showed 100% of the disk idle times to fall beneath one second, not nearly long enough to enter a disk's power-saving state without incurring a net power efficiency loss owing to energy required for transition.

The presentation detailed the design and implementation of Bursty, a mechanism providing highly speculative prefetching, a kernel interface to hint disk access patterns, and a daemon both to monitor and manage the system. Applications may hint improperly, or lack hints entirely, so the monitor must both generate and judge extant hints. The prefetching is also self-aware—the success rate of the algorithm is constantly measured to determine whether further adjustments are required. Additionally, per-application idle times are irrelevant if they are not in phase between applications; hence, the daemon also attempts to organize these patterns to allow for consistent disk avoidance between all applications. Once the predicted idle period is estimated to beyond the intersection of regular drive use and idle use combined with transition expenditure, the drive is powered down into an appropriate low-power mode. A variety of tests with different applications using a variety of workloads and disk access patterns (and memory configurations—Bursty is hungry!) found 60–80% energy savings, with negligible losses in efficiency.

Time-Based Fairness Improves Performance in Multi-Rate WLANs

Godfrey Tan and John Guttag, MIT

Modern Wireless networks theoretically maintain decent throughput when congested, though in practice the common utilization of rate diversity as an automatic signal strengthening scheme causes standard throughput-based fairness schemes to result in unexpectedly poor performance. This paper presents an overview of the problem, design and implementation of a solution—termed "time-based fairness"—and experimental verification.

802.11b was used as the test case. Though traditionally known to sport 11Mbps, the standard also defines three other rates: 1, 2, and 5.5. Vendors use these speeds when packet transmission failure becomes a problem, eventually bumping the speed to the slowest rate, which features the highest signal resilience. Current channel proportioning and access point downlink scheduling techniques result in throughput-based fairness, meaning a slower rate receives a larger portion of channel time, ostensibly aiding the feeble companion but causing the hare to be tied to the turtle.

Time-based fairness apportions channel use equally by time, resulting in much higher possible throughput. The average time for network tasks to complete is also reduced, obviously a benefit to many mobile users and definitely to anyone who would rather have things to do while a laggy transmission completes. The implementation is flexible enough to function properly on extant access points and does not need extensive modification on clients; adoption is easy and backwards-compatible.

EmStar: A Software Environment for Developing and Deploying Wireless Sensor Networks

Lewis Girod, Jeremy Elson, Alberto Cerpa, Thanos Stathopoulos, Nithya Ramanathan, and Deborah Estrin, UCLA

The burgeoning study, experimentation, and deployment of wireless sensor network applications is generating a need for full-fledged development suites. EmStar provides just that for 32-bit embedded MicroServer platforms: tools and libraries providing simulation, visualization, and emulation. Other functionality aids in development and testing of IPC and network communication.

Due to time constraints, a large portion of the talk focused on FUSD (Framework for User-Space Devices), which is a kernel module proxy to device file events. FUSD, though new, is already used by numerous applications to simplify communication with device nodes. It is essentially a micro-kernel extension to Linux. At the moment, performance is sufficient but unsatisfactory; read throughput, for instance, can be 3 to 17 times slower than analogous read performance without the FUSD proxy overhead.

EmSim and EmCee are simulation tools; these in turn are modular to allow for easy extension and minimized footprint. EmRun starts up, maintains, and shuts down an EmStar system according to a policy in a configuration file; it features process respawn, in-memory logging, fast startup, and graceful shutdown. All components (more than are listed here) are written with modularity in mind, and code is heavily reused. It has already proved useful in numerous projects at the CENS labs working with a variegated set of hardware.

SECURITY SIG

Summarized by Ming Chow

Panel: The Politicization of Security

Moderator: Avi Rubin, Johns Hopkins University

Panelists: Ed Felten, Princeton University; Jeff Grove, ACM; Gary McGraw, Cigital

The common theme of this panel was how politicized security, especially that relating to technology, has become. Professor Avi Rubin spoke of his experiences working on the issue of electronic voting (eVoting). He spoke about dealing with policy issues, and about how eVoting has become a partisan, politically charged issue and, as

such, is targeted for abuse. An example is that companies producing eVoting technologies and equipment have strong political ties. The goal from each political party is to "not have the other guy win." Professor Rubin has been on major news sources (e.g., CNN) speaking about technical issues of eVoting, and has received numerous telephone calls from both Democrats and Republicans. Professor Rubin recounted being called to testify in front of Congress about eVoting, and recalled the amount of fighting and bickering on both political sides dealing with the issue. He summed up the current state of politics by saying that "partisanship has never been worse."

