



The following paper was originally published in the
Proceedings of the Twelfth Systems Administration Conference (LISA '98)
Boston, Massachusetts, December 6-11, 1998

What to Do When the Lease Expires: A Moving Experience

Lloyd Cha, Chris Motta, Syed Babar, and Mukul Agarwal - Advanced Micro Devices, Inc.
Jack Ma and Waseem Shaikh - Taos Mountain, Inc.
Istvan Marko - Volt Services Group

For more information about USENIX Association contact:

1. Phone: 510 528-8649
2. FAX: 510 548-5738
3. Email: office@usenix.org
4. WWW URL: <http://www.usenix.org>

What to Do When the Lease Expires: A Moving Experience

Lloyd Cha, Chris Motta, Syed Babar, and Mukul Agarwal – Advanced Micro Devices, Inc.
Jack Ma and Waseem Shaikh – Taos Mountain, Inc.
Istvan Marko – Volt Services Group

ABSTRACT

Moving a division of approximately 200 employees from one building to another across town can be a daunting task. It involves coordination among teams from systems administration, networking, facilities, and security as well as support from management and cooperation of the employees being relocated. Contractors and subcontractors are frequently hired to handle physical relocation of goods from one location to another, construction of new server rooms, electrical rewiring, installation of new cooling systems, etc. This paper is the story of how we handled the move and reconfiguration of a network of approximately 1000 nodes over a long weekend in May 1998.

Previously published work has discussed some of the issues that challenged us here. The reconfiguration of large numbers of machines has been discussed in [Manning93, Riddle94, Shaddock95]. “Forklift” upgrades of new hardware [Harrison92] share some but not all of the problems we faced in our move. Implementation of new networking topology without the problems or schedules imposed by physical relocation has been discussed in [Limoncelli97].

We believe our work is unique in requiring all these tasks to happen on a large scale in a relatively short time. We were allocated only one workday in addition to a weekend to shutdown and relocate our computing environment. We were expected to have a fully functioning network at our new location the following Monday. Ordinarily the complete reconfiguration of a network this size would be a challenge in itself. For our project, we had to account for the time required to disconnect and pack machines, load them into trucks, transport them across town, unload, and reconnect them at the new building. As we will detail, the resulting window of time available to handle the reconfiguration of all these machines was very small.

Introduction

The California Microprocessor Division of AMD (hereafter referred to as CMD) uses a large computer network of nearly 1000 nodes. Of these, over 600 are Sun and HP UNIX workstations and servers, primarily high-end dual-cpu UltraSparc machines. Most of the remaining nodes are NCD X-terminals and x86 compatible machines running Linux and Microsoft Windows. The majority of the CPU power resides in our server room on our headless compute server ranch. User desktops are mostly older Sun Sparc20 and HP 715 workstations or NCD X-terminals. Over three terabytes of disk space are served by six Auspex file-servers and one Sun E4000 with two RSM2000 chassis attached.

Most of CMD’s engineers were originally employed by NexGen, which had been purchased by AMD in 1996. Our division was located in the former NexGen building in Milpitas. There had always been a desire by upper management to move CMD to the Sunnyvale main AMD campus to better facilitate interaction among the various groups within the company. As CMD began to reach the limits of its building capacity, the move from Milpitas to Sunnyvale became reality.

The engineers from the Milpitas building were to be relocated to a recently vacated AMD building (hereafter referred to as building 920) in Sunnyvale. We determined there was insufficient space to house CMD’s rapidly growing server room in this building and therefore opted to locate CMD’s compute and file servers in a new data center to be constructed within an existing facility (building 915) across the street from building 920. Having the server room in a different building required remote monitoring equipment as described later in this paper.

Our system administration team consisted of seven members with primarily UNIX expertise and two members who focus on PCs. Ideally a team of system administrators could be dedicated to move related tasks, but unfortunately the rest of the division was unwilling to forgo support while we prepared for the move. While preparations for the move were a high priority, it was definitely only a part-time job until the move date actually arrived.

