extending this vision into the future, Microsoft researcher Jim Gray foresees the development of highly secure, highly available, self-programming, self-managing, and self-replicating computer networks.[2] Gray imagines a system, akin to Bush's memex device, that can automatically organize, index, digest, evaluate, and abstract information. However, creating intelligent networks that can program, manage, and replicate themselves is a major challenge.

The China Knowledge Grid Research Group (http://kg.ict.ac.cn), established in 2001, is exploring the operating principles of this future interconnection environment.

TOWARD A NEW COMPUTING ENVIRONMENT

The emergence of the Web provided an unprecedented AI research and application platform. By providing access to human-readable content stored in any computer connected to the Internet, it revolutionized business, scientific research, government, and public information services around the globe.

However, because machines cannot yet understand human-readable Web pages, the current Web cannot adequately support intelligent applications. Such applications require a new Internet application platform to intelligently accommodate the development, deployment, interaction, and management of globally distributed e-services based on open standards such as the Web Services Description Language.

Scientists are using symbolic reasoning, text mining, information extraction and retrieval, and other cutting-edge technologies to improve or extend the Web. For example, IBM's WebFountain (www.almaden.ibm.com/webfountain) converts online content such as Web pages, e-mail, message boards, and chat into an XML-based form and analyzes it to identify its commercial value. The proposed Semantic Web[3] aims to enmesh online content more meaningfully using ontological and logical mechanisms as well as standard markup languages like the Resource Description Framework.

0018-9162/05/$20.00 © 2005 IEEE

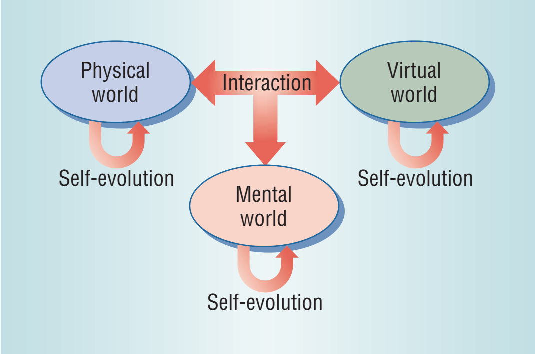


Figure 1. Future interconnection environment. The physical, virtual, and mental worlds will interact and evolve cooperatively.

In addition, the next-generation Internet2 will be hundreds of times faster and more secure than the current Internet. By providing a rich address space for advanced applications, it will push evolution of the Internet application platform, which in turn will inspire new applications.

Researchers are developing advanced functions in other types of artificial networks. For example, the Grid (www.gridforum.org) aims to share, manage, coordinate, and control distributed computing resources such as machines, networks, and data from any digital device plugged into it. The Open Grid Services Architecture (www.globus.org/ogsa) attempts to combine the Grid's advantages with those of the Semantic Web and Web services. However, OGSA is not suited to large-scale, unstable dynamic networks.

Peer-to-peer networking has emerged as a popular technology for sharing computing resources in such networks. However, while P2P networks are autonomous and scalable, they lack the required understanding, coordination, and scheduling capabilities to support advanced applications.[4]

The future interconnection environment must absorb AI and distributed systems, inherit the advantages of the Web, Semantic Web, Grid, and P2P technologies, and go beyond their scope with new principles.

PRINCIPLES, PARAMETERS, AND CHALLENGES

The computing environment has evolved from personal or centralized computers to distributed networks to human-computer environments. As Figure 1 shows, the future interconnection environment will be a large-scale human-machine environment that unites three worlds:

- *physical world*—nature, natural and artificial materials, physical devices, and networks;
- *virtual world*—the perceptual environment constructed mainly through vision (text, images, color, graphs, and so on) and hearing, and to some extent touch, smell, and taste; and
- *mental world*—ideals, religions, morals, culture, arts, wisdom, and scientific knowledge, which all spring from thought, emotion, creativity, and imagination.[5]

Ideally, this environment will be an autonomous, living, sustainable, and intelligent system within which society and nature evolve cooperatively. It will gather and organize resources into semantically rich forms that both machines and people can easily use. Geographically dispersed users will cooperatively accomplish tasks and solve problems by using the network to actively promote the flow of material, energy, techniques, information, knowledge, and services in this environment.

Principles

The future interconnection environment will evolve under the principles of openness, incremental development, economy, ecology, competition and cooperation, dynamic scalability, integrity, and simplicity.

Openness. Making the environment open prevents stagnation. Standards are essential for open systems and must be continually updated as the environment evolves.

Incremental development. The environment will move from a small, simple scale to a large, complex one, perhaps exponentially. The number and skills of developers will likewise increase. From the applications perspective, development should balance inheritance and innovation. Smooth upgrading of the work environment and paradigms will ensure effective use of new technologies.

Economy. Benefits to participants, resources, and the environment should be distributed reasonably and fairly. The market forces participants to reasonably adjust their decisions and behaviors—both producers and consumers look for satisfaction rather than maximization because they must come to agreement. This simple mechanism avoids complex computation.

Ecology. The future interconnection environment will foster a complex ecology that explores interactions between the natural world, the virtual world, and human society.

Competition and cooperation. Resources in the environment must compete for survival, rights, and reputation. At the same time, they should cooperate with and regulate one another to support the function and value of the services they use to compete.

Dynamic scalability. Participants and resources must be able to join or leave the environment without affecting its overall function. The network and its relational and organizational layers should support this dynamic scalability.

Integrity and simplicity. The environment's beauty lies in the integrity and simplicity of the underlying structures of itself, individuals, species, and society.

Parameters

The future interconnection environment will be a sustainable and harmonious system in terms of

- *space*—the capacity to encompass a great variety of individual and shared resources including material objects, information, knowledge, services, and physical space in the natural environment;
- *time*—the processes of evolution and degeneration;
- *structure*—the environment and resources in it;
- *relation*—the relationships between and among processes and resources; and
- *worth*—the status of and prospects for resources, processes, and their relationships.

Einstein's general theory of relativity reveals that space and time are malleable entities in the physical world. On the largest scale, space is dynamic, expanding or contracting over time.

The future interconnection environment will foster the growth of knowledge, a type of resource, by supporting social activities at different levels—from the physical level to the human-machine community level—and in different disciplines. As a natural product of society, knowledge will evolve and endure throughout the life of the human race rather than the life of any individual.

Human social activities have thus far largely relied on natural-language semantics. Future social activities will instead depend on a new kind of semantics that establishes an understanding between humans and inanimate resources. This human-machine semantics will make it possible to beneficially use and safely regulate services and knowledge.

Challenges

The future interconnection environment's variety and complexity will limit the ability of a single theory to support modeling. Gaining the insights needed to resolve a number of major challenges requires going beyond traditional disciplinary boundaries.

Reorganization of versatile resources. Accurate and complete resource management requires an organized approach. The relational model ensures successful database management but is unsuitable for managing complex and semantically rich resources in a dynamic environment. A new theory is needed for organizing resources in semantically rich forms and using them under integrity constraints. The Internet2's advanced characteristics make such a theory feasible.

Reconciling normalization and self-organization. Normalization reflects stability and order, while self-organization reflects dynamic order in unstructured phenomena. The "small world" phenomenon shows a kind of stability within a scale-free network.[6] The normalization of resource organization ensures accuracy in their operations; self-organization ensures autonomy, equality, and adaptability in managing resources. One way to reconcile normalization and self-organization is to impose normalized structure at the higher levels, allow self-organization at the lower levels, and maintain mapping and consistency between levels.

The future interconnection environment will be a sustainable and harmonious system.

Semantic interconnection. Consistently connecting various resources in many semantic layers to support intelligent applications is a challenge. The key is to construct a computing model that applies to explicit semantics as well as tacit semantics relating to sensation and emotion. The "sense and sensibility" of autonomous resources also play an important role in semantic interconnection.[7]

Clustering and fusing. Intelligent services rely on the ability to cluster and recluster heterogeneous resources. Because current passive resource models do not support active clustering, establishing an intelligent resource model is necessary. Fusing could occur among entities or among content.

Network degeneration. Researchers have extensively studied the Web's growth and distribution[8] but have largely neglected degeneration. In the real world, however, development of anything eventually reaches a limit. It is important to determine how degeneration might impact or limit evolution of the future interconnection environment.

Abstract flow modeling. Finding rules and principles common to material, information, knowledge, and service flows and discovering their logistics is another big challenge. Meeting this challenge requires an abstract process model with optimization and control methods.

Field theory. In the future interconnection environment, as in the physical world, resources will flow from high- to low-energy nodes. This automatically requires appropriate on-demand logistics because the energy difference reflects a need for flow. However, the real-world law of energy conservation does not hold: Copying or generating data does not incur a physical cost, nor does deleting data. The basic laws and principles governing this special field require much more investigation.

Abstracting resources. Abstraction is the basis of understanding, thinking, and problem solving. It is a challenge to automatically capture semantics

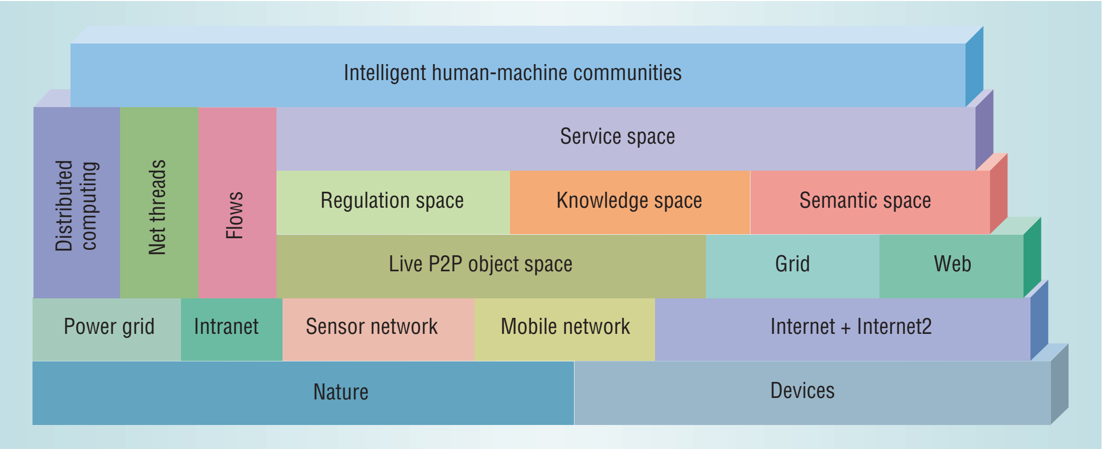


Figure 2. Future interconnection network reference architecture. Nature provides the most direct inspiration for ways to improve technology.

from resources and to reason and explain in a uniform semantic space. The environment needs a single semantic image mechanism[9] that establishes a common understanding for various resources. The mechanism's constraints and rules ensure valid resource usage in the semantic space.

Ecology. As elsewhere, resources in the future interconnection environment can be classed into species. Inheriting from existing species is the major way to form a new resource.[5] The evolution of species depends on flows between specimens. The challenge is to apply the methods and principles of natural ecology to help understand and explore the future interconnection environment ecology.

Dynamic inheritance. A common phenomenon in the organic world, inheritance is also the key mechanism for supporting reuse in object-oriented methodology and systems. How to realize the inheritance mechanism among evolving resources in an evolving environment is another challenge. Inheritance should accord with ecological and biological principles.

Biointerface. Sensor networks are gradually making the Internet ubiquitous. Scientists are embedding sensors into animals' bodies to integrate biological and electrical data, and they are using the human body to provide energy for computing and as part of a network. These sensors will be an integral part of the future interconnection environment. However, a huge gap exists between low-level information collected from sensors and high-level information that could be automatically understood and intelligently processed.

Organic architecture. A truly dynamic network should have organic characteristics such as self-protection, self-healing, fault tolerance, dynamic adaptation, self-replication, self-motivation, and self-fueling. This requires developing a system architecture analogous to anatomical structures—including, for example, immune, nervous, digestive, and circulatory systems.

Methodology. A large-scale, dynamic, and intelligent interconnection environment will require a multidisciplinary system methodology and an epistemology for guiding the development, operation, and maintenance of the network and its applications.

ARCHITECTURE AND INTERCONNECTION RULES

Applying the incremental development principle to the future interconnection network yields the layered reference architecture in Figure 2. The bottom layer is an interface between the physical world and the virtual world. Scientists use sensor networks to collect information and various devices to control small-scale natural environments—for example, to make rain.[10] Nature most directly inspires technological improvements, as evidenced by genetic computing, neuronal computing, swarm intelligence, and biomolecular computing.

The *live P2P object space* contains abstract representations of various environmental resources, and each object within it has a life span. *Net threads* carry running applications. Objects in the *regulation space* manage resources autonomously. The *knowledge* and *semantic spaces* overlay the live P2P object space and support the *service space*. Services can find requirements advertised by roles and resources. People in *intelligent human-machine communities* work, entertain, contribute, and enjoy services according to regulations and social principles. A person's role can move from one community to another through *flows* of material, information, knowledge, and services that link communities and exploit computing resources.

Object space growth

The live P2P object space is a relational network with live resource nodes connected by semantic links. The network is said to be alive because every node has a life span that lasts from "birth" (addition to the network) to "death" (removal from the network). Rules that govern network growth must consider the addition and removal of both nodes and links.

As a case study, investigators in the China Knowledge Grid Research Group compared the Web's hyperlink distribution with two models of the link distribution of the live semantic network—a stochastic growth model and a directed evolving

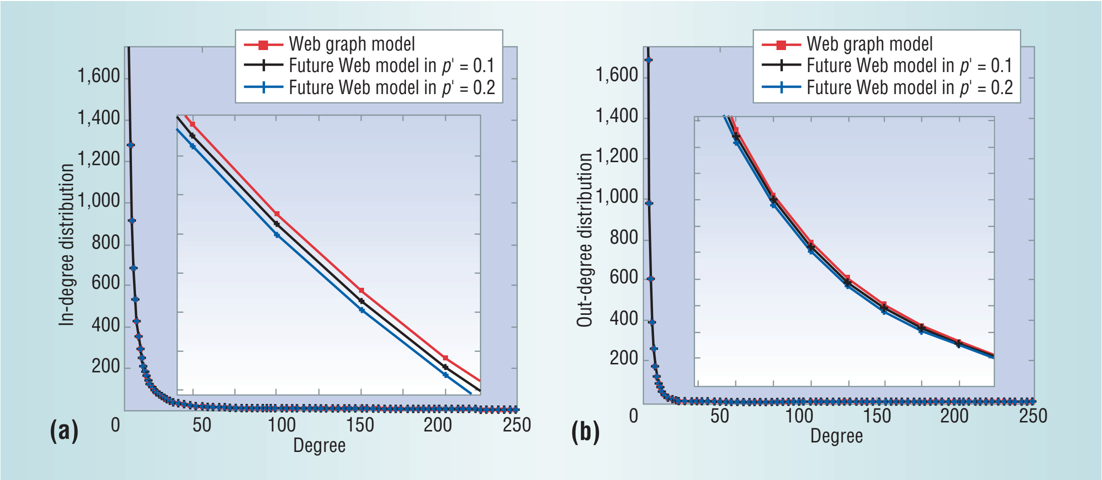


Figure 3. Live P2P object space growth. (a) In-degree and (b) out-degree distribution of live semantic network links using a stochastic growth model and a directed evolving graph model compared with Web data.

graph model—and obtained the same scale-free distribution rule from each.

Figure 3 shows the in- and out-degree distribution, respectively, of data obtained using the two models along with the Web data.[11] The models' loss of links and nodes accounts for the gap in results: the faster the removal, the larger the disparity. Further, the magnitude of the curve's slope for the investigated models is always smaller than that of the Web graph model. The number of links indicates a node's wealth: Rich nodes have more links than poor ones. Preferential attachment leads to a "rich get richer" phenomenon that increases the gap between rich and poor nodes.

Damping effect

Experience indicates that rich nodes will last longer than poor ones, but a cap on wealth tends to average out life spans. A simulation that blocked nodes from acquiring further in-links once they reached a certain level resulted in the distribution shown in Figure 4a, which is no longer a power law. Instead, the tail rises a little near the limit, suggesting that relatively rich nodes shared the blocked wealth.[11]

This *damping effect* also exists in many real-world networks, causing them to move from prosperity to degeneration. For example, in epidemic dissemination networks, nodes join when they become infected and leave when they recover or die. Figure 4b shows the damping effect of antibody development and community self-protection measures on the spread of the severe acute respiratory syndrome (SARS) epidemic in China in 2003.

In the future interconnection environment, resources will likewise compete under a damping effect. Because rich nodes emit more information, knowledge, and services, poor nodes will find it easier to get rich. Some social and natural rules will also apply to the interconnection environment.

Compression and expansion

Figure 5 presents a time-space model of the future interconnection environment. Compression and expansion pull and push development like the ebb and flow of tides. Compression intensifies competition among technologies (for example, the Internet, the Internet2, the Web, the Grid, and P2P networking), pushing some out and helping to generate new ones (such as the Semantic Grid and the Knowledge Grid), which leads to expansion.

The extent of expansion and compression influences sustainability. Achievements in sustainable development of the natural ecosystem provide insight into the future interconnection environment's sustainability.[12]

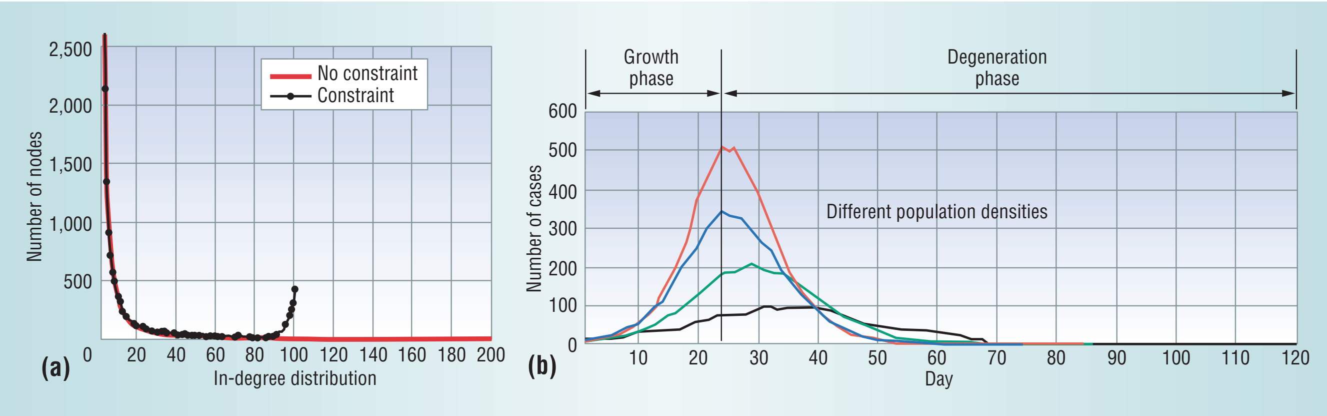


Figure 4. Damping effect. (a) In-degree link distribution under constraint. (b) SARS epidemic dissemination network.

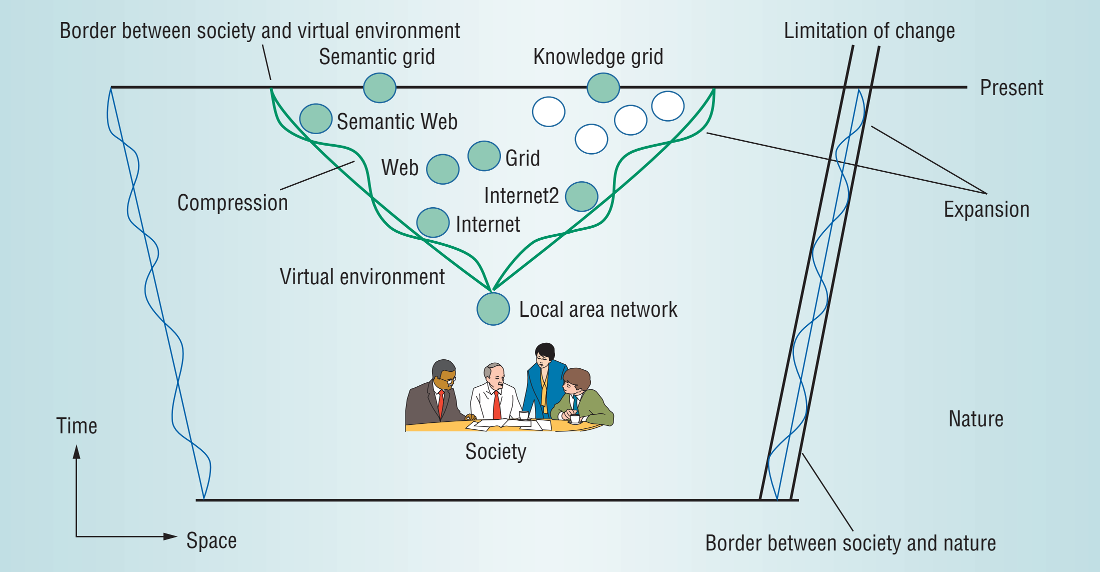


Figure 5. Time-space model. Compression and expansion pull and push development of the future interconnection environment like the ebb and flow of tides.

Various resources compete with one another in the interconnection environment for survival.[5] Some become dominant in the competition, as in the Web's "rich get richer" phenomenon.[8]

Compression and expansion change both the space and self-organization behavior. Social expansion and compression influences virtual environment development, which eventually will fuse with and keep pace with social development.

The border between society and the virtual environment will evolve from screens and keyboards to various mobile devices, sensors, and biointerfaces.

EVOLVING E-SCIENCE ENVIRONMENT

The China Knowledge Grid Research Group is developing the e-Science Knowledge Grid Environment as an experimental microcosm of the future interconnection environment.[9] This evolving, dynamic, self-organizing, self-managing, and scalable system is designed to support the development of diverse distributed and intelligent information, knowledge, and computing services.

The architecture, shown in Figure 6, includes five main components:

- *Imagine-Framework*—the core component that supports overall system development. This basic application development framework organizes and manages resources in a decentralized and autonomous way on a P2P network. It contains a class library, algorithm library, and template library for developing and managing high-level applications.
- *Imagine-Run*—an embedded platform that will support runtime management of the underlying P2P network and high-level applications, including network maintenance, application deployment, configuration, and execution control.
- *Imagine-Research*—a virtual network laboratory that will support monitoring, debugging, configuring, testing, and verification to hasten evolution of the environment. Researchers and users will be able to fully interact with one another to form a positive feedback cycle of requirements and technologies.
- *Imagine-Builder*—a platform that will include tools, source code, and virtual components to enhance development of distributed domain applications for large-scale networks.
- *EcoLab*—a virtual scientific research laboratory that geographically dispersed ecologists will use to efficiently publish, share, manage, and exploit distributed resources including computing power, data, information, and knowledge on a large P2P network. It will feed users' requirements back, thereby helping to improve both platforms and domain applications.

After developing and deploying Imagine-Run and Imagine-Research, we will use these as platforms to test and improve various technologies and software, and to extend the Imagine-Framework to different types of large-scale dynamic networks. Platform researchers, domain application developers, and end users will use Imagine-Framework, Imagine-Research, and Imagine-Run, respectively, in a cooperative way to improve the overall environment.

The China Knowledge Grid Research Group continues to look at ways to realize the ideal of the future interconnection environment. Ongoing work includes exploration of interconnection semantics, investigation of advanced high-level mechanisms such as dynamic soft-device inheritance, and application of research results in the development of systems for e-science and e-culture as well as supporting interscientific and intercultural research. ■

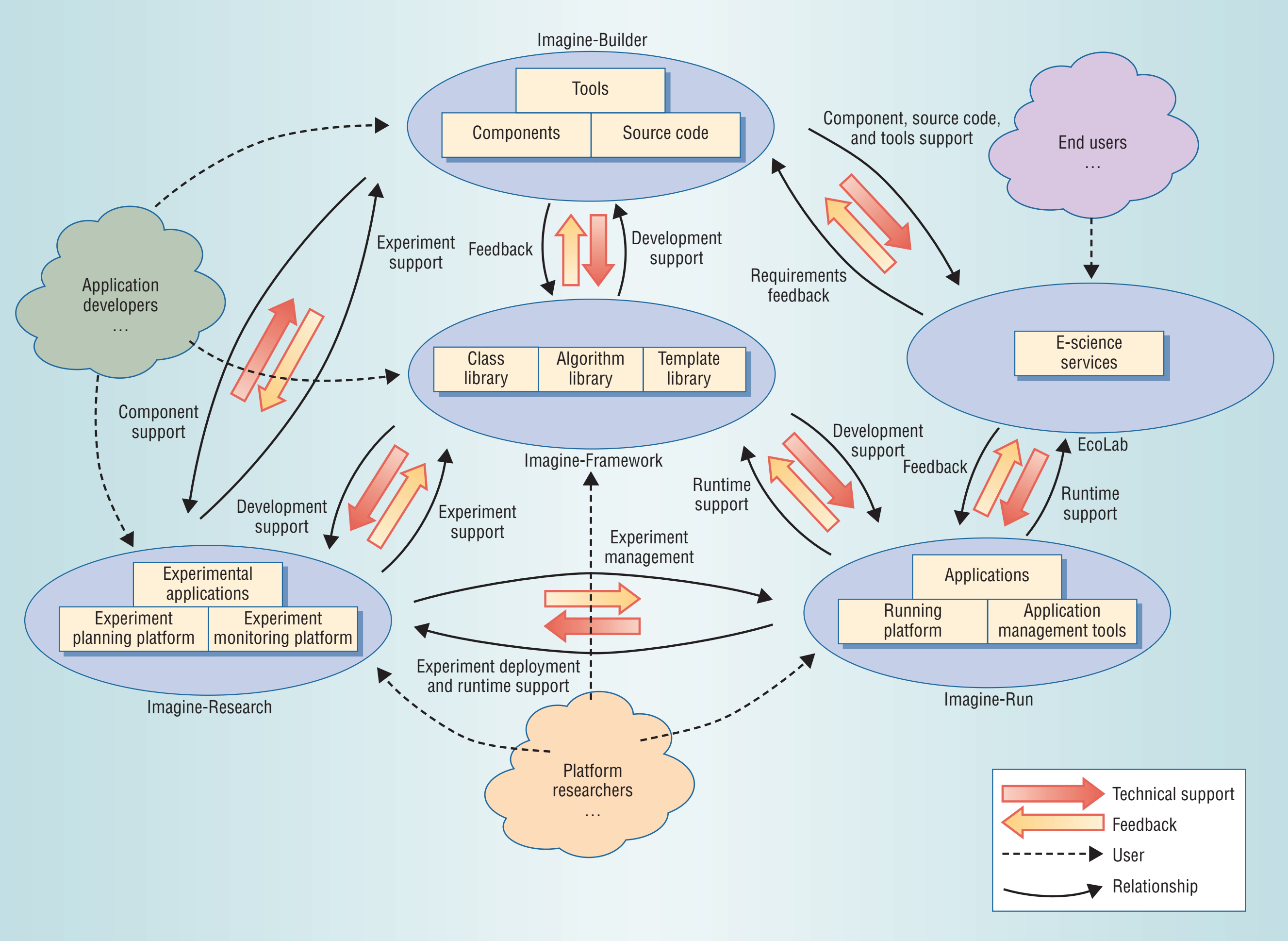


Figure 6. China's e-Science Knowledge Grid Environment. This evolving, self-organizing, self-managing, and scalable system is designed to support development of distributed ecological applications on a P2P network.

Acknowledgments

This article is based on research work supported by the National Basic Research and Development 973 Program (2003CB317001) and the National Science Foundation of China under grants 60273020 and 70271007.

References

1. D.L. Waltz, "Artificial Intelligence: Realizing the Ultimate Promises of Computing," *AI Magazine*, vol. 18, no. 3, 1997, pp. 49-52.
2. J. Gray, "What Next? A Dozen Information-Technology Research Goals," *J. ACM*, vol. 50, no. 1, 2003, pp. 41-57.
3. T. Berners-Lee, J. Hendler, and O. Lassila, "Semantic Web," *Scientific American*, vol. 284, no. 5, 2001, pp. 34-43.
4. H. Balakrishnan et al., "Looking Up Data in P2P Systems," *Comm. ACM*, vol. 46, no. 2, 2003, pp. 43-48.
5. H. Zhuge and X. Shi, "Eco-Grid: A Harmoniously Evolved Interconnection Environment," *Comm. ACM*, vol. 47, no. 9, 2004, pp. 79-83.
6. J. Kleinberg, "Navigation in a Small World," *Nature*, vol. 406, 2000, p. 845.
7. J. Kumagai and S. Cherry, "Sensors and Sensibility," *IEEE Spectrum*, July 2004, pp. 18-24.
8. L.A. Adamic and B.A. Huberman, "Power-Law Distribution of the World Wide Web," *Science*, vol. 287, no. 24, 2000, p. 2115.
9. H. Zhuge, "China's E-Science Knowledge Grid Environment," *IEEE Intelligent Systems*, vol. 19, no. 1, 2004, pp. 13-17.
10. S.K. Moore, "Electronic Rainmaking Technology Gets Mexico's Blessing," *IEEE Spectrum*, Apr. 2004, pp. 14-15.
11. H. Zhuge, *The Knowledge Grid*, World Scientific, 2004.
12. A.J. McMichael, C.D. Butler, and C. Folke, "New Visions for Addressing Sustainability," *Science*, vol. 302, no. 5652, 2003, pp. 1919-1920.