Gary McGraw spoke of the long history of politicization of scientific research and development and the degree to which current scientific research and development are influenced by politics (like Galileo and Darwin centuries ago). He stated that security and terrorism are sensitive subjects, and that "we should understand the problem, having worked in an asymmetric situation for years in computer security." McGraw also said that too often "individual rights can be trumped in the name of security" (e.g., DMCA and the Patriot Act).

Jeff Grove has worked with the government on Capitol Hill, and expressed his dissatisfaction on the number of bad laws being implemented, including the DMCA, and the regulation of P2P networks. Grove outlined how the Senate can address and jump on issues and make dumb laws. The problem persists because of bad conclusions, bad assumptions, and lack of basic understanding about technologies. In addition, there is a small handful of powerful players who are effective in influencing the government to create laws fitting their agendas. Bad laws expose developers to liabilities, even when there's no infringement, and provide civil enforcement by encouraging legal actions by the entertainment industry.

Professor Ed Felten spoke about the Digital Millennium Copyright Act (DMCA) and his work, which made national headlines several years ago. Professor Felten stated that the DMCA was created by negotiations in which computer scientists were not involved. His work with advisee John Halderman was discussed—the weak DRM technology created by SunnComm could be bypassed on Windows computers by holding down the shift key. The government was cracking down on Professor Felten's research and he was threatened by the RIAA under the DMCA. Professor Felten settled with both Princeton University and the government by creating educational packets for the government on security research. Professor Felten recalled testifying in Congress about a bill to limit developing tools on decoding technologies, and summarized the atmosphere in one word: "theater." Finally, he gave out his Web site: http://www.freedom-to-tinker.com.

The theme from all of the panel speakers was clear: "We (the computing and scientific communities) need to step up to the plate and educate people on technological issues." The goal can be accomplished by being more involved, by being partisan, and by talking to anyone who is curious. Openness and debate are encouraged and are healthy. It is critical to tell the truth and to convince people about what's really going on. Gary McGraw also said that attacking systems is a necessary part of security and that outlawing attacks makes little sense. Finally, media and politics are great investments: the "Slashdot effect" helps ridicule bad laws, and working even with your local government is a 10–15-year investment.

FREENIX OPENING REMARKS AND AWARDS

Summarized by Martin Michlmayr

Bart Massey, Portland State University; Keith Packard, Hewlett-Packard Cambridge Research Lab

Bart Massey and Keith Packard opened the FREENIX track, a forum devoted to free and open source software, by giving a brief summary of papers that were submitted this year. Out of 61 papers submitted, 15 were accepted. The organizers were happy to see that among the accepted papers, seven were from students, and seven were non-US papers. They said that the quality of all submitted papers was very high and that the review process was more formal than in the last few years, adding three external reviewers to the program committee. They also thanked DoCoMo for sponsoring student travel for the conference.

In this opening speech, two awards to papers in the FREENIX track were given. The Best Paper award went to "Wayback: A User-level Versioning File System for Linux," and the Best Student Paper was "Design and Implementation of Netdude, a Framework for Packet Trace Manipulation."

There will be another FREENIX track at USENIX '05 in Anaheim, California. Since future USENIX conferences will take place around April, the deadline for FREENIX submissions is October 22, 2004, rather than in December. More information on the next FREENIX track and Call for Papers can be found at http://www.usenix.org/events/usenix05/cfp/freenix.html.

