inside:

CONFERENCE REPORTS

USENIX 2002

conference reports

2002 USENIX Annual Technical Conference
MONTEREY, CALIFORNIA, USA
JUNE 10-15, 2002

ANNOUNCEMENTS

Summarized by Josh Simon

The 2002 USENIX Annual Technical Conference was very exciting. The general track had 105 papers submitted (up 28% from 82 in 2001) and accepted 25 (19 from students); the FREENIX track had 53 submitted (up from 52 in 2001) and accepted 26 (7 from students).

The two annual USENIX-given awards were presented by outgoing USENIX Board President Dan Geer. The USENIX Lifetime Achievement Award (also known as the "Flame" because of the shape of the award) went to James Gosling for his contributions, including the Pascal compiler for Multics, Emacs, an early

James Gosling

SMP UNIX, work on X11 and Sun's windowing system, the first enscript, and Java. The Software Tools Users Group (STUG) Award was presented to the Apache Foundation and accepted by Rasmus Lerdorf. In addition to the well-known Web server, Apache produces Jakarta, mod_perl, mod_tcl, and XML parser, with over 80 members in at least 15 countries.

Rasmus Lerdorf

KEYNOTE ADDRESS

THE INTERNET'S COMING SILENT SPRING
Lawrence Lessig, Stanford University

Summarized by David E. Ott

In a talk that received a standing ovation, Lawrence Lessig pointed out the recent legal crisis that is stifling innovation by extending notions of private ownership of technology beyond reasonable limits.

Several lessons from history are instructive: (1) Edwin Armstrong, the creator of FM radio technology, became an enemy to RCA, which launched a legal campaign to suppress the technology; (2) packet switching networks, proposed by Paul Baron, were seen by AT&T as a new, competing technology that had to be suppressed; (3) Disney took Grimm tales, by then in the public domain, and retold them in magically innovative ways. Should building upon the past in this way be considered an offense?

"The truth is, architectures can allow"; that is, freedom for innovation can be built into architectures. Consider the Internet: a simple core allows for an unlimited number of smart end-to-end applications.

Compare an AT&T proprietary core network to the Internet's end-to-end model. The number of innovators for the former is one company, while for the latter it's potentially the number of people connected to it. Innovations for AT&T are designed to benefit the owner, while the Internet is a wide-open board that allows all kinds of benefits to all kinds of groups. The diversity of contributors in the Internet arena is staggering.

The auction of spectrum by the FCC is another case in point. The spectrum is usually seen as a fixed resource characterized by scarcity. The courts have seen spectrum as something to be owned as property. Technologists, however, have shown that capacity can be a function of

architecture. As David Reed argues, capacity can scale with the number of users – assuming an effective technology and an open architecture.

Developments over the last three years are disturbing and can be summarized as two layers of corruption: intellectual-property rights constraining technical innovation, and proprietary hardware platforms constraining software innovation.

Issues have been framed by the courts largely in two mistaken ways: (1) it's their property – a new technology shouldn't interfere with the rights of current technology owners; (2) it's just theft – copyright laws should be upheld by suppressing certain new technologies.

In fact, we must reframe the debate from "it's their property" to a "highways" metaphor that acts as a neutral platform for innovation without discrimination. "Theft" should be reframed as "Walt Disney," who built upon works from the past in richly creative ways that demonstrate the utility of allowing work to reach the public domain.

In the end, creativity depends upon the balance between property and access, public and private, controlled and common access. Free code builds a free culture, and open architectures are what give rise to the freedom to innovate. All of us need to become involved in reframing the debate on this issue; as Rachel Carson's *Silent Spring* points out, an entire ecology can be undermined by small changes from within.

INVITED TALKS

THE IETF, OR, WHERE DO ALL THOSE RFCS COME FROM, ANYWAY?

Steve Bellovin, AT&T Labs – Research

Summarized by Josh Simon

The Internet Engineering Task Force (IETF) is a standards body, but not a legal entity, consisting of individuals (not organizations) and driven by a consensus-based decision model. Anyone who "shows up" – be it at the thrice-

annual in-person meetings or on the email lists for the various groups – can join and be a member. The IETF is concerned with Internet protocols and open standards, not LAN-specific (such as Appletalk) or layer-1 or -2 (like copper versus fiber).

The organizational structure is loose. There are many working groups, each with a specific focus, within several areas. Each area has an area director, who collectively form the Internet Engineering Steering Group (IESG). The six permanent areas are Internet (with working groups for IPv6, DNS, and ICMP), Transport (TCP, QoS, VoIP, and SCTP), Applications (mail, some Web, LDAP), Routing (OSPF, BGP), Operations and Management (SNMP), and Security (IPSec, TLS, S/MIME). There are also two other areas: SubIP is a temporary area for things underneath the IP protocol stack (such as MPLS, IP over wireless, and traffic engineering), and there's a General area for miscellaneous and process-based working groups.

Internet Requests for Comments (RFCs) fall into three tracks: Standard, Informational, and Experimental. Note that this means that not all RFCs are standards. The RFCs in the Informational track are generally for proprietary protocols or April first jokes; those in the Experimental track are results, ideas, or theories.

The RFCs in the Standard track come from working groups in the various areas through a time-consuming, complex process. Working groups are created with an agenda, a problem statement, an email list, some draft RFCs, and a chair. They typically start out as a BoF session. The working group and the IESG make a charter to define the scope, milestones, and deadlines; the Internet Advisory Board (IAB) ensures that the working group proposals are architecturally sound. Working groups are narrowly focused and are supposed to die off once the problem is solved and all milestones achieved. Working groups meet and work mainly through the email list,

though there are three in-person high-bandwidth meetings per year. However, decisions reached in person must be ratified by the mailing list, since not everybody can get to three meetings per year. They produce RFCs which go through the Standard track; these need to go through the entire working group before being submitted for comment to the entire IETF and then to the IESG. Most RFCs wind up going back to the working group at least once from the area director or IESG level.

The format of an RFC is well-defined and requires it be published in plain 7-bit ASCII. They're freely redistributable, and the IETF reserves the right of change control on all Standard-track RFCs.

The big problems the IETF is currently facing are security, internationalization, and congestion control. Security has to be designed into protocols from the start. Internationalization has shown us that 7-bit-only ASCII is bad and doesn't work, especially for those character sets that require more than 7 bits (like Kanji); UTF-8 is a reasonable compromise. But what about domain names? While not specified as requiring 7-bit ASCII in the specifications, most DNS applications assume a 7-bit character set in the namespace. This is a hard problem. Finally, congestion control is another hard problem, since the Internet is not the same as a really big LAN.

INTRODUCTION TO AIR TRAFFIC MANAGEMENT SYSTEMS

Ron Reisman, NASA Ames Research Center; Rob Savoye, Seneca Software

Summarized by Josh Simon

Air traffic control is organized into four domains: *surface*, which runs out of the airport control tower and controls the aircraft on the ground (e.g., taxi and takeoff); *terminal area*, which covers aircraft at 11,000 feet and below, handled by the Terminal Radar Approach Control (TRACON) facilities; *en route*, which covers between 11,000 and 40,000 feet, including climb, descent, and at-

altitude flight, runs from the 20 Air Route Traffic Control Centers (ARTCC, pronounced "artsy"); and *traffic flow management*, which is the strategic arm. Each area has sectors for low, high, and very-high flight. Each sector has a controller team, including one person on the microphone, and handles between 12 and 16 aircraft at a time. Since the number of sectors and areas is limited and fixed, there's limited system capacity. The events of September 11, 2001, gave us a respite in terms of system usage, but based on path growth patterns, the air traffic system will be oversubscribed within two to three years. How do we handle this oversubscription?

Air Traffic Management (ATM) Decision Support Tools (DST) use physics, aeronautics, heuristics (expert systems), fuzzy logic, and neural nets to help the (human) aircraft controllers route aircraft around. The rest of the talk focused on capacity issues, but the DST also handle safety and security issues. The software follows open standards (ISO, POSIX, and ANSI). The team at NASA Ames made Center-TRACON Automation System (CTAS), which is software for each of the ARTCCs, portable from Solaris to HP-UX and Linux as well. Unlike just about every other major software project, this one really is standard and portable; co-presenter Rob Savoye has experience in maintaining gcc on multiple platforms and is the project lead on the portability and standards issues for the code. CTAS allows the ARTCCs to upgrade and enhance individual aspects or parts of the system; it isn't a monolithic all-or-nothing entity like the old ATM systems.

Some future areas of research include a head-mounted augmented reality device for tower operators, to improve their situational awareness by automating human factors; and new digital global positioning system (DGPS) technologies which are accurate within inches instead of feet.

Questions centered around advanced avionics (e.g., getting rid of ground control), cooperation between the US and Europe for software development (we're working together on software development, but the various European countries' controllers don't talk well to each other), and privatization.

ADVENTURES IN DNS
Bill Manning, ISI

Summarized by Brennan Reynolds

Manning began by posing a simple question: is DNS really a critical infrastructure? The answer is not simple. Perhaps seven years ago, when the Internet was just starting to become popular, the answer was a definite no. But today, with IPv6 being implemented and so many transactions being conducted over the Internet, the question does not have a clear-cut answer. The Internet Engineering Task Force (IETF) has several new modifications to the DNS service that may be used to protect and extend its usability.

The first extension Manning discussed was Internationalized Domain Names (IDN). To date, all DNS records are based on the ASCII character set, but many addresses in the Internet's global network cannot be easily written in ASCII characters. The goal of IDN is to provide encoding for hostnames that is fair, efficient, and allows for a smooth transition from the current scheme. The work has resulted in two encoding schemes: ACE and UTF-8. Each encoding is independent of the other, but they can be used together in various combinations. Manning expressed his opinion that while neither is an ideal solution, ACE appears to be the lesser of two evils. A major hindrance getting IDN rolled out into the Internet's DNS root structure is the increase in zone file complexity.

Manning's next topic was the inclusion of IPv6 records. In an IPv4 world, it is possible for administrators to remember the numeric representation of an address. IPv6 makes the addresses too long and complex to be easily remembered. This means that DNS will play a vital role in getting IPv6 deployed. Several new resource records (RR) have been proposed to handle the translation, including AAAA, A6, DNAME and BIT-STRING. Manning commented on the difference between IPv4 and v6 as a transport protocol and that systems tuned for v4 traffic will suffer a performance hit when using v6. This is largely attributed to the increase in data transmitted per DNS request.

The final extension Manning discussed was DNSSec. He introduced this extension as a mechanism that protects the system from itself. DNSSec protects against data spoofing and provides authentication between servers. It includes a cryptographic signature of the RR set to ensure authenticity and integrity. By signing the entire set, the amount of computation is kept to a minimum. The information itself is stored in a new RR within each zone file on the DNS server.

Manning briefly commented on the use of DNS to provide a PKI infrastructure, stating that that was not the purpose of DNS and therefore it should not be used in that fashion. The signing of the RR sets can be done hierarchically, resulting in the use of a single trusted key at the root of the DNS tree to sign all sets to the leafs. However, the job of key replacement and rollover is extremely difficult for a system that is distributed across the globe.

Manning stated that in an operation test bed, with all of these extensions enabled, the packet size for a single query response grew from 518 bytes to larger than 18,000. This results in a large increase of bandwidth usage for high volume name servers and puts. Therefore, in Manning's opinion, not all of these features will be deployed in the near future. For those looking for more information, Bill's Web site can be found at *http://www.isi.edu/otdr*.

The Joy of Breaking Things

Pat Parseghian, Transmeta

Summarized by Juan Navarro

Pat Parseghian described her experience in testing the Crusoe microprocessor at the Transmeta Lab for Compatibility (TLC), where the motto is, "You make it . . . we break it!" and the goal is to make engineers miserable.

She first gave an overview of Crusoe, whose most distinctive feature is a code morphing layer that translates x86 instructions into native VLIW. Such peculiarity makes the testing process particularly challenging because it involves testing two processors (the "external" x86 and the "internal" VLIW) and also because of variations on how the code-morphing layer works (it may interpret code the first few times it sees an instruction sequence and translate and save in a translation cache afterwards). There are also reproducibility issues, since the VLIW processor may run at different speeds to save energy.

Pat then described the tests that they subject the processor to (hardware compatibility tests, common applications, operating systems, envelope-pushing games, and hard-to-acquire legacy applications) and the issues that must be considered when defining testing policies. She gave some testing tips, including organizational issues like tracking resources and results.

To illustrate the testing process, Pat gave a list of possible causes of a system crash that must be investigated. If it is the silicon, then it might be because it is damaged or because of a manufacturing problem or a design flaw. If the problem is in the code-morphing layer, is the fault with the interpreter or with the translator? The fault could also be external to the processor: it could be the homemade system BIOS, a faulty mainboard, or an operator error. Or Transmeta might not be at fault at all: the crash might be due to a bug in the OS or the application. The key to pinpointing the cause of a problem is to isolate it by identifying the conditions that reproduce the problem and repeating those conditions in other test platforms and non-Crusoe systems. Then relevant factors must be identified based on product knowledge, past experience, and common sense.

To conclude, Pat suggested that some of TLC's testing lessons can be applied to products that the audience was involved in, and assured us that breaking things is fun.

Technology, Liberty, Freedom, and Washington

Alan Davidson, Center for Democracy and Technology

Summarized by Steve Bauer

"Experience should teach us to be most on our guard to protect liberty when the Government's purposes are beneficent. Men born to freedom are naturally alert to repel invasion of their liberty by evil-minded rulers. The greatest dangers to liberty lurk in insidious encroachment by men of zeal, well-meaning but without understanding." – Louis Brandeis, *Olmstead v. U.S.*

Alan Davison, an associate director at the CDT (*http://www.cdt.org*) since 1996, concluded his talk with this quote. In many ways it aptly characterizes the importance to the USENIX community of the topics he covered. The major themes of the talk were the impact of law on individual liberty, system architectures, and appropriate responses by the technical community.

