conference reports

THANKS TO OUR SUMMARIZERS

LISA '08: 22nd Large Installation System Administration Conference

San Diego, CA
November 9–14, 2008

Summarized by Rik Farrow

Mario Obejas led off with thanks to the program committee members and USENIX staff for putting together another successful LISA. Then the SAGE award was given to the SAMBA group for its work on interoperability. SAMBA Team member (and USENIX Board Member) Jerry Carter accepted the award. The Chuck Yerkes award was given to Dustin Puryear for his helpful posts to sage-members.

The Best Student Paper award went to Xiaoning Ding of Ohio State University, Hai Huang, Yaoping Ruan, and Anees Shaikh of IBM T.J. Watson Research Center, and Xiaodong Zhang of The Ohio State University for "Automatic Software Fault Diagnosis by Exploiting Application Signatures." The Best Paper award went to Qi Liao, Andrew Blaich, Aaron Striegel, and Douglas Thain of the University of Notre Dame for "ENAVis: Enterprise Network Activities Visualization."

KEYNOTE ADDRESS

- ***Implementing Intellipedia Within a "Need to Know" Culture***
  Sean Dennehy, Chief of Intellipedia Development, Directorate of Intelligence, U.S. Central Intelligence Agency

Summarized by Andrew Hoblitzell (ahoblitz@cs.iupui.edu)

Sean Dennehy discussed technical and cultural challenges being introduced by the introduction of Web 2.0 tools in the United States intelligence community. Dennehy traced the start of the movement to Dr. Calvin Andrus's publication of his 2004 paper "The Wiki and the Blog: Toward a Complex Adaptive Intelligence Community" and showed the development of various Web 2.0 tools in the intelligence community up to the current day. The tools are used by intelligence analysts from 16 intelligence agencies, as well as other parts of government.

Dennehy began his presentation by showing a document from the Office of Strategic Services, the predecessor to the Central Intelligence Agency. The document described many of the characteristics of a typical bureaucracy, such as deferring decisions because of worries about management issues, conflicts about the specific wording of things, etc. Dennehy revealed that this document was produced by the Office of Strategic Services as a guide to sabotaging an organization. Dennehy proposed that Web 2.0 tools were a solution to these common problems.

In the intelligence community, information sharing is especially difficult, because information is typically available on a "need to know" basis. The information is classified to different levels, usually Controlled Unclassified Information, Secret, and Top Secret. Information may also be "compartmentalized" to further restrict its access. Two U.S. networks for transferring this information are SIPRNet for secret information and JWICS for top secret information. In contradiction to the common tenets of the intelligence community, Dennehy said, Intellipedia would aim information at the broadest audience possible. In this spirit, Dennehy said that the approach has been shared with allies and the media.

Dennehy pointed out a number of differences between Intellipedia and public Web 2.0 tools such as Wikipedia. Dennehy pointed out that although edits in Wikipedia may be anonymous, every edit to Intellipedia must be attributable. Other differences pointed out by Dennehy included that Intellipedia need not be encyclopedic and that Intellipedia only uses attributable point of view instead of Wikipedia's neutral point of view (NPOV). Dennehy said that these differences allowed Intellipedia to encourage professionalism, prevent anonymous edits, and encourage collaborative disagreement. Dennehy said the government's Web 2.0 tool set system had been further enhanced with a YouTube-like video channel, a Flickr-like photo-sharing feature, content tagging, blogs, RSS feeds, and many other Web 2.0 tools.

Dennehy noted that there would never be a "one stop shop" for the intelligence community and that Intellipedia was still in the early stages of development. Dennehy said that, despite this, Intellipedia would act as a powerful way to connect people. He said that people acting selfishly could in the end help each other and that Intellipedia would help connect analysts to information that they might not have been able to find otherwise.

Dennehy said that intelligence workers had started using Intellipedia as a replacement for their email and telephone conversations and are sharing it with each other in their personal blog postings. Dennehy said that Intellipedia would catch on because it has strong support from the top, including from some in the Joint Chiefs of Staff, but that it would ultimately have to become a self-sufficient grassroots movement kept active from all levels of the intelligence community.

In response to questions from the audience, Dennehy stated that old or dead data could automatically be tagged in the system after a given timeframe by using bots or similar technology. Dennehy also stated in response to a question from Dan Klein that he has seen more interagency cooperation as a result of the wikis. He noted that the intelligence community would have a strong interest in utilizing and adapting software that is currently used for Wikipedia and other Web 2.0 tools.

THINK ABOUT IT (META-ADMIN AND THEORY)

Summarized by Alex Boster (aboster@machineepsilon.com)

- ***Designing Tools for System Administrators: An Empirical Test of the Integrated User Satisfaction Model***
  Nicole F. Velasquez, Suzanne Weisband, and Alexandra Durcikova, University of Arizona

Nicole Velasquez explained that the motivation for this work came from a usability study at IBM done on system administration tools that yielded negative feedback from their target users. The conclusion of that work was that the tools were "built for grandma," not system administrators. A new, hybrid theoretical model, the Integrated User Satisfaction (IUS) model, was developed, combining the strengths of TAM and IS Success, older usability models. In this study, a Web-based survey was taken of system administrators, without respect to the specific type of administration they performed, using verified methodologies. Then this new usability model was tested using the survey results with various statistical methods to validate the model. The IUS model was validated by the survey.

Nonintuitively, the IUS model suggests that for system administration tools that use a GUI, currency (up-to-the-second information) and speed were not significant factors in final user attitude, despite the fact that the survey suggested that users thought they were important traits for a tool to have. Accessibility was not important—the tool users were all power users. Scalability did not factor in, since the tools in actual use were already selected by their users as being appropriate for the scale of their systems. Scriptability stood out as an important factor that is obvious to sysadmins, but less so to management and tool authors. The IUS model may be a useful way to validate and explain to management what makes one tool better than another. Known limitations of the study include all the issues involved in a voluntary response survey conducted at one point in time.

A questioner asked about the effect of already efficient tools being surveyed. Only tools that had been chosen for use were in the survey, so self-selection was a factor. Another questioner asked whether documentation was studied—it was not.

- ***Dynamic Dependencies and Performance Improvement***
  Marc Chiarini and Alva Couch, Tufts University

Marc Chiarini stated that traditional methods of performance improvement analysis are costly, so mostly ad hoc methods based on experience and intuition are used. These methods are not always reliable or generalizable. The authors' approach to performance improvement analysis used an exterior model, a technique that treats the system as a black box and is guided by analysis of changes in externally observable statistics as a probe load is varied. The authors factor the system into components (e.g., Web server, disk, CPU, memory, file server, etc.) under test and synthesize so-called steady-state behavior. This could be thought of as a typical "nonloaded" state. It is noteworthy that production systems are seldom in steady state, as factors such as time of day and day of the week become important.

In the dependency model, different servers are competing for the same finite resources. A resource is deemed critical if reductions in its availability affect the performance of some system component. Micro Resource Saturation Tests (mRST) are run in a steady-state environment. A resource is chosen to be made less available in various increments, then response times are monitored to see whether any go "off the charts." Statistical methods are used more rigorously to test the significance of the test results. It was noted that, to make these tests significant, large enough sample sizes must be used. Also, each probe of the system must be independent of the others; this is accomplished, for example, by spacing the probes a bit.

In conclusion, the speaker stated that this model allows us to reliably improve system performance without knowing too much detail—it's a guided approach that can help systems administrators know what to expand first. A questioner asked how to do this routinely. The speaker said that more work is needed, including tool development. Other questions focused on the particular statistical methods discussed in the paper.

- ***Automatic Software Fault Diagnosis by Exploiting Application Signatures***
  Xiaoning Ding, The Ohio State University; Hai Huang, Yaoping Ruan, and Anees Shaikh, IBM T.J. Watson Research Center; Xiaodong Zhang, The Ohio State University

  Awarded Best Student Paper!

Xiaoning Ding stated that the authors' goal was to create a black-box method to automatically diagnose application faults such as bugs and misconfiguration. It was noted that misconfiguration is far more common than bugs in production systems. The basic approach is to record the normal execution of an application into a trace. Over time, a series of normal traces becomes an application signature, which might be roughly thought of as a set union of normal traces.

Run-time attributes such as system call signatures, signals, and environment variables are used. This means there is no need to instrument the application. The collection of observed values is recorded. System calls provided a challenge. A graph is made of the system calls plus their call stacks. That way, repeated calls from the same point in the code are merged to one node in the graph. When a fault occurs, the trace of the fault is compared to the signature and mismatched attributes are flagged as potential root causes. Causes closer to the head of the call trace graph are given priority, as they are more likely to be root causes. For example, intermittently failing httpd daemons were diagnosed and the correct root cause (log files close to 2 GB in an older filesystem) was automatically diagnosed.

Experiments on real-world problems found from online sources such as Bugzilla and various forums, including CVS, Apache, and PostgreSQL, were used to test this technique. The experiments worked well except in the case of hardware faults. Performance overhead was initially pretty high, as much as 30%, but after optimization the overhead could be brought as low as 2%. A questioner asked about high-noise situations, such as several high-priority, possible root causes. Lots of traces are needed to help weed out the noise.

INVITED TALK

- ***Integrating Linux (and UNIX and Mac) Identity Management in Microsoft Active Directory***
  Mike Patnode, Centrify

  Summarized by Will Nowak (wan@ccs.neu.edu)

Mike Patnode, the Vice President of Technology of Centrify Corporation, gave a state of the world overview of integrating UNIX-based machines into an Active Directory environment.

Patnode started by discussing the open source and free solutions available. Generally these open/free tools were regarded as difficult to integrate, hard to find support for, and not always the right solution. Microsoft Services for UNIX (SFU) were discussed, although it was noted that these tools are limited in scope. Microsoft does provide support, but customers using SFU may find themselves wanting more. In addition to discussing various tools for solving integration issues, Patnode painted a picture of a typical organization's mess of authentication issues—local accounts, mixed directory systems, and no synchronization anywhere.

Patnode drew several conclusions. Integration is a good thing, and organizations should strive to reach this goal. User administration becomes easier, users are happier, and the all-important auditors are happier. Picking a tool or product for the job is difficult; it is up to the organization to decide whether it wants a commercially supported full-featured system such as Centrify's products or whether it is more comfortable tooling together something with freely available options. Patnode's presentation helped put the audience in the right mindset for making these important choices.

INVITED TALK

- ***Programming the Virtual Infrastructure***
  Paul Anderson, University of Edinburgh

  Summarized by Patrick Ntawuyamara (ntawuyamarp@hotmail.com)

Paul Anderson's presentation was about the complexity of configuring the virtual infrastructure. He identified some similarities and analogies with his experience programming early computers.

Anderson described how programming has evolved and how some of the problems and solutions mirrored those of system configuration. He emphasized that the virtual infrastructure does not fundamentally change the nature of the configuration problem but increases its complexity.

Referring to John Backus (developer of Fortran) and the early programmers, he pointed out that the properties of efficiency, correctness, portability, and usability that were important in the early days of programming are still relevant for the virtual infrastructure.

He stressed that programming languages have typically taken many years to become accepted practice and only survive when they can prove themselves practically. The same is likely to be true for system configuration tools such as LCFG, Cfengine, and Bcfg2.

He also discussed some different programming paradigms and pointed out features such as agile programming that have interesting analogies in the configuration context.

Declarative specifications have become the norm for system configuration, but the virtual infrastructure requires a different kind of approach to deal with its complexity. Fully automatic configuration within this framework will be difficult to achieve. Paul referred to the I-X framework, which supports a combination of human and automatic processes to achieve its goal, and suggested that this type of hybrid configuration may be a way forward for configuring complex virtual infrastructures.

The question still remains how best to coordinate efforts to find the language design that will deal with these complexities.

For an article based on the talk, see page 20 of this issue of *;login:*.

Paul Anderson's complete presentation can be read at http://homepages.inf.ed.ac.uk/dcspaul/publications/lisa-2008-talk.pdf.

