# conference reports

## THANKS TO OUR SUMMARIZERS

LISA ’09: 23rd Large Installation System Administration Conference . . . . . . . . . . . . . .100

Mark Burgess
Leah Cardaci
Rik Farrow
John Hewson
Rowan Littell
David Plonka
Shaya Potter
Andrew Seely
Josh Simon
Chuan Yue

## SPECIAL AWARDS AT LISA ’09

David Blank-Edelman received the 2009 SAGE Outstanding Achievement Award for his many contributions to the sysadmin community over the past quarter of a century. See http://www.sage.org/about/outstanding.html.

Luke S. Crawford won the Chuck Yerkes Award in recognition of outstanding individual contributions in online forums over the past year. See http://lopsa.org/node/1858.

## LISA ’09: 23rd Large Installation System Administration Conference

*Sponsored by USENIX and SAGE in cooperation with LOPSA and SNIA*

*Baltimore, MD*
*November 1–6, 2009*

Videos, MP3s, and PDFs of presentations are available online at http://www.usenix.org/lisa09/.

### KEYNOTE ADDRESS

- *Ahead in the Cloud: The Power of Infrastructure as a Service*

  *Werner Vogels, CTO, Amazon.com*

  *Summarized by David Plonka (plonka@cs.wisc.edu)*

Werner Vogels, who sometimes describes himself as the “system administrator of a large book shop,” gave this year’s LISA keynote address, an overview of Amazon’s Elastic Cloud Computing (EC2) as the preeminent example of infrastructure provided as a service.

First, Werner introduced an exemplary customer of Amazon’s services—“animoto” (http://animoto.com/), a company that developed an engine that automatically combines images, music, and video clips that you fashion into a polished production, something like a movie trailer. He said they are a New York-based enterprise that owns no servers. Instead, their product uses a four-step process (upload, analysis, rendering, and distribution) that employs Amazon cloud services: Amazon SQS to manage a queue of jobs, and Amazon EC2 to create and S3 to store content. When animoto launched a Facebook application for the service, they were able to immediately employ thousands of servers via Amazon EC2 servers to handle the influx of new users. He made the point that such a venture is revolutionary: you couldn’t secure start-up funding for 5000 or more servers simply to launch a free Facebook application. Thus, Werner describes this as a “democratization of business”: essentially, that the little guy can get the resources to launch something great.

Werner proceeded to describe how, in general, Amazon provides infrastructure as a service and that this is a significant foundation of Amazon’s structure, such that Amazon’s Web business, is a customer of this service too. Turning his comments to system administration, specifically, Werner said it is a myth that cloud computing puts sysadmins out of work. Indeed, sysadmins should have cloud computing in their portfolio, so that you can shift things there if and when you want. Since running an application on the cloud entails automation, cloud computing allows you to introduce more automation.

Werner prefers to use the term “infrastructure as a service” to “cloud computing” to disambiguate the concept from many things that are being called cloud computing, some of which go horribly wrong. The Gartner group’s definition of cloud computing is computing that is massively scalable, a service, over the Internet, and with multiple external customers. Werner added that infrastructure as a service needs two more characteristics: (1) resources available and releasable on demand (saving money and management troubles) and (2) the ability to pay as you go (so that there is not a massive expense when you are not using it).

Next, Werner described the implementation and evolution of each component of Amazon's infrastructure as a service to meet both Amazon's internal needs and those of its customers. The general infrastructure for Elastic Cloud Computing (EC2) runs on equipment managed using the Xen hypervisor. The storage offerings include Simple Storage Service (S3) for storage, SimpleDB for single index table DB operations, Elastic Block Storage (EBS) for operations requiring raw device access, and Relational Database Service (RDS) for custom MySQL instances that the customer can tune. Other offerings such as the Amazon Virtual Private Cloud (VPC) allow a customer to create a sort of "walled garden" using their own network addressing. Lastly, he mentioned that they've introduced "reserved instances" for customers wanting 24/7 use.

In closing, Werner outlined current usage trends for such cloud services, including load testing, collaborations, disaster recovery testing, large-scale analysis, marketing campaigns, and High Performance Computing (HPC). Amazon's cloud customers include Forbes for its stock quote streaming application, periodic streaming video for Indianapolis Motor Speedway events, Playfish for social network scaling, eHarmony to perform map/reduce matching daily, Netflix for video on demand, and Intuit for load testing to prepare for TurboTax downloads.

The session closed with questions from the audience. Rik Farrow asked, "What if MySQL goes away, given that it is the basis of Amazon's RDS service?" Werner replied that it is out of their control, but that Larry Ellison, CEO of Oracle Corp., the owner of MySQL, has committed to maintaining MySQL as a separate product. Brent Chapman asked about provisioning networks, and Werner replied that they don't currently expose this, even to allow customers to assign IP addresses, but there are a couple of tools for load balancing. Furthermore, each virtual machine runs its own firewall, and the customer doesn't have configurable control over all the network pieces. How does sensitive data (e.g., health care information) get handled in the cloud? Such security is an end-to-end issue and, generally, encryption is required to guarantee that data in the cloud can never be read by anyone else; there are strategies that involve using non-sensitive metadata as indexes to locate encrypted data. He also noted that some customers have achieved HIPAA and other regulatory compliance levels and that an Amazon white paper is available about HIPAA certification on cloud computing.

Contact information for Werner Vogels can be found at http://mynameise.com/werner.

## THE HUMAN SIDE OF SYSADMIN

*Summarized by Shaya Potter (spotter@cs.columbia.edu)*

- ***Pushing Boulders Uphill: The Difficulty of Network Intrusion Recovery***

  *Michael E. Locasto, George Mason University; Matthew Burnside, Columbia University; Darrell Bethea, University of North Carolina at Chapel Hill*

Michael Locasto focused on experience gained and lessons learned from analyzing a large-scale intrusion event that occurred at a large research university. This problem of network intrusion recovery "is a particularly thorny exercise in researching, designing, and creating usable security mechanisms," because intrusion recovery is an art, not a science. Each attack is different and each place an attack is successful is different. This means that each attack has to be handled differently. One thing that can help is to know more cases where intrusion occurs on a large scale to learn from those examples. However, there's a large stigma to admitting that a breach has occurred, meaning there are fewer stories to learn from. Michael hopes his paper can contribute to the lore and add to the publicized experience.

While Michael's paper included three intrusion recovery stories, his talks focused on the one that occurred in December 2007. Early in the year, a research lab at the university received new machines with high performance NVIDIA cards. As no official Linux drivers existed for them at the time, the lab used unofficial drivers. On December 6, the machines started crashing regularly. Installing the new official drivers did not help. They then pushed all the updates to the new machines via rdist, including the latest kernel, but this did not help either. It was assumed that there was a problem with the machines. However, on December 13, the rdist machines started crashing as well, and when they attempted to recompile its kernel, mkdir returned an error for directories containing only numbers. This led them to believe the machines had a rootkit, which they were able to confirm.

So what did they learn from this? First, they only discovered the intrusion by accident. Until a conflict cropped up and machines started crashing, they had no idea the machines had been exploited. Second, computer forensics is difficult to achieve. There is a tension between disabling an exploited host and keeping it up, considering, on the one hand, the impact on the reputation of those who host the machine as well as the risk to confidentiality, integrity, and privacy of the data on it versus the desire, on the other hand, to observe what is going on to learn more about it, as well as to keep available the service provided by the machine. Third, institutional memory is weak, meaning that goals can be forgotten or misunderstood, causing security gaps after other repairs are performed. Fourth, in many cases, informal preferences seemed to have a large impact on how to proceed in handling the intrusion. Finally, and most importantly, improvisation was common, as good tools did not exist to handle the challenges faced. This includes engineering challenges of repairing a network as well as management and usability challenges of dealing with people.

- ***Two-Person Control Administration: Preventing Administration Faults through Duplication***
  *Shaya Potter, Steven M. Bellovin, and Jason Nieh, Columbia University*

Shaya Potter focused on how one can apply the two-person control model to system administration. This is to help with administration faults that occur due to the fact that machines are complicated and complicated systems are prone to faults. Shaya focused his talk on two types of administrative faults that can occur, accidental and malicious. Accidental faults can be viewed as misconfigurations, while malicious faults are the result of an administrator leveraging privilege for malicious means.

To help prevent these type of faults from entering the system, Shaya proposed that the two-person control model should be applied to system administration, as this is known to be a good way to prevent faults in real systems, such as with nuclear weapons, bank checks, and pilots, all of which require two people to perform the same actions or just be available to observe the actions. To implement this, the authors created the I See Everything Twice (ISE-T) system to provide a general means of branching two computing environments and then comparing how they differ after they execute independently. This can be applied to system administration by branching the environments, allowing two system administrators to work independently and then compare their results for equivalence. If the environments are equivalent, the changes can be applied to the base system. Shaya noted that this can be an expensive solution, too expensive in many scenarios. However, he believes that the system can be used to improve other areas of system administration with little added cost, including the training of junior system administrators, providing an audit trail as well as combining it with configuration management.

ISE-T is implemented as a combination of three components. First, it includes an isolation mechanism that can be based on operating system containers, such as Solaris Zones, Linux Containers, or BSD's jails, as well as virtual machines such as VMware. These provide isolation for the cloned environment. Second, it includes a file system that both can branch itself quickly and can isolate the changes that occur after branch occurs, so that one can easily determine what has changed. In order to satisfy these requirements, they leveraged unioning file systems in order to enable quick branching, as well as isolating the changes to a new branch of the union. Finally, ISE-T includes a system service that compares the two environments for equivalence. This is difficult, because if one is limited to exact binaries, one will miss files that are semantically the same but differ in small ways, such as white space in scripts. However, for their prototype, they basically stuck to requiring that the files having the same binary content.