The first part of the talk provided the audience with an overview of legislation either already introduced or likely to be introduced in the US Congress. This included various proposals to protect children, such as relegating all sexual content to a .xxx domain or providing kid-safe zones such as .kids.us. Other similar laws discussed were the Children's Online Protection Act and legislation dealing with virtual child pornography.

Other lower-profile pieces covered were laws prohibiting false contact information in emails and domain registration databases. Similar proposals exist that would prohibit misleading subject lines in emails and require messages to clearly identify if they are advertisements. Briefly mentioned were laws impacting online gambling.

Bills and laws affecting the architectural design of systems and networks and various efforts to establish boundaries and borders in cyberspace were then discussed. These included the Consumer Broadband and Television Promotion Act and the French cases against Yahoo and its CEO for allowing the sale of Nazi materials on the Yahoo auction site.

A major part of the talk focused on the technological aspects of the USA-Patriot Act passed in the wake of the September 11 attacks. Topics included "pen registers" for the Internet, expanded government power to conduct secret searches, roving wiretaps, nationwide service of warrants, sharing grand jury and intelligence information, and the establishment of an identification system for visitors to the US.

The technological community needs to be aware and care about the impact of law on individual liberty and the systems the community builds. Specific suggestions included designing privacy and anonymity into architectures and limiting information collection, particularly any personally identifiable data, to only what is essential. Finally, Alan emphasized that many people simply do not understand the implications of law. Thus the technology community has a important role in helping make these implications clear.

Taking an Open Source Project to Market

Eric Allman, Sendmail Inc.

Summarized by Scott Kilroy

Eric started the talk with the upsides and downsides to open source software. The upsides include rapid feedback, high-

quality work (mostly), lots of hands, and an intensely engineering-driven approach. The downsides include no support structure, limited project marketing expertise, and volunteers having limited time.

In 1998 Sendmail was becoming a success disaster. A success disaster includes two key elements: (1) volunteers start spending all their time supporting current functionality instead of enhancing (this heavy burden can lead to project stagnation and, eventually, death); (2) competition for the better-funded sources can lead to FUD (fear, uncertainty, and doubt) about the project.

Eric then led listeners down the path he took to turn Sendmail into a business. Eric envisioned Sendmail Inc. as a small-ish and close-knit company but soon realized that the family atmosphere he desired could not last as the company grew. He observed that maybe only the first 10 people will buy into your vision, after which each individual is primarily interested in something else (e.g., power, wealth, or status). He warns that there will be people working for you that you do not like. Eric particularly did not care for sales people, but he emphasized how important sales, marketing, finance, and legal people are to companies.

The nature of companies in infancy can be summed up as: you need money, and you need investors in order to get money. Investors want a return on investment, so money management becomes critical. Companies therefore must optimize money functions. Eric's experience with investors were lessons all in themselves. More than giving money, good investors can provide connections and insight, so you don't want investors who don't believe in you.

A company must have bug history, change management, and people who understand the product and have a sense of the target market. Eric now sees the importance of marketing target research. A company can never forget

the simple equation that drives business: profit = revenue - expense.

Finally, the concept of value should be based on what is valuable to the customer. Eric observed that his customers needed documentation, extra value in the product that they couldn't get elsewhere, and technical support.

Eric learned hard lessons along the way. If you want to do interesting open source, it might be best not to be too successful. You don't want to let the larger dragons (companies) notice you. If you want a commercial user base, you have to manage process, watch corporate culture, provide value to customers, watch bottom lines, and develop a thick skin.

Starting a company is easy but perpetuating it is extremely difficult!

INFORMATION VISUALIZATION FOR SYSTEMS PEOPLE

Tamara Munzner, University of British Columbia

Summarized by J.D. Welch

Munzner presented interesting evidence that information visualization through interactive representations of data can help people to perform tasks more effectively and reduce the load on working memory.

A popular technique in visualizing data is abstracting it through node-link representation – for example, a list of cross-references in a book. Representing the relationships between references in a graphical (node-link) manner "offloads cognition to the human perceptual system." These graphs can be produced manually, but automated drawing allows for increased complexity and quicker production times.

The way in which people interpret visual data is important in designing effective visualization systems. Attention to pre-attentive visual cues are key to success. Picking out a red dot among a field of identically shaped blue dots is significantly faster than the same data repre-

sented in a mixture of hue and shape variations. There are many pre-attentive visual cues, including size, orientation, intersection, and intensity. The accuracy of these cues can be ranked, with position triggering high accuracy and color or density on the low end.

Visual cues combined with different data types – quantitative (e.g., 10 inches), ordered (small, medium, large) and categorical (apples, oranges) – can also be ranked, with spatial position being best for all types. Beyond this commonality, however, accuracy varied widely, with length ranked second for quantitative data, eighth for ordinal, and ninth for categorical data types.

These guidelines about visual perception can be applied to information visualization, which, unlike scientific visualization, focuses on abstract data and choice of specialization. Several techniques are used to clarify abstract data, including multi-part glyphs, where changes in individual parts are incorporated into an easier to understand gestalt, interactivity, motion, and animation. In large data sets, techniques like "focus + context," where a zoomed portion of the graph is shown along with a thumbnail view of the entire graph, are used to minimize user disorientation.

Future problems include dealing with huge databases, such as the Human Genome, reckoning dynamic data, like the changing structure of the Web or real-time network monitoring, and transforming "pixel bound" displays into large "digital wallpaper"-type systems.

FIXING NETWORK SECURITY BY HACKING THE BUSINESS CLIMATE

Bruce Schneier, Counterpane Internet Security

Summarized by Florian Buchholz

Bruce Schneier identified security as one of the fundamental building blocks of the Internet. A certain degree of security is needed for all things on the Net, and the limits of security will become limits of the Net itself. However, companies are

hesitant to employ security measures. Science is doing well, but one cannot see the benefits in the real world. An increasing number of users means more problems affecting more people. Old problems such as buffer overflows haven't gone away, and new problems show up. Furthermore, the amount of expertise needed to launch attacks is decreasing.

Schneier argues that complexity is the enemy of security and that while security is getting better, the growth in complexity is outpacing it. As security is fundamentally a people problem, one shouldn't focus on technologies but, rather, should look at businesses, business motivations, and business costs. Traditionally, one can distinguish between two security models: threat avoidance, where security is absolute, and risk management, where security is relative and one has to mitigate the risk with technologies and procedures. In the latter model, one has to find a point with reasonable security at an acceptable cost.

After a brief discussion on how security is handled in businesses today, in which he concluded that businesses "talk big about it, but do as little as possible," Schneier identified four necessary steps to fix the problems.

First, enforce liabilities. Today, no real consequences have to be feared from security incidents. Holding people accountable will increase the transparency of security and give an incentive to make processes public. As possible options to achieve this, he listed industry-defined standards, federal regulation, and lawsuits. The problems with enforcement, however, lie in difficulties associated with the international nature of the problem and the fact that complexity makes assigning liabilities difficult. Furthermore, fear of liability could have a chilling effect on free software development and could stifle new companies.

As a second step, Schneier identified the need to allow partners to transfer liability. Insurance would spread liability risk among a group and would be a CEO's primary risk analysis tool. There is a need for standardized risk models and protection profiles and for more securities as opposed to empty press releases. Schneier predicts that insurance will create a marketplace for security where customers and vendors will have the ability to accept liabilities from each other. Computer-security insurance should soon be as common as household or fire insurance, and from that development, insurance will become the driving factor of the security business.

The next step is to provide mechanisms to reduce risks, both before and after software is released; techniques and processes to improve software quality, as well as an evolution of security management, are therefore needed. Currently, security products try to rebuild the walls – such as physical badges and entry doorways – that were lost when getting connected. Schneier believes this "fortress metaphor" is bad; one should think of the problem more in terms of a city. Since most businesses cannot afford proper security, outsourcing is the only way to make security scalable. With outsourcing, the concenpt of best practices becomes important and insurance companies can be tied to them; outsourcing will level the playing field.

As a final step, Schneier predicts that rational prosecution and education will lead to deterrence. He claims that people feel safe because they live in a lawful society, whereas the Internet is classified as lawless, very much like the American West in the 1800s or a society ruled by warlords. This is because it is difficult to prove who an attacker is; prosecution is hampered by complicated evidence gathering and irrational prosecution. Schneier believes, however, that education will play a major role in turning the Internet into a lawful society. Specifically, he pointed out that we need laws that can be explained.

Risks won't go away; the best we can do is manage them. A company able to manage risk better will be more profitable, and we need to give CEOs the necessary risk-management tools.

There were numerous questions. Listed below are the more complicated ones in a paraphrased Q&A format:

Q: Will liability be effective; will insurance companies be willing to accept the risks? A: The government might have to step in; it needs to be seen how it plays out.

Q: Is there an analogy to the real world in the fact that in a lawless society only the rich can afford security? Security solutions differentiate between classes. A: Schneier disagreed with the statement, giving an analogy to front-door locks, but conceded that there might be special cases.

Q: Doesn't homogeneity hurt security? A: Homogeneity is oversold. Diverse types can be more survivable, but given the limited number of options, the difference will be negligible.

Q: Regulations in airbag protection have led to deaths in some cases. How can we keep the pendulum from swinging the other way? A: Lobbying will not be prevented. An imperfect solution is probable; there might be reversals of requirements such as the airbag one.

Q: What about personal liability? A: This will be analogous to auto insurance: liability comes with computer/Net access.

Q: If the rule of law is to become reality, there must be a law enforcement function that applies to a physical space. You cannot do that with any existing government agency for the whole world. Should an organization like the UN assume that role? A: Schneier was not convinced global enforcement is possible.

Q: What advice should we take away from this talk? A: Liability is coming. Since the network is important to our infrastructure, eventually the problems

will be solved in a legal environment. You need to start thinking about how to solve the problems and how the solutions will affect us.

The entire speech (including slides) is available at *http://www.counterpane. com/presentation4.pdf.*

LIFE IN AN OPEN SOURCE STARTUP
Daryll Strauss, Consultant

Summarized by Teri Lampoudi

This talk was packed with morsels of insight and tidbits of information about what life in an open source startup is like (though "was like" might be more appropriate); what the issues are in starting, maintaining, and commercializing an open source project; and the way hardware vendors treat open source developers. Strauss began by tracing the timeline of his involvement with developing 3-D graphics support for Linux: from the 3dfx voodoo1 driver for Linux in 1997, to the establishment of Precision Insight in 1998, to its buyout by VA Linux, to the dismantling of his group by VA, to what the future may hold.

Obviously, there are benefits from doing open source development, both for the developer and for the end user. An important point was that open source develops entire technologies, not just products. A misapprehension is the inherent difficulty of managing a project that accepts code from a large number of developers, some volunteer and some paid. The notion that code can just be thrown over the wall into the world is a Big Lie.

How does one start and keep afloat an open source development company? In the case of Precision Insight, the subject matter – developing graphics and OpenGL for Linux – required a considerable amount of expertise: intimate knowledge of the hardware, the libraries, X and the Linux kernel, as well as the end applications. Expertise is marketable. What helps even further is having a *visible* virtual team of experts, people who have an established track record of contributions to open source

in the particular area of expertise. Support from the hardware and software vendors came next. While everyone was willing to contract PI to write the portions of code that were specific to their hardware, no one wanted to pay the bill for developing the underlying foundation that was necessary for the drivers to be useful. In the PI model, the infrastructure cost was shared by several customers at once, and the technology kept moving. Once a driver is written for a vendor, the vendor is perfectly capable of figuring out how to write the next driver necessary, thereby obviating the need to contract the job. This and questions of protecting intellectual property – hardware design in this case – are deeply problematic with the open source mode of development. This is not to say that there is no way to get over them, but that they are likely to arise more often than not.

Management issues seem equally important. In the PI setting, contracts were flexible – both a good and a bad thing – and developers overworked. Further, development "in a fishbowl," that is, under public scrutiny, is not easy. Strauss stressed the value of good communication and the use of various out-of-sync communication methods, like IRC and mailing lists.

Finally, Strauss discussed what portions of software can be free and what can be proprietary. His suggestion was that horizontal markets want to be free, whereas vertically, one can develop proprietary solutions. The talk closed with a small stroll through the problems that project politics brings up, from things like merging branches to mailing list and source tree readability.

FILE SYSTEMS
Summarized by Haijin Yan

STRUCTURE AND PERFORMANCE OF THE DIRECT ACCESS FILE SYSTEM

Kostas Magoutis, Salimah Addetia, Alexandra Fedorova, and Margo I. Seltzer, Harvard University; Jeffrey Chase, Andrew Gallatin, Richard Kisley, and Rajiv Wickremesinghe, Duke University; and Eran Gabber, Lucent Technologies

This paper won the Best Paper award.

The Direct Access File System (DAFS) is a new standard for network-attached storage over direct-access transport networks. DAFS takes advantage of user-level network interface standards that enable client-side functionality in which remote data access resides in a library rather than in the kernel. This reduces the overhead of memory copy for data movement and protocol overhead. Remote Direct Memory Access (RDMA) is a direct access transport network that allows the network adapter to reduce copy overhead by accessing application buffers directly.

