;login:
APRIL 2013 VOL. 38, NO. 2
ELECTRONIC SUPPLEMENT

Conference Reports

LISA '12: 26th Large Installation System Administration Conference

San Diego, CA

December 9-14, 2012

Opening Remarks and Awards

Summarized by Rik Farrow (rik@usenix.org)

Carolyn Rowland, chair of LISA 2012 and appearing as energetic as ever, began the conference by saying that more than 1000 people attended LISA. Twenty-two papers were accepted for the papers track, with awards going to the Practice and Experience paper Lessons Learned When Building a Greenfield High Performance Computing Ecosystem by Andrew Keen et al., the best student paper going to Theia: Visual Signatures for Problem Diagnosis in Large Hadoop Clusters by Elmer Garduno et al., and best paper to Preventing the Revealing of Online Passwords to Inappropriate Websites with LoginInspector by Chuan Yue.

Next, John Mashey, appearing in a video, accepted the Lifetime Achievement award. Mashey is known for his work on the Programmer's Workbench (PWB) in the late '70s, contributing to the SPEC benchmark, the design of the MIPS RISC processor, and Silicon Graphics supercomputers. Arthur David Olson received the Software Tools Group award for his work on the Timezone DB. The LISA Outstanding Achievement award went to the developers of PowerShell: Jeffrey Snover, Bruce Payette, and James Truher. Finally, Phil Kizer, President of LOPSA, presented the Chucks Yerkes Award to David Lang for his work on the Linux kernel, rsyslog, and other projects.

Keynote Address: The Internet of Things and Sensors and Actuators!

Vint Cerf, VP and Chief Internet Evangelist, Google

Vint Cerf began by saying that as an Internet evangelist, he still has much work to do: the Internet has not yet reached everyone. Using domain names as a metric, there are 908.5 million machines visible on the Net, and 2.405 billion users. Only 1.5 billion of these are PC users, with much of the rest being users of mobile phone and devices.

IPv6 support got a lot better after the flag day: June 6, 2012. Today, about 25% of sites are visible via IPv6. With IPv4 addresses almost completely exhausted, IPv6 adoption must grow beyond the use of network address translation devices, which are fragile and don't do the job when cascaded.

ICANN (Cerf had been on the ICANN board for years) spent a lot of time and energy on getting support for Unicode for internationalized names. More recently, ICANN has collected more than $350 million in fees for non-generic top level domains, something Cerf said he is still skeptical about. (After all, how many people do much typing, especially of long domain names?) Cerf pointed out that DNS still has vulnerabilities and weaknesses, and that DNSSEC with its digital signatures will help. Cerf also mentioned using Digitally-Signed Address Registration (RPKI) to protect Internet routing from a serious vulnerability in BGP4, which has been around for decades.

Cerf commented that back in the early days (1980s), people joked about Internet-connected toasters. Today, we have Internet-connected picture frames and even light bulbs with IPv6 addresses. Cerf described how he monitors his wine cellar for temperature using a device that sends him text messages if the temperature goes over 60. He once received a message every five minutes for five days while he was traveling. Cerf also pointed out that his next steps would be adding RFID to each bottle, so that removed bottles get noted (he has teenagers!), and later planned to add sensors to the corks to monitor changes in wine caused by loss of temperature control.

As people and businesses add more sensors, Cerf told us that we need to be considering issues of authentication, authorization, security, along with ease-of-use. If you consider large environments, like a factory, it's not trivial to configure and manage a large network of devices (he mentioned Arch Rock's mesh networks). And what about devices in the home? If you allow auto-registration, what's to stop your neighbors from registering your devices? Who will you allow to monitor your devices, perhaps to add (not subtract from) your security? If these devices are wireless, which is much simpler, each needs its own address. Grouping devices by a controller (Arch Rock) seems like a good model, similar to the way we use ASNs today. Cerf included set-top boxes as other devices also in need of configuration.

Cerf is also a member of the Smart Grid Interoperability Panel (SGIP). While many US citizens consider having smart meters that can both monitor electric usage and (eventually) disable high current devices distasteful, Cerf pointed out that we use peak power only 2% of the time, but we pay to build out our generating capacity to support this tiny fraction of usage at great cost.

Cerf discussed the recent attempts to change how the Internet is governed. Certain countries had attempted to use the International Telecommunication Union as a forum to wrest control from nation-independent entities, such as the IETF, and to a lesser extent ICANN, so they can create new standards. Not that these standards will be "real," as their real purpose is

control, and the level of control desired already exists. You don't need a standard for doing deep packet inspection other than the existing standards that allow the Internet to work. Other policy challenges that exist today include the meaning of digital certificates, intellectual property, and preservation of data and software (Digital Vellum).

Cerf brought up other challenges to the Internet, some having to do with the future of routing (OpenFlow and BGP), rethinking the use of certificates and authorities, the role of trusted computing (TCM and the requirement to sign operating systems digitally), and inter-cloud protocols. He finished on a high note, by discussing the InterPlaNetary Internet (his capitalization). Because of the huge amounts of data acquired by remote sensors, such as Mars rovers or Cassini, and the large amount of time it takes for light to travel from the outer planets to Earth, new techniques are required. One thing that has worked so far is to store-and-forward messages, repurposing existing satellites—for example, in Mars orbits—to collect messages from surface-bound rovers, then send them using the more powerful radios. I found myself thinking, "What a great way to take advantage of bufferbloat!" and the reality is not that far off. The delays inherent in interplanetary TCP/IP really require different protocols, such as Custodial File Delivery Protocol.

Cerf only had time for a single question at the end, partially because he was urged to keep on speaking. When Patrick Cable came up to the mike, Cerf walked off the stage so he could watch Patrick ask his question, as Cerf said he has trouble hearing. Patrick asked Cerf about his thoughts on regulation in general, and are there regulations that make sense. Cerf responded that there are areas where international regulations do make sense. We can't do much about spam or Internet-based crime without the support of international law. We need international cooperation for many things. Then there are times when informal cooperation works best, like the organizations that worked together to track the Conficker botnet.

After his enthusiastic stump speech, Vint Cerf received a standing ovation from his equally enthusiastic audience. You can view the video or download the audio of this and the other presentations at www.usenix.org/conference/lisa12/tech-schedule/technical-sessions.

Papers and Reports: Storage and Data

Summarized by Lin Sun (sunlin530@gmail.com)

HSS: A Simple File Storage System for Web Applications

Daniel Pollack, AOL Inc.

Daniel Pollack explained that all Web applications need some sort of durable storage system to hold the content and, in some cases, the code that runs the Web application. At AOL, they looked at a variety of existing solutions, including cluster file systems, scalable NAS, and parallel file systems before deciding to build their own solutions. Their first attempt was iBrix, but it had both performance issues and required client-side support. Their second attempt was to build an object store using commodity hardware and open source software. Based on these experiences, they came up with a list of requirements, including scalable metadata, separate metadata, and data system components, both multi-site and multi-tenant capable.

The storage system presented seeks to improve on the availability and operational characteristics of the storage systems. A minimal set of operations are provided and they rely on external components for any additional functionality that an application may need. Additionally, several mechanisms are built into the system that provide data durability and recovery—for example, being aware of the physical makeup of the system for both reliability and hotspot reduction.

HSS uses MySQL for metadata storage, and stores content as objects. Each object is replicated, and the location of objects is updated in the MySQL database. A simple RESTful external API is presented to clients, and HSS fulfills requests.

A list of future planned improvements could be container files to address file management and performance concerns, lazy deletes to disconnect housekeeping operations from online operations, improved geographic awareness to improve access latency, and a policy engine to manage file placement and priority in the system.

IDO: Intelligent Data Outsourcing with Improved RAID Reconstruction Performance in Large-Scale Data Centers

Suzhen Wu, Xiamen University and University of Nebraska-Lincoln; Hong Jiang and Bo Mao, University of Nebraska-Lincoln

Bo Mao began by saying that there is much more disk failure in the real world than we used to imagine. Generally speaking, the complete disk failure rate is 2% to 4% on average, and after one disk fails, another disk failure will likely occur soon.

Due to these challenges, RAID reconstruction tends to be much more important to system reliability. There are two challenges for RAID reconstruction: real-time user performance and window of vulnerability. Diverting many user I/O requests from the degraded RAID directly affects the reconstruction performance.

The existing reconstruction approaches can be categorized into two types. The first type of reconstruction optimization improves the reconstruction performance by optimizing the reconstruction workflow, such as DOR, live-block recovery, and PRO. The second type improves reconstruction performance by reshaping the user I/O requests, such as MICRO, Work Out, and VDF. Based on new observations, they found that these optimizations are ineffective.

IDO (Intelligent Data Outsourcing), a proactive and zone-based optimization, can address this problem and significantly

improve online RAID-reconstruction performance. The main idea of IDO is to divide the entire RAID storage space into zones and identify the popularity of these zones in the normal operational state, in anticipation of data reconstruction and migration. Upon a disk failure, IDO reconstructs the lost data blocks belonging to the hot zones prior to those belonging to the cold zones and, at the same time, migrates fetched hot data to a surrogate RAID set.

IDO is an ongoing research project. They are working on the recovery algorithms in large-scale storage systems where the network bandwidth, the storage nodes, and the workloads are more complicated than the pure RAID-based storage systems.

Theia: Visual Signatures for Problem Diagnosis in Large Hadoop Clusters

Elmer Garduno, Soila P. Kavulya, Jiaqi Tan, Rajeev Gandhi, and Priya Narasimhan, Carnegie Mellon University

Awarded Best Student Paper!

Soila Kavuyla explained that problem diagnosis when using Hadoop is compounded by the overwhelming volume of monitoring data and complex component interactions that obscure root causes. Usually users want to distinguish between problems inherent in their job and problems due to infrastructure faults. Theia is a tool for visualizing anomalies in Hadoop clusters, targeting hardware failures, software bugs, and data skew. Its key requirements are an interactive interface that supports data exploration in which users drill-down from cluster- to job-level displays, a compact representation for scalability, and the ability to support clusters with thousands of nodes.

Theia's types of visualizations include anomaly heatmaps, job execution streams, and job execution details. The "anomaly heatmap" provides a high-density overview of cluster performance and summarizes job performance across nodes. It uses color variations to visualize anomalies. The "job execution stream" helps to visualize per-job performance across nodes and a scrollable stream of jobs sorted by start time. It displays performance of Map and Reduce phases and shows job execution traces in context: job name, duration, and status in addition to failed and killed task ratios and task duration anomalies. The "job execution detail" provides a detailed view of task execution but is less compact than the job execution stream. It displays job progress and volume of I/O; it is best-suited for detecting application problems, software bugs, and data skew.

Kavulya concluded that Theia visualizations for Hadoop are compact, interactive visualizations of job behavior. Theia distinguishes hardware failures, software bugs, and data skew in addition to evaluating real incidents in Hadoop clusters. Evaluating the effectiveness of a UI for diagnosis could be a further step taken by users.

Someone asked if they include tools for automatic problem classification. Kavulya said they are planning on adding those features. Someone else wondered how well they expect this to scale. Kavulya replied that they use Perl and batch processes, so this should scale. Marc Chiarini asked about the graphic display that uses sizes to indicate disparities across multiple jobs. Kavulya said that currently they just use visual clues to pick this out. Chiarini then suggested using a mouse-over script to provide more details on the node.

Invited Talks

OpenStack: Leading the Open Source Cloud Revolution

Vish Ishaya, Nebula, Inc.
Summarized by Andrew Hume (andrew@research.att.com)

Vish Ishaya started with an extended justification for clusters and clouds and the skunkworks-like genesis of OpenStack within NASA in April 2010. Things moved quickly: the first public cloud launched in October 2010 and Rackspace switched to OpenStack in August 2012.

He then installed OpenStack on his Mac laptop and started it running, logging into the console of a newly started VM. For many people, this was an amazing part of Ishaya's presentation.

So what is OpenStack? It is the APIs that let you manipulate Compute, Network, and Storage inside a cluster. OpenStack now has seven core components: compute, object storage, block storage, networking, (machine) image, identity, and dashboard. Vish gave brief overviews of each of these, and then touched on the management issues (550 developers) that led to the creation of the OpenStack Foundation.

He then described some of the projects in incubation, including heat (work on orchestrating groups of servers/VMs) and using bare-metal servers (and not just VMs).

