A Retrospective on Twelve Years of LISA Proceedings

Eric Anderson – University of California, Berkeley
Dave Patterson – University of California, Berkeley

ABSTRACT

We examine two models for categorizing tasks performed by system administrators. The first model is the traditional task based model. The second model breaks tasks down by the source of the problem. We then look at the historical trends of the last 12 years of LISA proceedings based on these models. Finally, we analyze some of the more important tasks done by system administrators and propose future research in those areas. Our hope is that some of the academic rigor in analyzing research can be brought to systems administration without losing the practicality that makes the research valuable.

Introduction

System administrators don't have a lot of time for introspection of their field. So work is repeated and new administrators, or people trying to do research on system administration, don't know where to start. To provide a starting point, we have examined the last twelve years of LISA proceedings and have categorized the papers in two separate ways. One categorization is by problem causes, and has the advantage that it will apply to any task in system administration. The second categorization is the traditional task breakdown, which shows us where past research has been focused.

In addition to categorizing the papers, we examine the trends in the categorization over the last twelve years. We find that some tasks were solved, and then, due to external changes, needed more work. We also find that some tasks have had a remarkable amount of effort focused on them without a complete solution. We then examine in more detail the more popular areas of research both to gain historical understanding and to consider future directions.

So that others can more easily build on our work, we make the complete set of data including both categorizations, brief summaries, and bibliographic information available on the web from http://now.cs.berkeley.edu/Sysadmin/categorization/.

The next two sections examine the two models, and then the fourth section shows historical trends. The fifth section focuses in on a number of important tasks and examines each in detail. The final section provides a few brief conclusions.

A Model of Tasks

The traditional approach for categorization is to group related papers by the task each targets. We did this for all of the papers. We continued the aggregation process starting with the list of tasks having at least two papers and built a hierarchy of tasks as shown below. The categories are sorted by popularity; the paper count is shown in brackets; ties are broken alphabetically. There were a total of 342 papers, and 64 separate categories.

- Services [75]
  - Backup [28]
  - Mail [20]
  - Printing [11]
  - News [5]
  - NFS [4]
  - Web [3]
  - DNS [2]
  - Database [2]
- Software Installation [57]
  - Application Installation [32]
  - OS Installation [14]
  - User Customization [8]
  - Software Packaging [3]
- Monitoring [44]
  - System Monitoring [14]
  - Resource Accounting [6]
  - Data Display [5]
  - Network Monitoring [5]
  - Benchmarking [4]
  - Configuration Discovery [4]
  - Host Monitoring [4]
  - Performance Tuning [2]
- Management [40]
  - Site Configuration [27]
  - Host Configuration [7]
  - Site Move [4]
  - Fault Tolerance [2]
- Miscellaneous [40]
  - Trouble Tickets [9]
  - Secure Root Access [8]
  - General Tool [6]
  - Security [6]
  - File Synchronization [4]
  - Remote Access [3]
  - File Migration [2]
  - Resource Cleanup [2]

- Management [35]
  - Accounts [23]
  - Documentation [4]
  - Policy [3]
  - User Interaction [3]
  - White Pages [2]
- Networking [19]
  - Network Configuration [9]
  - LAN [4]
  - WAN [4]
  - Host Tables [2]
- Improvement [18]
  - Self Improvement [7]
  - Models [5]
  - Software Design [4]
  - Training Administrators [2]
- one paper on topic [19]

We can see that there are many different types of tasks, but a few subjects are very popular: Backup, Mail, Application Installation, Site Configuration, and Accounts. We can also see that there is a remarkable amount of variability among the various tasks, not including the single-paper topics.

The taxonomy is useful because it helps to describe a skill set necessary for system administrators. We can see which areas system administrators have focused most of their efforts on, examine which areas have been successfully solved, and identify areas needing more work. Since this taxonomy is derived from papers, for completeness, it should be combined with tasks from time surveys [Ande95, Kols92] and interviews.

There are some potential concerns about this categorization. The simplest of which is that there were errors in the classification. There were about 350 papers, so a few errors probably occurred in classification. Furthermore, while the first author worked as a system administrator both at SURAnet, and at Carnegie Mellon University, he has clearly not personally performed all of the tasks described. The program committee may also have affected the papers accepted based on their views of what should be in the conference, or because of a limited selection of available papers. Finally, some papers may be missing because companies consider the information to be proprietary. We believe that the classification is useful, but keeping the weaknesses in mind will help prevent us from drawing incorrect conclusions.

A Model of Problem Sources

A second model based on the source of a problem is shown in Figure 1. The source of the problem is labeled on the edges leading out from the center (the happy state) of the state transition diagram. The edges leading back in to the center represent tasks performed to return the system to a happy state. The model was derived in part from the time surveys, which indicated that administrators spent about a third of their time on each of these tasks.

This model is fairly general and hence is able to cover all types of things done by administrators. Either administrators are trying to improve people (training) or trying to improve machines. If they're trying to improve the machines, it's either because the machines need to do something different (configuration management) or because they need to get back to doing what they used to do (maintenance).

Examination of the Different Categories

Configuration management tasks will remain so long as people change how they want to use the system. Only by freezing how the system is used can we eliminate configuration management tasks. Even a simple appliance like a toaster has a few configuration tasks (plugging it in, adjusting the amount of toasting). The tasks have been simplified by limiting choices; adding choices inherently increases complexity.

Maintenance tasks may be eliminated by building systems that recover from internal faults. If a maintenance task can't be eliminated (for example, purchasing replacement hardware), the goal should be

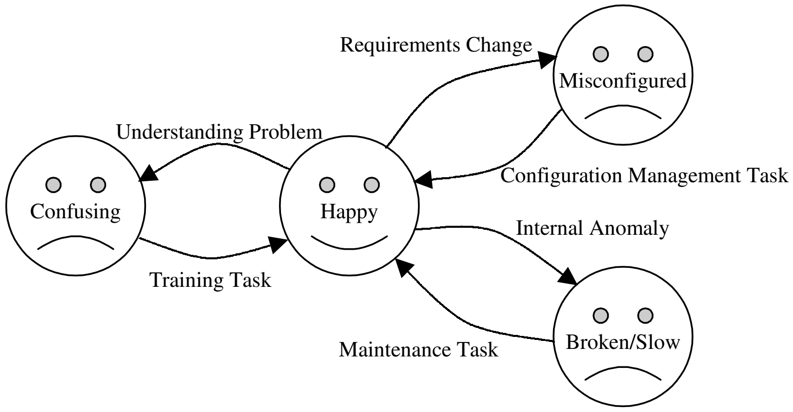

Figure 1: System state transition diagram. Edges out indicate problems that occur making the system less usable. Edges in indicate tasks performed by system administrators to restore the functionality of the system.

to make the task schedulable, rather than forcing an administrator to deal with the problem immediately. Reducing the number of interrupt-style tasks should lead to improving system administrator effectiveness.