The authors created a prototype which they used to test the feasibility of capturing changes and comparing for equivalence over a number of administrative tasks, including installing software, upgrading systems and making configuration changes, as well as having people create back doors. In general, the configurations were equivalent, except in places where they expected differences to occur, such as with the creation of encryption keys and back doors. In the few cases where other differences were detected, ISE-T was able to clearly show differences between the two administrations.

Shaya was asked about how this compares to using a source code control system for configuration management systems. He answered that this formalizes and enforces behaviors that might not exist with a source code control system.

- ***The Water Fountain vs. the Fire Hose: An Examination and Comparison of Two Large Enterprise Mail Service Migrations***
  *Craig Stacey, Max Trefonides, Tim Kendall, and Brian Finley, Argonne National Laboratory*

Craig Stacey spoke about his worst week as a system administrator. Argonne used to have a simple mail infrastructure. Mail came in and was handed off to a mail server using a system that lasted from 1998 to 2008, running an old version of Cyrus IMAP. The system worked very well until 2006, when they thought about replacing it, but other things took time away. Later on they piloted Zimbra to provide calendaring to the lab. Then they decided that as Zimbra can provide mail as well, they should consolidate everything onto it. They felt the deadline for moving was far away and wouldn't be hard to hit, as IMAP is simple: it's just mailbox data.

Plan A was to use imapsync to move data between the machines. However, the IMAP server was old. Its libraries were too old, and it was so overtaxed that it couldn't handle the load and they feared it was going to fall over. So they set up a separate machine to pull everything off. However, it turned out that they could only run two syncs in parallel without affecting service, which was very slow and wouldn't finish in time.

Plan B was to use rsync to sync the machines and then scripts could import the changes into Zimbra, and imapsync could then sync the mailbox flags between the two machines. However, the implementation did not hold up. The server they were trying to sync the mailboxes to was too slow, due to NFS problems. And there were many namespace collisions in Zimbra, due to its flat namespace structure, but they learned a lot in their test system.

In the end they had to start the real migration two weeks before the switchover date in April. Things weren't ready when the switchover came, but they threw the switch anyway to deliver to the new mail server while the sync was continuing. In the end, they created a new mailbox for each user and showed users how to access both mailboxes and move what mail they needed from one server to another if it was needed.

Craig noted that hindsight is 20-20 and that they could have done it more slowly with more testing, but that they were really afraid the system was going to fall down. Leveraging users to do the migration is useful, as not everything needs to migrate and the individual users know what should and shouldn't be migrated. The most important lesson is that one shouldn't get complacent when a system seems to just barely work, for if you don't keep it up to date on hardware and software, it will get to the point where you can't make any changes to it without fear that you'll break the system.

## INVITED TALK

- ***How to Build a PB Sized Disk Storage System***

  *Raymond L. Paden, HPC Technical Architect, IBM Deep Computing*

  *Summarized by John Hewson (john.hewson@ed.ac.uk)*

Petabyte-sized disk storage systems seemed unthinkable, but today they are increasingly common. Vendors are responding by making larger, less expensive disks and controllers that manage over a PB of data. Yet datacenter managers and application developers do not truly appreciate the complexity of these large storage systems. Too often they approach them as being peripheral rather than integral to the overall processing system. They give detailed attention to other things, but simply ask for a storage system like the ones they had before, only bigger. This often leads to an overly costly storage system that is a performance bottleneck. This talk will help you avoid the pitfalls and develop an efficient PB-sized system.

Paden talked about the issues encountered when building a petabyte (PB) sized storage system, and the problem that existing paradigms do not scale to the new sizes required. The key message of the talk was that to implement a PB storage system a number of questions need to be answered first: what is the I/O profile? Is a LAN or a SAN used? Can NFS or CIFS be used, or is a specialized file system needed?

Paden said that it is necessary to understand the profile of the system's users and their working set: what is the optimum working set size and cache locality? Temporal and spatial locality were examined, as were the implications of the storage access patterns of the users, highlighting differences such as streaming large files, small transactions for I/O processing, and transaction processing with temporal locality. Paden concluded that most environments have a mixed usage pattern and that it is best practice to develop use cases prior to making a purchase. Use cases provide a model of realistic use and require more time to evaluate than simple performance benchmarks; however, they provide a less synthetic method of evaluation.

Paden introduced the building-block strategy, where the smallest common storage unit consisting of servers, controllers, and disks is defined. This block is then used to construct a datacenter, ensuring that each device in the building block is balanced and optimized. Large building blocks are recommended for PB-scale systems.

The limitations of SAN file systems were mentioned, with LAN-based systems recommended as a cost-effective alternative. Paden described a taxonomy of file systems commonly used with clusters: conventional I/O, asynchronous I/O, networked file systems, network attached storage, basic clustered file systems, SAN file systems, multi-component clustered file systems, and high-level parallel I/O. Choice of the correct file system depends on the user profile.

Management of risk was briefly outlined, including disaster recovery and avoidance of single points of failure, such as the use of RAID 6 with SATA disks. The question and answer section identified managing the large number of cost-effective disks as a key issue for the future.

Doug Hughes, the Program Chair, asked if Paden had played with Lustre. Paden said he had not but Hughes was the second person to mention it. Hughes described Lustre as easy to install, reliable, and fast. Someone asked about flash drives and SSD. Paden replied that he was under non-disclosure but could say that these technologies are undergoing a period of rapid flux. Someone asked about the next big challenge to building file systems: It is managing the number of moving parts and having the tools to do so. The problems are "Texas big." As the session closed, people queued up to ask more questions.

## INVITED TALK

- ***Eucalyptus: An Open Source Infrastructure for Cloud Computing***

  *Rich Wolski, Chief Technology Officer and co-founder of Eucalyptus Systems Inc. and Professor of Computer Science at the University of California, Santa Barbara*

  *Summarized by Chuan Yue (cyue@cs.wm.edu)*

Professor Rich Wolski gave a wonderful talk on Eucalyptus, an open source infrastructure for cloud computing. Wolski first mentioned that cloud computing is a reality. People not only talk about it, but also spend money on it. Many companies, such as Amazon and Google, have tremendously powerful infrastructures in place today. Using Amazon as an example, Wolski pointed out that two important features make cloud computing work. One is SLA (service level agreement), which tells you what level of services you will get when you purchase. The other is the transaction, which has really driven the interest in cloud computing.

Wolski then explained why they decided to build Eucalyptus. People are using public clouds daily, but what really happens inside the public clouds is not transparent. There absolutely are many issues that distributed computing researchers should think about. Therefore there should be an open source infrastructure that allows people to experiment, play with, extend, and understand cloud computing. This infrastructure should not be a refactorization of previously developed technology: "It has to be developed from the first principle to be a cloud." This infrastructure should be open source so that its openness and exposure to the community can drive the technology forward.

Wolski emphasized that they borrowed something from the idea of the Turing test to make Eucalyptus a real cloud. What they did is to emulate an existing cloud—Amazon AWS—and support its APIs to the point that a user or a program cannot tell the difference between Amazon AWS and Eucalyptus. They built Eucalyptus to foster greater understanding of cloud computing and provide a technical overview platform for public clouds. But Wolski emphasized that Eucalyptus was not built to compete with or replace Amazon AWS or any other public cloud service. Eucalyptus needs to correctly implement Amazon's cloud abstractions and semantics, which are defined for scale. Eucalyptus also needs to be simple, scalable, system portable, and configurable.

The Eucalyptus infrastructure they have built is a bunch of open source Web service components that have been stuck together to make a cloud out of whatever resource users have. Eucalyptus has a layered architecture. The top layer is the translator that renders various client-side APIs into the internal representation of the cloud abstractions. The middle layer includes the VM controller and the storage management service. The bottom layer is the resource management.

After describing what they have done, Wolski went on to share what they have learned from building Eucalyptus. First, the notion of private clouds does not really exist; almost all the deployed private clouds are hybrid clouds. Second, storage architecture is hard to change in a company, because institutional knowledge and policy are embedded in the storage architecture. Third, cloud federation is a policy mediation problem. Last, a really new thing in cloud computing is that an application can measure the dollar cost associated with its execution. Wolski also argued against two myths of cloud computing: "Cloud computing is nothing more than virtualization" and "The cloud is grid." Using their careful performance comparison results, Wolski demonstrated that the third myth, "Cloud imposes a significant performance penalty," is also not true.

Finally, Wolski showed that Eucalyptus has been downloaded over 55,000 times from all around the world, and there are many real systems running on top of Eucalyptus. Wolski also showed their open source distribution effort and roadmap.

An audience member asked why open source is that important in Eucalyptus, and yet no open standard has been developed around it. Wolski replied that they say open source is important not only because they personally have benefited from open source, but also because they really think innovation will come from open source. It is a mistake to standardize too early; whether it is worth it to standardize should depend on users' needs. What are the challenges with eventual consistency in cloud computing? It is tricky but possible to put an SLA on the consistency semantics, and there is a tradeoff between this SLA and the scale of your applications. Is SLA supported in Eucalyptus? Not yet, but they certainly plan to support it.

## INVITED TALK

- ***The Advanced Persistent Threat***

  *Michael K. Daly, Director of Enterprise Security Services, Raytheon Company*

  *Summarized by Rik Farrow*

Daly made a no-nonsense presentation from the perspective of a security officer of a defense contractor, something that he told us would have been impossible just a few years ago. What's changed is the threat landscape, as well as how large US companies now feel about sharing security information. Part of the sea change includes the involvement of nation states in espionage and attacks.