LARGE-ISH INFRASTRUCTURE

Summarized by Matthew Sacks (matthew@matthewsacks.com)

- ***Petascale System Management Experiences***
  Narayan Desai, Rick Bradshaw, Cory Lueninghoener, Andrew Cherry, Susan Coghlan, and William Scullin, Argonne National Laboratory

In this talk, representatives from the Argonne National Laboratory spoke of some of their troubles and successes administering the Intrepid supercomputing cluster. Intrepid is an IBM Blue Gene/P system with proprietary job control and monitoring systems, which the Argonne engineering team describes to be at odds with the commodity cluster designs.

In the Argonne implementation the service infrastructure required much more manual servicing than the Blue Gene/P control system and the IBM Reliability, Availability, and Serviceability (RAS) infrastructure. The Argonne team concludes that more HPC systems will need to move to a more self-contained, self-healing systems such as the Intrepid system at Argonne.

- ***Rapid Parallel Systems Deployment: Techniques for Overnight Clustering***
  Donna Cumberland, Randy Herban, Rick Irvine, Michael Shuey, and Mathieu Luisier, Purdue University

The Purdue computing team presented their paper on how they were able to deploy the 7000-core "Steele" cluster in under 24 hours. Achieving this undertaking was a monstrous task, but the team was able to overcome most of the difficulty in meeting the time limit with manpower for unboxing and racking machines. Why would anyone want to or need to deploy a supercomputing cluster in under a day? Because research staff members want to be able to use the cluster as soon as possible to meet their deadlines, and the hardware is losing operational lifespan and warranty lifetime sitting in the box. It took a total of 190 volunteers just for the unboxing, racking, and waste-disposal effort. The physical deployment team started at 8 a.m. and was done by 1:30 p.m.

A small system administration team of four people, using extremely creative modifications to standard deployment methods, performed the nonphysical initialization of the computer cluster in under 15 hours. All hosts are installed using PXE and Red Hat kickstart; however, when kickstarting the immense volume of machines all at once, the team anticipated difficulties in setting up kickstart configuration files as well as network contention. This was addressed by using IPVS (IP Virtual Server) for load-balancing kickstart file-serving and a simplified kickstart configuration by using a single IP for the cluster.

Through the Purdue team's creative efforts, they were able to prove that even a small IT group can rapidly deploy large systems with the help of a large volunteer team to power through the manual labor portion of a large deployment.

- ***ENAVis: Enterprise Network Activities Visualization***
  Qi Liao, Andrew Blaich, Aaron Striegel, and Douglas Thain, University of Notre Dame

  Awarded Best Paper!

Qi Liao presented the ENAVis network visualization project. The ENAVis model addresses the issues that most conventional network monitoring and visualization systems lack, which is correlating data in three context rings: host, user, and application. Liao said a convention for such a model is the HUA model (host, user, application). ENAVis uses agents for data collection to overcome the limitation of using only network logs or SNMP for gathering and trending network activity. The amazing thing about the ENAVis application is that the agent is a simple bash script using commonly available commands, so it is flexible in terms of the systems it can collect data from. Liao argues that point-to-point (IP address and port) netflow-type data is not useful for monitoring and visualizing network activity, which holds much weight considering applications oftentimes do not run on standard ports. Applications, hosts, and users are all taken into account, giving a more accurate view of network activity. Data is then correlated and visualized on a central server.

Liao demoed some of the abilities of ENAVis by showing a file-sharing user and which applications he was using on the university network as he utilized a large amount of bandwidth. ENAVis introduces a more modern, thorough approach to trending network data and encourages better correlation of data to provide more accurate network monitoring. The code for ENAVis is freely available.

INVITED TALK

- ***How to Proceed When 1000 Call Agents Tell You, "My Computer Is Slow": Creating a User Experience Monitoring System***
  Tobias Oetiker, OETIKER+PARTNER AG

  Summarized by Marc Chiarini (marc.chairini@tufts.edu)

A few years ago Tobias Oetiker was called by some IT staff over at Swisscom, who asked if he would help them solve perceived performance problems with their CRM software. As Oetiker jokes, they were apparently laboring under the false assumption that he was an expert in this area. Nonetheless, he decided to take up the challenge and learned quite a few lessons in the process that he graciously shared with us in this invited talk.

Swisscom has a very complex Windows IT environment with a large call center where customer service agents deal with all manner of problems and requests. Agents were intermittently experiencing slowdowns, but no one on the IT staff was capable of pinpointing the cause. Using off-the-shelf products (sysinternals tools, winspy, various Perl modules, etc.), Oetiker initially created a three-part system called CPV to get to the root of the matter. CPV comprises a passive monitoring component (monitor), an interactive reporting component (reporter), and a data analysis component (explorer).

CPV monitor feeds filtered events (mouse clicks, log entries, network events) from each client in the call center (initially kept small for buy-in) into FSMs (finite state machines) that mimic the behavior of the call center software and look for invalid sequences of events. CPV reporter is a small tray tool that allows the agent to send a wealth of system information (crash logs, configuration, etc.) along with a message to the IT staff *when* the problem is occurring. CPV explorer allows the staff to perform in-depth visual and statistical analyses of the data sent from any combination of monitors over any time period.

Oetiker's first lesson was that the complexity of FSMs has the potential to grow quickly and without bound, making them slow and impractical. Over the course of building CPV and looking for an alternative, Oetiker investigated ways in which observability at the client could be increased. Further lessons ensued, including:

1. They determined why agents could sometimes not press the modal "send" popup button in Outlook. The CRM software used an outdated aspect of a Windows system call that preemptively stole the key state. The problem was solved with Perl.
2. Switching from the GetWindowText call to WMGetText in CPV monitor caused severe slowdowns because of undocumented behavior of the latter system call.
3. Once the monitor was running on ~1500 clients and other devices, too much data was produced (100 million samples in 40 days) and the central database used for analysis was unable to keep the entire DB index in RAM (4 GB). The solution was to perform index compaction, enabling analysis of up to 7 years of data at one time.
4. Threaded Perl on win32 does a full environment copy whenever a new thread is created, utilizing a large amount of memory. The solution was to only use threads where necessary.
5. Agents and IT staff were interested in how long it took to boot a system and then log in. When agents were on the line with a customer, they wanted to know how long they would need to stall before their system came back after a crash. Oetiker used the WMI (Windows Management Instrumentation) interface to identify the contributors to boot and login time.
6. Detecting crashes and (truly) hung applications does not work in the same way as with UNIX. In the first case, the monitor needed to be built to find the active window, attach via a process handle, and continuously poll for the appropriate exit code. In the case of hangs, the active window was identified and messages were sent to it until it returned. All this was necessary to distinguish between real crashes and users closing the CRM software or CPV monitor out of frustration.

Oetiker states that the learning process with CPV is ongoing. Some of the positives are that the reporter makes agents feel useful, the explorer gives effective access to previously hidden data, there is a closed feedback loop, and IT staff now have efficient tools to perform structured problem solving. Some of the stickier issues remain: What or who is being monitored? How do agents change their behavior? How does problem diagnosis move forward in the absence of previously available information?

INVITED TALK

- ***How to Stop Hating MySQL: Fixing Common Mistakes and Myths***
  Sheeri K. Cabral, The Pythian Group

  Summarized by Will Nowak (wan@ccs.neu.edu)

Sheeri Cabral presented a question: Why do we hate MySQL? Over the course of her in-depth examination of MySQL performance and configuration issues, she highlighted tips that should make any database administrator's life easier.

Cabral started with an overview of myths and rumors: Don't use ENUM, MySQL is slow, it uses too much memory, etc. For nearly all of these myths, Sheeri gave a simple and easy fix for the problem. One cool tip was to store IP addresses using INET_ATON and INET_NTOA. These two MySQL built-in functions convert a string representation of an IP address into an unsigned integer. This solves a performance issue with storing variable-length strings in a database. Throughout the course of the presentation numerous little hints like this were provided. More than just SQL, machine tuning was also covered, as were common issues with deciding whether or not to use RAID in a database server—why bother if you already have 10 replicas set up? Other software-independent issues were touched upon, such as network bottlenecks with multiple queries that could be reduced. A concrete example was given: Most people select from a database to see whether an ID exists before creating it, but you can combine those two operations into a single INSERT...ON DUPLICATE KEY UPDATE query, cutting the number of database connections in half.

TRUST AND OTHER SECURITY MATTERS

Summarized by Alex Boster (aboster@machineepsilon.com)

- ***Fast, Cheap, and in Control: Towards Pain-Free Security!***
  Sandeep Bhatt, Cat Okita, and Prasad Rao, Hewlett-Packard

Firewall ruleset analysis may yield significant gains in operational efficiency, Cat Okita said. Currently, most firewall rules are implemented once and never removed. There is a cargo-cult mentality because changes can have unpredictable effects. The authors' approach is to analyze the structural properties of security rulesets, define a standard grammar for ruleset actions, develop a set of ideal conditions for rulesets, and define metrics to measure the differences between rulesets.

Ideally, rules exhibit noninterference, simplicity, and consistency. The authors define noninterfering rules as those that do not interact with other rules. Simplicity posits that only triggered rules are defined, all defined objects are used, and rules match policy. Consistency requires objects to have consistent names. A new metric, effectiveness, is defined as a measure of the fraction of a given rule or object that is not interfered with. For example, if "permit ssh, telnet" is followed by "deny ssh," the second rule is less effective as it is partially obscured by the first rule. A tool has been written in Java 1.5 with CLI and Web interfaces, with Checkpoint devices fully supported and Cisco PIX/ASA, pf, and ipfilter in the proof-of-concept phase.

Several use cases have been explored. The tool can be used to compare differences in configurations, for example, for migration planning and verification, pre-change impact verification, post-change confirmation, and incident analysis. Ruleset remediation such as identification and removal of ineffective rules, resolution of rule conflicts, and improved accuracy of rule sets is possible. And the tool can be used for reporting.

The authors have made a number of discoveries. More rules lead to more issues. Humans are effective at finding completely ineffective rules, but they are bad at finding partially ineffective ones. Ruleset effectiveness declines over time. People clean up rules, but not objects.

A questioner asked about typical effectiveness ratings. These are 50% or more. Another questioner asked whether unused rules, like dead code, aren't really harmless. Compliers strip dead code but firewalls are still slowed down by ineffective rules.

- ***Concord: A Secure Mobile Data Authorization Framework for Regulatory Compliance***
  Gautam Singaraju and Brent Hoon Kang, University of North Carolina at Charlotte

Gautam Singaraju began by saying that regulatory compliance standards (RCS), such as the Feinstein bill and HIPPA, stipulate the safeguarding of data confidentiality and the reporting of incidents. In complex corporate environments threats come from a number of sources: servers with bad configurations, unpatched servers, spyware, and malware. Mobile devices have most of these issues, plus they can be lost or stolen. Furthermore, internal users are the largest threat, with 60% of attacks coming from disgruntled employees and other insiders.

The Concord framework addresses these issues by assigning trust to a collective interaction of system components, not just one component. Clients, including mobile clients, must get authorization before accessing even local data. Such access requires both user and system authorization. Such authorization can be revoked upon loss of a mobile device. In addition, accesses are logged, so organizations can discover what assets were compromised after an incident.

The prototype implementation is in Java, using Java ME for mobile platforms, and uses mRSA for encryption. Preliminary results yield a 7.5-second access and decryption time for a client with a mobile, disconnected enforcer device, but only 0.5 seconds for a connected agent. A questioner asked about real-life use cases, given the high overhead. This was acknowledged, but the system should be used when dealing with highly sensitive data such as social security numbers. A questioner asked about backups. Old keys must be backed up along with data—there is more work to be done in that realm.

- ***Authentication on Untrusted Remote Hosts with Public-Key Sudo***
  Matthew Burnside, Mack Lu, and Angelos Keromytis, Columbia University

Matthew Burnside explained that this paper was motivated by the following scenario: Consider logging in with ssh to a remote computer, then using sudo there. The traffic is encrypted, but if the target machine is compromised, the password has been revealed to the attacker. To address this, the authors have created SudoPK.

SSH provides agent-forwarding to allow chained ssh logins. The authors have built a generic interface to SSH-USER-AUTH that leverages agent-forwarding. Their authentication module makes running sudo like hopping to another ssh host. The password stays in the encrypted tunnel. This provides for ease of use and is no worse than plain agent-forwarding.

A questioner asked whether this has been submitted back to OpenBSD. It has been, but the status is unknown. Why is this preferable to putting someone's public key in root's authorized key file? Because sudo's grant/deny scheme is richer and more fine-grained. A questioner commented that this was like Kerberos. The speaker said that it was, but it is much simpler and therefore perhaps better suited to small networks.