Analysis of an Internet-Wide Stealth Scan from a Botnet

Alberto Dainotti, Cooperative Association for Internet Data Analysis
Summarized by Daniel-Elia Feist-Alexandrov (d.feistalexandrov@gmail.com)

Botnets are one of the most potent arrows in a cyber-criminal's quiver: not only are they responsible for large scale DDoS attacks, they can also be used to detect and exploit vulnerable machines on a massive scale. Alberto Dainotti presented the cooperative's analysis of a 12-day scan conducted by the Sality botnet against the SIP-calling infrastructure around the world. The scan Dainotti and his colleagues analyzed is exceptional not only because of its unprecedented size, but also because of its stealthy stratagem, which made it extremely hard to detect, despite covering the entire IPv4 address space.

The main tool Dainotti et al. used to identify the scan was the UC San Diego Network Telescope "darknet," a block of IPv4 addresses that are not assigned to actual hosts. Using a lot of "investigative" analysis and the fact that, by definition, every

packet that arrives at an address in this block is unsolicited, they identified a unique payload fingerprint and the UDP port 5060 as a common denominator of the scan. They determined that the 3 million IP addresses they registered were indeed unique machines by correlating their own data with that of the DShield project (an aggregator of dark and honeynet data) and data from a trans-Pacific link monitored by the MAWI/WIDE project. Another helpful fact was that all the bots that were geolocated to Egypt dropped out of the attack while the government suspended Internet connectivity during the Tahrir Square uprising.

Using a Hilbert curve and other visualization techniques to map the IPv4 realm to a two-dimensional space, Dainotti and his colleagues found that the scan's exceptional stealth was due to all bots choosing their next destination by incrementing target addresses in reverse-byte order. This meant that a generic /24 network would typically receive a total of 256 packets over 12 days from 256 different source addresses, thereby making it very hard to spot any connection between scans.

The first question concerned the fact that there was barely any scanning activity for several days during those 12 days. Dainotti confirmed that this was indeed due to a large number of bots halting their activity and speculates that this might have been due to sanitization efforts on behalf of law enforcement or antivirus companies performing routine botnet breakups. Another participant asked whether there were any similarities between the few bots that contacted an address in the dark net twice. Dainotti answered that there were no similarities. A possible explanation was that this was due to different versions of the bot's binary.

Someone asked whether the authors estimated the cost of leasing such a huge botnet. Dainotti responded that they didn't and hypothesized that this might have been a factor in the scan's intermittent flagging. The next participant commented on the possibility that the turnover in the botnet might have also been a result of the generally short lifecycle of a bot (due to eventual sanitization). Another participant speculated that the lack of bot activity in China could be caused by the "Great Firewall of China" and the government's repression of VoIP infrastructure.

Papers and Reports: Security and Systems Management

Summarized by Tim Nelson (tn@cs.wpi.edu)

Lessons in iOS Device Configuration Management

Tim Bell, Trinity College, University of Melbourne

Tim Bell presented iOS Configurator, a Django Web application used by students in Trinity College's foundational studies program. Foundational Studies is a one-year college-preparatory program, and each student receives an iPad. iOS Configurator allows those students to download fresh configurations for their iPads. Their original approach to configuration used a combination of manual edits and Python scripts, but that didn't scale very well. They needed the tool to be automatic and scale to several hundred students while allowing them to reconfigure their iPads at any time. Also, they needed to implement the replacement quickly with limited staff. iOS Configurator was the replacement.

Bell showed a screenshot of the login process. After a student logs in, the configurator authenticates her, then gets her group information and fetches a standard configuration file for that group. The configurator adds user-specific information, then downloads the profile. The app provides an administrator page that says who has downloaded their configuration and when.

iOS Configurator comprises 167 lines of Python (including comments) and 229 lines of settings (mostly boilerplate), and took a week to develop. The login process uses HTTPS with a commercial SSL certificate.

After completing the one-year program, students get to keep their iPad. Thus, Trinity did not want to restrict the students' use of their iPads. Because of this fact, Bell opted not to use mobile device management for this configuration process. Bell commented that Apple has since come out with Profile Manager in OS X Lion, which he might have used had it been available when he was creating the configurator app.

Paul Anderson asked whether students could override the settings in the downloaded configuration. Bell answered yes, and explained that that was part of their goal. Also, students can always re-download the configuration.

A Declarative Approach to Automated Configuration

John A. Hewson and Paul Anderson, University of Edinburgh; Andrew D. Gordon, Microsoft Research and University

John Hewson presented the ConfSolve tool. ConfSolve converts configuration goals into concrete configurations. Existing declarative configuration-management tools let you specify what you want, rather than how to accomplish it. ConfSolve builds off of the CM tools by inferring valid configurations from goals that are only partially specified.

ConfSolve uses a CSP (constraint satisfaction problem) engine as its workhorse. The tool has its own object-oriented language that it compiles into a CSP, and the solution that the solver provides is then translated into a concrete configuration.

Hewson showed an example of ConfSolve working on a virtual-machine specification, which showcased the tool's ability to handle constraints (e.g., "When assigning VMs to physical machines, don't exceed the physical machines' resources") and optimization ("Use our data center as much as possible before using the cloud").

ConfSolve scales fairly well; it produced a configuration for a thousand virtual machines in around 200 seconds, and there is still room for improvement.

John then commented that ConfSolve is not a replacement for mainstream declarative configuration-management tools. Instead, he would like to see those tools incorporate configuration inference. Tim Bell asked about the complexity of the problem, and Hewson replied that the general problem is NP-Hard. Tim Nelson speculated that some kinds of configuration inference may fall into a less difficult class.

Preventing the Revealing of Online Passwords to Inappropriate Websites with LoginInspector

Chuan Yue, University of Colorado at Colorado Springs

Awarded Best Paper!

Chuan Yue presented the LoginInspector tool. Passwords are still the dominant method of authentication on the Internet, yet they are vulnerable to phishing, reuse, and more. Detection of phishing sites relies on either a blacklist or heuristics, and can thus miss zero-day sites. Moreover, users who have forgotten their password can expose other passwords accidentally by trying their "usual passwords" in sequence. Yue's user-study information showed that users really do engage in this risky behavior.

LoginInspector keeps a database of which passwords have been used when logging in to which sites. (For security, it keeps only hashes in the database, not full passwords.) If a user tries to log in somewhere that he has logged in before, but with a different password, LoginInspector intercepts the login and shows a warning message to the user. If the user has not logged into the site before (i.e., it is a potential phishing site) a similar warning message is shown. The tool is implemented in JavaScript as a Firefox addon, using the SQLite database.

Yue showed that LoginInspector has low overhead, taking the longest when inserting new records into its database. He evaluated it on 30 real sites, 30 phishing sites, and one new phishing site. LoginInspector correctly gave warning messages on all phishing sites, where Firefox's phishing detection failed to catch seven. Chrome's failed to catch eight.

Yue commented that the effectiveness of the tool depends on users' ability to understand and heed the warnings; he intends to perform user studies to evaluate that next. Someone asked, if the site has multiple domains, will that result in multiple records in the database? Yue answered that it would. Mario Obejas asked when the tool would be available. Yue replied that it should be available in January 2013.

Invited Talks

Database Server Safety Nets: Options for Predictive Server Analytics

Joe Conway, credativ USA; Jeff Hamann, Forest Informatics, Inc.

Summarized by Cory Lueninghoener (cluening@gmail.com)

Joe Conway and Jeff Hamann started their session with a simple statement of their goal: to perform Postgres server monitoring using predictive analytics; however, they also noted that this project is really a wrapper around the underlying topic of using Postgres and R to do analysis of big data. With the stage set, they dove in to a technical description of how they are predicting congestion events on their database servers.

Joe began with a description of the tools they are using to perform their analysis: Postgres, a modern database server; R, a popular analytics engine; and PL/R, a module that runs R procedures inside of a Postgres process. He then listed the wide range of Postgres metrics they collect to perform their analysis: active and total Postgres sessions, blocks fetched and blocks hit, cache hit fraction, lock waits, free and cached memory, free swap space, I/O wait and CPU idle, blocks read and written per second, number of blocks read and written, and capture time.

After describing the tools and metrics they are interested in, Joe began a technical description of how the metrics are collected. He included several slides of PostgreSQL and R code examples that showed how the data is collected from the Postgres process, how it is automatically inserted into the database, and how R is used in this process. Joe also noted that the metrics gathering is triggered by a simple cron job, a decision they made to make the process simple, reliable, and transparent.

Following the technical dive, Joe described their method for testing their analysis process. This involved using pgbench, a tool that comes with the Postgres distribution, to simulate steady-state load and transient events on their servers. With the metrics collection pieces in place and a method of simulating events ready, the team was ready to start doing predictive analytics.

Jeff took over at this point to describe their methods. He started by stating the problem: can we do preemptive analytics work to sense when a server is going to experience congestion? By looking for causal factors, correlations, and leading indicators of system congestion, Jeff hoped that they could do just that.

After introducing their methodology with an example matrix plot and a series of plots showing correlation and time series data, Jeff showed a real example of their work using two basic metrics: swap and the number of active and total Postgres client sessions. He started by showing several graphs of this data, and then described how they built an initial model for this data using R. After comparing this initial model with the real data, he then described how they improved the model to get a better fit.

Once the model was complete, Jeff showed how it could be used to make predictions using principal component analysis and K-means clustering. This included a description of the built-in R functions that make this easy and several graphs that demonstrated its use.

Finally, Jeff gave a brief description of statistical process control and how it relates to predictive server analytics. He described R's statistical process control package, qcc, and how it can be used to glean more information from collected server metrics.

After describing their future work plans involving harvesting more data from the Postgres server, doing pattern recognition, and polling multiple servers, Joe and Jeff took questions. One attendee asked whether they were familiar with Baron Schwartz's work to collect data from a failure automatically. The speakers were not familiar with it, but thought it sounded interesting. Another attendee asked whether the source code for their project was available. Joe replied that it was not yet posted, but it would appear on joeconway.com after the conference. [Editor's note: Both code and slides were present on January 8, 2013]

Ceph: Managing a Distributed Storage System at Scale

Sage Weil, Inktank

Summarized by David Klann (dklann@linux.com)

Sage Weil wrote the Ceph distributed storage system and described it in this invited talk. Sage presented an articulate overview of Ceph and answered questions as if he wrote the software (see previous sentence).

Weil began his talk with a very brief historical roundup of storage systems he called "the old reality": directly attached storage, raw disks, RAID, network storage, and SAN. He quickly moved on to discuss new user demands including "cloud" storage and "big data" requirements. Requirements that include diverse use cases such as object storage, block device access, shared file systems, and structured data requirements. Scalability is also on the requirements list, including scale to exabytes on heterogeneous hardware with reliability and fault tolerance, and a "mish mash" of all the above technologies. And with all this comes a cost. Cost in terms of both time and dollars. Weil proceeded to describe these costs and then to describe Ceph itself.

Ceph is a unified storage system that incorporates object, block, and file storage. On the Ceph architecture slide, Weil showed the distributed object store base he calls RADOS, for Reliable Autonomic Distributed Object Store. Above RADOS live the API libraries and other interfaces to the object store: LIBRADOS (with the expected array of language support); RADOSGW, a REST interface compatible with Amazon's S3 and OpenStack's Swift; RBD (RADOS block device), the distributed block device; and Ceph FS, a POSIX-compliant distributed file system with Linux, a kernel client as well as a user-space file system (with FUSE). Weil emphasized the distributed nature of the Ceph system noting that Ceph scales from a few to tens of thousands of machines and to exabytes of storage. Weil noted that Ceph is also fault tolerant, self-managing, and self-healing. He pointed out that the collection of Ceph tools is an "evolution of the UNIX philosophy" in that each tool (control command and daemon) is designed to perform one task and to do it well.

Weil moved on to describe Ceph cluster deployment and management. He noted that the Ceph developers are working closely with the major Linux distributions to package the tool set for easy deployment. Ceph supports clusters with mixed versions of the code by checking program version numbers in regular internode communication. This facilitates rolling upgrades of individual cluster participants. The protocol also includes "feature bits," which enable integration of bleeding edge cluster nodes for the purpose of testing new functionality.

The Ceph configuration philosophy is to minimize local configuration. Options may be specified in configuration files and on the command line of the various tools.