Hai Zhuge *is a professor and director of the Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, and founder of the China Knowledge Grid Research Group. His research interests include the theory and methodology of the future interconnection environment and applications in China. Zhuge received a PhD in computer engineering from Zhejiang University, China. He is a senior member of the IEEE and a member of the ACM. Contact him at zhuge@ict.ac.cn.*

Overcoming the Internet Impasse through Virtualization

Most current Internet research involves either empirical measurement studies or incremental modifications that can be deployed without major architectural changes. Easy access to virtual testbeds could foster a renaissance in applied architectural research that extends beyond these incrementally deployable designs.

Thomas Anderson
University of Washington

Larry Peterson
Princeton University

Scott Shenker
UC Berkeley

Jonathan Turner
Washington University

The Internet's stunning success has changed the way we work, play, and learn. The Internet architecture, developed over 30 years ago, has proven its worth by the vast array of applications it now supports and the wide variety of network technologies over which it currently runs. Nonetheless, the Internet's increasing ubiquity and centrality has brought with it a number of challenges for which the current architecture is ill-suited. Although developers and researchers have shown increasing interest in new architectures that could address these challenges,[1-8] the prospects for significant change in its existing architecture appear slim. In addition to requiring changes in routers and host software, the Internet's multiprovider nature also requires that ISPs jointly agree on any architectural change.

The need for consensus is doubly damning: Not only is reaching agreement among the many providers difficult to achieve, attempting to do so also removes any competitive advantage from architectural innovation.

Short of the Internet's imminent collapse, there seems little hope for major architectural changes—those innovations that would alter its basic architecture. Worse, the situation continues to deteriorate.

The inability to adapt to new pressures and requirements has led to an increasing number of ad hoc workarounds, many of which violate the Internet's canonical architecture. While derided by architectural purists, these modifications have usually arisen to meet legitimate needs that the architecture itself could not. These architectural barnacles—unsightly outcroppings that have affixed themselves to an unmoving architecture—can serve a valuable short-term purpose, but they significantly impair the Internet's long-term flexibility, reliability, and manageability.

The daunting barriers to deployment of new architectures, while discouraging, do not directly hinder further research. Architectural invention continues without limitations, even if without hope of adoption. However, live experimentation with new architectures has proven more problematic. The main avenue for live experimentation, as opposed to simulation or emulation, is to use testbeds.

However, traditional testbeds have severe limitations that constrain our ability to evaluate new architectures.[9] Instead of being satisfied with paper designs that have no future, the design community should return to its roots of applied architectural research with the intention of once again changing the world.

THREE REQUIREMENTS

Overcoming the current impasse will not be easy and will require addressing three separate requirements:

- Researchers must be able to experiment easily with new architectures on live traffic.

Published by the IEEE Computer Society

- There must be a plausible deployment path for putting validated architectural ideas into practice.
- Instead of focusing on a single narrow problem, the proposed solutions should be comprehensive so that they can address the broad range of current architectural problems facing the Internet.

We propose to meet these three requirements by constructing a virtual testbed that will support multiple simultaneous architectures, serving all the communication needs of standard clients and servers. This virtual testbed approach provides a clean path for unilaterally and globally deploying new architectures. Because it does not require universal architectural agreement, this approach offers a more plausible deployment scenario for radical new designs that systematically tackle the complete set of problems facing the Internet today.

Central to our proposal is the concept that *virtualization*—as used in virtual memory, virtual machines, and elsewhere—is nothing more than a high-level abstraction that hides the underlying implementation details. With virtualization, nodes can treat an overlay as if it is the native network, and multiple overlays can simultaneously use the same underlying overlay infrastructure. Both aspects of virtualization are crucial to our virtual testbed proposal.

PHYSICAL TESTBEDS AND OVERLAYS

Before they can even consider deployment of a proposed architecture, researchers must adequately evaluate it. Although simulation and emulation are valuable tools for understanding new designs, they cannot substitute for experimentation with live traffic.

Preparing an implementation to deal with the real world forces designers to confront the many unpleasant realities that paper designs frequently avoid, such as multiple providers, legacy networks, anomalous failures and traffic conditions, and unexpected and diverse application requirements. Moreover, live traffic provides a fuller picture of how an architecture will perform, strengthening the case that the architecture will actually provide the claimed benefit.

Currently, researchers use physical testbeds and overlays to experiment with new architectures. Overlays have also found favor as a valid deployment path. Both of these approaches, however, have limitations.

Physical testbeds

The traditional platform for live experimentation, physical testbeds consist of leased lines connecting a limited set of locations. Testbeds can be roughly categorized as production- or research-oriented.

Traditional testbeds offer too little bang for the buck and clearly cannot lead us into the future.

Production testbeds, such as Internet2, support real traffic from real users, often in large volume and across many sites. As such, they provide valuable information about an architecture's operational behavior. However, a production testbed's users have no choice about participating in the testbed and usually don't even realize their traffic has become part of an experiment. They thus expect the performance and reliability to be no worse than the standard Internet. Production testbeds must therefore be extremely conservative in their experimentation, using well-honed implementations of incremental changes.

Research testbeds such as DETER (Defense Technology Experimental Research) do not carry traffic from a wide variety of real users. Instead, they are typically driven by synthetically generated traffic, a small collection of intrepid users, or both. Thus, they are more adventurous and capable of running first-cut implementations of radically new designs.

Unfortunately, this lack of real traffic also means that the results are less likely to be indicative of real operational viability. As a result, neither a production nor a research testbed can produce the data needed to adequately evaluate new architectures.

Further, because they utilize dedicated transmission links, both testbed categories involve substantial cost, which makes operating them on a large scale prohibitively expensive. This typically limits their use to a small geographic area and even then requires substantial funding support.

These factors make it difficult to build a compelling case for new architectural designs based on a testbed evaluation. Given their limitations, traditional testbeds offer too little bang for the buck and clearly cannot lead us into the future.

Overlays

Becoming more widespread recently, overlays are being used both as an experimental platform and a deployment path.[10-12] They are not limited geographically and their usage is voluntary. Moreover, overlays typically do not involve significant expenditures, thus avoiding many of the problems that plague traditional testbeds. With the advent of PlanetLab[13]—an open platform for developing, The virtual testbed approach provides a focal point for a new attitude toward overlays.

deploying, and accessing planetary-scale services—creating and maintaining an overlay has become a straightforward task. However, overlays still suffer from limitations of their own.

First, overlays have largely been seen as a way to deploy narrow fixes to specific problems in the Internet architecture, whether for performance,[10] availability,[11] denial of service,[11,14] content distribution, or multicast.[15] Researchers have viewed the solution to each of these problems as an isolated function, and they have done little to determine how any of the solutions might work together. More importantly, they have devoted little thought to identifying how a set of overlays might ultimately replace the underlying Internet architecture.

Second, to date, overlays have been architecturally tame. Because the emphasis has been on deployment in today's Internet rather than on architectural innovation leading to tomorrow's Internet, most current overlays typically assume IP or a close cousin as the architecture inside the overlay itself: the interoverlay node protocol. As such, overlays have not been the source of dramatic architectural advancement.

Thus, on their current trajectory, overlays will likely become just a better way of attaching yet another barnacle, rather than an agent of fundamental change. The field needs a philosophical revolution in how developers use overlays, not a technical alteration in how they build them. Therefore, the virtual testbed approach that we propose provides a focal point for a new attitude toward overlays rather than a technical advancement.

VIRTUAL TESTBED

To address these problems and provide an attractive platform for experimentation and possible deployment, we propose a *virtual testbed* approach. Virtual testbeds have two basic components: an overlay substrate and a client-proxy mechanism.

Key features

An *overlay substrate* provides a set of dedicated but multiplexed overlay nodes. By multiplexing these nodes, as first advocated in PlanetLab, multiple experiments can run simultaneously on the same infrastructure. The effort of instantiating and maintaining the overlay is amortized across the many concurrently running experiments, drastically lowering the barrier to entry that an individual researcher faces.

A host can use the *client-proxy mechanism* to opt in to a particular experiment running on a specific substrate overlay. This mechanism treats a nearby overlay node as the host's first-hop router without imposing any limitations on the experimental architecture. It also supports opt-in at a fine granularity by, for example, routing local traffic directly or determining participation on a per-application basis. These two features resolve the barrier-to-entry and architectural limitations that overlays faced.

To encourage the use of overlays for more radical architectures, we have deployed a prototype of this approach on PlanetLab. It is relatively primitive in its original incarnation. PlanetLab currently includes more than 529 nodes that span 252 sites and 28 countries on five continents.

Technology overview

We estimate that a PlanetLab node is within a LAN hop of more than one million users. As the "PlanetLab Computing Platform" sidebar describes, PlanetLab software architecture multiplexes multiple slices, each running a different network service, application, or architecture. Users can view each slice as a set of virtual routers connected by tunnels to whatever topology the architecture selects.

Mostly, PlanetLab leverages straightforward technologies, but we still have some issues to explore. For example, achieving sufficiently high throughput rates on PlanetLab nodes is challenging: Stock PlanetLab nodes can forward packets at 60 Mbps. While we expect to achieve gigabit rates with modest optimizations, PlanetLab nodes clearly cannot compete with custom hardware.

Similarly, an overlay's virtual links cannot compete with dedicated links. In cases where timeliness is crucial, an overlay could use techniques such as those incorporated in OverQoS[16] MPLS paths to provide better service than a naïve tunnel over IP.

Moderately developed, the proxy technology still needs work. Our prototype proxy can catch and forward packets into the virtual testbed from interposed proxies on any IP address or port that the legacy client software identifies. Given that most client applications use name translation as the first step in communication, the proxy interposes on DNS requests and either returns the server's true IP address if the packets are for the normal Internet or a fake IP address if the packets are for the virtual testbed.

By interposing on the fake IP addresses, the proxy can then forward the packets to the nearest virtual testbed node, the ingress node. The proxy is

PlanetLab Computing Platform

PlanetLab is a geographically distributed computing platform for deploying, evaluating, and accessing planetary-scale network services. PlanetLab is a shared community effort by researchers at 252 sites in 28 countries, each of whom gets access to one or more isolated "slices" of PlanetLab's global resources via a *distributed virtualization* concept.

To encourage infrastructure innovation, PlanetLab's *unbundled management* principle decouples the operating system running on each node from a set of multiple, possibly third-party, network-wide services that define PlanetLab.[1] PlanetLab services and applications run in a slice of the platform: a set of nodes on which the service receives a fraction of each node's resources in the form of a virtual machine.

What's new in PlanetLab is distributed virtualization: the acquisition of a distributed set of VMs that the system treats as a single, compound entity. PlanetLab isolates services and applications from one another, thereby maintaining the illusion that each service runs on a distributed set of private machines. The platform must deliver isolation of slivers—one constituent VM of a slice running on a single node—by allocating and scheduling node resources, partitioning or contextualizing system namespaces, and enforcing stability and security between slivers sharing a node. The actual contents of a sliver within the VM are of little concern to the platform; for example, it should not matter to the platform whether the code in the sliver is running in a Java VM or written in assembly language.[1]

Figure A illustrates the PlanetLab node architecture. At the lowest level, each PlanetLab node runs a virtual machine monitor that implements and isolates virtual machines. The VMM also defines the API that implements the services.

PlanetLab version 3.0 currently implements the VMM as a combination of the Linux 2.6 kernel and a set of kernel extensions—in particular, vservers 1.9, a Linux patch that provides multiple, independently managed virtual servers running on a single machine and the SILK (Scout in Linux Kernel) module that provides CPU scheduling, network accounting, and safe raw sockets.[2,3]

The node manager, a privileged root VM running on top of the VMM, monitors and manages all the VMs on the node. Generally speaking, the node manager enforces policies on creating VMs and allocating resources to them, with services interacting with the node manager to create new VMs rather than directly calling the VMM. Moreover, all interactions with the node manager are local: Only services running in another VM on the node are allowed to call the node manager, meaning that remote access to a specific node manager is always indirect through one of the services running on the node.

Currently, most policy is hard-coded into the node manager, but we expect that local administrators will eventually be able to configure the policies on their own nodes. This is the purpose of the local administrator VM shown in Figure A.[2]

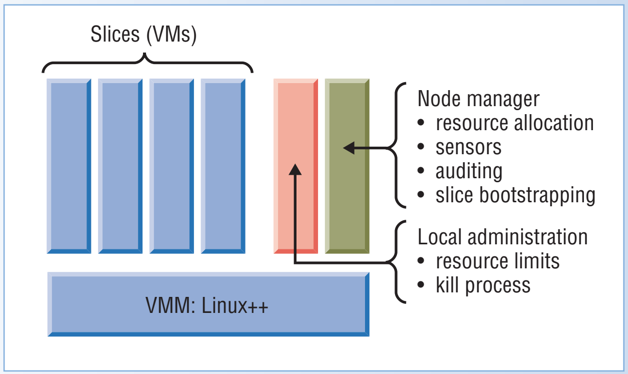


Figure A. PlanetLab node architecture. Each node runs a virtual machine monitor that implements and isolates virtual machines that the system treats as a single entity. Isolating services and applications from one another maintains the illusion that each service runs on a distributed set of private machines.

Example applications and services running on PlanetLab include network measurement, application-level multicast, distributed hash tables, storage schemas, resource allocation services, distributed query processing, content distribution networks, management and monitoring services, overlay networks, router design experiments, and federated testbeds, among others.[4]

References

1. L. Peterson and T. Roscoe, "The Design Principles of Planet Lab," PDN-04-021; www.planet-lab.org/PDN/PDN-04-021/.
2. A. Bavier et al., "Operating System Support for Planetary-Scale Network Services," *Proc. 1st Symp. Networked Systems Design and Implementation* (NSDI), Usenix, 2004, pp. 253-266.
3. PlanetLab v3.0, PDN-04-023; www.planet-lab.org/pdn.
4. T. Roscoe, "What Are People Doing in/on/with/around PlanetLab?"; http://www.planet-lab.org/Talks/2003-04-15-HP-PlanetLab-Users.pdf.

designed to do this in as architecturally neutral a way as possible. The virtual testbed can then do whatever it wants with the packets, using the IP or non-IP protocols it deems appropriate to service the packet, then tunneling over protocols it hopes to replace. Because gaining real users requires providing access to legacy servers, the node on the far end of the virtual testbed—the egress node—reconverts the packet into Internet format for delivery to the server. The egress node behaves as a network address translator, manipulating the source address to ensure that reply packets also enter the virtual testbed.

Service hosting

PlanetLab also can easily host a service within the virtual testbed that remains visible to nonparticipating clients. In this case, the virtual testbed provides DNS resolution to point the client to a nearby virtual testbed representative, in much the same way

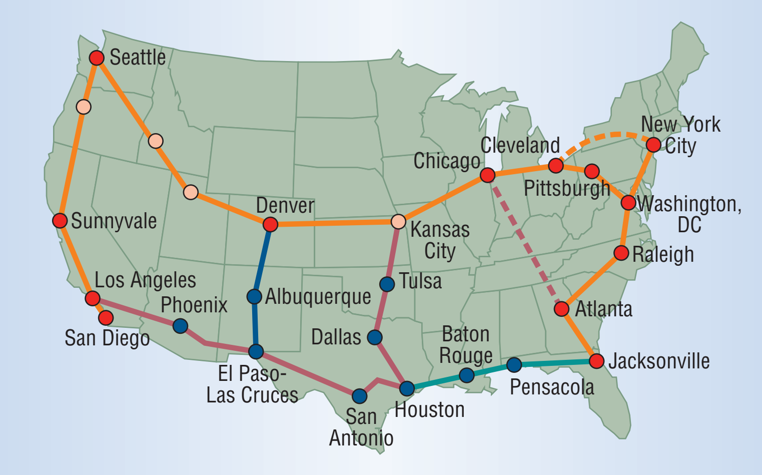


Figure 1. National LambdaRail. Using this backbone, the virtual testbed can support larger traffic volumes, with PlanetLab nodes aggregating traffic from local sites and feeding it to the backbone nodes.

that content delivery networks operate. The local representative can then translate the packets into an internal format for delivery to the server and translate the packets back to Internet format for the reply. In addition, developers can use this approach to point to multiple virtual testbeds.

Some security issues must still be resolved, particularly about how to respect server address-based policy restrictions when the overlay shields the source's IP address.

Quality of service

One drawback of the virtual overlay approach is that it cannot control the quality of service for packets traversing the virtual testbed. This limits the extent to which virtual testbeds or any overlay can test architectures for QoS. We do not consider this a fatal flaw, however, because an architecture deployed on a virtual testbed would still deliver relative QoS, as good a service as possible given the underlying link characteristics, even if it could not maintain the absolute QoS of a dedicated link in all cases.

Moreover, simulation and emulation can effectively evaluate QoS. Further, the enormous amount of literature on QoS in the past decade has made it the least mysterious aspect of new architectures. Many other issues that involve routing and addressing warrant more urgent attention and better suit the virtual testbed approach.

Inspiration

The virtual testbed borrows heavily from the ideas of the X-Bone[12] and the virtual Internet,[17] but we have a different emphasis. Because the X-Bone supports automated establishment and management of overlays, individual experiments running on the virtual testbed could use this suite of tools. The virtual testbed focus centers on virtualizing the overlay nodes themselves to support multiple simultaneous and potentially radically different architectures running on the same hardware. Although the X-Bone architecture supports this, it is not the major focus. The virtual Internet architecture,[17] based in part on the X-Bone work, allows multiple levels of virtualization. However, it remains closely tied to the current Internet architecture, which makes it unsuitable for experimenting with radical deviations from it.

Beyond this initial prototype, our future plans include a high-performance backbone, built using dedicated MPLS tunnels on Internet2, and then around a set of scalable substrate routers and links provided through the National LambdaRail (NLR), shown in Figure 1. With this backbone, the testbed will support larger traffic volumes, with PlanetLab nodes aggregating traffic from local sites and feeding it to the backbone nodes, while also enabling higher-bandwidth applications at sites close to backbone nodes. This hybrid approach captures the benefits of traditional testbeds without inheriting their flaws.

Fully utilizing the NLR backbone likely requires routers that also support virtualization. This can be accomplished at sufficient speeds using a pool of processing engines interconnected through a high-speed switch. We envision that most processing elements will include a network processor system capable of high-performance packet processing. A general-purpose processor will provide control functions, offer storage services, and facilitate migration from lower-performance sequential software designs to the parallelized designs needed to fully exploit network processor architectures.

Current-generation network processors provide enough processing resources to deliver approximately 3 to 5 Gbps of throughput for moderately complex applications. Thus, a backbone node capable of supporting 50 Gbps of throughput—three backbone links at 10 Gbps each, plus 20 Gbps of access bandwidth—will require 10 to 16 such processing engines. These engines could provide even higher performance by incorporating advanced field-programmable gate arrays that combine reconfigurable hardware and multiple processor cores in a single device.[18]

Our plan to integrate a high-speed backbone with PlanetLab has two major advantages over other purely physical testbeds. First, PlanetLab-based overlays serve as an access network for the backbone, bringing traffic from a large user community onto the backbone. Second, developing and deploying the hardware does not gate the architectural work. Researchers can first experiment with their architecture as an overlay and then later expand it to include the high-speed backbone as the platform supports it.

DEPLOYMENT

The traditional but now discredited deployment story predicted that, after having been validated on a traditional testbed, a next-generation architecture would, through some magical process of consensus and daring, be simultaneously adopted by ISPs and router vendors alike.

With this story no longer even remotely possible, can we find a *plausible* deployment alternative? We use the term plausible because adopting new technologies is an unpredictable process that confounds the expectations of even the most informed observers. Thus, we don't need to know precisely how, and certainly not which, new architectures developers might adopt. We require only that deployment be at least remotely possible.

Our deployment strategy leverages the strength of overlays, unconstrained by their previously limited ambitions. In this scenario, a *new-generation service provider* chooses a particular new architecture, then constructs an overlay supporting that architecture. The NGSP then distributes proxy software that lets anyone, anywhere, access its overlay. Those NGSP users not directly connected would still be purchasing Internet service from their ISP, but if the overlay is successful, either the NGSP would begin offering direct access to customers or current ISPs, seeing a viable competitive threat, would begin to support this new architecture.

Although we call this an overlay, the NGSP could easily support the new architecture natively on most of its network, so only the first-hop access for users not directly connected would use the architecture in overlay mode. Thus, developers could still deploy architectures that promised enhanced QoS this way.

This approach differs little from the normal overlay deployment story, except with regard to the proxy mechanism's non-IP-centric nature. Overlays offer an opportunity to radically change the architecture, not merely provide limited enhancements. A single daring NGSP could accomplish this. It might also arise more naturally, especially when we consider that a long-running experiment on a large, well-maintained virtual testbed constitutes nothing more than an NGSP.

If the architecture in question offers substantial advantages, it will attract an increasing number of users over time. The architecture could gradually and seamlessly migrate from the virtual testbed infrastructure to a more dedicated one, or even remain on a commercial version of a virtual testbed, just as many commercial Web sites reside on Web hosting services. This way, natural market forces could take us gradually into a new architectural world.

However, instead of resulting in a single, radical architectural winner, easing the creation of new overlays could result in a large, ever-changing collection of more narrowly targeted overlays. To avoid architectural chaos and achieve some form of synergy, overlay designers must consider how to bring this union of overlays together to form a coherent framework, thereby becoming more than the sum of their individual functions.

Overlays offer an opportunity to radically change the architecture, not merely provide limited enhancements.

Such joint deliberations on how to achieve synergy among overlays could require a sociological change in research community interaction. When designing a single Internet architecture, we could not afford to ignore each other, since there would be only one place where research advancements could take effect. Overlay deployments can occur independently, without any coordination between or even cognizance of other efforts, yet coordination is required if overlays are to lead to a substantially different future.

VIRTUALIZATION: MEANS OR ENDS

The virtual testbed approach uses virtualization in two crucial ways. First, within its confines, the client proxy coupled with the virtual links between overlay nodes is qualitatively equivalent to a native network. This frees users from the tyranny of their local ISP and network providers no longer need to deploy new functionality at every node. Second, multiplexing overlay nodes creates many virtual testbeds that operate simultaneously, which greatly reduces the barrier to entry for any particular experiment.

Facilitating revolution

Researchers use virtualization techniques for experimentation and perhaps deployment, but these techniques remain independent of the architectures being tested. If architectural changes are rare, with long periods of quiescence or incremental evolution between times of architectural revolution, virtualization simply provides a means to accomplish these architectural shifts.

Given this situation, developers would want every architecture to include the seeds of its own destruction, seamlessly supporting proxy-like functionality and other hooks to make overlay establishment easier, but it isn't necessary for virtualization to be more deeply embedded.

If the Internet is, instead, in a constant state of flux, with new architectures always competing against the old and with many narrowly targeted architectures existing simultaneously, virtualization can play a more central role. The functionality to support overlays—virtual link establishment and proxy-like reachability—could conceivably become the architecture's core functionality, its narrow waist. In this scenario, PlanetLab would become the new model for the Internet.

Redefining Internet architecture

A change this profound makes us question what we mean by the term architecture. The two extreme points in the spectrum frame this debate. Our diverse experience spans the entire range of this spectrum, so our extreme characterizations are meant not to belittle any opinion but to clarify, if somewhat overstate, the differences.

Internet purists have a monolithic view of architecture centered around a single universal protocol, currently IP, required in each network element and around which all else revolves. They consider overlays blights on the architectural landscape, at best necessary evils reluctantly tolerated. In this view, virtualization provides only a means to install new architectures, not a fundamental aspect of the architecture itself.

Others take a more pluralist approach to architecture, with IP being only one component of an overall system we call the Internet. Overlays offer just one more way to deliver the service users want and are no less appropriate than any other approach to providing functionality. In this view, the dynamic and evolving architecture can, at any point, be defined as the union of the various existing overlays and protocols. The ability to support these multiple coexisting overlays then becomes the architecture's crucial universal piece.

The purist/pluralist split is apparent not only when defining an architecture but also when evaluating it. Purists aim for architectural flexibility because the architecture will remain in place a long time. Often, however, this flexibility does not result in immediate user benefits. Pluralists, on the other hand, put more emphasis on short-term performance improvements, arguing that the desired flexibility derives from adding or augmenting overlays rather than from the nature of each individual overlay. Since a key challenge for pluralists is providing flexibility at the high speeds enabled by advances in optical networks, a hybrid approach is also possible—a pure architecture for the high-speed core and a more pluralist architecture closer to the edge.

We do not pretend to know which position is right. We anticipate, however, that the virtual testbed will serve as a fertile Petri dish, allowing the development of many different overlays, each with its different characteristics. Perhaps this process will itself be an experiment from which we can observe either a drive toward uniformity or instead a synergy out of dynamic diversity.

The purist/pluralist split is apparent not only when defining an architecture but also when evaluating it.

The canonical story about architectural research's potential impact has long maintained that if testbed experiments show an architecture to be promising, ISPs and router vendors might adopt it. This story might have been realistic in the early days of the Internet—certainly DARTnet and other testbeds played an important role in the development of IntServ and Multicast—but it no longer applies. We as a community have long known that any nonincremental architectural change has little chance of adoption.

Further, we are rapidly reaching consensus that traditional testbeds have ceased being an effective way of experimenting with new architectures. Consequently, the research community has greatly narrowed its focus. Most current Internet research involves either empirical measurement studies or incremental modifications that can be deployed without major changes to the architecture.

Although empirical, incremental research plays a valuable role, it cannot meet the broader and more fundamental challenges the Internet faces. By providing easy access to virtual testbeds, we hope to foster a renaissance in applied architectural research that extends beyond incrementally deployable designs. Moreover, by replacing a discredited deployment story with a plausible story closely linked to the experimental methodology, we hope to raise the research community's sights.

We dare not simply complain about our current impasse—we must directly confront and overcome it. ■

Acknowledgments

We thank Guru Parulkar for many helpful comments and conversations regarding this article. NSF planning grant CNS-0439842 funded this work in part.

REFERENCES

1. D.J. Wetherall, "Active Network Vision and Reality: Lessons from a Capsule-Based System," *Proc. 17th ACM SOSP*, ACM Press, 1999, pp. 64-79.

2. C. Tschudin and R. Gold, "Network Pointers," *Proc. ACM SIGCOMM*, ACM Press, vol. 33. no. 1, 2003, pp. 23-28.
3. T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet Denial-of-Service with Capabilities," *CCR*, vol. 34, no. 1, 2004, pp. 39-44.
4. I. Stoica et al., "Internet Indirection Infrastructure," *Proc. ACM SIGCOMM*, ACM Press, 2002, pp. 73-86.
5. M. Walfish et al., "Middleboxes No Longer Considered Harmful;" www.pdos.lcs.mit.edu/papers/doa:osdi04/.
6. H. Balakrishnan et al. "A Layered Naming Architecture for the Internet," *Proc. ACM SIGCOMM*, ACM Press, 2004, pp. 497-506.
7. D.R. Cheriton and M. Gritter, "TRIAD: A Scalable, Deployable NAT-Based Internet Architecture;" www-dsg.stanford.edu/papers/triad/triad.html.
8. D. Zhu, M. Gritter, and D.R. Cheriton, "Feedback-Based Routing," *CCR*, vol. 33, no. 1, 2003, pp. 71-76.
9. National Science Foundation, Report on NSF Workshop on Network Research Testbeds; http://gaia.cs.umass.edu/testbed_workshop.
10. S. Savage et al., "Detour: Informed Internet Routing and Transport," *IEEE Micro*, Jan./Feb. 1999, pp. 50-59.
11. D.G. Andersen et al., "Resilient Overlay Networks," *Proc. 18th ACM SOSP*, ACM Press, 2001, pp. 131-145.
12. J. Touch and S. Hotz, "The X-Bone," *Proc. 3rd Global Internet Mini-Conference at Globecom*, IEEE Press, 1998, pp. 59-68.
13. L. Peterson et al., "A Blueprint for Introducing Disruptive Technology into the Internet," *Proc. HotNets–I*, ACM Press, 2002.
14. A. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," *Proc. ACM SIGCOMM*, ACM Press, 2002; http://citeseer.ist.psu.edu/article/keromytis02sos.html.
15. K. Svetz, N. Randall, and Y. Lepage, *MBone: Multicasting Tomorrow's Internet*, IDG Books, 1996.
16. L. Subramanian et al., "OverQoS: An Overlay-Based Architecture for Enhancing Internet QoS," *CCR*, vol. 33, no. 1, 2003, pp. 11-16.
17. J.D. Touch et al., "A Virtual Internet Architecture," ISI tech. report ISI-TR-2003-570, Mar. 2003.
18. Xilinx, "Virtex-II Pro Platform FPGAs: Introduction and Overview;" www.mangocom.com/xilinx.asp.

Thomas Anderson *is a professor in the Department of Computer Science and Engineering at the University of Washington. His research interest is in the practical issues in constructing robust, secure, and efficient computer systems. Anderson received a PhD in computer science from the University of Washington. Contact him at tom@cs.washington.edu.*

Larry Peterson *is professor and chair of computer science at Princeton University. His research interests focus on end-to-end issues in computer networks and systems built around networks. Peterson received a PhD in computer science from Purdue University. Contact him at llp@cs.princeton.edu.*

Scott Shenker *is a professor in the Department of Electrical Engineering and Computer Science at the University of California, Berkeley. His research interests include Internet architecture, distributed systems, and game theory. Shenker received a PhD in physics from the University of Chicago. Contact him at shenker@icsi.berkeley.edu.*

Jonathan Turner *is a professor in the Department of Computer Science and Engineering at Washington University. His research interests include the design and analysis of high-performance routers and switching systems, extensible communication networks, and algorithm analysis. Turner received a PhD in computer science from Northwestern University. Contact him at jon.turner@wustl.edu.*

Emerging Grid Standards

Individual projects carried out to meet specific needs must interact as part of a larger Grid environment, but no international consensus exists as to which of the many ideas, proposed standards, and specifications are likely to dominate in the future.