Daly launched into a description of the Advanced Persistent Threat (APT) by describing a scenario in which a malware-infected PDF file is downloaded from the USENIX Web site. The file contains a trojan that upon opening installs malware that begins beaconing back, using an HTTP request. Once acknowledged, the beacon packets become rare, perhaps once a day, once a week, or even once every six months. Eventually, the attacker uses the beacon to connect to your infected system and start using your machine. The malware includes download capability, so it can update itself. The malware can do anything you can do, as well as use your system as a hop site (relay), a malware repository, an infection site, etc. For 2009, using data from F-Secure, Daly broke down the exploited software as: Acrobat 48.87%, Word 39.22%, Excel 7.3%, PowerPoint 4.52%. I personally thought that Flash belongs in the list as well, and Daly does mention Flash a bit later in his talk, with the use of zeroday exploits using Flash in Firefox.

Gh0stnet was a good example of APT, targeted specifically at the Dalai Lama and other Tibet-related groups, where 1300 systems were remotely controlled. Daly noted that he was not picking on China, but using a public domain report. You can find public documents about Computer Network Attack (CNA) and Integrated Network-Electronic Warfare (INEW) in China, and it's expected that this is going on in most other countries as well.

Daly provided a bullet list of things you can do to protect your networks:

Focus on traffic in and out of your networks, using network analysis tools; initiate awareness training, including targeted training for specific people; compartmentalize your environment and watch inputs and outputs via the network; drive down the dwell time, that is, the amount of time between malware installation and discovery; share and collaborate, working with other groups.

Daly pointed out that dynamic DNS gets used a lot for malicious purposes and that every site on the Russian Business Network can be considered bad. He's seen DNS used as a covert channel. Raytheon has blocked some popular attachment types, such as .zip, and people get used to it.

He also suggested looking for regularity in beacon packets, or for User-Agent strings in HTTP requests that have subtle changes, by using Pareto searches. Daly mentioned using Web proxy server logs to uncover systems that have visited malware download sites.

Rik Farrow suggested that dangerous apps only be run in sandbox environments, and Daly said they were working on that now. Tom Limoncelli worried about working with China, as Google does. Daly recommended segmentation, that is, not opening everything up to untrusted partners. They also have a clean laptop loaner program for visits to other countries, and they set up special Web-based email accounts for trips. If somebody demands your password, you can give it to them, because it won't work from the outside. Another person said that if the US government makes us take off our shoes and toss our water bottles before we can fly, they should be able to outlaw software that is too complex to run safely. Daly replied that people want cool features and that the government tries to keep costs down by using COTS software. Carson Gaspar asked if they are seeing obfuscated email and attachments, and Daly said they are even seeing steganography. Julie Bauer of AMD wondered how she could find out more about travel issues, and Daly suggested the US State Department Web site.

## PLENARY SESSION

- ***Google Wave***

  *Daniel Berlin and Joe Gregorio, Google, Inc.*

  *Summarized by Chuan Yue (cyue@cs.wm.edu)*

Joe Gregorio explained that Google Wave is a collaborative communication environment. They started from the question: "What would email look like if it were invented today?" Unlike email messages that are sent back and forth between users, every conversation in Google Wave is a collaboratively edited document. People involved in a conversation can add information to the document and see how this document evolves. The user interface of Google Wave is similar to Gmail and it has inbox, contacts, waves, and controls. Google Wave supports various content such as text, markup, image, and gadget, and it is more than just a collaborative text editing environment.

Gregorio differentiated the Google Wave product from underlying wave technology: "Wave is to Google Wave as email is to Gmail." The important part is federation, which is a technology that enables different wave providers to share waves with each other. Google Wave Federation Protocol (http://waveprotocol.org/) is open and iterating. It includes specifications and white papers. Federation is important both in avoiding fragmentation and in encouraging adoption. For example, Novell Pulse (http://www.novell.com/products/pulse/) has adopted the Google Wave Federation Protocol and can work seamlessly with Google Wave.

Going into the details of Google Wave, Gregorio described the wave data model. A wave is the container of wavelets, while each wavelet is actually a conversation. A wavelet contains a list of the conversation participants and a set of XML(ish) documents and annotations. A document is a linear sequence of items. It looks like an XML document but is not, because element and attribute names are well beyond what XML allows. Annotations are associated with the items in a document to provide various functionalities. The wavelet is the unit of concurrency. The wavelet is also the unit of federation.

Gregorio explained that in Google Wave, federation means sharing wavelets. Federation begins when a user adds someone outside the user's domain to a conversation. Wave servers run operational transforms to share the updates of the wavelets. Operational transforms incorporate operations made to a wavelet and then send transformed operations to each user so that all the users can end up with the same shared state. The Google Wave federation system sits on top of the XMPP technology: a wave server is an XMPP server, and federation is a service inside the XMPP server.

Gregorio mentioned that, so far, they have published two draft specifications: Google Wave Federation Protocol and Google Wave Conversation Model. A Java open source implementation of the specifications is also available (http://code.google.com/p/wave-protocol/). Gregorio showed that they have opened up a federation port on http://wavesandbox.com/, but it is still highly experimental. Gregorio also gave a few demos to illustrate the previously introduced concepts such as conversation, wavelet, and document. Finally, Gregorio said that they will open up an open federation port on http://wave.google.com/ once the specifications become stable; meanwhile, he mentioned that Google wants more people to participate in this project.

Someone asked whether they have explored the idea of making Google Wave completely peer-to-peer rather than using a client-server model. Gregorio replied that he was not involved in the early discussion of Google Wave, so he has no idea whether there is a discussion about peer-to-peer. But he argued that there has to be a centralized server to put together all the changes from the clients and then ask the clients to apply the changes before sending an update. Ari Redkin of UC Berkeley asked where the certificates used in Google Wave come from. Wave servers signed the certificates. Therefore, when a client tells the server to make changes to a document, the server knows where the request comes from. Gregorio acknowledged that currently a client is tied to a particular wave server because the server has the private key of the client. Has there been any thought or discussion about dealing with spam, malware, and virus in Google Wave? Certainly work needs to be done, and a white paper will come out soon talking about that. In Google Wave, at least all the operations are signed.

## PLENARY SESSION

- *Cosmic Computing: Supporting the Science of the Planck Space Based Telescope*

  *Shane Canon, Group Leader, Data System Group in NERSC, Lawrence Berkeley National Laboratory*

  *Summarized by Leah Cardaci (lcardaci@cs.iupui.edu)*

Shane Cannon discussed the Planck project and NERSC's work supporting research on the resulting data. Cannon began with a disclaimer that he was not an astrophysicist, cosmologist, or rocket scientist.

Cannon first presented an overview of the science of the cosmic microwave background (CMB) and why it is studied. The CMB is one of the first observable conditions from the Big Bang. Details about it can provide information about the Big Bang, the geometry, composition and shape of the universe, and the dynamics involved in the evolution of the cosmos. It was accidentally discovered in 1965 by Penzias and Wilson, physicists working for Bell Labs looking at types of background signals. They noticed an independent background signal that they could not eliminate. Eventually,they realized the signal was coming from space. In 1978, the pair won a Nobel Prize for their discovery. Since its initial discovery, the CMB has been studied in ground-based, balloon-based, and space-based projects.

Planck, a joint mission of the ESA and NASA, is a new space-based project that will look at fluctuations in the CMB to study the Big Bang and some basic properties of the universe. It will provide the biggest datasets for the CMB to date, with $10^{12}$ observations. The Planck satellite has both a low frequency and a high frequency bank of instruments. Advanced cryogenic shielding is needed to control the heat generated by the instruments. The ESA launched the Planck satellite on May 14, 2009. It is in orbit around the second Lagrangian Point (L2), a stable point relative to the Earth and the Sun. The results from Planck will allow for a more detailed map of the CMB to be built. As the dataset has grown, the analysis has had to move to more iterative methods that scale in a more sustainable manner.

The data is beamed from Planck to various ground stations. After some initial analysis, the low and the high frequency data are split off to separate processing groups. At this point, that data is sent to various places, including NASA's IPAC. NERSC is not a formal part of this pipeline but supports a lot of the analysis and handles both the high and low frequency datasets. CMB data is analyzed with time order data that is used to generate maps. Analysis of the CMB data is primarily concerned with how to remove the noise present in the measurements.

NERSC is the flagship computing center for the Department of Energy's Office of Science. It began in 1974 at Lawrence Livermore and moved to Berkeley Lab in 1996. It serves a large population of diverse interests. The center focuses on high end computing, storage, and networking. Because of the variety of groups using the resources, NERSC systems need to be flexible to meet multiple clients' needs. There are several groups of systems involved in this support. The flagship system is Franklin, a Cray XT4 massively parallel processing system with 9,740 nodes. In addition, a new Cray-based system is presently being built. NERSC also has some smaller clusters, a 400 TB global file system, and an archival storage system. The GPFS-based global file system was created to permit clients to avoid the burden of moving data to various systems but still allow high-performance access. It is a large SAN that connects to the various NERSC systems in a variety of optimized methods.

NERSC is supporting a variety of projects involving big datasets. Some projects are beginning to require data handling that almost outpaces the system's current abilities. Recent projects are expected to generate petabyte datasets. In addition, these datasets will be analyzed long after the original observations, creating a need to preserve the data for the long term. Cannon provided a select list of example big data projects at NERSC, pointing out that some new projects may require more high performance data support and not as much high performance computation support. KAMLAND is a neutrino detector experiment that has generated 0.6 TB of data in six years. ALICE is a soon-to-deploy collider experiment that is looking at QCD (quantum chromodynamics) matter. It is expected to generate around 600 TB of data in the first year, with a long-term estimate of 3.8 PB of data. Other future data-intensive projects include trying to build a model of global climate from 1871 to the present, building cloud resolving climate models, and the Joint Genome Institute looking at microbial DNA.