System Log Analysis Using BigQuery Cloud Computing

Gustavo Franco, Google Inc.

Summarized by Nick Felt (nfelt1@sccs.swarthmore.edu)

Gustavo Franco presented on Google's BigQuery service and how system administrators can apply it to speed up log analysis. This makes it easier to use logs not just for troubleshooting but also to drive product enhancements. Google has used BigQuery internally for a few years, and they've only recently opened it up to external use as an official product, so it's not yet widely known. Gustavo pointed out that traditional log analysis is tiresome because one writes analysis code and has to wait a few hours to get a response (in the case of large systems with lots of logs), which also makes fixing bugs in this code a day-long process. Using BigQuery, this process takes a matter of seconds, even for petabytes of raw data.

In comparison, Gustavo noted that other approaches to log analysis do not scale as well. Just using grep alone will not suffice once the setup involves several servers. Past that point, one can get by for a while sending log data to a MySQL database, but eventually the influx of data becomes so large that writes start to interfere with reads. One might consider the MapReduce distributed computation framework as an alternative, or the Sawzall programming language, which provides a script-like way to write MapReduce code (both developed by Google); however, although MapReduce is very flexible and useful for data analysis in general, it's a heavyweight solution for log analysis. For each MapReduce execution, the master has to spin up many mappers and reducers, each of which read and write to distributed storage, resulting in a significant time delay at the start in order to spin up workers and a lot of worker I/O overall.

BigQuery improves on all of these approaches by using Dremel, an internal Google framework explicitly designed for fast data analysis. Dremel uses a separate system to handle log injection, so this process doesn't interfere with running queries. On the query execution end, the major difference between Dremel and MapReduce is architectural: Dremel trades flexibility for raw power, allowing it to speed past MapReduce for certain kinds of data sets and queries. Dremel maintains a long-lived shared serving tree with always-running nodes that do not need to be spun up and can execute many queries at the same time. Each query starts at one of many top-level Mixer 0 nodes, which then sends requests to several Mixer 1 nodes, each of which farms out its portion of the request to many leaf nodes. The leaf nodes access distributed storage containing the data in records split up by column, a trick that speeds up the query. Then the leaf nodes send results back up the tree through Mixer 1 nodes to the Mixer 0 node, which reduces the data into a single result set. All data flow outside the leaf nodes occurs via RPC message passing and does not touch the disk, cutting I/O delays substantially.

At this point, Gustavo showed a live demo of BigQuery using the Web UI to execute a number of example queries on large sample data sets of dummy Web server and system logs. BigQuery uses a SQL-like query syntax intended to look familiar to users, and has a command-line UI and an API in addition to the Web UI. One of his example queries was "SELECT COUNT(*) as rows FROM Weblogs.lisa10," which took only 3.0 seconds to execute. The same query executed on the lisa163M table (which has 163 million rows instead of 10 million) took only 0.4 seconds longer. In another query, Gustavo demonstrated the ability to group Web requests to the top hits, which processed three gigabytes of data in 20.7 seconds. He emphasized that nothing in BigQuery is cached or indexed; the data is freshly scanned for each query. Someone in the audience asked whether BigQuery supports joins, and Gustavo said that it does if you establish the relationship in your logs, but the left side of the join must be smaller. Toward the end of the demo he also mentioned that BigQuery supports regex matching and various other features.

Gustavo wrapped up the presentation by explaining how system administrators could start using BigQuery to analyze their own Web server, application, and system logs. The first step is downloading the "bq" and "gsutil" command-line tools. For ongoing use, Gustavo recommended using logrotate and sharding logs into daily tables to improve performance unless the data set is fairly small. Logs should be uploaded to Google Cloud Storage in either CSV or JSON format, optionally gzipped. There was a question about the lack of support for syslog and other log formats; Gustavo said it's a work-in-progress. Once the data is uploaded, run "bq load" with a few arguments specifying what data to use and providing information about the columns, then you're ready to query away. You can even use the Google Visualization API to generate plots of results. Pricing information for BigQuery and Google Cloud Storage is available online, and both have free tiers as of December 2012. For BigQuery, the first 100 GB of data processed per month is free, and the cost per query is based just on how much data the query touches, not on the overall data size. Gustavo directed those who want to learn more about putting BigQuery to use to consult his "homework" page (http://goo.gl/JkhFC).

During Q&A, someone asked for elaboration about how Dremel scales, and what is processed by the leaves versus by the mixers. Gustavo responded that the leaves are only ones touching the shared storage; they do most of the data crunching and send results back to the mixers via RPCs, with different kinds of aggregation happening at different levels. Another attendee asked for the most interesting thing Gustavo had heard of someone doing with BigQuery. Gustavo said the Ads group at Google makes heavy internal use of BigQuery, but wasn't able to elaborate beyond "for cool stuff." Nick Felt (Swarthmore College) asked whether BigQuery can return an estimated time until completion for queries. Gustavo replied that it wasn't possible to propagate this information back up the tree, but queries are usually pretty fast to complete, although with large data sets or especially complex queries they can take longer than a minute. Someone else asked what to do to upload logs from an app with an idiosyncratic log format. Gustavo replied that any format is fine as long as it can be converted to a columnar structure, and the columns are given names and data types when loaded into BigQuery. Someone asked for a comparison with Splunk, a competing tool, but Gustavo declined to comment since he hasn't used it.

Plenary Session

Education vs. Training

Selena Deckelmann, PostgreSQL

Summarized by Jessica Hilt (Jhilt@ucsd.edu)

This talk concerned the controversial issue of formal education verses on-the-job training. Sysadmins might be divided about the topic to the extent that they talk about it at all.

Selena Deckelmann began with the continually pressing problem of scalability: we can't hire the numbers we need in the sysadmin field and we can't train people fast enough. Looking to the formal university setting, we see the ability to train larger numbers but we don't see the classes designed for sysadmins. On the job, we see the necessary skills being taught but only on a one-on-one basis. Certification programs tend to be scoffed at, and books and blogs tend to be the resources to which sysadmins turn but lack effectiveness in training large numbers.

Deckelmann explored the reason why sysadmins dismiss the university setting. Citing bad teachers, ineffective classes, and abstract theories, Deckelmann agreed that there is a lot not to like; however, Deckelmann cautioned against this mentality.

Instead of labeling formal training settings as snobby or impractical, she suggested we figure out how to share with teachers and universities what we need from them in order to make a great sysadmin and to bridge the gap between the training world and the education world.

In this vein, Deckelmann outlined steps to make a systematic training program that is effective. She started with a method she learned in a one-on-one setting early in her career. Use (1) defined steps with measurable outcomes; (2) explicit instructions with immediate feedback loops; and (3) pairing and modeling.

In outlining training in such a way, Deckelmann says, we are defining what success looks like to a student.

Next, Deckelmann sought to apply this method to a larger number of students taught at once, using a case study of teaching Python to non-programmers. She established a baseline for training by defining what the student comes into the class expected to know, teaching the gaps, and then having the students demonstrate the knowledge to the rest of the students. With this method, Deckelmann explained, the class proficiency rises.

Deckelmann made one strong recommendation throughout the presentation: In order to find solutions to the education debate, you need to start a fight with sysadmins about it today. She recommends fighting about the details of education (i.e., ethics versus risk reduction, nonprofit certification versus masters programs) in order to have an argument of value. Additionally, she promoted sharing existing training material and programs so that others can learn from your success, and stressed that we can't wait for people outside the industry to solve this problem.

This talk generated numerous questions as well as comments about current training programs or certification training programs. A questioner asked whether there was a current degree program that was respected for system administration. Deckelmann pointed to the Rochester Institute of Technology as a model. Another questioner asked whether requiring formal training would decrease diversity in the field and Deckelmann referred to the Py Ladies program as an example of increased diversity due to formal training. A questioner asked if there was a group that was discussing this for further conversation and they were referred to Carolyn Rowland who was creating a list.

Papers and Reports: Community and Teaching

Summarized by Barry Peddycord III (bwpeddyc@ncsu.edu)

A Sustainable Model for ICT Capacity Building in Developing Countries

Rudy Gevaert, Ghent University, Belgium

Rudy Gevaert discussed the efforts of his institution to improve the state of IT in the universities of developing countries. In the spirit of the other talks of the Community and Teaching track, the talk was not a technical talk, focusing instead on the human side of technology and computing.

Gevaert's university has taken part in an initiative to travel to universities in developing countries and improve their IT capabilities. Unlike many such initiatives that focus on delivering computing equipment to these universities, this initiative takes the efforts a step further by taking an active, hands-on role in training and mentoring the sysadmins of these developing institutions by teaching them how to utilize and troubleshoot the equipment they are given. As Gevaert stresses, the focus of the effort is not to build infrastructure, but to build capacity by focusing on ensuring that they are able to train the participants in these programs to become effective administrators and mentors to their peers.

One of the projects undertaken is to build an in-house email service or Web service. We take for granted the saturation of cloud applications for these purposes such as Google sites and Gmail, but in these nations, this saturation simply is not an option. Gevaert alluded to Vint Cerf's LISA keynote, where Cerf mentioned that the bandwidth of an interplanetary Internet connection might be 500 Kbps at best. In Cuba, one of the nations involved in this initiative, this is not a joke—it's reality. As bandwidth is extremely limited, efforts to conserve that resource are among the top priorities, so in-house solutions are preferred to cloud solutions, and filters to prevent extensive recreational usage of bandwidth must be put in place. These efforts help facilitate these practices and more.

Above all, the one takeaway from the talk was that any outreach effort like this absolutely must be designed with sustainability in mind. Volunteer efforts lose members for one reason or another and outreach initiatives lose government funding, meaning that the volunteers absolutely must not become a dependency. System administrators in developing countries must be trained to become self-sufficient and able to solve problems in their constrained environments. Turning them into mentors is important so that when the intervention ends, they can train their own colleagues.

Teaching System Administration

Steve VanDevender, University of Oregon

One doesn't have to be faculty to make a difference in teaching at a university. Steve VanDevender took the initiative of leading his own course in introductory system administration and presented his practice and experience report on how he developed the course and how it worked out.

A point that VanDevender stressed was that "you can't teach everything you want to teach." It's essential not to be too ambitious, and to focus on specific and attainable learning goals that can be accomplished in the time frame of the course being

taught. That being said, he still took risks, such as allowing students to work in teams, choose their own OS, and design their own final project for the course. Even though these were risky decisions, they ended up being very rewarding as the students really enjoyed being able to work on projects that they thought would be meaningful.

In his class, VanDevender also had the opportunity to do the opposite of what he disliked about classes when he was a student. He therefore focused on clear and well-defined assignments where the grading would be explicitly linked to the outcomes of the course and the objective performance of the projects. Research has shown that when assessment is objective and transparent, students are more likely to respect the instructor and take the course seriously. Despite the class being challenging, students appreciated the multi-modal learning, from reading chapters of the textbook to discussing materials in class to having the hands-on experience of working with their personal system—even when it came time for the surprise "System Failure" day.

One audience member highlighted that in university teaching, instructors don't get the guidance and support that they might expect. VanDevender's institution did not have as much oversight as he first thought they would, meaning that while he had a lot of freedom in how he led his class, he did not get much in the way of feedback and formal training. Many institutions have a Center for Faculty Development that offers workshops and mentorship to help improve teaching, which system administrators may find useful if they decide to attempt to take the initiative and do something similar at their own institutions.

Training and Professional Development in an IT Community

George William Herbert, Taos Mountain, Inc.

Professional development services are a major part of many companies. George Herbert shared the story of how his company treats professional development as a major company value and has offered such services to their consultants and contractors throughout the years, even in the face of the recent economic downturn.

Professional development services manifest themselves in many different ways. In addition to inviting guest speakers and providing mentoring, companies can offer reimbursements for taking classes or buying books, providing Safari accounts, and subsidizing travel to professional conferences like LISA. By subsidizing such services, companies keep their employees well-rounded and up-to-date on the latest technological developments.

Many companies don't treat their professional development services as the valuable asset that they are. In a field where retaining talent is so important, the attitude toward professional development can be a differentiator. Many employees find the fact that companies offer consistent professional development opportunities to be a reason to spend more time with their company rather than seeking another or going freelance. Furthermore, the professional development services at Taos only cost the company about $100 per employee each quarter, not nearly as much as one might think. Given the impact they have on the skills and morale of their employees, they are worth the investment.