Training tasks may be transferable out of the organization and into the schools. Users could be trained in the tools they will be using, and administrators could be trained in system administration. Earlier education would mean people would only have to learn the specifics of a site rather than the general knowledge. Alternately, the various tools that are being used could be improved to reduce the need for training. Researchers in Human Computer Interaction have been looking at this for some time, and have made a number of strides, but more work remains.

Historical Trends of the Conference

Given the two models, we can look at how the papers in the conference have fit into the models over the course of time. This will help us see if things have been changing from previous conferences.

Task Model Trends

Figures 2 and 3 show the papers over the last twelve years categorized by the Task Model. For completeness, we show all of the papers that were shown in the hierarchical categorization.

We can see that some tasks, such as backup, application installation and accounts alternated between very heavy and light years. This probably indicates some amount of duplicated effort in the very heavy years. In some cases (application installation, OS installation), this pattern indicates that good solutions have not been found, and people are still making new, slightly different attempts. In other cases (backup, accounts), it indicates that there was some change in the external world that caused previous solutions to stop working. For example, backup was a task that was successfully solved in the past, but with disk capacity and bandwidth growing faster than tape capacity and bandwidth, it has returned as a problem of dealing with larger scale.

We can see that some tasks, such as printing and trouble tickets, have received a little bit of work fairly steadily. This pattern is probably a good sign, as it means that slow and steady progress is being made without too much duplication of effort.

Mail alternated between the steady work and the heavy work models. Initial work was fairly steady until the explosion of the Internet increased the size of mailing lists, and commercialization resulted in problems with SPAM.

Similarly, some tasks, such as system monitoring and network configuration, see punctuated bursts of activity. This pattern probably indicates that the problem occurred simultaneously due to some external change such as sites scaling up, or new applications. It would be nice if there were some way for different people to coordinate their work as they simultaneously discover new problem areas. This would reduce the amount of duplicated work, and probably also

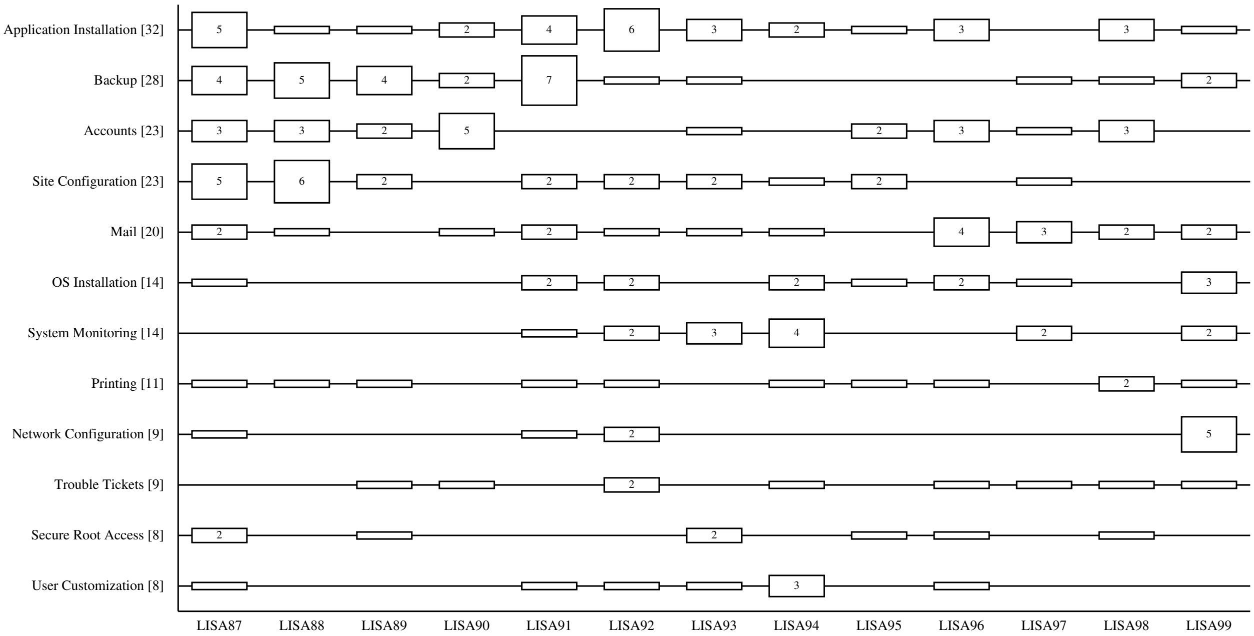

Figure 2: Breakdown of number of papers/conference/category for categories with at least eight total papers. Sorted by popularity of a category, ties broken alphabetically. Height of a box (and the number inside) indicates number of papers. Total number of papers in a category is shown in brackets after the category name. Remainder of categories are shown in Figure 3.

improve the resulting solution as it will deal with the idiosyncrasies of multiple sites.

It is not clear what we can learn from the tasks with fewer papers. In a few cases, we can infer that certain areas did not become problems until fairly recently. Web is an obvious example; configuration discovery, LAN, WAN, and NFS problems also appear to have only become problems recently.

Source Model Trends

Figure 4 shows the papers over the last twelve years categorized by the Source Model.

We can see that the number of training task papers has been remarkably small, and in fact, further examination of the papers in those categories indicates that they are mostly papers on improving the skills of administrators. The one oddity is LISA93, in which a third of the papers were on many different training issues. Some of the training papers cover software design issues for administrators, others suggest how to improve interactions with other administrators, users or managers. A few of the training papers cover how to train new administrators, but surprisingly none of the papers cover training users to take better advantage of software or provide better problem summaries. Training is an area where some work should be done, although it is more difficult to analyze because it involves the variability of people.