Mark Baker
University of Portsmouth

Amy Apon
University of Arkansas

Clayton Ferner
Jeff Brown
University of North Carolina at Wilmington

The Grid can be seen as a framework for "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources."[1] It allows researchers in different administrative domains to use multiple resources for problem solving and provides an infrastructure for developing larger and more complex applications potentially faster than with existing systems.

In general terms, the Grid has evolved from a carefully configured infrastructure, which supported limited Grand Challenge applications executing on high-performance hardware among numerous US centers,[2] to what we are starting to see today—a seamless and dynamic virtual environment being driven by international development and take-up.

As the Grid's potential started to become a reality over the past few years, industry has become increasingly involved. Commercial participation has accelerated development of hardened, industrial-strength software that supports Grid environments outside academic laboratories. This in turn has impacted both the Grid's architecture and the associate protocols and standards.

Most profoundly, the recent adoption of Web services, while bringing significant benefits, has also produced a somewhat fragmented landscape for application developers. Software and Grid services developers ideally seek to conform to conventions and standards widely adopted by their community. However, for various political and technical reasons, there are now competing views of how to implement the architecture and what standards to follow. This infighting is inhibiting Grid developers, who lack the assurance that future standards will support those used today.

GRID-RELATED STANDARDS BODIES

The Global Grid Forum (www.ggf.org) is the primary standards-setting body for the Grid. The GGF works with many organizations throughout industry that influence Grid standards and policies, including those for security and virtual organizations.

Other bodies include the Organization for the Advancement of Structured Information Standards, the World Wide Web Consortium, the Distributed Management Task Force, the Web Services Interoperability Organization, groups within Internet2 such as the Peer-to-Peer Working Group and the Middleware Architecture Committee for Education, and the Liberty Alliance.

Global Grid Forum

The GGF is a community-driven set of working groups that are developing standards and best practices for wide-area distributed computing. It was formed in 1998 from the merger of the Grid Forum in North America, the Asia-Pacific Grid community, and the European Grid Forum (eGrid).

In a process similar to that used for Internet standards, the GGF creates four types of documents that provide information to the Grid community:

- *informational*—a useful idea or set of ideas;
- *experimental*—useful experiments;
- *community practice*—common practices or processes that influence the community; and
- *recommendations*—specifications, which are analogous to Internet standards-track documents.

The GGF currently divides its efforts among seven areas—including, for example, architecture, data,

0018-9162/05/$20.00 © 2005 IEEE

OGSA defines the fundamental services that an e-business or e-science application would use.

and security—within which numerous working and research groups operate. Within the data area, standards under development include data access and integration services, Grid file systems, Grid FTP, grid storage, IPv6, and data replication. Nearly 30 research groups explore longer-term issues for which it may be premature to develop specifications.

Joining a GGF working group involves simply subscribing to its e-mail list. The project members, meeting agendas, and work progress are all posted online.

OASIS

A not-for-profit international organization that promotes industry standards for e-business, OASIS (www.oasis-open.org) was founded in 1993 as SGML Open and changed its name in 1998 to reflect its expanded technical scope. This includes developing standards such as those related to the Extensible Markup Language (XML) and the universal description, discovery, and integration (UDDI) service. OASIS produces Web services standards that focus primarily on higher-level functionality such as security, authentication, registries, business process execution, and reliable messaging.

Participants in OASIS can be either unaffiliated individuals or member-company employees. At least three organizations must implement a standard before OASIS will approve it.

World Wide Web Consortium

The W3C (www.w3.org) is an international organization initiated in 1994 by Tim Berners-Lee to promote common and interoperable protocols. It created the first Web services specifications in 2003 and initially focused on low-level, core functionality such as SOAP and the Web Services Description Language (WSDL). The W3C has developed more than 80 technical specifications for the Web, including XML, HTML, and DOM.

W3C members are organizations that typically invest significant resources in Web technologies. OASIS is a member, and the W3C has partnered with the GGF in the Web services standards area.

Distributed Management Task Force

The DMTF (www.dmtf.org) is an industry-based organization founded in 1992 to develop management standards and integration technologies for enterprise and Internet environments. DMTF technologies include the Common Information Model and Web-Based Enterprise Management. The DMTF formed an alliance with the GGF in 2003[3] for the purpose of building a unified approach to the provisioning, sharing, and management of Grid resources and technologies.

Web Services Interoperability Organization

WS-I (www.ws-i.org) is an open industry body formed in 2002 to promote the adoption of Web services and interoperability among different Web services implementations. Its role is to integrate existing standards rather than create new specifications. WS-I publishes profiles that describe in detail which specifications a Web service should adhere to and offer guidance in their proper usage. The goal is to provide a set of rules for integrating different service implementations with a minimum number of features that impede compatibility.

Internet2

Internet2 (www.internet2.edu) is a consortium of groups from academia, industry, and government formed in 1996 to develop and deploy advanced network applications and technologies.

The Middleware Architecture Committee for Education (http://middleware.internet2.edu/MACE) aims to create an interoperable middleware infrastructure for research and education. MACE develops good-practices documents, designs pilot projects and intercampus experiments, and recommends technical standards. Internet2 working groups related to Grid standards include the Higher Education PKI Technical Activities Group, the Peer-to-Peer Working Group, and the Shibboleth project.

Liberty Alliance

The Liberty Alliance (www.projectliberty.org) is an international alliance of companies, nonprofit groups, and government organizations formed in 2001 to develop an open standard for *federated identity management,* which addresses technical, business, and policy challenges surrounding identity and Web services. The Liberty Alliance has developed the Identity Federation Framework, which enables identity federation and management and provides interface specifications for personal identity profiles, calendar services, wallet services, and other specific identity services.

OPEN GRID SERVICES ARCHITECTURE

The most important Grid standard to emerge recently is the Open Grid Services Architecture, which aims to define a common, standard, and open architecture for Grid-based applications.

The GGF announced OGSA at Global Grid Forum 4 in February 2002, presented a draft overview four months later,[4] and created the OGSA Working Group in September 2002 to draft specifications.

At GGF10 in March 2004, the GGF declared OGSA to be its flagship architecture, and three months later, at GGF11, it released version 1.0.[5] OGSA v2.0, a proposed GGF recommendation, is scheduled for release in June 2005.

Service-oriented architecture

As Figure 1 shows, OGSA is a service-oriented architecture that specifies a set of distributed computing patterns realized using Web services. It aims to define all the fundamental services that an e-business or e-science application would use such as job and resource management, communications, and security, leaving various working groups within the GGF and other Grid-standards organizations to specify the services' interfaces, semantics, protocols, and other technical details.

Because the Grid is a dynamic environment in which service instances can come and go during task dispatching, resource configuration and provisioning, and system state changes, OGSA provides interfaces for lifecycle service management. It also supports state data associated with Grid services, an approach conceptually similar to traditional object-oriented programming environments. In addition, OGSA includes a callback operation in which clients can register interest in a service and receive notification of any change in that service.

Open Grid Services Infrastructure

OGSA instantiations depend on emerging specifications. The first instantiation was the Open Grid Services Infrastructure. OGSI was based on the concept of *Grid services,* enhanced Web services that provided a standard set of mechanisms to manage state. Released in July 2003, OGSI v1.0 defined a set of principles and extensions for using WSDL and XML Schema to enable stateful Web services.[6]

Critics identified several problems with OGSI.[7] First, many thought it was too large for one specification. In addition, because OGSI was not a pure subset of Web services, it required a modification to standard WSDL, called Grid WSDL, which would have necessitated extending current tools to parse and process WSDL for Grid services. Finally, even though many other Web services systems have object-oriented implementations, some viewed OGSI as too object oriented. To support transient, potentially short-lived instances, OGSI used OO concepts such as statefulness and the factory pattern to create Grid service instances.

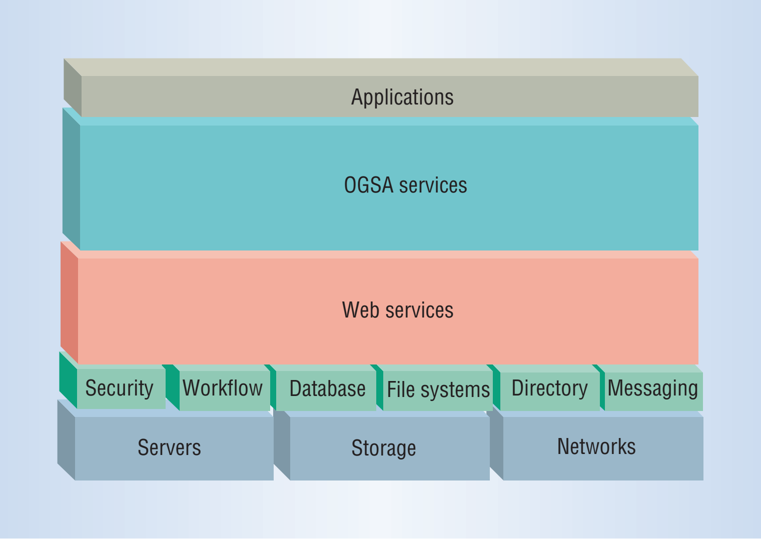


Figure 1. Open Grid Services Architecture. OGSA is a service-oriented architecture that specifies a set of distributed computing patterns realized using Web services.

WEB SERVICES RESOURCE FRAMEWORK

Widespread dissatisfaction with OGSI led to a collaborative effort among architects from the Grid and Web services communities to define an alternative infrastructure based on unadulterated Web services specifications. On 20 January 2004, Hewlett-Packard, IBM, Fujitsu, and the Globus Alliance announced the WS-Resource Framework (www.globus.org/wsrf). WSRF contains a set of specifications for expressing the relationship between stateful resources and Web services. The specifications define specific message exchange formats and related XML definitions.

After revising and updating the WSRF specifications based on industry feedback, a development team submitted the final results to two new OASIS technical committees, the WS-Resource Framework (WSRF) TC and the WS-Notification (WSN) TC.

The WSRF TC (www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf) was formed to standardize four specifications:

- WS-ResourceLifetime—describes how to manage the lifetime of a resource and specifies Web services operations used to destroy a WS-Resource;
- WS-ResourceProperties—defines how to query and modify WS-Resources described by XML Resource Property documents;
- WS-ServiceGroup—describes how to represent and manage collections of Web services and/or WS-Resources; and
- WS-BaseFaults—defines a base fault XML type for use when returning faults in a Web services message exchange.

The WSN TC (www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn) was created to standardize three other specifications defining Web services interfaces:

- WS-BaseNotification—handles asynchronous notification, including interfaces used by a notification producer or consumer;
- WS-BrokeredNotification—handles asynchronous notification; and
- WS-Topics—organizes and categorizes items of interest for subscription, known as topics.

Both technical committees republished the specifications as working drafts and started reviewing them in depth.

WSRF defines functions that allow interaction with WS-Resources such as query, lifetime management, and group membership.

Stateful resources

The OASIS WSRF TC aims to define a generic and open framework for modeling and accessing stateful resources using Web services. This includes mechanisms to describe views of state, support state management through properties associated with the Web service, and describe how these mechanisms are extensible to groups of Web services.

WSRF defines the means by which

- a Web service can be associated with one or more stateful resources;
- a service requestor can access stateful resources indirectly through Web services that encapsulate the state and manage all aspects of the service-based access to the state;
- the stateful resources can be destroyed, immediately or via time-based destruction;
- a stateful resource's type definition can be associated with a Web service's interface description and ensure well-formed queries against the resource via its interface;
- a stateful resource's actual state can be queried and modified via message exchanges;
- end-point references to a Web service that encapsulate stateful resources can be renewed when they become invalid due to, for example, a transient failure in the network; and
- the stateful resources can be aggregated for domain-specific purposes.

At the heart of WSRF is WS-Resource, which defines the relationship between Web services and stateful resources as an *implied resource pattern*. A WS-Resource is the "composition of a Web service and a stateful resource"[7] that can be described by an XML Schema associated with the Web services port type and addressed by a WS-Addressing EndpointReference.[8] WSRF defines functions that allow interaction with WS-Resources such as query, lifetime management, and group membership.

Currently, several early releases of WSRF-based systems are available, including Globus Toolkit 4 (www-unix.globus.org/toolkit) and WSRF.NET (www.cs.virginia.edu/~gsw2c/wsrf.net.html). Other development teams have implementations in progress such as WSRF::Lite (www.omii.ac.uk/mp/mp_wsrf_lite.htm), Unicore (www.unicore.org), and Python Globus (http://dsd.lbl.gov/gtg/projects/pyGlobus).

Event notification

Currently, two specifications describe event notification with respect to resources: WS-Eventing and WS-Notification. Originally released in January 2004, WS-Eventing[9] is a collaborative effort by Microsoft, IBM, BEA Systems, Computer Associates International, Sun Microsystems, and Tibco Software. Released around the same time, WS-Notification (www-106.ibm.com/developerworks/library/specification/ws-notification) is a joint initiative by Akamai Technologies, Computer Associates International, Fujitsu Laboratories of Europe, Globus, Hewlett-Packard, IBM, SAP AG, Sonic Software, and Tibco Software.

There is a move to merge these competing specifications, especially as IBM, Computer Associates, and Tibco contribute to both. The OASIS WSN TC is currently developing a standard based on WS-Notification.

WS-Eventing. This specification allows Web services to be notified of events that occur with other services. An *event source* is a Web service that produces notifications or event messages. An *event sink* is a Web service that receives notifications. A Web service subscribes itself or another service with a source to be a sink and thus receive events from that source. The subscription has an expiration time, which can be renewed, although it may have an indefinite termination.

WS-Eventing defines a *subscription manager,* which manages the subscriptions on behalf of an event source. It also includes the concept of *delivery mode,* which specifies how notifications should be delivered. For example, a source service can request that a notification be wrapped in a standard message. The only mode that the specification defines is *push mode,* which implies the delivery of individual, unsolicited, asynchronous SOAP messages. WS-Eventing also provides for source-side filtering of messages, such as using an XPath predicate expression.

WS-Notification. This family of specifications describes the mechanisms by which Web services can receive notification of an event related to a resource.[10] Web services that produce notifications are referred to as *notification producers,* while those that receive such notifications are *notification consumers*.

WS-Notification also describes a *subscription manager* as well as a *notification broker.* Using a separate notification broker can

- relieve the producer of the load needed to process notifications;
- reduce the number of interserver messages;
- provide a finder service, matching producers and consumers; and
- allow anonymous notification.

The notification producer can perform both of these roles, or a separate entity can offload these responsibilities from the producer. Unlike the subscription manager in WS-Eventing, this role in WS-Notification provides mechanisms to pause and resume subscriptions as well as to list them. Both specifications enable a separate entity to make subscription requests on behalf of a notification consumer.

Topics in WS-Notification support the hierarchical organization of notifications and offer a convenient way to locate notifications of interest. It is not clear whether topics provide greater functionality than XPath with respect to filtering XML documents, but topics should be applicable to other types of documents. Further, using topics in combination with the notification broker to pause and resume subscriptions enables demand-based publishing: If there are no subscribers, then nothing is published.

OTHER STANDARDS AND TRENDS

Despite the upcoming release of OGSA v2.0, some ongoing and recently initiated Grid projects cannot wait for production implementations of WSRF. Alternatives include WS-I's Basic Profile 1.0, the Web Services Grid Application Framework, and the Open Middleware Infrastructure Institute's WS-I+.

WS-I Basic Profile

In April 2004, WS-I published Basic Profile 1.0,[11] which contains guidelines for using SOAP, WSDL, and UDDI. BP1.0 has both recommendations and requirements for compliant services—for example, it recommends sending SOAP messages with HTTP/1.1 but requires the use of either HTTP/1.1 or HTTP/1.0.

WS-Notification describes the mechanisms by which Web services can receive notification of an event related to a resource.

Many applications other than Web services use HTTP, which has features that are appropriate in some environments but not in others. For example, HTTP cookies facilitate Web-based state management, but because cookies are not part of the SOAP envelope, BP1.0 mandates their use only in limited ways.

In some cases, BP1.0 tightens requirements in existing specifications. For example, SOAP 1.1 allows the use of the HTTP POST method as well as the HTTP Extension Framework's M-POST method, whereas BP1.0 permits only the former.

BP1.0 also clarifies ambiguities in some specifications. For example, a service sends a SOAP fault message when an error occurs. BP1.0 requires that the `soap:fault` element has no element children other than `faultcode`, `faultstring`, `faultactor`, and `detail`. Further, for extensibility the detail element can contain any type of element, thus a compliant service must accept such messages.

WS-I released Basic Profile 1.1[12] in August 2004. Some of the material in BP1.0 became Simple SOAP Binding Profile 1.0.[13] WS-I also released Attachments Profile 1.0[14] in August 2004.

Web Services Grid Application Framework

Grid services have requirements beyond those of standard Web services. The Web Services Grid Application Framework[15] proposes to meet the needs of Grid applications by extending basic Web services functionality.

The WS-GAF approach differs greatly from OGSI. Consider, for example, the problem of making services stateful. With OGSI, the user creates a service instance that generally only the creator uses. In contrast, WS-GAF uses the WS-Context specification,[16] which mandates that SOAP message headers carry service context information.

WS-I+

The UK e-Science Programme (www.rcuk.ac.uk/escience) has funded more than 100 separate projects that use a number of Grid technologies, many of which are based on Web services. It has also established the Open Middleware Infrastructure Institute (www.omii.ac.uk) to act as a center for expertise in Grid middleware and a repository for the software developed by the various projects. One goal of the OMII is to provide a relatively stable development environment for Grid-based enterprises.

The lack of Grid standards is a serious problem for the e-Science projects, some of which will be complete before specifications such as WSRF and WSN emerge. The OMII's approach is to build on WS-I profiles and create WS-I+,[17] which will identify existing standards that are considered safe and will potentially interoperate with emerging specifications. As in WS-I, the core of the service architecture consists of XML Schema Definition, WSDL 1.1, and SOAP 1.1.

The OMII expects to exploit BPEL's built-in extensibility mechanisms to support the scientific community's Web services needs.

For service discovery, the WS-I profiles include UDDI; WS-I+ might use UDDI, although the OMII is considering adopting registry service extensions that better suit scientific application needs. To address Grid workflow, WS-I+ uses the popular Business Process Execution Language (www-128.ibm.com/developerworks/library/specification/ws-bpel). The OMII expects to exploit BPEL's built-in extensibility mechanisms to support the scientific community's Web services needs.

Two competing specifications deal with addressing Grid services: WS-Addressing[8] and WS-MessageDelivery.[18] WS-Addressing has not been submitted to a standards body but is part of WSRF, while WS-MessageDelivery has been submitted to the W3C. WS-I+ will include WS-Addressing, which should facilitate future integration with WSRF.[19]

GRID SECURITY INFRASTRUCTURE

The Grid Security Infrastructure (https://forge.gridforum.org/projects/gsi-wg) implemented by the Globus Toolkit is a de facto standard for Grid security. GSI uses X.509 identity and proxy certificates, which provide a globally unique identifier that can authenticate and authorize an entity with accessed Grid resources.[20] In GSI, the owner typically grants use of a resource to individual users, who must have an account for each accessed resource. This becomes impractical as the number of users and resources grows.

Community Authorization Service

To overcome the access problem, the Community Authorization Service[21] provides an individual community identifier that authorizes a user for a resource. However, this solution requires additional Grid infrastructure and administration, which can lead to security problems when unknown users request a CAS account. For example, the CAS administrator might not know the person's institutional affiliation, which can be used to verify identity and trustworthiness.

GridShib and ESP-GRID

Two new projects are investigating alternative solutions that will impact the GSI standards. GridShib (http://grid.ncsa.uiuc.edu/GridShib) and ESP-GRID (http://e-science.ox.ac.uk/oesc/projects/index.xml.ID=body.1_div.20) will create new mechanisms and policies for distributed authorization and help Grid virtual organizations integrate with traditional organizations' security infrastructures. These projects should also lead to new tools and standards for administering user attributes and resource requirements. Both projects will leverage technologies in the Internet2's Shibboleth project (http://shibboleth.internet2.edu).

Shibboleth. Based on the Security Assertion Markup Language standard (www.oasis-open.org/committees/tc_home.php?wg_abbrev=security), this system is designed to exchange attributes between trusted organizations to authenticate and authorize users to remote resources. A user who desires to access a resource at a remote institution authenticates at a home institution, then the home institution passes the user's attributes securely through a trust relationship to the remote institution.

The remote institution authorizes access to the resource based on the user's attributes. For example, a member of a biomedical informatics research group could receive access to a remote institution's data set based on this group membership. The remote institution can require any number of user attributes before granting access to the resource, and users have the option of releasing attributes to particular resources, thereby maintaining privacy for access to some types of remote resources. Shibboleth's approach simplifies access control policies and makes them more scalable.

GridShib. Funded by the National Science Foundation Middleware Initiative, GridShib supports an identity federation between the Grid and higher-education communities by combining Shibboleth with GSI. Currently, Shibboleth only provides authorization and authentication for Web-based resources. In addition to using existing campus authentication and identity management infrastructures, GridShib plans to provide access to non-Web-based resources.

To accomplish this, GridShib will introduce two new modes of operation. In *pull mode*, a user with a GSI certificate contacts Shibboleth with a registration request and sends a key certificate to the target resource; the target resource contacts Shibboleth with a request for the user's attributes based on the user's key certificate. In *push mode*, Shibboleth passes the user's attributes to the target resource along with the key certificate when the user requests access to the resource.

An initial release of GridShib, planned for summer 2005, will support the pull-mode operation; follow-on releases will support the push mode and user pseudonymity. GridShib will likely be integrated with Globus Toolkit 4.2 or 4.4 and Shibboleth v1.3.

ESP-GRID. Funded by the UK's Joint Information Systems Committee, ESP-GRID is also investigating how Shibboleth can help provide solutions to Grid authentication, authorization, and security issues. In addition, ESP-GRID will reappraise public-key infrastructure use within the UK e-Science Programme and the Grid in general.

Any developer who wants to create Grid services or applications today faces the dilemma of deciding which of the many frameworks and specifications to follow, as currently there are no guarantees that industry and the open source community will embrace any one of them. Research is one thing, wide-scale deployment another.

Developers could use any of the competing frameworks and specifications to build wide-area infrastructure and associated applications. However, apart from the core Web services components—SOAP 1.2 and WSDL 1.1—all of the specifications are relatively new. In addition, many are drafts or in an early definition stage; even if a particular specification is accepted, a process that can take several years, its exact form is likely to differ from earlier versions. For these reasons, Savas Parastatidis and Jim Webber[22] argue that for production services the safest approach is to adopt existing and stable Web services specifications.

Despite resistance to new specifications, there is a growing demand for standards at the Grid's higher-level layers. For example, a recent effort among application developers to create a job and file management standard is gaining momentum. In December 2004, the GGF established a research group to begin developing the Simple API for Grid Applications (https://forge.gridforum.org/projects/saga-rg), which an application developer could use to specify a job request along with associated files and resources. This API will replace a number of incompatible tools that developers currently are using for such tasks. The SAGA research group will probably become a full GGF working group in the near future.

Another needed higher-level Grid standard would specify the type of information that a Grid monitoring system gathers. Although many open source and commercial tools are available for this purpose, the data returned varies considerably in content and detail. Lack of a standard makes monitoring heterogeneous resources difficult and limits the ability to assign tasks to resources and perform adaptive metascheduling.

For production services, the safest approach is to adopt existing and stable Web services specifications.

OGSA and WSRF represent significant cooperation among researchers in academia, government, and industry. These joint efforts point to a promising future for the Grid regardless of the uncertainties, inconsistencies, and interoperability problems developers currently face. ■

References

1. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputer Applications,* vol. 15, no. 3, 2001, pp. 200-222.
2. D. De Roure et al., "The Evolution of the Grid," *Grid Computing: Making the Global Infrastructure a Reality,* F. Berman, G. Fox, and A.J.G. Hey, eds., John Wiley & Sons, 2003, pp. 65-100.
3. "DMTF and GGF Announce Alliance Partnership," *Grid Today,* 5 May 2003; www.gridtoday.com/03/0505/101375.html.
4. I. Foster et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," draft document, 22 June 2002; www.globus.org/research/papers/ogsa.pdf.
5. I. Foster et al., *The Open Grid Services Architecture,* v1.0, Global Grid Forum, GWD-I (draft-ggf-ogsa-spec-019), 12 July 2004; www.ggf.org/documents/Drafts/draft-ggf-ogsa-spec.pdf.
6. S. Tuecke et al., *Open Grid Services Infrastructure (OGSI),* v1.0, Global Grid Forum, GFD-R-P.15 (proposed recommendation), 27 June 2003; www.ggf.org/documents/GWD-R/GFD-R.015.pdf.
7. K. Czajkowski et al., "From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution," v1.1, *IBM DeveloperWorks*, 5 Mar. 2004; /www-106.ibm.com/developerworks/library/ws-resource/ogsi_to_wsrf_1.0.pdf.
8. D. Box et al., *Web Services Addressing (WS-Addressing),* W3C member submission, 10 Aug. 2004; www.w3.org/Submission/2004/SUBM-ws-addressing-20040810.

9. D. Box et al., *Web Services Eventing (WS-Eventing)*, Aug. 2004; http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-eventing.asp.
10. S. Graham et al., *Web Services Notification (WS-Notification)*, v1.0, 20 Jan. 2004; http://ifr.sap.com/ws-notification/ws-notification.pdf.
11. K. Ballinger et al., *Basic Profile*, v1.0, Wed Services Interoperability Organization, 16 Apr. 2004; www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html.
12. K. Ballinger et al., *Basic Profile*, v1.1, Web Services Interoperability Organization, 24 Aug. 2004; www. ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html.
13. M. Nottingham, *Simple SOAP Binding Profile*, v1.0, Web Services Interoperability Organization, 24 Aug. 2004; www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html.
14. C. Ferris, A. Karmarkar, and C.K. Liu, *Attachments Profile*, v1.0, Web Services Interoperability Organization, 24 Aug. 2004; www.ws-i.org/Profiles/AttachmentsProfile-1.0-2004-08-24.html.
15. S. Parastatidis et al., *A Grid Application Framework Based on Web Services Specifications and Standards,* North East Regional e-Science Centre, School of Computing Science, Univ. of Newcastle upon Tyne, UK, 2003; www.neresc.ac.uk/ws-gaf/A%20Grid%20Application%20Framework%20based%20on%20Web%20Services%20Specifications%20and%20Practices%20v1.0.pdf.
16. M. Little, E. Newcomer, and G. Pavlik, *Web Services Context Specification (WS-Context)*, draft version 0.8, 3 Nov. 2004; http://xml.coverpages.org/WS-ContextCD-9904.pdf.
17. M. Atkinson et al., “Web Service Grids: An Evolutionary Approach,” Oct. 2004; www.omii.ac.uk/paper_web_service_grids.pdf.
18. A. Karmarkar et al., *WS-MessageDelivery* v1.0, W3C member submission, 26 Apr. 2004; www.w3.org/Submission/2004/SUBM-ws-messagedelivery-20040426.
19. N. Leavitt, “Are Web Services Finally Ready to Deliver?” *Computer,* Nov. 2004, pp. 14-18.
20. V. Welch et al., “X.509 Proxy Certificates for Dynamic Delegation,” *Proc. 3rd Ann. PKI R&D Workshop,* NIST, 2004; www.globus.org/Security/papers/pki04-welch-proxy-cert-final.pdf.
21. L. Pearlman et al., “A Community Authorization Service for Group Collaboration,” *Proc. IEEE 3rd Int’l Workshop Policies for Distributed Systems and Networks*, IEEE CS Press, 2002, pp. 50-59; www.globus.org/research/papers/CAS_2002_Revised.pdf.
22. S. Parastatidis and J. Webber, “Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications,” *Proc. 2004 IEEE Int’l Conf. Services Computing,* IEEE CS Press, 2004, pp. 65-72.

__Mark Baker__ is a reader in distributed systems and heads the Distributed Systems Group at the University of Portsmouth, UK. His research interests include wide-area resource monitoring, the integration of information services, and cross-environment application and service development. Baker received a PhD in maritime technology from Cardiff University, Wales. He is a member of the IEEE and the IEEE Computer Society, and is also cofounder and cochair of the IEEE Computer Society’s Technical Committee on Scalable Computing. Contact him at mark.baker@computer.org.

__Amy Apon__ is an associate professor in the Department of Computer Science and Computer Engineering at the University of Arkansas. Her research interests include cluster computing, security and privacy in Grid systems, and distributed file systems. Apon received a PhD in computer science from Vanderbilt University. She is a member of the IEEE, the IEEE Computer Society, and the ACM. Her research is supported by an Extending the Reach grant from Educause and grant number DUE-0410966 from the National Science Foundation. Contact her at aapon@uark.edu.

__Clayton Ferner__ is an assistant professor in the Department of Computer Science at the University of North Carolina at Wilmington. His research interests include Grid computing and parallel and distributed computing. Ferner received a PhD in mathematics and computer science from the University of Denver. He is a member of the IEEE, the IEEE Computer Society, and the ACM. Contact him at cferner@uncw.edu.