INVITED TALK

- ***Does Your House Have Lions? Controlling for the Risk from Trusted Insiders***
  Marcel Simon, Medco Health Solutions

No summary available: See www.usenix.org/lisa08/tech/ for presentation slides and live streaming of the talk.

INVITED TALK

- ***Spine: Automating Systems Configuration and Management***
  Rafi Khardalian, Ticketmaster

  Summarized by Marc Chiarini (marc.chairini@tufts.edu)

Rafi Khardalian presented Spine, Ticketmaster's homegrown system configuration management framework. Ticketmaster's worldwide IT infrastructure comprises roughly 4000 RPM-based Linux servers running different combinations of distributed and high-availability applications. Multiple small (3- to 7-person) system engineering subteams are responsible for managing over 100 different configurations, each with subtle variations. When setting out to create a useful tool, the Ticketmaster team had several goals in mind: the ability to define a hierarchical configuration that allowed for overrides from least to most specific; the ability to leverage existing tools with a small, straightforward, modular, and pluggable codebase; a simple management interface using text files; the need for admins to only understand simple programming constructs (conditions, flow control, etc.); the minimization of configuration data duplication; and, finally, versioning and rollback functionality. From the details of the invited talk, we can surmise that these goals have largely been met.

Spine is designed to be invoked on a regular basis after machines have already been provisioned (using another homegrown tool called Provision). It has two primary runtime phases, each consisting of several subphases. The first phase is configuration parsing, which includes discovery, hierarchy traversal, and key/value processing. Discovery consists of collecting network information, identifying the team (or business unit) that will manage the server, group type (e.g., prod, dev, qa), product, system class (e.g., proxy, app, cache), and the particular class instance. A single host can also be uniquely configured in circumstances where one-offs are unavoidable. Hierarchy traversal descends a class hierarchy that is represented naturally via a filesystem. Each directory in the hierarchy is considered a "node" and must contain at least two subdirectories, config/ and overlay/, which are allowed to be empty. In the key processing subphase, files found in the config/ directory are examined. The filename serves as the key, and the lines in the file represent values (or operators, such as <regexp> to remove all entries matching the regexp from a key). Values associated with nodes higher in the tree can always be overridden by those in lower nodes. To further improve the reusability of configuration data, other config nodes can be "grafted" into the tree via a config_groups/ directory at any point in the hierarchy.

The configuration application phase is the real workhorse of Spine. All configuration is applied via plug-ins, which allow for extensions to Spine without modifying the core code. The first step is to populate a temporary staging directory with all necessary configuration files and permissions relative to root. Then, templates written in Template Toolkit can apply the necessary tweaks according to the configuration hierarchy. Once the staging directory is ready, the overlay structure is applied directly to the node's root filesystem. The next steps install and update necessary RPM packages, remove packages not explicitly listed in the configuration (and dependencies), restart or execute commands based on configuration changes, enable or disable services, configure the default boot kernel, and apply basic security best practices.

To provide versioning and rollback, Spine also provides a *publisher* component that can pull an entire configuration "release" out of a SVN repository into an ISO image. Publisher can also request "configballs" via HTTP for either a previous release or the latest update. This approach is extremely flexible and allows for regular changes to be rolled out much like software upgrades.

On balance, Spine has been serving Ticketmaster very well. Some planned improvements include making the core more generic, adding a wider variety of package management substrates (currently only RPM with APT is supported), simplifying the installation process, improving user documentation, and providing detailed API docs for creating plug-ins and extending Spine. The team is actively soliciting large installations that would allow Ticketmaster to get them up and running with Spine. More information is available at http://code.ticketmaster.com.

PLENARY SESSION

- ***Reconceptualizing Security***
  Bruce Schneier, Chief Security Technology Officer, BT

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

Bruce Schneier examined the human element in security and how people make decisions about risk and threats. Schneier began with the idea that two types of security have to be examined: the feeling of security and the reality of security. It is important to understand when they are the same, when they are different, and how they can differ. This can be hard to discuss, because language does not easily handle this split.

Security decisions tend to involve some type of trade-off. For example, people living in gated communities trade convenience and privacy for some added security. These types of decisions are subjective, and they may differ from person to person. The evaluation is not based on how good the result is, but on whether the gain is worth the loss. When trade-offs don't make sense, it often means that the true trade-off being made is not visible from the outside. Some imperceptible factors are involved which may not have anything to do with the security gained from the decision.

Like all animals, humans make security trade-offs all the time; it is part of survival. Human beings should be good at making these types of decisions, but they are not. Humans react to the feeling of security, not to the reality of security. This approach will succeed only while the feeling and the reality of security are the same. It breaks down as they begin to diverge. Our intuitive responses work well for the security of pre-history society, but they do not work as well for modern security issues.

To help understand how these decisions are made, two portions of the brain were examined. The amygdala is a primitive part of the brain that is the center of the "fight or flight" reflex. It reacts faster than conscious thought, but it can be overridden in humans by training. The neocortex is a newer section of the brain where consciousness is centered. It is slower and uses heuristics and is the source of cognitive biases, which can fail.

Schneier described an experiment on how people react to risk. The test subjects were divided into two groups, with each group given a different choice. The first group was asked to choose between a sure gain of $500 and a 50% chance to gain $1,000. The second group was asked to choose between a sure loss of $500 and a 50% chance of losing $1,000. It would make sense for the two groups to have similar percentages of those willing to take risks and those unwilling to take risks. However, the results showed that although 84% will take a sure gain over the chance for a greater gain, 70% will take the chance of a greater loss over a sure, but lesser, loss. This makes sense on a survival level, where any loss or even the lack of a gain can mean death. Similar studies over varied demographics suggest that this is a general human response to risk.

Other biases are a tendency toward optimism, a greater fear of risks that can't be controlled, placing greater importance to risks from people than from nature, and putting greater importance on risks to children. People tend to think that something is more likely the easier it is for them to think of it.

People also tend to value something more if they already have it than if they want to have it. This can be seen in a study where the subjects were divided into two groups, one given mugs and one without. Those with mugs were asked how much they would sell the mug for and those without were asked how much they would pay for one. Those with a mug tended to ask twice as much as those without the mug were willing to pay. This experiment has been repeated with more expensive items and with groups with different levels of wealth.

To further complicate the discussion, security can be thought of as having three parts. The feeling of security is how people intuitively feel about their security. The reality of security is the real world. The model of security is an intelligent representation of reality. Models can be limited by lack of good information and lack of technology. When models differ from feelings, they tend to be rejected. Over time, the models can be accepted and will eventually disappear into feeling. However, it can take a long time for the model to be accepted. For example, it took decades for the model of the dangers of smoking to be accepted. This can be a problem in technical security because the fast growth of the technology means the model may be defunct by the time it is accepted.

People are a crucial part of security systems, and their role has to be considered. It is important to consider more than just the security reality when designing a system. The feeling of security will also figure into how people react to the system. In addition to designing better systems, people need to be presented with better models. All three parts of security—reality, feeling, and model—need to be considered.

When asked whether people who study statistics make better decisions using mental shortcuts, Schneier replied that they do not, but they tend to be surer of their decisions. It was suggested that part of the reason that these shortcuts are used is the cost of cognition. Schneier replied that this is a factor but at this point it is hard to understand how important the cost of cognition is in when shortcuts are made. He agreed that the way people think about security can be considered rational, but he pointed out that it is not analytical. Dan Klein mentioned that it was likely that many in the audience were INTJs on a Myers-Briggs typology and thus more likely to see the flaws in these shortcuts. He asked how such more analytical people can relate to security issues. Schneier replied that the human response to story-telling should be leveraged.

VIRTUALIZATION

Summarized by Andrew Hoblitzell

- ***STORM: Simple Tool for Resource Management***
  Mark Dehus and Dirk Grunwald, University of Colorado, Boulder

Dehus said his group defined an appliance as "a combination of operating system and application components that provide a specific computing task," such as spam filtering, a wiki, etc. He introduced STORM, his group's virtualization system, which was designed to simplify development, deployment, and provisioning for common applications and appliances. He showed that the system reacts to changes in system load to deploy additional services and that it also can dynamically power client machines using IMPI controls to promote energy savings. The system was demonstrated using a scalable mail appliance, and he said that they had designed the system to be easy to configure and maintain. He said the group planned to eventually place their project on SourceForge.

In response to questions from the audience, Dehus said that the layering used by his group's system was rather immediate and that disk images were kept separately, so overhead should be kept to a minimum. Dehus said his group had not had time to test what would happen if a very large number of virtual machines were requested at once and the system was unable to handle it. Another audience member mentioned that HP had conducted similar work.

An article with more details about STORM begin2 on page 16 of this issue.

- ***IZO: Applications of Large-Window Compression to Virtual Machine Management***
  Mark A. Smith, Jan Pieper, Daniel Gruhl, and Lucas Villa Real, IBM Almaden Research Center

Mark Smith presented IZO, a large-window de-duplication compression tool that his group developed which provides faster and increased compression over existing large-window compression tools. He said that large-window compression methods were becoming more and more relevant because of falling storage costs and larger storage applications and that this would also make his group's tool increasingly relevant. He showed that they had applied their method to a number of VM management domains (e.g., deep freeze, backup) and that his group's system would help administrators more effectively store, administer, and move virtual machines. He cited initial experiments which showed up to 86% storage savings for some scenarios, and he said that in future work his group was concerned with the rapid growth of metadata in backups, the ability to determine optimal chunk size automatically, and adding additional convenience features to their project.

In response to audience questions, Smith said that chunk size would be adjustable to allow for lowering the probability of hash collisions for specific applications. When he was discussing the current rapid growth of data in the computer industry, Smith said he was happy to see that a new linear accelerator was currently being constructed in France.

- ***Portable Desktop Applications Based on P2P Transportation and Virtualization***
  Youhui Zhang, Xiaoling Wang, and Liang Hong, Tsinghua University

This paper focused on play-on-demand for common desktop applications. The group's approach was based on lightweight virtualization and network transportation to allow a user to run personalized software on any computer, even if that computer doesn't have that software installed on it locally. In the group's method, registry, files/directories, environment variables, and such are sent to a portable device as they are needed. Access control methods could be added to the method to prevent illegal access, and other methods besides P2P could be used to implement the methodology. The group's method was said to be especially relevant for the developing world. In the future, the group planned to make the system work entirely over the network and to run in user-level mode instead of in administrator mode.

When answering questions from the audience, the speaker stated that dependent libraries could be detected during the installation of a product in a software environment. Further discussion about the issue took place offline.

INVITED TALK

- ***Mac OS X: From the Server Room to Your Pocket***
  Jordan Hubbard, Director, UNIX Technology Group, Apple, Inc.

Summarized by Matthew Sacks (matthew@matthewsacks.com)

Jordan Hubbard spoke about some of the innovations and latest developments in the Mac operating system. Hubbard provided much information about the security subsystems and modifications to the OS X operating system that were exclusive and difficult to find in the documentation. Some of the mysteries debunked included the sandbox profile language and file quarantine APIs. The sandbox system offers fine-grained controls over what sandboxed processes can do via system calls, including limiting what files or directories can be read or written and whether the process may have network access. The security system is based on TrustedBSD's Mandatory Access Control framework.

Hubbard also provided suggestions for accessing some of Apple's open source projects such as MacRuby, WebKit, and ZFS. MacRuby is a version of Ruby that runs directly on OS X's Objective C libraries. Hubbard spoke of a little-known library for MacRuby that makes Ruby development for OS X integrate into HotCocoa. HotCocoa makes it even easier to write Ruby code to quickly develop OS X applications.

Although there wasn't much talk about WebKit, which now powers Google Chrome, there was much interest in the topic of OS X supporting ZFS (as of Leopard 10.5). ZFS is now available on macosforge.org and is a read-only file system. Sun just recently announced bootable ZFS, so there may soon be ZFS support in an upcoming release of the Mac OS, although this was not discussed at the talk.

The iPhone was also a topic of discussion about the Mac OS X operating system. Mac OS X is officially a fully certified UNIX system. There were many developments with the iPhone's graphical API, Core Animation, that actually made it back into the regular operating system, deviating from the normal integration pattern of moving from desktop to mobile.