This paper explores the fundamental structural and performance characteristics of network file access using a user-level file system structure on a direct-access transport network with RDMA. It describes DAFS-based client and server reference implementations for FreeBSD and reports experimental results, comparing DAFS to a zero-copy NFS implementation. It illustrates the benefits and trade-offs of these techniques to provide a basis for informed choices about deployment of DAFS-based systems and similar extensions to other network file protocols such as NFS. Experiments show that DAFS gives applications direct control over an I/O system and increases the client CPU's usage while the client is doing I/O. Future work includes how to address longstanding problems related to the integration of the application and file system for high-performance applications.

CONQUEST: BETTER PERFORMANCE THROUGH A DISK/PERSISTENT-RAM HYBRID FILE SYSTEM

An-I A. Wang, Peter Reiher, and Gerald J. Popek, UCLA; Geoffrey M. Kuenning, Harvey Mudd College

Motivated by the declining cost of persistent RAM, the authors propose the Conquest file system, which stores all small files, metadata, executables, and shared libraries in persistent RAM; disks hold only the data content of remaining large files. Compared to alternatives such as caching and RAM file systems, Conquest has the advantages of efficiency, consistency, and reliability at a reduced cost. Using popular benchmarks, experiments show that Conquest incurs little overhead while achieving faster performance. Future work includes designing mechanisms for adjusting file-size threshold dynamically and finding a better disk layout for large data blocks.

EXPLOITING GRAY-BOX KNOWLEDGE OF BUFFER-CACHE MANAGEMENT

Nathan C. Burnett, John Bent, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau, University of Wisconsin, Madison

Knowing what algorithm is used to manage the buffer cache is very important for improving application performance. However, there is currently no interface for finding this algorithm. This paper introduces Dust, a simple fingerprinting tool that is able to identify the buffer-cache replacement policy. Dust automatically identifies the cache size and replacement policy based on the configuring attributes of access orders, recency, frequency, and long-term history. Through simulation, Dust was able to distinguish between a variety of replacement algorithm policies found in the literature: FIFO, LRU, LFU, Clock, Segmented FIFO, 2Q,and LRU-K. Further experiments of fingerprinting real operating system such as NetBSD, Linux, and Solaris show that Dust is able to identify the hidden cache replacement algorithm.

By knowing the underlying cache replacement policy, a cache-aware Web server can reschedule the requests on a cached request-first policy to obtain performance improvement. Experiments show that cache-aware scheduling improves average response time and system throughput.

OPERATING SYSTEMS (AND DANCING BEARS)

Summarized by Matt Butner

THE JX OPERATING SYSTEM

Michael Golm, Meik Felser, Christian Wawersich, and Jürgen Kleinöder, University of Erlagen-Nürnberg

The talk opened by emphasizing the need for, and the practicality of, a functional Java OS. Such implementation in the JX OS attempts to mimic the recent trend toward using highly abstracted languages in application development in order to create an OS as functional and powerful as one written in a lower-level language.

The JX is a micro-kernel solution that uses separate JVMs for each entity of the kernel, and, in some cases, for each application. The separated domains do not share objects and have no thread migration, and each domain implements its own code. The interesting part of the presentation was the discussion of system-level programming with Java; key areas discussed were memory management and interrupt handlers. The authors concluded by noting that their type-safe and modular system resulted in a robust system with great configuration flexibility and acceptable performance.

DESIGN EVOLUTION OF THE EROS SINGLE-LEVEL STORE

Jonathan S. Shapiro, Johns Hopkins University; Jonathan Adams, University of Pennsylvania

This presentation outlined current characteristics of file systems, and some of their least desirable characteristics. It was based on the revival of the EROS

single-level-store approach for current attempts to capitalize on system consistency and efficiency. Their solution did not relate directly to EROS but, rather, to the design and use of the constructs and notions on which EROS is based. By extending the mapping of the memory system to include the disk, systems are further able to ensure global persistence without regard for a disk structure. In explaining the need for absolute system consistency and an environment which by definition is not allowed to crash, the goal of having such an exhaustive design becomes clear. The cool part of this work is the availability of its design and architecture to the public.

THINK: A SOFTWARE FRAMEWORK FOR COMPONENT-BASED OPERATING SYSTEM KERNELS

Jean-Philippe Fassino, France Télécom R&D; Jean-Bernard Stefani, INRIA; Julia Lawall, DIKU; Gilles Muller, INRIA

This presentation discussed the need for component-based operating systems and respective structures to ensure flexibility for arbitrary-sized systems. Think provides a binding model that maps uniformed components for OS developers and architects to follow, ensuring consistent implementations for an arbitrary system size. However, the goal is not to force developers into a predefined kernel but to promote the use of certain components in varied ways.

Think's concentration is primarily on embedded systems, where short development time is necessary but is constrained by rigorous needs and limited resources. The need to build flexible systems in such an environment can be costly and implementation-specific, but Think creates an environment supported by the ability to dynamically load type-safe components. This allows for more flexible systems that retain functionality because of Think's ability to bind fine-grained components. Benchmarks revealed that dedicated micro-kernels can show performance improvements in comparison to mono-

lithic kernels. Notable future work includes building components for low-end appliances and the development of a real-time OS component library.

BUILDING SERVICES
Summarized by Pradipta De

NINJA: A FRAMEWORK FOR NETWORK SERVICES

J. Robert von Behren, Eric A. Brewer, Nikita Borisov, Michael Chen, Matt Welsh, Josh MacDonald, Jeremy Lau, and David Culler, University of California, Berkeley

Robert von Behren presented Ninja, an ongoing project that aims to provide a framework for building robust and scalable Internet-based network services, like Web-hosting, instant-messaging, email, and file-sharing applications. Robert drew attention to the difficulties of writing cluster applications. One has to take care of data consistency and issues of concurrency, and for robust applications there are problems related to fault tolerance. Ninja works as a wrapper to relieve the user of these problems. The goal of the project is building network services that are scalable and highly available; maintaining persistent data; providing graceful degradation; and supporting online evolution.

The use of clusters distinguishes this setup from generic distributed systems in terms of reliability and security, as well as providing a partition-free network. The next important feature of this project is the new programming model, which is more restrictive than a general programming model but still expressive enough to write most of the applications. This model, described as "single program multiple connection," uses intertask parallelism instead of multi-threaded concurrency and is characterized by all nodes running the same program, with connections delegated to the nodes by a centralized connection manager (CM). A CM takes care of hiding the details of mapping an external connection to an internal node. Ninja

also uses a design called "staged event-driven architecture," where each service is broken down into a set of stages connected by event queues. This architecture is suitable for a modular design and helps in graceful degradation by adaptive load shedding from the event queues.

The presentation concluded with examples of implementation and evaluation of a Web server and an email system called NinjaMail, to show the ease of authoring and the efficacy of using the Ninja framework for service development.

USING COHORT-SCHEDULING TO ENHANCE SERVER PERFORMANCE

James R. Larus and Michael Parkes, Microsoft Research

James Larus presented a new scheduling policy for increasing the throughput of server applications. Cohort scheduling batches execution of similar operations arising in different server requests. The usual programming paradigm in handling server requests is to spawn multiple concurrent threads and switch from one thread to another whenever a thread gets blocked for I/O. Since the threads in a server mostly execute unrelated pieces of code, the locality of reference is reduced; hence the effectiveness of different caching mechanisms. One way to solve this problem is to throw in more hardware. But Larus presented a complementary view where the program behavior is investigated and used to improve the performance.

The problem in this scheme is to identify pieces of code that can be batched together for processing. One simple way is to look at the next program counter values and use them to club threads together. However, this talk presented a new programming abstraction, "staged computation," which replaces the thread model with "stages." A stage is an abstraction for operations with similar behavior and locality. The StagedServer library can be used for programming in this model. It is a collection of C++

classes that implement staged computation and cohort scheduling on either a uniprocessor or multiprocessor. It can be used to define stages.

The presentation showed the experimental evaluation of the cohort scheduling over the thread-based model by implementing two servers: a Web server, which is mainly I/O bound, and a publish-subscribe server, which is mainly compute bound. The SURGE benchmark was used for the first experiment and the Fabret workload for the second. The results showed that cohort-scheduling-based implementation gave a better throughput than the thread-based implementation at high loads.

NETWORK PERFORMANCE
Summarized by Xiaobo Fan

ETE: PASSIVE END-TO-END INTERNET SERVICE PERFORMANCE MONITORING

Yun Fu, Amin Vahdat, Duke University; Ludmila Cherkasova, Wenting Tang, HP Labs

This paper won the Best Student Paper award.

Ludmila Cherkasova began by listing several questions most Web service providers want answered in order to improve service quality. She reviewed the difficulties of making accurate and efficient end-to-end Web service measurement and the shortcomings of currently available approaches. They propose a passive trace-based architecture, called EtE, to monitor Web server performance on behalf of end users.

The first step is to collect network packets passively. The second module reconstructs all TCP connections and extracts HTTP transactions. To reconstruct Web page accesses, they first build a knowledge base indexed by client IP and URL and then group objects to the related Web pages they are embedded in. Statistical analysis is used to handle non-matched objects. EtE Monitor can generate three groups of metrics to measure Web service performance: response time, Web caching, and page abortion.

To demonstrate the benefits of EtE monitor, Cherkasova talked about two case studies and, based on the calculated metrics, gave some insightful explanations about what's happening behind the variations of Web performance. Through validation they claim their approach provides a very close approximation to the real scenario.

THE PERFORMANCE OF REMOTE DISPLAY MECHANISMS FOR THIN-CLIENT COMPUTING

S. Jae Yang, Jason Nieh, Matt Selsky, Nikhil Tiwari, Columbia University

Noting the trend toward thin-client computing, the authors compared different techniques and design choices in measuring the performance of six popular thin-client platforms – Citrix MetaFrame, Microsoft Terminal Services, Sun Ray, Tarantella, VNC, and X. After pointing out several challenges in benchmarking thin clients, Yang proposed slow-motion benchmarking to achieve non-invasive packet monitoring and consistent visual quality. Basically, they insert delays between separate visual events in the benchmark applications of the server side so that the client's display update can catch up with the server's processing speed. The experiments are conducted on an emulated network over a range of network bandwidths.

Their results show that thin clients can provide good performance for Web applications in LAN environments, but only some platforms performed well for video benchmark. Pixel-based encoding may achieve better performance and bandwidth efficiency than high-level graphics. Display caching and compression should be used with care.

A MECHANISM FOR TCP-FRIENDLY TRANSPORT-LEVEL PROTOCOL COORDINATION

David E. Ott and Ketan Mayer-Patel, University of North Carolina, Chapel Hill

A revised transport-level protocol optimized for cluster-to-cluster (C-to-C) applications is what David Ott tries to explain in his talk. A C-to-C application class is identified as one set of processes communicating to another set of processes across a common Internet path. The fundamental problem with C-to-C applications is how to coordinate all C-to-C communication flows so that they share a consistent view of the common C-to-C network, adapt to changing network conditions, and cooperate to meet specific requirements. Aggregation points (AP) are placed at the first and last hop of the common data path to probe network conditions (latency, bandwidth, loss rate, etc.). To carry and transfer this information, a new protocol – Coordination Protocol (CP) – is inserted between the network layer (IP) and the transport layer (TCP, UCP, etc.). Ott illustrated how the CP header is updated and used when packets originate from source and traverse through local and remote AP to arrive at their destination, and how AP maintains a per-cluster state table and detects network conditions. Through simulation results, this coordination mechanism appears effective in sharing common network resources among C-to-C communication flows.

STORAGE SYSTEMS
Summarized by Praveen Yalagandula

MY CACHE OR YOURS? MAKING STORAGE MORE EXCLUSIVE

Theodore M. Wong, Carnegie Mellon University; John Wilkes, HP Labs

Theodore Wong explained the inefficiency of current "inclusive" caching schemes in storage area networks – when a client accesses a block, the block is read from the disk and is cached at both the disk array cache and at the client's cache. Then he presented the concept of "exclusive" caching, where a block accessed by a client is only cached in that client's cache. On eviction from the client's cache, they have come up with a DEMOTE operation to move the data block to the tail of the array cache.

The new exclusion caching schemes are evaluated on both single-client and multiple-client systems. The single-client results are presented for two different types of synthetic workloads: Random (transaction-processing type workloads) and Zipf (typical Web workloads). The exclusive policy was quite effective in achieving higher hit rates and lower read latencies. They also showed a 2.2 times hit-rate improvement over inclusive techniques in the case of a real-life workload, httpd, with a single-client setting.

The DEMOTE scheme also performed well in the case of multiple-client systems when the data accessed by clients is disjointed. In the case of shared data workloads, this scheme performed worse than the inclusive schemes. Theodore then presented an adaptive exclusive caching scheme – a block accessed by a client is placed at the tail in the client's cache and also in the array cache at an appropriate place determined by the popularity of the block. The popularity of the block is measured by maintaining a ghost cache to accumulate the number of times each block is accessed. This new adaptive scheme has achieved a maximum of 52% speedup in the mean latency in the experiments with real-life workloads.

For more information, visit *http://www.cs.cmu.edu/~tmwong/research* and *http://www.hpl.hp.com/research/itc/csl/ssp.*

BRIDGING THE INFORMATION GAP IN STORAGE PROTOCOL STACKS

Timothy E. Denehy, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau, University of Wisconsin, Madison

Currently there is a huge information gap between storage systems and file systems. The interface exposed by storage systems to file systems, based on blocks and providing only read/write interfaces, is very narrow. This leads to poor performance as a whole because of duplicated functionality in both systems and

reduced functionality resulting from storage systems' lack of file information.

