conference reports

- Our hearty thanks go:

  To the LISA '04 scribe coordinator:

  John Sechrest

  To the LISA '04 summarizers:

  Tristan Brown

  Rebecca Camus

  Andrew Echols

  John Hawkins

  Jimmy Kaplowitz

  Peter H. Salus

  John Sechrest

  Josh Simon

  Gopalan Sivathanu

  Josh Whitlock

  And to the EuroBSDCon 2004 summarizer:

  Jan Schaumann

LISA '04: 18th Large Installation System Administration Conference

Atlanta, Georgia

November 14–19, 2004

SPECIAL AWARDS

Doug Hughes was the recipient of the first Chuck Yerkes Award for Outstanding Individual Contribution on Member Forums, and Brent Chapman was the recipient of the SAGE Outstanding Achievement Award.

CONFERENCE SUMMARIES

- ***Configuration Management Workshop***

  Paul Anderson, University of Edinburgh

  Summarized by John Hawkins

  This year's configuration workshop was attended by 27 people, from a range of academic and commercial organizations with often widely differing requirements for their configuration management tools.

  In the three years since the first workshop, there has been great-er recognition that the configuration problem extends beyond that of configuring single machines toward methods of managing collections of nodes, often in a decentralized manner or by a devolved management team.

  This workshop took a slightly different form from previous ones. Each of the four sessions followed a different theme, concentrating on separate areas of the problem. The traditional presentations of attendees' tools were not present this time, helping to avoid the "tool wars" of previous workshops. It is now widely agreed that no current tool adequately solves all the problems in configuration management, and that collaboration between researchers in the field and a refocusing on the principles behind tool design, rather than the refinement of any one tool, are required for future progress.

  As usual, a range of polls were taken. The majority of attendees regard themselves as tool developers, but a much smaller number have written tools that are also used by others. Many manage either high-performance clusters or Windows machines, and well over half indicated a strong interest in the theoretical research issues in system configuration, in addition to practical concerns of tool development.

  Some attendees felt that those within the configuration community are approaching a consensus, but others thought there is some way to go yet. The problems involved are only beginning to be specified in words whose meanings are agreed on, and there is still much duplication of work.

  Attempts were made to define a number of terms in common usage but with slightly ambiguous or overloaded meanings. "Policy" was suggested to be a description of what a machine is intended to do, including intended collaborations as well as configurations. "Service Level Agreements" (SLAs) were defined as relationships promising service within specific tolerances.

  Effort is required on the specification of intermachine relationships. A tool may correctly configure a machine in isolation, but more complex ways of capturing the details of a service specification are needed to ensure that intermachine relationships hold, thus producing the desired service behaviors.

SLAs cross a dividing line, since they require active monitoring. While it is essential that this monitoring be integrated into the tool, it should be part of a layer distinct from the specification language currently used.

As soon as dynamic properties are introduced, much of the certainty that previously existed is lost, and it will no longer be possible to tell what's true or false at any particular time, thus moving into the realm of probabilities. This is a fundamental problem that must be lived with. It was suggested that this uncertainty has at least two dimensions, time and value uncertainty.

The possibility of a set of standards for configuration was discussed, with the POSIX standard for UNIX as an analogy. The use of a low-level API for configuration was also suggested.

Mark Burgess led a session on decentralized configuration, illustrated by Ed Smith's simulation of a decentralized service manager capable of reconfiguring in response to node failure.

A move away from centrally managed systems is required to cope both with problems of scale and with vulnerability to central points of failure that are becoming problematic for large sites. However, this move brings with it reductions in predictability, trust, and relative simplicity of management. Decentralized management allows autonomy where some configuration decisions are made by the system by use of protocols, including those for negotiation and service discovery. To govern this autonomous behavior, the system must have in place an awareness of its environment that is unnecessary under central management and policy control.

There was some discussion of pervasive computing, the management of large numbers of small devices, but this is currently an open problem and it may be too early to consider it in any detail since the range of requirements of such systems is not yet fully understood.

Luke Kanies and Alva Couch talked of the difficulty of achieving more widespread adoption of configuration management tools. Barriers to adoption include the hostility of system administrators used to the current ways of working, the complex task of respecifying the configuration of the site under the new tool, and unhelpful management attitudes not aided by poor cost models and lack of trust in often inadequately proven systems.

Alva's presentation described how the cost of configuration management goes through four phases, each phase reaching a point where the cost rapidly increases and a more sophisticated approach to configuration management must be adopted. Most sites have reached the point at the end of the second phase where the configuration is managed "incrementally," for example with cfengine, but encounters problems with hidden preconditions requiring bare-metal rebuilds. They are not aware of how to make the transition to a "proscriptive" management strategy.

Site managers need to know at what point it becomes more economical to adopt a heavyweight tool such as LCFG, with huge initial costs in setup and staff training but more robust results in handling large numbers of machines.

The problem of loss of institutional memory between the "incremental" and "proscriptive" phases would be alleviated if data mining techniques could be used to pull out a large proportion of the current configuration and convert this to configuration data for the new system. There was some discussion as to whether this is intractable.

A session on case studies with Steven Jenkins and John Hawkins attempted to increase the number of specific examples of configuration problems tool users are actually grappling with. Questionnaires were distributed, and although these have yet to be analyzed at the time of writing, the response was impressive, and it is hoped that this exercise will prove fruitful.

On devolved aspect resolution, it was suggested that the system should follow the human process of resolution. Political lines are important and should be reflected by the machine aspect resolution. An expert system could be utilized to predict the impact of the choice of value.

Suggestions about where research should focus from this point included the formalization and documentation of the collective knowledge so far, provision of limited user control, description of conflicts within configurations and procedures for their resolution, mechanisms for configuration transactions, and the identification of a wider range of case-study examples with a variety of candidate tools on which to try them.

This is likely to remain an active area for some time, but there was a general feeling of optimism at the workshop that solutions to many of the problems are reachable.

Sysadmin Education Workshop

John Sechrest, PEAK Internet Services; Curt Freeland, University of Notre Dame

Summarized by John Sechrest

The system administration education workshop addresses the process of system administration education at the university level. This was the seventh year of the workshop. Previous materials for system administration course content were discussed and an earlier curriculum of how many different courses might fit together into a degree program was reviewed.

A new Web site (http://education.sage.org) was unveiled as a starting point for information collection, and an online content management tool called Drupal was explored (http://education.sage.org/drupal).

The goal of the Web site is to enable more collaboration and cooperation between groups working on university sysadmin education.

Over the last two years, there has been a reduction in the number of universities that support system administration courses. The changes in the economy are being felt in the universities.

There was a substantial discussion of just-in-time learning materials for university training for existing system administration staff and how these materials might be shared in the context of other courses. There was also discussion of how online learning modules might be useful.

The system administration education mailing list is now at sysadm-education@sage.org.

- ***Advanced Topics Workshop***

Adam S. Moskowitz, Menlo Computing

Summarized by Josh Simon

The Advanced Topics Workshop started with a quick overview of the new moderation software and continued with introductions around the room—businesses (including consultants) outnumbered universities by about 2 to 1, and the room included three former LISA program chairs and six former members of the USENIX Board or SAGE Executive Committee.

Our first topic was introducing the concepts of disciplined infrastructure to people (i.e., it's more than just cfengine or isconf, or infrastructure advocacy, or getting rid of the ad hoc aspects). Some environments have solved this problem at varying levels of scale; others have the fear of change paralyzing the system administration staff. One idea is to offload the "easy" tasks either to automation (while avoiding the "one-off" problem and being careful with naming standards) or to more junior staff so that senior staff can spend their time on more interesting things. Management buy-in is essential; exposing all concerned to LISA papers and books in the field has helped in some environments. Like many of our problems, this is a sociological one and not just a technical one. Remember that what works on systems (e.g., UNIX and Windows boxes) may not work for networks (e.g., routers and switches), which may be a challenge for some of us. We also noted that understanding infrastructures and scalability is very important, regardless of whether you're in systems, network, or development. Similarly important is remembering two things: First, ego is not relevant, code isn't perfect, and a developer's ego does not belong in the code. Second, the perfect is the enemy of the good; sometimes you have to acknowledge there are bugs and release it anyway.

After the morning break, we discussed self-service, where sysadmin tasks are traditionally handed off (ideally in a secure manner) to users. Ignoring for the moment special considerations (like HIPAA and SOX), what can we do about self-service? A lot of folks are using a number of Web forms or automated emails, including the business process (e.g., approvals), not just the request itself. One concern is to make sure the process is well defined (all edge cases and contingencies planned for). We've also got people doing user education ("we've done the work, but if you want to do it yourself the command is…"). Constraining possibilities to do only the right thing, not the wrong thing, is a big win here.

Next we discussed metrics. Some managers believe you have to measure something before you can control it. What does this mean? Well, there are metrics for service goals (availability and reliability are the big two), in-person meetings for when levels aren't met, and so on. Do the metrics help the SAs at all, or just management? It can help the SAs identify a flaw in procedures or infrastructure, or show an area for improvement (such as new hardware purchases or upgrades). We want to stress that you can't measure what you can't describe. Do any metrics other than "customer satisfaction" really matter? Measure what people want to know about or are complaining about; don't just measure everything and try to figure out what's wrong from the (reams of) data. Also, measuring how quickly a ticket got closed is meaningless: Was the problem resolved, or was the ticket closed? Was the ticket reopened? Was it reopened because of a failure in work we did, or because the user had a similar problem and didn't open a new ticket? What's the purpose of the metrics? Are we adding people or laying them off? Quantifying behavior of systems is easy; quantifying behavior of people (which is the real problem here) is very hard. But tailor the result in the language of the audience, not just numbers. Most metrics that are managed and monitored centrally have no meaningful value; metrics cannot stand alone, but need context to be meaningful. Some problems have technical solutions, but metrics is not one of them. What about trending? How often and how long do you have to measure something before it becomes relevant? Not all metrics are immediate.