There were many exciting bits of information presented at this talk about some of the quieter innovations in OS X at Apple. Hubbard tantalized and informed the crowd with often-exclusive information about the Mac OS and the growing strength of mobile as a computing platform.

INVITED TALK

- ***An Open Audit of an Open Certification Authority***
  Ian Grigg, CAcert

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

Grigg discussed CAcert's open audit. He examined both the decision behind the choice of an open audit and how successful the open audit has been. Grigg began by discussing his role and background. He is the independent auditor for CAcert, with a background in cryptography, security protocols, and architecting systems.

Audits are often used to create a sense of trust when the security of a company is not transparent. CAcert is supposed to have an open process that will itself reveal flaws.

CAcert is an open group organized as a community. CAcert can be divided into three groups: the assurance group, the business group, and the tech group. All of the people in these groups are members of CAcert.

As CAs were introduced and their numbers increased, there came a need to manage how CAs were added to browsers. The method that emerged was a systems audit. However, audits have some problems: Audits are expensive, the audit statement is too brief to be useful, it often isn't clear, the process is closed, and the process has not been updated to reflect current threats. As Mozilla became popular, it needed to write a policy to determine the procedure for accepting CAs. Mozilla performed this task very well, appointing Frank Hecker to lead a project to create the policy for accepting CAs to the root list. Grigg called this project "one of the best open governance projects I've ever seen." This policy allowed open criteria and open reviews to qualify.

To meet Mozilla's requirements, CAcert needed to undergo an audit. In mid 2005 David Ross created the criteria for CAcert. Ian Grigg took over the audit in early 2006. An important feature of the David Ross Criteria is that they focus on the risks, liabilities, and obligations. This means that they focus on the interests of the end users. This is different from WebTrust, which seems to be set up more to protect the CA.

The industry standard is to set liability to zero. The problem with liability to end users is that the number of potential end users affected is so great that it would be an enormous cost to provide even a small amount of liability. The question is, if there is no liability, what is the value of a CA? CAcert answered this question by providing access to everyone but different levels of liability to members and nonmembers. Nonmembers may use CAcert, but they may not rely on it—there is no liability. Members are allowed to rely on CAcert certificates and accept some liability. Liability is capped at 1000 euros, is allocated from member to member, and will be decided by CAcert's own arbitration system. All members of CAcert are subject to arbitration. CAcert has an open process, which allows end users to decide whether CAcert is worthwhile.

CAcert uses a web of trust to assure identity. In CAcert's web of trust, only Assurers can make statements. This makes it important to properly validate the Assurers. To meet this need, the Assurer Challenge was introduced and passing it was required to continue as an Assurer. In addition, an Assurance Policy and Assurance Handbook were created to create a standard of assurance.

In the classical PKI model, the importance of a certificate is to provide a name. This allows trading with good people and a way to track down those who are not good. However, online markets, such as eBay, demonstrate the fact that having the "real" name itself is not as important as having a way to track reputation. The cost of resolution of disputes increase with distance, which is counterintuitive, as certificates should be more valuable as distance increases. Here the CAcert arbitration system is invaluable, because it limits the increased cost of resolution over greater distances.

The CAcert system consists of the end users on the Internet, the CAcert core server, which is accessible by the Internet, and the signing server, which is only connected to the core server by a serial link. The threats to CAcert may be grouped into bad certificates, data breaches, and the compromise of the root keys. CAcert's lines of defense are the community itself, reducing the data stored, typical online security technology, the security of the architecture, and the governance of the community.

Owing to a series of issues with hosting and problems relocating the machines in an auditably secure location, core machines failed the audit, as did the root key stored on some of those systems. Because of these events, the entire system administration team needed to be replaced with a new location and new team in the Netherlands.

Currently, physical security is ready to be audited, the logical security is close to being ready to audit, the documentation has a bug in CPS that needs to be fixed and the security manual needs some work, and eventually there needs to be some type of cross-team security. The next steps for the audit are to check documentation and assurance and to create new roots.

CAcert has done well by describing risks, liabilities, and obligations. It is also doing well by providing arbitration and progressing well with education. In the beginning, there were problems in the secrecy involved, but a move to openness has been made and CAcert has formally adopted a "no secrecy" policy.

After the talk, a questioner asked how, since in security secrecy is often considered a good thing, do you change that mindset? Grigg recommended small, gradual changes.

When asked what the plans were for physical security, Grigg replied that although the hosting building has a high security rating, there are weaknesses in the physical security. However, additional physical security is not the highest priority; instead, other security issues such as governance are being focused on now. CAcert has a different approach from other CAs; others seek to protect the root keys at all costs, which might be considered to be an absolute approach, whereas CAcert seeks to plan for all failures including compromise of roots, which Grigg referred to as disaster-style management. The latter approach is favored generally in audit criteria.

ON THE WIRE

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

- ***Topnet: A Network-aware top(1)***
  Antonis Theocharides, Demetres Antoniades, Michalis Polychronakis, Elias Athanasopoulos, and Evangelos P. Markatos, Institute of Computer Science, Foundation for Research and Technology (ICI-FORTH), Hellas, Greece

Demetres Antoniades presented Topnet, a modified Linux top command that can provide a process-oriented approach to network monitoring. Given that sysadmins use the top command to monitor process activity, it is perhaps equally accurate to characterize Topnet as implementing a network-centric approach to process monitoring. Topnet does this by augmenting the traditional top command to show two additional columns with each process's traffic inbound and outbound and, by introducing new hot keys, allowing the user to select how the processes are sorted by activity, among other things. The authors challenged themselves to essentially use only two sources of information: netstat (existing kernel-provided network statistics) and libpcap (the portable packet capture library used by tools such as tcpdump and wireshark). Thus Topnet does not require kernel modifications; instead it maintains a hash table that is used to correlate network traffic flows with processes based on the network endpoints bound to each process.

Demetres then presented results of various experiments to assess Topnet's performance. These focused on two areas: (1) the accuracy of Topnet's measurement of traffic volume over time, and (2) the load Topnet imposes on the system itself. The authors performed a number of TCP bulk-data transfers with both synthetic and real Web (via wget) and BitTorrent traffic. Overall, Topnet was shown to exhibit only small errors. Two sources of the discrepancies are (a) small differences in the binning into timeslots and (b) discrepancies that arise by observing traffic at the interface (Topnet) versus within the process (wget, etc.), such as retransmissions by the TCP layer. Regarding the load placed on the system when using Topnet, the results generally showed that although the CPU load can be prohibitively high at very high traffic rates, it grows linearly, being low at traffic rates that one would expect for most Linux machines. Thus the performance evaluation suggests that, like top, Topnet would be an effective sysadmin tool to run on demand for troubleshooting or information gathering in most environments.

One attendee asked how we might identify processes that elude Topnet, whether maliciously or otherwise. Demetres agreed that it is sometimes impossible for Topnet to associate some traffic with a process, such as when raw sockets or sockets with ephemeral bindings are employed; however, he noted that Topnet still reports such traffic as "orphaned," so the sysadmin is not left completely unaware. He further noted that it would likely require the addition of kernel support for the measurement of network traffic on a process basis to solve this problem completely. Demetres was also asked whether Topnet could report on threads rather than processes, to which he said no. Another attendee inquired whether Topnet can monitor traffic on loopback and alias interfaces. Demetres said that it can monitor any interface that libpcap can, so Topnet should be able to monitor those interfaces. Finally a number of attendees wondered when and where they could obtain Topnet. Demetres admitted that, although Topnet is based on freely available software, it is not yet available for download. This is, in part, because not all of its authors are currently available to work on it.

However, he invited interested parties to contact him by email. He can be reached at danton@ics.forth.gr.

- ***Fast Packet Classification for Snort by Native Compilation of Rules***
  Alok Tongaonkar, Sreenaath Vasudevan, and R. Sekar, Stony Brook University

Alok Tongaonkar presented work that improves the popular Snort Intrusion Detection System's (IDS) performance. Snort takes a hybrid approach to matching its rules to packet content; it first uses fast string matching techniques then follows with more general, but slower, regular expression techniques. Whereas much prior work has focused on improving only the latter, Alok and his collaborators focus on improving packet classification performance by compiling Snort rules into native code that is dynamically loaded. Their claim is that this strategy has at least three advantages: (1) It can speed up each of Snort's matching components; (2) the technique is also applicable to Deep Packet Inspection (DPI); and (3) these improvements could be used as a basis for IDS design.

Alok presented an overview of Snort's rule-based language, configuration, and variable declarations, then described how Snort uses an interpreter to process these at run time. This interpreter uses various sequentially accessed data structures and function pointers that, not unlike an interpreted programming language, limit its performance, for instance in how they force operations on the CPU data cache. The new technique achieves performance gains by having Snort use compiled rules rather than interpreted rules, in much the way that a compiled programming language and its resulting executables with native machine instructions typically yield better performance than interpreted languages. Since it would be difficult for network administrators to write, maintain, and evaluate IDS rules written directly in C code, the authors developed a compilation process that uses an intermediate language that they call the Packet Classification Language (PCL). Using PCL, the stepwise process is: (1) translate Snort rules to PCL using a PCL translator, (2) translate the PCL-specified rules to C code, and then (3) use the C compiler to create a shared object that is loaded by Snort, thus enabling Snort to perform tests to match packets to rules as native instructions. Finally, Alok presented an evaluation of the performance of this native compilation of rules versus traditional Snort, by testing 300 rules both with publicly available packet traces (DARPA '99) and using sample traffic from the campus. Overall, they achieved up to a fivefold improvement in Snort's packet classification (non–regular expression) performance for both the traces and campus traffic.

An audience member noted that PCL rules look very similar to user-supplied expressions for tcpdump and libpcap and wondered whether these improvements could be done in libpcap so that applications other than Snort might benefit. Alok said that, indeed, they have been considering how the native compilation technique could be used to improve Berkeley Packet Filter expressions.

INVITED TALK

- ***OpenSolaris and the Direction of Future Operating Systems***
  James Hughes, Sun Microsystems

Summarized by Alex Boster (aboster@machineepsilon.com)

The future of OpenSolaris, and indeed operating systems in general, is in providing the support necessary to effectively utilize parallel processing on systems with an ever-increasing number of processor cores. Sun's Batoka server with up to 256 threads and 0.5 TB of memory is an example. The search for so-called Goldilocks applications, such as MapReduce, that can make full use of such machines continues.

High-thread-count machines are the future and the winners will solve their problems with highly parallel solutions. Run-time parallelization remains the holy grail in the realm, with languages hiding the parallelism from the programmer. Operating systems must provide the tools, libraries, and developer support for this shift, by reducing complexity while enabling efficiency.

There are new features of Solaris that the audience might not be aware of. Solaris is now NUMA aware, including a NUMA memory management system that is fully transparent to the developer. ZFS is now a first-class, bootable filesystem usable as the root filesystem. Among other features, ZFS provides data integrity features, encryption, and snapshots. DTrace, familiar to this audience, allows complex debugging logic to be executed on production code. The future must enable bigger, faster applications, many of which, from simulation to games to finance, are numerically intensive.

Currently, there are several noteworthy parallel programming languages. Fortress is noteworthy for enabling implicit parallelism—hiding the complexity from the programmer. MapReduce is a real-world example in common use (e.g., by Google). Both Hadoop, a scalable Java MapReduce solution, and Phoenix, written in C, are available for OpenSolaris.

The coming revolution toward high-thread-count parallelism looks to be the biggest architectural change in computing history. Applications must become scalable or die. However, computers will be able to do a lot of this parallelization for the developer—such IT infrastructure will be a competitive advantage.

OpenSolaris is the leading edge of Solaris, with ZFS root and boot, new packaging and patching facilities (thanks to ZFS snapshots), and a userland more familiar to Linux users. It is possible that OpenSolaris will become Solaris 11.

New security features include encrypted storage, key management, and high assurance containment (e.g., running Windows in a Solaris TX labeled zone). With the spiraling issues surrounding securing data in a world with laptops, thumb drives, smartphones, and so on, encrypted ZFS (eZFS) is being developed. Most filesystem encryption systems still have a number of weaknesses, such as: root can read user data when the user is not logged in; deletion does not erase data; and raw data on RAID may be in the clear. eZFS addresses each of these issues by requiring the user's key to access data and zeroing data on deletion. Key management is a critical problem, but the solutions are not high-tech: Keys must be captured, categorized, and managed over their whole lifetime.