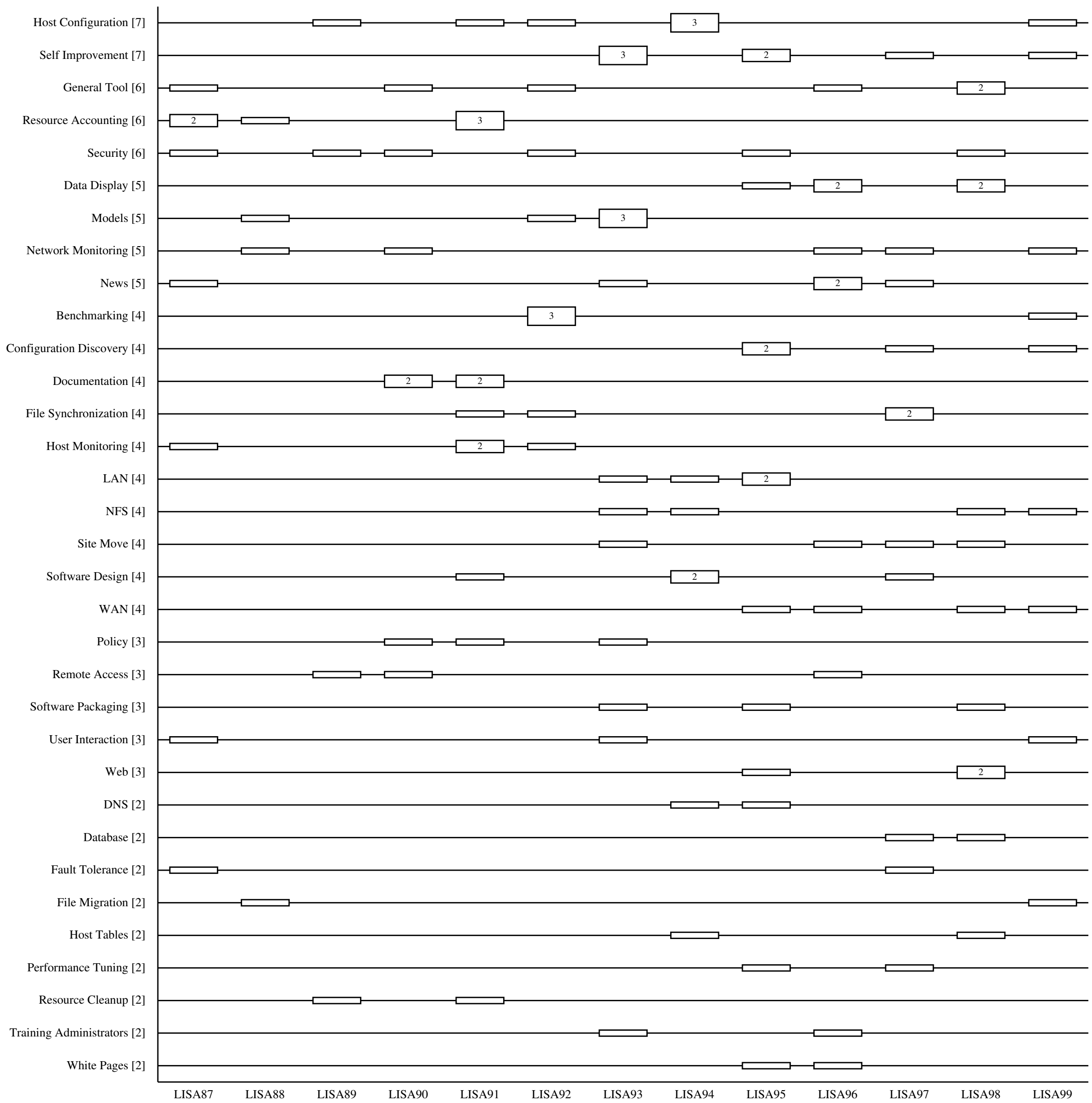

Figure 3: Continuation of Figure 2 for categories with 2-7 papers overall. Sorted by popularity of a category, ties broken alphabetically. This figure is included for completeness, care should be taken in drawing conclusions given the small number of papers.

We can also see that maintenance tasks comprise the second largest fraction of papers. Unfortunately, interrupt-style maintenance tasks contribute greatly to administrator stress. Beyond simply eliminating maintenance tasks by having systems automatically repair themselves, we should strive to convert maintenance tasks to schedulable tasks. If systems were designed to operate in degraded mode, then administrators would not have to respond immediately every time a problem occurred, but could instead work on related tasks at the same time.

Finally, we can see that configuration management tasks are the most prevalent of the papers, which is reasonable and unsurprising given that many tasks eventually require some change in configuration. Configuration tasks generally lead to results which can be more easily described in a paper than results from the other two categories.

Examination of Important Tasks

We now examine the important tasks performed by system administrators in more detail. We summarize the area, examine the research history, and propose directions for future research. Many of the directions would make good papers for future LISA conferences. In the research history, we reference some of the better papers on each topic, so that readers will know where to look for additional information.

Software Installation: OS, Application, Packaging and Customization

Software installation covers the problems of managing software installed on computers. There are four sub-categories of software installation: Operating System (OS) Installation, Application Installation, Software Packaging, and User Customization. Operating system installation deals with the problem of taking the raw machine and putting the operating system on it so it can boot. Application installation is the addition of optional (non-OS) packages to a machine. Software packaging is the step of creating an installable package. User customization happens when users need to change the way the software operates.

Research History

OS installation usually puts files in specific places and has limited support for multiple versions on a single machine. Research into operating system installation has taken a cyclic path. In the very beginning, the OS was installed by either cloning a disk and then putting it in the new machine, or by booting the new machine off some other media (e.g., floppy disk, network) and then copying an image to the local hard drive. Those solutions were then modified to support customization of the resulting installation and easier upgrades [Zwic92, Hide94]. The tools were then scaled to allow fast installation across the entire enterprise [Shad95]. By then large-scale PC OS installation needed to be supported, and the cloning solution [Troc96] reappeared.

Application installation usually puts packages into separate directories, and uses symbolic links to build composite directories, so multiple versions are easily supported, and programs can be beta tested easily before being made generally available. Application