After a little bit of network troubleshooting (someone's Windows XP box was probing port 445 on every IP address in the network from the ATW), we next discussed virtualized commodities such as User Mode Linux. Virtual machines have their uses—for

research, for subdividing machines, for providing easily wiped generic systems for firewalls or DMZ'd servers where you worry about them being hacked, and so on. There are still risks, though, with reliance on a single point of failure (the hardware machine) theoretically impacting multiple services on multiple (virtual) machines.

Next we discussed how to get the most out of wikis as internal tools. What's out there better than TWiki? We want authentication out of LDAP/AD/Kerberos, among other things. The conference used PurpleWiki, which seems to be more usable. There's a lot of push-back until there's familiarity. They're designed for some specific things, but not everything. You need to be able to pause and refactor discussions if you use it as, for example, an email-to-Wiki gateway. (There is an email-to-Wiki gateway that Sechrest wrote.) If email is the tool most people use, merging email into a wiki may be a big win. Leading by example—take notes in the wiki in real time, format after the fact, organize it after you're done—may help sell it to your coworkers.

Next we listed our favorite tool of the past year, as well as shorter discussions about Solaris 10, IPv6, laptop vendors, backups, and what's likely to affect us on the technology front next year. We finished off by making our annual predictions and reviewing last year's predictions; we generally did pretty well.

KEYNOTE ADDRESS

Going Digital at CNN

Howard Ginsberg, CNN

Summarized by Gopalan Sivathanu

Howard spoke about CNN's plan to move video storage, playout, and editing from physical media to file-based operations on disk. The main motivating factors behind this change are improving the speed of general operations like editing and transfer and bringing about a better means for locating archived data. He pointed out that through file operations, searching becomes much easier than pulling out physical videotapes.

According to CNN, using digital format for storing video simplifies access, eases manageability, and provides ready adaptability and a competitive edge in expediting the operations so as to bring news at the right time. Howard introduced the method that CNN plans to adopt to bring about this change: a system called "Integrated Production Environment" (IPE), whose operations would broadly be managing media assets, scheduling the various operations on the media, production, playout, and, finally, archiving the old data so that they can be accessed quickly whenever required.

While speaking about the complexity involved in such a huge transition, Howard presented numbers for the approximate storage space required for storing one day's video in high-res and low-res formats. He pointed out that, since high-res video might require around 22GB of storage per day, performing operations like editing and file transfer required for acquiring, archiving, and transferring data is a big challenge.

Howard then discussed the various pros and cons of making this transition. The major pros he pointed out are parallelism in recording and editing, sharing, versioning, flexibility, and transfer speed. The main disadvantages are absence of "at scale" reference architectures and the asynchronicity between deployment and technological improvement wherein technological development renders the system obsolete by the time it's deployed.

Howard pointed out that providing forward and backward compatibility between software and hardware is one of the important challenges in making the transition. He then described the project phases and workflow and gave a brief overview of the system architecture.

SPAM/EMAIL

Scalable Centralized Bayesian Spam Mitigation with Bogofilter

Awarded Best Paper!

Jeremy Blosser and David Josephsen, VHA Inc.

Summarized by Gopalan Sivathanu

Jeremy and David presented the paper together. Jeremy began by describing the spam problem and existing solutions such as checksumming and sender verification. Before introducing their own solution, he went into the history of Bayesian filtering and discussed its disadvantages. He then described their Bogofilter Bayesian classifier, which they implemented on a Linux and Qmail plaform. He gave the salient features of the method and its capabilities. They believe the success rate to be as high as 98–99%.

They displayed a graph showing the percentage of inbound mail blocked as spam to total inbound mail before and after implementing Bogofilter. It showed a sharp increase in the number of mails filtered after Bogofilter was installed, in late April of 2003, and the performance has persisted for over a year.

Jeremy and David described the training phase required for Bogofilter: sorting several days' worth of messages sorted into spam and non-spam teaches Bogofilter how to differentiate spam in a given environment. They explained that, after training, the parameters of Bogofilter have to be tuned by running it over a sample of presorted mail and log output and comparing error rates. They gave a 10-minute demonstra-

tion of their automated training scripts.

INVITED TALKS

- ***What Is This Thing Called System Configuration?***

  Alva Couch, Tufts University

  Summarized by Jimmy Kaplowitz

  Alva Couch talked about efforts to find a "good enough" path to configuration management. What would be ideal would be to describe to the computer the desired high-level behavior, but nowadays we still manually translate this into an implementation specification for the computers to interpret. Couch distinguished between host-based configuration, where individual machines are set up to cooperate in providing a service, and the more fault-tolerant network-based method of configuration, where a service is set up network-wide. The languages used in system configuration can either be procedural, an intuitive paradigm that shows the steps of implementation that lead to the desired result, or declarative, a nonobvious but clear form of language that simply states the desired intent and leaves it to software to determine the implementation. These contrasts reflect a continuum of rigor leading from manual and unstructured changes on individual hosts to declarative specifications applied to the entire fabric of the network. The most important aspect of a configuration system, however, is the discipline of the administrators adopting it. This is much more important than the software involved.

  Configuration specifications can either be proscriptive, specifying everything, or incremental, where some requirements are specified but others are not. Beginners, Couch says, are not proscriptive enough, allowing the presence of latent preconditions that differ among hosts. The next step beyond this is a federated network, where software can change the function of individual machines in response to usage patterns and other conditions. As network management systems gain complexity from ad hoc to federated, they become harder to start managing but easier to continue and finish managing. Simpler configuration management systems are better on smaller-scale networks and over shorter periods of time, but in large and long-lived networks these techniques help.

  Configuration languages, at their core, describe data and state, not the algorithms with which programming languages concern themselves. Existing languages to manage federated networks have classes of machines, but they are insufficient when the classes overlap. The best solutions use aspect composition to satisfy constraints that specify what properties the administrators require the network configuration to possess. These constraint-based languages are useful but hard. Configuration language theory is still in its infancy.

- ***Anomaly Detection: Whatever Happened to Computer Immunology?***

  Mark Burgess, Oslo University College

  Summarized by Jimmy Kaplowitz

  Anomaly detection in the context of computers is the process of monitoring systems and looking for anything unusual, or "funny," and maybe fixing it, too. To do this it is necessary to determine what an anomaly is. Are intrusions anomalies? What about viruses and malware? Should the system signal that regulation is needed? Regardless, the management of the anomaly detection system should be simple and flexible.

  Burgess emphasized the scientific approach to this problem. He says there are two types of knowledge: uncertain knowledge about the real world, such as that found in biology and experimental physics, and certain knowledge about the fantasy worlds of math and modeling. It is hard to define what a pattern is that an anomaly detection system would be looking for, but a rough guess would be a noticeable repetition or continuous variation. This is modeled with rules, and the knowledge we have about normal behavior is modeled with parameterized expressions.

  Anomalies can be modeled as discrete events. For this task, one can use languages anywhere on the Chomsky hierarchy of languages, ranging from regular languages to recursively enumerable languages. In practice, all discrete anomalies are finite in length, so regular languages and the regular expressions that comprise them will suffice. It is also possible to use techniques such as epitope matching or probabilistic searching of the pattern space.

  Continuous modeling of anomalies is a high-level way to attack the problem. This always involves intrinsic uncertainty, since each point represents a summary of many values. It is possible to include time in the model, considering sequences rather than individual anomalous symbols, but remembering all things doesn't usually help. On the other hand, forgetting temporal details implies certainty.

  In fact, whether the model is discrete or continuous, there will always be uncertainty, either over whether the symbol is correctly matched or over whether the shape of the trend is significant. How to deal with this uncertainty is an important matter of policy. One example of a policy question is where to set thresholds. Small variations should be ignored, but striking differences need to be dealt with. Also, one can focus on high-entropy anomalies, which are very broad, or specific and focused low-entropy anomalies.

Burgess' conclusion re-emphasizes that there is inevitable uncertainty in anomaly detection and that policy is necessary to resolve it. We lack causally motivated models to explain the reasons behind the anomalies that occur. Human beings are still the state of the art for anomaly detection. It is important not to make the sample size too large when determining whether an event is anomalous or not, since with a large enough sample nearly everything looks like normal variation. Burgess finished by saying that the next step for anomaly detection is to develop higher-level languages to describe policy.

- ***What Information Security Laws Mean for You***

John Nicholson, Shaw Pittman

Summarized by Rebecca Camus

With the prevalence of e-commerce over the past several years, more and more emphasis has been placed on guaranteeing consumers that their personal information will be secure in the business's databases and while the customer is performing online transactions. John Nicholson addressed the topic of information security by first giving the user a crash course in civics, then presenting both federal and state-level regulations and laws that businesses must comply with, and finally discussing what the government is doing to enforce these regulations.

Nicholson began by delineating the differences between state and federal regulations and how federal and state laws interact. Federal law has the power to preempt state law on controversial issues relating to individual protections. However, states have the power to make these laws stricter if they wish.

The next topic that Nicholson covered regarded federal laws. In the past few years, the federal government has passed several information security laws that all businesses must comply with. The ones presented in this lecture and brief explanations of each are:

Federal Information Security Management Act: requires each business to inventory their computer systems, identify and provide appropriate security protections, and develop, document, and implement information security programs.

Gramm-Leach-Bliley Act: requires financial institutions to protect nonpublic financial information by developing, implementing, and maintaining their information security programs.

Health Insurance Portability and Accountancy Act: requires all health-related industries (including any company that deals with any health-related businesses or companies that self-insure their employees) to install safeguards in protecting the security and confidentiality of their customers' identifiable information.

Sarbanes-Oxley Act: requires that all stored information must be accurate. In addition to the requirements of each law, they all require that companies perform frequent testing and risk assessments and fix all subsequent problems.