Cannon finished by relating general observations about dealing with data-intensive projects and handling large datasets. Successful projects use a shared data model, employ parallelism in multiple levels of the process, design to deal with failure, avoid the I/O bottleneck whenever possible, and consider the true lifecycle of the data. Technologies that enable such projects include scalable archive systems, parallel file systems, data management middleware, and visualization technology. Challenges include the continued imbalance between capacity and bandwidth, the fact that common utilities are not designed with the new storage approaches in mind, the lack of improvement in bit error rates, and the need to deal with the new requirements for long-term storage of these large datasets.

What is the name of the new NERSC computing system? Cannon said it was named Hopper. Had NERSC made any customizations to GPFS for their global file system? They basically ran the system out of the box, but had collaborated with IBM to meet their needs. However, they have customized the Cray system.

## INVITED TALK

- ***Storage and Analysis Infrastructure for Anton***

*Mark Moraes, Head of Anton Software Development and Systems Software, D.E. Shaw Research*

*Summarized by David Plonka (plonka@cs.wisc.edu)*

Mark Moraes introduced LISA attendees to the Anton supercomputer, a new machine meant to impact biology and chemistry by using advanced computational methods and hardware. First, to put their achievement in context, Mark introduced us to the fascinating world of computational chemistry and molecular dynamics (MD). MD involves simulating what molecules do, in the study of proteins, for example. Basically, they simulate the molecule surrounded by a cube of water to observe behaviors such as wiggling and folding that are key to the molecule's function. Simulation is an important technique because the alternative, experimental method, is difficult and involves error-prone purification and crystallization. The molecules themselves are complex: a modeled protein could become a 50,000 atom system when the water is modeled explicitly. To understand further why simulation on biological timescales is hard, he informed us that most organic molecules are held together by bonds that vibrate on the femtosecond ($10^{-15}$ of a second) timescale, while other important behaviors happen on the timescale of milliseconds.

Through the use of new MD algorithms that could scale to a large number of interconnected cores, they saw the possibility of running simulations 100 or 1000 times faster than previously feasible. However, modern commodity CPUs have limitations as a building block for such a system, since not much of the chip area is devoted to computation (lots of it is cache instead). Since MD computations involves a relatively small number (tens of thousands) of atoms, D.E. Shaw Research decided it was worth building a new machine based on custom-designed ASIC-based chips; while the resulting supercomputer may be less flexible, it would be dramatically faster for MD applications.

Thus Anton was born, coming online last year; it is used to study molecules, small proteins, and DNA via simulation. Its performance, as measured by a comparative benchmark, shows that it dramatically outperforms its predecessor, Desmond. For instance a 1024-core, 512-node cluster running Desmond might achieve about 500 nanoseconds of simulated behavior per day, whereas the Anton supercomputer, with its 512 ASICs, can achieve about 16,000 nanoseconds (16ms) of simulated behavior per day. Such dramatic performance gains in simulation are expected to change the way research chemists do their work. Indeed, in response to Anton's performance, one chemist remarked, "I'm going to have to think faster!"

Continuing with a system administration–specific portion of this talk, Mark described the significant infrastructure that supports Anton's supercomputing capability. For control and management, their front-end machine uses Linux running CentOS 4.x, and software such as syslog, ganglia, PostgreSQL, dhcpd, and tftp. There are also some custom Anton management components that employ JSON-RPC and the slurmd job queue system. The I/O and storage subsystem relies on NFS with the data resulting from runs accumulating into the hundreds of gigabytes to a terabyte per day. A custom file-based organization avoids having to search unnecessarily for data in volumes consisting of thousands of terabytes of storage. The parallel analysis portion of their infrastructure is a framework that they developed called HiMach; it is inspired by Google's MapReduce but is specific to the MD data structures they use. There are also Linux-based control processors on boards within the machine, which assist in monitoring and managing the supercomputer's custom ASIC processors.

Mark wrapped up his presentation with a fascinating animated movie clip, based on Anton simulation, of the folding of the villin protein headpiece. These and other sample animations dramatically show the behaviors of these microscopic structures at fine timescales, and researchers appear to be about to arrive at new scientific results that seemed impossible with the prior slower simulations. Thus, it appears that this new instrument, Anton, will allow researchers to arrive at useful results much more quickly and, in some cases, arrive at results that no one had the patience to develop by simulation before.

At the close of the session Mark fielded questions from the audience: How programmable are the ASICs employed in Anton? There is a flexible component, coded in C, that is often changed, especially for force fields; about half of the ASICs (12 cores) can be changed. The pipelines also have tables that allow their function to be changed somewhat. Is visualization performed in real time during a run? They can do this, but typically just a little bit of visualization is used during the run to determine if it's working correctly. Are custom or standard compilers employed for the programmable cores? They license four general-purpose cores from Tensilica, which provides customized compilers for these cores. D.E. Shaw Research designed some other components themselves, with custom instructions, and they now have a gcc 4.3 port that generates pretty good code for them. Mark added that it takes a long time to port gcc, so if you're in this situation, you should start early. How does Anton differ from the MD-GRAPE supercomputer? MD-GRAPE involves pipelines, deals with distant forces in hardware, and ignores the Amdahl's Law bottleneck. Anton ties computation together end-to-end.

## NETWORKS, NETWORKS, NETWORKS

*Summarized by Rik Farrow*

- ***Crossbow Virtual Wire: Network in a Box***

  *Sunay Tripathi, Nicolas Droux, Kais Belgaied, and Shrikrishna Khare, Sun Microsystems, Inc.*

  ***Awarded Best Paper!***

Nicolas Droux explained that virtualized environments and services need to share physical network interfaces. In this eight-year-long project, the big focus was on performance and also being able to take advantage of hardware that has multiple rings, DMA channels, and classifiers. Security was also important, so there is real isolation between flows and no ability to sniff or inject traffic into another flow.

The project built on previous work, such as nTop, streams, and Nemo, but needed to improve on management, which had been difficult in the past. Crossbow uses the notion of Virtual NICs (VNICs). Each VNIC is assigned a hardware lane that may consist of NIC resources (like rings and channels), and this lane, and any threads and interrupts, gets bound to a specific CPU. This binding avoids context changes and improves cache coherency. To control bandwidth in a VNIC, interrupts can be disabled and replaced with pulling chains of packets. Priority Flow Control (PFC) allows the use of VNICs with services instead of just VMs. The VNICs themselves connect via virtual switches, and these switches and the use of VNICs allow the modeling of a physical network within a single Solaris (or OpenSolaris) system. Droux pointed out that a student could be sitting in a cafe with a network model on his laptop, designing the next routing protocol.

Tom Limoncelli (Google) asked the only question, wondering if the bandwidth limits were absolute or if they allowed bursts? Droux answered that they were looking at bandwidth guarantees instead of limits, and when they have that, bandwidth use would become more flexible. See Peter Galvin's column on p. 79 for more details about Crossbox.

- ***EVA: A Framework for Network Analysis and Risk Assessment***

  *Melissa Danforth, California State University, Bakersfield*

Melissa Danforth described EVA (Evolutionary Vulnerability Analyzer) as an attack graph tool that supports a multitude of analysis modes. Host-based vulnerability scans produce information that is limited to each host. With EVA, the user can start by imagining an attack that provides a foothold within a single system, and see ways that the attack may spread to other systems.

Danforth used a diagram, built with EVA, that showed two groups consisting of a total of six systems. Two Internet-facing systems provided the initial foothold, and via vulnerabilities found on four internal servers an attack would eventually produce privilege escalation to root on the internal servers. EVA does this by producing attack graphs, where the nodes represent systems and the edges represent successful exploits. Exploits can be chained together to form templates: for example, a template for a vulnerable SSH server that provides both the compromise and privilege escalation (to root).

Nessus is used to scan for host-level vulnerabilities, and several attacker models are used: for example, an insider or an external attacker with no privileges. The Java Expert System (JES) is used to create the graphs, although the process is not automatic and requires some user input. The tool has been used to encourage sysadmins to patch systems they have been ignoring by showing how failing to install a patch can lead to many systems being exploited. EVA can also be used for forensics by uncovering possibly exploited systems and for network design.

- ***An Analysis of Network Configuration Artifacts***

  *David Plonka and Andres Jaan Tack, University of Wisconsin—Madison*

David Plonka and his co-author had the good fortune to be sitting on top of a 10-year repository of network configuration changes. When his university began building out their network, they also began using RCS as a revision control system. In this paper, the authors mined this repository for insights.

David said that they borrowed heavily from the world of programming, where many studies of source code changes had already been done. First, they converted the RCS records into CVS so they could use tools such as statCVS-XML and cvs2cl during the analysis. Then they began to pry out details of who made changes, when they made them, the types of changes made, and how quickly new revisions were made.

The campus network has over 3800 devices, with many being access layer (switches and wireless). Web interfaces allow 300 authorized users to make some changes, and other smaller groups, about 64 people in total, have root access. Still, 75% of all changes were made by just five people, all part of the network engineers group. They also found that 90% of revisions were interface changes, something that could be done by authorized users, and that VLANs were another good target for their Web interface. They could also see that there were many short-lived revisions as people tried changes, then backed them out or changed them, apparently because they didn't work. David pointed out that network management is different from programming because you are working on live systems.

One person asked about the use of RANCID (Really Awesome New Cisco confIg Differ), as RANCID doesn't include comments or the author of changes. David said he only mentioned RANCID and preferred his own scripts. What is the difference between the campus info and info gathered from a provider network (large ISP)? The difference is that the campus has lots of access-level devices.

## INVITED TALK

- ***Searching for Truth, or at Least Data: How to Be an Empiricist Skeptic***

  *Elizabeth D. Zwicky*

  *Summarized by Leah Cardaci (lcardaci@cs.iupui.edu)*

Elizabeth Zwicky opened the talk with a warning that all of the numbers in the talk were made up, but all of the stories were true. The talk was addressed to system administrators who look at information about technology. Skepticism towards data is a trait that good system administrators and good security people have in common. It involves the desire to learn about something, the ability to understand the difference between appearance and reality, and an ability to understand numbers.