The speaker presented two enhancements: ExRaid, an exposed RAID, and I.LFS, Informed Log-Structured File System. ExRAID is an enhancement to the block-based RAID storage system that exposes the following three types of information to the file system: (1) regions – contiguous portions of the address space comprising one or multiple disks; (2) performance information about queue lengths and throughput of the regions revealing disk heterogeneity to the file systems; and (3) failure information – dynamically updated information conveying the number of tolerable failures in each region.

I.LFS allows online incremental expansion of the storage space, performs dynamic parallelism using ExRAID's performance information to perform well on heterogeneous storage systems, and provides a range of different mechanisms with different granularities for redundancy of files using ExRAID's region failure characteristics. These new features are added to LFS with only a 19% increase in the code size.

More information:
http://www.cs.wisc.edu/wind.

MAXIMIZING THROUGHPUT IN REPLICATED DISK STRIPING OF VARIABLE BIT-RATE STREAMS
Stergios V. Anastasiadis, Duke University; Kenneth C. Sevcik and Michael Stumm, University of Toronto

There is an increasing demand for the continuous real-time streaming of media files. Disk striping is a common technique used for supporting many connections on the media servers. But even with 1.2 million hours of mean time between failures, there will be more than one disk failure per week with 7000 disks. Fault tolerance can be achieved by using data replication and reserving extra bandwidth during normal operation. This work focused on supporting the most common variable bit-rate stream formats (e.g., MPEG).

The authors used prototype media server EXEDRA to implement and evaluate different replication and bandwidth reservation schemes. This system supports a variable bit-rate scheme, does stride-based disk allocation, and is capable of supporting variable-grain disk striping. Two replication policies are presented: deterministic – data blocks are replicated in a round-robin fashion on the secondary replicas; random – data blocks are replicated on the randomly chosen secondary replicas. Two bandwidth reservation techniques are also presented: mirroring reservation – disk bandwidth is reserved for both primary and replicas of the media file during the playback; and minimum reservation – a more efficient scheme in which bandwidth is reserved only for the sum of primary data access time and the maximum of the backup data access times required in each round.

Experimental results showed that deterministic replica placement is better than random placement for small disks, minimum disk bandwidth reservation is twice as good as mirroring in the throughput achieved, and fault tolerance can be achieved with a minimal impact on the throughput.

TOOLS
Summarized by Li Xiao

SIMPLE AND GENERAL STATISTICAL PROFILING WITH PCT
Charles Blake and Steven Bauer, MIT

This talk introduced a Profile Collection Toolkit (PCT) – a sampling-based CPU profiling facility. A novel aspect of PCT is that it allows sampling of semantically rich data such as function call stacks, function parameters, local or global variables, CPU registers, or other execution contexts. The design objectives of PCT were driven by user needs and the inadequacies or inaccessibility of prior systems.

The rich data collection capability is achieved via a debugger-controller program, dbct1. The talk then introduced the profiling collector and data collec-

tion activation. PCT file format options provide flexibility and various ways to store exact states. PCT also has analysis options; two examples were given – "Simple" and "Mixed User and Linux Code."

The talk then went into how general sampling helps in the following areas: a debugger-controller state machine; example script fragments; a simple example program; and general value reports in terms of call-path histograms, numeric value histograms, and data generality. Related work, such as gprof and Expect, was then summarized. The contribution of this work is to provide a portable general value sampling tool. The limitations are that PCT does not support much of strip, and count inaccuracies happen because of its statistical nature. Based on this limitation, -g is preferred over strip.

Possible future directions included supporting more debuggers, such as dbx and xdb; script generators for other kinds of traces; more canned reports for general values; and a libgdb-based library sampler. PCT is available at *http://pdos.lcs.mit.edu/pct.* PCT runs on almost any UNIX-like system.

ENGINEERING A DIFFERENCING AND COMPRESSION DATA FORMAT
David G. Korn and Kiem Phong Vo, AT&T Laboratories – Research

This talk began with an equation: "Differencing + Compression = Delta Compression." Compression removes information redundancy in a data set. Examples are gzip, bzip, compress, and pack. Differencing encodes differences between two data sets. Examples are diff -e, fdelta, bdiff, and xdelta. Delta compression compresses a data set given another, combining differencing and compression, and reduces to pure compression when there's no commonality.

After presenting an overall scheme for a delta compressor and showing delta compression performance, this talk discussed the encoding format of a newly designed Vcdiff for delta compression. A

Vcdiff instruction code table consists of 256 entries of each coding up to a pair of instructions, and recodes I-byte indices and any additional data.

The talk then showed Vcdiff's performance with Web data collected from CNN, compared with W3C standard Gdiff, Gdiff+gzip, and diff+gzip, where Gdiff was computed using Vcdiff delta instructions. The results of two experiments, "First" and "Successive," were presented. In "First," each file is compressed against the first file collected; in "Successive," each file is compressed against the one in the previous hour. The diff+gzip did not work well because diff was line-oriented. Vcdiff performed favorably compared with other formats.

Vcdiff is part of the Vcodex package; Vcodex is a platform for all common data transformations, delta compression, plain compression, encryption, transcoding (e.g., uuencode, base64). It is structured in three layers for maximum usability. Base library uses Displines and Methods interfaces.

The code for Vcdiff can be found at *http://www.research.att.com/sw/tools.* They are moving Vcdiff to the IETF standard as a comprehensive platform for transforming data. Please refer to *http://www.ietf.org/internet-draft/ draft-korn-vcdiff.06.txt.*

WHERE IN THE NET . . .
Summarized by Amit Purohit

A PRECISE AND EFFICIENT EVALUATION OF THE PROXIMITY BETWEEN WEB CLIENTS AND THEIR LOCAL DNS SERVERS
Zhuoqing Morley Mao, University of California, Berkeley; Charles D. Cranor, Fred Douglis, Michael Rabinovich, Oliver Spatscheck, and Jia Wang, AT&T Labs – Research

Content Distribution Networks (CDNs) attempt to improve Web performance by delivering Web content to end users from servers located at the edge of the network. When a Web client requests content, the CDN dynamically chooses a server to route the request to, usually the one that is nearest the client. As CDNs have access only to the IP address of the local DNS server (LDNS) of the client, the CDN's authoritative DNS server maps the client's LDNS to a geographic region within a particular network and combines that with network and server load information to perform CDN server selection.

This method has two limitations. First, it is based on the implicit assumption that clients are close to their LDNS. This may not always be valid. Second, a single request from an LDNS can represent a differing number of Web clients – called the hidden load factor. Knowledge of the hidden load factor can be used to achieve better load distribution.

The extent of the first limitation and its impact on the CDN server selection is dealt with. To determine the associations between clients and their LDNS, a simple, non-intrusive, and efficient mapping technique was developed. The data collected was used to study the impact of proximity on DNS-based server selection using four different proximity metrics : (1) autonomous system (AS) clustering – observing whether a client is in the same AS as its LDNS – concluded that LDNS is very good for coarse-grained server selection, as 64% of the associations belong to the same AS; (2) network clustering – observing whether a client is in the same network-aware cluster (NAC) – implied that DNS is less useful for finer-grained server selection, since only 16% of the client and LDNS are in the same NAC; (3) traceroute divergence – the length of the divergent paths to the client and its LDNS from a probe point using traceroute – implies that most clients are topologically close to their LDNS as viewed from a randomly chosen probe site; (4) round-trip time (RTT) correlation (some CDNs select severs based on RTT between the CDN server and the client's LDNS) examines the correlation between the message RTTs from a probe point to the client and its local DNS;

results imply this is a reasonable metric to use to avoid really distant servers.

A study of the impact that client-LDNS associations have on DNS-based server selection concludes that knowing the client's IP address would allow more accurate server selection in a large number of cases. The optimality of the server selection also depends on the server density, placement, and selection algorithms.

Further information can be found at *http://www.eecs.berkeley.edu/~zmao/ myresearch.html* or by contacting *zmao@eecs.berkeley.edu.*

GEOGRAPHIC PROPERTIES OF INTERNET ROUTING
Lakshminarayanan Subramanian, Venkata N. Padmanabhan, Microsoft Research; Randy H. Katz, University of California at Berkeley

Geographic information can provide insights into the structure and functioning of the Internet, including interactions between different autonomous systems, by analyzing certain properties of Internet routing. It can be used to measure and quantify certain routing properties such as circuitous routing, hot-potato routing, and geographic fault tolerance.

Traceroute has been used to gather the required data, and Geotrack tool has been used to determine the location of the nodes along each network path. This enables the computation of "linearized distances," which is the sum of the geographic distances between successive pairs of routers along the path.

In order to measure the circuitousness of a path, a metric "distance ratio" has been defined as the ratio of the linearized distance of a path to the geographic distance between the source and destination of the path. From the data, it has been observed that the circuitousness of a route depends on both geographic and network location of the end hosts. A large value of the distance ratio enables us to flag paths that are highly

circuitous, possibly (though not necessarily) because of routing anomalies. It is also shown that the minimum delay between end hosts is strongly correlated with the linearized distance of the path.

Geographic information can be used to study various aspects of wide-area Internet paths that traverse multiple ISPs. It was found that end-to-end Internet paths tend to be more circuitous than intra-ISP paths, the cause for this being the peering relationships between ISPs. Also, paths traversing substantial distances within two or more ISPs tend to be more circuitous than paths largely traversing only a single ISP. Another finding is that ISPs generally employ hot-potato routing.

Geographic information can also be used to capture the fact that two seemingly unrelated routers can be susceptible to correlated failures. By using the geographic information of routers we can construct a geographic topology of an ISP. Using this we can find the tolerance of an ISP's network to the total network failure in a geographic region.

For further information, contact *lakme@cs.berkeley.edu.*

Providing Process Origin Information to Aid in Network Traceback

Florian P. Buchholz, Purdue University; Clay Shields, Georgetown University

Network traceback is currently limited because host audit systems do not maintain enough information to match incoming network traffic to outgoing network traffic. The talk presented an alternative, assigning origin information to every process and logging it during interactive login creation.

The current implementation concentrates mainly on interactive sessions in which an event is logged when a new connection is established using SSH or Telnet. The method is effective and could successfully determine stepping stones and reliably detect the source of a DDoS attack. The speaker then talked about the related work done in this area,

which led to discussion of the latest development in the area of "host causality."

The speaker ended with the limitations and future directions of the framework. This framework could be extended and could find many applications in the area of security. Current implementation doesn't take care of startup scripts and cron jobs, but incorporating the origin information in FS could solve this problem. In the current implementation, logging is just implemented as a proof of concept. It could be made safe in many ways, and this could be another important aspect of future work.

PROGRAMMING
Summarized by Amit Purohit

Cyclone : A Safe Dialect of C

Trevor Jim, AT&T Labs – Research; Greg Morrisett, Dan Grossman, Michael Hicks, James Cheney, Yanling Wang, Cornell University

Cyclone is designed to provide protection against attacks such as buffer overflows, format string attacks, and memory management errors. The current structure of C allows programmers to write vulnerable programs. Cyclone extends C so that it has the safety guarantee of Java while keeping C syntax, types, and semantics untouched.

The Cyclone compiler performs static analysis of the source code and inserts runtime checks into the compiled output at places where the analysis cannot determine that the code execution will not violate safety constraints. Cyclone imposes many restrictions to preserve safety, such as NULL checks. These checks do not exist in normal C.

The speaker then talked about some sample code written in Cyclone and how it tackles safety problems. Porting an existing C application to Cyclone is pretty easy, with fewer than 10% change required. The current implementation concentrates more on safety than on performance – hence, the performance

penalty is substantial. Cyclone was able to find many lingering bugs. The final step of the project will be to write a compiler to convert a normal C program to an equivalent Cyclone program.

Cooperative Task Management Without Manual Stack Management

Atul Adya, Jon Howell, Marvin Theimer, William J. Bolosky, John R. Douceur, Microsoft Research

The speaker described the definitions and motivations behind the distinct concepts of task management, stack management, I/O management, conflict management, and data partitioning. Conventional concurrent programming uses preemptive task management and exploits the automatic stack management of a standard language. In the second approach, cooperative tasks are organized as event handlers that yield control by returning control to the event scheduler, manually unrolling their stacks. In this project they have adopted a hybrid approach that makes it possible for both stack management styles to coexist. Thus, programmers can code assuming one or the other of these stack management styles will operate, depending upon the application. The speaker also gave a detailed example of how to use their mechanism to switch between the two styles.

The talk then continued with the implementation. They were able to preempt many subtle concurrency problems by using cooperative task management. Paying a cost up-front to reduce a subtle race condition proved to be a good investment.

Though the choice of task management is fundamental, the choice of stack management can be left to individual taste. This project enables use of any type of stack management in conjunction with cooperative task management.

IMPROVING WAIT-FREE ALGORITHMS FOR INTERPROCESS COMMUNICATION IN EMBEDDED REAL-TIME SYSTEMS

Hai Huang, Padmanabhan Pillai, Kang G. Shin, University of Michigan

The main characteristic of the real-time/time-sensitive system is its predictable response time. But concurrency management creates hurdles to achieving this goal because of the use of locks to maintain consistency. To solve this problem, many wait-free algorithms have been developed, but these are typically high-cost solutions. By taking advantage of the temporal characteristics of the system, however, the time and space overhead can be reduced.

The speaker presented an algorithm for temporal concurrency control and applied this technique to improve three wait-free algorithms. A single-writer, multiple-reader, wait-free algorithm and a double-buffer algorithm were proposed.

