Berkeley UNIX† on 1000 Workstations:

Athena Changes to 4.3BSD

G. Winfield Treese

Project Athena
Massachusetts Institute of Technology
Cambridge, MA 02139
treese@ATHENA.MIT.EDU

ABSTRACT

4.3BSD UNIX as shipped is designed for use on individually-managed, networked timesharing systems. A large network of individual workstations and server machines, all managed centrally, has many important differences from such a model. This paper discusses some of the changes necessary for 4.3 in this new world, including the file system layout, configuration files, and software. The integration with Athena's authentication system, name service, and service management system are also discussed.

1. Overview

‘‘By 1988, create a new educational computing environment environment at MIT built around high-performance graphics workstations, high-speed networking, and servers of various types.’’ This one-sentence statement is a high-level description of the technical goals of Project Athena. While the primary goals are to enhance education, attaining them has required a significant effort to engineer a software environment for use in a large network of workstations and servers.

The Athena hardware environment currently consists of approximately 650 workstations and 65 dedicated server machines. There are two kinds of workstations: DEC MicroVAX systems and IBM RT PC's. The servers are VAX 11/750's or dedicated workstations of either type. The operating system in use now is 4.3BSD UNIX on the VAX machines, and IBM's 4.3/RT UNIX for the RT PC systems. All systems include support for Sun Microsystem's Network File System (NFS).[1] The workstations and servers are connected to local-area Ethernet subnetworks, which are linked by a high-speed fiber optic ‘‘spine.’’ At present, there are twelve such subnetworks.

The problems of a distributed system are the scale of the operation and the role of the network as a fundamental component. UNIX systems have traditionally been managed on a ‘‘one system, one wizard’’ basis, but this is not acceptable at an eventual scale of 1000 workstations, 100 server machines, and 10,000 users. Two questions often asked are: ‘‘Does it scale?’’ and ‘‘Is it well-behaved on the network?’’ All too often, the answer to one or the other is ‘‘No,’’ and part of the system must be reworked to satisfy those constraints.

This paper describes the goals and constraints faced by Athena, as well as many of the solutions devised in building such a system. In particular, the next two sections examine the goals and evolution of the computing system side of the Project. Next is a discussion of the base operating systems in use, such as 4.3BSD. This is followed by a description of the constraints of scale and the network, as well as some of the solutions bounded by those constraints. Finally, system configuration issues and future directions

† UNIX is a Trademark of Bell Laboratories.

for enlarging the workstation base are discused.

2. Goals

The following are four important goals for engineering a Project Athena workstation system.

It must provide a coherent environment across heterogeneous hardware. When Project Athena was created, ‘‘coherence’’ was identified as an important characteristic of a successful workstation environment. Briefly, ‘‘coherence’’ means that the environment seen by users on different workstations should be as similar as possible. To a first approximation, this is achieved by using Berkeley UNIX systems on all workstations. There are many other constraints imposed by coherence, however; these are discussed in detail below.

It must provide rich computing environment for the Athena user community. To make a workstation optimal for educational use at MIT, a rich software environment must be available. If, for example, the software is not useful for building tools for teaching a thermodynamics class, that class will not use Athena workstations. Many third-party software packages, including an editor, text formatter, and spreadsheet, have been added to the standard UNIX system for this reason.

It must scale to 1000 machines such that the Athena Operations staff can manage them. The entire network of workstations must also be manageable. The Athena Operations staff is not large, but it is currently responsible for over 700 machines. Hence, a solution to a problem that requires any resources in proportion to the number of machines (e.g., having to visit each workstation) is unaffordably expensive. Wherever possible, operations tasks should be automatic; otherwise, they should at least be performed centrally.

It must behave well on the network. Athena workstations must also be ‘‘good neighbors’’ on the network; they should not generate problems for other systems on the network, such as by gratuitously consuming network bandwidth.

3. Evolution of Project Athena

At the time of Project Athena’s inception in 1983, workstations were still in the design phase. In the beginning, then, Athena used off-the-shelf hardware and software. Approximately 50 DEC VAX 11/750 systems were deployed as traditional timesharing systems running Berkeley UNIX (4.2BSD, later 4.3), and over 100 IBM PC/AT’s were deployed as networked single-user machines. Using both systems yielded much information about managing and engineering networked UNIX systems and single-user machines.