The project required an large influx of contract personnel to handle the more routine tasks required of the move, thereby freeing the regular system administration team up to perform the more involved tasks and to handle troubleshooting. The physical move of

goods from Milpitas to Sunnyvale was covered by our facilities department in the scope of the entire office move. We arranged with an outside contracting company to provide additional manpower on the weekend of the move. Their scope of work included project management of the network installation, purchase and installation of racks for the network and servers, the disconnection and reconnection of machines, a pre-move network audit, and additional people to help troubleshoot individual machine and network problems.

Our department hired our own contractor to provide project management and additional muscle. One team was responsible solely for disconnecting the machines in Milpitas and reconnecting them in Sunnyvale. Several contract systems administrators were brought in to help debug individual machine problems.

Planning

Planning for the move began approximately four months before the move date. Our first task was to define the scope of the project. In typical use, all of our UNIX CPUs see continual use on a 24 hour by 365 day schedule [Cha98]. We had not had a complete planned systems and network shutdown in over four years of operation. It was therefore tempting to incorporate a plethora of major system changes from our todo list that had accumulated over the years.

However, the amount of downtime we had available was severely limited by our tight schedule. We also had to balance the scope of any major changes against the risk of changing too many things at once. From past experience we knew that it was essential to allow at least one extra day to account for any unexpected problems and to tie up any loose ends.

We had to determine which tasks and system changes we would tackle, and which we would postpone until our next downtime (i.e., never). The merits of each potential change were weighed against the implementation time required and the overall risk to the project. We settled on the list presented in Figure 1. Inclusion or exclusion was based on several factors:

1. How much risk does it add to the project?
2. How much value does it add to the resulting environment? What are the consequences of not doing the proposed change?
3. How difficult would it be to do if postponed? Does it require a complete shutdown?
4. Is there sufficient time to complete the task?

There were many interdependencies among the included tasks. The new network design required new network interface cards (NICs) on the Auspex and the client machines. The new NICs required updated operating system installations on the Auspex servers and many of the client machines. The topology of the new network (detailed later in this paper) required Fast EtherChannel on the Auspex file servers. All these tasks were deemed essential to having a scalable network in place to accommodate future growth.

Changing the NIS domain name added relatively little value to the project, but added almost no risk and would be somewhat cumbersome to do at any other time. The excluded tasks required too many changes on both the part of the system administration team and the user community. The two tasks that were partially completed were deemed important, but failed to satisfy the fourth requirement in that we simply ran out of time.

Included tasks:

- Construction of new data center in building 915
- Implementing a new network infrastructure
- Changing all client CDDI NICs to 100baseT
- Upgrading the oldest Auspex (antigen) to a more modern chassis.
- Upgrading Auspex servers to 1.9.2 OS
- Changing all NIC cards in Auspex to full-duplex 100baseT
- Implementation of Fast EtherChannel in Auspex servers
- Implementation of terminal servers for remote console access
- Changing NIS domainname from systems to cpgca.amd

Excluded tasks:

- Eliminating filesystems mounted by direct automount maps
- Rearrangement of legacy filesystem organization
- Replacing Sun's automounter with the publicly available amd automounter

Partial tasks:

- Changing all client 10baseT machines to 100baseT
- Upgrading all client OS to a standard level

Figure 1: Move tasks.

Pre-move Tasks

In order to make the move and reconfiguration happen within the limited time available, it was critical that we perform as many tasks as possible in advance of the move date. We were able to take small groups of machines out of production in order to install new network interface cards and perform clean operating system installations. Having a consistent stable installation ensured that all machines would boot properly without encountering unusual problems that could have resulted from undocumented or erroneous changes to the machine through the years.

Unfortunately, critical machines such as the file-servers, license servers, and many of the more powerful compute servers could not be taken down prior to the move date. Upgrades to these machines had to be accounted for in the master move schedule.

All machines were labeled one week prior to moving. To avoid problems of illegible writing and inconsistent nomenclature, we used laser printed labels and distributed them to everyone involved. A sample label is shown in Figure 2. These printed labels were attached to larger color-coded moving company labels. Each color corresponded to a different zone in the new building to facilitate location of the equipment in the new building. The printed color-name on the label helped ensure that the right color label would be used.