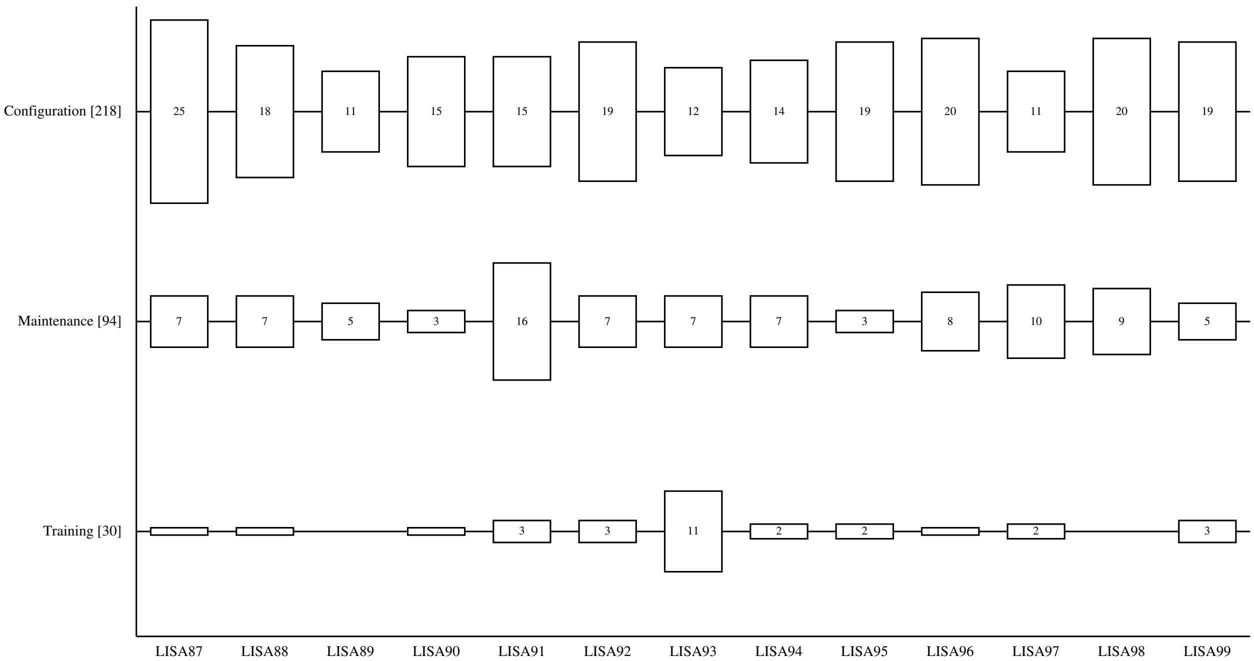

Figure 4: Breakdown of number of papers/conference/category. Sorted by popularity. Height of a box (and the number inside) indicates number of papers. Total number of papers in a category is shown in brackets after the category name.

installation has had many more papers written on it than OS installation, probably because vendors didn't supply tools to install additional applications. The initial solution was to build packages in separate directories and link them into a common directory [Manh90, Coly92]. These tools were then extended to support customization per host [Wong93]. Recently, the caching and linking pieces were untangled and refined into separate tools [Couc96b, Bell96].

Relatively few papers have been written on software packaging, probably because most of the application installation tools use source code trees rather than binary packages. These papers cover the patching of software for different host types, and the subsequent generation of installation packages [Stae98].

The papers on user customization cover two separate areas of customization: Selecting which packages are accessed by a user [Furl96, Will93], and customizing application behavior [Elli92]. The package selection tools started as simple shell scripts that adjusted environment variables to enable packages, and later were refined to work faster and more flexibly. The customization tools have dealt with different aspects of making it easer to control the behavior of programs and have been targeted at beginning users.

An Alternative Breakdown

There have been a remarkable number of papers in this area, many of which seem like slight variations of each other, which makes us wonder if the problem has been broken down poorly. We therefore propose a different breakdown into the following five pieces: Packaging, Selection, Merging, Caching, and End-User Customization.

The distinction between installing applications and the operating systems is probably unnecessary and a historical artifact. Some of the OS installation papers supported some limited number of additional packages, and recent OS installation programs [Hohn99] can install most of the packages available on the net. However, the distinction in functionality that was found in the OS/Application split still remains.

Packaging

Software packaging appears to be a mostly solved problem. There have been a few papers in the LISA conference on it, and the freely available Unix systems have associated packaging tools. Comparing these tools might pave the way to a single multi-platform tool.

Packaging usually binds pathnames into an application. This can limit how packages can be merged later (e.g., two versions both believe they own /usr/lib/package). Some packages allow environment variables to override pathname choices. Exploring the performance and flexibility of the different choices could help improve existing tools.

Selection

Package selection is part of all OS/Application installation tools. The key pieces for a selection tool are the need for per-machine flexibility and the need to support multiple collections. Both programmatic and GUI interfaces should be supported so that the tool is both easier to use and scriptable. The selection tool could then be integrated into some of the existing tools as a uniform front-end.

Merging

Merging packages remains a hard outstanding problem. Many tools just ignore the problem. A few have a configuration file to specify which package overrides another when conflicts occur. Merging is most difficult when packages are inter-related, as is the case with Emacs, Perl and Tcl with their various separate extensions; Tex/LaTeX; X windows with various applications that add fonts and include files; and shared library packages.

One unsatisfying solution is to pre-merge packages during packaging so that there are no inter-relations between packages. A modular solution would need to handle merging of files, for example generating the top level Emacs info file, or the X windows font directory files. Some programs include search paths, which might make the merging easier to handle, others require the execution of a program in the final merged directory.

If multiple versions need to be supported simultaneously, there is a more substantial problem. Supporting the cross product of all possibilities is not practical. However, there is no clear easy solution. Quite a bit of thought will be needed to find an adequate solution.

Caching

Caching to the local disk is beneficial for both performance and for isolating clients from server failures. Caching is a semi-solved problem. Some file systems cache onto local disk to improve performance (e.g., AFS, CacheFS, Coda). In general, caching merely requires mounting the global repository somewhere different and creating symlinks or copies as appropriate. There have been tools written to do just this [Couc96b, Bell96], and many of the general software installation tools have included support for caching [Wong93]. Making the caching fully automatic and fine grained will probably require some amount of OS integration.

End User Customization

End user customization has only been slightly examined. A few tools help users dynamically select the packages they want to use [Furl96]; most have fixed the choice on a per-machine basis. One old paper looked at how users customized their environment [Will93]. It would be nice for this area to be resurrected for research. Programs are becoming increasingly complex, especially as they add GUI interfaces, but the ease of customizing the programs has not kept up. Work in this area would require a large amount of interviewing users to determine what they would like to customize.

Backup

Backup addresses four separate, but related problems: User Error, Independent Media Failure, Correlated Media Failure (e.g., Site Failure, Software Error), and Long Term Storage. All the solutions are based on some type of redundant copy, but the particulars of each are different. Damage due to user error can be reduced by online filesystem snapshots. Independent media failure can be remedied by techniques like RAID. Correlated media failure requires use of additional uncorrelated media (e.g., Off-site tape, remote duplicates with different software). Finally long term storage requires very stable media, and an easily read format. Consider how many people can still read data written on punchcards, or even 9-track tape. Most of the focus in backup has been on independent media failure, usually by creating copies on tape, although people have looked at the other issues.

Research History

Research on backup has passed through many stages. The first was correctness: Does the right data get written? [Zwic91b] Are backups happening regularly and on schedule? [Metz92] Do restores work? Having achieved correctness, research turned to scaling backup solutions to the enterprise. The solution was staging disks so that backups could stream to tape [Silv93]. Having solved the correctness and scalability problems, research on backup paused. But then the onward march of technology reintroduced scalability as a problem. Disk bandwidth and capacity are starting to outstrip tape bandwidth and capacity leading to solutions requiring multiplexing of disks and tapes [Pres98].

Future Directions

Restores seem to be a somewhat overlooked part of the backup problem. Most backup papers deal in great detail with formats of dump tapes, scheduling of backups, streaming to tape. However, they usually only write a few paragraphs on the subject of restores, often ignoring the time taken to restore data. The whole purpose of backup is so that when something goes wrong, restores can happen! We would like a discussion of restore difficulty and measurements of restore performance in future papers. When something fails, there is a cost in lost productivity in addition to the direct cost of performing the repair.