__Jeff Brown__ is a professor in the Department of Mathematics and Statistics at the University of North Carolina at Wilmington, where he also heads the software development team for the GridNexus Grid computing project. His research interests include computer-aided geometric design and computational geometry. Brown received a PhD in mathematics from the University of Georgia. Contact him at brownj@uncw.edu.

IEEE/ACM Transactions on Computational Biology and Bioinformatics

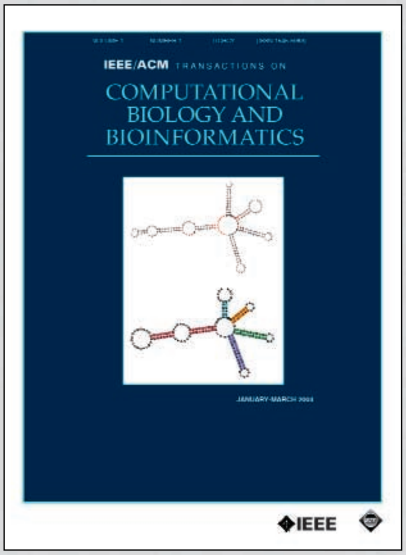
Brought to you by the IEEE Computer Society, IEEE Engineering in Medicine and Biology Society, IEEE Computational Intelligence Society, and the Association for Computing Machinery...

Stay on top of the exploding fields of computational biology and bioinformatics with the latest peer-reviewed research.

This new journal will emphasize the algorithmic, mathematical, statistical and computational methods that are central in bioinformatics and computational biology including…

- Computer programs in bioinformatics
- Biological databases
- Proteomics
- Functional genomics
- Computational problems in genetics

Publishing quarterly
Member rate:
$35 print issues
$28 online access
$46 print and online
Institutional rate: $375



Learn more about this new publication and become a subscriber today.

www.computer.org/tcbb

IEEE
IEEE COMPUTER SOCIETY
acm
EMB
IEEE Computational Intelligence Society

Figure courtesy of Matthias Höchsmann, Björn Voss, and Robert Giegerich.

RESEARCH FEATURE

Scaling Network Services Using Programmable Network Devices

The NEon system offers an integrated approach to architecting, operating, and managing network services. NEon uses policy rules defining the operation of individual network services and produces a unified set of rules that generic packet-processing engines enforce.

Christoph L. Schuba

Jason Goldschmidt

Michael F. Speer

Sun Microsystems Inc.

Mohamed Hefeeda

Simon Fraser University

Society increasingly relies on the Internet for communications, business transactions, information lookup, and entertainment, making it a critical part of our everyday life. The Internet's pervasiveness and its large-scale user base have prompted businesses and institutions to conduct many of their activities electronically and online, creating the need for efficient and reliable management of huge amounts of data.

A successful solution that has been adopted over the past several years is the concentration of critical computing resources in Internet data centers.[1] An IDC is a collection of computing resources typically housed in one physical location: a room, a floor in a building, or an entire building. Computing resources include Web, application, or database servers and network devices such as routers, firewalls, or load balancers. Large enterprises that rely heavily on the Internet and e-commerce applications typically operate their own IDCs, while smaller companies may lease computing resources within an IDC owned and operated by a service provider.

Computing resources in an IDC are typically organized into tiers. For instance, an IDC can dedicate one set of servers for Web access (Tier 1), a second set to run applications initiated by Web requests (Tier 2), and a third set to store data (Tier 3). Each tier is optimized for its own task: A Web server needs high-speed network access and the capacity to handle many concurrent connections, while a database server requires large storage capacity and fast I/O operations.

A tiered architecture allows incremental scaling of IDCs because the operator can independently upgrade each level. For example, if an IDC runs low on storage capacity, only the database server tier needs to be upgraded.

In addition to tiered architectures, IDCs employ other mechanisms to implement improved scalability and cost-effectiveness. One such mechanism is to offload especially expensive operations to special-purpose devices. For example, compute-intensive cryptographic engines often are used to protect client-server communications in financial transactions.

Instead of using expensive server cycles to perform cryptographic operations, highly optimized and less expensive devices can provide that functionality. These special-purpose Tier 0 devices, which precede the first server tier, are placed in the network before the end systems. Furthermore, the services that these devices provide are denoted as Tier 0 network services, or network services for short.

In addition to their use in Internet data centers, these network devices have been deployed in several other environments, including at the edge of Internet service provider networks, in storage area networks, and between tiers in server farms.

Published by the IEEE Computer Society

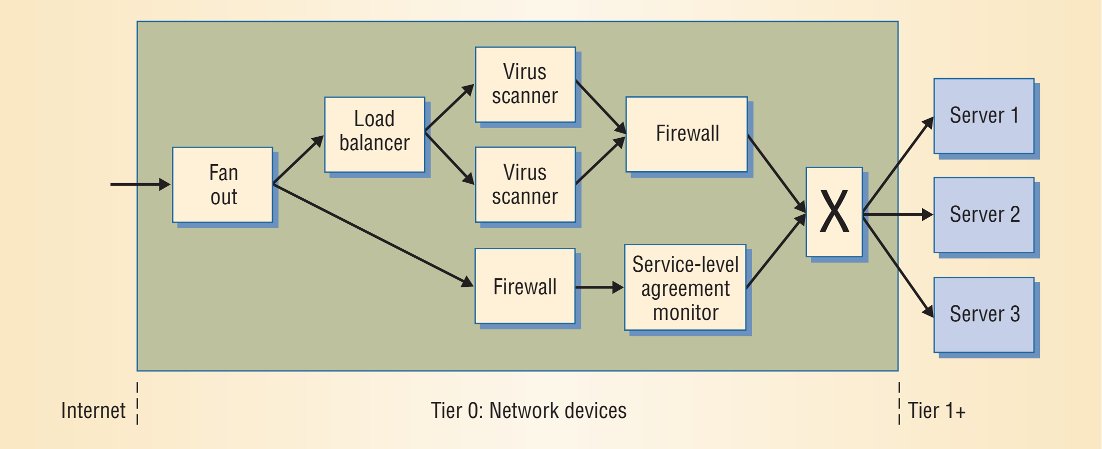


Figure 1. Network services deployment. The discrete approach implements network services as individual devices. As the number of network services increases, the discrete approach suffers from numerous scaling and manageability problems.

Network services are functions that network devices perform on packets before they reach their intended destination. These functions include firewalling, load balancing, intrusion detection, virus scanning, cryptographic acceleration, and service differentiation. Network devices are implemented using highly optimized software and custom hardware,[2] and they can be either standalone appliances or blades plugged into a blade chassis.

Figure 1 shows the discrete approach of deploying multiple network devices that each provide only one network service. However, as the number of network services increases, the discrete approach suffers from numerous scaling and manageability problems, including the following:

- *Fixed network service priorities*. Since network devices are physically cabled in a specific order, dynamically changing the flow processing order is difficult to accomplish. As network and business processing conditions change, dynamic alteration of priorities could provide new and valuable benefits in terms of enterprise security, competitiveness, and productivity.
- *Redundant packet classifications*. Each device performs packet classification and processing, essentially forcing a single flow to serially traverse processing stacks of the individual devices. Redundant processing is not only wasteful but also increases end-to-end latency, which has a negative impact on the user-perceived quality of service.
- *Multiple management consoles*. Each device requires a separate management console with its associated user interface and replicated administrative functions such as software updates and patching.
- *Lack of a feedback loop*. Applications running on servers may need to communicate with network devices that are processing pertinent packet flows. Such a feedback loop could significantly improve the performance of the network devices and the applications. However, it is difficult to establish this feedback loop in the discrete approach because an application would have to communicate with several heterogeneous devices, each with its own interface protocol.

The NEon architecture offers a novel approach for implementing network services. NEon is a paradigm shift away from special-purpose network devices, offering an integrated approach to architecting, operating, and managing network services. NEon employs new flow-handling mechanisms to integrate heterogeneous network services into one system.

NEON: AN INTEGRATED APPROACH

A NEon system accepts as input policy rules that define the operation of various network services and produces a unified set of rules that generic packet-processing engines can enforce. NEon uses rule unification (or crunching) to centralize the control of multiple network services. This centralization offers several advantages over the discrete approach for network services:

- *Flexible and dynamic network service priorities*. NEon merges network services rules together, with each rule possibly having a list of actions. These actions are ordered based on service priorities. Changing service priorities is a matter of changing the order of actions in the action list, which does not require recabling and can be done at runtime.
- *Single packet classification*. Each packet is classified only once before it is dispatched to the appropriate elements to perform the required actions, achieving a significant reduction in the packet processing delay.
- *Centralized management*. All supported network services are managed through one console through which the administrator inputs the rules and any configuration updates.
- *Single feedback point*. NEon servers tune the performance of network devices and applications at a single place. In contrast, in the discrete approach, applications are required to

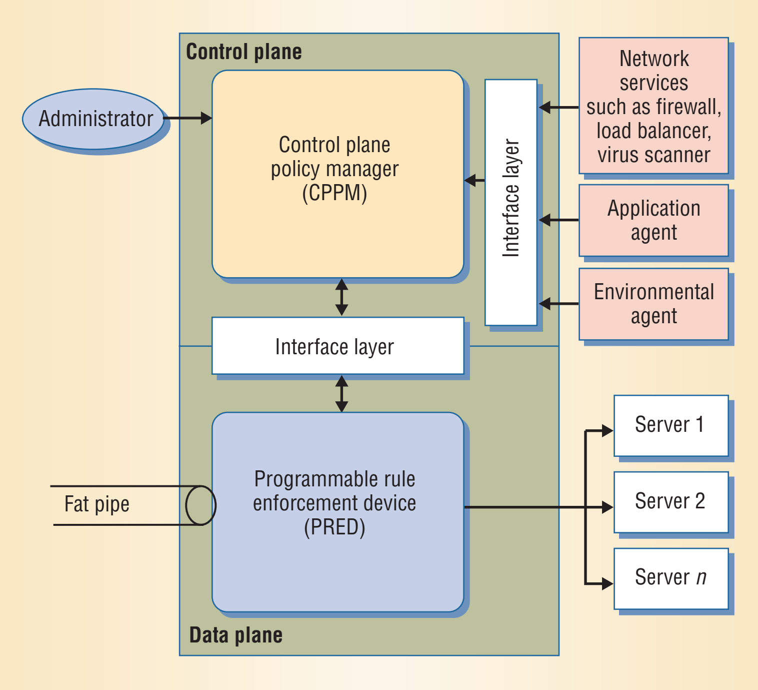

Control plane
Administrator
Control plane policy manager (CPPM)
Interface layer
Network services such as firewall, load balancer, virus scanner
Application agent
Environmental agent
Interface layer
Programmable rule enforcement device (PRED)
Fat pipe
Server 1
Server 2
Server *n*
Data plane

Figure 2. NEon architecture. Components comprise a control plane and a data plane separated by standards-compliant interface layers.

interact with several devices with different interfaces and communication protocols.

As Figure 2 shows, the NEon architecture's components are divided between two planes: control and data, delineated by standards-compliant interface layers.

The *control plane policy manager* is concerned with network service policy rules and metadata. The CPPM receives policy rules of different network services from system administrators and from management applications representing these network services. It integrates these rules into a unified set of rules that collectively implements the network services.

This unification of policies provides a virtualization of network services while preserving the policy enforcement and mapping of service semantics to physical hardware semantics. Rule unification is the heart of the CPPM, and it is accomplished through a rule-crunching algorithm.

The data-plane component, the *programmable rule enforcement device* (PRED), is a programmable classification and action engine that implements high-speed packet processing.[2] Packet processing is performed according to the rules that the CPPM prepares. Each rule consists of a filter and a list of actions.

The PRED checks data packets against the rule filters, and when a filter matches a packet, it applies the associated list of actions to that packet. PREDs use network processor technology to provide line-speed packet processing and the flexibility of programmable processors. Furthermore, network processors enable a PRED to support multiple network services concurrently.

The NEon architecture components communicate through interface layers that are designed to be compatible with open standards being developed by two industry-wide efforts: the Network Processing Forum (www.npforum.org) and the IETF Forwarding and Control Element Separation (ForCES) working group (www.ietf.org/html.charters/forces-charter.html). To further its efforts to accelerate the adoption of network processor technology in network devices, NPF publications identify the key elements in network devices and define standards for the hardware, software, and benchmarking aspects of building network devices. The ForCES working group defines a standard communication protocol between the control and data planes. The "Network Device Integration" sidebar describes other efforts to integrate the management of multiple network devices.

Standards-based separation of the NEon components offers a simplified management model and allows independent evolution of individual components. One interface layer resides between the CPPM and the PRED. This interface layer requires PREDs from different vendors to support a standard set of APIs that standards-compliant CPPMs will use. The other interface layer transforms input rules from various network services as well as application and environmental agents into the standard rule format that the CPPM supports. The NEon architecture uses application agents and environmental agents to enable dynamic adaptation and performance tuning of network devices.

Application agents form the feedback loop between applications running on the servers, in Tiers 1-3, and network devices. These agents run on servers and trap application-related events and forward them to the NEon CPPM. Typical examples of events that application agents gather are the number of connections opened, current CPU utilization, and memory usage.

Environmental agents provide input to the CPPM to adapt to environmental conditions such as server outages and link failures. Environmental agents allow NEon to dynamically steer the flow of packets to provide dynamic and highly available networked services.

NETWORK SERVICES INTEGRATION

The NEon approach integrates multiple network services and implements them in a single system.

The following are examples of features that different network services can have in common and of how their individual functions can be integrated.

- *Firewall*. A firewall checks individual data packets against preconfigured rule tables and either discards them or allows them to enter the network. An example of a rule in a firewall rule table looks like this: <TCP, 123.123.123.0/24, 0/0, 194.119.230.1/32, 80/16, ALLOW>, where the last field (ALLOW) represents the action that needs to be enforced on packets meeting the conditions specified in the preceding fields. That is, the rule allows the entry of TCP packets that come from the subnetwork 123.123.123.0/24 with any source port number and are destined to the HTTP server (port 80) running on the machine with IP address 194.119.230.1.
- *SLA monitor*. A service level agreement (SLA) monitor gathers statistics on the traffic flowing through a network. Its objective is, for example, to gather usage data for charging customers for the bandwidth used by their traffic. A configuration rule for an SLA monitor could look like this: <UDP, 123.123.123.0/24, 0/0, 0.0.0.0/0, 0/0, ACCT>, which means that statistics are to be collected on all UDP traffic originated from the subnetwork 123.123.123.0/24.
- *Load balancing*. A load balancer can be used in front of a number of servers to spread the load across them based on some criteria. The criteria could be the source address, destination address, protocol, or a combination thereof. A rule in the load balancer table may have the form: <TCP, 0.0.0.0/0, 0/0, 194.119.230.12/32, 23/16, LBGROUP1>, which means forward all TCP packets whose destination IP address is 194.119.230.12 and the destination port is 23 to a server that belongs to load-balancing group LBGROUP1.

Rule virtualization

Systems administrators must encode the network service semantics in the form of rules to communicate them to the hardware. To virtualize the rules of different network services, their semantics can be mapped into a single instance of generic packet processing hardware.

Most network services perform their functions by enforcing a set of rules, with each rule taking the form <Filter, ActionList>, where Filter defines a class of packets on which operations that ActionList specifies are to be performed.

A Filter is composed of several fields, each with a Field Name and a Pattern. The Field Name identifies the header field in a protocol layer—for example, the source address in the IP layer. The Pattern is composed of a bit string that determines which packets match this field.

The Pattern also specifies which bits in the string should be considered in matching packets and which bits can be ignored. One way to do that is by using the *mask length notion* (/len), which is commonly used in IP routing tables. For example, a pattern for a 32-bit IP address could be 128.12.30.0/24, which means that the leftmost 24 bits (128.12.30) should match the corresponding bits in packets, while the rightmost 8 bits can be ignored.

Network Device Integration

Previous efforts to integrate the management of multiple network devices includes the Open Platform for Security (www.opsec.com), which provides a standard framework for managing several independent security devices such as firewalls, intrusion detection, and authorization. However, OPSEC integrates the devices only at the management level, whereas NEon integrates both the management and enforcement (hardware) levels.

The rule-crunching algorithm has some similarities to high-speed packet-classification algorithms.[1] Packet classification means finding which rule matches a given packet. If multiple rules match, a conflict occurs and must be resolved so that only one rule applies. Rule conflicts can occur in NEon, but they are handled differently: Rules are merged or modified, and sometimes new rules are created.

An algorithm for detecting conflicts between two *k*-tuple filters creates a new filter when a conflict occurs; therefore, the total number of rules increases exponentially as the number of conflicts increases.[2] A more scalable conflict-detection algorithm builds binary tries for each filter field; each level of the trie is one bit of the field.[3] The algorithm computes a bit vector from this trie to aid in conflict detection.

The growth in the total number of rules is a critical issue in PREDs with limited memory. Because our rule-crunching algorithm merges and prioritizes the actions of conflicting rules, it does not incur exponential growth. An efficient data structure for detecting rule conflicts that is based on rectangle geometry works only for two-dimensional classification.[4] However, because classifying based on two fields is not sufficient for many network services, our rule cruncher uses five fields. Finally, an algorithm for removing redundant rules can be used as a preprocessing step for the rule cruncher to eliminate unnecessary service rules.[5]

References

1. D.E. Taylor, *Survey and Taxonomy of Packet Classification Techniques*, tech. report WUCSE200424, Dept. Computer Science and Eng., Washington Univ., 2004.
2. A. Hari, S. Suri, and G. Parulkar, "Detecting and Resolving Packet Filter Conflicts," *Proc. IEEE Infocom 00*, IEEE Press, 2000, pp. 1203-1212.
3. F. Baboescu and G. Varghese, "Fast and Scalable Conflict Detection for Packet Classifiers," *Computer Networks*, Aug. 2003, pp. 717-735.
4. D. Eppstein and S. Muthukrishnan, "Internet Packet Filter Management and Rectangle Geometry," *Proc. 12th ACM SIAM Symp. Discrete Algorithms* (SODA 01), ACM Press, 2001, pp. 827-835.
5. A. Liu and M. Gouda, *Removing Redundancy from Packet Classifiers*, tech. report TR0426, Dept. Computer Sciences, Univ. Texas at Austin, 2004.

The NEon architecture can apply several rules from different services to the same packet.

The simple mask-length approach applies only when bits that need to be matched are contiguous. To support more complex patterns such as those that match intermediate bits, the field pattern should specify the full bit mask (not only the length). To match a rule, a packet must match all fields of the rule's filter. To check a data packet against a specific field, the network device extracts the corresponding bits in the packet. If all bits in the field pattern match the extracted bits, a field is matched.

The ActionList is a series of processing actions that the network device performs on a packet. Actions are, for example, dropping a packet, gathering statistical information, controlling timer functions, modifying or marking a packet with metadata—such as inserting new fields or overwriting existing fields—or passing the packet on (doing nothing).

We can abstract the functioning of a network service as follows:

```
NetworkService  :=  < Rule > +
Rule            :=  < Filter, ActionList >
Filter          :=  < Field > +
Field           :=  < FieldName, Pattern >
ActionList      :=  < Action > +
Action          :=  Drop | Allow | Mark |
                    Overwrite | ...
```

This abstraction allows combining rules from several network services into a unified set of rules. However, some network services such as stateful services do not readily lend themselves to using the algorithm for performing this unification.

Rule crunching

The rule-crunching algorithm uses service abstraction to take rules from multiple services and generates a consistent rule set that a PRED can apply. Two concepts underlie this algorithm: rule merging and rule enforcement order.

Rule merging. Because the NEon architecture integrates multiple network services in one device, it can apply several rules from different services to the same packet. Rule merging occurs when two or more rules would match the same packet. The rule-crunching algorithm merges rules based on the set of packets each rule influences.

Consider two rules *r* and *r′* that belong to two different services. We define *S* as the set of packets that match rule *r*. Similarly, *S′* is the set of packets that match *r′*. Five relationships are possible between *S* and *S′*: EQUAL, DISJOINT, SUBSET, SUPERSET, and INTERSECT. Rule merging creates the following relationships:

- $S = S'$ (EQUAL). This relationship indicates that *r* has the same filter as *r′*, but they may have different actions. The result of merging is one rule denoted by *cr*. Rule *cr* will have the common filter and the concatenation (denoted by the pipe symbol) of the actions of *r* and *r′*. That is, $cr.filter = r.filter = r'.filter$, and $cr.ActionList = r.ActionList \mid r'.ActionList$.
- $S \cap S' = \phi$ (DISJOINT). This relationship means that the two rules *r* and *r′* are to be applied on different, nonintersecting sets of packets. In this case, no merging will happen, and the two rules *r* and *r′* are used in their original format.
- $S \subset S'$ (SUBSET). In this case, packets in the set *S* should be subjected to actions of rule *r* as well as rule *r′*. Moreover, packets in the set $S' - S$ should be subjected only to the action list of *r′*. Merging creates two rules cr_1 and cr_2, where (1) $cr_1.filter = r.filter$ and $cr_1.ActionList = r.ActionList \mid r'.ActionList$; and (2) $cr_2 = r'$. Note that the device stores cr_1 and cr_2 in a way that ensures that packets are checked against cr_1 before cr_2. Therefore, cr_2 will be applied only to packets that do not match cr_1 but match cr_2—that is, packets belonging to the set $S' - S$.
- $S \supset S'$ (SUPERSET). This case is equivalent to $S' \subset S$ (SUBSET) and handled accordingly.
- $S \cap S' \neq \phi$ (INTERSECT). Merging in this case results in three rules *cr*1, *cr*2, and *cr*3, where (1) $cr_1.filter = r.filter \cap r'.filter$ and $cr_1.ActionList = r.ActionList \mid r'.ActionList$; (2) $cr_2 = r$; and (3) $cr_3 = r'$. Again, cr_1 should be checked before both cr_2 and cr_3.

For each relationship, the algorithm creates equivalent crunched rules whose filters match the same set of packets as the original service rules. Moreover, the crunched rules perform the same actions as the original rules, and any existing ambiguity among them has been removed.

Rule enforcement order. A programmable rule enforcement device checks every packet flowing through it against the set of crunched rules stored in its table. The order of checking rules against packets is critical to the PRED's correct operation because a packet can match more than one crunched rule when only one rule should be applied. For example, a packet can match two rules, one of which contains more specific filters because it matches source IP address and port number rather than just the source IP address. Clearly, the more specific rule—the first one—should be applied.

The algorithm uses two approaches to determine the order of checking rules against packets: *ordered precedence* and *longest prefix matching*. Ordered precedence matching places rules with more specific filters earlier in the rule table. Rules are considered one at a time in the order specified: The first matching rule fires, and its action list is executed. In longest prefix matching, the algorithm applies the rule that shares the longest prefix with the corresponding fields in the packet.

Typically, the algorithm stores rules in a data structure that facilitates longest prefix matching, such as binary tries. Longest prefix matching assumes matching contiguous bits.

Rule-crunching algorithm

The input to the rule-crunching algorithm is the *service rule database* (srdb), a list of rules of individual network services. The administrator assigns a unique priority to each network service. All rules of the same service get the same priority. The srdb is ordered based on this priority, which ensures that all rules of the same network service come after each other. We illustrate this algorithm using high-level pseudocode:

```
1. $crdb \leftarrow r_1$; /* $r_1$ is the first rule
                    in $srdb$. */
2. foreach $r \in srdb - \{r_1\}$ do
3.   foreach $cr \in$ crdb do
4.     $rel \leftarrow DetermineRel(r, cr)$;
5.     if $rel ==$ DISJOINT
6.       add $r$ to $crdb$;
7.     else
8.       $MergeRules(r, cr, rel, crdb)$;
9. return $crdb$;
```

The algorithm's output is a unified set of rules: the *crunched rule database* (crdb). The algorithm subsequently removes rules from the srdb and adds them to the crdb until there are no more rules to move. A rule *r* in srdb is compared with every rule *cr* in crdb for possible merging.

The *DetermineRel()* function invoked in line four determines the relationship, *rel*, between the two rules by comparing their corresponding field filters. If the two rules cannot be applied on any packet simultaneously—that is, their packet sets are DISJOINT—no rule merging is performed, and *r* is added to crdb. Otherwise, the *MergeRules()* function is invoked to merge the two rules based on their relationship, *rel*. For example, if *rel* = EQUAL, function *MergeRules()* adds the *r* action list to the *cr* action list and adjusts the priority of the modified *cr*. Note that rule *r* itself is not added to crdb, which reduces the total number of rules in crdb.

Algorithm analysis. As the discussion on rule merging indicates, the DISJOINT, SUBSET, and SUPERSET relationships do not add new rules to the crdb; they merely move the service rules from the srdb to the crdb and can modify the rule's filter or action lists.

If the EQUAL relationship occurs between two rules, we add only one of them to the crdb—that is, the number of rules is reduced by one. If the INTERSECT relationship occurs between two rules, we add three rules to the crdb, which increases the total number of rules by one.

Our worst-case analysis assumes that no EQUAL relationships occur. Therefore, the crdb's final size is equal to the number of the original service rules (n) in addition to the maximum number of rules created from all possible INTERSECT relationships. To count the maximum number of INTERSECT relationships, the main factor in determining the relationship between two rules is their IP source and destination addresses because these fields can use address range wild-carding (for example, a.b.c.d/16 or a.b.c.d/24). Other fields will either match or will not.

For two rules to intersect, their IP source (or destination) addresses must share a common prefix, while their other fields can differ. Since IP addresses have a fixed number of bits k ($k = 32$ for IPv4), the common prefix can range from 1 bit to a maximum of k bits. In addition, crdb prefixes are unique because if two rules have the same prefix, they must have been merged in a previous iteration.

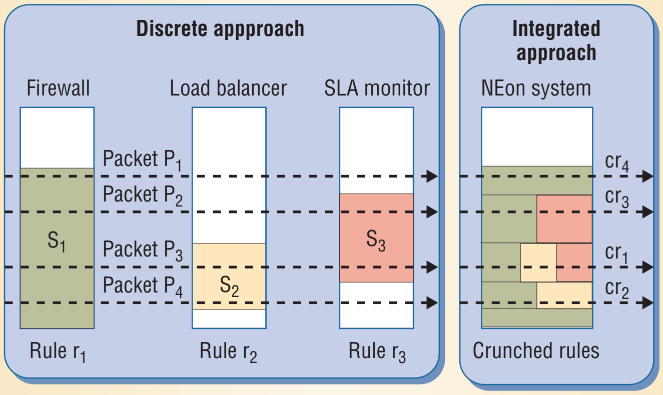
When the rule-crunching algorithm compares one rule from the srdb against all entries in the crdb, there can be at most $(k + k) \times 2 \times c = O(1)$ INTERSECT relationships, where c is the number of fields in the filter other than the IP addresses ($c = 3$ in the 5-tuple rules). Because c fields may or may not match in the INTERSECT relationship, we must include the factor $2 \times c$. The factor 2 comes from the fact that there are two IP addresses: source and destination. Therefore, each iteration of the algorithm's outer loop (line 2) will add a maximum of $O(1)$ rules to the crdb, which results in a space complexity of $n \times O(1) + n = O(n)$.

The same arguments apply for determining the algorithm's time complexity. Because the inner loop

The rule-crunching algorithm stores rules in a data structure that facilitates longest prefix or ordered precedence matching.

The input service rules are

r_1 = <TCP, 0.0.0.0/0, 0/0, 1.1.1.0/24, 80/16, [ALLOW]>
r_2 = <TCP, 0.0.0.0/0, 0/0, 1.1.1.7/32, 80/16, [LBGROUP1]>
r_3 = <TCP, 2.2.2.0/24, 0/0, 1.1.1.0/24, 80/16, [ACCT]>



The resultant crunched rules are

cr_1 = <TCP, 2.2.2.0/24, 0/0, 1.1.1.7/32, 80/16, [ALLOW, LBGROUP1, ACCT]>
cr_2 = <TCP, 0.0.0.0/0, 0/0, 1.1.1.7/32, 80/16, [ALLOW, LBGROUP1]>
cr_3 = <TCP, 2.2.2.0/24, 0/0, 1.1.1.0/0, 80/16, [ALLOW, ACCT]>
cr_4 = <TCP, 0.0.0.0/0, 0/0, 1.1.1.0/24, 80/16, [ALLOW]>

Figure 3. Rule crunching in NEon. The algorithm merges the input service rules for firewalling, load balancing, and service level agreement monitoring into a set of four crunched rules allowing different actions on incoming packets.

is invoked at most n times, the algorithm's overall time complexity is $O(n^2)$.

The rule-crunching algorithm is a part of the CPPM, which operates in the control plane, not in the performance-critical, high-speed data plane. Moreover, the CPPM does not necessarily run on the network device's hardware. The CPPM can run on a monitoring or management server that controls several network devices.

The management server provides a single point to feed and update rules for various network services. It also performs the rule crunching once for all attached network devices and then pushes the crunched rules to each network device. Furthermore, because it has the crunched rules for all devices, the management server can perform some optimizations such as removing redundant rules.[3]

Example. Figure 3 illustrates the crunching of three rules r_1, r_2, and r_3 belonging to three different services: firewalling, load balancing, and SLA monitoring, respectively. The color of each area in the figure represents the set of packets that matches a rule whose action is represented by the same color. For instance, the green area represents the set of packets on which the firewall performs the action ALLOW. The word ALLOW in the action list of the rules is also colored in green. Let S_1, S_2, and S_3 represent packet sets that match rules r_1, r_2, and r_3, respectively. The figure shows various relationships between the packet sets. For example, S_1 is a SUPERSET of S_2 and S_3, while the relationship between S_2 and S_3 is INTERSECT.