Herbert has done some initial analysis of the data from attendance sheets, compiling how well certain events have gone over. In general, employees prefer professional development with a social element to them, such as having guest speakers or having special classes on specific topics rather than buying books and resources for self-directed study. As much of the data has yet to be compiled in a way that can be easily analyzed, most of the lessons learned are anecdotal; however, this still leads to his big takeaway: any professional development initiative should be documented and measured so that its value can be clearly represented to the management in order to sustain it over the long term. Herbert looks forward to coming back next year with empirical data to back up his hypotheses about the relationship between professional development, morale, and retention.

Invited Talks

Dude, Where's My Data? Replicating and Migrating Data Across Data Centers and Clouds

Jeff Darcy, Red Hat

Summarized by Daniel-Elia Feist-Alexandrov (d.feistalexandrov@gmail.com)

Jeff Darcy discussed the basic problems faced when distributing and migrating data around the cloud. To start off he cautioned the audience that while there is no silver bullet in data maintenance, there are optimized replication infrastructures for several usage profiles. After giving a high-level overview of consequences that come with large data and varying basic environmental parameters, he gave an introduction to a very basic UNIX tool for synchronization, rsync, and its strengths and issues. Darcy followed up by exploring different replication topologies and strategies, wrapping up the talk with a discussion of different available distributed file systems.

The basic problem that Darcy observes is that computing cycles in the cloud move quickly, while the data that is needed to perform those isn't necessarily able to follow suit. This is because replication and migration of large data is complicated by its size, rapid turnover, and variety, which becomes especially complicated when dealing with large distances and replication across multiple domains. In such scenarios keeping the data in sync across the network nodes is a formidable challenge.

The first replication tool that Darcy explores is the simple UNIX rsync, which is used in production environments such

as the back-end of the Dropbox service. Although rsync performs well with large files, its downfall comes with a sensitivity to geographical distance and synchronization across multiple domains. Its architecture also entails high divergence between node states. But we learn from this simple case that the initial sync of our nodes is the least of our worries since it only occurs once. Darcy concluded this section of the talk with some advice on how to optimize the initial synchronization (such as packing files into larger archives and transferring in parallel).

Darcy then compared different replication types. He first explored synchronous replication, which, while keeping divergence at a minimum, is extremely latency sensitive. He then explored both ordered and unordered asynchronous replication. The former continuously logs the changes and thus only transmits what was changed and lowers divergence. The latter only scans the data periodically for diffs, which results in high divergence and is thus the less preferable of the two. Darcy then explored these two basic premises of replication by logging and replication by scanning in further detail.

In the next section, Darcy went back to rsync and proposed improvements to this simple tool. He concluded that scanning is inherently inefficient and introduced some well-known distributed file systems, such as AFS and Coda. Darcy championed the less well known but powerful XtreemFS file system. He concluded that all these file systems handle the challenges that come with large data volumes rather well and must be chosen according to environmental circumstances (such as bandwidth and distance) and what parameters are critical to the user.

Darcy finished the talk with a recapitulation of the lessons learned: Initial synchronization is the smallest worry in data replication, whereas staying in sync is hard. Conflict resolution is a major challenge and is best approached by segregating data by consistency requirements and choosing requirements on what is "just enough" consistency.

There were no questions.

Rolling the D2O: Choosing an Open Source HTTP Proxy Server

Leif Hedstrom, Cisco Systems
Summarized by Dybra Grande (granded@coyote.csusb.edu)

Leif Hedstrom discussed the problems systems administrators face when choosing an open source HTTP proxy server. One of the many issues administrators face is choosing the correct proxy server(s) to use from the overwhelming quantity on the market. Some of the proxy servers available are either commercial or open source, and some products offer caching, proxying, or do both. Hedstrom says, "This is where system administrators get lost in choosing the correct solution that works for them, and end up visiting social media sites to get opinions from 'reliable' sources such as other administrators who work on Netflix, Facebook, Twitter, Google+, and Usenet, where they prescribe solutions to problems which are sometimes irrelevant to implement due to the fact one is running different types of systems and applications than theirs."

Hedstrom included a crash course in his presentation before discussing his research on the different types of intermediaries available. Forward Proxy basically uses the user agent "the browser," which cooperates with the proxy server itself asking it to process a request on its behalf. This can improve performance because it allows you to cache and use this data. Reverse Proxy does the opposite of Forward Proxy because it acts on behalf of the servers. The administrator is responsible for setting the rules, such as when and where to cache. On Intercepting Proxy, the user is oblivious to the system administrator's set up. This is used by businesses and educational institutions who do not want users to visit Web site content that can be dangerous to their network or systems. It can also be used to block users from getting into their Facebook or Google+ accounts from their workplace.

Hedstrom covered several HTTP proxy servers, such as ATS, Nginx, SQUID, and Varnish, emphasizing the benefits and failures of these intermediaries. ATS offers good HTTP/1.1 support and includes SSL. Hedstrom mentioned the benefits of its ability to tune itself to the system and its excellent cache features and performance. The problems with ATS is that the load balancing is incredibly lame and difficult to set up, the developer community is small, and the code is complicated. Also, there are many configuration files, and there is still legacy code that must be replace or removed. Nginx, on the other hand, has a code base and architecture that is easy to understand. It has an excellent Web and application server that includes commercial support available from its creators. Nginx's problems are that adding extensions implies rebuilding the binary, it does not make good attempts to tune itself to the system, and there is no support for conditional requests or protocols.

Of the bunch, Squid has by far the most HTTP features, and it is the best HTTP conformant proxy today. Squid is widely used because of its mature features, which work pretty well out of the box. One of the negative issues with Squid is that it is based on old code and the cache is not particularly efficient. It has also been traditionally prone to instability and complex configurations. Varnish uses its own configuration language and has a clever logging mechanism, which supports several commercial entities. The problem with Varnish is that it does not support SSL, and protocol support is weak. In the end, Hedstrom reminded the audience that performance itself is rarely a key differentiator, but latency and feature correctness should be.

An attendee asked about issues with logging in Varnish. Hedstrom replied that Varnish can produce several hundred lines of logging with each request because it logs everything that

happens. The attendee thought this could be a vulnerability, and Hedstrom replied that varnishlog can cause latency by hammering on the disk or virtual disk.

Advancing Women in Computing (Panel)

Moderator: Rikki Endsley, USENIX

Panelists: Jennifer Davis, Yahoo, Inc.; Elizabeth Krumbach, Ubuntu, Adele Shakal, Metacloud, Inc.; Nicole Forsgren Velasquez, Utah State University; Josephine Zhao, Prosperb Media and AsianAmericanVoters.org

Summarized by Aileen Alba (acalba@csupomona.edu)

Jennifer Davis, Elizabeth Krumbach, Adele Shakal, Nicole Forsgren Velasquez, and Josephine Zhao all came together to answer questions Rikki Endsley had about women in computing. Some of the topics ranged from mentoring, networking, recruiting, and advice for women and their colleagues. Although it was a panel of women discussing women in computing, many men attended as well to find out more about how women work, think, and even feel. Each one of the women took turns answering questions and discussed their own experience. Rikki first asked, "What makes a good mentor and what skills are good?" Nicole Forsgren Velasquez made a great point when she said we all should have different types of mentors. She went on to explain, "If we only have a cheerleader mentor we miss the holes in our work, and if we only have a skeptic mentor we are always discouraged."

The panel continued with a discussion on women in the workplace from recruiting to advice for male colleagues. Elizabeth Krumbach pointed out that when creating the requirements for a job you must be realistic. Many women will not apply to a job if they don't have all the requirements, so employers need to make sure they clarify this in their requirements. Jennifer Davis also explained that interviewers should be aware that women communicate differently from men. When women say "we" it doesn't mean that they didn't contribute to the project, it just means that they don't take credit for all of the work. Understanding women in the workspace is hard for some men because they are not accustomed to having women in their companies. One of the best pieces of advice for men during this panel was that they should not comment on a woman's appearance unless they have an established relationship with her (Jennifer). Some terms people should be aware of when it comes to women are "derailing," "gas lighting," and "imposter syndrome." Adele made it clear that these terms will help men better understand how women feel.

Carolyn Rowland pointed out that women tend to internalize failure and externalize success. She continued by saying that women tend to give credit to everyone else even if we are the ones who lead something, but if we do something wrong we take blame for it alone. We must all be aware that women seldom brag or take credit for the work they do. Another attendee asked, "How should we help women have a more positive view of themselves?" Elizabeth said women sometimes just need a push and some positive advice. Josephine also explained that sometimes it takes a woman to change herself and also advertise for herself. Women need to be less shy about themselves; it might take time for this to happen but the more we all do this the faster we will see the change.

Carat: Collaborative Energy Debugging

Adam Oliner AMP Lab, University of California, Berkeley

Summarized by Tim Nelson (tn@cs.wpi.edu)

Adam Oliner presented Carat (carat.cs.berkeley.edu), an app that helps smartphone users improve their battery life. Carat is different from other such apps because it does not just advise people to use their smartphone less; it gives targeted advice based on statistical information gathered from many users. Carat looks at how much power each app uses, not for specific, pre-defined problems.

Carat does collaborative energy diagnosis. It collects power data from each phone on which it is installed and uploads the data to the cloud, where the data are compiled into a statistical profile of power use, broken down by app installation, OS version, and more. The collaborative approach is important for many reasons: different devices are used differently, and looking at a single device in isolation would not reveal that an app on one person's mobile is consuming more power than normal. Also, more data means a more accurate statistical profile.

Carat distinguishes between energy hogs (apps that use more power than other apps across the community) and energy bugs (apps that use more power than other instances of the same app across the community). Carat provides lists of these, along with estimated power savings if the user kills the hog or buggy app. It also gives a "J-Score"—a unified score that gives the percentile battery life relative to other users of Carat.

To use Carat, just install it and open the app about once a day, to seed data about power use. After about a week, you will start receiving reports that suggest what apps to close, whether you need to upgrade, etc. Carat is available for both iOS and Android, and is free on the Apple App Store and the Google Play Store. The app code is open source, although the analysis code is proprietary. No jailbreaking is required. They evaluated whether Carat actually improved users' battery life. After 10 days, users saw a 10% increase on average. After 90 days, they saw a 30% improvement on average.

Their initial deployment had 100 sign-ups, 75 of which installed the app. Developers got 10,000 samples. Over two weeks, they found 35 energy bugs, including popular apps such as Facebook. They also evaluated Carat by injecting three bugs into the Wikipedia app, all of which were detected. After releasing the apps on their respective app stores, they were featured on several online news sites, and found themselves with more than 100,000 users. Now they have more than 450,000. They have detected 11,256 energy hogs and nearly half a million energy bugs, some of which

were quite surprising. For instance, they found a case where turning on WiFi could improve battery life.

Cory Lueninghoener noted that some of Carat's recommendations involved updating the phone's OS version. He asked whether Carat ever recommended that users not upgrade, because the upgrade consumed more power. Oliner replied that that was something that they had discussed, but decided not to do; upgrading tends to install security patches, and so it provides an important benefit that isn't related to power use.

Alva Couch asked whether Carat was aware of application-specific settings, and whether it could make recommendations at that level. Oliner answered that Carat does not, but that is something that they want to provide via a developer API.

Soila Kavulya asked whether Carat could compare the power consumption of platforms as a whole. Is Android better than iOS, or is the reverse true? Oliner answered that it would be very tricky to tell, since the two platforms tend to use different hardware and different batteries. User behavior is another factor; some people will constantly reload news feeds, etc.

Tim Nelson asked whether they received useful negative feedback, or if it was mostly trolling. Oliner replied that yes, the negative feedback often gave them insight, even if it was not directly related to Carat.

Rik Farrow asked who paid for Carat. Oliner explained that Amazon Web Services provided a large amount of resources free of charge to his research group, and he expressed gratitude.

Plenary Session

NSA on the Cheap

Matt Blaze, with Sandy Clark, Travis Goodspeed, Perry Metzger, Zach Wasserman, and Kevin Xu, University of Pennsylvania
Summarized by Rik Farrow (rik@usenix.org)

Matt Blaze started with a reprise of his presentation at Security 2011, but that was not the scary part. Matt began with some background behind the project into open telecommunications networks with the aim of improving security for various wireless networks. University of Pennsylvania's focus is on two-way public safety radio. And, as this is NSF-funded and not classified, they are obligated to publish their findings.