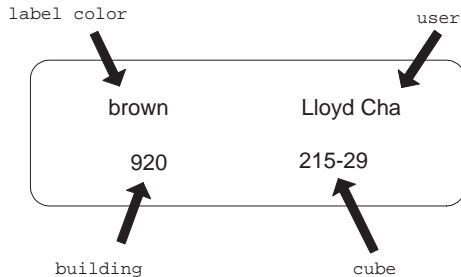


Figure 2: Sample label.

Typical moving labels are designed so that they can be peeled off furniture and the like after the move without leaving much residue. Unfortunately, we found that they stuck poorly to the cases of most of computer equipment. We ended up running around with rolls of clear tape to make sure that nothing fell off during the move.

The task of labeling the computers also provided a check to see what machines may have been missed in our master move list. In addition, we had included a detailed hardware audit in the scope of work that our contractor was to perform.

Unfortunately, the resulting audit turned out to be woefully incomplete. Due to some miscommunication, the audit was performed by starting with a list of

machines generated from our DNS database and then ascertaining their locations. While this did help qualify the DNS database, it did not perform the essential function of identifying hardware that had somehow been missed in the DNS listing. These machines included those that had returned from repair and had swapped names, machines that had recently been acquired, and machines operating without proper DNS registration (mostly renegade PCs).

We should have insisted on an audit that would physically cover every square foot of our building in a systematic way to identify every piece of computer hardware attached to our network. This would have allowed us to double-check our existing machine databases, identify archaic hardware that we should dispose of, and bring renegade machines to justice. This would also have given us a definitive checklist to go by to ensure that nothing had been missed by the movers.

Networking Considerations

Our existing network was not optimally configured as a result of extremely rapid growth with little available downtime. On several occasions over 100 machines at a time were added in batches with minimal disruption to the existing production network. This resulted in an irregular and poorly balanced network. Budget and downtime limitations prevented us from overhauling the entire network, which resulted in a hodgepodge of modern and outmoded equipment. Our backbone was a hybrid of ATM and FDDI, and client machine network interface cards (NICs) included 10baseT, 100baseT, CDDI, and FDDI.

Our options during the move were either to build a new network from newly purchased equipment and stage it several weeks before the move, or to tear down our old network the weekend of the move and use the components as a basis for building the new network. We were able to convince management that pursuing the latter option was too risky and would require too much downtime. We were able to justify the cost impact by noting that our older network equipment could be re-deployed to upgrade other even older equipment in use by other divisions within our company.

Changing technologies and the opportunity to build a scalable network from scratch resulted in a vastly different network design from our existing network. CMD's powerful workstations and implementation of LSF [Platform97] for load sharing demanded high-throughput between all machines with a particular emphasis on NFS traffic. Our new network is based primarily on 100baseT, with extensive use of Cisco's new FastEtherChannel technology [Cisco97, Cisco98a] for interswitch and fileserver [Auspex98] links. Further information about our new network topology can be found in [Cha98].

Our New Data Center

This move gave us the opportunity to construct a new data center in building 915 with many features that would improve our efficiency in maintaining the network. A large UPS system that covered the entire server room replaced a hodgepodge collection of single-system UPS units that were scattered around our existing data center.

Our old server room configuration had keyboards attached to every machine and a terminal that was wheeled around on a cart to attach to the console port of any machine that required attention. This was both messy and inefficient. For our new data center, we opted to use Xyplex 40-port terminal servers to provide remote console access. We reserved one port to possibly connect a modem for access to the machine consoles in the event that the network was completely down. Details of our implementation are available in [Cha98].

Our Sun servers automatically use the serial port if a keyboard is not detected upon bootup. One problem with Sun servers is that a break signal is perceived by the host if the serial port is connected to a terminal server that has been power cycled. Unfortunately, the team testing to see that the terminal servers would boot properly when power cycled was not aware of this fact and managed to simultaneously halt every machine connected to the terminal servers during their testing.

Fortunately our terminal servers are protected by the same UPS system that protects the Sun machines, so this problem should not be an issue except in case of operator error. As of this writing we have not put much effort into implementing a solution, however some suggestions are available in [Cisco98b].