Using their transformation mechanism, they achieved an improvement of 17–66% in ACET and a 14–70% reduction in memory requirements for IPC algorithms. This mechanism is extensible and can be applied to other non-blocking IPC algorithms as well. Future work involves reducing the synchronizing overheads in more general IPC algorithms with multiple-writer semantics.

MOBILITY

Summarized by Praveen Yalagandula

ROBUST POSITIONING ALGORITHMS FOR DISTRIBUTED AD-HOC WIRELESS SENSOR NETWORKS

Chris Savarese, Jan Rabaey, University of California, Berkeley ; Koen Langendoen, Delft University of Technology

The Pico Radio Network comprises more than a hundred sensors, monitors, and actuators equipped with wireless connectivity with the following properties: (1) no infrastructure, (2) computation in a distributed fashion instead of centralized computation, (3) dynamic topology, (4) limited radio range,

(5) nodes that can act as repeaters, (6) devices of low cost and with low power, and (7) sparse anchor nodes – nodes with GPS capability. A positioning algorithm determines the geographical position of each node in the network.

In two-dimensional space, each node needs three reference positions to estimate its geographical position. There are two problems that make positioning difficult in the Pico Radio Network type setting: (1) a sparse anchor node problem, and (2) a range error problem.

The two-phase approach that the authors have taken in solving the problem consists of: (1) a hop-terrain algorithm – in this first phase, each node roughly guesses its location by the distance calculated using multi-hops to the anchor points; and (2) a refinement algorithm – in this second phase, each node uses its neighbors' positions to refine its own position estimate. To guarantee the convergence, this approach uses confidence metrics: assigning a value of 1.0 for anchor nodes and 0.1 to start for other nodes and increasing these with each iteration.

A simulation tool, OMNet++, was used for both phases and in various scenarios. The results show that they achieved position errors of less than 33% in a scenario with 5% anchor nodes, an average connectivity of 7, and 5% range measurement error.

Guidelines for anchor node deployment are: high connectivity (>10), a reasonable fraction of anchor nodes (> 5%), and, for anchor placement, covered edges.

APPLICATION-SPECIFIC NETWORK MANAGEMENT FOR ENERGY-AWARE STREAMING OF POPULAR MULTIMEDIA FORMATS

Surendar Chandra, University of Georgia; and Amin Vahdat, Duke University

The main hindrance in supporting the increasing demand for mobile multimedia on PDA is the battery capacity of the

small devices. The idle-time power consumption is almost the same as the receive power consumption on typical wireless network cards (1319mW vs. 1425mW), while the sleep state consumes far less power (177mW). To let the system consume energy proportional to the stream quality, the network card should transition to sleep state aggressively between each packet.

The speaker presented previous work, where different multimedia streams were studied for PDAs: MS Media, Real Media, and Quicktime. It was found that if the inter-packet gap is predictable, then huge savings are possible: for example, about 80% savings in MS media streams with few packet losses. The limitation of the IEEE 802.11b power-save mode comes into play when there are two or more nodes, producing a delay between beacon time and when the node receives/sends packets. This badly affects the streams, since higher energy is consumed while waiting.

The authors propose traffic shaping for energy conservation where this is done by proxy such that packets arrive at regular intervals. This is achieved using a local proxy in the access point and a client-side proxy in the mobile host. Simulations show that traffic shaping reduces energy consumption and also reveals that the higher bandwidth streams have a lower energy metric (mJ/kB).

More information is at *http://greenhouse.cs.uga.edu.*

CHARACTERIZING ALERT AND BROWSE SERVICES OF MOBILE CLIENTS

Atul Adya, Paramvir Bahl, and Lili Qiu, Microsoft Research

Even though there is a dramatic increase in Internet access from wireless devices, there are not many studies done on characterizing this traffic. In this paper, the authors characterize the traffic observed on the MSN Web site with both notification and browse traces.

Around 33 million browsing accesses and about 3.25 million notification entries are present in the traces. Three types of analyses are done for each one of these two services: content analysis, concerning the most popular content categories and their distribution; popularity analysis, or the popularity distribution of documents; and user behavior analysis.

The analysis of the notification logs shows that document access rates follow a Zipf-like distribution, with most of the accesses concentrated on a small number of messages; and the accesses exhibit geographical locality – users from same locality tend to receive similar notification content. The browser log analysis shows that a smaller set of URLs are accessed most times, though the access pattern does not fit any Zipf curve; and the highly accessed URLS remain stable. A correlation study between the notification and browsing services shows that wireless users have a moderate correlation of 0.12.

FREENIX TRACK SESSIONS

BUILDING APPLICATIONS
Summarized by Matt Butner

INTERACTIVE 3-D GRAPHICS APPLICATIONS FOR TCL
Oliver Kersting, Jürgen Döllner, Hasso Plattner Institute for Software Systems Engineering, University of Potsdam

The integration of 3-D image rendering functionality into a scripting language permits interactive and animated 3-D development and application without the formalities and precision demanded by low-level C/C++ graphics and visualization libraries. The large and complex C++ API of the Virtual Rendering System (VRS) can be combined with the conveniences of the Tcl scripting language. The mapping of class interfaces is done via an automated process and generates respective wrapper classes, all of which ensures complete API accessibility and functionality without any signifi-

cant performance retribution. The general C++-to-Tcl mapper SWIG grants the necessary object and hierarchal abilities without the use of object-oriented Tcl extensions. The final API mapping techniques address C++ features such as classes, overloaded methods, enumerations, and inheritance relations. All are implemented in a proof-of-concept that maps the complete C++ API of the VRS to Tcl and are showcased in a complete interactive 3-D-map system.

THE AGFL GRAMMAR WORK LAB
Cornelis H.A. Coster, Erik Verbruggen, University of Nijmegen (KUN)

The growth and implementation of Natural Language Processing (NLP) is the cornerstone of the continued evolution and implementation of truly intelligent search machines and services. In part due to the growing collections of computer-stored human-readable documents in the public domain, the implementation of linguistic analysis will become necessary for desirable precision and resolution. Subsequently, such tools and linguistic resources must be released into the public domain, and they have done so with the release of the AGFL Grammar Work Lab under the GNU Public License, as a tool for linguistic research and the development of NLP-based applications.

The AGFL (Affix Grammars over a Finite Lattice) Grammar Work Lab meshes context-free grammars with finite set-valued features that are acceptable to a range of languages. In computer science terms, "Syntax rules are procedures with parameters and a non-deterministic execution." The English Phrases for Information Retrieval (EP4IR), released with the AGFL-GWL as a robust grammar of English, is an AGFL-GWL generated English parser that outputs "Head/Modified" frames. The sentences "CompanyX sponsored this conference" and "This conference was sponsored by CompanyX" both generate [CompanyX,[sponsored, confer-

ence]]. The hope is that public availability of such tools will encourage further development of grammar and lexical software systems.

SWILL: A SIMPLE EMBEDDED WEB SERVER LIBRARY
Sotiria Lampoudi, David M. Beazley, University of Chicago

This paper won the FREENIX Best Student Paper award.

SWILL (Simple Web Interface Link Library) is a simple Web server in the form of a C library whose development was motivated by a wish to give cool applications an interface to the Web. The SWILL library provides a simple interface that can be efficiently implemented for tasks that vary from flexible Web-based monitoring to software debugging and diagnostics. Though originally designed to be integrated with high-performance scientific simulation software, the interface is generic enough to allow for unbounded uses. SWILL is a single-threaded server, relying upon non-blocking I/O through the creation of a temporary server which services I/O requests. SWILL does not provide SSL or cryptographic authentication but does have HTTP authentication abilities.

A fantastic feature of SWILL is its support for SPMD-style parallel applications which utilize MPI, proving valuable for Beowulf clusters and large parallel machines. Another practical application was the implementation of SWILL in a modified Yalnix emulator by University of Chicago Operating Systems courses, which utilized the added Yalnix functionality for OS development and debugging. SWILL requires minimal memory overhead and relies upon the HTTP/1.0 protocol.

NETWORK PERFORMANCE

Summarized by Florian Buchholz

LINUX NFS CLIENT WRITE PERFORMANCE

Chuck Lever, Network Appliance; Peter Honeyman, CITI, University of Michigan

Lever introduced a benchmark to measure an NFS client write performance. Client performance is difficult to measure due to hindrances such as poor hardware or bandwidth limitations. Furthermore, measuring application performance does not identify weaknesses specifically at the client side. Thus a benchmark was developed trying to exercise only data transfers in one direction between server and application. For this purpose, the benchmark was based on the block sequential write portion of the Bonnie file system benchmark. Once a benchmark for NFS clients is established, it can be used to improve client performance.

The performance measurements were performed with an SMP Linux client and both a Linux NFS server and a Network Appliance F85 filer. During testing, a periodic jump in write latency time was discovered. This was due to a rather large number of pending write operations that were scheduled to be written after certain threshold values were exceeded. By introducing a separate daemon that flushes the cached write request, the spikes could be eliminated, but as a result the average latency grows over time. The problem could be traced to a function that scans a linked list of write requests. After having added a hashtable to improve lookup performance, the latency improved considerably.

The improved client was then used to measure throughput against the two servers. A discrepancy between the Linux server and the filer test was noticed and the reason for that traced back to a global kernel lock that was unnecessarily held when accessing the network stack. After correcting this, performance improved further. However,

the measurements showed that a client may run slower when paired with fast servers on fast networks. This is due to heavy client interrupt loads, more network processing on the client side, and more global kernel lock contention.

The source code of the project is available at *http://www.citi.umich.edu/projects/nfs-perf/patches/*

A STUDY OF THE RELATIVE COSTS OF NETWORK SECURITY PROTOCOLS

Stefan Miltchev and Sotiris Ioannidis, University of Pennsylvania; Angelos Keromytis, Columbia University

With the increasing need for security and integrity of remote network services, it becomes important to quantify the communication overhead of IPSec and compare it to alternatives such as SSH, SCP, and HTTPS.

For this purpose, the authors set up three testing networks: direct link, two hosts separated by two gateways, and three hosts connecting through one gateway. Protocols were compared in each setup, and manual keying was used to eliminate connection setup costs. For IPSec the different encryption algorithms AES, DES, 3DES, hardware DES, and hardware 3DES were used. In detail, FTP was compared to SFTP, SCP and FTP over IPSec, HTTP to HTTPS and HTTP over IPSec, and NFS to NFS over IPSec and local disk performance.

The result of the measurements were that IPSec outperforms other popular encryption schemes. Overall, unencrypted communication was fastest, but in some cases, like FTP, the overhead can be small. The use of crypto hardware can significantly improve performance. For future work, the inclusion of setup costs, hardware-accelerated SSL, SFTP, and SSH were mentioned.

CONGESTION CONTROL IN LINUX TCP

Pasi Sarolahti, University of Helsinki; Alexey Kuznetsov, Institute for Nuclear Research at Moscow

Having attended the talk and read the paper, I am still unclear about whether the authors are merely describing the design decisions of TCP congestion control or whether they are actually the creators of that part of the Linux code. My guess leans toward the former.

In the talk, the speaker compared the TCP protocol congestion control measures according to IETF and RFC specifications with the actual Linux implementation, which does conform to the basic principles but nevertheless has differences. A specific emphasis was placed on retransmission mechanisms and the congestion window. Also, several TCP enhancements – the NewReno algorithm, Selective ACKs (SACK), Forward ACKs (FACK) – were discussed and compared.

In some instances, Linux does not conform to the IETF specifications. The fast recovery does not fully follow RFC 2582 since the threshold for triggering retransmit is adjusted dynamically and the congestion window's size is not changed. Also, the roundtrip-time estimator and the RTO calculation differ from RFC 2988 since it uses more conservative RTT estimates and a minimum RTO of 200ms. The performance measures showed that with the additional Linux-specific features enabled, slightly higher throughput, more steady data flow, and fewer unnecessary retransmissions can be achieved.

XTREME XCITEMENT

Summarized by Steve Bauer

THE FUTURE IS COMING: WHERE THE X WINDOW SHOULD GO

Jim Gettys, Compaq Computer Corp.

Jim Gettys, one of the principal authors of the X Window System, outlined the near-term objectives for the system, primarily focusing on the changes and infrastructure required to enable replication and migration of X applications. Providing better support for this functionality would enable users to retrieve or duplicate X applications between their servers at home and work.

CONFERENCE REPORTS

One interesting example of an application that currently is capable of migration and replication is Emacs. To create a new frame on DISPLAY try: "M-x make-frame-on-display <Return> DISPLAY <Return>".

However, technical challenges make replication and migration difficult in general. These include the "major headaches" of server-side fonts, the nonuniformity of X servers and screen sizes, and the need to appropriately retrofit toolkits. Authentication and authorization issues are obviously also important. The rest of the talk delved into some of the details of these interesting technical challenges.

HACKING IN THE KERNEL
Summarized by Hai Huang

AN IMPLEMENTATION OF SCHEDULER ACTIVATIONS ON THE NETBSD OPERATING SYSTEM
Nathan J. Williams, Wasabi Systems

Scheduler activation is an old idea. Basically, there are benefits and drawbacks to using solely kernel-level or user-level threading. Scheduler activation is able to combine the two layers of control to provide more concurrency in the system.

In his talk, Nathan gave a fairly detailed description of the implementation of scheduler activation in the NetBSD kernel. One important change in the implementation is to differentiate the thread context from the process context. This is done by defining a separate data structure for these threads and relocating some of the information that was embedded in the process context to these thread contexts. Stack was especially a concern due to the upcall. Special handling must be done to make sure that the upcall doesn't mess up the stack so that the preempted user-level thread can continue afterwards. Lastly, Nathan explained that signals were handled by upcalls.