FREENIX INVITED TALK

Summarized by Martin Michlmayr

The Technical Changes in Qt Version 4

Matthias Ettrich, Trolltech Linux/Open Source

Matthias Ettrich, founder of the KDE project and a main developer on Qt, gave an overview of the next generation of Qt, a cross-platform C++ GUI toolkit. Qt supports X11, Microsoft Windows, Mac OS X, and embedded Linux, and offers native look and feel on each of these platforms. Qt provides single-source compatibility: one source code compiles on all target platforms. While Qt mainly offered GUI functions in the past, it is much more than a GUI library these days: It also supports I/O, printing, networking, SQL, process handling, and threading. One aim of Qt is to provide an excellent programming experience.

Qt introduced the signals-and-slot concept in order to allow different GUI components to communicate. You can connect any signal to any number of slots in any module, and communication is done at run time. The sender and receiver don't need to know each other. In version 4, connections can be either synchronous or asynchronous ("equal connections"); this will allow thread communication. Arthur is Qt's paint subsystem, and version 4 will offer several new features: linear gradient brushes, alpha-blended drawing, anti-aliased lines, painter paths, and an OpenGL backend.

Interview is a model/view framework for tree views, lists views, and tables. In the Model-View-Controller (MVC) paradigm, all these components are separated from each other. The model contains data, the view renders data, and the controller transforms interaction with the view into actions to be performed on the model. Qt 4 will introduce semi-transparent windows and flicker-free painting, and it will also implicitly provide double-buffering for all built-in and custom widgets: this is transparent, and no code needs to be rewritten. In addition, it will allow large windows (even modern window systems limit a widget's coordinate system to 16 bit, but Qt 4 won't have this limitation), and there will be improvements in size and performance. Qt 3 was originally designed for desktop computers (with fast CPUs with FPUs, lots of RAM and disk space). On the other hand, Qtopia was designed for embedded systems. Qt 4 aims at merging the benefits of both product lines into one.

In summary, Qt 4 will provide a number of new features that will offer new possibilities for cross-platform development. Ettrich hopes that a first technology preview will be made really soon, with another one following in Q3 2004. A beta of Qt 4 should be released in Q4 2004, with the final version following in Q1 2005.

SECURITY SIG

Summarized by Ming Chow

Panel: Wireless Devices and Consumer Privacy

Organizers: Ari Juels, RSA Laboratories; Richard Smith, Consultant

Panelists: Markus Jakobsson, RSA Laboratories; Frank Schroth, uLocate; Matthew Gray, Newbury Networks

The panel talked about wireless technologies, including GPS and RFID, and privacy issues concerning them. Frank Schroth discussed uLocate's wireless technology, which enables small business users to view the location of phones in their account, including maps and routes. uLocate's technology is based on GPS on the Nextel Network. The interface on cellular phones is a Java-based application that transmits data to server via UDP. Permissioned users can view information on their phone or via Web. The benefits of the technologies to small business include convenience, efficiency, and safety. The security of the uLocate service consists of two layers: a carrier level and an application level. Mr. Schroth also discussed privacy concerns about the technology, namely, addressable IP addresses, privacy, and leadership and ownership of risks.

Matthew Gray of Newbury Networks discussed his concerns about location-tracking technologies. The goal at Newbury Networks is to see what people are accessing without interfering with larger networks (e.g., Starbucks) and to eliminate such false positives to enhance security and privacy. Mr. Gray noted that security and privacy are in opposition to each other. He said that consumers and regulators must understand the risks of location-tracking technologies.

Marcus Jakobsson of RSA Laboratories listed three ways in which location privacy can be violated: active attacks (keep asking a device), passive attacks (listen to communication from other devices), and remote attacks (infer location from public information). He said that legislation for location privacy is necessary and meaningful. However, such law will be difficult to enforce because detecting abuse by institutions is hard, and it is even harder to detect abuse by individuals. He noted that countermeasures have been proposed but not deployed. In conclusion, Mr. Jakobsson listed several things that must be done immediately: the threats of location privacy must be studied and understood, legislation must be enacted, and countermeasures must be implemented.