The “100 Server” Move

We were permitted downtime on one hundred of our compute servers one weekend in advance of the primary move date. These machines were already located in Sunnyvale in a room adjacent to our new data center due to a lack of available space in Milpitas and were therefore not covered in the scope of our physical relocation.

We were taught an early lesson as to why extensive testing of scripts is essential before widespread implementation. The reconfiguration script had been only proofread but had not been extensively tested due to lack of time. It would have been time well spent. Our reconfiguration script suffered from two embarrassing and critical typos. The first problem was that the new default route was incorrectly entered, making the machines unreachable from anything but the local subnet. The second problem was that the NIS domain-name was misspelled, which caused every machine to hang at waiting for an NIS server to become available.

What’s worse, these errors caused name service to fail on these clients rendering the .rhosts file useless. To avoid having to write an Expect script to login to each of the machines, we temporarily stuffed our root passwords into a .netrc file so that we could use rexec to run our “cleanup” script to fix all the errors. We also added at least one IP address to each of our client .rhosts files so that we would not be stuck with this again.

This experience taught us that it’s worthwhile to run all procedures on one machine in “live” mode no matter how far behind schedule we were. No amount of proofreading is an adequate substitute for actually running the script or procedure and testing the results.

Schedule Overview

Our overall schedule looked like this:

- Thursday 6 pm:
All users logged off. Any jobs still running may be terminated. Shutdown script run on all desktop machines. Desktop machines powered off and packed.
- Thursday 9 pm:
Shutdown scripts run on all compute servers.
- Friday 3 am:
Final full backup before move for all file servers complete. File servers shutdown and packed for moving.
- Friday 6 am:
File servers begin loading onto moving vans, followed by compute servers and desktops.
- Friday 9 am:
File servers arrive in Sunnyvale and are unloaded and placed in new location for reinstallation. Aupsex begins reinstallation and required hardware changes.
- Friday noon:
Hardware changes to file servers complete. Compute servers and desktops begin to arrive at new building.
- Friday 6 pm:
Operating system upgrades to file servers complete. File servers should now be fully operational. Compute servers and desktops should all be in place. Reconnection of compute servers and desktop machines begins.
- Saturday 6 am:
All compute servers and desktops operational. Exceptions to be noted and addressed by systems admin. Global script run to check health of all machines.

We opted to set a very aggressive schedule and allow sufficient time for slips in the schedule rather than setting a relaxed schedule with very little room for error. Had we following a relaxed schedule, we would run the risk that one of the tasks in the late stages would slip leaving little room for recovery.

That was our plan. The actual timeline of events which we will detail below turned out to be somewhat different.

Shutting Down

Our shutdown scripts had two main objectives: reconfiguration and shutdown. First, the shutdown script modified each machine's startup files such that on power up the system would be configured for the new network. At a minimum the IP address, netmask, and NIS domainname would need to be changed. Some machines would require new network interface

cards. After applying all these changes, the scripts would 'sync' and 'halt' each machine in preparation for being powered down and disconnected.

A single master reconfiguration file drove all the IP address changes. All the shutdown scripts consulted this file, as did the script that we used to generate our new DNS source files. Using a single master file allowed for easy insertion of last minute changes and also ensured that the local host files and DNS database would be consistent.

Our original plan called for deploying the reconfiguration scripts to the local /tmp disk of every

```
#!/bin/csh
# Script for halting a machine
#
if (-x /usr/local/bin/get_arch) then
    set archtype = '/usr/local/bin/get_arch'
else
    set archtype = "unknown"
endif
switch ($archtype)
    case 'sunos':
        /usr/etc/halt >& /dev/null
        exit 0
        breaksw
    case 'solaris':
        /usr/sbin/halt >& /dev/null
        exit 0
        breaksw
    case 'hp9':
        /usr/sbin/shutdown -h -y 0 >& /dev/null
        exit 0
        breaksw
    case 'hp10':
        /usr/sbin/shutdown -h -y 0 >& /dev/null
        exit 0
        breaksw
    case 'linux':
        /sbin/halt >& /dev/null
        exit 0
        breaksw
    default:
        exit
endsw
```

Figure 3a: Sample shutdown script.

```
#!/bin/csh
# Wrapper script for shutting down machines
/usr/local/admin/move/scripts/halt_sys >& /dev/null &
exit 0
```

Figure 3b: Sample wrapper for shutdown script.

machine. We would then start a process that would run in an infinite loop, waking up periodically to look for the local hostname in a central control file. Once the hostname appeared, the reconfiguration script would be launched out of /tmp and the machine would shut itself down.