Workstations became available to Athena staff in late 1985. At that time, Athena attempted to apply the lessons learned from running timesharing systems to distributed workstation systems, as well as to develop solutions to the new problems posed by workstations. Much of the design work was done ‘‘on the fly’’ in order to make usable workstations available to staff, and eventually to users, as quickly as possible.

The first student workstation clusters, running a prototype system, were opened for use in the fall of 1986. Since then, the number of workstations available has steadily increased. During the 1986-87 academic year, both timesharing and workstation systems were in use. Over the summer of 1987, the Project shifted almost entirely to the workstation environment, with only a few timesharing systems in use for classes with specific requirements (e.g., those using a non-networked database system). The majority of the timesharing systems were converted to NFS file servers, primarily for student ‘‘lockers,’’ or file storage areas.

4. Hardware

The typical Athena workstation is roughly a ‘‘3M’’ machine; that is, it has a 1 million-instructions-per-second processor, a megapixel (1000 x 1000) display, and three or four megabytes of memory. It also has a mouse, a local disk (typically 30 - 70 megabytes), and an Ethernet network interface. Actual hardware in use includes DEC VAXstation II and VAXstation 2000 systems, and IBM RT PC desktop models. Various other configurations of MicroVAX and RT PC systems are used for development and as dedicated servers.

The heterogeneous hardware base has imposed certain constraints on Athena software. For example, the VAX and the RT PC architectures use different byte orders to represent integers. Hence, if data is to be exchanged between the two architectures, the software must be prepared to handle this difference. This is true both of network protocols and of files used for data storage, since such files may be transferred between machines.

The hardware differences also make the goal of coherence difficult to reach. For example, the compilers available on the two machines are somewhat different. While the differences are minor, they can be most annoying. Since the same source code is used on both machines whenever possible, the standard source pool has been partitioned into sections of machine-independent and machine-dependent sources to simplify the system build process. The compilers, for example, are machine-dependent; a text editor such as */bin/ed* is not.

Of course, hardware from other vendors meets the specifications for the basic Athena workstation. In the future, the ‘‘Athena Environment’’ will be produced as a layer on top of vendor-supplied systems that meet certain fundamental requirements.

5. Base Software Systems

The basic software used on all Athena workstations at this time is 4.3BSD UNIX with machine-dependent software supplied by the hardware vendors. All systems include NFS client support and use the X Window System.[2] The standard Berkeley distribution is augmented by several third-party software packages and local Athena modifications and additions.

RT PC System. The RT PC kernel is taken from 4.3/RT. NFS support has recently been integrated into that kernel at Athena. The necessary machine-dependent utilities (e.g., the C compiler) were also drawn from the 4.3/RT system.

VAX System. The basic VAX system is 4.3+NFS from the University of Wisconsin. Some device drivers from Digital’s Ultrix-32 have been added to handle some of the hardware in use at Athena.

6. Constraints of Coherence.

Providing a coherent environment across the different types of workstations imposes several constraints on constructing the system. These constraints include the following.

The same basic system must be available on all platforms and must evolve in synchrony. To promote both coherence and maintainability, all software is periodically built from the source code. This ensures that all changes to header files, libraries, etc., are actually reflected in the running system. Maintaining and building sources for two different architectures turns out to be no small task. Unfortunately, *make* and *rdist* alone are not sufficient to maintain source code on multiple architectures, especially when some changes (e.g., kernel modifications) must be made to code that is similar but not identical. New tools to automate this procedure are under development.

Data must be interchangeable between the systems. Since different machines may use different internal data formats, applications must be prepared to cope with such variations. This is particularly true of network services, since services may be available from different architectures. Fortunately, this constraint is not difficult to observe, *provided it is anticipated*.

7. Constraints of Scale

Scale is one of the driving considerations in building the Athena system. Unfortunately, not all constraints of scale are obvious at first glance (or even the second or third). Some of the specific constraints of a large scale system include the following.

Resources cannot be expended in proportion to the number of workstations. This may seem obvious, but it is a constant problem. Athena Operations, for example, does not have resources that grow in proportion to the number of workstations installed. The only way to support those workstations is to make operational tasks more efficient. To use economies of scale, one must first find the economies.