APCO (Association of Public Safety Communication Officers) Project 25 (P25) is a standard for two-way digital radio, replacing the older analog radios. There are issues with backward capability, which is what Matt spent the next 45 minutes talking about. Because compatibility is the key to standards, multiple vendors' products have similar user interfaces as well as complying with the on-air protocols.

The P25 is used by police, fire departments, ambulances, but also the FBI, Secret Service, Treasury, postal inspectors, and even the US military. The P25's digital radio broadcasts on a narrow (12.5 KHz) channel, with each 180 ms of speech converted into 1728-bit voice frames encoded using the IMBE vocoder. Security is an option, which Matt said makes him excited because he will get to write a paper. For the most part, local emergency services don't want encryption. Federal services generally do, and this is where the problems appear.

The P25 uses symmetric encryption, and cryptokeys must be loaded into the radios in advance of being used. Matt explained that they can be loaded using a big cumbersome keyloader device or over-the-air rekeying, which allows updating of keys only if keys have previously been installed. There are no communication sessions, so the sender sets his radio to select the crypto mode and key, and the receiver must recognize the mode and have the right key loaded for this to work. Matt explained that the design errs on the side of allowing things to happen: radios play plaintext by default, and will also play any encrypted broadcasts for which they have the key. There is no authentication, so there is no protection against replay attacks or falsifying credentials (radio ID).

Matt described the P25 as an "ad hoc design," and there were some things that were done correctly. For example, the radios do not reuse initialization vectors, a common mistake in stream encryption protocols. There are mistakes in other areas: radio unit IDs are sent unencrypted even when in encrypted mode, silent radios can be made to respond (giving away their presence), and the design is very vulnerable to denial-of-service attacks.

Because there is no authentication, an attacker can replay messages, even encrypted ones. Matt joked that he got the FBI off his back by constantly replaying the message: "That Matt Blaze guy has gone to bed, so we can stop watching him." Matt later explained that Travis Goodspeed had discovered that there is a $15 toy that contains a transceiver chip that can be reflashed so it detects when an encrypted broadcast is occurring and can jam those transmissions by overriding the first 64 of the 1728 bits.

Next, Matt talked about passive analysis, looking for patterns of who is talking to whom, even if the content is encrypted. That type of analysis can be more powerful than actual content, as traffic analysis can be automated. And the P25 has a 24-bit unit ID assigned by the US government, which identifies the agency that owns the radio, and sometimes also the office and even the squad or person who owns a radio. The standard does support encrypting the unit ID, but we've never seen this and have been able to keep track of these IDs over time. If you add a pair of phased directional antennas, you can also locate radios as they transmit. Matt reminds us that the military are using these radios as well, and pointed out that even idle radios can be tricked into replying.

The radios also suffer from usability issues. The transmit crypto switch is an obscurely marked toggle switch, and that switch's state has no effect on received audio. Received audio is played if the signal is in the clear or if the signal is encrypted and the receiver has the key. Finally, rekeying is difficult and unreliable, and many agencies use short-lived keys.

Matt explained that one of the first things they decided to do in the field was to see how often people were using P25 radios in the clear. With a handful of grad students, and several thousand dollars for equipment, they were able to find out that quite a bit of federal agency and law enforcement traffic is in the clear. They decided they would focus on the federal government, by listening to just the frequencies used by sensitive organizations (so not the Park Service, but the Secret Service). There are 2000 channels allocated to the federal government, and they could determine the sensitive ones by watching for those that normally used encryption. They used an off-the-shelf hobby scanner, the Icom R-2500, which includes a P25 option, and is legally available to anyone.

They found friends with homes in high places, installed R-2500 receivers, antennas, and PCs with some software that collects metadata from received transmissions, and uploaded this information once a day. They typically got about 30 minutes of in-the-clear sensitive transmissions per city per day. By listening to this plaintext over time, his analysts, the grad students, identified which channels are used by which agencies. They also heard names of confidential informants, wiretap subjects' activities, about a wide range of crimes, and plaintext from every agency in the federal government with the exception of one—Postal Inspection.

Matt said that the friendly people in legal at U Penn found out that there is a law that specially allows people to listen to law enforcement radio traffic as long as it is not encrypted. They have tried to help, but the usability issues have prevented radio users from successfully improving the rate of encryption (about 90%). They did learn that by being a bit more systematic about their interception systems, they could learn a lot more. They also observed that various security folklore, such as change your passwords/keys often, actually makes security worse.

Mark Staveley pointed out that you don't have to jam every packet, but Matt said that just jamming every 100th packet would introduce a little stutter in the transmission. Because you only need to jam 64 bits out of 1728, and those 64 bits always follow a synchronizing frame, it is still just a tiny fraction of the energy needed compared to jamming the entire frame. Mark then asked about the cipher (a streaming cipher) and suggested the super-secret agency Matt wondered about what would be inside the Postal Inspection office. Someone asked whether they found any evidence that the vulnerabilities they discovered were actually used by black hats. Matt did hear a couple of times about "targets being sophisticated and taking counter-measures," but these messages were in the clear. Another person said that if the encryption algorithm was developed in the '90s, it could be decrypted. Matt pointed out that although DES could be decrypted with an exhaustive search of the key-space, AES, which uses a 256-bit key, is certainly out of range of an exhaustive search so far. Someone asked about the postal inspectors and their rekeying habits, and Matt said they don't rekey over the air, and perhaps are not changing their keys as often. The same person wondered whether perhaps they should produce a device with some useful function but that also did some jamming on the side. Matt said, "You're evil. Let's talk some later."

Rik Farrow asked how long have they been talking about the use of the receivers, and Matt said for about a year publicly. Rik then asked if they were scanning, and Matt said, yes, they are essentially sampling. Rik said that Matt and his graduate students have now collected enough information to make them an interesting target for Advanced Persistent Threat-style actors. Matt replied that they took a fair amount of care that the machine the data is uploaded to moves the data behind a firewall quickly. The easiest attack against us would be to apply to grad school at Penn and get accepted. It would probably be easier to get your own radios, Matt suggested.

Doug Hughes wondered about the first time they talked to the FBI and said that they wanted to record your over-the-air traffic. Matt said that they didn't have that conversation. They did tell the FBI, but only after they had been doing this for a while. They did talk to their IRB (Institutional Review Board) because they were collecting personally identifying information, which must remain private. They are also identifying federal agents who are making mistakes while using their radios and could get in trouble if they were identified. So they are prohibited from sharing that information with the authorities by their IRB. Someone else asked whether they are still collecting information, and Matt said they have two more radios than when they started, and that he always asks for a room on a high floor whenever he stays in a hotel.

They have shared their software with the government, but there is a problem: their software runs on Debian, which is not approved software. They have to "smuggle" their software in, so it can be used.

Papers and Reports: Content, Communication, and Collecting

Summarized by Dybra Grande (granded@coyote.csusb.edu)

What Your CDN Won't Tell You: Optimizing a News Website for Speed and Stability

Julian Dunn, SecondMarket Holdings, Inc.; Blake Crosby, Canadian Broadcasting Corporation

When the Canadian Broadcasting Corporation (CBC) reimplemented their content delivery network (CDN) architecture and changed it to a static delivery system, they never imagined the resulting scalability issues caused by the redesign of the configurations on their CDN and servers. To remove any possibility of downtime the writers of CBC agreed origin stability, content freshness, system complexity, and cost were significant. Originally, CBC's Web site was driven by J2EE applications that rendered news content from a relational database, which was difficult to maintain and meet business requirements. CBC's origin systems now consist of an Apache Server farm with no dynamic modules. Stories are now generated on content management systems (CMS) and converted into HTML fragments containing headlines, story body, associated links, and other user-displayed metadata. These HTML fragments are then wrapped by a "story wrapper" template using Server Side Includes (SSI) in which story variables are then injected accordingly throughout the CBC Web site.

Julian Dunn explains, the implementation parameter CBC uses to optimize CDN content freshness is based on setting almost all objects except HTML to a global site TTL of 20 seconds. In the act of achieving a high origin offload, edge servers will issue GET requests to the origin with an If-Modified-Since (IMS) header to ensure object bodies were updated and not sent unnecessarily through the system. Objects such as Site Icons, JavaScript, and CSS had a rigorous change control process in which expiration is organized through file systems with top-level directories. Also, to enable last mile acceleration and origin compression, the CDN's edge server will use gzip compression to send content to end-users without needing to recompress them. To enable HTTP-persistent connections and set appropriate timeouts, the CDN will attempt to keep a pool of connections open to avoid cache misses. These measures help ensure origin stability, content freshness, system complexity, and abolish downtime.

Dunn mentioned that SSI technology suffers criticism due its lack of incorporation of languages such as PHP and Ruby. Although SSI does not incorporate more complex languages, it provides security and performs well under high loads. It also protects the company and its employees by providing a good audit trail. Dunn concluded his presentation with several general lessons learned: (1) keep cache rules simple; (2) keep tuning knobs at origin if you can; (3) organize and categorize content; and (4) understand what "TTL" actually means. After the presentation, Brent Chapman, the session chair, asked whether the CBC ever considered automating the turnout process? Dunn responded, "It's a bit of a judgment call. There is a way to do it one way, turn off site features, or increase TTL. In the end, we will need developers to intervene. Yes, we can automate, but it is not just one knob."

Building a 100K log/sec Logging Infrastructure

David Lang, Intuit

David Lang discussed the need for Intuit to create a high volume logging infrastructure that can handle large batches of logs. In previous years, logs grew 60% per year, and traffic has only become more concentrated over time. Additionally, 75% of the possible logs were not being fed into the system. In 2005, vendor-neutral solutions such Arcsight, Sensage, Splunk, Nitrosecurity, and Greenplum were evaluated. The result was that none of the vendors were able to handle a 100K logs/sec load or the desired alerting/reporting functions. The architecture that Digital Insight decided to use was rsyslog for gathering and transporting logs.

Before Digital Insight decided to work with rsyslog, they tested several services such as sysklogd, syslog-ng, and rsyslog. Sysklogd daemon lost thousands of logs/sec under increasing traffic volumes. Syslog-ng hit a wall around 1K logs/sec and just dropped messages above this rate. Rsyslog handled short peaks of 30K logs/sec as it processed incoming messages on the memory queue, with the restriction of being able to write only a few thousand logs/sec. If traffic spiked and was greater than the memory could handle, however, it will would start losing lots of log messages. When it came to transporting logs due to the large number of networks, they decided to implement a set of syslog relay servers. These syslog relay servers were built in HA pairs and were set on an interface of 90 while accepting the risk of the unreliability of the UDP syslog messages being blocked by the router choke points from other networks. For delivering logs, they needed a reliable solution that could support multiple copies of logs of 100K logs/sec and a GigE wire speed of 400K logs/sec. They ended up going with Multicast MAC traffic software called CLUSTERIP using Linux. CLUSTERIP's role is to hash the connection of information and divide the resulting bases into a number of buckets, which are assigned to the local machine up the stack.

Someone asked how receptive the developers and management were. Lang replied that syslog developers were very receptive to the changes of the core syslogs, but that it was difficult getting approval or understanding of the importance of the project from the management and finance departments. Another audience member asked whether there were any nasty surprises or disappointments. Lang replied that their biggest hurdle was dealing with proprietary software, but they were pleasantly surprised

with the results. Another audience member asked whether their data center had a virtual center. Lang replied, "Our data center did not have a virtual center. We are interested with what happens with virtualization, but with everything else it is a factor. You really will have to do some testing, you will need more machines and more instances."

Building a Protocol Validator for Business to Business Communications

Rudi van Drunen, Competa IT B.V.; Rix Groenboom, Parasoft Netherlands

Rudi van Drunen described the design and implementation process of a system that tests and validates secure communication using XML messages through the AS2 Standard. This system provides a way for XML data to enter encrypted through an authenticated receiver using S/Mime. This protocol is essential to enable the deregulation of the energy market in the Netherlands. The goal of this project is to provide a test environment that can be used to certify more than 100 market parties to adhere to the new XML definition during the migration process. During this process more than 50 applications and protocol test scenarios will be verified before they are certified to participate in the new communication infrastructure.