Examining technology trends and technology options would help identify future backup challenges before they occur. The technology involved has reasonably predictable future performance in terms of bandwidth, latency, and capacity. Somewhat weaker predictions can be made about the growth in the storage needs of users. Given this information, a prediction can be made about the required ratio of hardware in the future. In addition, alternatives to tape backup such as high capacity disks and writable cds/dvds may become viable in the future. One advantage of random access media is that data can be directly accessed off the backup media to speed up recovery.

Backup by copying to remote sites is very different from traditional approaches. A few companies are dealing with the possibility of a site failure by performing on-line mirroring to a remote site over a fiber connection. It may be possible to decrease the required bandwidth by lowering the frequency of the updates, so that this approach is practical for people unable to purchase a dedicated fiber.

Backups also present special security concerns. A backup is typically an unprotected copy of data. If anyone can get access to backup tapes, they can read critical data. How can encryption be used to solve the security problem? Will encryption enable safe web backup systems?

Another interesting question is how to handle backup for long-term storage. Some industries have legal requirements to retain documents for a long (indefinite) time. There are two related problems. First, media needs to be found which is stable enough to last a long time. Second, it seems wise to rely on conversion to a common format because it is never clear what software will still work in 20-50 years. How can these two concerns be integrated into a backup solution?

Configuration: Site, Host, Network, Site Move

Configuration tasks are modification to the setup of hardware and software so that the environment matches the requirements of a particular organization. These tasks can range from simply installing the appropriate exports and resolv.conf files to complicated tasks like migration from an MVS platform to a UNIX one.

Research History

The first few LISA conferences included many papers which summarized their site's configuration. Research then forked in two directions. Some papers looked at how to store and extract configuration information from a central repository, either using available tools such as SQL [Fink89], or by designing their own language [Roui94a]. Other papers looked at using a level of indirection to make configuration changes transparent to users [Detk91].

The great growth spurt in the computer industry lead to complete site moves, either as part of a merger, separation, or just to handle growth [Schi93]. Similarly, the great amount of research in this area led some people to examine the question, “What properties of site design make it easier to administer?” [Trau98]. Recently, a mobile user base caused dynamic network re-configuration to become a problem [Vali99].

Categorization Commentary

This is probably the weakest categorization. The original intent was that host configuration would cover host issues, network configuration would cover network issues, and site configuration would cover global site issues. However, the line between host and site is at best blurry. We therefore believe that someone should re-examine the papers in these areas, and see if they can find a better categorization.

Future Directions

The key to host configuration seems to be having a central repository of information that is then pushed or pulled by hosts. Most of the papers did some variant of this. Two areas remain to be refined: First, someone should analyze exactly what information should be in the central repository, and how it can be converted to the many different types of hosts in use. Second, someone should write a tool to automatically create the repositories so that the start-up cost to using a configuration tool is lower.

Site configuration tools vary widely, probably because of the different requirements at each site (e.g., a wall street trading firm vs. a research lab). [Evar97] surveyed the current practices, and [Trau98] studied the best practices for certain environments. Combining these two directions by identifying the best practices based on the requirements of a site would help all sites do a better job of configuration.

Network configuration is a fairly recent topic, so proposing directions by analyzing the papers is risky. However, we can still look at analogies to previous work. First, we want to build abstract descriptions of the system. Second, the models should be customizable; early configuration tools didn't support much customization, so later ones had to add it. Third, a survey paper, analogous to [Evar97] would help identify the problems in network configuration research.

Accounts

Managing user accounts at first seems very simple. But further examination indicates that there are additional subtleties because an account identifies users, and therefore has lots of associated real world meaning. Therefore, authentication, rapid account creation, and managing the associated user information become important.

Research History

Accounts research started with the goal of simplifying the account creation process. Scripts were designed that automated the steps of accumulating the appropriate information about users, adding entries to password files, creating user directories, and copying user files [Curr90]. Because the scripts were site-specific, they were able to do better error checking. Once creating accounts became easy, accounts research paused until enough people needed accounts that scalability became a concern. Sites with thousands of accounts, usually schools, needed to create lots of accounts quickly because of high turnover in the user population. Their solutions tended to have some sort of central repository storing account information (often an admissions' database), with complementary daemons on client nodes to extract the needed parts of the database [Spen96]. Some of the recent papers considered auxiliary details such as limiting accounts to certain hosts, account expiration, and delegating authority to create accounts [Arno98].

Future Research Opportunities

Surveying account creation practices would help identify why no tool has evolved as superior despite many papers on this subject. We believe this is because of unrecognized differences in the requirements at each site. With all the requirements explicitly described, it should be possible to build a universal tool.

A related topic is the examination of specific issues related to account creation. For example, many of the papers ignored the question of how to limit accounts to specific machines. Is a simple grouping as was done for host configuration sufficient, or is some sort of export/import setup needed? Sharing accounts across administrative boundaries within an organization will make this problem even more difficult.

Another specific issue is delegation of account creation. The one tool to do this [Arno98] assumed all the employees were trusted to enter correct account information. Clearly this solution will not work at all sites. There may be synergy with the secure root access papers that looked at delegation.

Mail

Electronic mail has been one of the driving applications on the Internet since its inception. This makes it unsurprising that it ranks extremely high on the list of applications. It is the highest of the applications that are used by end-users on a regular basis. There is a vast amount of email, traveling around the world-wide network, leading to a lot of effort in interoperability and scalability.

Research History

Very early research in mail targeted interoperability between the wide variety of independently developed mail systems. This research and the reduction in variety over time, combined with SMTP as a standard mail interchange protocol, solved the interoperability problem. Research then turned to flexible delivery and automating mailing lists [Chap92]. There was then a brief pause in the research. However, as the Internet continued to grow, research on scaling delivery of mail both locally and in mailing lists [Kols97] was needed. At the same time, commercialization caused SPAM to become a problem [Hark97].

Future Research

The biggest remaining problem is dealing with SPAM. The correct solution is probably dependent on trading off difficulty in being reached legitimately with protection from SPAM. Some possible approaches are: acceptance lists with passwords, a list of abusers that are automatically ignored (this is being done), a pattern matcher for common SPAM forms, and receive-only/send-only addresses. Finding a good solution will be challenging.

Scalability and security still need some work. Scalability of mail transport and mail delivery may be possible by gluing together current tools into a clustered solution. Both problems partition easily. Handling more types of security threats also remains open; [Bent99] has done some initial work securing MTA ⇔ MTA transfers.