Prompted by the increase in identity thefts, California has led the way with state-level information security laws. Even businesses not located in California but with customers who are California residents must comply with California's new information security laws. According to California's SB 1386, a resident cannot have his or her full name or first initial and last name connected with either his or her Social Security Number, driver's license number, California identification card number, or account/
credit card/debit card number, access code, or password. In addition to SB 1386, California has also passed AB 1950, which requires all businesses (even third-party companies who have purchased information about California residents) dealing with California residents to implement and maintain reasonable security procedures to protect the information from unauthorized access or modification.

All of these regulations are being heavily monitored and enforced by the Federal Trade Commission. Businesses that violate any of these laws or regulations are subject to heavy fines and possible lawsuits.

INTRUSION AND VULNERABILITY DETECTION

Summarized by Josh Whitlock

- ***A Machine-Oriented Vulnerability Database for Automated Vulnerability Detection and Processing***

Sufatrio, Temasek Laboratories, National University of Singapore; Roland H. C. Yap, School of Computing, National University of Singapore; Liming Zhong, Quantiq International

The motivation for the work is that CERT-reported vulnerabilities have increased greatly in the period from 1995 to 2002. The increase, at first glance, is exponential; there have been decreased reports in the time period, though. To address this increase in vulnerabilities, system administrators look at vulnerability reports and use databases of vulnerabilities. However, such human intervention is not scalable to the increase in the number of vulnerabilities. The solution is to increase the speed of addressing new alerts by moving from human language to machine language reports. The goal of the work is to create a new framework and database that can be used by third-party tools by making the framework declarative and flexible.

Related work on this subject includes Purdue Coop Vulnerability database, ICAT, Windows Update, and Windows Software Update Server. ICAT is not designed for automatic responses

but is geared more toward mere vulnerability searches. Windows Update and Windows Software Update Server, two automatic patch update systems, are closed and have black-box updates. Issues with these products mentioned above include concerns that black-box scanners and updaters might leak information and do not address problems such as what is affected by the vulnerability in a system.

To produce a prototype of the framework, approximately 900 CERT advisories were examined manually. Information scavenged from the advisories included conditions for vulnerabilities to exist and the impact of vulnerabilities. A scanner written in Perl and a mySQL database were used as the prototype. A symbolic language was created and used to store a vulnerability source, an environment source, a vulnerability consequence, and an exploit.

The conclusions reached were that manual translations of vulnerabilities were not feasible due to the increasing size; the key to a better framework is the abstraction of vulnerabilities instead of new tools, and this issue needs the involvement of many organizations, including CERT, BugTraq, and SANS.

- ***DigSig: Runtime Authentification of Binaries at Kernel Level***

  Axelle Apvrille, Trusted Logic; Serge Hallyn, IBM LTC; David Gordon, Makan Pourzand, and Vincent Roy, Ericsson

  DigSig is a kernel module whose main intent is to provide protection from trojan horses. The module achieves this goal using public key cryptography, signing known trusted programs using a hidden private key, and verifying signatures at program load using the public key. Previous attempts include CryptoMark and modules from IBM Research, but DigSig seeks to contribute by being practical, simple to install, and efficient.

  DigSig uses SHA-1 and RSA encryption, supports signature revocation, and has good performance. The main problems for DigSig include the lack of support for network file systems, lack of script handling (allowing for trojaned scripts going uncaught), and no protection against vulnerabilities such as buffer overflows.

- ***I³FS: An In-Kernel Integrity Checker and Intrusion Detection File System***

  Swapnil Patil, Anand Kashyap, Gopalan Sivathanu, and Erez Zadok, Stony Brook University

  The motivation for I³FS (pronounced I cubed FS) is that intrusions are on the rise; prevention is nearly impossible, and effective intrusion handling requires timely detection. The most common ways to detect intrusions are to establish invariant, run-scheduled integrity checks and to use non-kernel programs for the detection. I³FS does all this by existing in the kernel at the file system layer and using checksums for integrity checks. A look at other tools shows that many are user tools, including Tripwire, Samhain, and AIDE. The Linux Intrusion Detection System is one of the few kernel tools. I³FS was designed with a threat model of unauthorized replacement of key files, modification of files through raw disk access, and modification of data in the network. I³FS has a stackable file system. The administrator can choose which files to protect and the policy for protecting them. The policies range from checking inode fields to the frequency at which checks are performed.

  I³FS is implemented using an in-kernel Berkeley database and a B+ tree format. With caching, performance is good.

INVITED TALKS

- ***LiveJournal's Backend and memcached: Past, Present, and Future***

  Lisa Phillips and Brad Fitzpatrick, LiveJournal.com

  Summarized by Andrew Echols

  LiveJournal started out as a college hobby project, providing blogging, forums, social networking, and aggregator services. Today, LiveJournal boasts over 5 million user accounts and 50 million page views every day. LiveJournal quickly outgrew its capacity several times over, starting with a single shared server and scaling up to five dedicated servers.

  Each step along this path had problems with reliability and points of failure, with each upgrade merely trying to remedy the problem at hand. Eventually, the user database was partitioned into separate user clusters, but as growth continued, this system became chaotic as well.

  Today, there is a global database cluster with 10 individual user database clusters, as well as separate groups of Web servers, proxy servers, and load balancers. Furthermore, master-master clusters were implemented so that each database master has an identical cold spare ready in the event of a hardware failure.

  LiveJournal also uses a large amount of custom software that has been open sourced. Perlbal is a load balancer written in Perl that provides single-threaded, event-based load balancing. Perlbal also has multiple queues for prioritizing free and paid users.

  MobileFS is a distributed user-space file system where files belong to classes. The file system tracks what devices files are on and utilizes multiple tracker databases.

  Memcached provides a distributed memory cache, taking advantage of unused system memory to reduce database hits. Memcached

is now in use at many sites, including Slashdot, Wikipedia, and Meetup.

- ***NFS, Its Applications and Future***

Brian Pawlowski, Network Appliance

Summarized by John Sechrest

Brian Pawlowski is active in the development of NFS version 4. NFS is a distributed file system protocol started in 1985. The current version is version 3; the new revision of NFS, version 4, was influenced by AFS. Brian outlined how NFS was used in the past and contrasted it with how it is being used today. Grid computing and the larger Internet are putting more demands on distributed file systems.

NFSv4, an openly specified distributed file system with reduced latency and strong security, is well suited for complex WAN deployment and firewalled architectures. NFSv4 represents a huge improvement in execution and coordination over NFSv3, also improves multi-platform support, and is extensible. It lays the groundwork for migration/replication and global naming.

NFSv4 adds a Kerberos V5 (RFC1510) authentication system as a way to create a secure distributed file system. It also provides finer grained control for access, including optionally supporting ACLs.

NFSv4 is available now in some platforms. More will be coming out over the next six months. It is available for Linux 2.6, but not by default; you must add it explicitly.

Future development in NFS may include session-based NFS, directory delegations, migration/replication completion, failover, proxy NFS, and a uniform global namespace.

Brian spent some time outlining how NFS works in a grid or clustered environment.

NFSv4 seems like a substantial transformation, which will make a significant difference for distributed file systems.

CONFIGURATION MANAGEMENT

Summarized by Josh Whitlock

- ***Nix: A Safe and Policy-Free System for Software Deployment***

Eelco Dolstra, Merijn de Jonge, and Eelco Visser, Utrecht University

Transferring software from machine to machine is hard, especially when packages don't work right. There can be difficulties with multiple versions and unreliable dependency information. The central idea of Nix is to store all packages in isolation from one another so that dependency and version problems evaporate. This is done by storing the path of each package as an MD5 hash of all the inputs used to build the package, thus producing automatic versioning. When a Nix package is installed, if there are dependencies, those packages are automatically installed too. Versioning is dealt with by having symlinks that point to the current environment. To roll back to a previous version means changing a symlink to point to the old version. To delete previous versions, the symlink to the old version is removed and the version is removed from disk. Nix thus allows for safe coexistence of versions, reliable dependencies, atomic upgrades and rollbacks, and multiple concurrent configurations.

INVITED TALK

- ***Documentation***

Mike Ciavarella, University of Melbourne

Summarized by Rebecca Camus

Mike Ciavarella presented on the importance of documentation to system administrators. However, instead of simply stating how essential documentation is to system administrators, he sparked many people's interests by comparing the work of a system administrator to Alice from the book *Through the Looking Glass*.

He began the discussion by reinforcing the fact that system administrators are very different from the users they support. Many times these differences lead to a lack of communication between the user and the system administrator, which, in turn, causes the sysadmin to feel underappreciated and leads to a sense of frustration. Such sysadmins often feel little motivation to use valuable time documenting their system.

It is essential for system administrators always to document their work, for multiple reasons. A system administrator will often be working on several projects at once, and it is very easy to forget what one has previously done on a project when dealing with several other concurrent projects. Also, the system administrator has to think of the future. It is common for things to go wrong in a system that has not been worked on recently. It is helpful to be able to reference documentation that will help solve the problem at hand. And a system administrator may move on to other projects. It is helpful to those dealing with the system to be able to reference the material that was written when the system was created.

Overall, documentation will improve the way system administrators are perceived, emphasizing their professionalism and ultimately saving time and reducing stress.

GURU SESSION

- ***Linux***

Bdale Garbee, HP Linux, CTO/Debian

Summarized by Tristan Brown

Topics included HP's commitment to Linux, stabilization of the Linux kernel, and future directions for Linux.

HP has a huge commitment to Linux across their product line.

Every division uses Linux to some degree, although some (such as the Laptop division) would like to see more. Linux represents a source of future growth, given its current immature state in the market. One problem currently faced is the variety of distributions that are used and supported, even at HP. Not only are tier-A distributions used (e.g., SuSE or RedHat), but so are near-tier-A distributions, such as some international distributions. For many people, a noncommercial distribution like Debian is required. Efforts to create one Linux strategy are underway at HP.

The discrepancies between versions of the Linux kernel can be a problem, as many libraries and applications need to be compiled against a specific kernel. This has traditionally been a problem for Linux, although recent efforts to stabilize major interfaces have resulted in less churn. This is a good thing, due to the tendency for closed-source drivers to be compiled for specific versions. Certification is also easier, since certification is typically done for a specific kernel. The end result is that the kernel can now be considered a serious tool for end users.