Due to last minute changes in the reconfiguration scripts, we did not have enough time to distribute the changed scripts to each machine. The fallback was to rsh to each machine using a simple foreach loop and call the reconfiguration script from a central location. The problem with this approach is that some machines would hang on the rsh during the shutdown call, and the foreach would not continue. A more robust solution is to rsh to the machine with a call to background the actual halt script using csh, which would allow the foreach to continue. Sample scripts are provided in Figure 3.

Once the shutdown scripts executed, teams of contractors moved through the building powering down and disconnecting machines. Peripheral components such as keyboards and mice were collected into labeled clear plastic bags and were left with the machines for pickup by the movers.

Bringing Up the Fileservers

The critical path in our move schedule centered around the move and reconfiguration of the Auspex servers. They were the last machines to be shutdown in the old building and the first that would be required in the new location. In addition, substantial hardware changes were required to support the new network hardware.

We hired Auspex to handle the packing and unpacking of our Auspex servers. Packing materials designed specifically for the equipment we were moving were shipped well in advance to our old address and staged in preparation for the move. We completed a full level 0 backup just before shutting the file servers down. Everything had gone smoothly up to this point.

The disassembly of the Auspex servers took longer than expected, and the movers were several hours late in delivering the goods to the new location. It appeared that both our moving company and Auspex had underestimated the amount of manpower needed to complete the tasks required in the allotted time. This put us several hours behind on the most critical path of the schedule. We encountered additional delays in getting the new hardware installed in the Auspex servers required to support our new network. By midnight Friday night, the file servers still had not been powered up. By this time most of the vendor technicians and support engineers who had been working since 3am the previous night were exhausted. Powering up the servers brought more unpleasant surprises. Three of the seven servers failed to boot due to improperly installed hardware. We

spent most of the night debugging server problems.

The one Auspex server that we were upgrading during this move was an exception to the usual process. This was a “forklift-style” upgrade in which the new server was delivered with the old server being returned for credit at a later date. The new server was ready by the move weekend awaiting the final step of copying over 300GB of data from the old server to the new. We considered several methods:

1. Pre-load the new server from the last full backup (one week before the move). Run an additional incremental backup on the old server before shutting down and apply it to the new server on the weekend of the move.
2. Pre-load the new server by copying data over the Milpitas-Sunnyvale WAN (OC-3). During the move, use rdist to update any files that have changed.
3. Do nothing before the move. During the move, attach both the old and new disk chassis to the new server and copy the data locally from disk to disk. This can be done rapidly on the Auspex architecture by using the dedicated storage processors and bypassing the UNIX kernel altogether.

We had initially attempted to use option 1, but our efforts were derailed due to issues with the backup software and licensing that severely impacted performance. Option 2 proved to be impractical due to long transfer times over the WAN link. We ended up using option 3, the direct transfer from disk to disk.

Option 3 provided the fastest and most foolproof method of transferring the data but had a critical disadvantage that it would have to be completely done during the move weekend. Unfortunately the process also required a lot of manual work by the operator, some of which could probably have been scripted in advance although we failed to do so. The hardware and disk swaps necessary to the procedure could not be automated anyways. The entire procedure became the single-most time consuming task of the entire weekend. The complete copy did not finish until sometime Saturday afternoon.

Lessons learned:

- Be sure that adequate manpower is available, with additional help available to be called in if needed. Overworking the move team is unwise as it increases the probability of mistakes.
- Script out and document changes in detail. Even the simplest procedures become error prone when operating under pressure or sleep deprivation. Leave nothing to chance.
- Be realistic about schedules and have a contingency plan in place should things fall far behind.
- Don't drive yourself home after staying up for 30 consecutive hours.