The HTTPS and AS2 communications are handled by an Open Source Enterprise Service Bus (UltraESB). UltraESB passes the XML payload to a product called Virtualize, which is used as a virtualization engine to test validity in XML messages. Virtualize handles responses while storing data in a MySQL database. Information stored in the database includes meta information on business partners or timestamps. When it came to authentication and encryption of XML messages on the AS2 level, a Public Key Infrastructure (PKI) was used by the Dutch energy market and maintained by the government.

Someone asked whether there were ongoing certifications for certain versions. Drunen replied that recertification is necessary when vacancies or software updates occur and that a new partner coming to a new environment would need to be recertified, but it varies with different protocols. The same person asked whether they would use the two-way two-level encryption authentication scheme again within their database. Drunen replied yes, it was important to secure their database using a two-way two-level encryption authentication scheme.

Invited Talks

Surviving the Thundering Hordes: Keeping Engadget Alive During Apple Product Announcements

Valerie Detweiler and Chris Stolfi, AOL

Summarized by Nick Felt (nfelt1@sccs.swarthmore.edu)

Valerie Detweiler and Chris Stolfi, both AOL veterans of about a decade, jointly described the experience of keeping the popular tech Web site Engadget up and serving requests during peak traffic (i.e., when Apple announces new products). Chris noted that Engadget runs on the same shared publishing platform as more than 800 other AOL sites, but it got 4.4 billion requests in two hours during the last iPhone announcement, which is more than most AOL sites get during a week. Since 2007, Engadget has run a live blog for high-profile news stories such as Apple product events, using a revamped framework that has supported an increasing number of updates per event (reaching 973 updates for the iPhone 5). Chris observed that the condensed traffic surges triggered by these events can at least be anticipated, which allows them to prepare—and this is vital because the tech blog industry hasn't always been able to weather these events, meaning even more traffic for Engadget when the competition goes down.

The overall approach that Engadget takes to withstanding these traffic surges relies on a fairly traditional LAMP stack, with several layers of protection against high traffic. Live at the event, Engadget reporting staff submit new content to the CMS, which gets passed back to MySQL (for text) and media store (for photographs). At the same time, users' Web requests arrive and either hit the CDN or go straight to the load balancer, which has its own three-second TTL cache. Behind the load balancer is a LAMP front-end for MySQL and Apache for the media store, plus nginx as a proxy and cache for external API calls (so that Engadget can at least still serve stale content if partners go down). Memcache protects MySQL with about three gigabytes' worth of cache per server, which Chris said is generally more than enough. He emphasized the importance of having multiple layers of caching, which together allow them to get by with only one relatively modest machine as a MySQL server per data center. Valerie showed a chart of traffic during a peak event, explaining that the goal is to have the CDN handle most incoming requests and then serve the majority of those that pass it from the load balancer cache, thus leaving only a small fraction that actually hit the Web server.

Besides the core stack, Engadget has developed strategies for withstanding traffic surges based on lessons learned from previous events. One of these is simply to lighten the load by sending fewer bits to the client; for the iPhone 4's event they had to serve 100 Gbps, but they actually reduced this substantially for the subsequent iPhone 5 event despite having more updates and more readers. They accomplished this by switching their live blog page to update itself incrementally instead of requiring the user to refresh the page. This allows Engadget's servers to send back single live blog updates via JSON instead of the entire page, sustaining rates of more than 100,000 JSON calls per second because the calls are lightweight enough to be cached easily by the load balancer. Another strategy is to reduce complexity, favoring availability over performance. This means relaxing geolocation constraints and serving some users from relatively far away data centers. (Valerie recounted how for one early keynote, requests were so concentrated in California that

the Mountain View data center was overwhelmed with traffic, leading to a domino effect as the traffic then hit the next-nearest data center, and then the next.) It also meant removing extraneous beacons and ads from the live blog page, because third-party infrastructures would fall over under the heavy load. With these techniques, Engadget was able to stay up successfully during the entirety of the last iPhone announcement.

Chris Reiser (Groupon) asked, given all the caching, whether it was possible to clear the cache in the case of a bug. Chris Stolfi answered that it's a non-issue for the live blog's JSON calls because they're only cached for three seconds, and for the CDN they can use a tool to clear it; Valerie noted that because objects are versioned, the preferred option is to do a new publish. Jake Richard (Yahoo) asked how they determine at least an order of magnitude scale for what they need to have to handle the traffic. Chris pointed out that until recently Engadget had never stayed up the whole time, and thus hadn't known many people had tried to visit the site. Now that they do have this benefit, he said it was pretty much just a matter of doing standard load-testing to get a unit of scale and then estimating how many people they expect to come. He noted that Engadget does get influxes of new people as other sites fail, but they can compensate for fluctuations in traffic by adjusting the live blog client's query interval, say from the three second default up to five seconds, in order to control the rate of JSON calls.

Vitess: Scaling MySQL at YouTube Using Go

Sugu Sougoumarane and Mike Solomon, YouTube

Mike Solomon and Sugu Sougoumarane of YouTube discussed their recent work building Vitess, a project in the Go language designed to improve the scalability of MySQL. They divided up the talk such that Mike covered the MySQL aspect and Sugu addressed their experience using Go. Mike began, explaining that YouTube had originally scaled MySQL up to the cluster level with a collection of homegrown scripts that could be difficult to use. Vitess was born of the desire to distill those scripts down to the simplest way to manage a sharded MySQL instance. They wanted to stick with MySQL because it's popular, easy to use, and reliable, but doing so at a large scale required overcoming obstacles: making the system relatively self-managing to reduce the time needed to manage hardware, and increasing efficiency to support greater throughput without needing thousands of connections to the database. They decided Vitess would use external replication with eventual consistency in order to get data out in near-real time, and would provide automated reparenting of slaves so that a wide array of operations could be performed conveniently by doing them in the background on a replica and then failing over to a new master. The database would be sharded primarily by the leading edge of the primary key, and would not provide cross-shard transactions, which Mike said might seem like cheating, but in their experience was a reasonable limitation.

Given these constraints, the implementation strategy for Vitess was to make minimal changes to the MySQL codebase—just two 25-line patches—and rely instead on an external tablet manager and an external query shaper, both written in Go. The tablet manager maintains the sharding of the entire space of primary keys up into individual tablets, and lives on every box running MySQL. It stores coordination data directly in Apache Zookeeper, a highly reliable notifying file system from the Hadoop project. The query shaper provides an RPC front-end to MySQL, and has been serving all of YouTube's MySQL queries in production for more than a year. It manages a pool of database connections, and provides a number of fail-safes, including query consolidation in the case of duplicate queries, row count limits for the number of rows to return, and a SQL parser that allows it to intelligently reshape queries on the fly. Besides the tablet manager and query shaper, each tablet server also provides an update stream of primary key change notifications derived from the database binary log. Work is in progress developing a row cache to support better random access performance than MySQL's traditional page cache.

Sugu described some of the highlights and lowlights of using Go for the Vitess project. He noted that the main benefit has been in productivity: writing code went quickly because the language is much more expressive than other widely used compiled languages, falling closer to Python than C++ in that regard. Go's quick compilation (for example, the Vitess tablet server compiles in less than three seconds) and well-designed set of libraries also saved development time. He touched on his appreciation for several of Go's helpful language features, including an intuitive approach to interfaces, first class concurrency via lightweight goroutines, syntactic elegance with defers and closures, and the ability to call into C code. At the same time, he also pointed out some of Go's rough edges, including mismatch between string types, lack of agreement on how to handle errors, and deficiencies in the garbage collector and scheduler (although work continues on both components and he expects them to improve). Mike spoke on deploying Go code in production, saying it was relatively easy to debug—sending SIGABRT tells you the state of every goroutine stack—and casting it as a good experience overall. He and Sugu said Vitess recently picked up three new committers for a total of five, and invited involvement in the project at http://code.google.com/p/vitess/.

Someone asked for details about the version of MySQL that Vitess uses, and Mike said they were using the community build of MySQL 5-point-something with a few small patches, rather than the Google internal build. Someone asked whether they'd looked at 5.6 and GTIDs (Global Transaction IDs). Mike said the route they've chosen is applying GTIDs to the 5.1 tree using

Google's stable internal patch. Vince Clark (VMware) asked about issues in debugging code with goroutines, particularly interactively. Mike answered that although Go can be massively concurrent, it has good primitives with a clear memory model, so it's generally not a problem. Triggering a panic produces a full stack trace of every goroutine in flight, which then just needs to be examined carefully to diagnose problems. Asked as a follow-up whether the lack of visible thread IDs hampered inspection, he explained that the reduced exposure hasn't been limiting, because the specific thread only mattered when interacting with certain kinds of C code. Sugu also remarked that for their one tough deadlock issue, they still just needed to crash the server and then examine the stack trace. Finally, Kent Skaar (VMware) asked whether they had used the SSL support in Go. Sugu answered that they've tried SASL and messed with the crypto package but haven't used SSL.

Ganeti: Your Private Virtualization Cloud "the Way Google Does It"

Thomas A. Limoncelli, Google, Inc.

Summarized by Andrew Hume (andrew@research.att.com)

Tom Limoncelli started with an overview of Ganeti, a management tool for clusters of VMs (either Xen or KVM). The fundamental terminology that Tom used is node equals physical server and instance equals VM. VMs can use a SAN or local disk. When using local disk, they use RAID to mirror across two nodes so that the disk is in two places, and thus we can move the VM. Moving VMs is based on two primitives: move a VM and move virtual disk (storage).

Tom said that being able to move VMs provides real benefits if the VM needs more memory than on the node it's currently on, or in the case that a disk or node is failing. He then described various roles in Ganeti, such as the master node and some processes—for example, the node daemon and Ganeti watcher—and different sized configurations (small, medium, and huge). The scaling issues involve an administrative lock on node/instance operations (not any VM internals).

Google tends to use Debian-based Xen in para-virtualization mode, with DRDB (Distributed Replicated Block Device) and local disks (no SAN). Google operates "huge" clusters in a few data centers for self-service, and one or so medium cluster per office ("office in a box"). Tom then described a bunch of management tools and how Google manages their clusters (e.g., clusters are generally tuned for one of a few different workloads). Tom finished with a live demonstration using Ganeti on a test cluster.

The code can be found at http://code.google.com/p/ganeti.

DNSSEC: What Every Sysadmin Should be Doing to Keep Things Working

Roland Van Rijswijk, SURFnet bv IPv6 and DNSSEC

Summarized by Steve VanDevender (stevev@hexadecimal.uoregon.edu)

You might already be using DNSSEC and not know it. Traditional DNS does not provide authenticity or integrity information, but with DNSSEC, domain owners can digitally sign zone data, and resolvers can check those signatures to verify authenticity and integrity of DNS data.

The EDNS0 standard provides support for DNSSEC by specifying additional flags and larger UDP replies of 4096 bytes (by default) for DNS information, and is enabled by default in modern DNS server software (such as BIND, Unbound, and Microsoft Server 2008R2 and 2012). In particular a client resolver can set the "DNSSEC OK" flag to request a DNSSEC reply, and this is also enabled by default in many recursive resolving servers. Even if a client resolver doesn't ask for DNSSEC, it may use a name server that is one of the 70% of all recursive resolvers on the Internet that do have DNSSEC enabled, and 90% of those use the 4K default maximum reply size. Typical DNSSEC replies may return more than 3K bytes of data and therefore may broken into three or more IP fragments.

Fragmentation causes problems because some firewalls drop fragmented DNS replies, originally in response to some security attacks common in the 1990s, and such configurations still exist because of outdated recommendations from vendors and auditors to block fragmented DNS UDP replies and disallow TCP DNS replies. If a resolver makes a DNSSEC request behind such a misconfigured firewall, it never receives a complete reply, and the resolver eventually sends an ICMP fragment assembly timeout message back to the server. Monitoring these ICMP messages allowed SURFnet to estimate that 1% of resolvers contacting them had this problem. Other research suggests 2% of all Internet hosts and 2–10% of recursive resolving name servers may have this problem. Resolvers may also experience serious performance issues if DNS fragments are blocked, as they will eventually retry using TCP but can take several seconds to do so.