As an example of the difference between appearance and reality, Zwicky related  how a former coworker who knew how to pick locks was asked by the company CFO to break into her office. The coworker pointed out this wasn't really a lock-picking problem, popped out a raised floor tile, and used a coat hanger to pull the handle and open the door. While the office appeared to have a solid wall, it was easily bypassed.

Taking this approach to finding and considering data can prevent logistical nightmares, is part of troubleshooting and security, is fun, and prevents you from falling for pseudo-science. When looking at potential data, determine if it is really data, consider what it is data about, and ask what conclusion can be drawn from the data. Zwicky went through several examples of potential data, such as "Brand A's router has an error rate 200% worse than Brand B" and discussed the value of each example. Hearsay, numbers without context, and conclusions are not data. Observations, numbers with context, and self-reports are data.

Basic statistics can help one understand whether given numbers have appropriate contexts. When looking at averages, it is useful to know what kind of average is meant. One common average is the mean. Mean is interesting when discussing a bell curve graph that is fairly symmetric, but not with other distributions. Graphs related to machines do not usually have that shape, so mean is not a useful measurement. Other measurements such as the median, quartiles, and percentiles, or the entire graph are more relevant. If only a mean is available, looking at the standard deviation will show how the distribution is skewed.

When given a rate, asking "Compared to what?" can provide information with needed context. For example, a 200% increased risk of being hit by a meteor is still very low, because the initial odds are incredibly low. Correlation does not equal causation. Two correlated events can both be affected by other unmentioned factors. If someone is looking to make a point, they will likely be able to find some correlation to support the claim.

Zwicky showed an example comparing numbers of users versus network usage per month at an ISP. While the initial months appeared to be following a clear predictable curve, the seasonal activity surrounding Christmas caused a change in behavior. Two significantly different predicted curves for the activity show how people can interpret the same numbers in different ways. Without the appropriate context, it is not easy to know what the given data is really showing. Seeing data without knowing what the data is really about does not provide much information.

There are a variety of ways to gather data. You need a programming language to process the data, preferably one that manipulates text well. You need some tools to look at the internals of what is happening with programs and network traffic. Some examples of such tools are trace, dtrace, truss, wireshark, tcpdump, and Windows Sysinternals. Spreadsheet programs, GraphViz, and gnuplot can be used to make pictures of the data you gather. Some basic knowledge for handling data is basic statistics, SQL and XML, and writing regular expressions.

There are multiple ways to find data sources, including mining existing data or learning new data. For example, look at logfiles. If there are no existing sources, collect new data. Data can be collected with logging, tracing or sniffing, or running tests. Simulate data or extrapolate data from some known information. See if data can be gathered from published sources and colleagues. If all else fails, "make stuff up" by guessing. This process can be at least slightly improved by collecting a variety of guesses, basing guesses on known information, and testing various guesses. When collecting data about people, be prepared to go through a Human Subjects Board. It can be difficult to design an unbiased survey. In such a case, it may be better to gather descriptions rather than numerical measures.

Once you have the data, the interesting part may be obvious, but you will likely need to analyze it. Sometimes you need to check the data to be sure it makes sense and measures what you want to measure. Once you have the data it is important to know what questions you want to ask when analyzing the data. Humans are good at certain types of pattern recognition, such as recognizing abrupt change, noticing correlation, and seeing faces. They are not good at understanding probability, seeing when things aren't related, noticing slow change, and understanding a delayed correlation.

To show data well, know what the message is and limit any extra facts shown. To avoid lying with graphs, understand how people perceive graphs. Humans do not perceive area well. For this reason, pie charts are not an effective tool.

Zwicky gave a detailed example of the problem measuring performance of a help desk. Time to completion was not an effective metric, because it encouraged workers to prematurely label a job or try to hand off jobs to others. An alternative measure was a customer satisfaction survey. When considering the results, it was important to think about the behavior of the people who filled out the form. Most would be people who felt strongly about the help desk service and not those with an average experience. As the majority of employees were not likely to follow instructions for no reason, the survey results were skewed. Zwicky showed a variety of different graphs of the survey results. Looking at the data in multiple ways showed information that was not visible in the most basic view of the results.

One person asked how to deal with management's insistence that a known bad metric is better than no metric at all. Zwicky suggested trying to replace the bad metric with a better metric. Making the new metric more appealing in some way can help.

## SECURITY, SECURITY, SECURITY

*Summarized by Shaya Potter (spotter@cs.columbia.edu)*

- ***Secure Passwords Through Enhanced Hashing***
  *Benjamin Strahs, Chuan Yue, and Haining Wang, The College of William and Mary*

Chuan presented PasswordAgent, which is meant to provide secure passwords for Web site usage. This is important, as passwords are the most common online authentication method and will remain dominant for the foreseeable future. However, passwords are crackable if weak, and vulnerable to theft, especially as users use the same password at many sites.

Many approaches have been used to try and secure passwords, including password managers, but those lack mobility, and single sign-on systems, but these provide a single point of failure. Password hashing, taking a weak password and making it strong by hashing it with other data, is what the authors built on.

PasswordAgent is based on PwdHash, a Web browser-based solution that creates a unique password for each site based on a hash of the password and the domain name of the site being accessed as a salt. PwdHash is meant to focus on phishing attacks, since it would give a phishing site an incorrect password from a different domain name. Rather than using the domain name to hash together with the password as PwdHash does, PasswordAgent creates random salts that it hashes together with the password. It stores these salts in a salt repository, accessible from many different machines, and in multiple repositories so that one doesn't have a single point of failure. PasswordAgent is built as a Firefox extension and hooks into password fields, enabling the password to be replaced when the password is protected by PasswordAgent.

PasswordAgent protects passwords in many different scenarios. For instance, it doesn't have a master password to steal. Furthermore, even if the plaintext password passed to a Web site is compromised, the real password is still protected as long as the salt remains secure. It also reduces the risk of weak passwords, as hashing them together with the random salt increases their security. Even if an attacker could guess the user's password, they would have to iterate against every possible salt. Finally, PasswordAgent protects against phishing by notifying users when they enter their passwords into sites that have not been set up to be protected by PasswordAgent. If the user expected this site to be protected, it's indicative that this is a phishing site.

Limitations of PasswordAgent include vulnerability to malware on the system that can see the salts as well as the passwords before they are hashed, as well as its dependence on the salt repository. If the repository becomes unavailable, one will not be able to create the hashes.

What happens if domain names change, such as one Web site being purchased by another company? The password would have to be manually reset, which would generate a new salt.

- ***SEEdit: SELinux Security Policy Configuration System with Higher Level Language***
  *Yuichi Nakamura and Yoshiki Sameshima, Hitachi Software Engineering Co., Ltd.; Toshihiro Tabata, Okayama University*

Yuichi presented SEEdit, a tool to improve and simplify the configuration of Security Enhanced Linux (SELinux), which provides least privilege via type enforcement and mandatory access control. However, while useful, SELinux has a bad reputation; in fact, many recommend disabling it if one has problems. The reason for this is that security policy configurations are difficult.

Refpolicy is the current way to configure systems. It is developed by the community, policies for many applications are included within it, and it works well when the system is used as expected. However, it fails when used in other ways. Furthermore, because it's such a big policy, it's very difficult to understand how to change it to fit one's needs when they differ from the expected scenarios.

SEEdit tries to fix this problem by letting one write SELinux policies in SPDL, which is a higher-level language that hides many of the complexities of SELinux's policy language. After writing the policy in SPDL, it's translated into SELinux's own policy language. Instead of having to create type labels manually, SEEdit automatically generates types for permission rules listed in the SPDL language.

Yuichi demonstrated that SEEdit is able to create complete configurations using many fewer lines than are required by Refpolicy to describe the same security policy, making it much easier for a user to read and grasp, as well as verify. Furthermore, smaller policies enable SELinux to work in embedded environments where space can be at a premium. For instance, Refpolicy-based security policies can take a few megabytes of space, while a SEEdit-created policy only took up 71k of space.

However, the problem with SEEdit is that its current approach integrates multiple SELinux permissions into one higher level which are merged, reducing granularity. For instance, reading a regular file and symbolic links are a single permission in SEEdit. They would have to expand SEEdit to understand more permissions.

Could SEEdit be used to manage the Refpolicy itself once its few issues are worked out? It would be possible to use SEEdit to manage the policy. A question was asked about the difficulty of debugging misconfigurations if one's configuration doesn't work as expected. The bug could be a result of the conversion into SELinux's language, so one might not know which SPDL rule created the SELinux policy rule that caused the problem. Yuichi agreed that this is important and needs to be worked on.

- ***An SSH-based Toolkit for User-based Network Services***

  *Joyita Sikder, University of Illinois at Chicago; Manigandan Radhakrishnan, VMware; Jon A. Solworth, University of Illinois at Chicago*

Jon Solworth spoke about securing user-based network services (UBNS) easily. UBNS involves authenticating users, encrypting communications, and authorizing and customizing the services based on the user authenticated. Different users have different access permissions.

In general, password authentication is used to authenticate to these services, but it isn't that secure. For instance, Dovecot mail service has a good reputation for security. It was built using four different process types for privilege separation, and about 37% of the source code is just implementing authentication and encryption, ignoring external libraries such as OpenSSL. This portion was implemented by a security expert and is not so easy to implement for regular programmers.