There are many niches Linux is now or will be entering, beyond the typical desktop or server setup. A significant portion of high-performance computing is done using Linux clusters, and Linux is beginning to enter the financial markets. At the other end of the spectrum, HP is beginning to work with Linux in the embedded space, although not much for real-time uses. They are also taking Linux to the DSP market.

NETWORKING

- ***autoMAC: A Tool for Automating Network Moves, Adds, and Changes***

  Christopher J. Tengi, Joseph R. Crouthamel, Chris M. Miller, and Christopher M. Sanchez, Princeton University; James M. Roberts, Tufts University

  Summarized by Tristan Brown

  Network administration at Princeton's Computer Science department, with more than 1500 hosts and 100 subnets, is less than trivial. To manage this system, the autoMAC tools were developed. The toolset replaces a largely manual process of entering host information into a database, configuring a switch, and connecting a patch cable with a fully automatic approach. Key to this system are several tools:

  - Web registration system. This allows administrators and users to enter configuration information that is validated and automatically entered into the DHCP, DNS, and NIS databases. This replaces a previous manual system that involved editing a file using vi.
  - NetReg server. This server answers DHCP requests for all hosts and acts as a gateway for unauthenticated users. When an unrecognized host is attached to the network, all HTTP requests are intercepted and redirected to a registration page.
  - FreeRADIUS. This technology, originally created for wireless access points, allows a device's MAC address to be used as an authentication key without any special configuration from the client end. When the switch sees a device, it asks the server for information about its MAC address. Known devices are automatically switched to their appropriate VLAN, while unknown devices end up on the registration VLAN.

    With these tools, adding a host to the database is simple to perform, and moving from one Ethernet port to another requires no configuration changes at all. Future improvements include automatic virus scanning that moves infected hosts to a quarantine VLAN. More information is available at http://www.cs.princeton.edu/autoMAC/.

- ***More Netflow Tools for Performance and Security***

  Carrie Gates, Michael Collins, Michael Duggan, Andrew Kompanek, and Mark Thomas, Carnegie Mellon University

  Network analysis for large ISP networks can involve massive data sets consisting of over 100 million flow records per day stored for a period of months. To deal with this demand for storage and processing, the SiLK tools were developed. Based on Netflow logs, the custom SiLK format takes less than half the space and has several tools to perform statistical analysis on the data.

  The SiLK logfile format starts with the directory tree, with data segregated by log date and type of traffic. Several fields from the Netflow log are removed, and others, such as time, are stored with reduced range or precision. HTTP traffic is isolated from the rest, allowing the transport type and port number fields to be reduced to two bits of data, specifying only the TCP port used. This results in a two-byte savings over other packet types and reduces each entry to less than half the size of the original Netflow record.

  To complement the compact logfile format, SiLK includes four tools to analyze traffic patterns and seven utilities for summarization. The tools all operate across the directory structure and can work with sets of IP addresses. Analysis performed with the tool set can be used by administrators in a variety of ways, such as detecting virus traffic on the network.

  The SiLK tool set is available at http://silktools.sourceforge.net.

INVITED TALK

Flying Linux

Dan Klein, USENIX

Summarized by Tristan Brown

This talk poses a simple question: Is Linux robust enough to power the computers responsible for fly-by-wire in state-of-the-art aircraft? These computers operate with real-time constraints and no tolerance for unexpected failure. Linux faces several difficulties meeting these demands.

Linux developers work on activities they enjoy, not necessarily the drudge-work required to eliminate elusive, rarely encountered bugs. Corporate developers are paid to do the work the Linux developers don't do.

The Linux source contains millions of lines of code, many of them experimental, untested, or irrelevant to a fly-by-wire system. Knowing which components to compile in is a difficult task, and tracking the interrelations between these systems is all but impossible. A GraphViz graph of the FreeBSD kernel (fewer than half the lines of code of Linux) results in a graph so complex that individual nodes and edges are no longer distinguishable.

Linux is the most exploited operating system, counting only manual attacks. The kernel receives hundreds of patches from all over the world. Which contributors can you trust? The NSA has decided to harden Linux, but are we sure vulnerabilities haven't slipped in through a back door?

The loose, open development model of Linux isn't ideal for creating a secure, stable platform. Ignoring this, however, there is yet another problem. Fly-by-wire systems are computers: they can react only to the situations they have been programmed for. Mechanical or sensor failures can result in the computer taking the wrong course of action. Human pilots have the ability to adapt to new situations much better than their electronic counterparts, although this advantage is steadily shrinking.

The unfortunate conclusion is that Linux, like its penguin mascot, may never fly. The difficult task of creating a hardened, real-time system can be left to purpose-built software, and Linux can remain the general-purpose operating system it was designed to be.

SPAM MINI-SYMPOSIUM

Filtering, Stamping, Blocking, Anti-Spoofing: How to Stop the Spam

Joshua Goodman, Microsoft Research

Summarized by John Hawkins

This talk gave a broad picture of the difficulties currently being caused by widespread proliferation of spam, the techniques recently adopted by spammers to get around ever more complex spam-filtering tools, and possible solutions to further deal with spam.

A number of statistics were presented, including the results of a poll in which 40% of respondents stated that spam overload was the biggest problem faced by IT staff. Spam has increased from around 8% of all email in 2001 to at least 50% currently. Wasted time dealing with unwanted messages, offense caused by often obscene content, and a lowering of trust in email systems make spam a major problem.

Examples of different spammer techniques were given, such as enclosing content in an image, swapping letters of words, and using coded HTML to confuse filters. Techniques for gaining access to computers to turn them into spam relays were also mentioned.

A range of methods to deal with spam were discussed, some of which may be combined to increase effectiveness: Better filtering and use of machine-learning techniques will identify more complex patterns: using honeypots, which should never receive "good" mail, to identify messages that are definitely spam; charging small amounts per message as an economic barrier to profitable spam; computational challenges requiring a calculation on the client, making it difficult to send many messages simultaneously. Many of the techniques discussed are only applicable in certain situations, and most have significant drawbacks. Some proved unpopular with the audience, which was composed mainly of email admins.

Approaches for anti-spoofing email addresses, types of non-email spam, and legal approaches to tackling spam, such as the Can-Spam Act, were also mentioned.

The speaker pressed the case for a specific conference to cover spam-related issues, due to the increasing seriousness and complexity of the problem.

Lessons Learned Reimplementing an ISP Mail Service Infrastructure to Cope with Spam

Doug Hughes, Global Crossing

Summarized by John Hawkins

In contrast to the previous talk by Joshua Goodman, which gave a wide-angle view of current spam issues, this talk discussed the specifics of dealing with spam while managing an engineering mail platform with 200 users, a number of ISPs including one with 1096 users, and 4 to 6 million mails processed a day.

The talk began with some statistics, identifying the source and nature of the spam dealt with and how some of the trends are changing. Of the mail received from Italy, for example, 90% is currently being blocked as spam. It was found that 98% of spam could be blocked by examining the headers alone.

The main part of the talk was more technical, covering how the sites' mail systems were configured to deal with spam. The majority of the filtering employs a modified

version of smtpd using a binary search tree to store logs, which vastly reduces search time. Sender addresses can be compared with those previously seen to assess whether a message is likely to be genuine.

The system was shown to be very effective, blocking up to 98% of spam while giving virtually no false positives.

INVITED TALK

- ***Grid Computing: Just What Is It and Why Should I Care?***

Esther Filderman and Ken McInnis, Pittsburgh Supercomputing Center

Summarized by Josh Whitlock

There are different types of grid computing, including utility computing (where processing cycles are for sale), distributed computing, and high-performance resource sharing. What grid computing actually is lies between these types. Examples of grids include the CERN Large Hadron Collider Computing Grid, supporting approximately 12 petabytes of data per year with 6000+ users, and the Electronic Arts Grid for the Sims Online game, with approximately 250,000 players. The challenges of using grid computing include flexible virtual organizations having different site policies and disparate computing needs.

Components of grids include security, toolkits, job management, data movement, and Web portals. Security for grid computing is basic X.509. The standard toolkit is Globus. Condor is a job manager for grids, and while flexible and portable, it is not open source or simple to use. Data movement is supported with programs such as GridFTP, which is built on top of regular FTP but is used for high-performance, secure, and reliable data transfer. Web portals are used to provide a consistent front to a grid that makes maintenance less difficult.

Grids are most successful when they have a purpose: setting up a grid at a university that everyone can use for their own purpose is not a good idea, but setting one up for a research project is a good idea. Political challenges in running a grid include coordinating work between sites that don't trust one another and implementing common policies. Technical challenges include having interoperable software between sites and synchronizing upgrades to the grid software.

MONITORING AND TROUBLESHOOTING

Summarized by Andrew Echols

- ***FDR: A Flight Data Recorder Using Black-Box Analysis of Persistent State Changes for Managing Change and Configuration***

Chad Verbowski, John Dunagan, Brad Daniels, and Yi-Min Wang, Microsoft Research

The Flight Data Recorder concept presented in this talk targets auditing, configuration transaction, and "what if" scenarios. In auditing scenarios, the FDR would assist in troubleshooting by identifying what is on a machine, and by answering questions like what is installed, where, when, and by whom. Changes are then grouped logically into change actions.

These change actions can be treated as transactions, allowing groups of changes to be automatically rolled back if they are no longer wanted, poorly done, had an adverse effect, or were malicious.

The information gathered also aids in answering "what if" questions regarding the impact of certain changes. This requires the formation of "application manifests," which are records of an application's frequency of interactions, all states read, all states written, and how often it is run.

The approach taken in the research is to get into the lowest level of the OS. There, state interactions can be monitored and understood. To be useful, information gathered must then be analyzed and presented in a way useful to humans. This is implemented as a service that logs gathered information locally. Periodically, the logs are uploaded to central storage, where they can be processed and inserted into a database. Finally, a client may view this information online or retrieve it for offline use.