Differences in software, including configuration files, must be minimized. One way to minimize operational tasks is to minimize the differences between workstations. There are more than twenty different configuration files in a standard 4.3 system. Workstation systems are much easier to manage when most of these are identical from workstation to workstation. On Athena public workstations, all configuration files are identical except for */etc/rc.conf*, which contains the hostname and network address. Operators can be trained to understand two or three configurations, but expecting them to understand the subtleties of 1000 is too much. If configuration files are identical, an operator can simply look at the configuration file that seems to be incorrect, realize that it is not standard, and copy the correct version from a standard source.

There are, of course, some small exceptions to this rule. In particular, a workstation must be able to find the nameservers in the first place. At this time, a list of nameservers is maintained on the workstation’s local disk. In the future, this should not be necessary (see ‘‘Future Plans’’).

Network services must be redundant to tolerate network and server failures. If a central name or authentication service is not available, the world will grind to a halt. Providing redundant servers minimizes the probability of a given service being unavailable, as well as distributing the load over several different servers when all is well. In contrast, it is not always possible, nor is it necessary, to provide redundant servers for modifying central database information that is later distributed to appropriate servers (e.g., the central nameserver database). As a corollary to this, critical network services should also rebound quickly after system crashes or power failures.

A centrally-managed name service should provide information for finding network services. Suppose the FOO service is provided by machine HERA.MIT.EDU. If this service moves to ZEUS.MIT.EDU, it should not be necessary to update configuration files on 1000 workstations. Indeed, experience has shown that there would be workstations with the wrong information months after the change takes place. A name service provides a much easier way to manage this problem.

Software installation must be quick and painless. Installing new software on a workstation should not take much time, nor should it require many props. Since workstation systems at Athena are installed to a standard form, the installer need perform very little customization; the rest of the procedure is automatic. In fact, for most workstations, only the hostname must be customized.

Software update must be quick and painless. Automatic update is even better, if it works correctly. In an ideal world, software update would be completely automatic. In an automatic update, a workstation compares its software to a central library and makes any necessary changes. Some software must be updated with great care, however. For example, a new kernel requires that the workstation be rebooted, and often requires that various kernel-dependent utilities, such as *ps*, be updated at the same time. The tools for doing this are not yet entirely reliable.

Configuring servers should be quick, painless, and reproducible. Service machines (e.g., printer servers) are implemented as a set of differences from a basic workstation system. Converting a standard workstation to a server is then an straightforward process. This allows a vanilla machine to be ‘‘swapped in’’ quickly for a server should hardware problems arise. It also simplifies updating servers with the latest software release.

The workstation root password is not a secret. In a large environment such as Athena’s, the root password to workstations cannot be different on each machine. Indeed, with the workstation physically available to a user, it is not necessary to know the root password; booting the machine in single-user mode is sufficient to gain root access. A user may therefore easily modify the software local to the workstation, and network services must not blindly trust root users from workstations.

Network services must be managed centrally. Trying to manage a large number of network services can become quite difficult. When only 10 or 100 workstations are involved, it may be possible to manage services on one or two machines, a job that can be done by hand. On a larger scale, the number of servers must also be considered. To this end, Athena has designed the Service Management System (SMS),[3] which consists of a central database of service information, tools for manipulating that information, and tools for extracting appropriate information for the services themselves. A particular advantage of maintaining this database is that it reduces to one the number of different places a given piece of information must be stored: it is kept once in the database and is provided to servers as they need it.

8. Constraints of the Network

Working in a networked environment also imposes several constraints on the overall system. One of the most important for users is that several services available on timesharing systems should be available on workstations as well. In some cases, these services behave somewhat differently than in the timesharing world, but functionality is preserved. Some of the services include the following.

System Libraries. Timesharing systems have many programs available for a user to execute; most of these are traditionally found in */usr*. Workstations do not typically have the hundreds of megabytes necessary to keep the complete system library available on a local disk, and it would be impossible to keep the software up to date if they did. To solve this problem, Athena has provided a ‘‘Remote Virtual Disk’’ (RVD) service. RVD was originally developed at MIT’s Laboratory for Computer Science and significantly enhanced at Project Athena. An RVD ‘‘pack’’ appears to a workstation just as a local physical disk device does. An RVD server only supplies requested disk blocks; all filesystem information is manipulated by the client workstation. As a result of this, an RVD pack may be used by many clients in a read-only mode, or by a single client in an exclusive read-write mode. Another effect of the block-level service is that a single VAX 11/750 server can support 75 client workstations with quite acceptable performance.