A questioner asked about performance testing of eZFS on x86 hardware. The presenter noted that ZFS is cache-hungry and asynchronous, so if the load fits in cache, latency is low—also, encryption is four times faster than compression. To a question about large vendor support for OpenSolaris, Hughes said that when there is enough community demand, support should be forthcoming. A questioner asked how one does backups on eZFS user data if it is encrypted from the admin user. The backup system must have user keys, Hughes explained, or the users must handle their own backups. In response to whether Linux is seen as a competitor to OpenSolaris, Hughes agreed that it is.

INVITED TALK

- ***Auditing UNIX File Systems***
  Raphael Reich, Varonis

  Summarized by Matthew Sacks (matthew@matthewsacks.com)

Reich presented some of the reasons for auditing and common pitfalls in auditing UNIX file systems and how auditing fits into the broader topic of IT governance. The context of auditing UNIX file systems in this talk was focused on unstructured data, such as documents, images, blueprints, or any kind of data stored on a file share. IDC estimates that unstructured data represents about 80 percent of all business data. Reich claims that unstructured data is overly accessible in general.

The problem with auditing file systems is that it is difficult to understand who owns data, and using metadata is a poor method for understanding the owners and consumers of data. This problem becomes increasingly complex when data is being migrated across different locations. IT professionals often do not have the business context for the data, although they are often the individuals managing where the data is stored.

Auditing usually involves capturing live, real-time information, which causes poor performance on the system being audited. As a result, the methodologies for auditing data are typically episodic, so data is only captured when an incident happens or is reported and defeats the purpose of auditing or results in too much information to sift through.

The system for solving this problem is to use probes and a central database to constrain the auditing information. Resolution of incidents where a user has access to data that he or she should not have is done by only running a simple command or, if using a graphical interface, by clicking.

The end goal is that people who need to have access to data only have access to the data they need, and those who do not need to access certain data will not be able to. Without a system or way to audit this process, it is an inefficient, manual process. Reich admits that he works for a vendor whose focus is an auditing product, but that this type of auditing awareness and implementation is important regardless of where the auditing solution comes from.

GETTING STUFF DONE

No summary available: See www.usenix.org/lisa08/tech/ for the papers in HTML and PDF.

INVITED TALK

- ***WTFM: Documentation and the System Administrator***
  Janice Gelb, Sun Microsystems

  Summarized by Alex Boster (aboster@machineepsilon.com)

Most system administrators fear and hate documentation, both writing and reading it. This talk attempted to alleviate that frustration by explaining why system administration documentation is important, showing how to resolve common documentation problem areas using real-world examples, and describing how to improve product documentation from your company and from companies that make products you use.

Documenting systems and procedures can reveal missing information, gaps, and inefficiencies. What should be documented? The list includes project plans, diagrams, infrastructure details, feature and equipment requests, server log formats, and backup and restore procedures, as well as user documentation.

If starting from scratch, consider hiring or assigning a technical writer for the initial wave of documentation. Make an outline of the document and perhaps use existing documentation structures, such as those suggested by Network DNA (http://network-documentation.com/). Structuring before you write helps make sure your documentation covers all necessary material. Making a documentation template helps a lot. Keeping documentation up to date is also very important.

A long list of detailed techniques and dos and don'ts was presented, along with real-world examples from commercial product documentation. Problems with procedure lists included failing to number sequential steps or numbering steps that aren't sequential, burying actions in paragraphs, and failing to note important information before the step in which it applies. When and how to use lists, tables, FAQs, and illustrations was discussed.

A questioner asked about wikis. These are good for community involvement, but not for product documentation—as such, though, they are often useful for system administration documentation. A questioner suggested that style guides, such as Strunk and White, were useful resources. The speaker recommended their use and also mentioned *Read Me First! A Style Guide for the Computer Industry*, for which the presenter was the project lead, but which is not Sun specific. In response to a question about how to hire help on a shoestring budget, starving students and students in organizations devoted to tech writing were mentioned as good resources.

INVITED TALK

- ***Fighting Spam with pf***
  Dan Langille, Afilias USA, Inc.

  Summarized by Qi Liao (qliao@nd.edu)

Dan Langille talked about how to use a firewall to fight off spam. The specific firewall in question presented in this talk is the Packet Filter (pf) in OpenBSD and FreeBSD. There has been increasing adoption among system administrators toward using "greylisting" to fight spam. Dan showed how to achieve this greylisting functionality in pf.

If the sender is known to be good, its packets will be directly passed to the local MTA (whitelisting). If the sender is known to be bad, the connection is terminated (blacklisting). Greylisting is a practice between whitelisting and blacklisting, in the sense that the receiver is not sure about the sender. So if the sender is new to the receiver, the pf will redirect the connection request to TCP port spamd, which temporarily refuses and asks the sender to resend at a later time. The basic assumption here is that the spammer is lazy! A spammer would move on rather than requeueing the messages. If the sender does come back after several retries, spamlogd will then move items from the greylist to the whitelist stored in spamdb.

Several concerns were raised by the audience:

- High latency for emails. This is the top concern among most system administrators. In many cases, researchers collaborate over a document and circulate it around. It would be unacceptable for users to be waiting for such important emails while they are going through the greylisting process.
- Lack of standards compliance.
- Lack of empirical evidence on the false positive rates.
- Lack of a reputational score in a dynamic environment where the reputation of the sender can change over time, as it can be infected by viruses and worms.

Although this solution is not perfect, it does have the advantage of simplicity, requires no changes to existing mail servers, and provides a cheap way of defending against spam. For administrators who are interested in setting up pf/spamd, the detailed instructions can be found at http://www.freebsddiary.org/pf.php.

PLENARY SESSION

- ***The State of Electronic Voting, 2008***
  David Wagner, University of California, Berkeley

  Summarized by Alex Boster (aboster@machineepsilon.com)

How did we get here? Florida, 2000. A voting system vendor sent a plea for help: A machine had recorded negative 16,022 votes for Gore. Reflected in media vote totals, this caused Al Gore to call George Bush to concede the election. An aide managed to catch Gore while en route to give his concession speech—and the rest is history. No explanation was ever given for the –16,022 votes.

This incident, along with the Florida 2000 recount generally, was a "nuclear bomb" in election systems. In response, Congress passed the Help America Vote Act (HAVA), requiring most counties to replace old voting systems, such as punch cards. The act had a firm, use-it-or-lose-it funding deadline. This led to a rapid acceleration of the adoption of electronic voting systems, including so-called direct recording electronic (DRE) systems, typically electronic touchscreen machines with no paper record.

One example of voting system problems occurred in a 2006 U.S. House election in Sarasota, Florida (FL-13). The margin in the district was 369 votes (0.15%), whereas no vote in the House race was recorded for 18,412 ballots. In the county in question, this undervote amounted to 14% of all ballots, but other counties in the district had a more typical 2%–3% undervote. Paper ballots also showed a 2%–3% undervote. Trouble logs show lots of voter complaints in precincts that used the suspect DRE system. It is suspected that the issue was a bad ballot layout that made it easy to overlook the U.S. House race. This, along with the famous "butterfly ballot" in Florida 2000, demonstrates the major importance of usability considerations.

Reliability issues have plagued electronic voting systems. The 2004 elections yielded several examples. In North Carolina, at least 4500 votes were lost after the capacity of the machine was exceeded, yet the machine failed silently while appearing to record votes. In Columbus, Ohio, a memory card failed on a machine that did not checksum its data—it recorded more votes for Bush than there were voters in the precinct. In Broward County, Florida, negative votes were recorded after about 32,000 had been cast—a clear sign of a 16-bit signed integer overflowing.

Major security issues have also been discovered in electronic voting systems. After Diebold accidentally made an archive of its election systems software public via anonymous FTP, researchers at Johns Hopkins University quickly released a report detailing many serious security issues. Diebold refused to allow academics or government representatives to examine its hardware or software, claiming they were proprietary. In 2006, Princeton researchers gained physical access to a machine and discovered a buffer overrun exploit that allowed a software upgrade to be loaded from the memory card without authorization. One hotel minibar–style key could open the physical locks on all machines; copies have been made from photos of these keys.

In 2007, California Secretary of State Debra Bowen commissioned a top-to-bottom review of voting systems and required vendor cooperation in order to recertify their machines—and, indeed, she decertified vendors that did not cooperate. The speaker was the principal investigator for this study. The commission discovered serious security issues with all the systems studied. Cryptographic issues included trivial password encodings, hard-coded passwords, and unencrypted passwords. Many simple bugs were uncovered, such as the misuse of || versus && and using a simple char as a buffer instead of char[]. All systems allowed malicious code to propagate virally, allowing one malicious memory card to infect not just one machine but the county central machines, and then on to all voting machines. No vendor followed standard defensive and secure programming techniques. As a result of these findings, Secretary Bowen decertified all of the machines until voter verifiable paper trails with audits were in place.

How do things stand going forward? The most important attributes of voting systems are auditability, recountability, and transparency. However, one-fourth of states still use paperless DREs—the very same systems the commission studied. There is no silver bullet: Elections are complex and are run by an army of trained volunteers.

A questioner asked about any partisan correlation to the pattern of states that still use paperless DRE machines, but the speaker doubted it—other factors are more probable. Another questioner noted that other secretaries of state think Minnesota (undergoing a statewide recount at the time) was lucky to have a paper trail, but politicization had delegitimized paper. In response to a question about the mechanical lever machines used for many years, the speaker noted that they had many of the same issues with known methods of cheating, but that only one machine at a time could be rigged—there is no viral capability. A non-American audience member asked about federal standardization. Wagner noted that although there are many things the federal government could do but has not yet done, the states and counties run elections in the United States. It was noted that the Constitution gives Congress the authority to decide how its members are elected, but Wagner pointed out that there are many political constraints on this. As to whether using PDFs or TIFFs instead of paper was a option, the speaker replied that it was very hard to find a suitable replacement for paper.

WORK-IN-PROGRESS REPORTS (WIPS)

Summarized by Alex Boster (aboster@machineepsilon.com)

David Pullman of the National Institute of Standards & Technology presented "Moving to a Geographically Distributed HA Cluster Using iSCSI." An old HA cluster was presented for comparison, along with a diagram and details of the new HA cluster housed in two different buildings for better reliability. Objectives for the new system included using commodity hardware, leveraging existing Veritas/Symantec expertise, and moving to an iSCSI SAN architecture. Details of the particular hardware chosen were presented. The system is in place and running with a few post-implementation issues—including vendor issues—some of which are due to pushing the envelope on iSCSI.

Dan Kegel of Google described a pre-commit autotest system. Automatic testing on nightly builds or after commit is common, but it comes at a cost—any issues found are already in the source tree. The speaker's experience with the system he built is that pre-commit testing leads to a cleaner source tree and happier developers. The worst problems he encounters are broken tests, particularly those that only break sporadically. A questioner asked how changes are propagated to the build machine. Each developer could get a separate branch, or patches can be sent via mailing list. Two questioners asked about merging issues, but merging is a problem that always requires human intervention.

Joe Muggli of the University of Illinois at Urbana-Champaign discussed SELS, a Secure (Encrypted) Email List System (http://sels.ncsa.uiuc.edu). SELS is based on proxy encryption techniques, which enable the transformation of cipher-text from one key to another without revealing the plaintext. Exchanging emails using SELS ensures confidentiality, integrity, and authentication. This includes ensuring confidentiality while in transit through the list server, a functionality that is uniquely supported by proxy encryption. SELS makes use of OpenPGP and Mailman and is compatible with common email clients including Outlook, Thunderbird, Mac Mail, Emacs, and Mutt. Several national and international incident response teams are using SELS. A questioner asked about the use of Mailman, a system with a history of security issues. The Mailman service sees only encrypted messages, so this is not considered a major problem.