Monitoring: System, Network, Host, Data Display

Monitoring solutions help administrators figure out what is happening in the environment. There are problems of system, network and host monitoring, and the associated problem of data display. Monitoring solutions tend to have two variants: instantaneous and long term.

Research History

Research in monitoring has progressed along a number of axes. First, there has been work in monitoring specific sources from file and directory state [Rich91] to OC3 links [Apis96]. Simultaneously, generic monitoring infrastructure [Hard92, Ande97a] has been developed. Finally, as the amount of data available has increased, some work on data display has been done [Oeti98a].

Categorization Commentary

The categorization here was by the type of thing being monitored (host, network system). Perhaps a better classification would be by the axes described in the research history.

Future Directions

There has been a lot of work on gathering data from specific sources, but in most cases, the overhead for gathering data has been high, so the interval is usually set in minutes. Reducing this overhead is important for allowing finer grain monitoring [Ande97b]. In addition, we would like to vary the gathering interval so that the overhead of fine-grain gathering is only incurred when the data would be used. In addition to just gathering the data, having a standard form for storing the data efficiently would be very useful. Combining these two issues should lead to a nice universal tool with pluggable gathering modules.

Data analysis and data reduction have not received nearly the attention they deserve. The data collection techniques are only useful if the data can be used to identify problems. But beyond averaging time-series data, very little automated analysis has been done. An examination of methods for automated analysis, for example, looking at machine learning techniques, could prove fruitful.

Data visualization has started to get some examination in the system administration field. There is a vast amount of literature on various forms of visualization in the scientific computing field. We believe that a survey of existing techniques would lead to tools that allow visualization in system administration to be both more effective and more scalable.

Printing

Printing covers the problems of getting print jobs from users to printers, allowing users to select printers, and getting errors and acknowledgements from printers to users.

Research History

Early research in printing merged together the various printing systems that had evolved [Flet92b]. Once the printing systems were interoperable, printing research turned to improving the resulting systems, making them easier to debug, configure, and extend [Powe95]. As sites continued to grow, scaling the printing system became a concern, and recent papers have looked into what happens when there are thousands of printers [Wood98].

Future Directions

Printing research seems to be in fairly good shape. Scaling print systems is still not completely done, debugging problems and selecting the right printer is still challenging. Perhaps printer selection could be done by property (e.g., color, two sided). Finally, the path for getting information from printers back to users has not been well examined. A notification tool to tell users the printer's status, such as print job finished or out of paper, would be useful. The notification tool might also help in debugging printing problems.

Trouble Tickets

Trouble ticket tools simplify the job of accepting a problem report, assigning the problem report to an administrator, fixing the problem, and closing the problem's ticket. Trouble ticket systems usually have a few methods for getting requests into the system (e-mail, phone, GUI), and provide tools for querying and adjusting the requests once they are in the system.

Research History

Trouble ticket systems began as email-only submission tools with a centralized queue for requests [Galy90]. Later, the systems were extended so that users could query the status, and tickets could be assigned to particular administrators [Kobl92]. The systems were improved to support multiple submission methods such as phone [Scot97] and GUI, and to support multiple request queues [Ruef96].

Future Directions

There seems to be a fair amount of overlap in the research on trouble tickets. Many of the tools were created from scratch, only occasionally building on the previous research. Examining the existing tools should identify the different requirements that have led to all these systems and to a more general tool.

A second direction to extend trouble ticket systems would be to build in a knowledge of the request handling process. [Limo99] examines the process of handling problem reports, but doesn't propose tools. A trouble ticket system supporting the process would be quite valuable.

Secure Root Access

Secure root access is the general problem of providing temporary privileges to a partially trusted user. Many actions need to be taken as root, and giving out the root password is clearly a poor decision. The questions then are how to give out privileges, how to track their use, and how to retain some amount of security.

Research History

Research in secure root access has gone down two separate paths. One path has been to examine how to provide secure access to commands within a host. This has gone through many iterations, slowly adding in more complex checking of programs and arguments [Mill99, Hill96]. The other has been to provide secure access remotely [Ramm95].

Future Directions

The unfortunate effect of having the two separate paths of research is that neither handles all the problems easily. The remote tools are more flexible, but harder to configure, and don't support logging well. The local tools have a more natural interface, but don't have as much power to provide partial access. Combining these two paths of research should lead to a more powerful and flexible tool.

A second direction to consider is toward providing finer-grain access control. [Gold96] did this by securely intercepting system calls. Further work could lead to having something like capabilities in the OS, allowing very precise control over the access granted to partially-privileged users.

Conclusions and Analysis

We have categorized all of the papers in the LISA conference according to two separate models. We have made the categorization available so that others can examine our choices, correct mistakes, or provide better categorizations. Hopefully this paper will encourage people to think differently about the field and problems that it presents, and as a result build better tools and processes.

We would like to see other people examine some of the other conferences that may publish relevant papers. The USENIX general conference, SIGCOMM, and SANS are a few places to start looking. There is likely to be some useful information present in those conferences which was not covered in this paper.

We have examined the historical trends of the LISA conference according to the two models. This has helped us see that some areas are under served, and some are probably over-served. We can also see the bursty nature of research in system administration (probably because the same problem occurs to everyone at the same time). As a result we recommend that a central clearinghouse of problems be created to facilitate collaboration and improve the resulting tools.

Finally we examined some of the important task areas. Based on our analysis, we have proposed a number of papers to be written. We believe that this sort of analysis should be performed every few years. The Database community gets together and decides which areas of research were successful, and which require more work [Silb91, Silb96]. Their reports have helped their community show their results and focus their efforts. Hopefully this analysis of system administration will help do the same for ours.

Acknowledgements

We would like to thank Evi Nemeth, Kim Keeton, Drew Roselli, Aaron Brown, and David Oppenheimer, and the anonymous reviewers for their comments on the paper. Their comments have improved both the ideas and the readability of the paper immensely. This work was supported by DARPA under grant DABT63-96-C-0056.

References

The entire database of categorized papers is available from http://now.cs.berkeley.edu/Sysadmin/categorization/ .

[Ande95] Eric Anderson. "Results of the 1995 SANS Survey" *;login:*, October 1995, Vol20, No. 5, http://now.cs.berkeley.edu/Sysadmin/SANS95-Survey/index.html; Found weak correlation between number of machines and number of admins; many important tasks.

[Ande97a] Eric Anderson and Dave Patterson, "Extensible, Scalable Monitoring for Clusters of Computers," *Proceedings of the Eleventh Systems Administration Conference* (LISA '97), San Diego, California pp 9-16; http://now.cs.berkeley.edu/Sysadmin/esm/intro.html; Tool for monitoring and displaying cluster statistics.