- ***Real-Time Log File Analysis Using the Simple Event Correlator (SEC)***

John P. Rouillard, University of Massachusetts, Boston

All forms of log analysis result in false reports, but the Simple Event Correlator aims to perform automated log analysis while minimizing errors through proper specification of recognition criteria. Information can be gained from logs by looking for certain patterns. These patterns may appear across multiple logs. Absences of events and relationships between events should also be noted.

Relationships between events may take the form of one event taking place before or after another, or in a sequence. Events may also be coincident within a certain period of time. If there is some event that occurs periodically, its absence would be significant.

Relationships between events are important for combining events. For example, some program may emit an error without a username attached to it, but only after another related event that contains a username. Correlation may take place by watching for one event, then expecting another within a certain amount of time that can be matched with the first.

SEC allows for analysis of logs in real time using simple rules for single events, as well as many complex rules which support

matching multiple events within given windows or thresholds and as pairs. With proper use of these rules, relationships between events such as those mentioned above may easily be detected.

INVITED TALK

- *A New Approach to Scripting*

Trey Harris, Amazon.com

Summarized by Josh Whitlock

People think of scripts being different from programs. People think of Perl, Python, and bash shell code as scripts, while they think of C, C++, and Java code as programs. Scripts are interpreted, while programs are compiled. In fact, scripts are a subset of programs. They are programs that make heavy use of their environment (files and directories), define few complex data structures, have no outer event loop, are run by either the author or administrator, and have a primary purpose of ensuring a given state is achieved in a system. Scripts should be distinguished from programs because good programming methodology is well researched while good scripting methodology is not.

As an example, consider a script that wants to mount a remote file system via NFS. If the script does nine things, there are nine places where the script can fail. Restarting the script when it fails could result in unwanted and incorrect duplication of tasks (e.g., multiple mountings). Adding error checking code to the script simply convolutes the code. The problem with error checking code is that it is syntactic and not based on implementation. The error checking needs to be semantic. The Commands::Guarded command set for Perl is available from CPAN. Each guard consists of two parts: a condition, or guard statement, and an executable statement. For example:

```
ensure { $var == 0 }
using { $var = 0};
```

means "if $var is not zero, then set $var equal to zero." To effectively use guarded commands, decompose the code into atomic steps. For each step, write the necessary and sufficient condition for the step to complete. Use the guards as the conditions for each step.

There are many benefits to using guarded commands. They make scripts more resilient because they make testing semantic. If the environment changes, the script is more likely to continue to function.

Guarded commands reduce the amount of error-checking code, thus making the code easier to read and maintain. If a script terminates prematurely, the script will pick back up exactly where it left off, thanks to the guards.

GURU SESSION

- *AFS*

Esther Filderman, The OpenAFS Project

Summarized by Peter H. Salus

Esther (Moose) Filderman knows more about AFS than anyone else I've ever encountered. She illustrated this knowledge and her quick wit for about an hour, at which time the session dissolved into a sequence of queries from the floor, comments, and idle remarks.

She began by noting that "AFS was rather clunky and difficult to administer, but then the universe changed." What happened, of course, was that it moved from Carnegie Mellon to a startup called Transarc, which open-sourced "Andrew."

Originating in 1986, the core has been maintained and guided since 1988 (when Moose joined the crew). It is now "tightly secure" with a Kerberos-like mechanism. "AFS is a good complement to Kerberos," Moose said. It has the structure of a distributed file system, and users can't tell what's what: everything looks like /afs.

There are lots of protections available, but AFS is still "slow," though "speed is coming along nicely." The biggest problem (of course) is with the I/O. "Read times are better; write times are getting better."

Apparently, when there are lots of path or sharing changes, one should opt for AFS over NFS.

Definitely interesting and worthwhile.

SYSTEM INTEGRITY

Summarized by John Sechrest

- ***LifeBoat: An Autonomic Backup and Restore Solution***

Ted Bonkenburg, Dejan Diklic, Benjamin Reed, Mark Smith, Steve Welch, and Roger Williams, IBM Almaden Research Center; Michael Vanover, IBM PCD

LifeBoat is a backup solution designed for ease of use. The goal is to reduce the total cost of ownership for a system by reducing client support costs.

Over 50% of current support costs involve PC clients. While these PC clients often have mission critical software on them, the underlying server automation generally does not reach down to the PC clients. LifeBoat is aimed at resolving this by creating a complete rescue and recovery environment.

Targets for the backup system can be network peers, dedicated servers, or local devices. This means that the backup must include file data as well as file metadata. In order for it to be user friendly, it must enable users to restore individual files. More critically, it must support special processing for open files locked by Windows OS. LifeBoat provides a kernel driver to obtain file handlers for reading locked files.

In order to manage the backed-up data, LifeBoat uses an object storage device called SCARED that organizes local storage into a flat namespace of objects. The data is

not interpreted; it can be encrypted at the client and stored encrypted on the storage devices. These storage devices authenticate clients directly, supporting a method of keeping the data pathways secure.

LifeBoat can use centralized servers, peers, or local devices for backup. This offers some flexibility in the overall problem of finding a good place to put data when you are on the road with your laptop.

LifeBoat is packaged as a Linux boot CD providing software used for maintenance. This allows for a rescue and restoration of a system when there is a problem.

While perhaps the term "autonomic" is being overused in this case, the system looks as though it enables users to easily back up systems, including mobile systems. This is one of the great problems in an organization supporting PC systems. And the solution looks as though it will integrate well in a corporate environment supporting autonomous server configuration strategies.

- ***PatchMaker: A Physical Network Patch Manager Tool***

  Joseph Crouthamel, James Roberts, Christopher M. Sanchez, and Christopher Tengi, Princeton University

  What do you do when you have lots of networks and wires, with 1500 hosts, 675 switch ports, and 1100 patch ports, and you want to keep track of what is plugged into what? Often a huge amount of the time spent debugging a problem involves locating the port causing the problem. Hand-tracing old wires is very time-consuming.

  There used to be a Perl CGI for patch information, but it did not scale very well. As the port count increased, they felt they needed a better tracking system. By putting a Web interface on the front end, it became more useful for a broad group of people.

  This program has a patch database, which is searchable. This is support port monitoring and management and enables direct changes in the VLAN information. It allows for the cables to be documented and for the switch to be managed directly and to track information about each host. It makes it more effective by presenting a visual view of a patch panel.

  This is an open source package written in mySQL, PHP, DHTML, and CSS. It supports SNMP/Sflow and uses Mrtg/nMon to monitor network activities.

  They keep information on patches, racks, hosts, panels, port count, SNMP, and more.

  In the future, they hope to add GUI creation, user authentication and privilege levels, change logging, QoS/rate limiting, security ACL management, switch configuration, file management, and end-user PortMaker.

  This project looks like a local solution that is trying to evolve into a set of broader network management tools. That the whole site can be managed through managing the configuration is an important step forward toward bringing automation to network management. It is too bad that they did not then integrate this data and structure with a configuration management system. When hosts are deployed by a management system, you need to deploy the network as well. This package will work for people to patch by hand, but it has not progressed to the point where a content management system could do it.

  http://www.cs.princeton.edu/patchmaker

- ***Who Moved My Data? A Backup Tracking System for Dynamic Workstation Environments***

  Gregory Pluta, Larry Brumbaugh, William Yurcik, and Joseph Tucek, NCSA/University of Illinois

  NCSA has a lot of highly mobile computers, employees, and students. With increased laptop use, in particular, they found that they had to rethink their approach to data management.

  Far too many people look at laptops as a way to provide persistent data storage, and this puts the data at risk. As the systems move in and out of the network, the backup process can easily fail.

  NCSA started to look at the percentage of laptops vs. percentage of backup success rate. They wanted to be able to find the important data that was vital to the organization.

  They set up a backup tracking system, with authentication servers. This integrates with a Web-based application to go through the data and search for material. It makes a list of machines and users who have not been backed up and allows them to work to increase the backup success rate.

  This talk clearly shows that for organizations that have information as their main product, it is vital to work out a meaningful backup process. And for the most part, people do not back up PCs and they don't back up laptops. NCSA's system provides an organizational framework to enable these backups. This was a good measure for NCSA, but it leaves me wishing for a good distributed file system that knows what to do with mobile users.

SPAM MINI-SYMPOSIUM

- ***What Spammers Are Doing to Get Around Bayesian Filtering & What We Can Expect for the Future***

  John Graham-Cumming, Electric Cloud

  Summarized by Jimmy Kaplowitz

  Graham-Cumming started by describing several spam-related trends that have occurred in the past year or so. Most major email clients support adaptive filtering, with the notable exception of Outlook. Spam is getting slightly sim-

pler and less tricky, but not very fast. Spam is also using Cascading Style Sheets.

The speaker then described specific tricks spammers are using, ranging from hard-to-read color and Internet Explorer's odd handling of hexadecimal color codes to Web bugs and nonexistent zero-width images. From July 2003 to April 2004, email software firm Sophos has noted the relative frequencies of certain tricks. Spammers try to hide red-flag words and include innocuous words. About 10% of spam uses obscure Web site addresses, 20% uses simple trickery such as the insertion of spaces or punctuation characters into red-flag words, and another 20% inserts HTML comments to break up red-flag words. About 80% of spam uses trickery of some sort, but the percentage that doesn't is increasing. Even some bulk email software manuals point out that anti-filtering tricks often make spam easier to filter. There are even spam-sending programs that include SpamAssassin technology to allow the spammer to test and tweak their mail for maximum penetration. A very common occurrence in spam is for the text/plain MIME part to contain very different content from the text/HTML MIME part.