Bringing Up the Client Machines

Our general plan was for contractors to oversee the routine powering up and booting of client machines. They would login to a test account which would run a script that would test for basic functionality. Exceptions would be flagged and handled by our system administration team.

The number of system administration problems encountered in bringing up client machines was overshadowed by logistical problems related to the physical move itself. Our moving company had underestimated the amount of manpower necessary to relocate the goods while keeping to our schedule. Many machines were delayed in their arrival, and those that did arrive were often placed incorrectly despite being properly and clearly labeled. System configuration problems were rapidly fixed, but valuable time was lost scouring the building for misplaced monitors, keyboards, and other parts.

Results

Despite our schedule setbacks, we were able to provide a fully functional computing environment for our users by Monday morning. Our LSF reports (our LSF setup is discussed in [Cha98]) indicated that fewer than 5% of the machines were unavailable on Monday. None of these machines was critical and the majority of these were due to circumstances somewhat beyond our control (e.g., hardware failures, lost peripherals, etc). While we were happy with the end results, we realize that many of the lessons learned from our experience could have saved a lot of extra effort and stress. We hope that readers of our paper will be able to benefit from both our successes and failures.

Suggestions for a Successful Move

In this section we provide additional tips and techniques that we have learned from our experience that were not already covered above. Some of the suggestions reflect things that worked well for us, while others were mistakes we learned to avoid. Several are common sense suggestions that could easily be overlooked in the rush. In no particular order:

- Keep copies of essential data related to the move and reconfiguration in a central location. Don't forget to copy them to an external or stand-alone system before powering down since you won't have access to them while your network is in transit.
- Make a priority list and make sure everyone on the project has a copy. Solve problems that affect the most systems first. Document dependencies (eg. NIS server needs to be available before NIS clients are booted). Much of this seems obvious, but remember that in a reasonably stable environment these requirements are often taken for granted.

Typically file servers, DNS servers, and NIS servers should be attended to first followed perhaps by the mail server, license servers, and web servers. In our particular case, the LSF master server [Cha98] was also a high priority since we used it to collect data on the general health of our computing cluster.

The priority list should also take into account the relative importance of the various services provided to the users. If it becomes impossible to have everything available on schedule, you will at least want to have the essential functions available.

- Have a healthy stock of basic office supplies and deploy them in accessible places at your new location. You want to avoid wasting time looking for a pen and paper, and you certainly don't want important notes left untaken for lack of a writing instrument. Permanent markers are invaluable.
- Communication is essential. Pagers, cordless phones, and walkie-talkies are all useful although be aware that some older computer hardware may be sensitive to radio transmissions. Setup a command center with a whiteboard or bulletin board in a visible area that is kept up-to-date with the latest status and whereabouts of key personnel.
- Arrange a blanket purchase order or other means of purchasing necessary items that may have been overlooked in earlier planning. We were fortunate to be located near a large Silicon Valley electronics store that was able to handle our emergency supply needs.
- Provide meals and snacks on-site. This keeps critical people from wandering off to lunch or dinner and becoming unavailable for consultation.
- Laptops are invaluable for storing information (IP lists, phone lists, etc), connecting to outside systems, substituting for dumb terminals, downloading patches and licenses, testing the network, etc.
- Script and automate as much as possible. Document routine procedures that cannot be scripted. Spread knowledge around as much as possible.
- Label label label! Audit and inventory all items. Label individual components and SCSI ID and DIP switch settings that may get inadvertently bumped during transportation. If possible, have the audit performed by someone familiar with the hardware involved to prevent headaches caused by improper or ambiguous information.
- Test whatever can be tested. Stage a dry run or preliminary move if possible.
- After the move, provide information to users on whom to contact for problems. Setup a hotline to coordinate trouble requests if one doesn't

exist.

- Cleanup as much as possible before the move!
- Give your users incentive to co-operate. Get support from management to send them to a movie or other company sponsored events to keep them out of the way. Set reasonable expectations.

Availability

Please contact the authors at <lisa98@cmd-mail.amd.com> regarding availability of scripts referenced in this paper.