AUTHORIZATION AND CHARGING IN PUBLIC WLANS USING FREEBSD AND 802.1X
Pekka Nikander, Ericsson Research NomadicLab

802.1x standards are well known in the wireless community as link-layer authentication protocols. In this talk, Pekka explained some novel ways of using the 802.1x protocols that might be of interest to people on the move. It is possible to set up a public WLAN that would support various charging schemes via virtual tokens which people can purchase or earn and later use.

This is implemented on FreeBSD using the netgraph utility. It is basically a filter in the link layer that would differentiate traffic based on the MAC address of the client node, which is either authenticated, denied, or let through. The overhead for this service is fairly minimal.

ACPI IMPLEMENTATION ON FREEBSD
Takanori Watanabe, Kobe University

ACPI (Advanced Configuration and Power Management Interface) was proposed as a joint effort by Intel, Toshiba, and Microsoft to provide a standard and finer-control method of managing power states of individual devices within a system. Such low-level power management is especially important for those mobile and embedded systems that are powered by fixed-capacity energy batteries.

Takanori explained that the ACPI specification is composed of three parts: tables, BIOS, and registers. He was able to implement some functionalities of ACPI in a FreeBSD kernel. ACPI Component Architecture was implemented by Intel, and it provides a high-level ACPI API to the operating system. Takanori's ACPI implementation is built upon this underlying layer of APIs.

ACCESS CONTROL
Summarized by Florian Buchholz

DESIGN AND PERFORMANCE OF THE OPENBSD STATEFUL PACKET FILTER (PF)
Daniel Hartmeier, Systor AG

Daniel Hartmeier described the new stateful packet filter (pf) that replaced IPFilter in the OpenBSD 3.0 release. IPFilter could no longer be included due to licensing issues and thus there was a need to write a new filter, making use of optimized data structures.

The filter rules are implemented as a linked list which is traversed from start to end. Two actions may be taken according to the rules: "pass" or "block." A "pass" action forwards the packet and a "block" action will drop it. Where more than one rule matches a packet, the last rule wins. Rules that are marked as "final" will immediately terminate any further rule evaluation. An optimization called "skip-steps" was also implemented, where blocks of similar rules are skipped if they cannot match the current packet. These skip-steps are calculated when the rule set is loaded. Furthermore, a state table keeps track of TCP connections. Only packets that match the sequence numbers are allowed. UDP and ICMP queries and replies are considered in the state table, where initial packets will create an entry for a pseudo-connection with a low initial timeout value. The state table is implemented using a balanced binary search tree. NAT mappings are also stored in the state table, whereas application proxies reside in user space. The packet filter also is able to perform fragment reassembly and to modulate TCP sequence numbers to protect hosts behind the firewall.

Pf was compared against IPFilter as well as Linux's Iptables by measuring throughput and latency with increasing traffic rates and different packet sizes. In a test with a fixed rule set size of 100, Iptables outperformed the other two filters, whose results were close to each

other. In a second test, where the rule set size was continually increased, Iptables consistently had about twice the throughput of the other two (which evaluate the rule set on both the incoming and outgoing interfaces). A third test compared only pf and IPFilter, using a single rule that created state in the state table with a fixed-state entry size. Pf reached an overloaded state much later than IPFilter. The experiment was repeated with a variable-state entry size and pf performed much better than IPFilter for a small number of states.

In general, rule set evaluation is expensive and benchmarks only reflect extreme cases, whereas in real life, other behavior should be observed. Furthermore, the benchmarks show that stateful filtering can actually improve performance due to cheap state-table lookups as compared to rule evaluation.

ENHANCING NFS CROSS-ADMINISTRATIVE DOMAIN ACCESS

Joseph Spadavecchia and Erez Zadok, Stony Brook University

The speaker presented modification to an NFS server that allows an improved NFS access between administrative domains. A problem lies in the fact that NFS assumes a shared UID/GID space, which makes it unsafe to export files outside the administrative domain of the server. Design goals were to leave protocol and existing clients unchanged, a minimum amount of server changes, flexibility, and increased performance.

To solve the problem, two techniques are utilized: "range-mapping" and "file-cloaking." Range-mapping maps IDs between client and server. The mapping is performed on a per-export basis and has to be manually set up in an export file. The mappings can be 1-1, N-N, or N-1. In file-cloaking, the server restricts file access based on UID/GID and special cloaking-mask bits. Here users can only access their own file permissions. The policy on whether or not the file is visible to others is set by the cloaking mask, which is logically ANDed with the

file's protection bits. File-cloaking only works, however, if the client doesn't hold cached copies of directory contents and file-attributes. Because of this the clients are forced to re-read directories by incrementing the mtime value of the directory each time it is listed.

To test the performance of the modified server, five different NFS configurations were evaluated. An unmodified NFS server was compared against one server with the modified code included but not used, one with only range-mapping enabled, one with only file-cloaking enabled, and one version with all modifications enabled. For each setup, different file system benchmarks were run. The results show only a small overhead when the modifications are used, generally an increase of below 5%. Another experiment tested the performance of the system with an increasing number of mapped or cloaked entries on a system. The results show that an increase from 10 to 1000 entries resulted in a maximum of about 14% cost in performance.

One member of the audience pointed out that if clients choose to ignore the changed mtimes from the server and thus still hold caches of the directory entries, the file-cloaking mechanism could be defeated. After a rather lengthy debate, the speaker had to concede that the model doesn't add any extra security. Another question was asked about scalability of the setup of range mapping. The speaker referred to application-level tools that could be developed for that purpose.

The software is available at *ftp://ftp.fsl.cs.sunysb.edu/pub/enf.*

ENGINEERING OPEN SOURCE SOFTWARE

Summarized by Teri Lampoudi

NINGAUI: A LINUX CLUSTER FOR BUSINESS
Andrew Hume, AT&T Labs – Research; Scott Daniels, Electronic Data Systems Corp.

Ningaui is a general purpose, highly available, resilient architecture built

from commodity software and hardware. Emphatically, however, Ningaui is not a Beowulf. Hume calls the cluster design the "Swiss canton model," in which there are a number of loosely affiliated independent nodes, with data replicated among them. Jobs are assigned by bidding and leases, and cluster services done as session-based servers are sited via generic job assignment. The emphasis is on keeping the architecture end-to-end, checking all work via checksums, and logging everything. The resilience and high availability required by their goal of 8-5 maintenance – vs. the typical 24-7 model where people get paged whenever the slightest thing goes wrong, regardless of the time of day – is achieved by job restartability. Finally, all computation is performed on local data, without the use of NFS or network attached storage.

Hume's message is a hopeful one: despite the many problems encountered – things like kernel and service limits, auto-installation problems, TCP storms, and the like – the existence of source code and the paranoid practice of logging and checksumming everything has helped. The final product performs reasonably well, and it appears that the resilient techniques employed do make a difference. One drawback, however, is that the software mentioned in the paper is not yet available for download.

CPCMS: A CONFIGURATION MANAGEMENT SYSTEM BASED ON CRYPTOGRAPHIC NAMES
Jonathan S. Shapiro and John Vanderburgh, Johns Hopkins University

This paper won the FREENIX Best Paper award.

The basic notion behind the project is the fact that everyone has a pet complaint about CVS, and yet it is currently the configuration manager in most widespread use. Shapiro has unleashed an alternative. Interestingly, he did not begin but ended with the usual host of reasons why CVS is bad. The talk instead began abruptly by characterizing the job

of a software configuration manager, continued by stating the namespaces which it must handle, and wrapped up with the challenges faced.

X meets Z: Verifying Correctness in the Presence of POSIX Threads

Bart Massey, Portland State University; Robert T. Bauer, Rational Software Corp.

Massey delivered a humorous talk on the insight gained from applying Z formal specification notation to system software design rather than the more informal analysis and design process normally used.

The story is told with respect to writing XCB, which replaces the Xlib protocol layer and is supposed to be thread friendly. But where threads are concerned, deadlock avoidance becomes a hard problem that cannot be solved in an ad-hoc manner. But full model checking is also too hard. In this case Massey resorted to Z specification to model the XCB lower layer, abstract away locking and data transfers, and locate fundamental issues. Essentially, the difficulties of searching the literature and locating information relevant to the problem at hand were overcome. As Massey put it, "the formal method saved the day."

FILE SYSTEMS
Summarized by Bosko Milekic

Planned Extensions to the Linux Ext2/Ext3 Filesystem

Theodore Y. Ts'o, IBM; Stephen Tweedie, Red Hat

The speaker presented improvements to the Linux Ext2 file system with the goal of allowing for various expansions while striving to maintain compatibility with older code. Improvements have been facilitated by a few extra superblock fields that were added to Ext2 not long before the Linux 2.0 kernel was released. The fields allow for file system features to be added without compromising the existing setup; this is done by providing

bits indicating the impact of the added features with respect to compatibility. Namely, a file system with the "incompat" bit marked is not allowed to be mounted. Similarly, a "read-only" marking would only allow the file system to be mounted read-only.

Directory indexing changes linear directory searches with a faster search using a fixed-depth tree and hashed keys. File system size can be dynamically increased, and the expanded inode, doubled from 128 to 256 bytes, allows for more extensions.

Other potential improvements were discussed as well, in particular, pre-allocation for contiguous files which allows for better performance in certain setups by pre-allocating contiguous blocks. Security-related modifications, extended attributes and ACLs, were mentioned. An implementation of these features already exists but has not yet been merged into the mainline Ext2/3 code.

Recent Filesystem Optimisations on FreeBSD

Ian Dowse, Corvil Networks; David Malone, CNRI, Dublin Institute of Technology

David Malone presented four important file system optimizations for FreeBSD OS: soft updates, dirpref, vmiodir, and dirhash. It turns out that certain combinations of the optimizations (beautifully illustrated in the paper) may yield performance improvements of anywhere between 2 and 10 orders of magnitude for real-world applications.

All four of the optimizations deal with file system metadata. Soft updates allow for asynchronous metadata updates. Dirpref changes the way directories are organized, attempting to place child directories closer to their parents, thereby increasing locality of reference and reducing disk-seek times. Vmiodir trades some extra memory in order to achieve better directory caching. Finally, dirhash, which was implemented by Ian Dowse, changes the way in which entries

are searched in a directory. Specifically, FFS uses a linear search to find an entry by name; dirhash builds a hashtable of directory entries on the fly. For a directory of size n, with a working set of m files, a search that in certain cases could have been $O(n*m)$ has been reduced, due to dirhash, to effectively $O(n + m)$.

Filesystem Performance and Scalability in Linux 2.4.17

Ray Bryant, SGI; Ruth Forester, IBM LTC; John Hawkes, SGI

This talk focused on performance evaluation of a number of file systems available and commonly deployed on Linux machines. Comparisons, under various configurations, of Ext2, Ext3, ReiserFS, XFS, and JFS were presented.

The benchmarks chosen for the data gathering were pgmeter, filemark, and AIM Benchmark Suite VII. Pgmeter measures the rate of data transfer of reads/writes of a file under a synthetic workload. Filemark is similar to postmark in that it is an operation-intensive benchmark, although filemark is threaded and offers various other features that postmark lacks. AIM VII measures performance for various file-system-related functionalities; it offers various metrics under an imposed workload, thus stressing the performance of the file system not only under I/O load, but also under significant CPU load.

Tests were run on three different setups: a small, a medium, and a large configuration. ReiserFS and Ext2 appear at the top of the pile for smaller and medium setups. Notably, XFS and JFS perform worse for smaller system configurations than the others, although XFS clearly appears to generate better numbers than JFS. It should be noted that XFS seems to scale well under a higher load. This was most evident in the large-system results, where XFS appears to offer the best overall results.

THINGS TO THINK ABOUT

Summarized by Bosko Milekic

SPEEDING UP KERNEL SCHEDULER BY REDUCING CACHE MISSES

Shuji Yamamura, Akira Hirai, Mitsuru Sato, Masao Yamamoto, Akira Naruse, Kouichi Kumon, Fujitsu Labs

This was an interesting talk pertaining to the effects of cache coloring for task structures in the Linux kernel scheduler (Linux kernel 2.4.x). The speaker first presented some interesting benchmark numbers for the Linux scheduler, showing that as the number of processes on the task queue was increased, the performance decreased. The authors used some really nifty hardware to measure the number of bus transactions throughout their tests and were thus able to reasonably quantify the impact that cache misses had in the Linux scheduler.

Their experiments led them to implement a cache coloring scheme for task structures, which were previously aligned on 8KB boundaries and, therefore, were being eventually mapped to the same cache lines. This unfortunate placement of task structures in memory induced a significant number of cache misses as the number of tasks grew in the scheduler.

The implemented solution consisted of aligning task structures to more evenly distribute cached entries across the L2 cache. The result was, inevitably, fewer cache misses in the scheduler. Some negative effects were observed in certain situations. These were primarily due to more cache slots being used by task structure data in the scheduler, thus forcing data previously cached there to be pushed out.

WORK-IN-PROGRESS REPORTS

Summarized by Brennan Reynolds

RESOURCE VIRTUALIZATION TECHNIQUES FOR WIDE-AREA OVERLAY NETWORKS

Kartik Gopalan, University Stony Brook

This work addressed the issue of provisioning a maximum number of virtual overlay networks (VON) with diverse quality of service (QoS) requirements on a single physical network. Each of the VONs is logically isolated from others to ensure the QoS requirements. Gopalan mentioned several critical research issues with this problem that are currently being investigated. Dealing with how to provision the network at various levels (link, route, or path) and then enforce the provisioning at run-time is one of the toughest challenges. Currently, Gopalan has developed several algorithms to handle admission control, end-to-end QoS, route selection, scheduling, and fault tolerance in the network.