The rules r_1, r_2, and r_3 are initially stored in the srdb, and the crdb is empty. Line 1 of the algorithm moves r_1 to the crdb. Then, it merges r_2 with r_1. Since $S_2 \subset S_1$, r_2 is added to the crdb after concatenating the action list of r_1 to its action list, r_2 will have <ALLOW, LBGROUP1> as its action list and it will be inserted before r_1 in the crdb. Then, r_3 will be merged with the modified r_2. This is an INTERSECT relationship.

The merging produces a new rule with the filter <TCP, 2.2.2.0/24,0/0, 1.1.1.7/32, 80/16>, which is the intersection of the r_2 and r_3 filters. The new rule has the action list <ALLOW, LBGROUP1, ACCT>. In the final step, r_3 is merged with r_1, which modifies r_3 by concatenating the action list of r_1 to r_3's before adding r_3 to the crdb.

The resultant crunched rules are shown in the bottom part of Figure 3. Figure 3 also demonstrates the flow of sample packets P_1 to P_4 through the discrete network devices and the NEon system. The NEon system performs exactly the same actions on each packet that the discrete network devices perform. For example, packet P_3 matches the three rules r_1, r_2, and r_3 of the discrete devices and at the same time matches cr_1, which has the same actions as r_1, r_2, and r_3.

PROTOTYPE SYSTEM

To validate the NEon concept, we developed a complete prototype system. The prototype has been tested with commercial hardware devices using synthetic data traffic as well as configuration files from operational network devices.

CPPM code

The CPPM code is implemented in Java and has two main parts: a *service listener* and a *rule cruncher*.

The service listener receives policy rules of individual network services from the administrator and from the software agents representing network services. It is implemented as a pool of threads, one for each active network service. The service listener stores the received rules and rule updates in the srdb. Rules in the srdb are ordered based on their defined network service priorities.

The rule cruncher applies the rule-crunching algorithm to the srdb to create the crunched rule database. The rule cruncher can be invoked periodically (every few minutes or seconds), upon a rule update (insert, delete, or modify a rule), or explicitly by the administrator. The automatic and periodic invocation of the rule cruncher allows fast propagation of updated configurations to the enforcement devices.

PRED

Because there are currently no commercially available generic PREDs that can apply rules from multiple network services, for our prototype we modified two different hardware products to serve as PREDs.

We have tested our prototype on PolicyEdge, a network processor chip emulator from FastChip, and on the Sun Fire Content Load Balancer Blade (B10n) from Sun Microsystems. Successfully modifying these devices to run the NEon prototype demonstrates that the integrated approach for network services is both feasible and viable.

The B10n is a networking product that provides content load balancing for blade servers and horizontally scaled systems. The B10n operates at the data center edge, applies user-specified rules to classify client-side inbound traffic at wire speed, and applies load-balancing actions on the data traffic.

The B10n was designed to provide only content load balancing—that is, only one action is associated with each rule. We have augmented the firmware and the data structures to support multiple actions for each rule.

The B10n's rule-matching technique is based on two fields: IP source address and source port. In NEon, we use 5-tuple rules to represent a wider range of network services. We have changed the rule structure to support five layer-4 fields: source IP, destination IP, source port, destination port, and protocol. In addition, any field can be either fully specified or wild-carded.

Results

Several NEon system parameters were tested including the rule-crunching algorithm's runtime, the relationship between the number of crunched rules versus the number of raw service rules, and the number and type of merging relationships occurring among rules.

To perform the experiments, we used configuration files from deployed network services, handcrafted scenarios, and generated data. The configuration files had a small number of unique service rules—63 on average, which is typical for many Internet data centers. The crunching algorithm took about 6 milliseconds on a Sun Fire Ultrasparc 240 server-class machine and produced 148 crunched rules on average.

We also verified that the INTERSECT relationship, which increases the size of the crdb, does not occur frequently between rules: only 9.9 percent of the merging relationships were INTERSECT. More than 87 percent of the relationships were DISJOINT, with the small remaining percentage distributed among EQUAL, SUBSET, and SUPERSET relationships.

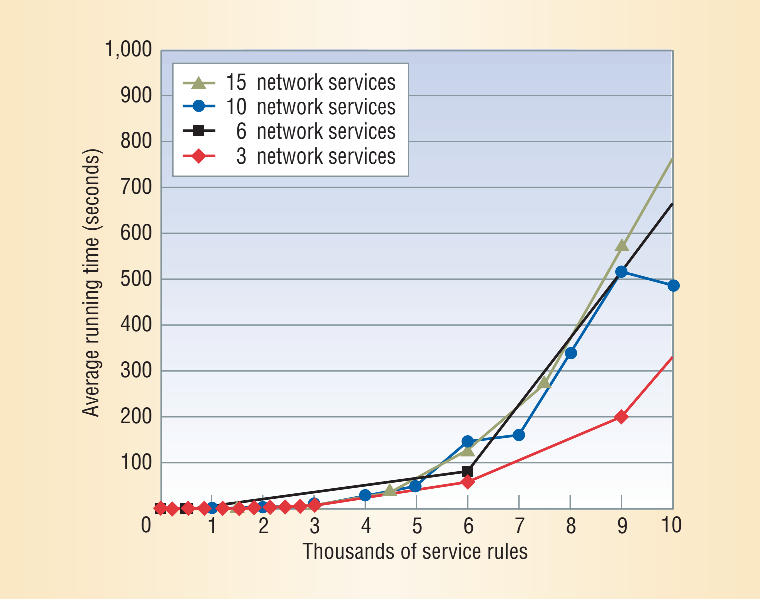


Figure 4. Rule cruncher runtime. The rule-crunching algorithm terminates in less than one minute for up to approximately 4,000 rules.

To test the scalability of the approach, we simulated various combinations of network services with a large number of rules. The rule filters were generated randomly within the appropriate ranges. For example, the transport protocol field was chosen randomly from either TCP or UDP. Random generation of rules stresses the rule-crunching algorithm because it produces more INTERSECT relationships than in typical configuration files; therefore, it pushes the running time and the size of the crunched rule database toward their worst cases.

Figure 4 shows the rule cruncher's average running time as the number of service rules increases from 0 to 10,000. The total number of service rules was divided among the simulated number of network services. For example, if we simulate 8,000 rules and 10 network services, each network service will have, on average, 800 rules. The algorithm terminates in less than one minute for up to approximately 4,000 rules. For larger numbers of rules, the running time is still on the order of minutes. This is acceptable given that the rule cruncher is not invoked frequently: It is needed initially and when rules or rule sets are modified.

The NEon integrated approach offers numerous advantages over the current practice of implementing network services as discrete devices, including

- flexibility of dynamically changing service priorities;
- single-packet classification, which leads to shorter end-to-end delay;
- centralized management; and
- a single point for applications to establish a feedback loop with network devices.

Testing a NEon prototype architecture with commercial hardware devices and data collected from operational network devices demonstrates that this system offers a feasible and viable approach for implementing sophisticated network services.

This work can be extended in several directions. The rule-crunching algorithm reconstructs the entire crunched rule database upon the modification of any service rule. Currently, this is not a major concern because rule updates are infrequent and occur on a time scale of hours or days. However, in the future, more dynamic environments in which service rules can be updated on a shorter time scale will pose a challenge for the rule cruncher. One possible solution is to design a differential rule cruncher, which performs the minimum amount of work to adjust the crunched-rule database to reflect changes in service rules.

Another future direction that would broaden the scope and applicability of the NEon approach is to extend the rule-unification mechanism to accommodate more network services such as stateful services, services that access data beyond packet headers (application data), and services such as the network address translator that can change a packet's identity.

A complete testbed is needed to thoroughly compare NEon with the discrete approach. Such a test bed would contain several network devices connected in the traditional discrete way and an equivalent NEon system. It would be interesting to measure performance parameters such as average packet processing time, the time needed to reconfigure the network devices, and how quickly the system can propagate rule updates to the rule-enforcing hardware. ■

Acknowledgments

We appreciate the insightful discussions, suggestions, and support from Robert Bressler, Danny Cohen, Cindy Dones, Kevin Kalajan, Lisa Pavey, and Raphael Rom. We also thank Ahsan Habib, Nan Jin, Sumantra Kundu, Santashil PalChaudhuri, George Porter, Ioan Raicu, and Aaron Striegel for their contributions to the NEon project during their summer internships at Sun Microsystems Inc.

References

1. M. Arregoces and M. Portolani, *Data Center Fundamentals*, Cisco Press, 2003.
2. D. Comer, *Network Systems Design Using Network Processors*, Prentice Hall, 2004.
3. A. Liu and M. Gouda, *Removing Redundancy from Packet Classifiers*, tech. report TR0426, Dept. Computer Sciences, Univ. Texas at Austin, 2004.

***Christoph L. Schuba** is a senior staff engineer in the Security Program Office at Sun Microsystems Inc. His research interests include network and computer system security, high-speed networking, and distributed systems. Schuba received a PhD in computer science from Purdue University. He is a member of the Internet Society and Usenix. Contact him at christoph.schuba@sun.com.*

***Mohamed Hefeeda** is an assistant professor in the School of Computing Science at Simon Fraser University, Canada. His research interests include computer networks, multimedia networking, peer-to-peer systems, and network security. Hefeeda received a PhD in computer science from Purdue University. He is a member of the IEEE and the ACM Special Interest Group on Data Communications. Contact him at mhefeeda@cs.sfu.ca.*

***Jason Goldschmidt** is a technical staff engineer, Network Systems Group, Sun Microsystems Inc. His research interests include network processing, protocols, and scalable architectures. Goldschmidt received a BS in computer science and engineering from Bucknell University. Contact him at jason.goldschmidt@sun.com.*

***Michael F. Speer** is a senior staff engineer, Netra Systems and Networking, Sun Microsystems Inc. His research interests include scalable network systems and services architectures for network deployments. He received a BS in computer engineering from the University of California, San Diego. He is a member of the IEEE Computer Society, the ACM, and Usenix. Contact him at michael.speer@sun.com.*

Leveraging Social Networks to Fight Spam

Social networks are useful for judging the trustworthiness of outsiders. An automated antispam tool exploits the properties of social networks to distinguish between unsolicited commercial e-mail—spam—and messages associated with people the user knows.

P. Oscar Boykin
University of Florida

Vwani P. Roychowdhury
University of California, Los Angeles

The amount of unsolicited commercial e-mail—spam—and, more importantly, the portion of e-mail that is spam, has increased dramatically in the past few years. A recent study showed that 52 percent of e-mail users say spam has made them less trusting of e-mail, and 25 percent say that the volume of spam has reduced their e-mail use.[1]

This crisis has prompted proposals for a broad spectrum of potential solutions, ranging from more efficient antispam software tools to antispam laws at both the federal and state levels.

While the jury is still out on how widely such antispam laws will be enacted and how effectively they would stem the flow, the objective of the various legal and technical solutions is the same: to make sending spam unprofitable and thereby destroy the spammers' underlying business model.

Achieving these goals requires widespread deployment and use of antispam techniques. To gain user confidence, which is a prerequisite for wide deployment, the tool must be accurate, user friendly, and computationally efficient.

Our technique simultaneously achieves all three requirements. This technique is predicated on recognizing the unique characteristics inherent to social networks and the proven wisdom of using such trust networks to make the right choices. The reliability of our decisions, then, depends strongly on the trustworthiness of our underlying social networks. Thus, we seem to have evolutionarily developed several interaction strategies that can generate a trustworthy network.

A commonly espoused rule suggests that trust is built based not only on how well you know a person but also on how well others in your network know the person. This interaction dynamic results in close-knit communities in which, if Alice knows Bob and Charlotte, it's highly likely that Bob and Charlotte also know each other.

We show that this natural instinct to form close-knit social networks operates in cyberspace as well and can be exploited to provide an effective and automated spam-filtering algorithm.

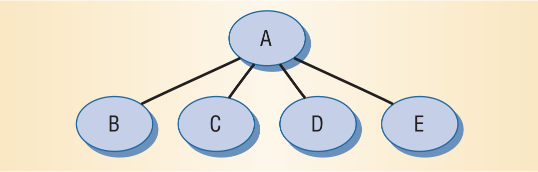


Figure 1. Star subnetwork. Subgraph represents e-mail interaction generated by a message sent from A to B and C and cc'd to D and E.

PERSONAL E-MAIL NETWORKS

Some researchers have constructed e-mail graphs based on e-mail address books[2] or complete e-mail logs of sets of users.[3-5] We based our network on the information available to just one user of an e-mail system—specifically, the headers of all of the e-mail messages in that user's inbox. Each e-mail header contains the sender's e-mail address, stored in the "From" field, and a list of recipient addresses, stored in the "To" and "Cc" fields.

To construct a personal e-mail network, we first created nodes representing all of the addresses appearing in all of the e-mail headers of the messages in the user's inbox. We added edges between pairs of addresses appearing in the same header—that is, addresses of individuals who have communicated via the user. For example, suppose Alice sends a message "To" Bob and Charlotte, with a "Cc" to David and Eve. We represented this e-mail interaction using a star subnetwork, as Figure 1 illustrates.

0018-9162/05/$20.00 © 2005 IEEE

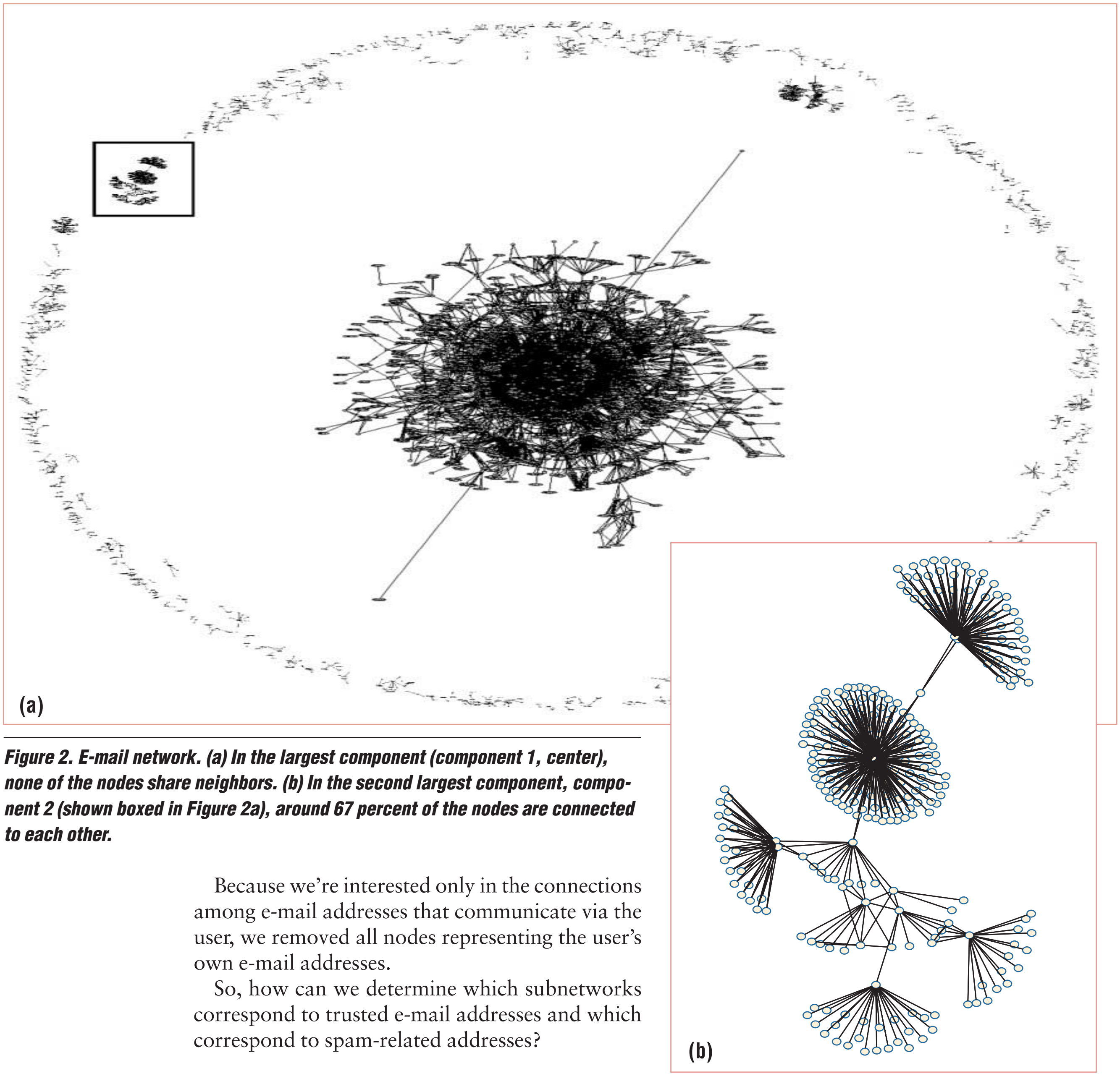


Figure 2. E-mail network. (a) In the largest component (component 1, center), none of the nodes share neighbors. (b) In the second largest component, component 2 (shown boxed in Figure 2a), around 67 percent of the nodes are connected to each other.

Because we're interested only in the connections among e-mail addresses that communicate via the user, we removed all nodes representing the user's own e-mail addresses.

So, how can we determine which subnetworks correspond to trusted e-mail addresses and which correspond to spam-related addresses?

SOCIAL NETWORKS AND WHITE LISTS

Many recent studies have identified quantitative measures of a community's closeness and have used these measures to distinguish empirically observed social networks from less-known, nonsocial networks.[4,6,7]

A social network's most distinctive property is the tendency to cluster. For example, if Alice knows Bob and Eve in a social network, Bob is considerably more likely to know Eve than in a random network with similar degree distribution.

To define a qualitative expression for a network's clustering coefficient, we begin by counting all pairs of nodes that are part of a wedge—that is, each node in the pair has a direct edge to a third node.

According to the intuitive notion of clustering, in a graph with a high clustering coefficient, many of these pairs will also be connected by an edge—that is, many of the wedges also form triangles. Hence, we can express the clustering coefficient—sometimes called transitivity—C of a graph as:

$$C = \frac{3 \times \text{(number of triangles in the graph)}}{\text{number of wedges}} \quad (1)$$

This expression should provide an idea of the clustering coefficient's physical meaning in social networks.

We use a quantitative definition of the clustering coefficient that involves counting the fraction of a node's neighbors that are also each other's neighbors.

Specifically, a node with degree k_i has k_i neighbors. Each k_i neighbor is potentially connected to the others. A total of $k_i(k_i - 1) = 2$ possible connections exists between the neighbors. Counting the number of connections E_i and dividing by $k_i(k_i - 1) = 2$ gives us the node's clustering coefficient.

Because the quantity for nodes of degree 1 is undefined, we only count nodes of degree greater than 1. The clustering coefficient for the entire graph is the average of the clustering coefficient for each node (of degree greater than 1)

$$C = \frac{1}{N_2} \sum_i \frac{2E_i}{k_i\,(k_i - 1)} \qquad (2)$$

where N_2 is the total number of network nodes of degree 2 or greater.

Other researchers have applied this metric to e-mail graphs and found that the clustering coefficient is more than 10 times larger than we would expect from a random graph with the same degree distribution.[4]

To demonstrate how to use a clustering coefficient to distinguish between spam and nonspam e-mail, consider the connected components 1 and 2 in the personal e-mail network consisting of 5,486 messages that is depicted in Figure 2. Interestingly, and perhaps contrary to intuition, the largest connected component in this particular network corresponds to spam-related e-mail.

Component 1 (Figure 2a) has a clustering coefficient of 0: Exactly zero nodes share neighbors with any of their neighbors. On the other hand, component 2 (Figure 2b), which is smaller in size, has a clustering coefficient of 0.67: Around 67 percent of each node's neighbors are connected to each other.

Figure 3, a subgraph of component 1, and Figure 4, a subgraph of component 2, show the relative incidence of the triangle structures that characterize close-knit communities.

Given that social networks have high clustering coefficients, we can be confident that the e-mail addresses in component 2 are part of the user's cyberspace social network. Thus, we can classify any e-mail with nodes from the second component in its header as nonspam. The e-mail addresses associated with these nodes comprise the user's *white list*.

BLACK LISTS AND SPAM COMPONENT FORMATION

If we can likewise conclude that spam generates the first component, which has a low clustering coefficient, we can label any e-mail sent or coreceived by a node inside the first component as spam.

Indeed, a detailed bookkeeping of the inbox shows that the e-mail addresses in component 1 are always related to spam, just as those in component 2 are always part of the user's social network. In other words, the e-mail addresses in the first component comprise a *black list*.

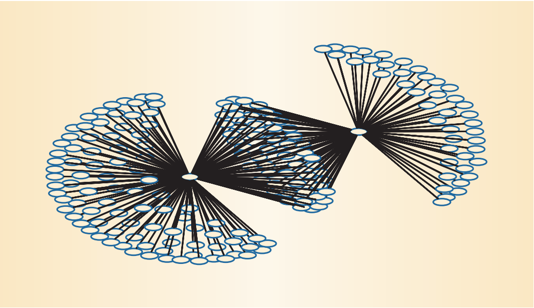


Figure 3. Subgraph of a spam component. (a) Two spammers share many corecipients (middle nodes). In this subgraph, no node shares a neighbor with any of its neighbors.

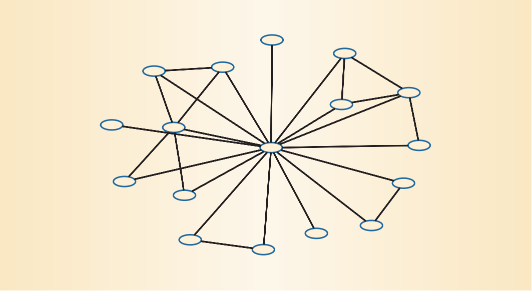


Figure 4. Subgraph of a nonspam component. The nonspam graph shows a higher incidence of triangle structures (neighbors sharing neighbors) than the spam subgraph.

Although we can expect networks of friends to have a high clustering coefficient, it is important to understand why a large subnetwork with a low clustering coefficient is likely to be spam-induced.

A careful analysis of component 1 reveals that it was created by a *dictionary attack*, a common spamming technique in which spammers send messages to corecipients sorted by e-mail address. For example, adam@example.com will likely have corecipients with alphabetically similar e-mail addresses, such as arthur@example.com, alex@example.com, and avid@example.com. Because of sorted recipient lists, corecipients are often repeated in different spam e-mail messages, causing the disconnected spam components to merge into larger components.

The spam messages form a bipartite graph, with two types of nodes representing spammers and spam recipients. Because the spammers don't spam each other, and the corecipients of spam messages don't know each other, this graph will always have a clustering coefficient of 0.

To determine how quickly the spammer components will merge, we can examine the probability that two different spammers send messages to the same corecipient.

Figure 5 shows the complementary cumulative distribution function (CCDF) of the number of corecipients per spam message. In this data, the average number of recipients of a spam message is 3.87. As a model, we assume that each spammer uses a "From" address only once and sends the spam to l recipients, which the spammer chooses at random. Based on our data, we choose $l \approx 3.87$.

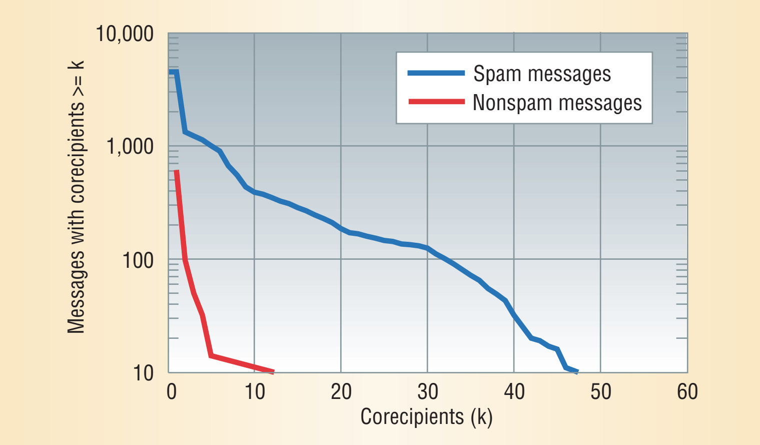


***Figure 5.* Complementary cumulative distribution function (CCDF) of the number of corecipients per message for spam and nonspam messages. The x-axis shows the number of corecipients, and the y-axis shows the number of messages with at least that number of corecipients. The mean number of corecipients for the spam messages is 3.87; for nonspam, the number is 1.71.**

The model assumes that k e-mail addresses are near the user's address in the sorted address space.

To solve for the size of the largest connected component, we define S_i as the size of the largest connected component after i messages. The probability that each recipient is already in the largest component is $S_i = k/q$. The probability that m recipients are in the largest component is

$$p_m = \binom{l}{l-m}(1-q)^{l-m}q^m$$

Combining this in a rate equation, we find

$$\begin{aligned} S_{i+1} &= \sum_{m=0}^{l} p_m\,(S_i + (l-m)) - p_0 l \\ &= S_i + (1-p_0)l - ql \\ &= S_i - \frac{l}{k}S_i + (1-(1-q)^l)l \end{aligned}$$

$$\frac{dS_i}{di} \approx (1-(1-q)^l)l - \frac{l}{k}S_i$$

We approximate this equation in two regimes—unsaturated ($q << 1$) and saturated ($q \approx 1$)—to get two different solutions. In the unsaturated regime,

$$\begin{aligned} \frac{dS_i}{di} &\approx (1-(1-q)^l)l - \frac{l}{k}S_i \\ &\approx l^2 q - \frac{l}{k}S_i \\ &= \frac{l(l-1)}{k}S_i \\ \Rightarrow S_i &= le^{\frac{l(l-1)}{k}i} \end{aligned}$$

In the saturated regime,

$$\begin{aligned} \frac{dS_i}{di} &\approx (1-(1-q)^l)l - \frac{l}{k}S_i \\ &\approx l - \frac{l}{k}S_i \\ \Rightarrow S_i &= k(1-e^{-\frac{1}{k}i}) + le^{-\frac{1}{k}i} \end{aligned}$$

Clearly, after $O(k/l^2)$ messages, the spam components should start to join. This analysis underestimates the joining rate because it assumes that a message only joins with the largest component and never adds more than one component, which clearly only increases the rate that the spam components grow in size.

We've also ignored the fact that the nearer the addresses are by alphabetical measure, the more likely they are to be corecipients of an e-mail message. Instead, we approximate that k nearby addresses exist and that the spammer selects them randomly.

GRAY LISTS AND AMBIGUOUS SUBGRAPHS

We can also use this scheme to classify all components of a user's personal network.

For each component, we note the size, maximum degree (k_{max}), and clustering coefficient. We expect a minimum number of nodes for which we can reliably measure a component's clustering coefficient and exclude components smaller than this cutoff size (S_{min}) from our classification scheme. We also know that graphs with power-law exponents greater than or equal to –2.0 will have a maximum degree on the order of the graph's size.[6] Like previous work,[4,5] we find degree distributions with exponents greater than –2.0.

We use this fact to introduce another cutoff parameter. To limit a single node's impact on the statistics, we disregard all components with a clustering coefficient equal to 0 and (k_{max} + 1)/size greater than K_{frac}. We assume the remaining components with clustering coefficients less than a critical value C_{min} are spam components and write all nodes in these components to the black list.

If a component's clustering coefficient is greater than C_{max}, we write all nodes to the white list. In rare cases, we include a component with a clustering coefficient between C_{min} and C_{max}, as we discuss later.

To choose the cutoff parameters S_{min} and K_{frac}, we used several criteria. Single messages can make isolated components, which are difficult to classify a priori, because every message with k corecipients can create a component with a clustering coefficient equal to 0.0 and size k. Setting K_{frac} to less than 1.0 (0.6 to 0.8 works well in practice) ensures that the algorithm won't consider a component from a single message.

A more direct route (but one that can't be realized using purely graph-theoretic methods) is to only consider components formed by N or more messages. The size cutoff should come from a message's expected number of recipients. Setting this far above the mean also ensures that each component has several messages. For the data we examined, a cutoff size of 10 to 20 messages seemed to work well.

Choosing C_{min} and C_{max} involved another set of criteria. Although we expect $C = 0$ for spam components, in our data, as in previous studies,[4,6] the clustering coefficient of the social graphs was an order of magnitude larger than we would expect for a random graph with the same degree distribution. We found that $C_{min} = 0.01$ and $C_{max} = 0.1$ produced excellent results.

Further research on personal e-mail networks will help improve the methods for choosing parameters. This research will provide a better understanding of the statistical properties of these networks over a larger set of users.

Purely by chance, a spammer might send a spam message to a user and one of his or her friends (we assume here that spammers don't know the e-mail addresses of a user's friends). This event will be rare because we imagine there is a very small probability that each corecipient on a spam message is a friend. Hence, spam components usually stay disconnected from friend components.

The possibility exists, however, that a spam message will have a corecipient in common with a nonspam message. The cross-component connections are most likely in spam components with a large number of corecipients. This means that chance connections will result in a nonspam component joined with a large spam component through a small number of edges (the chance corecipient connections).

From a graph-theoretic perspective, this situation corresponds to two connected communities that have few edges between them and different clustering coefficients. Large components with intermediate clustering coefficients typically signal these cases. For example, if the component size is large but the clustering coefficient is less than C_{max}, we can assume a joined spam component.