They've developed the SSH-based UBNS toolkit, which makes it much easier to provide all the requirements of UBNS with minimal changes to existing service applications. It doesn't require any knowledge of cryptographic libraries. It isolates the UBNS functionality into address spaces separate from the service functionality to have the OS enforce the isolation. By building on top of SSH, it doesn't require a global namespace, but instead allows each user to create their own public/private key pair.

The implementation is a modification to SSH. In a traditional SSH tunnel to a running service, the running service has no direct knowledge of the user who set up the tunnel, but with UBNS this information is available to the service. On the server side they provide an inetd type super-server, unetd, to manage connection to their managed UBNS services and require a simple modification to applications in the accept() function to make it UBNS aware.

On the client side, no modifications have to be made; instead the client connects to a local port on the client, which initiates an SSH connection to the unetd super-server to instantiate the requested services as the correct user and sets up the tunnel

## PLENARY SESSION

- ***Towards Zero-Emission Datacenters Through Direct Reuse of Waste Heat***

  *Bruno Michel, IBM Zurich Research Laboratory*

  *Summarized by Rik Farrow*

Bruno Michel began by telling us that the energy consumption in datacenters has doubled in the past two years. Using a chart from the International Technology Roadmap for Semiconductors, Michel pointed out that energy leakage, manifest as energy wasted as heat, will only get worse as chip features continue to shrink. He also mentioned that he is a mountaineer and has personally seen the shrinkage of glaciers in the Alps, an obvious effect of global warming.

There have been improvements in energy usage, spurred on in part by the awareness of the impending crises. The Green 500 ranks supercomputers by the amount of useful work produced per watt. The number one supercomputer in 2002 produced 3.4 Mflops/watt, while the current fastest one produces 445 Mflops/watt, while being ranked number four in the Green 500 list.

The thrust of his presentation had to do with using water for cooling. The circulation of blood is efficient in transferring both nutrients and heat. Water itself has tremendous thermal capacity, much higher than refrigerants. IBM began using water to cool chips with its 3070 mainframe back in the '80s. Each processor chip (including those that controlled data transfer like today's northbridge) had a piston that rested on the chip, with an armature sitting above the set of pistons for circulating water.

Working from biology for modern design, Michel described fluid channels built right into chip carriers, with tiny channels nearest heat-producing parts flowing into larger channels above. Chip design needs to consider the position of cooling channels for areas that will be the hottest sections of chips. IBM has planned for stacked chips, with vertical interconnecting pins, interlaced with water cooling, providing shorter signal paths and much more efficient cooling.

Using this design allows water to reach 60°C (170°F), a point where it becomes feasible to resell the heat produced as a side-effect of computing. In parts of Europe, they can sell this "waste" heat for about half the cost of producing the heat via electricity or gas. Even where the climate is hot, like Dubai, waste heat can be used to preheat seawater for evaporative desalination. The goal is to reach a PUE/reuse ratio of less than one. Prototypes built using these designs are expected to cost 30% more, 10% when mass-manufactured, but these costs should come down to perhaps 3% over time.

Andrew Hume wondered if we will be needing plumber's putty when pulling boards? Michel said that admins will hardly notice the difference, as there will be connections for water and electricity. Hume asked if they had considered other fluids, because of bacteria? Michel said that corrosion is a problem, but that they have used systems like this for over ten years.

Someone asked about radially oscillating flow cooling being used in cell phones, but Michel pointed out that the cost would be quite high. Hamil Howell asked what C4 technology meant. Michel said this is a technical term,  controlled collapse chip connect, which uses solder balls that melt to make connections instead of wires. Doug Hughes of D.E. Shaw Research asked about using outside air: what's the overall efficiency? Michel said he was not an expert, but it is better to build a datacenter where you can disable the heat pump and use outside air. Doing so requires a large enough gradient for this to work, but you also have to filter out dust, which requires more power to pump the air. Steven Spackman said he grew up in Quebec where they use hydroelectric power. Michel pointed out that electrical energy is more valuable than heat energy, and that you need to understand the concept of exergy. DCs have 100% exergy, but if you can heat houses, you can get your exergy down to 10%.

## ON THE FRINGE

*Summarized by John Hewson (john.hewson@ed.ac.uk)*

- ***Federated Access Control and Workflow Enforcement in Systems Configuration***

  *Bart Vanbrabant, Thomas Delaet, and Wouter Joosen, K.U. Leuven, Belgium*

  ***Awarded Best Student Paper!***

Bart Vanbrabant discussed the current situation, in which most system configuration data is stored in source control repositories, with limited, directory-based access control being highlighted as a weak point in security or reliability. Examples were given, such as testing or development repositories, and shared configuration across grid computing sites.

Vanbrabant introduced ACHEL, a tool to integrate fine-grained access control into existing configuration tools. Using this tool, changes that are checked in to a source code repository are checked before being uploaded to a server. ACHEL is mostly language-agnostic, only requiring administrators to create regular expressions in order to apply access control to existing code. An email workflow is provided so that repository admins can approve changes made requiring higher privileges. The implementation of ACHEL was presented, with language-agnostic components discussed, as well as the language-specific abstract syntax tree parser component. Vanbrabant presented a prototype implementation as a test case, using Mercurial source control, Bcfg2 deployment, and a simple custom configuration language. A sample junior and senior sysadmin workflow was created. Finally, a larger federated example from BEGrid was presented.

The question of whether access control could be integrated into existing configuration languages was raised. Vanbrabant said that there could be better language integration but that it was desirable for access rules to be separate from the specification itself.

- ***CIMDIFF: Advanced Difference Tracking Tool for CIM Compliant Devices***

  *Ramani Routray, IBM Almaden Research Center; Shripad Nadgowda, IBM India Systems and Technology Lab*

Ramani Routray gave an overview of the DMTF Common Information Model (CIM), a standard for vendor-neutral exchange of management information, including the underlying XML, and its use in Web-based management. The problem of identifying meaningful semantic differences between the XML documents containing the CIM data was discussed, and the goal of performing meaningful difference tracking was identified. Routray then presented CIMDIFF, a tool that identifies semantic differences between devices which implement CIM, allowing system administrators to discover differences between systems.

Routray then summarized details of the implementation of CIMDIFF with overviews of the hash maker, knowledge base, and difference tracker.

- ***Transparent Mobile Storage Protection in Trusted Virtual Domains***

  *Luigi Catuogno and Hans Löhr, Ruhr-University Bochum, Germany; Mark Manulis, Technische Universität Darmstadt, Germany; Ahmad-Reza Sadeghi and Marcel Winandy, Ruhr-University Bochum, Germany*

Luigi Catuogno gave an overview of the problems with existing data protection on mobile devices such as memory cards and USB sticks. He discussed the issues of untraceability, lack of effective security, and management overhead on users. The authors offer Trusted Virtual Domains (TVDs) as a way to attain free and transparent deployment of mobile storage within an organizational network.

Catuogno introduced TVDs as a coalition of virtual machines with mutual trust, an enforced security policy, and the ability to span physical infrastructure. Catuogno introduced an extension of TVD which covers mobile storage devices, adding device identification and dynamic device management, as well as transparent mandatory encryption of sensitive data stored on mobile devices. Encryption keys are stored in a centralized key management database, and access control is integrated into the TDV policy. Offline access to mobile storage is provided by delayed re-encryption and delayed revocation of encryption keys. As a further reference, Catuogno mentions a prototype implementation based on the Turaya security kernel.

## INVITED TALK

- ***Visualizing DTrace: Sun Storage 7000 Analytics***

  *Bryan Cantrill, Sun Microsystems*

  *Summarized by Rik Farrow*

I had heard that Cantrill was an interesting speaker, and that turned out to be an understatement. Cantrill both entertained and enlightened us with his funny, fast-paced talk on DTrace.

Although ostensibly a talk about DTrace in the 7000, Cantrill started with the story behind the creation of the tool. In 1997 he was part of a team working to tune a Sun Enterprise 10000 (E10K), a million-dollar server with up to 64 UltraSPARC CPUs, to run a TPC benchmark. The E10K worked well for a while, then performance "went sucky" for about four minutes, before resuming at a benchmark record-breaking level.

Cantrill wrote kernel modules to debug the problem, not something Sun's ordinary customers would even dream of doing, and eventually discovered that the mainframe-like machine had been misconfigured to act as a backup router, and would do so when the real router would crash. This killed performance until after the router rebooted.

Cantrill went on to write DTrace with Mike Shapiro and Adam Leventhal (see their 2004 Annual Tech paper at http://www.usenix.org/event/usenix04/tech/general/full_papers/cantrill/cantrill_html/), a project that earned the STUG award for them in 2006—not that Cantrill mentioned this, as he was much more interested in demonstrating DTrace on the Mac he was using for his presentation.

Cantrill started with a simple DTrace command, dtrace -n syscall:::entry'{trace(execname)}', which lists the names of programs executed. But this gave him a way of pointing out that DTrace has an awk-like syntax, with a probe pattern, followed by an optional predicate, and an action, surrounded by curly braces, that executes only when the probe triggers. Cantrill next demonstrated aggregation and histograms using slightly more complex examples. While typing, he quipped, "Using DTrace is a way of seeking food, not mates. It is not an aphrodisiac. Just ask your wife while reading the manual out loud."

Learning that the action expressions used a C-like language, called D, that borrows features from awk actually helped me to understand the previously impenetrable DTrace examples.

Cantrill continued his presentation by firing up the demo VM of the 7000 storage appliance so he could show how DTrace helps debug storage system issues. He told the story of the famous YouTube video that shows a Sun engineer shouting at a disk array (Just a Bunch of Disks, JBOD), resulting in longer latency. They had observed some unusual latency and tracked that back using DTrace tools within the 7000 to a single drive in the JBOD which turned out to have three of the four mounting screws missing. Replacing the screws fixed the latency issue. But to reproduce it they loosened all the drive screws, tried various ways to vibrate the array using synthesized sound, then finally discovered that simply screaming at the array produced increased latencies that clearly show up in the 7000 GUI. Cantrill pointed out that the camera is clearly shaking during the video because he is still laughing.