For more information, visit *http://www.ecsl.cs.sunysb.edu/.*

VISUALIZING SOFTWARE INSTABILITY

Jennifer Bevan, University of California, Santa Cruz

Detection of instability in software has typically been an afterthought. The point in the development cycle when the software is reviewed for instability is usually after it is difficult and costly to go back and perform major modifications to the code base. Bevan has developed a technique to allow the visualization of unstable regions of code that can be used much earlier in the development cycle. Her technique creates a time series of dependent graphs that include clusters and lines called fault lines. From the graphs a developer is able to easily determine where the unstable sections of code are and proactively restructure them. She is currently working on a prototype implementation.

For more information, visit *http://www.cse.ucsc.ecu/~jbevan/evo_viz/.*

RELIABLE AND SCALABLE PEER-TO-PEER WEB DOCUMENT SHARING

Li Xiao, William and Mary College

The idea presented by Xiao would allow end users to share the content of the Web browser caches with neighboring Internet users. The rationale for this is that today's end users are increasingly connected to the Internet over high-speed links, and the browser caches are becoming large storage systems. Therefore if an individual accesses a page which does not exist in their local cache, Xiao is suggesting that they first query other end users for the content before trying to access the machine hosting the original. This strategy does have some serious problems associated with it that still need to be addressed, including ensuring the integrity of the content and protecting the identity and privacy of the end users.

SEREL: FAST BOOTING FOR UNIX

Leni Mayo, Fast Boot Software

Serel is a tool that generates a visual representation of a UNIX machine's boot-up sequence. It can be used to identify the critical path and can show if a particular service or process blocks for an extended period of time. This information could be used to determine where the largest performance gains could be realized by tuning the order of execution at boot-up. Serel creates a dependency graph expressed in XML during boot-up. This graph is then used to create the visual representation. Currently the tool only works on POSIX-compliant systems, but Mayo stated that he would be porting it to other platforms. Other extensions that were mentioned included having the metadata include the use of shared libraries and monitoring the suspend/resume sequence of portable machines.

For more information, visit *http://www.fastboot.org/.*

Berkeley DB XML

John Merrells, Sleepycat Software

Merrells gave a quick introduction and overview of the new XML library for Berkeley DB. The library specializes in storage and retrieval of XML content through a tool called XPath. The library allows for multiple containers per document and stores everything natively as XML. The user is also given a wide range of elements to create indices with, including edges, elements, text strings, or presence. The XPath tool consists of a query parser, generator, optimizer, and execution engine. To conclude his presentation, Merrell gave a live demonstration of the software.

For more information, visit *http://www.sleepycat.com/xml/.*

Cluster-on-Demand (COD)

Justin Moore, Duke University

Modern clusters are growing at a rapid rate. Many have pushed beyond the 5000-machine mark, and deploying them results in large expenses as well as management and provisioning issues. Furthermore, if the cluster is "rented" out to various users it is very time-consuming to configure it to a user's specs regardless of how long they need to use it. The COD work presented creates dynamic virtual clusters within a given physical cluster. The goal was to have a provisioning tool that would automatically select a chunk of available nodes and install the operating system and middleware specified by the customer in a short period of time. This would allow a greater use of resources, since multiple virtual clusters can exist at once. By using a virtual cluster, the size can be changed dynamically. Moore stated that they have created a working prototype and are currently testing and benchmarking its performance.

For more information, visit *http://www.cs.duke.edu/~justin/cod/.*

Catacomb

Elias Sinderson, University of California, Santa Cruz

Catacomb is a project to develop a database-backed DASL module for the Apache Web server. It was designed as a replacement for the WebDAV. The initial release of the module only contains support for a MySQL database but could be extended to others. Sinderson briefly touched on the performance of her module. It was comparable to the mod_dav Apache module for all query types but search. The presentation was concluded with remarks about adding support for the lock method and including ACL specifications in the future.

For more information, visit *http://ocean.cse.ucsc.edu/catacomb/.*

Self-Organizing Storage

Dan Ellard, Harvard University

Ellard's presentation introduced a storage system that tuned itself, based on the workload of the system, without requiring the intervention of the user. The intelligence was implemented as a virtual self-organizing disk that resides below any file system. The virtual disk observes the access patterns exhibited by the system and then attempts to predict what information will be accessed next. An experiment was done using an NFS trace at a large ISP on one of their email servers. Ellard's self-organizing storage system worked well, which he attributes to the fact that most of the files being requested were large email boxes. Areas of future work include exploration of the length and detail of the data collection stage, as well as the CPU impact of running the virtual disk layer.

Visual Debugging

John Costigan, Virginia Tech

Costigan feels that the state of current debugging facilities in the UNIX world is not as good as it should be. He proposes the addition of several elements to programs like ddd and gdb. The first addition is including separate heap and stack components. Another is to display only certain fields of a data structure and have the ability to zoom in and out if needed. Finally, the debugger should provide the ability to visually present complex data structures, including linked lists, trees, etc. He said that a beta version of a debugger with these abilities is currently available.

For more information, visit *http://infovis.cs.vt.edu/datastruct/.*

Improving Application Performance Through System-Call Composition

Amit Purohit, University of Stony Brook

Web servers perform a huge number of context switches and internal data copying during normal operation. These two elements can drastically limit the performance of an application regardless of the hardware platform it is run on. The Compound System Call (CoSy) framework is an attempt to reduce the performance penalty for context switches and data copies. It includes a user-level library and several kernel facilities that can be used via system calls. The library provides the programmer with a complete set of memory-management functions. Performance tests using the CoSy framework showed a large saving for context switches and data copies. The only area where the savings between conventional libraries and CoSy were negligible was for very large file copies.

For more information, visit *http://www.cs.sunysb.edu/~purohit/.*

Elastic Quotas

John Oscar, Columbia University

Oscar began by stating that most resources are flexible but that, to date, disks have not been. While approximately 80 percent of files are short-lived, disk quotas are hard limits imposed by administrators. The idea behind elastic quotas is to have a non-permanent storage area that each user can temporarily use. Oscar suggested the creation of a /ehome directory structure to be used in deploying elastic quotas. Currently, he

has implemented elastic quotas as a stackable file system. There were also several trends apparent in the data. Many users would ssh to their remote host (good) but then launch an email reader on the local machine which would connect via POP to the same host and send the password in clear text (bad). This situation can easily be remedied by tunneling protocols like POP through SSH, but it appeared that many people were not aware this could be done. While most of his comments on the use of the wireless network were negative, the list of passwords he had collected showed that people were indeed using strong passwords. His recommendations were to educate and encourage people to use protocols like IPSec, SSH, and SSL when conducting work over a wireless network, because you never know who else is listening.

SYSTRACE

Niels Provos, University of Michigan

How can one be sure the applications one is using actually do exactly what their developers said they do? Short answer: you can't. People today are using a large number of complex applications, which means it is impossible to check each application thoroughly for security vulnerabilities. There are tools out there that can help, though. Provos has developed a tool called systrace that allows a user to generate a policy of acceptable system calls a particular application can make. If the application attempts to make a call that is not defined in the policy, the user is notified and allowed to choose an action. Systrace includes an auto-policy generation mechanism which uses a training phase to record the actions of all programs the user executes. If the user chooses not to be bothered by applications breaking policy, systrace allows default enforcement actions to be set. Currently, systrace is implemented for FreeBSD and NetBSD, with a Linux version coming out shortly.

For more information, visit *http://www. citi.umich.edu/u/provos/systrace/.*

VERIFIABLE SECRET REDISTRIBUTION FOR SURVIVABLE STORAGE SYSTEMS

Ted Wong, Carnegie Mellon University

Wong presented a protocol that can be used to re-create a file distributed over a set of servers, even if one of the servers is damaged. In his scheme the user must choose how many shares the file is split into and the number of servers it will be stored across. The goal of this work is to provide persistent, secure storage of information, even if it comes under attack. Wong stated that his design of the protocol was complete and he is currently building a prototype implementation of it.

For more information, visit *http://www. cs.cmu.edu/~tmwong/research/.*

THE GURU IS IN SESSIONS

LINUX ON LAPTOP/PDA

Bdale Garbee, HP Linux Systems Operation

Summarized by Brennan Reynolds

This was a roundtable-like discussion with Garbee acting as moderator. He is currently in charge of the Debian distribution of Linux and is involved in porting Linux to platforms other than i386. He has successfully help port the kernel to the Alpha, Sparc, and ARM and is currently working on a version to run on VAX machines.

A discussion was held on which file system is best for battery life. The Riser and Ext3 file systems were discussed; people commented that when using Ext3 on their laptops the disc would never spin down and thus was always consuming a large amount of power. One suggestion was to use a RAM-based file system for any volume that requires a large number of writes and to only have the file system written to disk at infrequent intervals or when the machine is shut down or suspended.

A question was directed to Garbee about his knowledge of how the HP-Compaq merger would affect Linux. Garbee thought that the merger was good for Linux, both within the company and for the open source community as a whole. He stated that the new company would be the number one shipper of systems with Linux as the base operating system. He also fielded a question about the support of older HP servers and their ability to run Linux, saying that indeed Linux has been ported to them and he personally had several machines in his basement running it.

A question which sparked a large number of responses concerned problems people had with running Linux on mobile platforms. Most people actually did not have many problems at all. There were a few cases of a particular machine model not working, but there were only two widespread issues: docking stations and the new ACPI power management scheme. Docking stations still appear to be somewhat of a headache, but most people had developed ad-hoc solutions for getting them to work. The ACPI power management scheme developed by Intel does not appear to have a quick solution. Ted T'so, one of the head kernel developers, stated that there are fundamental architectural problems with the 2.4 series kernel that do not easily allow ACPI to be added. However, he also stated that ACPI is supported in the newest development kernel, 2.5, and will be included in 2.6.

The final topic was the possibility of purchasing a laptop without paying any charge/tax for Microsoft Windows. The consensus was that currently this is not possible. Even if the machine comes with Linux, or without any operating system, the vendors have contracts in place which require them to pay Microsoft for each machine they ship if that machine is capable of running a Microsoft operating system. The only way this will change is if the demand for

other operating systems increases to the point that vendors renegotiate their contracts with Microsoft, and no one saw this happening in the near future.

LARGE CLUSTERS
Andrew Hume, AT&T Labs – Research

Summarized by Teri Lampoudi

This was a session on clusters, big data, and resilient computing. Given that the audience was primarily interested in clusters and not necessarily big data or beating nodes to a pulp, the conversation revolved mainly around what it takes to make a heterogeneous cluster resilient. Hume also presented a FREENIX paper on his current cluster project, which explained in more detail much of what was abstractly claimed in the guru session.

Hume's use of the word "cluster" referred not to what I had assumed would be a Beowulf-type system but to a loosely coupled farm of machines in which concurrency was much less of an issue than it would be in a Beowulf. In fact, the architecture Hume described was designed to deal with large amounts of independent transaction processing, essentially the process of billing calls, which requires no interprocess message passing or anything of the sort a parallel scientific application might.

Where does the large data come in? Since the transactions in question consist of large numbers of flat files, a mechanism for getting the files onto nodes and off-loading the results is necessary. In this particular cluster, named Ningaui, this task is handled by the "replication manager," which generated a large amount of interest in the audience. All management functions in the cluster, as well as job allocation, constitute services that are to be bid for and leased out to nodes. Furthermore, failures are handled by simply repeating the failed task until it is completed properly if that is possible, and if it is not, then looking through detailed logs to discover

what went wrong. Logging and testing for the successful completion of each task are what make this model resilient. Every management operation is logged at some appropriate level of detail, generating 120MB/day, and every single file transfer is md5 checksummed. This was characterized by Hume as a "patient" style of management, where no one controls the cluster, scheduling behavior is emergent rather than stringently planned, and nodes are free to drop in and out of the cluster; a downside to this model is the increase in latency, offset by the fact that the cluster continues to function unattended outside of 8-to-5 support hours.

In response to questions about the projected scalability of such a scheme, Hume said the system would presumably scale from the present 60 nodes to about 100 without modifications, but that scaling higher would be a matter for further design. The guru session ended on a somewhat unrelated note regarding the problems of getting data off mainframes and onto Linux machines – issues of fixed- vs. variable-length encoding of data blocks resulting in useful information being stripped by FTP, and problems in converting COBOL copybooks to something useful on a C-based architecture. To reiterate Hume's claim, there are homemade solutions for some of these problems, and the reader should feel free to contact Mr. Hume for them.

WRITING PORTABLE APPLICATIONS
Nick Stoughton, MSB Consultants

Summarized by Josh Lothian

Nick Stoughton addressed a concern that developers are facing more and more: writing portable applications. He proposed that there is no such thing as a "portable application," only those that have been ported. Developers are still in search of the Holy Grail of applications programming: a write-once, compile-anywhere program. Languages such as Java are leading up to this, but virtual

machines still have to be ported, paths will still be different, etc. Web-based applications are another area that could be promising for portable applications.

Nick went on to talk about the future of portability. The recently released POSIX 2001 standards are expected to help the situation. The 2001 revision expands the original POSIX sepc into seven volumes. Even though POSIX 2001 is more specific than the previous release, Nick pointed out that even this standard allows for small differences where vendors may define their own behaviors. This can only hurt developers in the long run. Along these lines, it was pointed out that there may be room for automated tools that have the capability to check application code and determine the level of portability and standards compliance of the source.

NETWORK MANAGEMENT, SYSTEM PERFORMANCE TUNING
Jeff Allen, Tellme Networks Inc.