Name Resolution. On a timesharing system, names are often translated to machine-usable data by configuration files. For example, */etc/hosts* maps machine names to numerical Internet addresses. Another file might contain the name of the RVD server that provides the system libraries. In a large, dynamic environment, this reliance on static files causes innumerable problems. To manage changing information, it is necessary to have a service that provides name translation. Athena undertook some minor extensions to the BIND[4] nameserver package from 4.3BSD to provide generalized name service; the resulting software is known as *Hesiod*.[5] *Hesiod* provides information on users, locations of user lockers, locations for various network services, etc. Changes in the *Hesiod* database, which is generated by the Service Management System, are available to all workstations within a few hours.

Authentication Service. UNIX systems have traditionally stored encrypted passwords in */etc/passwd* on each machine. It is quite difficult to maintain password and group files on 50 timesharing machines and completely impractical to provide complete password and group files for each machine in a large network, so an alternative authentication system is required. Athena has implemented a system known as *Kerberos*[6] to handle authentication for network services, including workstation login.

File Service. A user’s files should be available for use on any workstation. At Athena, each user has a ‘‘locker’’ for personal storage. User lockers are distributed across many file servers, but the appearance to the user is that the home directory is as expected, and it is the current directory at login, just as on a timesharing system. One difference is that other users’ home directories are not immediately available but must explicitly attached to the workstation’s directory hierarchy. The net effect, however, is positive: in the earlier days of Athena timesharing, a user could not gain access another’s files if they were on a different machine. Security and privacy considerations are observed, however, and the usual UNIX protections are enforced.

Printing. Typical timesharing systems have printers available locally. The environment assumed for 4.3BSD makes some attempt to provide networked printing facilities, but the *lpr* system is still difficult to manage. Two of its major problems are local queuing and local configuration. Normally, *lpr* drops a print request in a queue on the local machine, and a line printer daemon either prints it or queues it to a remote machine, as specified in */etc/printcap*. *lpr* gives the user no indication whether or not the machine physically connected to the printer is actually up and accepting requests. In the Athena workstation world, this can cause a file to be left in a local queue on a workstation, with no guarantee that it will ever be printed. To circumvent this problem, *lpr* has been modified to queue directly to the remote printer server. If it is not available, the user receives an immediate error and may try to find another printer to use.

The second problem is */etc/printcap* itself. As a static configuration file, the copy on every workstation must be updated when a new printer is added or an old one moved. *lpr* has been modified to query *Hesiod* for information about printers in order to find a printer server. Printer servers themselves behave more traditionally; they use a local */etc/printcap* for detailed information about their own printers.

Electronic Mail. Timesharing systems provide a convenient maildrop for users. The address is typically *user@machine*, and local users can be addressed simply as *user*. Where, then, should mail be kept for a user who might login on workstation ABC one day and workstation XYZ the next, especially when none of the workstations has sufficient local disk space to store mail? Athena has adopted the concept of a ‘‘post office,’’ which holds mail for a user until he picks it up. The Post Office Protocol[7] software supplied with the MH mail system[8] has been modified for use with *Kerberos*, and the retrieval software finds the mailbox using *Hesiod*.

The changes are transparent, so the user simply uses the same commands as he did on the timesharing system, without having to worry about the details. At this time, Athena has three post offices in use, serving approximately 8000 users.

Sending mail to Athena users is also straightforward. The address is simply *user* within Athena, or *user@ATHENA.MIT.EDU* from outside. All mail is routed by a central mail hub, which uses a master list of users at various post offices. The list is provided by SMS. This machine also handles distribution to mailing lists within Athena. When a user sends mail from a workstation, it is queued to the mail hub for forwarding to a post office, mailing list, or outside machine, as appropriate. Note that *all* mail goes to the mail hub; workstations do not try to contact foreign machines themselves, nor do they run a *sendmail* daemon to receive incoming mail. At this time, sending mail does suffer from one of the same problems as *lpr*: mail is queued locally on the workstation if the mail hub is not responding. In such a case, there is no guarantee that it will ever be delivered; the next user may willfully or mistakenly delete any mail in the spool directory.