Cantrill concluded that looking at latency issues really helps uncover problems and that every aspect used in the GUI is there because they needed to understand performance.

Someone asked if the Sun 4500, an obsolete member of the Enterprise line, could be used as a NAS appliance? Cantrill answered that this is not supportable as a NAS appliance, but with their new masters they were going to look at how they actually make money. Was the 7000 was going to support FC (Fibre Channel)? Cantrill said he was working on an FC target that should be ready by Christmas. Mark Staveley asked about FCoE (FC over Ethernet), and Cantrill went off about how this was possible but something he considered ridiculous, as it meant replacing your FC infrastructure investment, so why not just go to 10 Gigabit Ethernet?

## INVITED TALK

- ***Above the Clouds: A Berkeley View of Cloud Computing***

  *Armando Fox, Reliable Adaptive Distributed Systems Lab, University of California, Berkeley*

  *Summarized by John Hewson (john.hewson@ed.ac.uk)*

Armando Fox introduced the idea that the datacenter is the new server, albeit one requiring a large investment in infrastructure. He outlined the RAD Lab's five-year mission to enable a single individual to develop next-generation Internet apps. He introduced cloud computing as a new concept, separate from SaaS, which he describes as predating Multics. Pay-as-you-go utility computing, with the illusion of on-demand infinite resources, was identified as being the novel discriminator between SaaS and cloud computing. Fox described the advantages of provisioning in the cloud—better matching between capacity and demand makes a cloud datacenter considerably more efficient than a traditional datacenter, with perhaps 20% average resource utilization, where remaining resources are reserved for peaks.

Fox also presented some unique advantages of clouds. Cloud computing transfers risk for large capital expenditures where resource demand is unknown; cost becomes associative, as the cost of 1000 servers for an hour is equal to the cost of one server for 1000 hours, allowing academics to perform experiments on larger numbers of servers than was previously possible. He used the example of Animoto, which scaled from 50 to 3500 servers in three days and then scaled back down, using Amazon Web Services. Notably, the economies of scale from existing large infrastructure such as Amazon and Google, and their operational expertise, are cited as the prime reason for the current trend toward cloud computing.

Fox next mentioned challenges and opportunities, with the primary challenge being the incompatibility of different types of cloud: low-level instruction-set VMs such as Amazon EC2 at one end, and Google's framework-based AppEngine at the other. There is scope for vendor lock-in, but also opportunities for the development of open APIs and free and open source software. Other uniquely cloud-based issues were presented: the costs of moving data, including physically shipping it, and the proliferation of non-relational scalable storage such as Cassandra, Hypertable, and Amazon SimpleDB.

Fox discussed deciding whether or not to "cloudify" an application. Authentication and data privacy are paramount when data is placed in a public cloud, and it is unlikely that the weakest link will be technical. For those considering building private clouds, he notes that other than efficient utilization, they are unlikely to be as cost-effective as public clouds. Overheads, such as a billing system, and incentives to release idle resources also need to be addressed.

The role of academics in cloud computing is seen as promising. Fox mentioned UC Berkeley's own 40-node Eucalyptus cluster, with a workload that can overflow onto Amazon EC2, on which they routinely perform experiments on hundreds of servers. He commented on the difficulty of incorporating cloud computing into the funding/grants culture and providing accounting models for cloud usage. Statistical machine learning is an area for future research, as the difficulty of resource optimization increases.

Mark Burgess asked whether weak coupling between cloud components was desirable and whether a change in software writing strategy is needed. Fox agreed that scale independence in software was both necessary and desirable. Alva Couch asked whether low cost would override privacy concerns, with Fox replying that legislation is always behind technology; however, for certain applications a new high-trust cloud might be needed, as the public cloud is out of bounds.

## PLENARY SESSION

- ***Frank Lloyd Wright Was Right***

  *Daniel V. Klein, Consultant*

  *Summarized by Mark Burgess (Mark.Burgess@iu.hio.no)*

Closing sessions at LISA have been varied since the demise of the LISA Game Show, but Dan Klein is known for his snappy and erudite lectures that dance between playful analogy and serious commentary. With an impressively polished script and dazzlingly crisp diction, he didn't disappoint this time in his flawless execution of the closing session.

The talk was subtle but firm in bemoaning a lack of leadership we often exercise in design or repair of the infrastructure around us. "We," he suggested, "are collectively responsible for the messes we make, and by golly we are responsible for some corkers."

Dan likened our efforts to implement computer security, in particular, to the Byzantine ruminations of civil engineering in Pittsburgh and to US tax legislation, as well as to the bizarre Heath-Robinson-like contraptions we build from Web services using today's so-called state-of-the-art technologies. He showed a humorous battery of examples and illustrations of things gone awry.

The talk was "cantilevered" with examples and quotations from visionary architect, philanderer, and borderline fraudster Frank Lloyd Wright, who suggested, like Ayn Rand's fictional Howard Roark, that rather than feign "hypocritical humility" in building the world, "honest arrogance" might be a cleaner approach to infrastructure building. Thus he proposed to "raze it and start over" in the face of mistaken design, rather than keep people in ungainful employment, perpetrating "feeping creaturism" that creates monsters from incessant patching.

Wright proposed to merge form and function seamlessly from the start, to keep design as simple as possible, but no simpler. He posited that "less is only more when more is no good." We build, Dan suggested, some comically inept systems that suffice often more by luck than judgment, and we could take a lesson from Wright and try to build systems that work *with* rather than *against* the environment they live in.

The talk was a fast and festive romp through cultural and technical folly. There were more than a few laughs from the appreciative audience. The undercurrent was essentially this: as engineers, we should exercise more leadership in mending broken designs rather than using plasters and chewing gum to keep them together. Raze it and start over is an underestimated strategy, Dan suggested. We needn't fear it if we phase in major change incrementally, because, as Wright exclaimed, "Belief in a thing makes it happen."

## LISA '09 WORKSHOPS

- ***University Issues***

  *Workshop Organizer: John "Rowan" Littell, California College of the Arts*

  *Summarized by Rowan Littell (rowan@hovenweep.org) and Josh Simon (jss@clock.org)*

The fifth annual University Issues workshop had six participants (down from 15 last year), three from the US, two from Canada, and one from Norway, with institutional sizes from hundreds to tens of thousands of users. The format of the workshop was a semi-structured roundtable discussion.

In the morning session, after brief introductions, we started with a discussion of personnel management issues. Some of us have managers who don't understand technology or can't communicate or don't trust (either their own people or other groups). We had a mix of centralized and decentralized institutions, which affects the management styles; one example was of a central IT group that takes over faculty- and departmental-run technologies.

This segued into a discussion on project management. Smaller groups tend to either not have any or to self-manage projects, and larger institutions tend to have a dedicated project management team. Project management tends to be expensive, since there is a big time and communications commitment in doing it well. There are also issues involved with funding: is project management funded centrally or is the funding grant- or project-based? In general, management, both personnel and project, needs to communicate with the employees or users in order to make the right decisions for the needs of those employees and users.

We next had a brief discussion on virtualization. Most people are using either VMware or Hyper-V. Several places are looking at offering virtualization as a service. This segued into uses for virtualization. Eeducational institutions have liability and security concerns (in the US, the Family Educational Rights and Privacy Act (FERPA) is the big one to worry about); the issues and risks of exposing confidential information are certainly concerns when virtualizing on a third-party vendor. There are also the usual reliability and security concerns. Outsourcing IT to virtualization services may be good for smaller environments.

After the break, we moved from soft topics to more technical ones. We continued discussing virtualization technologies, touching on guest OS and network interactions (particularly network storage) and patch and update management; it was noted that Solaris zones make patching the guest zones at different schedules difficult. The topic of virtualization as a service led to a discussion of how to provide administrator rights to such systems, which and how many operating systems to support, and how to integrate hosted virtual machines with the rest of the department's computing resources.

The discussion then moved into various security concerns. Phishing scams and password management were noted as prominent concerns. Some institutions have implemented outbound rate limiting on email and scanning messages for known passwords. Password policies, including change frequency and complexity, were also discussed. Some attendees reported having to deal with students or other parties who have captured and cached passwords, usually for non-malicious purposes such as creating services that let people log in to multiple Internet services at once.

The workshop ended by moving back to a discussion of money. American institutions have, of course, been struggling under the current economy, foreign ones less so (with the Norwegian representative noting that students do not pay tuition per se and funding comes from other state sources). Money is easier to access for hardware purchases than for personnel, raises, or training, although some have changed how they budget for hardware (e.g., leasing machines so that the cost is part of the operations budget rather than the capital budget).

Although the workshop closed at lunch with a half-day format, all attendees agreed that it had been beneficial. There was a short discussion about the small size of the workshop: people agreed that a larger attendance would have been better but that the small group allowed for good discussions.

- ***Government and Military System Administration Workshop***

  *Workshop Organizer: Andrew Seely*

  *Summarized by Andrew Seely (seelya@saic.com)*

The Government and Military System Administration Workshop was attended by representatives from the Department of Defense, Department of Energy, NASA, Food and Drug Administration, National Oceanic and Atmospheric Administration, Raytheon, and Science Applications International Corporation. This was the second year the Gov/Mil workshop has been held at LISA.