The speaker presented seven tough questions to ask vendors of anti-spam software. The first question is, "How do you measure your false positive rate?" Claims of 99.999% accuracy are meaningless, since a mail filter that deletes all mail can claim to have removed 100% of all spam. Ask what the false positive and false negative rates are, and how the vendor knows. Another question is, "How often do you react to changes in spammer techniques?" The third question, designed to cut through the hype, is, "What are your top two ways of catching spam?" In order to prevent spammers from receiving feedback on your reading preferences, another important behavior to confirm is that the software prevents Web bugs from firing. It also must properly handle legitimate bulk mailings as non-spam, deal gracefully with user reports of legitimate mail as spam, and have a good method to separate non-English spam from other non-English mail.

At the end of the talk, Graham-Cumming predicted that spammers will continue to reduce their use of obfuscations and other trickery and that the set of popular spam tricks will continue to change.

PLENARY SESSION

- ***A System Administrator's Introduction to Bioinformatics***

Bill Van Etten, The BioTeam

Summarized by John Sechrest

BioTeam is an organization of scientists, developers, and IT professionals who have an objective—vendor-agnostic professional services aimed at solving large-scale bioinformatics problems for large companies.

Bill Van Etten provided a quick but detailed introduction to genetics and genomics. He then went on to talk about how computing is impacting current biological activities.

HISTORY

- 1866 Genetic theory published—Mendel
- 1869 DNA discovered—Miescher
- 1952 DNA is genetic material—Hershey
- 1953 DNA structure—W&C
- 1959 Protein structure determined—Perutz, Kendrew
- 1966 Genetic code
- 1977 DNA sequenced—Sanger
- 1988 Human Genome Project started
- 2001 Human genome sequenced

GENETICS TRIVIA

- Everything you are is either protein or the result of protein action.
- Proteins are folded strings of amino acids (20).
- Proteins' structure is important.
- Genes are a hunk of DNA that defines a protein.
- There are 3 billion DNA letters (made up of only 4 characteristics) in the human genome.
- Five percent of human DNA contains genes.
- 999/1000 DNA is identical between any two people.
- Human genes are 98% the same as those of a chimpanzee.
- Human genes are 50% similar to those of a banana.

In 1953, it was discovered that DNA structure is a 3D structure in the form of a double helix. In 1968, the DNA code was broken using Sanger DNA sequencing. DNA of all possible lengths from a known starting point is treated. Each strand ends with a radioactive dideoxy nucleotide which terminates the chain. The strands are weighted.

You get something like:

ACTGAGTGAGCTAACTCA
CATTAATTGCGTT

It all happens in a single capillary tube, and the results are read via laser spectrometer. This leads to high throughput sequencing (see http://www.sanger.ac.uk/Info./IT).

And this leads to an explosion of public sequence databases, which, in turn, leads to a large number of computing related activities: genetic mapping sequence analysis, genome annotation, functional genomics, comparative genomics, expression analysis, coding regions, and genetic coding.

The growth of the genetic scientific inquiry has followed the growth in computing power. All of this information is able to be processed through computer-aided data analysis.

There are some interpersonal differences between wetlab people and computer scientists: Wetlab people know that biology is unreliable but think computers work

well all the time. Computer people think that biology works great but that computers have problems all the time.

The half-life of scientific information is five years.

SEQUENCING SEARCHING ALGORITHMS

There are several sequencing searching algorithms, which build and map sequences. These include Blat, Blast, and HMMR (Hidden Markoff Matching => HMM). They involve many pattern-matching approaches. Each tool is particular to each pattern and structure.

BioInformatics is full of problems that are embarrassingly parallel. There is a great deal of data, but it is generally easy to decompose the problem into a number of little problems. This type of problem leads to a great many cluster and grid-style solutions.

But because many of the people who are trying to use these systems are scientists who have a hard time with computing details, they found that a portal architecture was a fabulous step forward.

All the gory details of LSF, Grid Engine, and Condor are really distractions for scientists who are just trying to get data back out.

While clusters are cheap ways to increase computing power, they are often hard to build, manage, and use. It is also difficult for scientists to map computing to computers in the cluster, making it hard to achieve high throughput and high performance .

BioTeam has developed a tool called iNquiry, which is a rapid cluster installer with a persistent graphical user interface. It is built around the Apple Xserver. You can see an example of it at http://workgroupcluster.apple.com. (Ask for a login.)

- It supports the automatic deployment of a cluster for biocomputing, including: network services; DRM—LSF, etc.; admin tools; monitoring tools; biological applications (200).

  The most impressive idea that I got from this talk was that they wrote a DTD to create a command line interface and write an XML description of the command-line which was used to generate Web interfaces for command lines. This hides system from the scientists in an elegant way that has leverage.

  As a serious win on the cool marketing idea, they used an Apple iPod to build the cluster. All the data is on the iPod, and it drives the boot and install process (and the sysadmin gets to keep the iPod).

  While this works well on the Mac OS X-based Xserver, it works less well on some Linux system hardware platforms. It uses a system imager.

  The core work for the DTD/XML solution was done by Catherine Letondal. It is called pise:

- http://www.pasteur.fr/recherche/unites/sis/Pise/CCP11/s6.html
- http://www.pasteur.fr/recherche/unites/sis/Pise/
- http://bioinformatics.oupjournals.org/cgi/reprint/17/1/73.pdf

  For more details that might interest you, read Bill Bryson's book *A Short History of Nearly Everything*.

  Web site: http://bioteam.net

SECURITY

Summarized by Tristan Brown

- ***Making a Game of Network Security***

  Marc Dougherty, Northeastern University

  The idea is to make a closed, firewalled network with a collection of hosts running various unpatched and insecure services. Give each team a host, and give the teams two days to defend their machines while exploiting everyone else. The result is Capture the Flag tools, a framework for online war games. Run services on your host and gain defensive points. Compromise other people's servers and gain offensive points. The toolkit facilitates the competition in two ways:

  1. Service verification. A flag service is run on each host, allowing the central server to perform confirmation of the actual flag file on the server. The process allows the central server to determine what team is in control of each server and to defeat simple cron jobs designed to ensure that one team's flag file stays in place. This is a complex procedure and is currently being overhauled to become even more robust.
  2. Scoring. A scoreboard showing each team's progress is kept available. This simple touch is an effective motivator to keep people working on the project.

  The tool framework has room for future expansion. Different games can be implemented (verification is based on Perl scripts), and a larger game based on a large network of networks or autonomous systems is being planned. More information can be found at http://crew.ccs.neu.edu/ctf/.

- ***Securing the PlanetLab Distributed Testbed: How to Manage Security in an Environment with No Firewalls, with All Users Having Root, and No Direct Physical Control of Any System***

  Paul Brett, Mic Bowman, Jeff Sedayao, Robert Adams, Rob Knauerhause, and Aaron Klingaman, Intel Corporation

  PlanetLab is an environment with unique security requirements—the system must remain secure despite the fact that owners and users of each system have root access on the machines. To ensure availability, several tools are used. Nodes run virtual servers to allow users to operate as root in their virtual server but not in any oth-

ers. Packets are tagged to identify the originator of all data, thus ensuring accountability among the research-ers. Traffic control is utilized to ensure that no one node uses excessive system bandwidth. In the event of a compromise, systems can be remotely rebooted with a magic packet, causing them to start off of a CD. The machine then enters debug mode, restricting access and allowing remote forensics to be performed. Patches may be applied, bringing the node back to a known, safe state.

Because of all the measures taken to protect the network, a large compromise resulting in rooted nodes was detected and contained within minutes. Further security reviews since have resulted in an effort to decentralize administration and eliminate reusable passwords. To support future expansion, a federated decentralization scheme is being implemented.

- ***Secure Automation: Achieving Least Privilege with SSH, Sudo, and Suid***

  Robert A. Napier, Cisco Systems

  Automation and scripting across the boundary between hosts is a difficult process often subject to security vulnerabilities. Although insecure tools such as Telnet and RSH may no longer be used, SSH and sudo can still be used to exploit a system. A design paradigm of small, simple scripts that do one task very well and with the least privilege possible can help to prevent unauthorized access due to an improperly written script. Several techniques can be used to harden scripts:

  - SSH command keys can be used to perform a command on another host without the possibility of using the session to execute a shell. This ensures that a script can only perform its single task on the remote machine, and it keeps attackers from performing unwanted actions on the remote machine.
  - Properly configured sudo can give scripts the ability to do just what they need and no more. It should not be used to give scripts complete root access.
  - Scripts should have a unique user ID to isolate their domain on the system even further.
  - Setuid and setgid can both be used to allow a script to work as another user. Setgid has the advantage that fewer exploits have been performed against it. Setuid to root is generally a bad idea for a script.

  By following the principle of least privilege and asking, "How much access do I need?" rather than "How much security do I need?" overall vulnerability due to scripts can be reduced.

INVITED TALK

- ***System Administration and Sex Therapy: The Gentle Art of Debugging***

  David Blank-Edelman, Northeastern University

  Summarized by Peter H. Salus

  Blank-Edelman's underlying thesis is that we can improve our system administration through learning from other fields. He likes sex therapy. This is because:

  - Debugging is getting harder.
  - Debugging is not merely binary.
  - Both fields (sysadmin and sex therapy) deal with complex systems.

  The phenomena are harder because of interdependency; because we don't control the "parts" and haven't written the software; as availability increases, the level aided decreases.

  We all feel that "sex should just work." Blank-Edelman listed a number of male and female myths concerning sex that he found in several therapy books. He then moved to "system administration should just work," where the principal assumptions involve:

  - Plug and play
  - It's just (merely)…
  - No administration toolkit
  - Printing
  - The Internet
  - "You have the source, right?"

  Blank-Edelman analogized between sex therapy and Agans' (2002) "Nine Indispensable Rules":

  1. Understand.
  2. Make it fail.
  3. Look at it.
  4. Divide and conquer.
  5. Change one thing at a time.
  6. Keep an audit trail.
  7. Check the plug.
  8. Get a fresh view.
  9. If you didn't fix it, it isn't fixed.

  Good advice.

  The comparative metaphor was attractive and Blank-Edelman is a good presenter, but it wore thin long before the 90 minutes were up, and there were too many old and trite jokes and anecdotes.