Finally, Ari Juels and Richard Smith discussed radio frequency identification (RFID) tags and privacy concerns about the technology. Mr. Juels presented a brief tutorial of the RFID technology: an RFID tag

uses a chip (IC) antenna slightly larger than a quarter. Currently, many people have tools and gadgets that have RFID tags, such as E-ZPass, Mobil SpeedPass, and physical-access cards. RFID tags are seen as next-generation barcodes; Mr. Juels listed the benefits of RFID tags over barcodes (they're fast, efficient, mobile, can uniquely specify objects, and require little computational power). What this means is that the world will consist of billions of $0.05 computers. The major privacy problem concerning RFID technology is that it can be used to profile a person incredibly easily and quickly, providing detailed information, for example, on artificial body parts and other details of a person. Mr. Juels noted that approximately 42% of Goggle hits on a search for RFID contain the word "privacy." The solution to the privacy problem is to kill RFID tags. However, RFID tags are too useful. Mr. Juels concluded his talk by saying that there is serious danger to privacy if the technology is deployed naively, but the danger can be mitigated to strike a technical balance with society.

At the end of the talk, the panel discussed what must be done now to mitigate privacy concerns. One question to the panel was whether policy legislation hurts or helps technical development. A member of the panel suggested that a policy of saving data for 90 days would be sufficient. There was also a discussion about disclosure of information to consumers. Mr. Schroth responded that it has been startling to him how people do not care about disclosing information about themselves: people are willing to give lots of information to companies, including passwords.

FREENIX SESSION: SERVER

Summarized by Matus Telgarsky

Migrating an MVS Mainframe Application to a PC

Glenn S. Fowler, Andrew G. Hume, David G. Korn, and Kiem-Phong Vo, AT&T Labs

Rotting at the hearts of many old institutions' organizational frameworks are mainframes and their respective applications. Though the software may be relatively dependable, operational costs are prohibitive (the task emulated within this paper is estimated at $20,000 per month just for mainframe use), and the code consists of thousands, even millions of lines of ancient COBOL and JCL, on a system without a hierarchical file system. Emulating the process is a feasible and cost-effective alternative.

The MVS was to handle a mammoth of data, so a variety of tools were written to efficiently compress it prior to transmission. The OpenCOBOL compiler was extended to handle a few language extensions and different character sets, parse compressed data directly, and also receive a few performance enhancements. An extended sort program was built to enable MVS features, and a flexible JCL interpreter was built with handy features such as ksh script generation. An unsophisticated scheduler was developed to emulate MVS handling of processes.

David Korn took a moment to quip that a 25-year-old tip indicated that sort is optimally performed on a UNIX machine by transferring it to tape, performing it on a mainframe, and transporting it back—yet today the situation is reversed. Two 2.8GHz Pentium 4 machines were used to emulate the mainframes, at under $4,000 total. The 60-hour MVS task took 19 hours on the shiny new silicon. Data transmission ballooned surprisingly to nearly 24 hours due to tapes acting—predictably—unpredictably fussily.

C-JDBC: Flexible Database Clustering Middleware

Emmanuel Cecchet, INRIA; Julie Marguerite, ObjectWeb; Willy Zwaenepoel, EPFL

The general trend in modern high-power computing is toward clusters of commodity machines, achieving a significantly superior price-power ratio over more traditional and expensive many-CPU SMP machines. Though many tiers of server applications have extended to utilize this trend, in general RDBMS installations have lagged behind. Limited support has come from Oracle and IBM, but open source databases either relied on simplistic master-slave replication services or other similar compromises.

Clustered JDBC (C-JDBC) is an open source database middleware which abstracts pools of databases into a virtual database, complete with load balancing, query caching, logging, checkpointing, scheduling, authentication, and other features. JDBC is used to connect to virtually any database (as they basically all provide JDBC drivers), and allows for seamless integration of heterogeneous database farms into single resources. Performance has also been considered deeply: For most workloads, increasing nodes results in linear benchmark improvements, meaning superbly minor overhead and excellent scalability.

Fault tolerance and redundancy are not only accounted for with a flexible load balancer, but C-JDBC controllers themselves can be stacked horizontally to virtualize the same databases and seamlessly provide redundancy. Arbitrary trees may be constructed by attaching C-JDBC controllers as client databases to other C-JDBC controllers. Though only 10 months have passed since its initial beta release, C-JDBC has already been downloaded more than 15,000 times.