The workshop concept was to create a forum to discuss common challenges, problems, solutions, and information unique to the government sector, where participants would be able to gain and share insight into the broad range of government system administration requirements. LISA was an opportunity for diverse government and military organizations to come together in a unique forum; it's not common to have highly technical staff from DoD, DoE, FDA, NASA, NOAA, and industry at the same table to candidly discuss everything from large datasets to organizational complexity. All expected to find similarities and hoped to be exposed to new ideas, and no one went away disappointed. The day's specific agenda was developed in the weeks leading up to the workshop, with each attendee identifying specific topics for the workshop to address. The agenda was adjusted as the workshop progressed in order to capture emergent issues.

The day started with roundtable introductions and a reminder that the environment was not appropriate for classified or sensitive topics. For system administrators outside the government sector this could seem like an unusual caveat, but for people who work in classified environments it is always a good idea to state what the appropriate level of discussion is for any new situation, especially when the discussion is about government systems and capabilities. The group agreed that the day would be strictly UNCLASSIFIED and that no For Official Use Only or higher material would be discussed.

The day was loosely divided between technical and organizational topics. Technical topics discussed included PKI and identity management systems, integrating authorization and authentication systems, and managing large datasets. More detailed and wide-ranging this year were topics centering on policy, procedure, and organizational structure, with heavy focus on industrial security and government/military security policy implementation. Detailed discussions on certification and accreditation highlighted the surprising differences between government agencies.

All attendees presented what types of personnel their respective sites or companies are seeking to hire. Over half had positions to fill, and almost all required security clearances. DoE and DoD were generally hiring, while FDA, NASA, and NOAA were not. Hiring information and career Web sites were shared.

The final topic of discussion was to determine if there would be sufficient interest in this workshop to repeat it at LISA '10. It was agreed that it was a valuable experience for all attendees and that all would support a follow-on workshop. A Gov/Mil wiki was established at http://gov-mil.sonador.com/ to provide a collaboration area to help develop this workshop into a viable community of interest.

- ***Advanced Topics***

  *Workshop Organizer: Adam Moskowitz*

  *Summarized by Josh Simon (jss@clock.org)*

Adam Moskowitz was the host, moderator, and referee of the Advanced Topics Workshop once again. We started with our usual administrative announcements and the overview of the moderation software for the three new attendees. Then, we went around the room and did introductions. In representation, businesses (including consultants) only outnumbered universities by about 2 to 1 (down from 4 to 1); over the course of the day, the room included six LISA program chairs (past, present, and future, up from five last year) and nine past or present members of the USENIX, SAGE, or LOPSA Boards (the same as last year).

Our first topic was cloud computing. We discussed the various takes on it, and one of the clearest issues is one of definition: Technologists and non-technical end-users have different definitions of what it means. Comparisons to grid computing were made; the consensus was that grid is for high performance computing (HPC), cloud computing isn't, and that grid and cloud are solving different problems. When discussing cloud computing you need to determine if your definition includes the server/OS (be it physical or virtual), the applications, or the data itself. Then there's the issue of the data: who owns it, maintains it, backs it up, is responsible for restores as needed, and deletes it when you're done with it. So far, in general, cloud computing is good in that you and your company can save money on hardware and possibly on support (licensing, maintenance, and staff) costs, but so far we've ignored the security aspect. Contracts are all well and good, but there are legal and regulatory and security issues regarding certain types of data (student records, health records, personally identifying information, access control for research data, and so on) that make it a bad idea for some environments, industries (health and financial), and applications. How do you audit your cloud provider?

Next we did a lightning round of cool new-to-individuals tools or technologies. The most common response was a programming language (Erlang, Python, and Ruby); others were Bugzilla, iPhone, memcache, nfswatch, RRDtool, XMPP for system-to-system messaging, and ZFS. One person mentioned his new Viking 6-burner range.

After our morning break, we resumed with a discussion of file systems. Some are looking for alternatives to NFS that scale better; most seemed to like GPFS, and others mentioned OCFS2 and GFS2. In all cases, you need to look at your requirements to find the one that best suits your needs; for example, OCFS2 doesn't scale beyond 7 or 8 nodes in a cluster of virtual machines, but if you only have 3 or 4 it might be sufficient. This segued into a distribution discussion regarding what needs to be local and what can be remote, as well as what needs to be read-write (more expensive) versus read-only. From there we segued into charge-back. Can you charge back to other departments or users the cost of your file services (and, indeed, other services), and if so, how? Most people are looking at tiered models, such as "dumb SATA is free; if you want RAID or backups it costs more." The problem is that end-users can add cheap disk to their systems and not see the difference between disk (the physical device and its data) and storage (the infrastructure for availability, retention, and recovery). Some folks are charging back what they can even though it's not enough to cover the hardware costs, let alone the staff costs. It was stressed that you have to proportionally reflect your costs or the users will game the system, and you have to be careful not to oversubscribe.

Our next major discussion topic was career paths. Management is still the most common option career path for ever more senior people. In education, it's pretty much the only option, as you have to become a manager to grow into any CTO/CIO/Dean/Provost roles. In industry there's no well-defined career path; there's junior to intermediate to senior, but then it can tend toward either management or architecture/design. One possibility is "minister without portfolio," where you're known internally as a senior resource and various departments bring you the hard problems for advice if not outright solution, and otherwise you just do what needs doing. Some noted that manager-or-techie may be the wrong view. Leadership is the issue: does your organization provide a way to foster and grow leadership skills? It seems that "architect" is the "leader who isn't a manager" title. In addition to growth, the concept of job satisfaction came up. Some are satisfied more by title, some by compensation (salary or benefits), some by growth, and some by having interesting problems to solve. Where are you on that scale, and can your current organization satisfy you? If not, it may be time to find one that can.

After our lunch break, we had a discussion on automation. We talked about some of the differences between host and network based configuration tools, and how at the baseline you need to get a set of consistent configuration files for all the devices at a given point in time. The next problems are to get that configuration information to those devices, then move from that set to another set from a different point in time. Do you keep the configuration data and metadata and state all in the same system or not? Are the tools topology-aware or not? We should move away from the procedural specification and more toward a declarative mode (e.g., "build a VPN between point A and point B"), letting the tools figure out the "right" way to do it for your environment. Abstracting up to a declarative level will be helpful in the long run but getting there is going to be challenging. The mental model for automation sits at the intersection of "how people think about their systems" and "what data the tool provides" or "what function the tool performs."

We next had a quick survey on the hot new technologies or big things to worry about in the coming year. Answers included automating failover and self-healing automation; changing the way people think; chargeback and resource allocation; cloud computing; finding a new job, getting out of the rut, having new challenges; getting useful metrics; outsourcing; politics at work; and rebuilding community and improving communications between IT and their users.

Our next topic was communications, both between technical and nontechnical (be they business or faculty as relevant) and between groups within an organization. Having an advocate for IT in the remote business group has been helpful for some people; holding tours of the datacenters for non-technical users has helped others. Empowering users to help themselves, such as with self-service Web sites or kiosks, helps as well. To get IT recognized as helping the business accomplish its goals and not as obstructions or obstacles, IT has to understand those business goals better. It's not that IT should say "No," but, rather, "Here's a better way to accomplish that" or "Here's what I need before I can say yes." Technical people need to remember that just because someone isn't technical doesn't mean they're stupid. One additional note is that, like nurses, we often see people on the worst day of their lives: something is broken, they have a deadline, and we have to fix what's wrong so they can get on with it.

After the afternoon break, we resumed with a discussion on mobility. Laptops and mobile phones are commodities now, so what policies do people have for managing them? There was a reminder that if the policy is too complicated, then it'll just be ignored or worked around. Most places have the management of laptops (both corporate and visitor) controlled by now, but handhelds are a newer problem. In general, the policy needs to scope what is and isn't allowed and to focus on what is and isn't within the control of the people enforcing it. This completely ignores the supportability aspects. Are VPNs the answer? DMZs for unauthenticated devices? As with everything else, "it depends."

The security issues involved in managing mobile devices segued into a discussion of identity management; it seems that many people are falling for phishing despite education, outreach, and announcements. Several have implemented email filters to look for personally identifying information in outbound email to try to prevent account compromises.

Security in general is about the same as a year ago (one person said, "It's better now"). It's still often an afterthought for infrastructure projects. We tried to brainstorm on how to get people to incorporate security. You need management buy-in and to change the culture, whether it's for regulatory reasons or not. It helps if there are policies to point to and guidelines to follow. Security is a method or a process, not a result. It does get better when more people understand *and* follow it.

Next we discussed our preferred scripting languages. Perl, Python, and Ruby continue to be the big winners, with shell scripting (and VBscript for those on Windows) trailing behind. Others include Erlang and Haskell.

We next discussed outsourcing. There's been a rise in many places of the percentage of finance-based managers who don't understand engineering or information technology. Outsourcing is only good in those cookbook situations where there's easily identified cause and effect, and specific tasks to accomplish. Companies don't think they're giving up control when they outsource. There are two different kinds of outsourcing: ongoing operations, where automating may be better (but since that's hard, it's next-better to implement it via API as request ticket to your outsourcing company), and project-based, where a particular project is given to the outsourcing company. However, it should be pointed out that in India, the average time-in-post is only three months, so a one-year project means you'll have four different project teams on average. That gives you lower quality, and going through a non-technical project manager gives you even less control over the implementation.

Companies are good at looking at the next quarter but not at long-term costs. One trick mentioned is to realize that the outsourcing company has limited bandwidth which you can fill up with less important projects, showing yourself to be willing and getting goodwill, even though you're keeping the more important projects in-house; and it's good to use some metrics to show how excellent you are at those in-house projects. Finally, we recommended that you keep your core business in-house; anything else is asking for trouble.

Our last discussion was a lightning round about the biggest technical or work-related surprises this past year. Some of the surprises included company closures despite profit and growth, company relocations, retiring executive management and changes in management or reporting structures, and responsibility changes.

Due to software issues, this year's Talkies Award could not be awarded. Last year's winner was D.J. Gregor, but he was not present this year.