Brent Chapman of Netomata presented "Automating Network Configuration: Netomata Config Generator (NCG)." NCG generates config files for all network services using templates and a generator engine—think of it as Puppet or Cfengine for networks. Automatic generation leads to greater reliability through more consistent configuration and better scalability owing to the ease of adding devices and services. It is written in Ruby and is in alpha now. One questioner asked where to draw the line between devices and hosts. NCG is agnostic on this issue. Another questioner asked about conflicting rules. Since NCG is template-based, the user must review the output to make sure it's sensible.

Dave Plonka of the University of Wisconsin discussed "Network Admins Are Programmers: An Analysis of Network Management Artifacts," which asks what programming tools and analysis techniques are useful in network management, with tens of thousands of devices to be managed. Device configuration files are placed in a CVS repository and various standard analysis tools are then used to analyze the repository. Each device corresponds to one CVS file, and files are grouped into "modules," which correspond to geographic areas. So, for example, one can track the activities of one, or all, users in a day and discover that Wednesday is the main day network changes are made. Lines of "code" turn out not to be a good measure for complexity, but churn (changes per day) is.

Jason Faulkner of mailtrust.com discussed some new strategies for networking and load balancing. The requirements of this system included the use of commodity hardware, use of Linux, and that the backend software be aware of the external user's IP address. There were issues that excluded the use of SNAT, DNAT, shared IP, multiple load balancers, and layer 7 proxies. This technique uses one-to-one IP to firewall mapping, and it takes advantage of the fact that Linux supports 255 routing tables. Each IP is mapped to a different routing table. Downsides include that round-robin DNS must be used and there is no way for the firewall to know about backend server load and adjust accordingly.

Will Nowak of Northeastern University discussed migrating his college to OpenLDAP from legacy systems. To that end, a modular Python-based conversion and syncing tool was written to push from NIS to LDAP. Details of the college's old SunOne LDAP cluster and the new OpenLDAP cluster were presented. Code is available at http://code.google.com/p/nisldapsync/.

Chris McEniry of Sony Computer Entertainment America presented a password management tool that uses a generic command-line interface to Web services called shiv. Knowledge of particular commands accepted by a Web service is not needed, so the CLI client is decoupled from the service. A group password storage utility has been written on top of this system, allowing all the benefits of a CLI. Other Web services, such as inventory management and auto discovered switch information, are also accessed with shiv. Chris was hopeful that shiv would be generally available soon.

Nicole Velasquez of the University of Arizona presented a configuration management tool. A test lab within IBM has developed a tool to monitor and manage its configurations from the port level. The tool is built with MySQL, Apache, and PHP. This tool has allowed users in three countries and two continents to share hundreds of servers, more than 100 switches, and over 60 enterprise storage boxes seamlessly, resulting in greater system utilization and higher availability. Clean, color-coded interfaces allow users to visualize the system at a glance and flexible reporting keeps management at bay. Workflow management functionality has been included in the tool to allow for the request and tracking of configuration changes.

INVITED TALK

- ***Deterministic System Administration***
  Andrew Hume, AT&T Labs—Research

Summarized by Marc Chiarini (marc.chairini@tufts.edu)

Andrew Hume gave an invited talk on his vigorous attempts to combat "pixies" and entropy at the data centers used by his AT&T Research Lab. The symptoms of entropy include, but are not limited to, senior architects producing nothing but complicated drawings of rack layouts, excessive use of large cable lengths, dormant racks, attached disk arrays without power, and an inability to understand what connects to what. AT&T commissioned a study that found that 100% accurate physical inventories eventually declined to 60%–70% over one year. Hume insists that we can do better.

After a brief detour into logical positivism (which describes a worldview built only on empirical statements), Hume offers a naive solution: Produce a description of what you want, place equipment in a pile in the middle of the machine room, select a piece of stuff, and verify it is connected correctly; if not, make it so. Repeat until everything is verified. The biggest problem is that, in the presence of pixies, one must verify endlessly. Because of various constraints, this is not usually done.

Fettle is Hume's primary contribution to the battle against entropy. Given a moderately simple textual description of component definitions, cable types and lengths, and specifications for layout and connectivity, Fettle is able to produce a visual representation of the desired rack design, a list of cable orders, power analyses, and more. Written in Ruby, the program is designed to provide bookkeeping functionality and quick feedback, and Hume has been pleased with the results so far. Although he admits to not caring much about networking, Hume turns to PXE booting and two more Ruby tools, Limn and Treetop (a little language), to help with the tasks of logical network layout and giving devices most of the necessary configuration information to start up.

In Fettle one can specify several (somewhat hierarchical) elements when defining one or more racks. Racks, bars (tie bars), boxes, and ports are instantiated and "anchored" (given a location according to name and dimensions). Wire elements connect any two port elements. A rack element contains the names, dimensions, and 3D locations of one or more primary doors, bars, boxes, and access ports. Bar elements specify the names, dimensions, locations, and type of cable held by each cabling channel. Boxes declare the names and dimensions of each physical device placed in a rack. Port elements describe the names and dimensions of each port assembly on a box and also the name, type, and dimensions of each individual port. Finally, wire elements describe the length and type of cable between two ports and what bar to run along. Optionally, macros can be defined to instantiate multiple servers plus the wires they need.

Once a specification is processed, Fettle outputs VRML visualizations that can be manipulated in 3D to see exactly how different racks and cabling should be laid out. This allows comparisons to be made between what should be (the model) and what is (the reality of the machine room). Cable labels are also generated from the diagram, and certain types of connectivity can be easily spot-checked (e.g., power, KVM). Fettle itself produces other kinds of output, such as automated routing of cables (which Hume admits is slow in Ruby), and machine-readable tabular summaries of the specs.

All in all, Hume considers Fettle a lifesaver, although more work needs to be done to speed up automated cable routing and improve networking aspects. He hopes that others will pick up where he leaves off when the tools are ready for general consumption. Despite its successes, tools such as Fettle can only aid the system administrator in matching reality to the ideal. Hume remains amazed at the consistency and speed with which configurations decay.

INVITED TALK

- ***Designing, Building, and Populating a 10-Megawatt Datacenter***
  Doug Hughes, D.E. Shaw Research, LLC

  Summarized by Ben Allen (bsa8923@rit.edu)

Doug Hughes from D.E. Shaw Research, a firm that develops novel algorithms and machine architectures for high-speed molecular dynamics simulations of proteins and other biological macromolecules, presented a talk on building his company's new data center and many of the choices, gotchas, and tips he found along the way.

Hughes went on to show the benefits and differences between wet and dry cooling systems. The main benefit of a wet cooling system is efficiency. However, water pipes take quite a bit of space and pipes can leak. Electronics and water do not mix well. In addition, humidity control with a wet system can be problematic. Finally, if your environment freezes in the winter, you must ensure that cooling water pipes do not freeze during maintenance periods. Dry systems, in contrast, can control humidity well using waste heat and can be placed almost anywhere with no major structural pipes required. However, dry systems are less efficient.

In a data center there are two competing factors: human comfort and maximizing the temperature difference (ΔT) between the inlet and the outlet. No one wants to work in a frigid environment nor in a blistering hot one. Creating a high ΔT results in the coldest possible "cold aisle" and hottest possible "hot aisle." The best cooling design is a compromise between human comfort and ΔT.

Humidification is used in a data center to moderate static electricity. Various sources of humidification are available, including plant steam, steam canisters, and infrared or ultrasonic systems. A general recommendation of 40%–60% humidification is the industry standard, but Hughes believes this tolerance is a bit tight.

Various economizing techniques are available to decrease the cost of cooling a data center. On the air side, venting waste heat directly outside can lead to considerable cost savings. In addition, humidification of cooler air is more efficient, as cooler air's dew point is lower. On the water side, a heat exchanger placed outdoors can increase savings, especially in colder regions.

Hughes recommends a few things when using a wet cooling system. First, disable humidification on all but one of your cooling units, or use a specific-purpose humidifier. The other option is to ensure precise calibration of all cooling units. Hughes recommended this because the cooling units will fight to humidify and dehumidify. Next, keep an eye out for changing conditions in your environment such as increased load on servers, increased or decreased number of servers, and changes in outside temperatures. Next, disable reheat. Reheating is the process of dehumidification where the system chills the air down to the dew point and then reheats it. It is much more efficient to have one unit bypass warm air from outside.

A number of points must be considered for the flooring of a data center. Generally two choices are available: cement/epoxy floors and raised floors. A cement/epoxy floor has a high weight load, but it is bad for chilled water cooling, as pipes are usually run under the floor. A raised floor is more expensive, but it allows room for chilled water piping and cables. A raised floor becomes more expensive when high load capacities are needed and as the height of the floor increases. In addition, when using a raised floor with chilled water cooling, consider that leaking water pipes and power cables do not mix well.

Hughes presented various fire-suppression techniques and technologies available for data centers. A pre-action system is a water-based system with sprinklers. However, the pipes of the system are filled with air during normal operation. When smoke is detected, a pump is turned on and floods the pipes. Water is only released when the individual sprinkler heads reach a certain temperature. The biggest problem with using a water-based system is cleanup. In addition to cleanup, water is not efficient at putting out interior fires or fires inside contained areas, as it takes a little while for the water to reach inside the contained area. The next type of fire suppression is a dry-agent-based system. The benefits of a dry system are minimal downtime, as there's no dry-out period, and that interior fires are extinguished quickly. However, some systems require room sealing and have a corrosion potential. A new, potassium-based system named Aero-K is safe for humans and hardware.

A number of suggestions were presented on power efficiency. High voltages and fewer voltage conversions equal better efficiency. Typically, each power conversion wasted about 1% to 2% of the energy. The use of a three-phase system allows 173% more power to be carried in a power line than with a single-phase system. When buying equipment, insist on high efficiency and correctly sized power supplies from vendors with power factor correction. Finally, use only as much redundancy as required.

Another consideration when building a data center is the density required (e.g., how many kilowatts per rack). Hughes noted that although blade servers offer cabling advantages and space savings, their typical power requirements per compute unit are about the same as new servers. In general, as density increases, cooling and power requirements increase too.

During the Q&A session, the suggestion was made by an audience member that sump pumps and drip pan pumps should be on protected power. A question of flywheel use in Hughes's data center was brought up. Flywheels are used as an alternative to uninterruptible power supplies. Although they only offer a short duration of power, this is typically enough time for generators to turn on. Flywheels are less toxic, require less maintenance, and are more efficient than uninterruptible power supplies.

LUNCHTIME TALK

- ***"Standard Deviations" of the "Average" System Administrator***
  Alva L. Couch, Tufts University

No summary available: See www.usenix.org/lisa08/tech/ for the presentation slides.

INVITED TALK

- ***System Administration and the Economics of Plenty***
  Tom Limoncelli, Google NYC

  Summarized by Qi Liao (qliao@nd.edu)

Tom Limoncelli talked about how changing resources changes the practice of system administration. Computing resources are getting cheaper and are ample nowadays. This movement from scarcity in the past to the current land of plenty has a significant impact on system administration policies. For example, CPU resources were once scarce, so administration mainly focused on fair timesharing. Now, since everyone has his or her own CPUs on PCs, modern policy switches to focus on desktops. Having cheaper servers shifts the dominant cost from hardware to power. Therefore, modern policy becomes focused on green power. Cheaper and larger storage makes it a community resource. Increasing network bandwidth results in the modern policy of dedicated port per user while keeping bandwidth shaping. Finally, helpdesks evolved from a rigid, do-not-want-to-be-abused attitude to a more user-friendly environment.

The role of the system and network administrator as a gatekeeper has been made obsolete by Internet resource abundance (e.g., there are 11 hours of video uploaded each minute onto YouTube, blogs, etc.). The gatekeeping role is shifting to a curator one, in a process of disintermediation (removing the middleman). The traditional model, in which the IT department picks the apps and controls everything rather than the users, is no longer valid.

Hosted applications (or the fancy name Software as a Service, SaaS) and cloud computing are gaining popularity these days. The question is, "Is cloud-based computing the end of system administration?" Tom suggested several directions in system administration: cloud systems, legacy apps, desktop life-cycle management, help desks, monitoring and SLAs, IT coordinators, change management, release engineering, and security and compliance.

One audience member was opposed to getting rid of the gatekeeper. Tom agreed and further emphasized that having the right choice of gatekeeper (what to block or allow) is often a difficult task. Another audience member offered that what he got from the talk was that the scarcity of resources is changing from hardware/bandwidth to electric power and system admins' time. Tom commented that he would like to talk about green computing and time management of system admins another time.