Edge betweenness is a proposed metric for identifying edges between communities.[3,8,9] The betweenness of an edge within a network is the number of shortest paths between all of the pairs of nodes in the network that include the edge.

All paths linking the nodes in the spam community to nodes in the nonspam community in a joined component will include one of the edges corresponding to the chance connections between the two communities. Therefore, these edges will have a much higher betweenness than the edges connecting members in the same community. Therefore, we split joined components into two communities by removing the edges with the highest betweenness until we have two distinct components.

Table 1. Algorithm results for three data sets. (Data set 1 is from a six-week period; sets 2 and 3 are from five-week periods.)

Data	Black list	White list	Gray list	Total
Spam 1	1,664	0	2,841	4,505
Nonspam 1	0	331	282	613
Spam 2	2,988	0	1,142	4,130
Nonspam 2	0	66	215	281
Spam 3	785	0	297	1,082
Nonspam 3	0	88	461	549

Newman and Girvan's algorithm[8] uses the same step; however, we don't execute the step on components with a high clustering coefficient. In fact, Newman and Girvan's community-finding algorithm tends to cut these e-mail graphs into many communities, whereas we're only interested in finding the split between spammers and nonspammers.

Although we rarely need the separation technique, it's a robust method for separating joined components, as long as spammers don't have access to the user's white list. For example, in our analysis of a data set containing all of a user's saved e-mail messages over a period of three years, we found only one example in which there was an edge between a spam and nonspam community. We used the separation technique to remove this edge.

In the absence of a complete experimentally verified theory for how personal e-mail networks are formed, we must rely on empirical observations of such networks. Although no theory exists that requires the graph parameters we assume here, some models based on underlying community structure have sought to explain the high clustering coefficient.[6,7] Access to mail headers of many different users might help researchers find a theory to explain the mechanisms giving rise to the properties our algorithm uses.

Presumably, all users are subject to a similar spammer environment—that is, spammers attack all users similarly. As such, until spammers change their tactics, the algorithm's black-listing power should apply to all users. If some users' social structures don't match those previously measured—specifically, if their personal e-mail networks exhibit extremely low clustering—the algorithm might misclassify nonspam messages as spam (false positives).

EXPERIMENTAL RESULTS

Table 1 shows the results of using the algorithm on three data sets covering three users over five- and six-week periods. Averaging across users, 34 percent of the nonspam is on the white list, 56 percent of the spam is on the black list, and 47 percent

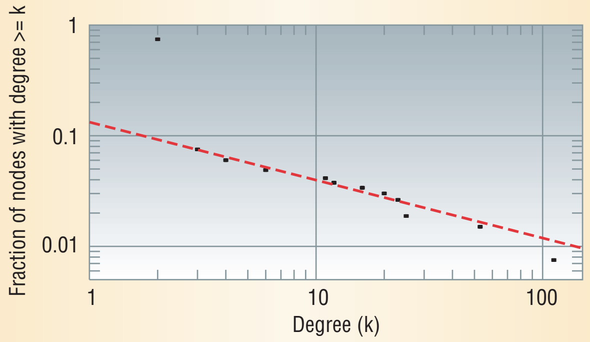


Figure 6. Degree CCDF for the largest nonspam component. The line corresponds to $p_k \propto k^{-1.52}$. Nodes with degrees 1 and 2 don't fit the tail's power law. This occurs because the personal e-mail network construction adds an edge only when the user is a recipient and doesn't see the edges that would exist between two users if it considered both users' e-mail traffic.

of the messages are on the gray list. Thus, the algorithm correctly identifies the message about half the time; in the remaining half, it can't classify the message. Over the test data, it correctly classifies all messages.

To improve our algorithm, we added more graph-theoretic parameters to the classification scheme. Figure 6 shows the CCDF of the largest nonspam component. The degree distribution for the CCDF's tail is $p_k \propto k^{-1.52}$. The nodes with degrees 1 and 2 don't fit the tail's power law. Making a subgraph of the entire e-mail network from a single user's view produces a degree distribution consistent with work based on multiple users' e-mail.[4,5]

Previous studies found degree distributions following power laws with exponents from –1.3 to –1.8. In our data sets, the spam components had power laws with exponents between –1.8 and –2.0.

Thus, because spam components seem to have unusually high power-law exponents, we might be able to use a component's degree distribution to improve the distinguishing power and thus reduce the likelihood of error.

USING THE ALGORITHM

Our e-mail-network-based spam-filtering algorithm automatically constructs white lists with a 34 percent success rate, black lists with a 56 percent success rate with no false classifications, and leaves 47 percent of the e-mail unclassified. It achieves these hit rates without any user intervention.

Table 2. Results of a Bayesian classifier trained on the data from the graph algorithm.

Data	Percent classified as spam	Percent classified as nonspam
Spam 1	80.8	19.2
Nonspam 1	0.3	99.7
Spam 2	97.6	2.4
Nonspam 2	48.8	51.2
Spam 3	95.6	4.4
Nonspam 3	4.7	95.3

Because the only information the algorithm needs is available in the user's e-mail headers, we can easily implement it at various points in the filtering process.

Existing white-list systems use a list of acceptable e-mail addresses or mail servers and assume that all senders not on the list are spam. Similarly, existing black list systems block mail from e-mail addresses or mail servers on a list, and assume all other senders are nonspam. These lists are currently created in an ad hoc manner, based on global prior knowledge and user feedback.

Thus, we can easily integrate our technique, based purely on graph-theoretic methods, with existing methods. For example, administrators of large e-mail servers, such as corporate e-mail servers or Internet service providers, can use our algorithm because they can generate a personal e-mail network for each user as their mail servers receive mail. The ability to generate white lists and black lists for all users will greatly improve the central mail servers' ability to reduce the number of spam messages that reach end users.

The algorithm is also virtually immune to false positives, suggesting that our method can also significantly increase the ease of use of content-based antispam tools, which classify e-mail as spam or nonspam based on content rather than the sender's address.

To illustrate how we integrate our graph-based algorithm with a learning algorithm, we created a simple example using the CRM114 Bayesian classifier.[10]

When performing Bayesian learning,[11,12] it's important not to overtrain and, in the case of e-mail, not to prefer false positives (spam misclassified as nonspam) to false negatives (nonspam misclassified as spam).

We had the classifier learn all of the messages classified as nonspam, and we had it classify each spam message. If the classifier misclassified one of these messages, we set it to learn the message as spam. In other words, we trained the classifier on all of the white list, but only used train-on-error for the black list. It eventually classified all messages as spam or nonspam. Table 2 lists the results.

In this case, data sets 1 and 3 performed rather well, but data set 2 performed quite poorly. We believe this is because data set 2 had the fewest nonspam messages to learn and also the worst ratio of spam to nonspam: 14.7, compared to 7.35 and 1.97 for data sets 1 and 3.

These preliminary results are only estimates of possible performance. Future research should produce more sophisticated training schemes to better combine our algorithm with content-based learning methods.

COUNTERMEASURES

Spammers, of course, will try to defeat antispam tools. A few countermeasures might foil our algorithm. The most obvious countermeasure is to never use multiple recipients in a spam message's "To" or "Cc" headers. This would make the spam components isolated nodes in our graph, which the algorithm would disregard when constructing the black list. However, this wouldn't impact users' ability to automatically generate white lists, which are extremely valuable in improving the accuracy of content-based filtering systems.

Spammers could also attempt to make the algorithm misclassify them as nonspammers. For example, a spammer could use spyware to try to learn e-mail addresses for each user's friends. The spammer could then include the user's friends as corecipients of a spam message, thus posing as a member of the user's social network. If spammers can impersonate members of a white list, however, they can damage the effectiveness of any white-listing scheme, not just our algorithm.

In another countermeasure, spammers construct an artificial social network and attempt to have their messages white-listed. The spammer then sends e-mail messages in which the "From," "To," and "Cc" headers match a real social network's structure. The spammer could include the spam targets—those users for whom the spam is intended—as "Bcc" recipients, which don't show up in the headers. Although our algorithm could label this component as part of the white list, this countermeasure itself could be countered. (Recall that we use only incoming mail headers, not outgoing mail headers—that is, the mail sent by algorithm users.)

By using outgoing mail headers, we can disregard any nonspam components to which the user has never sent any mail. Although this countermeasure can damage the user's ability to construct black lists, it doesn't appear to damage the user's ability to construct white lists. Hence, at this time, we see no countermeasures that could destroy the algorithm's white-listing aspect.

Many areas are open for improving our spam-filtering algorithm. Most obviously, we need to include more component parameters to distinguish spam from nonspam. An algorithm that incorporates more parameters could potentially reduce the probability of misclassification even further.

Unfortunately, significantly decreasing the gray list's size doesn't appear to be a promising option because the gray list components are very small and thus inhabit a small parameter space. Hence, the ability to strongly distinguish between gray-listed components appears to be beyond purely graph-based techniques.

Using cryptographic white lists, in which a user only accepts messages cryptographically signed by authenticated keys, is clearly a solution to the spam problem. Unfortunately, without an infrastructure to make the scheme accessible to most end users, its immediate potential for widespread use is highly questionable. Until this infrastructure is in place, end users must rely on less-perfect solutions, and they will benefit from any effort that makes these solutions more user friendly and easier for mail servers and ISPs to broadly distribute.

As long as spammers continue to adapt their strategies for defeating antispam tools, improving these tools is a subject that will warrant further study. However, even if the spam problem is solved, our e-mail network tool will become increasingly useful. In particular, using a personal e-mail network has the potential to capture the community structure in cyberspace. It's possible that better e-mail message management can be achieved if e-mail clients are aware of the user's various social groups. Our scheme for generating personal e-mail networks could provide such information without any changes to Internet e-mail protocols. ■

Acknowledgments

We thank Jesse Bridgewater for use of his e-mail, Joseph Kong and Kamen Lozev for their comments, and Julie Walters for use of her e-mail and her helpful suggestions. US National Science Foundation grant ITR0326605 partially supported this work.

References

1. D. Fallows, *Spam: How It Is Hurting E-Mail and Degrading Life on the Internet*, tech. report, Pew Internet and American Life Project, Oct. 2003; http://www.pewinternet.org/reports/toc.asp?Report=102.
2. M.E.J. Newman, S. Forrest, and J. Balthrop, "E-Mail Networks and the Spread of Computer Viruses," *Physical Rev. E*, vol. 66, no. 035101, 2002.
3. J.R. Tyler, D.M. Wilkinson, and B.A. Huberman, "E-Mail as Spectroscopy: Automated Discovery of Com-

munity Structure within Organizations," preprint, http://xxx.lanl.gov/abs/cond-mat/0303264.

4. H. Ebel, L.-I. Mielsch, and S. Bornholdt, "Scale-Free Topology of E-Mail Networks," *Physical Rev. E*, vol. 66, no. 035103, 2002.
5. G. Caldarelli, F. Coccetti, and P.D.L. Rios, "Preferential Exchange: Strengthening Connections in Complex Networks," *Physical Rev. E*, vol. 70, no. 027102, 2004.
6. M.E.J. Newman and J. Park, "Why Social Networks Are Different from Other Types of Networks," *Physical Rev. E.*, vol. 68, no. 036122, 2003.
7. M.E.J. Newman, "Assortative Mixing in Networks," *Physical Rev. Letters*, vol. 89, no. 208701, 2002.
8. M.E.J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical Rev. E*, vol. 69, no. 026114, 2004.
9. M.E.J. Newman, "Fast Algorithm for Detecting Community Structure in Networks," *Physical Rev. E*, vol. 69, no. 066133, 2004.
10. W.S. Yerazunis, "Sparse Binary Polynomial Hashing and the CRM114 Discriminator," presented at the MIT Spam Conf., 2003; http://crm114.sourceforge.net/CRM114_paper.html.
11. M. Sahami et al., "A Bayesian Approach to Filtering Junk E-Mail," *Learning for Text Categorization: Papers from the 1998 Workshop*, AAAI tech. report WS-98-05, AAAI Press, 1998.
12. D. Heckerman, "A Tutorial on Learning with Bayesian Networks," *Learning in Graphical Models*, M. Jordan, ed., MIT Press, 1998.

***P. Oscar Boykin** is an assistant professor of electrical and computer engineering at the University of Florida. His research interests include quantum cryptography, quantum information and computation, peer-to-peer computing, and neural coding. Boykin received a PhD in physics from the University of California, Los Angeles. Contact him at boykin@ece.ufl.edu.*

***Vwani P. Roychowdhury** is a professor of electrical engineering at the University of California, Los Angeles. His research interests include models of computation, including parallel and distributed processing systems, quantum computation, and information processing; adaptive and learning algorithms; nonparametric methods and algorithms for large-scale information processing; and information theory. Roychowdhury received a PhD in electrical engineering from Stanford University. Contact him at vwani@ee.ucla.edu.*

ADVERTISER / PRODUCT INDEX

APRIL 2005

Advertiser / Product	Page Number
Addison-Wesley	**Cover 4**
Borland Software	75
DE Shaw & Company	**70**
IBM	75
IEEE Computer Society Membership	4-6
NextCom	75
Parasoft	75
Themis Computer	75
WinHEC 2005	**Cover 2**
Classified Advertising	***70***

Boldface denotes advertisements in this issue.

FUTURE ISSUE

May 2005

Virtualization

Virtualization technologies are becoming pervasive in commercial products. These technologies decouple user-perceived behavior of hardware and software resources from their physical implementation. Examples include virtual machines that enable multiple operating systems to coexist on the same physical machine as well as mechanisms that virtualize data and architecture-independent applications that can be used anywhere, anytime.

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 0161
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)
Robert Zwick
Phone: +1 212 419 7765
Fax: +1 212 419 7570
Email: r.zwick@ieee.org

Northwest (product)
Peter D. Scott
Phone: +1 415 421 7950
Fax: +1 415 398 4156
Email: peterd@pscottassoc.com

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)
Josh Mayer
Phone: +1 972 423 5507
Fax: +1 972 423 6858
Email: jm.ieeemedia@ieee.org

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: +1 212 419 7766
Fax: +1 212 419 7589
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

D. E. Shaw Research and Development

Systems Architects and ASIC Engineers

Specialized Supercomputer for Computational Drug Design

Extraordinarily gifted systems architects and ASIC design and verification engineers are sought to participate in the development of a special-purpose supercomputer designed to fundamentally transform the process of drug discovery within the pharmaceutical industry. This early-stage, rapidly growing project is being financed by the D. E. Shaw group, an investment and technology development firm with more than US $11 billion in aggregate capital. The project was initiated by the firm's founder, Dr. David E. Shaw, and operates under his direct scientific leadership.

This project aims to combine an innovative, massively parallel architecture incorporating 90-nanometer "system on a chip" ASICs with novel mathematical techniques and groundbreaking algorithmic advances in computational biochemistry to direct unprecedented computational power toward the solution of key scientific and technical problems in the field of molecular design. Successful candidates will be working closely with a number of the world's leading computational chemists and biologists, and will have the opportunity not only to participate in an exciting entrepreneurial venture with considerable economic potential, but to make fundamental contributions within the fields of biology, chemistry, and medicine.

The candidates we seek will be unusually intelligent and accomplished, with a demonstrated ability to design and implement complex, high-performance hardware solutions based on the latest semi-custom technologies. We are prepared to reward exceptionally well-qualified individuals with above-market compensation.

Please send resume, along with GPAs, standardized test scores (SAT, GRE), and compensation history, to ieeecomputer@desrad.deshaw.com.

D. E. Shaw Research and Development, LLC does not discriminate in employment matters on the basis of race, color, religion, gender, national origin, age, military service eligibility, veteran status, sexual orientation, marital status, disability, or any other protected class.

D E Shaw & Co

TECH SUPPORT ANALYST for clients. of Noviant in NYC. 6 months experience & experience with AutoCad; BS Computer Science or Comp. Aided Dsgn. Resumes to: K. Wang, Noviant, 32 Broad St., Ste. 509, NY, NY 10004.

FINANCE - NY, NY: 2 positions available: Chief Fin. Officer responsible for overall financial activity for the parent company as well as an American and Israeli subsidiary. Req. BS in Acctg/Finance/Econ + related exp. Must be fluent in Hebrew. Chief Tech. Officer to plan & direct development of software systems, hardware deployment, & other technology related projects. Req. MBA or MS in Comp Sci. + exp. Must be fluent in Hebrew. Resumes to: HR Director, XMPie, 41 Madison Ave, NY, NY 10010.

COMPUTER PROGRAMMER. Texmate Inc. in Vista, California needs a computer programmer to write programs; update and modify, expand and repair, existing software programs and to develop programs for maintaining computer systems with operator interfaces in English and Chinese characters; prepare instructions for computer operators in English and Chinese, master's degree in computer science related field, six months experience and fluency in Chinese Mandarin are required. Prevailing salary and working environment. Please fax resume to (760) 598-9828.

INTERNATIONAL CHRISTIAN UNIVERSITY, Division of Natural Sciences invites applications for a tenure track position in Information Science at the rank of Assistant Professor or above with responsibility for teaching undergraduate and graduate courses in Computer Science. Ph.D. in the field of specialization is required, publications and teaching experience preferred. Expertise in administering computer systems and networks is strongly desirable. Fluency in English is required. Knowledge or willingness to learn Japanese is preferred. Applicants should send a cover letter, CV, and names and contact information for three academic references. Deadline for applications is June 10, 2005. The initial date of employment is April 1, 2006. Please send documents to: Chair, Division of Natural Sciences, 3-10-2 Osawa, Mitaka, Tokyo 181-8585, JAPAN. For more information, see www.icu.ac.jp or email to: cs-recruit@icu.ac.jp.

SOFTWARE ENGINEER (WEB BASED APPLICATIONS) - R'sch, dsgn, & dev. web based comp. apps. Using ATG Dynamo, J2EE, EJB, JavaScript, & XML Req'd: BS & 3 yrs. exp. Resumes: Kaplan, Inc. 888 7th Ave., NY, NY 10106. Attn: P. Torres.

Game Programming Gems 5, Kim Pallister, editor. This fifth volume in a series provides a roadmap through the vast array of development challenges facing today's game programmers. The book's 62 newly unearthed gems offer practical insights and techniques that can solve current problems and help inspire future games.

The academic and industry contributors cover general programming topics such as parsing text data in games, using templates for reflection in C++, and a generic pager. The mathematics topics include geometric algebra for computer graphics, minimal acceleration hermite curves, and minimal numerical approximation. Other topics covered in the book include artifical intelligence, physics, and graphics. Network and multiplayer topics such as keeping a massively multiplayer online game live and persistent and audio topics such as multithreaded audio and using 3D surfaces as audio emitters are also discussed.

Charles River Media; www.charlesriver.com; 1-58450-352-1; 791 pp.; $69.95.

Fuzzy Database Modeling with XML, Zongmin Ma. Research into fuzzy conceptual models and object-oriented databases, in addition to fuzzy relational database models, has focused on applying the databases to the distributed information systems environment. Because developers commonly employ databases to store and manipulate XML data, they need additional requirements to model fuzzy information with XML.

The outgrowth of the author's recently conducted research, this book introduces state-of-the-art database research information, while at the same time providing a useful tool for the information technology professional faced with a nontraditional application that defeats conventional approaches.

Springer; www.springeronline.com; 0-327-24248-1; 216 pp.; $89.95.

Debugging by Thinking: A Multidisciplinary Approach, Robert Charles Metzger. This book applies the wisdom of six disciplines—logic, mathematics, psychology, safety analysis, computer science, and engineering—to the problem of debugging. It uses the methods of literary detectives, mathematical problem-solving techniques, the results of research into cognitive psychology, the root-cause analyses of safety experts, the compiler analyses of computer science, and modern engineering processes to define a systematic approach to identifying and correcting software errors.

The author also provides examples in Java and C++; complete source code that show actual bugs, rather than contrived examples; and a thought-process diary that shows how he resolved problems as they occurred.

Elsevier Academic Press; http://books.elsevier.com/; 1-55558-307-5; 600 pp.; $49.95.

Game Coding Complete, 2nd ed., Mike McShaffry. This substantially expanded edition provides new material on game interface design, 3D programming, network and multiplayer gaming, sound effects and music, programming event handlers and scripts, and game physics and AI. The author uses his experience as a lead programmer for Origin Systems, Microsoft, Ion Storm, and Breakaway Games to illustrate real-world game programming insight and solutions.

Second edition highlights include using C++ and DirectX 9 to present specific programming concepts, a comprehensive discussion of game code architecture, programming insights for both console and PC developers, and expanded 3D programming coverage.

Paraglyph Press/O'Reilly; www.oreilly.com; 1-932111-91-3; 936 pp.; $44.99.

Hardening Windows Systems, Roberta Bragg. The author urges a proactive approach to network security that involves hardening Windows systems against attacks before they occur. This hands-on resource provides concrete steps that can be taken immediately as well as ongoing actions to ensure long-term security.

The book provides complete details on how to systematically harden a network from the ground up, whether it consists of a single Windows server or a hundred. It also provides strategies for getting company-wide support for the security plan and covers Windows 95/98/NT 4.0/2000/XP and Windows Server 2003.

The four-part hardening methodology starts with a checklist of immediate steps to take to lock down a system from further attack; a systematic approach to hardening the enterprise from the top down that focuses on authentication, access controls, borders, logical security boundaries, communications, storage, and administrative authority; an ongoing monitoring and assessment plan to keep the network secure, including patch management and auditing; and strategies for getting budget approval, management buy-in, and employee cooperation for security programs.

McGraw-Hill Osborne; www.osborne.com; 0-07-225354-1; 544 pp.; $39.99.

Editor: Michael J. Lutz, Rochester Institute of Technology, Rochester, NY; mikelutz@mail.rit.edu. Send press releases and new books to *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; fax +1 714 821 4010; newbooks@computer.org.

CALLS FOR IEEE CS PUBLICATIONS

IEEE Pervasive Computing magazine seeks articles for an October-December special issue on methods and tools for rapid prototyping, a central instrument for research and development in ubiquitous computing. *PvC* welcomes contributions that approach this theme from angles including hardware platforms and support for custom developments, software toolkits and frameworks, architectures and conceptual abstractions, experiments and evaluation based on prototypes, and methods for prototyping.

Submissions are due by **1 May.** To view the complete call for papers, including author guidelines, visit www.computer.org/pervasive/edcal0405.htm.

OTHER CALLS

HiPC 2005, 12th IEEE Int'l Conf. on High-Performance Computing, 18-21 Dec., Goa, India. Papers due **2 May.** www.hipc.org/hipc2005/papers.html

ICDM 2005, 5th IEEE Int'l Conf. on Data Mining, 26-30 Nov., New Orleans. Papers due **1 June.** www.cacs.louisiana.edu/~icdm05/cfp.html

ICDE 2006, 22nd IEEE Int'l Conf. on Data Eng., 3-7 Apr. 2006, Atlanta. Papers due **14 June.** http://icde06.cc.gatech.edu/cfp.html

ICTAI 2005, 17th Int'l Conf. on Tools with AI, 14-16 Nov., Hong Kong. Papers due **15 June.** http://ictai05.ust.hk/frame/fm-callpaper.html

Key West 2006, IEEE Key West Computer Elements Workshop, 8-11 Jan. 2006, Key West, Fla. Proposals due **24 June.** www.unf.edu/ccec/ieee/IEEE-2006-KeyWest-Call.pdf

Submission Instructions

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting. Complete instructions for submitting conference or call listings are available at www.computer.org/conferences/submission.htm.

A more complete listing of upcoming computer-related confeences is available at www.computer.org/conferences/.

CALENDAR

MAY 2005

1 May: DBT 2005, IEEE Int'l Workshop on Current & Defect-Based Testing (with VTS), Rancho Mirage, Calif. www.cs.colostate.edu/~malaiya/dbt.html

1 May: NanoArch 2005, IEEE Int'l Workshop on Design & Test of Defect-Tolerant Nanoscale Architectures (with VTS), Rancho Mirage, Calif. www.nanoarch.org/

1-5 May: VTS 2005, 23rd IEEE VLSI Test Symp., Rancho Mirage, Calif. www.tttc-vts.org/

8-11 May: SP 2005, IEEE Symp. on Security & Privacy, Berkeley, Calif. www.ieee-security.org/TC/SP2005/

9-12 May: CCGrid 2005, 5th IEEE Int'l Symp. on Cluster Computing & the Grid, Cardiff, UK. www.cs.cf.ac.uk/ccgrid2005/

10-13 May: SPI 2005, IEEE 9th Workshop on Signal Propagation on Interconnects, Garmisch-Partenkirchen, Germany. www.spi.uni-hannover.de/

11-13 May: NATW 2005, IEEE 14th North Atlantic Test Workshop, Essex Junction, Vt. www.ee.duke.edu/NATW/

15-16 May: IWPC 2005, 13th Int'l Workshop on Program Comprehension (with ICSE), St. Louis. www.ieee-iwpc.org/iwpc2005/

15-21 May: ICSE 2005, 27th Int'l Conf. on Software Eng., St. Louis. www.cs.wustl.edu/icse05/Home/index.shtml

16-19 May: ISEE 2005, IEEE Int'l Symp. on Electronics & the Environment, New Orleans. www.regconnect.com/content/isee/

18-20 May: ISORC 2005, 8th IEEE Int'l Symp. on Object-Oriented Real-Time Distributed Computing, Seattle. http://shay.ecn.purdue.edu/~isorc05/

18-21 May: ISMVL 2005, 35th Int'l Symp. on Multiple-Valued Logic, Calgary, Canada. www.enel.ucalgary.ca/ISMVL2005/

22-25 May: ETS 2005, 10th European Test Symp., Tallinn, Estonia. http://deepthought.ttu.ee/ati/ETS/

25-26 May: EBTW 2005, European Board Test Workshop (with ETS 2005), Tallinn, Estonia. www.molesystems.com/EBTW05/

30-31 May: EMNETS-II 2005, 2nd IEEE Workshop on Embedded Networked Sensors, Sydney, Australia. www.cse.unsw.edu.au/~emnet/

JUNE 2005

1-3 June: PADS 2005, 19th ACM/IEEE/SCS Workshop on Principles of Advanced & Distributed Simulation, Monterey, Calif. www.pads-workshop.org/pads2005/index.html

4-8 June: ISCA 2005, 32nd Ann. Int'l Symp. on Computer Architecture, Madison, Wis. www.cs.wisc.edu/~isca2005/

5-8 June: SWTW 2005, Southwest Test Workshop, San Diego, Calif. www.swtest.org/

6-8 June: Policy 2005, IEEE 6th Int'l Workshop on Policies for Distributed Systems & Networks, Stockholm. www.policy-workshop.org/2005/

6-9 June: ICDCS 2005, 25th Int'l Conf. on Distributed Computing Systems, Columbus, Ohio. www.cse.ohio-state.edu/icdcs05/

7-11 June: JCDL 2005, IEEE/ACM Joint Conf. on Digital Libraries, Denver. www.jcdl2005.org/

12-13 June: MSE 2005, Int'l Conf. on Microelectronic Systems Education (with DAC), Anaheim, Calif. www.mseconference.org/

12-15 June: Complexity 2005, 20th Ann. IEEE Conf. on Computational Complexity, San Jose, Calif. www.computationalcomplexity.org/

13-15 June: WETICE 2005, 14th IEEE Int'l Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Linköping, Sweden. http://siplab.csee.wvu.edu/wetice05/

13-16 June: ICAC 2005, 2nd IEEE Int'l Conf. on Autonomic Computing, Seattle. www.autonomic-conference.org/

13-16 June: WOWMOM 2005, Int'l Symp. on A World of Wireless, Mobile, & Multimedia Networks, Taormina, Italy. http://cnd.iit.cnr.it/wowmom2005/

13-17 June: SMI 2005, Int'l Conf. on Shape Modeling & Applications, Cambridge, Mass. www.shapemodeling.org/

16-20 June: ICECCS 2005, Int'l Conf. on Eng. of Complex Computer Systems, Shanghai. www.cs.sjtu.edu.cn/iceccs2005/

19-24 June: Int'l Symp. on Emergence of Globally Distributed Data, Pula, Italy. www.storageconference.org/

20-22 June: CSFW 2005, 18th IEEE Computer Security Foundations Workshop, Aix-en-Provence, France. www.lif.univ-mrs.fr/CSFW18/

20-26 June: CVPR 2005, IEEE Int'l Conf. on Computer Vision & Pattern Recognition, San Diego, Calif. www.cs.duke.edu/cvpr2005/

22-24 June: CGI 2005, Computer Graphics Int'l Conf. & Workshops, Stony Brook, N.Y. www.cs.stonybrook.edu/~cgi05/

23-24 June: CBMS 2005, 18th IEEE Symp. on Computer-Based Medical Systems, Dublin, Ireland. www.cs.tcd.ie/research_groups/mlg/CBMS2005/index.html

26-29 June: VCEW 2005, IEEE Vail Computer Elements Workshop, Vail, Colo. www.unf.edu/ccec/ieee/vail_2005wkshps.html

26-29 June: LICS 2005, 20th Ann. IEEE Symp. on Logic in Computer Science, Chicago. http://homepages.inf.ed.ac.uk/als/lics/lics05/

27-29 June: IMSTW 2005, 11th Ann. Int'l Mixed-Signals Testing Workshop, Cannes, France. http://tima.imag.fr/conferences/IMSTW05/

27-29 June: ARITH-17, 17th IEEE Symp. on Computer Arithmetic, Cape Cod, Mass. http://arith17.polito.it/

27-29 June: CollaborateCom 2005, 1st IEEE Int'l Conf. on Collaborative Computing, Cape Cod, Mass. www.collaboratecom.org/

27-30 June: ISCC 2005, 10th IEEE Symp. on Computers & Communications, Cartagena, Spain. www.comsoc.org/iscc/2005/

28 June-1 July: DSN 2005, Int'l Conf. on Dependable Systems & Networks, Yokohama, Japan. www.dsn.org/

29-30 June: WTW 2005, 4th Workshop on Test of Wireless Circuits & Systems (with IMSTW), Cannes, France. http://resmiq.grm.polymtl.ca/WTW/2005/

30 June-1 July: DCOSS 2005, Int'l Conf. on Distributed Computing in Sensor Systems, Marina del Rey, Calif. www.dcoss.org/

JULY 2005

5-8 July: ICALT 2005, 5th IEEE Int'l Conf. on Advanced Learning Technologies, Kaohsiung, Taiwan. www.ask.iti.gr/icalt/2005/

6-8 July: ICME 2005, IEEE Int'l Conf. on Multimedia & Expo, Amsterdam. www.icme2005.com/

6-8 July: IOLTS 2005, 11th IEEE Int'l On-Line Testing Symp., Saint Rafael, France. http://tima.imag.fr/conferences/IOLTS/iolts05/Index.html

2005 IEEE International Conference on Advanced Learning Technologies

The fifth IEEE International Conference on Advanced Learning Technologies will bring together top professionals who are building the next generation of e-learning systems and technology-enhanced learning environments.