[Ande97b] Jennifer M. Anderson, Lance M. Berc, Jeffrey Dean, Sanjay Ghemawat, Monika R. Henzinger, Shun-Tak A. Leung, Richard L. Sites, Mark T. Vandervoorde, Carl A. Waldspurger, and William E. Weihl. "Continuous Profiling: Where Have All the Cycles Gone?" *16th ACM Symposium on Operating Systems Principles,* Saint-Malo, France, http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1997-016.html .

[Apis96] Joel Apisdorf, k claffy, Kevin Thompson, and Rick Wilder, "OC3MON: Flexible, Affordable, High Performance Statistics Collection," *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 97-112, http://www.nlanr.net/NA/Oc3mon/; HW & SW for monitoring and analyzing traffic on an OC3 link.

[Arno98] Bob Arnold, "Accountworks: Users Create Accounts on SQL, Notes, NT, and UNIX," *Proceedings of the Twelfth Systems Administration Conference* (LISA '98), Boston, Massachusetts, pp 49-61.

[Bell96] John D. Bell. ''A Simple Caching File System for Application Serving,'' *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 171-179; Automatic caching of applications from a remote server to local disk.

[Bent99] Damien Bentley, Greg Rose, and Tara Whalen. ''ssmail: Opportunistic Encryption in sendmail,'' *Proceedings of the Thirteenth Systems Administration Conference* (LISA '99), Seattle, Washington.

[Chap92] D. Brent Chapman. ''Majordomo: How I Manage 17 Mailing Lists Without Answering '-request' Mail,'' *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 135-143, ftp://ftp.greatcircle.com/pub/majordomo.tar.Z .

[Coly92] Wallace Colyer and Walter Wong, ''Depot: A Tool for Managing Software Environments,'' *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 153-162, ftp://export.acs.cmu.edu/pub/depot/; Build merged tree by copy/link from packages; conflict resolution by package preferences.

[Couc96b] Alva L. Couch, ''SLINK: Simple, Effective Filesystem Maintenance Abstractions for Community-Based Administration,'' *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 205-212, ftp://ftp.cs.tufts.edu/pub/slink; Flexible sym-linking/copying for merging software repositories.

[Curr90] David A. Curry, Samuel D. Kimery, Kent C. De La Croix, and Jeffrey R. Schwab, ''ACMAINT: An Account Creation and Maintenance System for Distributed UNIX Systems,'' *Proceedings of the Fourth Large Installation Systems Administrator's Conference* (LISA '90), Colorado, pp 1-9.

[Detk91] John F. Detke. ''Host Aliases and Symbolic Links -or- How to Hide the Servers' Real Name,'' *Proceedings of the Fifth Large Installation Systems Administration Conference,* (LISA '91), San Diego, pp 249-252; Use host aliases & symbolic links to allow mount points & servers of exported FS to move w/o needing client changes.

[Elli92] Richard Elling and Matthew Long. ''user-setup: A System for Custom Configuration of User Environments, or Helping Users Help Themselves,'' *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 215-223, ftp://ftp.eng.auburn.edu/; Extension to Modules [Furl91] system, menu driven script to select applications & configure apps.

[Evar97] Rémy Evard. ''An Analysis of UNIX System Configuration,'' *Proceedings of the Eleventh Systems Administration Conference* (LISA '97), San Diego, California, pp 179-193; Examination of current configuration practices at nine different sites.

[Fink89] Raphael Finkel and Brian Sturgill. ''Tools for System Administration in a Heterogeneous Environment,'' *Proceedings of the Workshop on Large Installation Systems Administration III* (LISA '89), Austin, Texas, pp 15-29; Relational structure stores host, file information. Tables can be generated at runtime. Schema describes relation structure & constraints. Query language queries & executes.

[Flet92b] Mark Fletcher, ''nlp: A Network Printing Tool,'' *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 245-256; Centralized print server database, uses lpd protocol to transfer files.

[Furl96] John L. Furlani and Peter W. Osel, ''Abstract Yourself With Modules,'' *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 193-203, http://www.modules.org/; Per-user flexible configuration of accessible packages.

[Galy90] Tinsley Galyean, Trent Hein, and Evi Nemeth, ''Trouble-MH: A Work-Queue Management Package for a >3 Ring Circus,'' *Proceedings of the Fourth Large Installation Systems Administrator's Conference* (LISA '90), Colorado, pp 93-95.

[Gold96] Ian Goldberg, David Wagner, Randi Thomas, and Eric A. Brewer, ''A secure environment for untrusted helper applications: confining the wily hacker,'' *Sixth USENIX Security Symposium, Focusing on Applications of Cryptography,* San Jose, California, http://www.cs.berkeley.edu/~daw/janus/ .

[Hard92] Darren R. Hardy and Herb M. Morreale, ''buzzerd: Automated Systems Monitoring with Notification in a Network Environment,'' *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 203-210; Central monitoring server, remote monitoring daemons, paging on problems, users can put in notification, filtering, and escalation.

[Hark97] Robert Harker, ''Selectively Rejecting SPAM Using Sendmail,'' *Proceedings of the Eleventh Systems Administration Conference* (LISA '97), San Diego, California, pp 205-220, http://www.harker.com/sendmail/anti-spam; Configuring sendmail to reject spam messages.

[Hide94] Imazu Hideyo. ''OMNICONF – Making OS Upgrads and Disk Crash Recovery Easier,'' *Proceedings of the Eighth Systems Administration Conference* (LISA '94), San Diego, California, pp 27-31; Calculate a delta between two configurations, store the delta, apply it later.

[Hill96] Brian C. Hill, ''Priv: Secure and Flexible Privileged Access Dissemination,'' *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 1-8, ftp://ftp.

ucdavis.edu/pub/unix/priv.tar.gz; Secure ability to run programs as root with flexible command checking.

[Hohn99] Dirk Hohndel, "Automated installation of Linux systems using YaST," *Proceedings of the Thirteenth Systems Administration Conference* (LISA '99), Seattle, Washington.

[Kobl92] David Koblas, "PITS: A Request Management System," *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 197-202; Users can query database of open tickets, centralized assignment of new tickets, request editing tool.

[Kols92] Rob Kolstad, "1992 LISA Time Expenditure Survey," *;login:*; Administrator time spread over many tasks.

[Kols97] Rob Kolstad, "Tuning Sendmail for Large Mailing Lists" *Proceedings of the Eleventh Systems Administration Conference* (LISA '97), San Diego, California, pp 195-203; Configuring sendmail to increase performance.

[Limo99] Thomas A. Limoncelli. "Deconstructing User Requests and the 9-Step Model," *Proceedings of the Thirteenth Systems Administration Conference* (LISA

[Manh90] Kenneth Manheimer, Barry A. Warsaw, Stephen N. Clark, and Walter Rowe, "The Depot: A Framework for Sharing Software Installation Across Organizational and UNIX Platform Boundaries," *Proceedings of the Fourth Large Installation Systems Administrator's Conference* (LISA '90), Colorado pp 37-46.