Tom Limoncelli is the author of *The Practice of System and Network Administration* and a few other books. For more information, visit www.EverythingSysadmin.com.

- ***Inside DreamWorks Animation Studios: A Look at Past, Present, and Future Challenges***
  Sean Kamath and Mike Cutler, PDI/DreamWorks

No summary available.

INVITED TALK

- ***Beyond VDI: Why Thin Client Computing and Virtual Desktop Infrastructures Aren't Cutting It***
  Monica Lam, MokaFive and Stanford University

  Summarized by Ben Allen (bsa8923@rit.edu)

Monica Lam presented current issues and myths of Virtual Desktop Infrastructure (VDI) and MokaFive's LivePC product. MokaFive can be found at mokafive.com, where the player portion of LivePC is available for free.

There are several reasons why thin-client computing does not reduce the cost of hardware. First is the reduced cost of a PC today and the similar cost of thin-client hardware. Monika gave the example of a suitable PC costing $499 and thin-client hardware costing $300 plus $60 a year. In addition, employers can depend on employees using their own computers. Another reason is that moving desktop virtualization into a data center incurs additional datacenter operational costs (e.g., having to provide cooling and power). If the virtualization is running at the end point, often passive cooling can be used. Finally, when designing the systems for servers in a data center to support virtualization you must provision for the "Super Bowl effect" or the theoretical event when all your users log in and use their virtualized desktops at the same time.

Centralized management does not have to lead to a bad user experience. VDI, where the virtual machines are run on a central server, introduces a few factors that lead to a bad user experience. First, VDI has overhead requirements: Running multiple virtual machines on a single server causes resources to be shared among many users. Next, because the user is running on a remote display, all display information has to be sent and received across whatever network connection the user is on. Often this leads to very slow interaction performance. In addition, 3D graphics or other graphic-intensive applications are very difficult to interact with over a remote desktop.

MokaFive's LivePC product attempts to solve the problems of VDI by offering a centralized management interface that allows administrators the ability to create, update, and publish virtual machines. The virtual machines are published to an HTTP server and made available for downloading by the client. The client portion of the product lives on the user's machine and downloads and runs the virtual machine. Virtual machines created by LivePC maintain two virtual hard drives: one managed by the administrator remotely and another used to store any local changes to the virtual machine. LivePC will automatically pull differential updates to the first virtual hard drive as it is updated by the administrator. In addition, LivePC allows the user to revert back to the original virtual machine, undoing any changes to the operating system.

In response to a question about the case of users needing shared access to large pools of data, Monica noted that this is the one application where VDI is quite useful. Another question was asked how to back up local changes in the LivePC product. Monica responded that MokaFive decided to let users decide how to back up local changes, as most enterprises have their own backup solutions in place. A security-based question arose regarding whether LivePC prevents the host OS from screen capturing the guest OS. Monika said MokaFive treats any data that is displayed on the screen as gone and not securable. She then noted that the only real solution to this problem is a trusted computing environment. The last question of the session was whether MokaFive offered a bare-metal install of its LivePC product. Monika answered that MokaFive initially developed a bare-metal installation before going to using the virtual machine player model, and it is still available.

UNIVERSITY ISSUES WORKSHOP

Summarized by Rowan Littell (rowan@hovenweep.org), with help from Josh Simon

In its fourth year, the University Issues workshop included 15 participants from a variety of higher education institutions, primarily in the United States, with representation this year also from Finland, Australia, and Slovenia. Schools represented ranged in size from a few hundred students to tens of thousands, some with single campuses and others with over 20 campuses. The format of the workshop was a semi-structured round table discussion; one participant described it as Advanced Topics for people in education.

One of the topics that generated the most discussion was organizational structure. Many larger schools have divisions between central computing and various academic and administrative departments. These divisions lead to tensions and challenges, such as who owns the network equipment within a research cluster or how cost recovery is performed for shared services. Being able to compare methods of dealing with these was a highlight of the workshop. Participants stressed the importance of good communications channels within IT departments and with the rest of the institution and having SLAs to structure support agreements with other departments.

An ongoing area of discussion from previous years was the outsourcing of core software services to such places as Google Apps or other hosted providers. One participant described her institution's project to provide the option of Google Apps for all students, noting that the challenges were only partially technical and that of greater importance is having a policy and governance structure for the outsourcing of a core IT service. Regarding such efforts, others brought up concerns about document-retention policies, particularly for public institutions, and information classification and protection. A slightly different area of concern, particularly regarding email, was the fact that students are starting to prefer newer forms of instant communication over email and not seeing the value in an email account provided by the school; some places have considered providing different communications solutions for faculty and staff versus students or even not providing students with local email accounts unless requested.

Identity management systems were mentioned at several times during the day, as they have implications for many of the other topics of discussion. Although some institutions are able to consolidate all authentication and identity management into one system such as Active Directory, others use a variety of solutions, including different LDAP implementations, Kerberos, and Shibboleth, most of which are tied together with local tools. Service authorization is still a problem area; traditional UNIX groups, even in LDAP, have limits. One institution is using Shibboleth-style entitlements and LDAP ACLs to limit access to services, and others are using Cosign in conjunction with single sign-on.

A number of other topics were discussed briefly, including virtualization systems, print management, feedback structures for users, and open source software. Several places are starting to use virtualization for production systems, usually lightweight servers (Web, DNS, DHCP); however, its use is sometimes limited by the political need to tie a particular project to identifiable hardware. Many institutions are inclined to use open source software; however, management still wants the kind of supportability and accountability that seems to come from commercial vendors.

The workshop closed with participants briefly describing some of the things that they've found that really work well, whether they're large systems or simple tools; the most popular suggestion in this discussion was the use of IPSca for free educational SSL certificates. In response to a final question, most participants said they expected to be working in academia a year from now.

GOVERNMENT AND MILITARY SYSTEM ADMINISTRATION WORKSHOP

Summarized by Andrew Seely (seelya@saic.com)

The Government and Military System Administration Workshop was attended by representatives from the U.S. Department of Defense, U.S. Department of Energy, NASA, Raytheon, CSC, Science Applications International Corporation, Advanced Concepts, Los Alamos National Lab, Internet Software Consortium, Equilibrium Networks, Makena Technologies, Cfengine AS, and USENIX. Although there have been .gov BoFs in the past, this was the first time a workshop with this focus 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 allowed diverse government and military organizations to come together in a unique forum; it's not common to have highly technical staff from DoD, DoE, NASA, and industry at the same table to candidly discuss everything from power supplies to policy. All expected to find similarities and hoped to discover exposure to new ideas, and no one went away disappointed. The day's specific agenda was developed in the weeks before the workshop through email, with each attendee providing a short introduction and identifying a specific goal that he or she hoped the workshop would address. The agenda was adjusted as the workshop progressed, in order to capture emergent topics.

While the workshop met goals for general discussion, it also produced several "lightbulb" moments that were taken away for action: Three potential corporate partnerships were established. Datacenter environmental modeling tools were introduced. Information on setting up a corporation for holding security clearances was shared. A thin-client bug fix applicable to a military command in Europe was uncovered. Useful ideas for productivity while awaiting a clearance and practical ideas for coping with frustrations at the constraints of the government environment were introduced by people who have discovered creative solutions to these hard problems.

The day started with introductions and a reminder that the environment was uncleared and that non-U.S. people were in the room. For system administrators outside the government sector this would seem like an unusual caveat, but for people who work in classified environments it is always a safe reminder 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 products, challenges, solutions, and future issues with multilevel security systems (MLSs), PKI and identity management systems, and infrastructure issues, including cooling, plumbing, and how HVAC in some cases has overcome CPU as a primary metric for procurement. Open source and software development issues in the government domain were also addressed, as were setting up and maintaining development labs and using virtual networking.

A short presentation on DNSSEC was provided by the ISC. OMB Memorandum M-08-23, which mandates the use of DNSSEC on all .gov domain systems by the end of 2009, was introduced and its impact discussed. New software opportunities and small business initiatives were discussed, with Cfengine AS and Equilibrium Networks representing two small companies who are posturing to do business with the government sector. This led to a detailed discussion on accreditation of software and systems and a survey of issues surrounding how a corporation can successfully interface with a government entity.

Organizational topics discussed included time management issues, general rules and regulations for systems and personnel in government and military facilities, challenges of a nonclassified government environ0ment, working with vendors who are unfamiliar with classified systems and environments, working with non-U.S. or noncleared service providers or customers, and the contractor experience (working for two or more masters while keeping the mission in focus).

Debate was opened about the presence of non-U.S. attendees in the workshop. Certain levels of security clearance require the reporting of foreign contacts, which could discourage some people from attending this workshop in the future. It was agreed by all that the presence of non-U.S. attendees did not in any way detract from any discussion, that no additional information would have been discussed given a U.S.-only room, and that the inclusion of our non-U.S. attendees contributed significantly to our overall discussion. It was agreed that any future workshops would explicitly state that non-U.S. attendees were welcome and that all workshop discussion would be unclassified and unrestricted.

All attendees presented what types of personnel their respective sites or companies are seeking to hire. Over half had positions to fill but almost all required clearances for defense work. DoE and NASA were not generally hiring, but defense organizations were. Hiring information and career Web sites were shared.

The final topic was to determine whether there would be sufficient interest in this workshop to repeat it at LISA '09. It was agreed that it was a valuable experience for all attendees and that all would support a follow-on workshop. This workshop was a small step forward in shaping our profession of system administration in the government and military sector.

ADVANCED TOPICS WORKSHOP

Summarized by Josh Simon (jss@clock.org)

Tuesday's sessions began with the Advanced Topics Workshop; once again, Adam Moskowitz was our host, moderator, and referee. We started with our usual administrative announcements and the overview of the moderation software for the five new folks. Then we went around the room and did introductions.

For a variety of reasons, several of the Usual Suspects weren't at this year's workshop. Despite this, in representation, businesses (including consultants) outnumbered universities by about 4 to 1 again; over the course of the day, the room included 5 LISA program chairs (past, present, and future, up from 4 last year) and 9 past or present members of the USENIX, SAGE, or LOPSA Boards (down from 11 last year).

Our first topic was a round-the-room survey of the biggest problem we'd had over the past year. Common threads include career paths, such as whether to leave system administration for management or development, stay in system administration and motivate both yourself and your employees in a stagnating position, or find a job that's a better fit; reorganizations and lack of structure; doing more with less; and writing tools to automate tasks.

Our next topic was a brief comment about the effective expiration of RAID 5 as disk sizes increase. When you have a 5+1 RAID 5 array of terabyte drives, recomputing the checksum during recovery requires reading 5 TB of data; any unrecoverable error means data loss. Using 2 TB or larger disks means that the odds of unrecoverable errors rise to 80% or higher. Andrew Hume said, "By the end of 2009, anyone still using RAID-5 storage on large drives will be professionally negligent."

We next discussed storage. There was some question as to the best way to make very large data sets available to multiple machines at the same time. Some sites are stuck with NFS version 3 because of interoperability issues. The consensus is that NFS is like VHS: It's not the best technology, but it's what we've got: Use it if you want it, or don't use it and write your own. If you're doing high-performance computing (HPC), GFS, Lustre, and ZFS may be worth investigating, depending on your requirements. The consensus on using iSCSI heavily in production server environments is "Don't."

Our next discussion was automation. We started with automation of network configurations, since all the good solutions now in that space cost money. There should be a free tool, such as Cfengine or Puppet, explicitly for networking: It should inspect your environment and all its configurations. The goal here is more about managing racks, power, load balancers, VLAN configurations, ACLs on the switches, NAT on the firewall, updating the monitoring (Nagios), and trending (MRTG) tools. Other automation tools mentioned include Augeas and Presto.

The mention of integrating with monitoring led to a discussion as to when it's appropriate to add a new server or service to your monitoring system. One argument is to deploy both service and monitoring updates at the same time, with alerts silenced until it goes into production because it tightly couples deployment and monitoring. Another argument is only to add the new service to monitoring when it's ready to go into production, although this is harder to automate in a one-stop-shop mode, since there can be a long time between initial deployment and going into production. There is no single right answer, since the pros and cons tend to be environment-specific.