Summarized by Matt Selsky

Jeff Allen, author of Cricket, discussed network tuning. The basic idea of tuning is measure, twiddle, and measure again. Cricket can be used for large installations, but it's overkill for smaller installations, for which Mrtg is better suited. You don't need Cricket to do measurement. Doing something like a shell script that dumps data to Excel is good, but you need to get some sort of measurement. Cricket was not designed for billing; it was meant to help answer questions.

Useful tools and techniques covered included looking for the difference between machines in order to identify the causes for the differences; measuring but also thinking about what you're measuring and why; remembering to use strace and tcpdump to identify problems. Problems repeat themselves; performance tuning requires an intricate understanding of all the layers involved to solve the problem and simplify the automation. (Having a computer science

background helps but is not essential.) If you can reproduce the problem, you then should investigate each piece to find the bottleneck. If you can't reproduce the problem, then measure. You need to understand all the system interactions. Measurement can help determine whether things are actually slow or if the user is imagining it.

Troubleshooting begins with a hunch, but scientific processes are essential. You should be able to determine the event stream, or when each event occurs; having observation logs can help. Some hints provided include checking cron for periodic anomalies, slowing down /bin/rm to avoid I/O overload, and looking for unusual kernel CPU usage.

Also, try to use lightweight monitoring to reduce overhead. You don't need to monitor every resource on every system, but you should monitor those resources on those systems that are essential. Don't check things too often, since you can introduce overhead that way. Lots of information can be gathered from the /proc file system interface to the kernel.

BSD BOF

Host: Kirk McKusick, Author and Consultant

Summarized by Bosko Milekic

The BSD Birds of a Feather session at USENIX started off with five quick and informative talks and ended with an exciting discussion of licensing issues.

Christos Zoulas for the NetBSD Project went over the primary goals of the NetBSD Project (namely, portability, a clean architecture, and security) and then proceeded to briefly discuss some of the challenges that the project has encountered. The successes and areas for improvement of 1.6 were then examined. NetBSD has recently seen various improvements in its cross-build system, packaging system, and third-party tool support. For what concerns the kernel, a zero-copy TCP/UDP implementation has been integrated, and a new pmap

code for arm32 has been introduced, as have some other rather major changes to various subsystems. More information is available in Christos's slides, which are now available at *http://www.netbsd.org/ gallery/events/usenix2002/.*

Theo de Raadt of the OpenBSD Project presented a rundown of various new things that the OpenBSD and OpenSSH teams have been looking at. Theo's talk included an amusing description (and pictures!) of some of the events that transpired during OpenBSD's hack-a-thon, which occurred the week before USENIX '02. Finally, Theo mentioned that, following complications with the IPFilter license, the OpenBSD team had done a fairly extensive license audit.

Robert Watson of the FreeBSD Project brought up various examples of FreeBSD being used in the real world. He went on to describe a wide variety of changes that will surface with the upcoming FreeBSD release, 5.0. Notably, 5.0 will introduce a re-architecting of the kernel aimed at providing much more scalable support for SMP machines. 5.0 will also include an early implementation of KSE, FreeBSD's version of scheduler activations, large improvements to pccard, and significant framework bits from the TrustedBSD project.

Mike Karels from Wind River Systems presented an overview of how BSD/OS has been evolving following the Wind River Systems acquisition of BSDi's software assets. Ernie Prabhakar from Apple discussed Apple's OS X and its success on the desktop. He explained how OS X aims to bring BSD to the desktop while striving to maintain a strong and stable core, one that is heavily based on BSD technology.

The BoF finished with a discussion on licensing; specifically, some folks questioned the impact of Apple's licensing for code from their core OS (the Darwin Project) on the BSD community as a whole.

CLOSING SESSION

HOW FLIES FLY

Michael H. Dickinson, University of California, Berkeley

Summarized by J.D. Welch

In this lively and offbeat talk, Dickinson discussed his research into the mechanisms of flight in the fruit fly (*Drosophila melanogaster*) and related the autonomic behavior to that of a technological system. Insects are the most successful organisms on earth, due in no small part to their early adoption of flight. Flies travel through space on straight trajectories interrupted by saccades, jumpy turning motions somewhat analogous to the fast, jumpy movements of the human eye. Using a variety of monitoring techniques, including high-speed video in a "virtual reality flight arena," Dickinson and his colleagues have observed that flies respond to visual cues to decide when to saccade during flight. For example, more visual texture makes flies saccade earlier (i.e., further away from the arena wall), described by Dickinson as a "collision control algorithm."

Through a combination of sensory inputs, flies can make decisions about their flight path. Flies possess a visual "expansion detector," which, at a certain threshold, causes the animal to turn a certain direction. However, expansion in front of the fly sometimes causes it to land. How does the fly decide? Using the virtual reality device to replicate various visual scenarios, Dickinson observed that the flies fixate on vertical objects that move back and forth across the field. Expansion on the left side of the animal causes it to turn right, and vice versa, while expansion directly in front of the animal triggers the legs to flare out in preparation for landing.

If the eyes detect these changes, how are the responses implemented? The flies' wings have "power muscles," controlled by mechanical resonance (as opposed to the nervous system), which drive the

wings, combined with neurally activated "steering muscles," which change the configuration of the wing joints. Subtle variations in the timing of impulses correspond to changes in wing movement, controlled by sensors in the "wing-pit." A small, wing-like structure, the haltere, controls equilibrium by beating constantly.

Voluntary control is accomplished by the halteres, whose signals can interfere with the autonomic control of the stroke cycle. The halteres have "steering muscles" as well, and information derived from the visual system can turn off the haltere or trick it into a "virtual" problem requiring a response.

Dickinson has also studied the aerodynamics of insect flight, using a device called the "Robo Fly," an oversized mechanical insect wing suspended in a large tank of mineral oil. Interestingly, Dickinson observed that the average lift generated by the flies' wings is under its body weight; the flies use three mechanisms to overcome this, including rotating the wings (rotational lift).

Insects are extraordinarily robust creatures, and because Dickinson analyzed the problem in a systems-oriented way, these observations and analysis are immediately applicable to technology, the response system can be used as an efficient search algorithm for control systems in autonomous vehicles, for example.

THE AFS WORKSHOP
Summarized by Garry Zacheiss, MIT
Love Hornquist-Astrand of the Arla development team presented the Arla status report. Arla 0.35.8 has been released. Scheduled for release soon are improved support for Tru64 UNIX, MacOS X, and FreeBSD, improved volume handling, and implementation of more of the vos/pts subcommands. It was stressed that MacOS X is considered an important platform, and that a GUI configuration manager for the cache

manager, a GUI ACL manager for the finder, and a graphical login that obtains Kerberos tickets and AFS tokens at login time were all under development.

Future goals planned for the underlying AFS protocols include GSSAPI/SPNEGO support for Rx, performance improvements to Rx, an enhanced disconnected mode, and IPv6 support for Rx; an experimental patch is already available for the latter. Future Arla-specific goals include improved performance, partial file reading, and increased stability for several platforms. Work is also in progress for the RXGSS protocol extensions (integrating GSSAPI into Rx). A partial implementation exists, and work continues as developers find time.

Derrick Brashear of the OpenAFS development team presented the OpenAFS status report. OpenAFS was released immediately prior to the conference; OpenAFS 1.2.5 fixed a remotely exploitable denial-of-service attack in several OpenAFS platforms, most notably IRIX and AIX. Future work planned for OpenAFS includes better support for MacOS X, including working around Finder interaction issues. Better support for the BSDs is also planned; FreeBSD has a partially implemented client; NetBSD and OpenBSD have only server programs available right now. AIX 5, Tru64 5.1A, and MacOS X 10.2 (a.k.a. Jaguar) are all planned for the future. Other planned enhancements include nested groups in the ptserver (code donated by the University of Michigan awaits integration), disconnected AFS, and further work on a native Windows client. Derrick stated that the guiding principles of OpenAFS were to maintain compatibility with IBM AFS, support new platforms, ease the administrative burden, and add new functionality.

AFS performance was discussed. The openafs.org cell consists of two file servers, one in Stockholm, Sweden, and one in Pittsburgh, PA. AFS works reasonably well over trans-Atlantic links,

although OpenAFS clients don't determine which file server to talk to very effectively. Arla clients use RTTs to the server to determine the optimal file server to fetch replicated data from. Modifications to OpenAFS to support this behavior in the future are desired.

Jimmy Engelbrecht and Harald Barth of KTH discussed their AFSCrawler script, written to determine how many AFS cells and clients were in the world, what implementations/versions they were (Arla vs. IBM, AFS vs. OpenAFS), and how much data was in AFS. The script unfortunately triggered a bug in IBM AFS 3.6–derived code, causing some clients to panic while handling a specific RPC. This has since been fixed in OpenAFS 1.2.5 and the most recent IBM AFS patch level, and all AFS users are strongly encouraged to upgrade. No release of Arla is vulnerable to this particular denial-of-service attack. There was an extended discussion of the usefulness of this exploration. Many sites believed this was useful information and such scanning should continue in the future, but only on an opt-in basis.

Many sites face the problem of managing Kerberos/AFS credentials for batch-scheduled jobs. Specifically, most batch processing software needs to be modified to forward tickets as part of the batch submission process, renew tickets and tokens while the job is in the queue and for the lifetime of the job, and properly destroy credentials when the job completes. Ken Hornstein of NRL was able to pay a commercial vendor to support Kerberos 4/5 credential management in their product, although they did not implement AFS token management. MIT has implemented some of the desired functionality in OpenPBS, and might be able to make it available to other interested sites.

Tools to simplify AFS administration were discussed, including:

- AFS Balancer. A tool written by CMU to automate the process of balancing disk usage across all

servers in a cell. Available from *ftp://ftp.andrew.cmu.edu/pub/ AFS-Tools/balance-1.1b.tar.gz.*

- Themis. Themis is KTH's enhanced version of the AFS tool "package," for updating files on local disk from a central AFS image. KTH's enhancements include allowing the deletion of files, simplifying the process of adding a file, and allowing the merging of multiple rule sets for determining which files are updated. Themis is available from the Arla CVS repository.

Stanford was presented as an example of a large AFS site. Stanford's AFS usage consists of approximately 1.4TB of data in AFS, in the form of approximately 100,000 volumes. 3.3TB of storage is available in their primary cell, ir.stanford.edu. Their file servers consist entirely of Solaris machines running a combination of Transarc 3.6 patch-level 3 and OpenAFS 1.2.x, while their database servers run OpenAFS 1.2.2. Their cell consists of 25 file servers, using a combination of EMC and Sun StorEdge hardware. Stanford continues to use the kaserver for their authentication infrastructure, with future plans to migrate entirely to an MIT Kerberos 5 KDC.

Stanford has approximately 3400 clients on their campus, not including SLAC (Stanford Linear Accelerator); approximately 2100 AFS clients from outside Stanford contact their cell every month. Their supported clients are almost entirely IBM AFS 3.6 clients, although they plan to release OpenAFS clients soon. Stanford currently supports only UNIX clients. There is some on-campus presence of Windows clients, but Stanford has never publicly released or supported it. They do intend to release and support the MacOS X client in the near future.

All Stanford students, faculty, and staff are assigned AFS home directories with a default quota of 50MB, for a total of approximately 550GB of user home directories. Other significant uses of AFS

storage are data volumes for workstation software (400GB) and volumes for course Web sites and assignments (100GB).

AFS usage at Intel was also presented. Intel has been an AFS site since 1994. They had bad experiences with the IBM 3.5 Linux client; their experience with OpenAFS on Linux 2.4.x kernels has been much better. They use and are satisfied with the OpenAFS IA64 Linux port. Intel has hundreds of OpenAFS 1.2.3 and 1.2.4 clients in many production cells, accessing data stored on IBM AFS file servers. They have not encountered any interoperability issues. Intel has some concerns about OpenAFS; they would like to purchase commercial support for OpenAFS and to see OpenAFS support for HP-UX on both PA-RISC and Itanium hardware. HP-UX support is currently unavailable due to a specific HP-UX header file being unavailable from HP; this may be available soon. Intel has not yet committed to migrating their file servers to OpenAFS and are unsure if they will do so without commercial support.

Backups are a traditional topic of discussion at AFS workshops, and this time was no exception. Many users complain that the traditional AFS backup tools ("backup" and "butc") are complex and difficult to automate, requiring many home-grown scripts and much user intervention for error recovery. An additional complaint was that the traditional AFS tools do not support file- or directory-level backups and restores; data must be backed up and restored at the volume level.

Mitch Collinsworth of Cornell presented work to make AMANDA, the free backup software from the University of Maryland, suitable for AFS backups. Using AMANDA for AFS backups allows one to share AFS backup tapes with non-AFS backups, easily run multiple backups in parallel, automate error recovery, and provide a robust degraded mode that prevents tape errors from

stopping backups altogether. Their implementation allows for full volume restores as well as individual directory and file restores. They have finished coding this work and are in the process of testing and documenting it.

Peter Honeyman of CITI at the University of Michigan spoke about work he has proposed to replace Rx with RPCSEC GSS in OpenAFS; this would allow AFS to use a TCP-based transport mechanism, rather than the UDP-based Rx, and possibly gain better congestion control, dynamic adaptation, and fragmentation avoidance as a result. RPCSEC GSS uses the GSSAPI to authenticate SUN ONC RPC. RPCSEC GSS is transport agnostic, provides strong security, is a developing Internet standard, and has multiple open source implementations. Backward compatibility with existing AFS servers and clients is an important goal of this project.