To avoid problems on your recursive resolving servers, van Rijswijk recommended verifying that your resolvers can receive large and fragmented UDP DNS replies. DNS-OARC provides a tool for this at http://www.dns-oarc.net/oarc/services/replysizetest. You should also configure firewalls not to drop fragmented DNS replies and not to block TCP DNS replies on port 53. You can also reduce your EDNS0 maximum reply size to 1472 (below the Ethernet MTU) or 1232 (below the IPv6 MTU) to reduce problems with fragments.

Another problem encountered by SURFnet after enabling DNSSEC was network amplification denial-of-service attacks from DNSSEC UDP queries with forged source IP addresses. A query of 68 bytes can return a reply of 3300 bytes, resulting in an

almost 50-fold increase in bandwidth between the attacker and the attack target. One attack was observed to generate 38 Gbps of traffic toward a target, with their name servers receiving 10 Kbps and sending 50 Mbps to the target.

One way to prevent such amplification attacks is to implement IETF BCP38 http://tools.ietf.org/html/bcp38 to filter spoofed traffic. DNSSEC server operators should also monitor for such attacks and filter them. Rate-limiting DNS can also help, but rate-limiting is not yet available in all name server software and may affect legitimate traffic if not implemented carefully.

DNSSEC Deployment in .gov: Progress and Lessons Learned

Scott Rose, National Institute of Standards and Technology (NIST)
Summarized by Steve VanDevender (stevev@hexadecimal.uoregon.edu)

The US federal government has mandated that .gov DNS zones are to be digitally signed and served with DNSSEC. This was originally motivated by Dan Kaminsky's presentation on DNS spoofing at Black Hat 2008, followed shortly by OMB-08-23, which mandated that the .gov zone was to be signed by January 2009. The rest of the federal executive branch was to be signed by December 2009, and DNSSEC was added to the FISMA standard requirements for all federal information systems.

Progress on DNSSEC deployment was not as rapid as was hoped. The .gov zone was not actually signed until February 2009, and only 30% of the subzones met the original deadlines. Furthermore, 10% of the zones that were served with DNSSEC had various errors, although only a few were noticed by operators or client resolvers. Some of these errors persisted for about two weeks until they were corrected.

A number of challenges made deployment and maintenance of DNSSEC difficult. Besides trying to meet the initial deadlines, DNS data has to be re-signed periodically even if the zone data did not change. DNSSEC also required more interactions between parent and child zones, with child zone keys needing to be uploaded to parents whenever they change. Existing DNS operators also had to learn DNSSEC and sometimes had to change their service plans or even obtain new equipment. This led to some consolidation and reorganization of existing DNS service.

To address problems with lagging deployment and failed security audits, a "DNSSEC tiger team" was formed in April 2011 by the federal CIO council and staffed by volunteers. The teams hold monthly meetings to discuss progress and problems with deployment. They produced training material and monitoring tools and created discussion forums for other government system administrators. They also produced reports for departmental CIOs, but these were not always handled quickly and may not have been all that helpful.

The "tiger team" did produce an improvement in the number of signed and valid DNSSEC domains under .gov, although the number of "island domains" and domains with errors remained fairly consistent. Currently 70% of .gov domains are signed.

Between August 2011 and March 2012 there were frequent problems with DNSSEC errors, although the rate fluctuated significantly. The most common errors were expired signatures. Centralization of some services led to bursts of errors when subzone operators forgot to sign their zones. Many of these correlated with holidays when operators were unavailable to renew signatures. Other problems included bad key rollover, when keys were mismatched between parent and child zones, or mismatched timestamps, where a child zone appeared to have been signed before its parent; however, in the first year after the "tiger team" was formed, response to errors improved significantly, particularly with the common problems of no or expired signatures on domains, with error rates reduced to about 20% of their initial levels and problem resolution times cut in half.

Rose drew a number of lessons from the US federal government's DNSSEC deployment efforts. Monitoring to report errors was the first step to indicate the scope of problems. Getting organizations to provide current points of contact for DNS and security operations improved resolution times. Operators were encouraged to automate the error-prone aspects of DNSSEC operation, especially zone re-signing. Fostering an internal community for DNS administrators made it easier to share information and solve operational problems.

Papers and Reports: If You Build It They Will Come

Summarized by Steve VanDevender (stevev@hexadecimal.uoregon.edu)

Building the Network Infrastructure for the International Mathematics Olympiad

Rudi van Drunen, Competa IT; Karst Koymans, University of Amsterdam

The International Mathematics Olympiad is an annual event held in a different country each year, with more than 600 international high school students as competitors, more than 100 jury members, and with more than 60 different languages represented. The contest itself occupies two days, but an additional five days are involved in preparation, translation, and correction and scoring of papers. The contest held in the Netherlands was hosted at two Amsterdam hotels, an Amsterdam sports complex where the competition was held, and an Eindhoven hotel for the jury members. All of these sites were networked together for communication among the contestants and jury members, although traffic had to be isolated between those groups for security, and with substantial flexibility to allow for considerations such as people moving between hotel rooms.

The network they developed used VLANs to isolate traffic while allowing it to be consolidated on common physical links. A VPN system based on FreeBSD OpenVPN was used to provide

security and allow more flexible access. Traffic was also isolated from the general Internet using NAT with a gateway at the University of Amsterdam. A backup network using 3G was also available in case their telco connections went down, and "warm standby" host replacements were available at each site. Site setup took six people working over four days, involving lots of improvisation and thorough documentation maintained in a wiki.

van Drunen drew several lessons from their experience. Expect the unexpected in site surveys. Use a wiki for documentation. Use DNS for all host information. Label everything—cables, hosts, equipment. Use open-source tools such as FreeBSD, OpenVPN, and Wireshark. Provide hand tools at all sites for hardware fixes. Be flexible by design, such as putting all VLANs on all switches and avoiding complicated procedures and layers of management. Test everything. Allow enough time for building your network. Have multiple communication methods available. Take time to prepare and build your network.

Lessons Learned When Building a Greenfield High Performance Computing Ecosystem

Andrew R. Keen, Dr. William F. Punch, and Greg Mason, Michigan State University

Awarded Best Practice and Experience Report!

Building a high performance computing (HPC) environment involves more than just getting the most FLOPS (floating-point operations per second), but in making it an effective tool for its users. HPC is critical to research and often provides a competitive advantage, but it also requires substantial funding from university administration to create a useful resource. A first attempt to build an HPC system with the involvement of a major vendor appeared to have great benchmarks, but it underperformed in real use mainly due to inadequate I/O bandwidth for storage.

Storage for HPC needs to be fast but also safe to protect user data. For their environment the team used Lustre over Infiniband for storage with the ZFS file system. This provided for snapshots and off-site replication of data for backups. To improve responsiveness, solid-state disk (SSD) was added for caching. Later, the storage system was designed to allow for failover, and it had increased CPU capacity and less use of SSDs since they found that RAID caching was not being well used.

Their HPC system had to fit in a small machine room, with lots of power dissipation and need for cooling. Spot cooling was used to deliver cold air to system intakes, and inexpensive heat containment was obtained by using cardboard to route airflow (later upgraded to Plexiglas). They found that using standard IPMI instead of proprietary management tools made hardware management much easier; firmware updates and configuration could be easily managed remotely, and with better consistency than manual updating. Software management was done using configuration management systems—for consistency, systems were never managed "by hand." They also found that using a single OS across the entire cluster made management easier, and an open-source distribution like ROCKS has already solved many HPC design problems. Job queuing was request-driven and allowed for managing multiple jobs in parallel; however, queue selection was automatic for users, so they did not have to learn details of the queuing system to manage their own jobs. Systems were monitored using in-band methods to track performance, Cacti to do out-of-band monitoring, and Nagios for failure alerting.

Someone asked how to set up trust properly between systems. Keen replied that one example is allowing management hosts to ssh to managed hosts in the cluster. The assumption that hosts in the cluster should trust each other is not a good one, however.

Building a Wireless Network for a High Density of Users

David Lang, Intuit

Lang attended SCALE (Southern California Linux Expo) in 2008, and like many attendees at many conferences found that the wireless network didn't work very well. He volunteered to help design a better wireless network in 2010, with his technical expertise including experience as an amateur radio operator, after the original wireless vendor backed out shortly before the conference started.

Wireless networks that appear to work in early testing often collapse when lots of people try to use them. Technical conference networks are especially problematic because there are lots of people—and, more importantly, gadgets—in a small area. Collapse is inevitable when fundamental limits on the amount of radio airtime available are reached, but it is possible to delay that collapse, sometimes by doing counterintuitive things.

WiFi has the same problem as radio in that only one device of any sort can be "talking" at any one time on a channel. In high-density areas there are also "hidden transmitters" where devices on one side of an AP may not be able to detect devices talking on the far side. It takes little interference to corrupt transmitted packets. Wireless devices may try to reduce transmission speed to overcome interference, but that just increases the airtime they use. Even regular housekeeping traffic may use up too much available airtime to allow devices to transmit data. Many OSes use large network buffers, and this "bufferbloat" may cause a device to transmit for long periods and retransmit more when reception fails. Turning up transmitter power on APs usually doesn't help since it just increases interference between APs and doesn't help with receiving data from low-power mobile devices. So-called "enterprise-class" APs often don't help because they typically concentrate many radios in one place and use directional antennas that create more hidden transmitters.

There are a variety of methods for solving these radio problems. Doing a site survey of your venue helps by allowing you to determine better AP placement, especially if you measure signal strength using mobile devices and tools such as MySpy and Kismet. You can also find the wired network jacks that actually work. As much as possible use 5 GHz WiFi, which has 8–18 available channels (depending on sources of interference at a location) instead of the three available in 2.4 GHz. Use lots of APs, and set them to use lower transmitter power, especially no more power than client devices use. Use existing transmission obstacles such as walls or the presence of crowds to avoid hidden transmitter problems. Placing APs closer to the floor may also help. Advanced antennas should be used carefully to direct signals away from areas rather than toward them, and directionality can also help to avoid cross-floor interference.

Using a single SSID can allow devices to roam between APs, but have separate SSIDs for 2.4 GHz and 5 GHz if both are supported since devices may give up before finding 5 GHz and obtaining better performance. Enable wireless isolation so APs don't relay traffic between mobile devices. Reduce the "beacon interval" so less airtime is used for housekeeping. Using distinct prime number intervals across multiple APs also avoids collisions between beacon broadcasts. Disable slower speeds (i.e., 1–11 MHz). On APs, reduce kernel network buffering, disable memory-intensive connection tracking, and use short inactivity timeouts to forget about inactive devices sooner.

Invited Talks

Disruptive Tech Panel

Summarized by Barry Peddycord III (bwpeddyc@ncsu.edu)
Moderator: Narayan Desai, Argonne National Laboratories
Panelists: Vish Ishaya, RackSpace; Jeff Darcy, Red Hat; Adam Oliner, University of California, Berkeley; Theo Schlossnagle, OmniTI

Each panelist began by talking about his predictions for the next 10 years of system administration. Adam Oliner predicted that there is going to be a paradigm shift where system administration will take on a more substantial role in software development. Most of the development that administrators do is in the form of scripting because the APIs for the tools deployed will not necessarily be consistent from environment to environment. With the push to cloud infrastructure, the interfaces to these resources—and, by extension, the solutions developed on top of these resources—have started to converge. While scripts are appropriate when each system is wholly unique, the growing trend toward unified APIs means that solutions will be more generalizable, and it will be more effective for system administrators to share their approaches with one another so that they can help stand on each other's shoulders. It is at this point that scripts become software projects, and system administrators begin to become developers.

Theo Schlossnagle disagreed, arguing that APIs are only meaningful when they serve as a layer of abstraction on top of a reliable resource. He asserted that the role of a system administrator is to make an unreliable infrastructure less unreliable for the benefit of their developers and users. The cloud, despite being widespread, is no more reliable than any of the other resources that make up computer infrastructure, and, in fact, sharing solutions that leverage a common API simply makes it more likely that common mistakes will be shared as well. When a script that solves a problem exists, it is very attractive even if it is inefficient or inappropriate for the usage scenario.

Whereas Oliner and Schlossnagle believed that predicting the future is easy, Vish Ishaya wasn't so sure, stating that visionaries have been making poor predictions of the future for decades. He said that rather than looking at what people are adopting, the best way to see the effect of disruptive technology is to look at what people are not doing anymore. He alluded to how C was disruptive in an era when programmers were using assembly language to accomplish tasks on their machines, as it stopped the practitioners from doing what they were used to doing in their daily jobs. He mentioned that there are many jobs that system administrators do that may be abstracted away, such as managing databases, Web services, and distributed systems. Echoing Oliner, he cited the explosion of APIs as an indicator of things to come.