Conference organizers have solicited papers on topics that include educational modeling languages, socially intelligent agents, artificial intelligence tools for contextual learning, and mobile learning applications. Workshops on semantic Web technologies for e-learning and applications of tablet PCs in engineering education are also scheduled.

ICALT 2005, set for 5-8 July in Kaohsiung, Taiwan, is sponsored by the IEEE Computer Society in cooperation with the IEEE Technical Committee on Learning Technology.

For more conference details, including travel and venue information, visit www.ask.iti.gr/icalt/2005/.

11-14 July: ICPS 2005, IEEE Int'l Conf. on Pervasive Services, Santorini, Greece. www.icps2005.cs.ucr.edu/

11-14 July: MEMOCODE 2005, 3rd ACM-IEEE Int'l Conf. on Formal Methods and Models for Codesign, Verona, Italy. www.irisa.fr/manifestations/2005/MEMOCODE/

12-15 July: ICWS 2005, 3rd IEEE Int'l Conf. on Web Services, Orlando, Fla. http://conferences.computer.org/icws/2005/

12-15 July: SCC 2005, IEEE Int'l Conf. on Services Computing (with ICWS 2005), Orlando, Fla. http://conferences.computer.org/scc/2005/

18-19 July: WMCS 2005: 2nd IEEE Int'l. Workshop on Mobile Commerce and Services (with CEC-05), Munich. www.mobile.ifi.lmu.de/Conferences/wmcs05/

19-22 July: CEC 2005, 7th Int'l IEEE Conf. on E-Commerce Technology, Munich. http://cec05.in.tum.de/

20-21 July: WRTLT 2005, Workshop on RTL & High-Level Testing, Harbin, China. http://wrtlt05.hit.edu.cn/

20-22 July: ICPADS 2005, 11th Int'l Conf. on Parallel & Distributed Systems, Fukuoka, Japan. www.takilab.k.dendai.ac.jp/conf/icpads/2005/

23-25 July: ASAP 2005, IEEE 16th Int'l Conf. on Application-Specific Systems, Architectures, & Processors, Samos, Greece. www.ece.uvic.ca/asap2005/

24 July: CLADE 2005, Workshop on Challenges of Large Applications in Distributed Environments (with HPDC-14), Research Triangle Park, N.C. www.cs.umd.edu/CLADE2005/

24-27 July: HPDC-14, 14th IEEE Int'l Symp. on High-Performance Distributed Computing, Research Triangle Park, N.C. www.caip.rutgers.edu/hpdc2005/

27-29 July: NCA 2005, 4th IEEE Int'l Symp. on Network Computing & Applications, Cambridge, Mass. www.ieee-nca.org/

AUGUST 2005

2-4 Aug: ICCNMC 2005, Int'l Conf. on Computer Networks & Mobile Computing, Zhangjiajie, China. www.iccnmc.org/

4-5 Aug: MTDT 2005, IEEE Int'l Workshop on Memory Technology, Design, & Testing, Taipei, Taiwan. http://ats04.ee.nthu.edu.tw/~mtdt/

8-10 Aug: ICCI 2005, 4th IEEE Int'l Conf. on Cognitive Informatics, Irvine, Calif. www.enel.ucalgary.ca/ICCI2005/

8-11 Aug: CSB 2005, IEEE Computational Systems Bioinformatics Conf., Palo Alto, Calif. http://conferences.computer.org/bioinformatics/

17-19 Aug: RTCSA 2005, 11th IEEE Int'l Conf. on Embedded & Real-Time Computing Systems & Applications, Hong Kong. www.comp.hkbu.edu.hk/~rtcsa2005/

22-24 Aug: Tabletop 2005, IEEE Int'l Workshop on Horizontal Interactive Human-Computer Systems, Mawson Lakes, Australia. Contact bruce.thomas@unisa.edu.au.

29 Aug.-2 Sept: RE 2005, 13th IEEE Int'l Requirements Eng. Conf., Paris. http://crinfo.univ-paris1.fr/RE05/

SEPTEMBER 2005

7-9 Sept: SEFM 2005, 3rd IEEE Int'l Conf. on Software Eng. & Formal Methods, Koblenz, Germany. http://sefm2005.uni-koblenz.de/

12-14 Sept: IWCW 2005, 10th Int'l Workshop on Web Content Caching & Distribution, Sophia Antipolis, France. http://2005.iwcw.org/

15-16 Sept: AVSS 2005, Conf. on Advanced Video & Signal-Based Surveillance, Como, Italy. www-dsp.elet.polimi.it/avss2005/

18-21 Sept: CDVE 2005, 2nd Int'l Conf. on Cooperative Design, Visualization and Eng., Palma de Mallorca, Spain. www.cdve.org/

19-22 Sept: Metrics 2005, 11th IEEE Int'l Software Metrics Symp., Como, Italy. http://metrics2005.di.uniba.it/

19-22 Sept: WI-IAT 2005, IEEE/WIC/ACM Int'l Joint Conf. on Web Intelligence & Intelligent Agent Technology, Compiegne, France. www.comp.hkbu.edu.hk/WI05/

19-23 Sept: EDOC 2005, 9th Int'l Conf. on Enterprise Computing, Enschede, Netherlands. http://edoc2005.ctit.utwente.nl/

20-22 Sept: WRAC 2005, 2nd IEEE/NASA/IBM Workshop on Radical Agent Concepts, Greenbelt, Md. http://aaaprod.gsfc.nasa.gov/WRAC/home.cfm

21-24 Sept: VL/HCC 2005, IEEE Symp. on Visual Languages & Human-Centric Computing, Dallas. http://viscomp.utdallas.edu/vlhcc05/

25-30 Sept: ICSM 2005, 21st IEEE Int'l Conf. on Software Maintenance, Budapest. www.inf.u-szeged.hu/icsm2005/

26-29 Sept: MASCOTS 2005, Int'l Symp. on Modeling, Analysis, & Simulation of Computer & Telecomm. Systems, Atlanta. www.mascots-conference.org/

27-30 Sept: Cluster 2005, IEEE Int'l Conf. on Cluster Computing, Boston. www.cluster2005.org/

NextCom Consolidates Mobile Workstation and Server

FlexPCServer from NextCom is an all-in-one mobile server and graphics workstation designed to match the performance and flexibility of large rack-mount server architectures in a rugged, briefcase-like package.

The product offers single or dual Intel Xeon processors running at up to 3.06 GHz, 8 Gbytes DDRAM, 200 Gbytes removable IDE storage, five internal media bays, and dual-slot PCI expansion to leverage off-the-shelf I/O technologies. It also features a 15-inch TFT 1600 × 1200 LCD, ATI Mobility M7 graphics with 32 Mbytes of VRAM, and flip-down keyboard.

FlexPCServer supports numerous operating systems including SuSE Linux Enterprise Server and Desktop Linux; Red Hat Enterprise Linux; Fedora Core Linux; Windows Server 2003, XP Pro, and 2000 Pro; Solaris 9x86, Win-UX (VMWare) concurrent-use Linux and Windows; and various dual-boot configurations.

Contact the company for more information; www.nextcomputing.com.

Themis Launches 64-Bit VMEbus Computers

Themis Computer's PPC64 is the first of a new family of single-board VMEbus computers based on the 1.8-GHz IBM PowerPC 970FX processor, which is designed to deliver maximum performance for existing 32-bit and new 64-bit applications. The product is available in single-slot uniprocessor and two-slot dual-symmetric multiprocessing configurations.

The single-processor PPC64 offers up to 4 Gbytes DDR400 SDRAM and two Gigabit Ethernet ports, two Ultra320 SCSI channels, one AC97 audio port, one 10/100 Ethernet port, six USB ports, and two serial ports. It also includes a high-performance Universe II VME64x interface and Linux support. An additional processor can be added in a second VME slot, communicating with the baseboard via a HyperTransport link.

PPC64 VMEbus computers will be available in Q2 2005, with single-processor models priced below $6,000 in OEM quantities; www.themis.com.

Please send new product announcements to products@computer.org.

Parasoft Upgrades Java-Testing Tool

Jtest from Parasoft is an automated Java unit testing and coding standard analysis tool designed to improve Java code reliability, functionality, security, performance, and maintainability throughout the software lifecycle. Version 6.0 includes new capabilities such as increased coverage for complex code with configurable scenarios, more realistic and flexible testing, faster generation of automated JUnit tests, more than 500 built-in Java development rules, the ability to parameterize rules, a security module that delivers more than 100 vulnerability rules, .jsp file testing, and prioritization of unit-testing errors.

Jtest 6.0 is available for Windows 2000 and XP, Linux, and Solaris, with prices starting at $3,495 for a single-user, machine-locked license; www.parasoft.com.

IBM's Emerging Technologies Toolkit Gets Update

The Emerging Technologies Toolkit is a collection of downloadable tools, example code, documentation, and executable demos from IBM's alphaWorks designed to help developers create and deploy autonomic and Web services technologies. ETTK demos and functions run on both the Linux and Windows operating systems.

Version 2.2 of the ETTK provides an early look at the Generic Manageability Library, WS-Resource wrapper for CIM, Semantic Web Services, DNS-EPD (EndPoint Discovery), WS-Agreement, WSDL Port Aggregator Tool, Web Services Navigator, WS-Metadata Exchange (Sept. 2004 spec.), WS-ResourceFramework 1.2, WS-Notification 1.2, and WS-Addressing (Aug. 2004 spec.).

The ETTK is distributed with a license that allows 90 days of free usage; www.alphaworks.ibm.com/tech/ettk.

Borland Releases Integrated Role-Based SDP

The Borland Core Software Delivery Platform from Borland Software is an integrated suite of role-based software development tools designed to foster collaborative application lifecycle management. Borland Core SDP lets distributed teams customize workflow processes, proactively manage change across roles, and capture and analyze ALM metrics. It supports the company's own JBuilder IDE as well as the open source Eclipse development framework, with future support planned for Microsoft .NET.

Licences for Borland Core SDP are priced according to the platform and number of users in each job category; www.borland.com.

NextCom's FlexPCServer combines a mobile server with a high-performance mobile workstation in one portable, rugged package.

William J. Dally Earns 2004 Cray Honor

In recognition of groundbreaking achievements in high-performance computer processing, the IEEE Computer Society recently presented Stanford University's William J. Dally with the 2004 Seymour Cray Computer Science and Engineering Award. Dally's citation highlights his "fundamental contributions to the design and engineering of high-performance interconnection networks, parallel computer architectures, and high-speed signaling technology."

The chair of the Computer Science Department at Stanford University, Dally is principal investigator on the Imagine processor. This programmable signal and image processor achieves the performance of a special-purpose processor. Imagine is the first of its kind to use stream processing and partitioned register organization.

In addition to the Imagine processor, Dally has been instrumental in developing a "streaming supercomputer" capable of scaling easily from a single chip to thousands of processors. In contrast to conventional cluster-based supercomputers, the streaming supercomputer uses stream processing combined with a high-performance network that accesses a globally shared memory.

Working with Cray Research and Intel researchers, Dally has incorporated many of his innovations into commercial parallel computers. With Avici Systems, Dally has brought his technologies to Internet routers.

At Stanford, Dally is a member of the Computer Systems Laboratory, leads the Concurrent VLSI Architecture Group, and holds the Willard R. and Inez Kerr Bell Professor of Electrical Engineering and Computer Science position.

William J. Dally has made fundamental contributions to high-performance computer processing technology.

Dally has played a key role in founding Stream Processors, a company whose mission is to commercialize stream processors for embedded applications. Prior to joining Stanford's faculty, Dally worked at Bell Labs, Caltech, and MIT. At Bell Labs, he contributed to the design of the Bellmac32 microprocessor and designed the Mars hardware accelerator. At MIT, Dally led development of the Reliable Router high-performance routing chip and of the M-Machine, a fine-grained multicomputer project that later moved to Stanford.

A fellow of the IEEE and the ACM, Dally recently received the 2005 ACM/Sigarch Maurice Wilkes Award for outstanding contributions to computer architecture.

The Seymour Cray Computer Science and Engineering Award recognizes individuals whose contributions to high-performance computing systems best reflect the creative spirit of supercomputing pioneer Seymour Cray. Recipients of the Cray Award receive a crystal memento, an illuminated certificate, and a $10,000 honorarium. Recent recipients include John Hennessy, Monty Denneau, and Burton Smith.

For further information on the Cray and other Computer Society awards, visit www.computer.org/awards/. ■

Computer Society Recognizes Outstanding Students

The future of computer engineering depends upon nurturing talented students who can bring a fresh perspective and enthusiasm to a profession challenged by shifting global priorities.

In recognition of the impact of education on future professionals, the IEEE Computer Society rewards student achievers with scholarships, promotes innovation through events like the Computer Society International

Design Competition (CSIDC), supports student chapter activities, and sponsors awards for precollege science fair participants.

The IEEE Computer Society recently presented student awards to two outstanding undergraduates.

LANCE STAFFORD LARSON OUTSTANDING STUDENT PAPER AWARD

Akin Günay of Eastern Mediterranean University in Northern Cypress won a $500 scholarship for submitting the winning entry in this year's Lance Stafford Larson best paper contest. The contest, open only to student members of the Computer Society, rewards a future computing professional who demonstrates exceptional skill in writing and communication. Judges score entries on writing proficiency, technical content, and overall presentation. To be eligible for the Larson Award, student members must maintain a minimum 3.0 GPA.

UPSILON PI EPSILON AWARD FOR ACADEMIC EXCELLENCE

In cooperation with international computing honor society Upsilon Pi Epsilon, the Computer Society presents the Upsilon Pi Epsilon Award for Academic Excellence to students who demonstrate high academic achievement and participate in computer-related extracurricular activities.

This year, the IEEE Computer Society presented the UPE Award to Neha Jain of North Carolina State University. Jain is also a 2004 winner of the Google-sponsored Anita Borg Scholarship.

Up to four UPE awards of $500 each are presented annually. Winners also receive their choice of either a Computer Society book or a one-year subscription to a Computer Society periodical.

Computer Society volunteers Lowell Johnson, Fiorenza Albert-Howard, and Murali Varanasi served as judges for the Larson and Upsilon Pi Epsilon scholarships. Applications for next year's scholarships are due by **31 October**.

Each year, the IEEE Computer Society also offers up to 10 Richard Merwin Student Scholarships to student chapter leaders. The $4,000 Merwin prizes are paid out to individual winners in four installments. The deadline to apply is **31 May**.

For more information or to apply for Computer Society student awards programs, visit www.computer.org/students/schlrshp.htm. ■

Collaborative Research Program Broadens Its Outreach to Students

The Computing Research Association's Committee on the Status of Women in Computing Research (CRA-W) recently announced an initiative designed to involve larger numbers of women and minority undergraduates in cooperative, hands-on research. By offering such research opportunities, the CRA-W aims to encourage more women and minorities underrepresented among computer science and engineering undergraduates to continue on to graduate-level study.

Operating for six years as the Collaborative Research Experience for Women (CREW) program, the initiative's scope now includes other populations not commonly found among computer engineering professionals. Participants in the Collaborative Research Experience for Undergraduates in Computer Science and Engineering (CREU) program work on research projects at their home institutions during the academic year, in groups of two or three juniors or seniors. The students collaborate with one or two sponsoring faculty members on a project for which financial support would be otherwise unavailable. Each student receives a stipend of $1,000. In addition, participants can request up to $500 per project for special equipment, travel, or supporting materials.

At the end of the project, students submit a one-page summary of their work for posting on the CRA-W Web site. Most students also submit papers or otherwise present their work to relevant journals and conferences.

Joining the CRA-W in sponsoring the CREU initiative are the National Science Foundation, Usenix, and the National Science Foundation's Partnership for Advanced Computational Infrastructure.

Proposals for 2005 CREU projects must be submitted by **1 July**. To support the exchange of shared common experiences, individual teams should be homogeneous with respect to minority status or gender. Teams consisting of all women or all underrepresented minorities are especially encouraged to apply.

Prospective CREU participants must have completed two years of undergraduate study at the college level, including at least four courses in computer science or computer engineering.

For more information on the CRA-W CREU project, including detailed eligibility requirements and student project summaries from past years, visit www.cra.org/Activities/craw/creu/.

Editor: Bob Ward, *Computer*; bnward@computer.org

Computer Society Honors Berger with Fernbach Award

Marsha Berger, a professor of computer science at New York University's Courant Institute, has been honored with the 2004 IEEE Computer Society Sidney Fernbach Memorial Award.

In the awards citation, the award committee praised Berger's "many contributions to, and enormous influence on, computational fluid dynamics, including adaptive mesh refinement methods, Cartesian grid methods, and practical mathematical algorithms for solving significant and previously intractable problems."

Berger's research focuses on scientific computing applications in fluid dynamics and encompasses areas of computer science, numerical analysis, and applied mathematics.

Berger earned the 2002 NASA Software of the Year Award for her collaboration on Cart3D, a package for automated Cartesian grid generation and aerodynamic database creation. Her other honors include the 2000 NYU Sokol Faculty Award in the Sciences, the NSF Faculty Award for Women, and the NSF Presidential Young Investigator Award. Berger was elected to the National Academy of Sciences in 2000.

The Fernbach Award recognizes individuals who have made notable strides in developing applications for high-performance computing. An awards committee associated with the annual SC high-performance computing, networking, and storage conference evaluates nominations. The Fernbach winner receives a certificate of recognition and a $2,000 honorarium during a special ceremony at the conference.

Nominations for the next year's Fernbach Award are due by **31 July**. To nominate a potential recipient or to learn more about any IEEE Computer Society award, visit www. computer.org/awards/. ■

Game Engine Virtual Reality with CaveUT

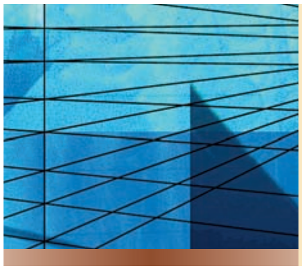
Jeffrey Jacobson and Michael Lewis, University of Pittsburgh

CaveUT, an open source freeware project (http://PublicVR.org/ut/CaveUT.html), uses game technology to make immersive projection-based virtual reality affordable and accessible.

Relatively simple, the current public release of CaveUT works well for low-cost displays. The most advanced version, which will be publicly available in the fall of 2005, supports real-time spatial tracking and stereographic imaging. It is currently installed and working in the SAS-Cube (www.alterne.info/techn_platform.html), a CAVE-like display.

Computer games with the most advanced simulation and graphics usually employ a *game engine*, a commercially available software package that handles basic functions. For example, the first-person shooter *Unreal Tournament* for the PC employs the Unreal Engine to provide richly detailed graphics, high-speed processing performance, a built-in physics engine, a scripting language interpreter, and robust networking for shared environments.

CaveUT modifies *Unreal Tournament* to let it display in multiscreen enclosures suitable for immersive virtual reality applications. VR applications developed with CaveUT inherit all the Unreal Engine's capabilities along with *Unreal Tournament*'s authoring support, open source code, content library, and large user community.



Based on *Unreal Tournament*, the CaveUT game engine gives developers a high-performance, low-cost VR alternative.

ORIGIN AND DEVELOPMENT

In 1990, researchers developed the original Cave Automatic Virtual Environment (http://cave.ncsa.uiuc.edu/about.html), a partial cube approximately three meters per side, with each wall functioning as a rear-projection screen illuminated by a projector. A mainframe drives all of the CAVE's projectors, displaying a contiguous visual image across all screens to produce a virtual landscape. Stereographic imaging makes the virtual objects look more three-dimensional, while real-time spatial tracking lets users interact with the objects and navigate the space.

Throughout the 1990s, researchers developed many VR applications for the CAVE and similar displays. They either programmed these applications directly, starting from OpenGL or a similar graphics library, or wrote them using advanced authoring kits. Despite the process's difficulty and expense, it usually produced applications with poor and often primitive graphics, low performance, and limited networking functionality.

In 1997, Paul Radjlich produced a version of *Quake* for CAVE (http://brighton.ncsa.uiuc.edu/~prajlich/caveQuake/) that inherited the game's authoring support, networking, and other features. Unfortunately, CaveQuake could not benefit from *Quake*'s game engine, which was PC-based.

By 2000, the game industry had driven significant advances in graphics hardware for the PC, and CAVE owners began replacing their mainframes with PC networks thanks to the PC's low cost and increasing graphics power. Further, the leading first-person shooters, *Quake* and *Unreal Tournament*, surpassed the traditional CAVE-based applications in graphics quality, performance, animation, and networking. These games also had respectable authoring support, built-in physics, partially open source code, a large base of existing content, and an active developer community.

Despite these developments, the games still lacked the ability to perform real-time shape generation for scientific visualization applications. Their interface and animation support also proved limiting for applications using a different paradigm.

Thus, in 2000 we decided to adapt *Unreal Tournament* to the BNAVE, a PC-based CAVE-like display. With Michael Lewis's support and guidance, Jeffrey Jacobson and Zimmy Hwang invested a year of careful study, then solved the multiscreen display problem by inserting just six lines into the game's open source code.

We packaged this as the first version of CaveUT, which we made into a freeware project to attract collaborators.

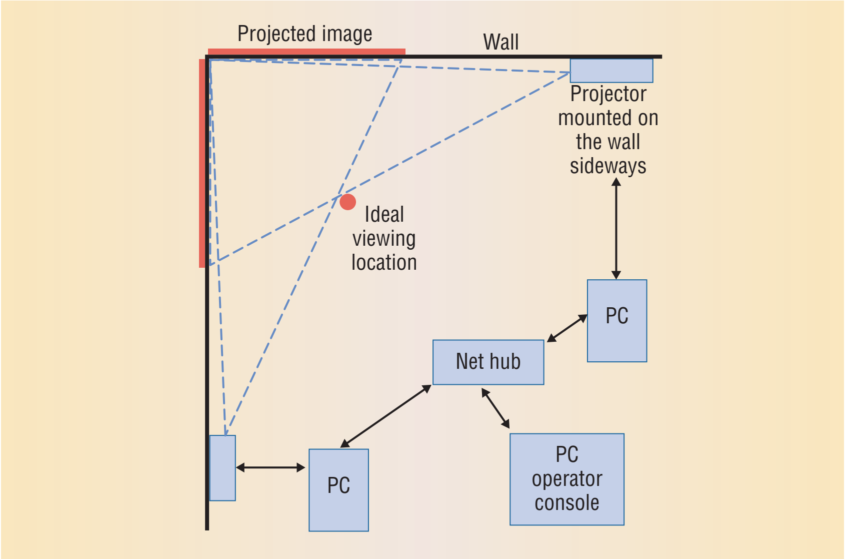


Figure 1. V-Cave schematic. In this simple CaveUT installation, each screen is turned 45 degrees. The Virtual Theater is similar, but with four screens instead of two.

Since then, CaveUT has become far more capable and complex through independent code contributions.

Today, most software development for CAVE still uses the traditional VR authoring tools, which have improved considerably but remain expensive. However, the game engines have maintained their list of advantages and increased their lead in graphics quality and performance, thanks to massive funding by a game industry that has grown larger than Hollywood.

Already, many developers in science and industry take advantage of game technology for other uses (www.seriousgames.org/), with developers basing more and more immersive VR applications on games. Meanwhile, CaveUT has attracted a growing list of collaborators, which should lengthen with the release of the project's stereographic code.

HOW IT WORKS

A multiscreen display based on CaveUT requires a server computer that connects by a standard LAN to several client computers. Each client drives one screen of the display, usually a projection screen illuminated by a digital projector.

The operator begins a multiplayer game of *Unreal Tournament* with one normal player on the server and one spectator player on each of the clients. Each spectator's view duplicates the view seen by the player on the server. On each client, the CaveUT code rotates the view—the screen's window—into the virtual world so that each screen shows its part of the composite view. Figure 1 shows the view of one screen turned 45 degrees to the left, while the other screen has been turned 45 degrees to the right.

To handle perspective correction, CaveUT employs Willem de Jonge's VRGL, an OpenGL library modified for VR applications. For an installation with no head tracking, the user must specify a single ideal viewing location for the whole display. As long as the user's head stays at or very near this point, the view will remain unified and undistorted. If the installation does have a head tracker, CaveUT can correct the perspective in real time to effectively follow the user's movement.

This arrangement uses an unmodified Unreal Engine, with its internal functioning unaffected by CaveUT and VRGL. This lets CaveUT remain open source and easily upgradable to each new engine version.

Unreal Tournament provides CaveUT's good performance by ensuring that each machine's copy maintains a complete instance of the virtual world and performs all its own graphics rendering, physics, animation, and related operations. Client-server communication is confined to fast and simple state-change updates such as, "The player is now here and is moving in this direction at this speed."

CaveUT can support at least 32 independent view screens for a single application, in any configuration, and with multiple real players. For example, developers could configure a six-walled enclosure with all its views centered on the first user, two more four-walled enclosures with views that center on the second and third users, and single-desktop arrangements for eight more users. All 11 users could share a single virtual environment. Some could be students, others instructors, and yet others acting as intelligent scenery. In this scenario, only imagination and budget limit the possibilities.

MINIMUM REQUIREMENTS

A monoscopic nontracking CaveUT display can be configured cheaply from common office equipment and simple hardware. Front-projection screens can consist of any clean, white surface, while rear-projection screens can be made from any translucent material. Each screen requires one common DLP or LCD projector driven by a standard personal computer with a good video card and an installed copy of *Unreal Tournament*, which currently retails for around $20.

An additional computer hosts the server and acts as the operator console. The server connects to the clients through an Ethernet switch and the appropriate cables. CaveUT/VRGL is currently limited to OpenGL and the Windows OS, although porting it to Linux/Unix/MAC-OSX would not be difficult. Configuring the Virtual Theater shown in Figure 2 cost only

$25,000 and required no special programming.

With CaveUT, the installation developer can upgrade steadily from an inexpensive display to a fully capable CAVE-like interface. The display supports spatial trackers useful for head-tracking and advanced controls. For example, the user can manipulate virtual objects or navigate largely by pointing with the tracked controller.

CaveUT also supports stereographic imaging at a per-screen cost of two computers with specialized video cards and a stereographic projector for active stereo or a pair of monoscopic projectors for passive stereo. Although less expensive than active stereo, passive stereo is limited to displays that array the screens horizontally around the viewer because the illusion falls apart if the user's head tilts. When used in combination with a head tracker, active stereo allows unrestricted viewing angles.

A CaveUT installation could be used to interact with most content written for *Unreal Tournament* and most applications built on the game. The highly localized code changes that CaveUT introduces are unlikely to conflict with the *UT*-based application's code changes. This is an important advantage because the large community of *Unreal Tournament* gamers and researchers produces a great deal of artistic content, animation, and code modifications for the game engine. CaveUT developers can use most of this material, in pieces or in whole applications, and benefit from the *Unreal* development community's support and cooperation.

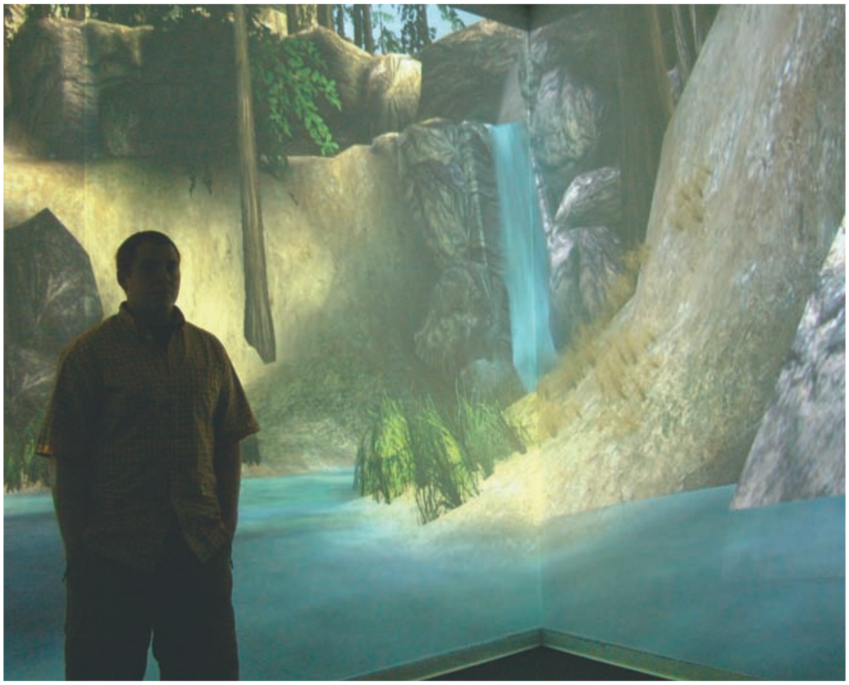


Figure 2. Virtual Theater, University of Pittsburgh's Information Sciences Department. The user can manipulate virtual objects or navigate using standard game peripherals.