After our morning break, we resumed with a round-robin list of the latest favorite tools. This year, the list included BLCR (Berkeley Lab Checkpoint/Restart), C++, Cfengine, git, IPMI (Intelligent Platform Management Interface), MacBook, pconsole, pester, pfsense, Roomba, Ruby, SVN, Slurm, TiddlyWiki, Tom Binh backpacks, virtualization (including OpenVZ and Parallels), and ZFS.

Our next discussion was on cloud computing and virtualization. We're seeing it more and more and are wondering where the edge cases are. It tends to work well at the commodity level but not for certain services (such as file services). Managing virtual machines can be problematic as well. Some people are too optimistic with what they think they can gain; some folks are over-allocating machines, which can lead to outages. Managing the loads is a hard problem, since the tools may not show accurate information. HPC shops don't see much gain from virtualization, since they tend to be very computation-intensive and the ability to relocate virtual machines to different physical machines may not be enough of a gain to be worthwhile. The consensus was that virtualization is more useful in smaller development and Web service shops, especially in providing QA or test environments that look like their production counterparts. It's also useful to try something new: Take a snapshot, then work on it, and if you destroy it (or the software install wipes it out, or the patch blows up horribly) you can roll back to the snapshot you took. Finally, server consolidation (especially in data centers) and reducing power consumption is a big driver.

We next talked about career satisfaction and the lack thereof. Some senior folks (in both engineering and operations sides of the shop) are writing policies instead of doing "real work." This is similar to the shift from technical to management career paths; it works for some but not for others, in part because the work itself is different and in part because the reward is different. There's some concern that, as we age, we may lose touch with technology, in which many of us have bound our self-identity or self-worth. This is more problematic for those without similarly inclined peers with whom to discuss issues. Part of the problem is also that we as a profession are moving from server- or service-focused roles to more of a business focus; we exist to keep the business running, not to play with the cool toys. Some people have come back to system administration from management and feel that having the experience on the business side has been a huge benefit and makes them better system administrators. Additional satisfaction can come from mentoring.

This segued into a discussion about when it's appropriate to break policies when they're preventing the work from getting done. The summary is that rewriting them to avoid the problem or amending them to allow for IT-based exceptions was the best course of action.

After our lunch break, we talked more about monitoring. The best practices seem to include using both black-box monitoring tools (in which closed systems pretend to be the user) and white-box ones (in which one collects statistics and analyzes them later). Also, keeping historical data around is required if you want to do any trending analysis or need to audit anything. One argument is to capture everything, since you don't necessarily know what you'll want next month or next year; however, the counter-argument to just capture what you need for business purposes has advantages of using less disk space and making data unavailable for legal discovery later. It all depends on what you care about, and what level of failure in data collection you can live with. Clusters have interesting challenges; how much of your limited CPU (and cache and network bandwidth and so on) are you willing to allocate to monitoring, since that impacts your ability to process real data? Monitoring should not be an afterthought but an integrated part of any solution.

It should be noted that monitoring, checking the availability or function of a process, service, or server, is a different problem from alerting, telling someone or something the results of a monitoring check. This led to a discussion about not getting alerted unnecessarily. In one environment, the person who adds the monitoring rule is responsible for documenting how the Help Desk escalates issues, with the last-resort rule of "Contact the developer." This becomes more complicated in multi-tier environments (e.g., if you are monitoring in development and QA as well as production) and in environments with no 24/7 support access.

Maybe 5 of the 30 attendees were satisfied with the state of the monitoring in their environments.

Our next discussion topic arose out of the previous satisfaction issues involving bad policies. The right answer in most cases is that the policies need to be fixed, and that requires escalating through your management chain. Most policies in this context boil down to risk management for the business or enterprise. Security as it affects risk management needs to be functional instead of frustrating; there needs to be an understanding of the business needs, the risks, and how to work both within and around the policies as needed. We need to move away from the us-versus-them mentality with security for things like this. This might even include getting written exceptions to the policy (e.g., "No downloading of any software is allowed, except for the IT group whose job it is to do so"). Note also that some suppliers are more trustworthy than others, so some stuff can be fast-tracked. Document that into the policy as well. Policies should have owners to contact for review or explanation.

Next we did another round-robin on the next big things on our plate for the next year. For us, it includes automating manageability, backing up terabytes per day, building out new and consolidating existing data centers, centralizing authentication, dealing with globalization and reorganizations, designing a system correctly now to deploy in three years, doing more with less, excising encroaching bad managers from our projects, finding a new satisfying job, mentoring junior administrators, moving back to technology from management, remotely managing a supercomputing center, rolling out new services (hardware, OS, and software) the right way, scaling software to a hundred thousand nodes, transitioning from a server/host-based to a service-based model, virtualizing infrastructure services, and writing policies.

After the afternoon break we had a brief discussion about IPv6. A little fewer than half of us are doing anything with it, and those mostly on the client side. The consensus is that there's no good transition documentation explaining what providers need to do to transition from v4 to v6. It was noted that you need to say, "Here's the specific thing we need IPv6 to accomplish," then you'll be able to move forward instead of being thought of as the crazy one.

Next we discussed chargebacks; people seem to find it mostly useful. Some places have problems with their internal auditors. It was noted that chargeback encourages perverse behavior, such as doing what's easiest to measure and not what's necessarily useful or desired. Also, some departments tried to charge the time it took to convert to a new system to the department that rolled that system out. Some want to use chargebacks to provide accounting and force departments to forecast quantities, such as CPU time or disk space utilization. Charging for the technology resource at some rate (such as per CPU hour or per gigabyte per month) tends to work well, but that cost needs to include the human cost and yet not be so large as to discourage users from using your service.

Our next discussion was on professionalism and mentoring. How do we attract new blood into system administration? There's no good answer; in some environments, clearances are needed; in universities, many of the technically interested people go into development rather than system administration. Hiring student interns who want to be system administrators can help (if you're in a position to hire students), or going to local user groups, but good people are hard to find.

It may be that market forces will help; the demand for system administrators will drive up salaries in the long run. In tandem, recent articles mention that system administrators, network administrators, and database administrators are recession-proof jobs. But money talks. However, it's hard to get quality if you're interested more in money than the work. There's also conflation of the term "system administrator": Is it working with big, cool machines, or supporting users, or fixing printers, or being the computer janitor? People are starting to recognize that professionalism is important. Expectations as to IT staff behavior are higher than in the past: Use full sentences, be polite, answer questions, and help solve problems.

This boils down to how we get people into the profession. They're already maintaining their own desktop and so they're not seeing any of the cool side. People come in through the help desk and programming, but what other vectors are there (and how common are they)? It used to be hard to provide certain services that are now trivially easy. For example, mirroring is easy now (using rsync and cheap disk).

Our last discussion was on power consumption and "green" computing. Many places are running out of space, power, or both, and they need to increase efficiency. Most non-HPC places are just starting to look at the whole issue, although there's general consensus that it makes sense, both in terms of environmental issues (reduce, reuse, recycle) and economic issues (lower power bills with more instructions per kilowatt). Suggestions included defaulting to duplex printing, powering off desktops and monitors overnight, raising the cold aisle temperatures one degree in your data centers, running three-phase 208V power, virtualization of services that can be, and not allowing "throw hardware at it" as a solution. Low-power CPUs and variable-speed disk drives may help out as well.

This year's Talkies Award goes to DJ Gregor. Last year's winner, David Williamson, was not present; Andrew Hume, a former Talkies Award winner, was in the bottom five this year. (On a personal note, I actually managed to get my name in the speakers' queue on a relevant issue, surprising our moderator.)

VIRTUAL INFRASTRUCTURES WORKSHOP

Summarized by Kyrre Begnum (Kyrre.Begnum@iu.hio.no)

Virtualization was a key topic at this year's LISA conference, with virtualization-specific tutorials nearly every day. Paul Anderson decided to run a workshop with virtual infrastructures in mind. The workshop aimed at identifying the present-day challenges in integrating and running virtualization in large infrastructures. He did a lot of work during the planning phase to get people of different fields to give short presentations. In the end Kyrre Begnum chaired the workshop.

After the presentations we did some quick polls to identify the types of attendees. The group could be divided into three general classes: practitioners, who currently were using virtual machines (often on a large scale), researchers, whose main concern was management of virtual machines and automated service deployment, and sysadmins, who were going to deploy virtualization at some point and wanted to learn more.

Most of the practitioners were using more than one virtualization technology. Everyone believed that the number of virtual machines was going to expand in the future.

The workshop was organized around short presentations with following discussions. The presentations were divided into three subjects: deployment and performance, service management, and virtual machine management. The first subject was initiated by Gihan Munashinge, who presented his real-life experience with hosting virtual machines for customers. This talk helped set the tone for the rest of the workshop. Some very important discussion topics surfaced quickly, such as storage and lack of technology-independent management tools. All practitioners considered storage a major factor in the success of the virtual infrastructure. Three dimensions of storage were discussed: reliability, performance, and management. Most large infrastructures depended on redundant storage. ISCSI and NFS were common, but with low performance in the latter. Some had created their own storage solution, such as the layered approach used in the STORM project (see the article in this issue of *;login:*).

Next, Lamia Youseff presented performance results from using Xen virtual machines for HPC clusters. The lack of significant performance penalties intrigued the audience, and the discussion turned toward comparing experiences and impressions on real-life performance of VMs. One interesting topic here is the way in which VMs underperform compared to a traditional hardware-based server. Performance degradation appeared to be more dramatic after a certain threshold was crossed. The lack of publications comparing performance of different technologies was discussed briefly.

Deployment issues were laid to rest and focus shifted toward deploying services and approaches to create autonomic tools. The first presenter was Andy Gordon from Microsoft Research. The focus was on describing both the setup and the operational logic of a running service. A prototype system was presented, Baltic, where the overall functioning of a service was described in F#. Features such as automated scaling were supported and could be described in operational terms. Along similar lines, Nigel Edwards from HP Labs presented his experience deploying SAP on virtual machines. He shared with us some interesting real-life issues with dynamic services and cloud-like scenarios, such as added complexity in management, software licensing, and loss of control. Both presentations illustrated the potential in automated scenarios, but most practitioners used manual operations today to roll out new VMs. For some, scripts or configuration management tools inside the virtual machine would do the individualization of the VM.

Licensing was also discussed in this context. Many licenses were VM-unaware; this created problems for sysadmins. One example is a license that is hardware-profile aware. In such a case, moving the software over to a VM from a physical server would be problematic. Also, cloning VMs would potentially violate single-copy licenses.

The last topic was management and security. Anna Fischer from HP Labs talked about how to achieve secure communication among virtual machines, even when they are on different servers. Her architecture used MAC-address rewriting to create transparent communication among individual virtual machines. Several networking-related problems were discussed in relation to Fischer's work (e.g., the problem of inserting security tools into the servers in order to protect virtual machines). Further, enabling QoS on the network in order to quench VM traffic was discussed. Most practitioners did not enforce QoS on the virtual machines; instead they had several separate networks: SAN, management, and LAN.

Richard Elling from Sun talked briefly about reliability and fault tolerance in virtual infrastructures. He then proceeded to discuss bundling demos into virtual machines with regard to a new storage product released by Sun.

Kyrre Begnum talked about approaches for virtual machine management. His argument was that creating architectures for load-balancing services and virtual machines was very difficult, and he saw little adoption by the community. A different approach would be to put much of the monitoring and decision-making into the virtual machine itself, letting the underlying servers play a more passive role. There was a lively discussion around this approach, where trade-offs between the two approaches were analyzed. Many found a so-called hybrid approach interesting, where the virtual machines assisted the decision-making of the servers based on their individual policies.

This workshop provided an excellent opportunity for practitioners, researchers, and curious minds to exchange ideas and experience. The discussions were fruitful, with most of the 27 participants chiming in on various subjects.

Management of virtual machines seemed to be one of the key issues for most practitioners. Decoupling the management interface from the virtualization technology would be one way in which different management approaches could be tried on the same infrastructure without switching the underlying virtualization layer. Describing the behavior of services on a high level and transforming this description into real deployments are research challenges. Still, people from each camp came together in breaks and continued discussions also after the workshop. Many were interested in keeping in touch later and updating each other on new developments.

Our thanks are owed to Patrick Ntawuyamara and Nii Apleh Lartey for taking notes during the workshop.