Jeff Darcy changed the tone by looking at a more specific technical issue, primarily the changes to storage over time, with storage behaving more like memory and being distributed across multiple machines in networks. When storage essentially becomes permanent memory, many assumptions about the behavior of the storage can no longer be made. Although good security practices often involve clearing passwords and keys stored in memory, new practices for protecting sensitive data have to be addressed when the abstractions about where memory is located and where it is copied no longer hold. Furthermore, as more memory is distributed, the issue of desynchronization has to be considered as well. When data diverges across sibling nodes in a network or between caches held by systems, assumptions made by applications about consistent data in memory may not hold, and the decision to read invalid data or block until the data is valid can be a hard one to make in some scenarios.

In addition to predicting the changes to the field of system administration, the panelists also talked about what they were most excited about in the future of system administration, and the responses were all over the board. Oliner was most excited about the idea that, because that technology has started hitting the limits set by the laws of physics, the next generation of administrators, developers, and academics are facing a new set of constraints that must be worked around. Because latency is the unsurpassable bottleneck of networking, the next steps

are not in improving speed, but improving prediction—moving computational power to take advantage of Big Data. Schlossnagle was also excited about optimizing, with the potential for full recreation of systems from the bare metal, while Ishaya was more interested in seeing how these newly created systems would be treated as systems of their own, building abstractions. Darcy closed the discussion by saying that he was looking forward to advances in asynchronicity. As mobile devices increase in number, and latency grows due to the geographical concerns of a globally connected world, dealing with asynchronous storage and communication is the current problem that needs to be faced.

TTL of a Penetration

Branson Matheson, NASA

Summarized by Mario Obejas, Raytheon

Branson is a 24-year IT veteran who loves a "You can't do that" challenge. He began by asking what kinds of sysadmins were in the audience, and when he asked how many security administrators were present, only a few hands went up. In a sense it's a trick question since, as Branson asserted, security should be a component of every system administrator's duties.

Branson then presented a series of referenced statistics to create a context for the talk by comparing the number of sysadmins supporting associates in a business to those in particular roles: 1 to 30 for sysadmin, 1 to 200 for network admin, and 1 to 1200 for security.

Branson defined black hats as individuals trying to impact an organization negatively, providing the usual list of suspects: script kiddies, bored students, hacktivists, governments, and organized crime. Branson also said that vendors, developers, and users are also often unintentional black hats. And, as always, users are the weak point in the system, subject to social engineering.

Eighty percent of US households have at least one computer. These have a plethora of operating systems, with a profusion of services and applications. The attack surface is huge. He estimated that there are 141 million workplace users in the US and 20 million of these are government users. There are more than 240 million home users, with 85 million on broadband. Given these statistics, Branson asserted that a penetration test (aka pen test) should go after users more than infrastructure.

Branson estimates there are 5 million real hackers in the US alone. Black hats have the advantage (tools, knowledge, sites, conferences, certifications, etc.). Training for the latter includes ShmooCon, DefCon, B-Sides, LISA, etc. Branson said that a person using Metasploit can easily penetrate a network-connected WindowsXP box in <1 second. Aircrack can crack a WEP key in 6 seconds.

Unlike white hat rules, there are no black hat rules other than "Don't get caught." The bar for entry into the black hat world gets lower because cool tools come along every day, and existing ones get better. Survival time of an unpatched machine directly connected to the Internet is on average less than five minutes now.

Branson described five pen testing/attack steps: 1. Target reconnaissance: Pig, Maltego, Netcraft, Google are tools of the trade. Social Engineering is another standard tool: call a sup-port line, change a password; also, use public knowledge to answer security questions. 2. Probing: where are the holes? 3. Exploit: (Metasploit, hydra, custom hacking scripts). 4. Once in, cover your tracks: clean logs, hide code, install root kits, obfuscate network traffic, disable monitoring, pivot to spread to other systems.

With knowledge of the pen tester's list of actions, the sysadmin/victim needs to be on the lookout for unwarranted increases in support calls, spikes in Web traffic, increases in "Friend" requests, increased probe/suspicious traffic (as noticed with Snort), increased load, increases in httpd-error and EventLog (Windows) activities. After an exploit is successful, the victim may see the following symptoms: changed files on file system, changed system behavior (possibly compiler use), and network traffic changes.

Prevention starts with baselining a good system. This is in fact the primary overt message from this talk. Do the same thing your adversary would do—do reconnaissance on yourself, and know what is out there about you. Be aware of what operating systems you have, which services are available, and what levels of traffic are "normal." You also want to baseline your ticketing trends.

You will want to use IDS (such as Snort) inside your networks, and to perform log analysis using such tools as awstats, Webalyzer, kernel/security log reduction (via Splunk, for example), log watch, and ntop. It's very important to centralize logs and aggregate the data reduction. Tools such as OSSIM and Bo are integral and will save you time. You can also use configuration management to notice baseline changes, as well as to recover from penetrations.

Near-Disasters: A Tale in 4 Parts

Doug Hughes, D.E. Shaw Research, LLC

Summarized by Yakira Dixon (dixony@coyote.csusb.edu)

Doug Hughes began his talk with powerful slide images of disasters (a flooded Verizon data center during Hurricane Sandy, the rubble of the 1906 San Francisco earthquake, the aftermath of Fukushima) before he discussed four unrelated near-disasters that he and his team experienced at the beginning of 2012. The first issue involved a degraded WAN. The network between their primary office and primary data center is an OC-12, an optical, leased line. If the OC-12 link went down, failover to a backup connection would cause a jump in latency. It took some

investigation to figure out that mismatched fiber-optic transceivers between the partner-provided OC-12 router and the carrier-provided OC-12 equipment was causing the primary network link to go down. The carrier replaced the long range receiver with an intermediate range transceiver and the primary link was up and running again. Both transceivers were eventually replaced with short range transceivers so they could not overpower each other.

The second issue involved archive failure. A mega RAID controller for a backup storage server lost knowledge for a group of eight adjacent disks. Other disks on the controller had no issues. They tried to resolve the issue by reseating the disks and by power cycling the server. When the disks were relabeled with RAID controller logical unit (LUN) labels, the disks were visible with large integer labels instead of controller numbers, but there was output showing the label that the disk used to have. They attempted to relabel the disks with dd using this output but ran into namespace collisions with the new mapping. The issue was fixed by removing the eight disks, rebooting, re-inserting the disks, and restoring the label to a factory default using the Solaris command format -e. Then the labels were fixed one at a time. Some things that were learned: ZFS can repair disk labels wiped by dd. ZFS output about what disks used to be labeled can't always be trusted.

The third issue began when one of the primary application servers went offline, leaving half of their cluster machines handling NFS application requests. They began troubleshooting by running diagnostics on hardware and swapping the RAID card with a card from another machine, but the server remained stuck during power cycles. The server was able to boot normally after removing an SSD that would fail and hang the SATA bus. Doug recommended having spare RAID cards and SSDs on hand, as well as having machines available that allow for the swapping of parts.

The fourth issue was the largest and could have resulted in massive data loss—640 TB of primary storage. Doug provided some background on the storage system architecture: four Linux hosts serve GPFS and NFS to clients and communicate with two storage cabinets. The first cabinet has two storage controllers connected to disk shelves. The second cabinet has shelves that connect to their corresponding shelves in the first cabinet. If a storage controller fails, half of the paths to storage are lost. A controller failed and the vendor shipped a replacement. Shortly after the first controller was swapped out, the second controller failed, and they were told by the vendor to power cycle the two storage shelves. When that action went awry, Doug performed an emergency shutdown of the GPFS nodes to preserve the integrity of the file system. The system was brought back up and the I/O card in the storage shelf had to be replaced. Things were stable, but broken disks needed repair. LUNs that had journals with information on how to rebuild disks were rebuilt first, and the vendor had Doug and his team perform some undocumented methods for fixing disks lost in the RAID-6 stripes.

Doug presented some meta-ponderables, things to consider based on all of the issues his team dealt with. How much information should be communicated to management during a near-disaster? Can your tape data be restored easily? Doug asked the audience how many people had tested their backups and about 12 individuals raised their hands. He added that squirrels tend to be responsible for many power outages. Doug jokingly said that squirrels were the worst natural disaster in IT history.

Closing Plenary Session

15 Years of DevOps

Geoff Halprin, The SysAdmin Group

Geoff Halprin opened his presentation with a two-part thesis: (1) in the next decade, operations in general will look a lot like operations at Google or other major .coms, and (2) software development has changed forever, and system administration must do the same. From this point, Geoff briefly discussed the evolution of software development, starting from the waterfall model. This method of developing software was broken because it made a lot of incorrect assumptions about the development process. The response to the failings of the waterfall method was to build new methodologies, such as extreme programming, and agile development practices that embodied principles like daily collaboration between business people and developers, continuous delivery of valuable software, and using working software as a primary measure of progress.

Geoff then turned to a discussion of the incorrect assumptions made about operations. He emphasized that documents that define the waterfall software development life cycle did not mention operations at all; it was assumed that programmers dealt with operation aspects of the system. The greatest assumption made was that operations was involved with the development team in determining project requirements. DevOps is important because it makes the assumption true. It requires an operations person or team to be involved in the development process.

After a quick apology about the brevity and subjective nature of this part of the talk, Geoff talked about the history of system administration. In the past, system administrators dealt with large systems, UNIX variants and networking variants. A community of programmers, mathematicians, and scientists who found themselves doing administration work came together at LISA conferences and user groups to spread ideas about the professions. Geoff discussed his attempt at defining the role of a system administrator, the System Administration Body of Knowledge (SA-BOK). He continued his history as he talked about the rise of the Web and the three-tier infrastructure model, the commoditization of hardware, virtualization of

servers, and the emergence of the cloud, which involves infrastructure on demand, infrastructure as code and provided APIs, and automation driven by scale.

Geoff said that DevOps is to system administration what Agile is to software development; it integrates development and operations teams so they can collaborate more effectively. DevOps is a culture that encourages teams to learn from one another and make their workflow visible. Developers start writing infrastructure as code and are involved in production support. Operations contributes stories to development and uses Kanban walls to keep the team informed on what tickets are being worked on. Geoff quickly went over some DevOps tools and cloud infrastructure frameworks.

Geoff noted that DevOps isn't a complete model and that not all products or environments will fit with the model. It will take time to learn how to scale the model in traditional enterprise environments. Geoff asked whether there were any audience members who thought DevOps was a great way to run their core systems and there were no hands raised. There are a lot of problems that DevOps does not solve, such as determining serviceability criteria or how to monitor services. It doesn't look at the entire Operations life cycle or teach professionalism or ethics.

Geoff stressed that DevOps is not a new concept. While showing various slides from 1997, he talked about his past practices and tools for configuration and systems management. Later he made the point that DevOps is a continuation of a path that system administration is currently taking. This path leads to software-defined data centers where virtualization is necessary and automation is critical. Cloud standardization is changing and the next generation of cloud frameworks will be more generic. Geoff wrapped up his presentation with a statement to ponder: companies that aren't moving toward a cloud model for IT service delivery are setting themselves up to be outsourced.

Jay told Geoff that he has trouble with how DevOps changes how developers do things, and wanted to know how to keep a system safe from a developer working on systems code. Geoff said to be careful of a false dichotomy, to stop talking about the situation with a "them vs. us" (devs vs. ops) perspective, and to focus on ensuring the integrity of changes to a code repository, irrespective of who makes those changes. Someone commented that Geoff's slides about cloud-based startups were chilling and asked the audience if anyone was involved with a company that was completely cloud-based. A couple of people raised their hands in response. This person noted that cloud-based startups were a trend in the San Francisco Bay Area. An attendee told Geoff that he worked at an Agile shop for development and was curious about resources for maintaining an Agile workflow for sysadmins. Geoff couldn't suggest any specific resources but discussed different categories of workflow and how to organize those categories using a Kanban wall or ticketing system. Garrett Wollman from MIT said that his organization doesn't produce code, and wanted to know how relevant DevOps is for his environment. Geoff recommended that people in non-enterprise environments (HPC, research, and university) look at the practices that one gets from DevOps and determine which practices apply.