[Metz92] Melissa Metz and Howie Kaye, "DeeJay – The Dump Jockey: A Heterogeneous Network Backup System," *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 115-125, ftp://ftp.cc.columbia.edu/.

[Mill99] Todd Miller, Dave Hieb, Jeff Nieusma, Garth Snyder, et al., "Sudo: a utility to allow restricted root access," http://www.courtesan.com/sudo/ .

[Oeti98a] Tobias Oetiker, "MRTG – The Multi Router Traffic Grapher," *Proceedings of the Twelfth Systems Administration Conference* (LISA '98), Boston, Massachusetts, pp 141-147, http://ee-staff.ethz.ch/˜oetiker/webtools/mrtg/3.0/ .

[Powe95] Patrick Powell and Justin Mason, "LPRng – An Enhanced Printer Spooler System," *Proceedings of the Ninth Systems Administration Conference* (LISA

[Pres98] W. Curtis Preston, "Using Gigabit Ethernet to Backup Six Terabytes," *Proceedings of the Twelfth Systems Administration Conference* (LISA '98), Boston, Massachusetts, pp 87-95.

[Ramm95] Karl Ramm and Michael Grubb, "Exu – A System for Secure Delegation of Authority on an Insecure Network," *Proceedings of the Ninth Systems Administration Conference* (LISA '95), Monterey, California, pp 89-93, ftp://ftp.duke.edu/pub/exu; A tool for providing fine-grain root access via authenticated, privileged scripts.

[Rich91] Kenneth Rich and Scott Leadley, "hobgoblin: A File and Directory Auditor," *Proceedings of the Fifth Large Installation Systems Administration Conference* (LISA '91), San Diego, pp 199-207, ftp://cc.rochester.edu/ftp/pub/ucc-src/hobgoblin; list of files/dirs + attributes => model. Checks for correctness, autogenerated from tar or ls listings.

[Roui94a] John P. Rouillard and Richard B. Martin, "Config: A Mechanism for Installing and Tracking System Configurations," *Proceedings of the Eighth Systems Administration Conference* (LISA '94), San Diego, California, pp 9-17, ftp://ftp.cs.umb.edu/pub/bblisa/talks/config/config.tar.Z; Update target machines using rdist + make with master repository in CVS, look for changed files with tripwire.

[Ruef96] Craig Ruefenacht, "RUST: Managing Problem Reports and To-Do Lists," *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 81-89, ftp://ftp.cs.utah.edu/pub/rust; Manages trouble ticket reports via e-mail.

[Schi93] John Schimmel, "A Case Study on Moves and Mergers," *Proceedings of the Seventh Systems Administration Conference* (LISA '93), Monterey, California, pp 93-98, How the merger of SGI & Mips was handled, physical move & computer configuration issues.

[Scot97] Peter Scott, "Automating 24x7 Support Response To Telephone Requests," *Proceedings of the Eleventh Systems Administration Conference* (LISA '97), San Diego, California, pp 27-35, Phone system for receiving problem reports and paging people.

[Shad95] Michael E. Shaddock, Michael C. Mitchell, and Helen E. Harrison, "How to Upgrade 1500 Workstations on Saturday, and Still Have Time to Mow the Yard on Sunday," *Proceedings of the Ninth Systems Administration Conference* (LISA '95), Monterey, California, pp 59-65; Tools & processes used to quickly upgrade an entire site.

[Silb91] Avi Silberschatz, Michael Stonebraker, and Jeffrey D. Ullman, "Database Systems: Achievements and Opportunities," *Communications of the ACM*, 34(10), pp 110-120.

[Silb96] Avi Silberschatz, Michael Stonebraker, and Jeffrey D. Ullman, "Database Research: Achievements and Opportunities into the 21st Century," Stanford Technical Report, http://elib.stanford.edu/ as CS-TR-96-1563.

[Silv93] James da Silva and Ólafur Guômundsson, "The Amanda Network Backup Manager," *Proceedings of the Seventh Systems Administration Conference* (LISA '93), Monterey, California, pp 171-182, ftp://ftp.cs.umd.edu/pub/amanda; Network backup by staging to a holding disk & streaming to tape, flexible scheduling.

[Spen96] Henry Spencer, "Shuse: Multi-Host Account Administration," *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 25-32, Centralized account management for rapid account creation.

[Stae98] Carl Staelin, "mkpkg: A software packaging tool," *Proceedings of the Twelfth Systems Administration Conference* (LISA '98), Boston, Massachusetts, pp 243-252, http://www.hpl.hp.com/personal/Carl_Staelin/mkpkg; Automatically generate manifest & install scripts; backend makes package.

[Trau98] Steve Traugott and Joel Huddleston, "Bootstrapping an Infrastructure," *Proceedings of the Twelfth Systems Administration Conference* (LISA '98), Boston, Massachusetts, pp 181-196.

[Troc96] Jim Trocki, "PC Administration Tools: Using Linux to Manage Personal Computers," *Proceedings of the Tenth Systems Administration Conference* (LISA '96), Chicago, Illinois, pp 187-192; Installation of DOS/Windows using Linux boot disk.

[Vali99] Peter Valian and Todd K. Watson, "NetReg: An Automated DHCP Network Registration System," *Proceedings of the Thirteenth Systems Administration Conference* (LISA '99), Seattle, Washington.

[Will93] Craig E. Wills, Kirstin Cadwell, and William Marrs, "Customization in a UNIX Computing Environment," *Proceedings of the Seventh Systems Administration Conference* (LISA '93), Monterey, California, pp 43-49; Study of how users customize their environment (copied from friends, then changed).

[Wong93] Walter C. Wong, "Local Disk Depot – Customizing the Software Environment," *Proceedings of the Seventh Systems Administration Conference* (LISA How to cache packages onto the local disk in the depot [Coly92].

[Wood98] Ben Woodard, "Building An Enterprise Printing System," *Proceedings of the Twelfth Systems Administration Conference* (LISA '98), Boston, Massachusetts, pp 219-228, http://pasta.penguincomputing.com/pub/prtools .

[Zwic91b] Elizabeth D. Zwicky, "Torture-testing Backup and Archive Programs: Things You Ought to Know But Probably Would Rather Not," *Proceedings of the Fifth Large Installation Systems Administration Conference* (LISA '91), San Diego, pp 181-189; Static & dynamic tests for many backup programs (dump, cpio, tar, etc.) shows many problems.

[Zwic92] Elizabeth D. Zwicky, "Typecast: Beyond Cloned Hosts," *Proceedings of the Sixth Systems Administration Conference* (LISA '92), Long Beach, California, pp 73-78, ftp://ftp.erg.sri.com/pub/packages/typecast .