GURU SESSION

- ***RAID/HA/SAN (with a Heavy Dose of Veritas)***

  Doug Hughes, Global Crossing; Darren Dunham, TAOS

  Summarized by Andrew Echols

  The RAID/HA/SAN guru session consisted of a Q&A, primarily about using Veritas products for RAID, high availability, and SAN setups.

  The gurus recommended a few mailing lists for questions related to the session topics. Each list is at <list-name> mailman.eng.auburn.edu, and uses <list-name>-request mailman.eng.auburn.edu to subscribe:

  - Veritas Applications: veritas-app
  - Veritas Backup Products: veritas-bu

- Veritas High Availability Products: veritas-ha
- Veritas VxVM, VxFS, and HSM: veritas-vx
- Sun storage appliance: ssa-managers

A small sample of the topics discussed follows.

Regarding experience with VxVM and Sun Cluster, it was noted that Solaris 10 has a new file system, ZFS, which has its own volume management. There are known issues, but Sun is likely motivated to drop the requirement for VxVM to use Sun Cluster. Sun is also pushing ZFS as something that just does the right thing and hides the details.

In another case, a user recently started using Veritas Foundation Suite 4 and wanted to reduce the number of inodes. There appeared to be a very high amount of overhead, but it was suggested that there might be differences between definitions of gigabytes (2^30 vs. 10^9) between programs. It was also noted that there may be problems with the size of the private region.

Commenting on the maturity of the Linux Volume Manager, the gurus felt that it had some nice features, primarily a combination of those provided by Veritas and AIX. The naming scheme is good and fairly robust. It still needs mirroring, but that can be worked around at the RAID level.

THEORY

Summarized by John Sechrest

- ***Experience in Implementing an HTTP Service Closure***

Steven Schwartzberg, BBN Technologies; Alva Couch, Tufts University

This talk is a part of an ongoing discussion in configuration management. How can you create a system which can programmatically be configured in a coherent and consistent way?

Two concepts put forward are "closures" and "conduits." Closures act as an element of the network and conduits are the mechanism by which closures talk to each other.

A closure is a self-contained application that is self-managing and self-healing and where all operations affecting the closure must be within this or another closure.

If this is so, then the closure will:

- Provide consistency
- Create a common command language
- Operate the same way regardless of OS or other applications
- Provide security
- Provide a single interface into the closure
- Detect intrusion and potentially repair modifications
- Hide complexity
- Understand the dependencies and be a mediator
- Provide configuration language in plain English
- Eliminate typographical errors

All of these things are good ideas.

This was proposed in an earlier paper, but how does it work in reality? This talk covered the experience of building an HTTP closure:

1. Start with RH 9 w Apache.
2. Create a protected directory.
3. Wrap Apache binaries with scripts.
4. Move all HTML, logs, configuration files, and modules into the repository.
5. Create a Perl script to allow management of the closure.

At the first iteration of the closure language, they supported a limited command set:

- assert—Create a virtual domain.
- post—Upload a file into the domain.
- retract—Remove a file from the domain.
- allow—Grant permission to users and modify configuration options.
- Deny—Remove permissions as above.

For example:

```
Assert foo.edu
post info.html
www.foo.edu/info.html
allow cg www.foo.edu/cgi-bin
retract www.foo.edu/
information
```

But this leads to problems and ambiguity. They had problems with how they specified files vs. directories, the renaming of files, and dealing with indexes.

In order to address these problems, they tried to gain idempotence (making order not matter) by hiding the complexity of the commands and trying to create statelessness. Post must only deal with directories, so you load the whole site each time.

This was an interesting experiment. It showed that there were many places where procedural complexity and all the details of a configuration make it hard to present a simplified solution. This was a good first step in understanding how a closure might be built. However, Apache seems like a difficult example to process. Perhaps a simpler service would be a better starting point for illustrating the idea of closure.

- ***Meta Change Queue: Tracking Changes to People, Places, and Things***

Jon Finke, Rensselaer Polytechnic Institute

At RPI all of the data about all of the different systems is stored in databases.

Getting this data where it needs to go, and doing it in a timely manner, has taken some thought. Many of the issues are related to who owns the data and who needs the data. Often these are in different parts of the organization.

Different data sets—CMMS, LDAP, ID card, Space—need to be integrated into the process. By setting up a Meta_Change_Queue Table, the changes to the data sets can be identified. A Listener can be set up to catch and process these data changes in a low-overhead way.

This table uses a vendor-provided trigger to input real-time data. This protects the business logic of the program, which allows third-party programs to work completely and still be integrated into the larger campus data process.

Through this mechanism a single sign-on system for campus has been created and maintained.

In order to do this, Jon needed to:

- Index a key column and set to null after processing
- Import with a trigger, which works well with Oracle target systems
- Include a low-priority batch queue

A key idea is to use a trigger to catch deletions and move things to the history table on deletes.

All source code is available on Jon's Web site (www.rpi.edu/~finkej/), but it is not packaged and is mostly in PLSQL.

- ***Solaris Zones: Operating System Support for Consolidating Commercial Workloads***

Daniel Price and Andrew Tucker, Sun Microsystems, Inc.

Solaris Zones are a way to run multiple workloads on the same machine. They are akin to Jails in BSD. Because they all use the same kernel, they gain efficiency, but they do not gain the independence of multiple kernels like User Mode Linux. These are not virtual machines.

A set of processes look the same at the base machine. It supports a reduced root-permissions structure, in order to prevent escaping from a zone as you can from a chroot.

They support per-zone resource limits.

This was an interesting introduction to Solaris Zones. However, it did end up coming off as a sales talk more than a comparison with the state of the art. Solaris Zones are a good thing, just as Jails are a good thing. I wished there had been a broader perspective in this talk.

INVITED TALK

- ***Used Disk Drives***

Simon Garfinkel, MIT Computer Science and Artificial Intelligence Laboratory

Summarized by Jimmy Kaplowitz

Simon Garfinkel explained that much data is frequently left behind when a computer owner tries to remove it from the hard drive. This is because of the common choice to use quick but vastly incomplete erasure methods such as the DEL and FORMAT commands of Micro-soft operating systems. The data left behind by these tools stays around for a long time, since hard drives are quite long lived. Physical security and OS-level access controls fail to ensure data privacy when drives are repurposed or given to a new owner. Cryptography, which would provide that assurance, is rarely used on-disk.

The author did a study of 235 used hard drives, some of which were obtained for as little as $5. The contents of the drives included runnable programs and private medical information. The data was deleted to different degrees, so the author developed several different levels of deletion to keep these separate. Level 0 files are ordinary files in the file system. Level 1 files are temporary files in standard temporary directories. Level 2 includes recoverable deleted files, such as those in DOS where only the first letter of the filename is lost. Level 3 is partially overwritten files.

An overview of digital forensics tools was given. Among hard-drive forensics tools, the two main categories were consumer and professional tools. All of them were able to undelete files at level 2 and search for text in level 3 files. The professional tools also had knowledge of file formats, were able to perform hash-based searching, and could provide reports and timelines useful for auditing and legal testimony purposes.

The author then described the challenge of performing forensics on many drives at once with very little time for each drive, explaining some of his choices of tools as well as some of the gotchas he had to overcome. In total, he worked with 125GB of data. The data included very common parts of the OS, Web caches, authentication cookies, credit card numbers, email addresses, medical records, personal and HR correspondence, personnel records, and similar items.

Garfinkel told several more horror stories about really sensitive data being revealed, and suggested the creation of a US privacy commissioner or some other federal waste-auditing role. He cautioned that all of his warnings about recoverable data on hard drives applies equally well to USB drives, digital cameras, and other types of media. However, when faced with a police officer ordering deletion of (for example) a picture on a digital camera, the recoverability of dirty deletion is usable to preserve one's data.

The author mentioned the Department of Defense's standard for sanitizing media containing non-classified data and then proceeded to discuss other tools for overwriting media. The tools discussed range from standard UNIX dd to commercial tools such as Norton Disk Doctor. It was pointed out that Windows XP and NT hide a little-

known sanitization program in their installation; it's called cipher.exe.

The author discussed the exotic threat of hard drives with hostile firmware that lie about their state and snoop on other hard drives in the same system. It is even possible for such hard drives to infect other drives with hard-drive viruses. There is a level 4 part of the disk, the vendor area, which is where the firmware lives. This can be overwritten by really expensive sanitization tools.

The talk concluded with a sobering question: If the author was able to get so much private info for under $1000, who else is doing this?

- ***Work-in-Progress Reports***

  Summarized by John Sechrest

  WiPs are brief introductions to topics that people are currently working on. Seven people stood up and gave rapid descriptions of ongoing projects.

  - *OGSAConfig*—A proof of concept implementation of an architecture to make grid fabrics more manageable

    Edmund Smith, University of Edinburgh

    Split the problem down for a grid by allowing each node in the grid to get a different profile. You can do this with a constraint-resolution problem.

    OGSA config project: http://groups.inf.ed.ac.uk/ogsaconfig/

  - *CfengineDoc*—A tool which uses GraphViz and NaturalDocs to document the import execution order of cfengine scripts

    Christian Pearce, Perfect Order

    How to describe the meaning of a cfengine file. Used Inkscape to draw a cfengine config file and then discovered GraphViz. Then got AutoDoc and found NaturalDocs. Use it to describe SysNav Interface.

    http://cfenginedoc.com
    http://naturaldocs.com
    http://autodoc.com
    http://cfwiki.org
    http://sysnav.com
    http://inkscape.com

  - *Verifiable Secure Electronic Voting*—An electronic voting system that allows voter verification of individual votes, and makes a passing nod towards being a secure and private voting environment

    Zed Pobre, Louisiana State University

    Use a UID + voting ID + MD5. Supports many different voting styles.

  - *Grand Unified Documentation Theory*, or What's wrong with current documentation standards. Can we fix it?