Notification. On a timesharing machine, it is easy to notify a user asynchronously of some event: one need simply find the entry in */etc/utmp*, if one exists, and send a message to the appropriate terminal. Where, however, does one find a user somewhere in the forest of 1000 workstations? And where does one deliver a message in a thicket of windows? To solve this problem, Athena has developed a notification system known as *Zephyr*,[9] which can be used both to locate users and to send messages to them. One simple example is the command *zwrite treese*, which will deliver a message to user *treese* on his workstation if he is logged in somewhere on the Athena network (within constraints of permission and privacy, of course).

Remote Access. An Athena workstation is designed to be a single-user machine, if for no other reason than that it is easy to gain root access on one. By default, Athena workstations do not permit remote logins, remote shells, or remote file access, since that may harm the current user of the workstation (i.e., the one logged in on the console). A user can defeat this protection and allow access to a workstation during a session if desired. The major disadvantage of this restriction is that operations personnel cannot remotely login to repair problems; experience has shown that this is an acceptable limitation.

On-Line Consulting. On a UNIX timesharing system, one can usually find the system manager or other knowledgeable user when one runs into problems. Finding assistance is much more difficult when there are 5000 active users of the system, with the Athena staff system wizards somewhere on the other side of the campus. To solve this problem, Athena implemented an ‘‘On-Line Consulting System’’ (OLC) that can be used to ask questions of consultants and other knowledgeable users logged in elsewhere one the network. Questions are saved until a consultant becomes available, and answers are often returned by electronic mail if the question remains unresolved when the user logs out.

Network Etiquette. Athena workstations must be good neighbors on the network. Services that require the use of Ethernet broadcast packets are almost always unacceptable by that measure. They raise two problems: first, the broadcasts are limited to the local net (as a matter of policy) and can therefore reach only a small fraction of the workstations. Second, they tend to generate a great deal of network traffic when many machines are involved, consuming valuable network bandwidth and local processing cycles. One example is the *rwho/ruptime* software from 4.3BSD; handling the packets from nearly a hundred workstations on the same local net seriously affected workstation performance in the early days of workstation use at Athena. Athena machines no longer run that software.

9. Configuration Changes

In satisfying these constraints, Athena has made several changes to the standard UNIX configuration for use on workstations. These fall into four categories: changes to the directory hierarchy, introduction of ‘‘activated’’ and ‘‘deactivated’’ states for a workstation, modification of the login procedure, and changes to configuration files. Strictly speaking, many of these changes are not necessary for the public workstations most common at Athena, but are included for use on non-public machines, such as those in the offices of MIT faculty members.

Directory structure. Most of the changes in the directory structure are in */usr*. Because some standard subdirectories are used for executable programs and some for spooling and administration, */usr* on an Athena workstation actually contains a set of symbolic links. Directories such as *adm*, *spool*, and *crash* are stored on the local disk on */site*; other directories are actually subdirectories of */urvd*, a read-only RVD system library mounted at activation.

Over time, the root filesystem has also overflowed its size. To avoid reconfiguring all machines with a larger filesystem, many programs in */etc* or */bin* are actually symbolic links to a second RVD system library, mounted on */srvd*. These programs are those which are not essential for a workstation during its boot procedure or for repairing a workstation in single-user mode. For example, the C compiler and the assembler fall into this category. The */srvd* system library also has the latest versions of programs that should reside on the root; it is used as a reference both for installing and for updating workstations.

Activating a Workstation. As with all systems, it is occasionally necessary to update the software available on a workstation. One limitation of read-only file service such as the RVD system is that this cannot be done while a workstation has attached the system libraries. Of course, it is possible to make the new libraries available and wait until all of the workstations have booted again to start using them. This solution takes too long to work probabilistically and so implies a visit to each workstation to boot it. This, too, is unacceptable.

When not in use, an Athena workstation is in a ‘‘deactivated’’ state. No system libraries are attached and the window system is not running. In place of a *getty* on the console, a program known as *toehold* waits for a keypress from a user who wishes to login. *Toehold* then executes a shell script that attaches the system libraries. If this succeeds, *toehold* starts the X Window System, and a login window appears for the user.

After a user logs out, *toehold* ‘‘deactivates’’ the workstation. This includes detaching any attached filesystems, including the system libraries and filesystems that the user may have attached during his session, ensuring that remote access is impossible, cleaning the temporary storage areas (*e.g., /tmp*), and killing the window system.