Author Information

Lloyd Cha is a MTS CAD Design Engineer at Advanced Micro Devices in Sunnyvale, California. Prior to joining AMD, he was employed by Rockwell International in Newport Beach, California. He holds a BSEE from the California Institute of Technology and a MSEE from UCLA. He can be contacted by USPS mail at AMD, M/S 361, PO Box 3453, Sunnyvale, CA 94088 or by electronic mail at <lloyd.cha@amd.com> or <lloyd.cha@pobox.com>.

Chris Motta is the manager of the CMD Systems and Network Administration department. He holds a BSME from the University of California at Berkeley. He has held a variety of systems administration positions including UNIX and networking consulting. Electronic mail address is <Chris.Motta@amd.com>, and USPS mail address is M/S 366, PO Box 3453, Sunnyvale, CA 94088.

Syed Babar received his master's degree in computer engineering from Wayne State University in Detroit, Michigan. He works at Advanced Micro Devices in Sunnyvale, California as a Senior CAD Systems Engineer. He can be contacted via e-mail at <Syed.Babar@amd.com> or <Syed_Babar@hotmail.com>.

Mukul Agarwal received his MSCS from Santa Clara University. He joined Nexgen, Inc. in Milpitas, California as a CAD Engineer in 1993. He switched to systems and network administration in 1995 and has been a System/Network Administrator ever since. Reach him via e-mail at <mukul.agarwal@amd.com>.

Jack Ma holds a BSCS from Tsinghua University and a MSCS from Computer Systems Engineering Institute. He was a UNIX software developer at Sun Microsystems before joining Taos Mountain at 1995, where he now works as a networking/UNIX system consultant. He can be reached electronically at <ylma@netcom.com>.

Waseem Shaikh holds a master's degree in computer engineering from the University of Southern California and received his bachelor's degree in electrical engineering from University of Engineering and Technology in Lahore, Pakistan. He was a

System/Network Engineer at Steven Spielberg's Holocaust Shoah Foundation, a System Consultant at Stanford Research Institute, and is now working as a System/Network Consultant with Taos Mountain. He can be reached at <shaikh@netcom.com>.

Istvan Marko is a self-educated Computer Specialist currently working as a System Administrator employed through Volt Services Group. He can be contacted via e-mail at <imarko@pacificnet.net>.

References

- [Auspex98] "Auspex Support For Cisco Fast EtherChannel," *Auspex Technical Report #21*, Document 300-TC049, March 1998.
- [Cha98] Cha, Lloyd, et al., "The Evolution of the CMD Computing Environment: A Case Study in Rapid Growth," *LISA XII*, Boston, MA 1998.
- [Cisco97] Cisco Systems, Inc. *Fast EtherChannel*, Cisco Systems Whitepaper, 1997.
- [Cisco98a] Cisco Systems, Inc. *Understanding and Designing Networks Using Fast EtherChannel*, Cisco Systems Application Note, 1998.
- [Cisco98b] Cisco Systems, Inc. *Terminal Server Break Character on Cisco Access Servers*, Cisco Systems Field Notice, April 21, 1998. URL: <http://www.cisco.com/warp/public/770/fntsbreak.html>.
- [Harrison92] Harrison, Helen E, "So Many Workstations, So Little Time," *LISA VI Proceedings*, Long Beach, October 1992.
- [Limoncelli97] Tom Limoncelli, Tom Reingold, Ravi Narayan, and Ralph Loura, "Creating a Network for Lucent Bell Labs Research South," *LISA XI Proceedings*, San Diego, CA, October 1997.
- [Manning93] Craig Manning and Tim Irvin, "Upgrading 150 Workstations in a Single Sitting," *LISA VII Supplementary Materials*, Monterey, CA, November 1993.
- [Platform97] Platform Computing, "AMD's K6 Microprocessor Design Experience with LSF," *LSF News E-mail Newsletter, Platform Computing*, August 1997.
- [Riddle94] Paul Riddle, "Automated Upgrades in a Lab Environment," *LISA VIII Proceedings*, San Diego, CA, September 1994.
- [Shaddock95] Michael Shaddock, Michael Mitchell, and Helen Harrison, "How to Upgrade 1500 Workstations on Saturday and Still Have Time to Mow the Yard on Sunday," *LISA IX Proceedings*, Monterey, CA, September 1995.