    David Nolan, CMU

    Trying to create an interface to a single place for documentation using a wiki front end, with some customizations for raw access and a choice of editor.

    vitroth+lisa04@cmu.edu
    sadm-l@lists.andrew.cmu.edu

  - *CADUBI: Creative ASCII Drawing Utility by Ian*

    Ian Langworth, Northeastern University

    Received Best WiP Award!

    CADUBI is in all kinds of ports. It is Old Crap, very old.

    CADUBI 1.2: two maintainers to the last version.

    Go AWAY . . . Use TetraDraw.

    (This was incredibly humorous.)

  - *Geographic Firewalling*

    Peter Markowski, Northeastern University

    Wireless firewalls are natively geographic entities. This way you can choose where people get services. Limit your perimeter.

  - *Portable Cluster Computers and InfiniBand Clusters*

    Mitch Williams, Sandia National Laboratories

    Received 2nd Best WiP Award!

    A cluster in a box. It is one foot tall, has four CPUs, and by using LinuxBios it can boot in 12 seconds. Working on adding infiniband and creating embedded Linux clusters.

INVITED TALKS

- ***Lessons Learned from Howard Dean's Digital Campaign***

  Keri Carpenter, UC Irvine;
  Tom Limoncelli, Consultant

  Summarized by Andrew Echols

  Howard Dean's presidential campaign was unconventional in its approach to putting the campaign online. Traditionally, candidate activities take the form of getting out to the people, giving speeches, and kissing babies. Online campaigning takes this approach further, putting more content online and organizing supporters over the Web.

  Joe Trippi, the Dean campaign manager, a veteran of the dot-com boom, wanted to run an "open source" campaign. Traditional campaigns have been online before, but they have limited themselves to creating, essentially, a brochure Web site. The Dean campaign, however, created an online social movement with tools like blogs, meet-ups, and mailing lists. Furthermore, control of the message was opened up to Web site visitors. Supporters were trusted and ex-pected to aid in crafting the movement.

  The campaign blog provided up-to-the-minute articles and discussion of activities. It provided a central online community for the campaign. It also allowed visitors to post comments, opening up the dialog.

  The potential of the online campaign is apparent when one considers where it took the campaign. In January 2003, the campaign

had only 432 known supporters and $157,000 in the bank. By the end of the campaign, it had raised over $50 million from 300,000 individuals, with 640,000 supporters on a mailing list and 189,000 meet-ups. The online campaign's success story is that it took Howard Dean from obscurity to being, for a time, the front-running contender for the Democratic nomination.

- ***Storage Security: You're Fooling Yourself***

  W. Curtis Preston, Glasshouse Technologies

  Summarized by Josh Whitlock

  Until recently, storage people did not have to worry that much about security. Because storage was not meant to be accessible on a network, security was not designed into storage technology. Now that network security is becoming more of an issue, storage security must be reconsidered. One problem is that storage and security people speak different languages. Storage people talk in terms of zones, LUNs, and mirroring, while security people talk in terms of ACLs, VLANs, and DMZs. By learning how each other operate and working together, storage and security people will be better able to locate and address vulnerabilities.

  There are a number of security issues for storage. People can get inside the SAN and there have full access to the data by exploiting configuration mistakes such as open management interfaces and default passwords. Management interfaces are often plugged into the corporate management backbone, are visible from the LAN, use plaintext passwords, and often have default passwords. This problem can be solved by putting management interfaces on a separate LAN and upgrading to interfaces like SSL and SSH that don't use plaintext passwords. The backup person can be the greatest security risk of the SAN, because that person has root access. Depending on the backup person's security astuteness, a hacker could gain access to the backup server via social engineering and insecure server configuration.

  A hacker with access to the backup server has complete control over all the data. The greatest threat to tape backups is theft. Social engineering could be used to sway a low-income employee transporting the backups into stealing the tapes. Discarded media that is not sanitized properly before being sold could be scavenged for data. SAN issues with security deal with zones. For enforcement methods, hardware-enhanced zones are the only type of zoning that offers a meaningful level of access control. LUN masking hides LUNs from specific servers and should not be considered a security mechanism, but commonly is.

  Suggestions for improving storage security include shutting down whatever is non-essential, testing NAS servers for weaknesses, and putting the NAS on a separate LAN. Learning about weaknesses, planning for security, and working with vendors to make storage technologies more secure are other means of improving security.

IMPRESSIONS FROM A LISA NEWBIE

Chris Kacoroski, Comcast
kacoroski@comcast.net

Hi,

As a LISA '04 newbie with a focus on configuration management training, I attended tutorials every day and missed many of the technical sessions. What follows are my opinions, which you are obviously free to agree or disagree with.

Overview:

Having all areas covered by wireless really changes how the audience listens to the speakers. I would go to a session and while the speaker was setting up, I would go online and check out the servers at work, along with my favorite Web sites. I noticed several other people also doing this. If the speaker was covering something I was not interested in, I responded to emails. With the IRC channel, I saw some folks holding side chats while speakers were talking. This is a good thing for folks like me who do not have a backup in their job (one day I was able to fix a problem with our name servers), but from a speaker's viewpoint, I would think it could be very frustrating that some people are not paying atten-tion.

I was impressed by LISA's leadership and staff's emphasis on learning from each other. They encouraged this via the LISA bingo game, where you had to get signatures of people, the lunches provided with the tutorials, and the acknowledgment of the importance of "hallway track," where I definitely learned the most. I was also impressed by the folks who taught the tutorials; these are people who write books (e.g., O'Reilly's *LDAP* by Gerald Carter).

The days were very intense, with sessions from 9 to 5:30. Then there were BoFs from 7 to 11 p.m. After that the hallway track typically continued from 1 to 3 a.m. I averaged about five hours of sleep a night. I definitely had information overload.

The high points for me were:

- *System Logging and Analysis Tutorial*

  Marcus Ranum

  I liked the idea of "artificial ignorance," where if you see a log message that is okay, you filter it out but keep a count of how often it happens. In a short period of time, the only messages that get past the filters are ones that you have not seen before and that you need to look at. By keeping a count of messages, you can be alerted if a

message that normally happens 100 times a day jumps to 1000 (something probably happened). I also learned just how unreliable UDP really is and ways around this problem (using TCP or local server filtering).

Marcus briefly mentioned logwatch, plog, and logbayes, and went into detail on his NBS tool, which is used to implement the idea of Artificial Ignorance. Marcus is a good presenter who makes the material real through several real-life examples.

- *Cfengine Tutorials*

  Mark Burgess

  The introductory tutorial was not that interesting, but the advanced tutorial was excellent. I picked up enough new ideas and hints that I really need to go back and re-read the manual (some of this is due to a new version that has some significant changes, such as subroutines). Mark even created a patch to re-solve a specific problem I have been having with my use of cfengine.

- *Change Management Guru Session*

  Gene Kim

  The book provided with this session is very good, as it provides a recipe that my management can use to start to implement good change-management practices. Because this book is based on real data about the impact of unmanaged changes, I think I will have a good chance of getting it implemented.

- *Configuration Management Session*

  Alva Couch/Mark Burgess

  Mark's discussion on computer immunology did not grab me—too far out in the future, but Alva's discussion about the phases of configuration management and the difficulty in moving from one level to another was excellent, as he also listed the benefits of moving to the next level. I will be able to use this to show my management where I am trying to go. The phases are:

  Level 0—Ad hoc: Uses ad hoc tools to manage the systems. Pretty much anything goes. There are no barriers, since everybody does their own thing. Also, there is no documentation, reproducibility, or interchangeability of systems. This starts to fail when the number of systems goes above 50–100.

  Level 1—Incremental: Uses incremental management—implements a tool such as cfengine and just manages a few things on the machines. One barrier is that each sysadmin has to be trained on the tool you use. This level provides documentation of the systems and ease of updating 1000s of systems. This starts to fail when you need to manage entire systems, not just parts of them.

  Level 2—Proscriptive: Controls the entire system by building from bare metal for each system. A barrier is loss of memory of how a machine was set up (e.g., why a machine was configured in a certain manner). Lots of work to document exactly how the system was set up before it was rebuilt. Need to do a lot of testing to ensure that the bare-metal rebuild will work. This provides guaranteed reproducibility, where you can create any system from bare metal.

  Level 3—Enterprise: Divorces the hardware from the machine use so machines are interchangeable—we can move services between machines as needed. All machines are created alike. A barrier is the loss of freedom to take care of your own machine (e.g., you do not know which machine will be the Web server). The benefit is that any machine is interchangeable with another. Kind of like thin clients, where a person can log on to any terminal and get their own environment. Companies like CNN use this for their Web server farms.

  Alva then went on to show how each organization will need to determine which level has the best payoff. For example, his group has decided to stay at level 2, because they rebuild machines every six months and it makes no sense to go to level 3. Paul Anderson's group has moved to level 3, because they do not rebuild machines every six months, so the cost-benefit equation makes sense.

- *BoFs & Hallway Track*

  The asset management BoF had the idea of tracking MAC address to IP address to switch port in order to alert you to any changes in the data center. I really liked what one participant had to say: he used SNMP to get all kinds of data from his servers—the only thing he was lacking was an automated process to determine what power plug a computer was using.

  As I said at the beginning, I focused on configuration management. My environment consists of about 100 servers (Mac, Linux, Solaris, Windows), 1500 need-to-be-managed Mac workstations, and 700 Windows workstations. What I learned from the BoFs, hallway track, and sessions was:

- Cfengine is a good choice for me to manage my systems.
- I need to move to automated builds of systems.
- Cfengine has a lot of functionality that I was not aware of. I had been trying to figure out hacks for certain items that cfengine does out of the box.
- Many folks who use cfengine do not really use it in depth but only use a small portion of it. Mark Burgess was the only person I found who really understood all of it.
- I have about a two- to three-year project ahead of me to get everything managed.
- A monitoring system that uses SNMP to tie MAC address to IP address to switch port has lots of benefits for asset management and

sending alerts when things change.

- Change management is critical. Tracking the success rate of approved changes really makes it much easier to diagnose and resolve problems, since most problems result from recent changes.

cheers,

ski