CURRENT PROJECTS

Several projects already use CaveUT, including the following:

- The ALTERNE project's artists use CaveUT for interactive art installations, storytelling, and information visualization (www.alterne.info/).
- Researchers at the University of Pittsburgh's Visual Information Sciences Center (http://visc.exp.sis.pitt.edu/projects/707.asp) use CaveUT to prototype systems for first-responder emergency training, virtual museums, way-finding applications, and architectural planning.
- Researchers at the University of Pittsburgh's Usability Lab (ULAB) are developing CaveUT projects for robotic simulations (http://usl.sis.pitt.edu/ulab/CotrollingRobotl.htm) and virtual archeology (http://planetjeff.net/#HorusUnreal).
- Stagecraft designers at the University of Southern California's Institute for Creative Technologies use CaveUT for interface prototyping (www.ict.usc.edu/disp.php?bd=proj_flatworld).
- A low-cost portable CaveUT display (publicvr.org/ut/CUT4Cave.html) has been demonstrated at conferences such as CHI 2002, HFES 2002, VR 2003, and I3D 2003.
- The artists at Elumenati (www.elumenati.com/) are helping to develop CaveUT's close cousin, DomeUT (http://planetjeff.net/#DomeUT), for applications in all-digital dome displays.
- Military researchers are using CaveUT as an immersive interface for their Unreal-Engine-based training simulators.

This list will continue to expand because CaveUT's low initial cost makes it accessible and attractive to students, educators, artists, developers, gamers, and small businesses working in a wide range of disciplines.

Along with the powers of the Unreal Engine, CaveUT also inherits its biases. The engine can support large virtual environments, but it works best with relatively small worlds that are rich in detail and activity. For example, the engine can support a dozen or so amazingly detailed and lifelike humanoid agents more efficiently than a large number of simple ones. All game engines work with predefined objects and cannot use a data stream to continuously generate visual effects. However, they can be used to display effects transmitted to the engine from other software or generated from data in advance. CaveUT also inherits the advantages and limitations of *Unreal Tournament*, but it could easily be adapted to any game based on the Unreal Engine. Similarly, CaveUT could be adapted for other graphics systems to provide more options for the developer.

Nevertheless, CaveUT now supports a wide range of applications. It continues to develop through contributions of all kinds, such as the following:

- Marc Le Renard and Jean Lugrin will release their stereographic extension for CaveUT this fall, and Le Renard will also release his spatial tracking for CaveUT then.
- Demiurge Studios is adapting VRGL—and therefore DomeUT/CaveUT—for use with multiprojector curved screen and partial-sphere displays.
- Ivor Diosi is working on a method for streaming video from an external source onto a surface in the virtual environment.
- Jacobson and Demiurge are developing optimal ways to use commercially available game peripherals in an immersive display.

All these new features will require documentation, tutorials, testing, and extension to a variety of platforms, especially the low-cost ones. We always welcome new users and collaborators as we build up the CaveUT user community. ■

__Jeffrey Jacobson__ is a graduate student and __Michael Lewis__ is an associate professor, both in the University of Pittsburgh's Information Sciences Department. Contact Jacobson at jeff@planetjeff.net and Lewis at ml@sis.pitt.edu.

Editor: Michael Macedonia, Georgia Tech Research Institute, Atlanta; macedonia@computer.org.

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 The IEEE Computer Society's Web site, at **www.computer.org**, offers information and samples from the society's publications and conferences, as well as a broad range of information about technical committees, standards, student activities, and more.

OMBUDSMAN Members experiencing problems—magazine delivery, membership status, or unresolved complaints—may write to the ombudsman at the Publications Office or send an e-mail 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 the Publications Office:

- 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)

To check membership status or report a change of address, call the IEEE toll-free number, +1 800 678 4333. Direct all other Computer Society-related questions to the Publications Office.

PUBLICATIONS AND ACTIVITIES

Computer. The flaship 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 15 magazines and 14 research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The IEEE Computer Society Conference Publishing Services publishes more than 175 titles every year.

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

Technical Committees. TCs provide professional interaction in over 30 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education. The society holds about 150 conferences each year and sponsors many educational activities, including computing science accreditation.

EXECUTIVE COMMITTEE

President:
GERALD L. ENGEL*
Computer Science & Engineering
Univ. of Connecticut, Stamford
1 University Place
Stamford, CT 06901-2315
Phone: +1 203 251 8431
Fax: +1 203 251 8592
g.engel@computer.org

President-Elect:
DEBORAH M. COOPER*

Past President:
CARL K. CHANG*

VP, Publications:
MICHAEL R. WILLIAMS (1ST VP)*

VP, Electronic Products and Services:
JAMES W. MOORE (2ND VP)*

VP, Chapters Activities:
CHRISTINA M. SCHOBER*

VP, Conferences and Tutorials:
YERVANT ZORIAN†

VP, Educational Activities:
MURALI VARANASI†

VP, Standards Activities:
SUSAN K. (KATHY) LAND*

VP, Technical Activities:
STEPHANIE M. WHITE†

Secretary:
STEPHEN B. SEIDMAN*

Treasurer:
RANGACHAR KASTURI†

2004–2005 IEEE Division V Director:
GENE F. HOFFNAGLE†

2005–2006 IEEE Division VIII Director:
STEPHEN L. DIAMOND†

2005 IEEE Division V Director-Elect:
OSCAR N. GARCIA*

Computer Editor in Chief:
DORIS L. CARVER†

Executive Director:
DAVID W. HENNAGE†

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

BOARD OF GOVERNORS

Term Expiring 2005: *Oscar N. Garcia, Mark A. Grant, Michel Israel, Robit Kapur, Stephen B. Seidman, Kathleen M. Swigger, Makoto Takizawa*

Term Expiring 2006: *Mark Christensen, Alan Clements, Annie Combelles, Ann Q. Gates, James D. Isaak, Susan A. Mengel, Bill N. Schilit*

Term Expiring 2007: *Jean M. Bacon, George V. Cybenko, Richard A. Kemmerer, Susan K. (Kathy) Land, Itaru Mimura, Brian M. O'Connell, Christina M. Schober*

Next Board Meeting: *10 June 2005, Long Beach, CA*

EXECUTIVE STAFF

Executive Director: DAVID W. HENNAGE
Assoc. Executive Director:
ANNE MARIE KELLY
Publisher: ANGELA BURGESS
Assistant Publisher: DICK PRICE
Director, Administration:
VIOLET S. DOAN
Director, Information Technology & Services:
ROBERT G. CARE
Director, Business & Product Development:
PETER TURNER

COMPUTER SOCIETY OFFICES

Headquarters Office
1730 Massachusetts Ave. NW
Washington, DC 20036-1992
Phone: +1 202 371 0101 • Fax: +1 202 728 9614
E-mail: hq.ofc@computer.org

Publications Office
10662 Los Vaqueros Cir., PO Box 3014
Los Alamitos, CA 90720-1314
Phone:+1 714 821 8380
E-mail: help@computer.org
Membership and Publication Orders:
Phone: +1 800 272 6657 Fax: +1 714 821 4641
E-mail: 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
E-mail: tokyo.ofc@computer.org

IEEE OFFICERS

President and CEO :
W. CLEON ANDERSON

President-Elect:
MICHAEL R. LIGHTNER

Past President:
ARTHUR W. WINSTON

Executive Director:
TBD

Secretary:
MOHAMED EL-HAWARY

Treasurer:
JOSEPH V. LILLIE

VP, Educational Activities:
MOSHE KAM

VP, Publication Services and Products:
LEAH H. JAMIESON

VP, Regional Activities:
MARC T. APTER

VP, Standards Association:
JAMES T. CARLO

VP, Technical Activities:
RALPH W. WYNDRUM JR.

IEEE Division V Director:
GENE F. HOFFNAGLE

IEEE Division VIII Director:
STEPHEN L. DIAMOND

President, IEEE-USA:
GERARD A. ALPHONSE

IEEE

Social-Mobile Applications

Ian Smith, Intel Research Seattle

Mobile communications devices and applications are primarily designed to increase efficiency and productivity for professionals on the go. However, users invariably appropriate such technology to meet their social needs as well. For many people, particularly younger users, BlackBerry devices, Hiptops, and other handhelds primarily have a social function.

For a growing number of mobile users, handhelds primarily have a social function.

A few small companies are beginning to exploit the growing demand for social-mobile applications, also known as mobile social-software services.

One of the most popular MoSoSos applications is dodgeball (www.dodgeball.com), a New York-based social-mobile network with thousands of users in 22 cities across the US. After a registered user "checks in," friends receive a text message indicating the check-in location and time in case they want to get together. The service will also notify a user if a friend, friend of a friend, or "crush" is within 10 blocks. In addition, dodgeball users also can broadcast messages, or "shout," to those in their network.

Plazes (www.plazes.com) is a location-aware interaction system that helps mobile users hook up with friends or other like-minded people anywhere on the globe. Jambo Networks (www.jambo.net/web-site/Home.htmland) uses Wi-Fi-enabled laptops, cell phones, and PDAs to match people within walking distance who have similar interests and would like to meet face to face. In the UK, playtxt (www.playtxt.net) helps mobile users locate nearby friends, friends of mutual acquaintances, or even strangers with matching preferences.

RENO

During the past year, Intel Research Seattle has designed, studied, and built several applications to support a specific type of social interaction, the *rendezvous,* in which two or more people meet at the same location. Intel researchers chose this scenario for their initial test deployments because many people already use mobile devices to coordinate such meetings—for example, to notify others that they're running late.

With the Reno mobile phone application users can query other users about their location and disclose their own, either in response to another query or unprompted. Unlike other MoSoSos applications that support rendezvous, Reno is location aware. For example, dodgeball and playtxt require the user to manually type in an identifier—a place name or postal code, respectively—and send this data to a central server that performs the location calculation.

As Figure 1 shows, Reno calculates the device's approximate location locally using Global System for Mobile Communications (GSM) technology and then presents the user with a short list of nearby locations sorted by proximity. The user then selects the most appropriate place from the list, an easier and significantly faster process than typing text.

In addition to location awareness, Reno incorporates three design factors that are critical to the success of social-mobile applications: privacy, practicality, and specific value.

Privacy

MoSoSos applications must give users sufficient control of their personal data or risk being rejected as agents of Big Brother. Therefore, Intel researchers incorporated a number of privacy features into Reno up front, including user control of the disclosure of location information.

Reno's location algorithm binds specific features of the wireless GSM environment that the mobile device can sense to simple data strings the user chooses; it uses no other strings to reveal location information. Users need not label any place they regard as private and can be confident that the application will not disclose it to others.

In addition, Reno doesn't employ a tracking system that would enable others to ascertain a user's location without that person's knowledge. Rather, users disclose their location at a time they choose. For example, if Alice wants to know where Bob is, she must first request his location, then Bob must take some action to reveal it. Further, Bob only needs to choose a nearby location that he is comfortable with from the list that Reno presents.

Practicality

People often use mobile devices on the spur of the moment—for example, between pressing work engagements or while in transit. Social-mobile applications must therefore be simple to use and quick to operate or people will choose another form of communication, such as making a phone call.

Reno's design exploits the traditional "inbox" metaphor to let people quickly glance at requests for their location and disclose it to others. In a small deployment in the Seattle area, several test users commented that sending a Reno message was much easier than sending a traditional SMS message or making a call. Message recipients also found Reno less intrusive than a phone call for coordinating a rendezvous.

Specific value

A social-mobile application must offer some key benefit to be "sticky"—that is, convince users to repeatedly choose it over other communication techniques.

To test one common rendezvous scenario, coordinating family tasks with teenagers, Intel researchers targeted subjects with children who in most cases were not old enough to drive. The families used Reno an average of 2.4 times per day per user, which was encouraging given the inherent difficulty in using a new application on unfamiliar mobile phones. If it's possible to sustain or slightly exceed this level of use in future deployments, Reno has the potential to become the rendezvous tool of choice.

BEYOND RENO

Commercial and academic researchers are exploring numerous exciting opportunities for using MoSoSos applications. These include new kinds of dating applications, sales-force automation, finding a restaurant recommended by friends, and monitoring human-rights workers.

Since the initial deployment of Reno in 2004, Intel researchers have redesigned the rendezvous application to support more complex coordination scenarios. Like its predecessor, this application doesn't enable users to surreptitiously track others. Rather, the device display shows a map of the user's region with icons representing the most recent information that others have disclosed about themselves. For example, someone who is late to a meeting might elect to continually update his location so others can estimate his time of arrival.

Intel researchers are also prototyping Houston, an application designed to investigate the utility of mobile social-support networks. Houston is oriented toward physical fitness and weight management, but the general principles apply to many other areas where friends share experiences and get mutual support.

With Houston, group members share step counts from their pedometers automatically via mobile phones. Users control what to disclose about themselves and can view what others choose to reveal—for example, "Joe made it to 10,000 steps today!" The goal is to determine whether this approach changes behavior more effectively than traditional social-support networks such as commercial weight-management groups or at least offers similar support at a lower cost in time and money.

Following the PC in the 1980s and the World Wide Web in the 1990s, the mobile device is emerging as the next general-purpose computing platform, most likely in a form similar to today's cell phone.

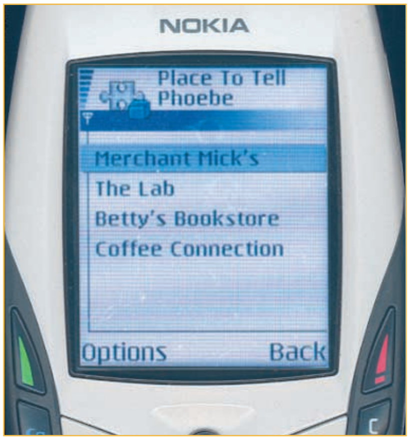


Figure 1. Social-mobile application. Reno calculates the user's location and sorts nearby places by proximity.

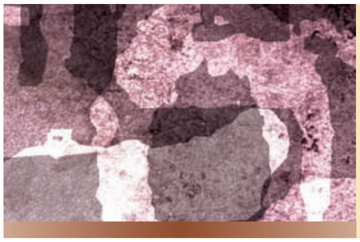
Mobile phone designers weren't as agnostic as their predecessors; they had at least one specific application in mind—namely, mobile telephony. Nevertheless, within the next decade numerous highly specialized classes of applications will emerge for the mobile platform. Intel researchers expect social-mobile applications to be one of these classes. ■

Ian Smith is a senior researcher at Intel Research Seattle. Contact him at ian.e.smith@intel.com.

Editor: Bill Schilit, Intel Research Seattle; bill.schilit@intel.com

In Defense of Spam

Neville Holmes, University of Tasmania



Legislators' well-meaning attempts to eradicate spam are woefully misdirected.

Denigrating spam has become a popular activity, if an ill-directed one. My experience with defending PowerPoint tells me that, before I begin defending spam, I should emphasize that by doing so I am not necessarily praising it.

In the letters column in *Computer*'s July 2004 issue, correspondent Davy Cheung concluded, "Does anyone really believe that anti-spam—or 'unsolicited communications,' to be exact—laws are not necessary?" In the October 2004 issue, Brian Whitworth and Elizabeth Whitworth spelled out why "passing laws in virtual worlds has several problems" ("Spam and the Social-Technical Gap," pp. 38-45). After describing four major problems, they observed that "the long arm of the law struggles to reach into cyberspace."

Indeed, it seems that antispam legislation has been largely ineffective. How can this impasse be broken?

DEFINING SPAM

The word *Spam* is a registered trademark (www.rsi.com/spam/) long owned by Hormel Foods LLC. Kenneth Daigneau, a New York actor and the brother of a Hormel executive, coined the trademarked term *Spam*, which came into successful commercial use in 1937. Some sources suggest that the term derived from a contraction of "spiced ham." During World War II, Spam—not being rationed as beef products were—was consumed widely, especially in the armed forces. Spam became so ubiquitous that the medal given by some governments to all those who served in that war at home or abroad was colloquially called "the spam medal." This sense of unlimited dispersal appropriately describes some varieties of the electronic messages now called spam.

Wikipedia (en.wikipedia.org/wiki/Spamming) defines spam as "the use of any electronic communications medium to send unsolicited messages in bulk" and refers to five different media: e-mail, messaging, newsgroups, mobile phones, and Internet telephony. Spam also refers to Web site interference that, for example, increases a product's search engine ranking through spamdexing. According to Wikipedia, blog, wiki, guestbook, and referer spam are all prevalent as well.

ANTISPAM LEGISLATION

Spam has been targeted by special legislation that seeks to control it, although legislators disagree about what it is and why it must be controlled. In the European Union, for example, the Privacy and Electronic Communications Directive 2002/58/EC is inclusive within the general scope of regulating the use of many kinds of personal data. Article 13(1), a minor exception aside, "prohibits the sending of unsolicited commercial communications by fax or e-mail or other electronic messaging systems such as SMS or MMS unless the prior consent of the addressee has been obtained…."

The US based its legislation—the Controlling the Assault of Non-Solicited Pornography and Marketing Act of 2003—on the determination that commercial electronic mail should be regulated nationwide, that senders should not mislead recipients, and that recipients have a right to decline receiving further e-mail from a sender.

The Australian Spam Act of 2003 seeks to regulate *commercial* e-mail and other types of *commercial* electronic messages, forbids these when unsolicited (with exceptions), requires the sender to be identified and the receiver to be able to opt out, forbids address-harvesting programs and their output, and emphasizes that the main remedies are civil.

This legislation is typically voluminous and difficult to understand in full, which perhaps explains why there are many accompanying documents that, in particular, explain how businesses can continue their use of the Internet for marketing.

All of which makes it difficult to understand why—assuming lawmakers considered the existing legislation that relates to the control of marketing inadequate—they chose not to amend this legislation so that it would be adequate.

It's almost as though US and Australian legislators felt that the Internet itself, not the marketers, pose an extraordinary threat to users. After

Continued on page 86

Continued from page 88

all, the legislation carefully provides for Internet marketing to continue, proving that lawmakers do not consider marketing itself a problem.

JUSTIFICATION

The explanatory memorandum justifying the Australian legislation cites spam's effect on several aspects of Internet use:

- *User confidence.* "Today, the problem of spam has … a significantly negative effect on users' confidence in using e-mail." But why is this a government concern? Given that e-mail is a commercial service, user confidence should properly be the service provider's concern.
- *Network integrity.* "There are clear signs of a deleterious impact on the performance of the global e-mail network … [which] could mean the end of e-mail as an effective form of communication." But surely if the network fails to function satisfactorily, its commercial owners should use technologists to fix it from the inside, not the government to fix it from the outside.
- *Privacy.* "There are significant privacy issues surrounding the manner in which e-mail addresses and personal information are collected and handled." Is this peculiar to the Internet? Shouldn't the Internet's owners be responsible for that medium's methods of handling personal information? Certainly the European Union legislators think so.
- *Content.* "There are obvious … concerns with the illicit content of a considerable amount of spam—including those that promote pornography, illegal online gambling services, pyramid selling, get rich quick schemes or misleading and deceptive business practices." If such content is harmful, should it make a difference whether it appears on the Internet? If many are gullible enough to be taken in by spam, shouldn't the government be concerned about its constituents' gullibility rather than the bait? Shouldn't government ask instead why the education system has let society down?
- *Spoofing.* "Spammers may use spoofing to route spam through a reputable organization in an attempt to entice recipients to open and respond to their messages." Isn't this a technical problem that should be dealt with by having the Internet protocols prevent false addressing?
- *Financial costs.* "These [estimated to be huge] costs are usually borne by Internet users (and/or) employers … Spammers … bear relatively small costs in sending these messages." Surely, this amounts to saying that the business model is wrong. Why should the government try to cover up business problems? Won't this merely delay the development of a more viable and amenable Internet?

Software cannot undermine society, although _people_ can use it to do so.

After some discussion of spam statistics, the Australian report eventually tackles the basic issue: Why is antispam legislation necessary? The report gives the following reasons:

- most spammers are not subject to codes of practice,
- applying present content laws to spam could be expensive, and
- technical solutions are imperfect and can't relieve the overload on the Internet.

These reasons raise more questions than the report answers. What is the law but a code of practice? How will the spammers be constrained to obey the law anyway?

If this is all primarily about unsolicited broadcast commercial electronic messages, and the government seeks to protect me from them, why won't it try to protect me from the unsolicited broadcast of commercial electronic messages that overwhelm television, particularly around seven in the evening and during major sporting events? Government makes the commercial television stations responsible for the advertising they accept. Why don't they put the same responsibility on the Internet owners?

If applying present content laws to spam would be expensive, why not improve the present laws rather than come up with new law specific to the Internet? After all, the Internet isn't the only game in town. Will there be new laws regulating content over mobile phone transmissions? What happens if RFID technology gets extended to sending messages to mobile phones in the neighborhood—will the spam laws then need further expansion?

Surely the focus should be on the content itself rather than on the particular medium.

Technical solutions are always imperfect—at least to some degree, as Bob Colwell will tell you. This provides a compelling reason to improve the technology, not to resort to legislation.

I'm puzzled by the talk of overload on the Internet backbone. How could e-mail overwhelm dense wavelength division multiplexed optical fiber? Doesn't the repetitive downloading of the complex and largely pointless graphics that adorn most Web pages place a far greater burden on the Internet, not to mention on the user's patience? What's going to happen when the browsers start interleaving fancy commercials with their browsings?

Perhaps these questions explain why antispam legislation doesn't seem to be having much effect and why spam filtering and blocking remain the primary tools for countering spam. Even these measures don't stop the spam from being sent, and spammers can usually work around the countermeasures. They can also have side effects, such as contributing to the digital divide (www-staff.it.uts.edu.au/~lueg/papers/asistam04.pdf).

SYMPTOM OR DISEASE?

In treating diseases, a physician might succeed in alleviating the symptoms but will always realize professionally that eradicating the illness requires seeking the cause. If the cause is a continuing one, the disease can only be conquered by removing that cause. If it's not, merely knowing what caused the illness can help to more quickly and thoroughly restore the patient's health.

Much the same principles apply to alleviating malfunctions in the use of digital technology. In a thoughtful and prescient article, "Copy Protection Technology Is Doomed" (*Computer*, Oct. 2001, pp. 48-49), Dan S. Wallach argued that copyright violations constitute a commercial disease and that the "only way to prevent teenage girls from freely sharing boy-band MP3s will be to provide reasonably priced service that's irresistibly better than free file sharing." This seems to be the direction the recorded music industry is finally moving in, albeit reluctantly and ponderously.

The disease behind illegally copying digital entertainment or software is, however, in contrast to that behind spam, spim, viruses, and hacking. The first is theft, because legislators have seen fit to create monopolistic property rights over intangibles. The second is intangible assault even if, as in the case of phishing, it's with a view to theft. Someone sends the spam to someone else or damages or abuses someone else's computer without permission. The first is impersonal, the second personal; the first is a commercial disease, the second a social one.

Computing professionals are not responsible for diagnosing the social disease behind digital assault. This problem is arguably only one symptom of a disease that includes everything from telemarketing and littering to massacre and terrorism. However, digital assault is easier to study than other social malaises, and computing professionals should team with social scientists to help investigate social phenomena.

It is proper, even mandatory, for computing professionals to design and implement systems that make digital assault more difficult to commit. The Whitworths focused on fairness and legitimacy as aims that digital technology can support to discourage such assault. They concluded that "If software is to support society, not undermine it, legitimacy concepts must be taught in core information system design courses, as a social-technical requirement."

Nevertheless, at the social level, digital systems serve merely as intermediaries in digital interaction, and designing them to make digital assault more difficult would only treat the symptom. *Software* cannot undermine society, although *people* can use it to do so. Digital technology supports people, and people, in turn, can support or attack society.

Sending someone an unsolicited commercial electronic message is illegitimate only if done with malicious intent. Digital systems can be designed to deter certain kinds of digital communication, but this does nothing to deter malice and could even amplify it.

If all professionals learned about social actualities as well as concepts, they would be better placed to choose, design, and implement procedures and systems that lessen the causes of malice in society. ■

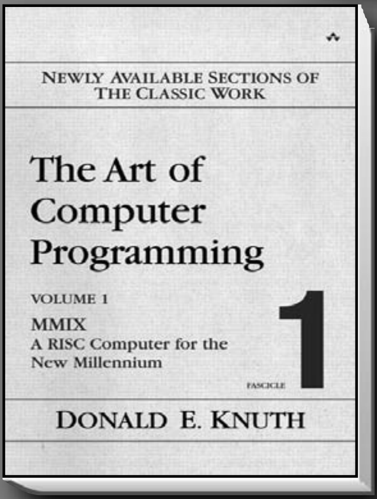
Neville Holmes *is an honorary research associate at the University of Tasmania's School of Computing. Contact him at neville.holmes@utas.edu.au. Details of citations in this essay, and links to further material, are at www.comp.utas.edu.au/users/nholmes/prfsn.*

Editor: Neville Holmes, School of Computing, University of Tasmania; neville.holmes@utas.edu.au. Links to further material are at www.comp.utas.edu.au/users/nholmes/prfsn.

The Art of Computer Programming

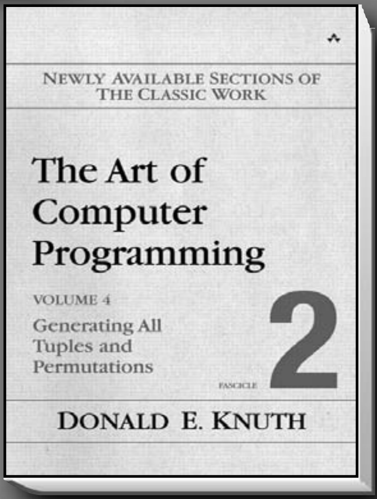
RETURNS!

Now Available



NEWLY AVAILABLE SECTIONS OF THE CLASSIC WORK
The Art of Computer Programming
VOLUME 1
MMIX
A RISC Computer for the New Millennium
FASCICLE
1
DONALD E. KNUTH

ISBN: 0-201-85392-2



NEWLY AVAILABLE SECTIONS OF THE CLASSIC WORK
The Art of Computer Programming
VOLUME 4
Generating All Tuples and Permutations
FASCICLE
2
DONALD E. KNUTH

ISBN: 0-201-85393-0

TO BEGIN THE FOURTH and later volumes of this renowned set, and to update parts of the existing three volumes, DONALD E. KNUTH has created a new series of small books called fascicles.

More new titles from Addison-Wesley

C++ PRIMER, 4th Edition

Stanley B. Lippman • Josée Lajoie • Barbara E. Moo

This popular tutorial introduction to standard C++ has been completely updated, reorganized, and rewritten to help programmers learn the language faster and use it in a more modern, effective way.

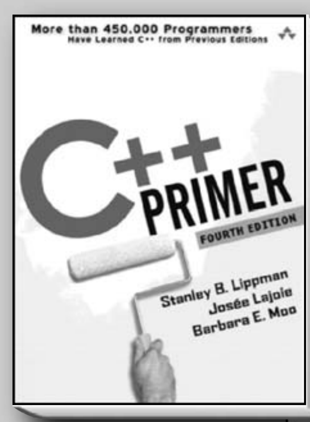
- *READ CHAPTER 1: "Getting Started" online at: www.awprofessional.com/title/0201721481*

C++ COMMON KNOWLEDGE: Essential Intermediate Programming

Stephen C. Dewhurst

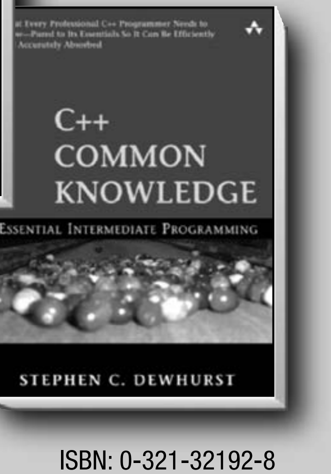
What every professional C++ programmer needs to know, pared to its essentials for efficient and accurate understanding.

- *READ ITEM 2: "Polymorphism" online at: www.awprofessional.com/title/0321321928*



More than 450,000 Programmers Have Learned C++ from Previous Editions
C++ PRIMER
FOURTH EDITION
Stanley B. Lippman
Josée Lajoie
Barbara E. Moo

ISBN: 0-201-72148-1



What Every Professional C++ Programmer Needs to Know—Pared to Its Essentials So It Can Be Efficiently and Accurately Absorbed
C++ COMMON KNOWLEDGE
ESSENTIAL INTERMEDIATE PROGRAMMING
STEPHEN C. DEWHURST

ISBN: 0-321-32192-8

Available at technical bookstores everywhere, or order online at ***www.awprofessional.com***

Addison
Wesley