Toehold also has an alternate entry: typing control-P (ˆP) on the console allows one to login directly on the console without activating the workstation. This is particularly useful for repairing a workstation that is not working properly.

Logging In. The login process is considerably more complicated now. */bin/login* now includes the following functions:

1. It authenticates the user with *Kerberos*. To limit access on certain workstations, if the file */etc/nocreate* exists, an account is not automatically created; the user must already be listed in */etc/passwd*.

2. It adds entries for the user to */etc/passwd* and */etc/group*, using information from *Hesiod*. This makes the information available to programs that require it for the duration of the session.

3. It attaches the user’s home directory on */mit/<username>*. If the directory is unavailable for some reason (e.g., the appropriate NFS server is down), a temporary home directory is created for the user in */tmp*. In this case, the user is notified of the situation and may choose to abort the login. If the file */etc/noattach* exists, the home directory is not automatically attached, and a local home directory is assumed to exist.

/bin/login then continues with the normal execution of the shell. Note that it does no longer simply *exec()* the shell; it forks before executing the shell so it can perform some cleanup operations when the user logs out. The cleanup includes deletion of *Kerberos* information and detaching the user’s home directory.

As part of the login process, a *Zephyr* windowgram client is started and the *Zephyr* server informed of the login.

Configuration Files. Many UNIX configuration files have been heavily modified for workstation use; some of the most important are described here.

Boot-Time Configuration. As usual, the script */etc/rc* is executed when a workstation boots. The *rc* script on Athena workstations, however, has been extensively reworked to provide a great deal of flexibility. There are actually four files involved: *rc, rc.conf, rc.net*, and *rc.local*.

/etc/rc does most of the work, and it calls each of the other three as required. It first performs disk checks and resets the password file appropriately. It then calls *rc.conf* to obtain configuration information for the workstation and *rc.net* to initialize the network subsystem. Next, it spawns various daemons and further cleans up from the last session, including flushing all connections to file servers (both RVD and NFS). Finally, it calls *rc.local* for any workstation-specific tasks.

rc.conf sets a number of configuration variables for the workstation, including its hostname and network address. These are used to determine which daemons should be run, what configuration should be done, etc. This defines the supported set of differences between workstations and servers, and confines their specification to a single file.

rc.net performs network initialization, including configuring network interfaces, setting default routes, and starting the local nameserver. These functions are isolated in a single file to simplify starting the network in single-user mode; trying to repair a workstation often requires use of some service on the network. A program named *machtype* is used to determine the type of the machine so a single version of the file can be used on all Athena machines.

rc.local is reserved for local configuration on non-public machines; standard server configurations are handled by */etc/rc* itself.

Nameserver Configuration. Because of the *Hesiod* extensions to */etc/named*, an additional standard configuration file has been added: */etc/named.hes*. It contains the names and addresses of the *Hesiod* servers. (Note: the addresses of *Hesiod* servers are included because some software attempts to resolve names with a class ANY query. If the addresses are not present, this will result in a response that does not include the address, and the desired host cannot be contacted.) For local priming of the nameserver cache, the file */etc/named.local* is available. On public workstations it is empty.

Sendmail Configuration. The files *aliases, aliases.dir, aliases.pag, sendmail.cf,* and *sendmail.fc* in */usr/lib* are actually symbolic links into */site/usr/lib* to allow local configuration changes. On public workstations, the aliases files are empty, and the *sendmail.cf* file is a standard Athena version. *Sendmail.cf* is configured only to send mail, not to receive it. In addition, it rewrites ‘‘from’’ addresses to be from *user@ATHENA.MIT.EDU* and rewrites unqualified usernames to be to *user@ATHENA.MIT.EDU*. *Sendmail* does not run as a daemon on Athena workstations; it is started on demand to send mail and periodically by *cron* to attempt to send any queued mail.

/etc/passwd and /etc/group. These files are updated at login time from information supplied by *Hesiod*. When a workstation deactivates, */etc/passwd.local* and */etc/group.local* are copied over these files, undoing any changes. Public workstations also initialize these files from */srvd* at boot time to prohibit other changes.

10. Future Plans

Implementation of the workstation environment at Project Athena is not yet finished. Some known areas of work include the following topics.

Network Error Logging. As the number of workstations and servers grows, it becomes more and more difficult to monitor what is happening. By logging error messages and status information across the network to a central machine, it may be possible to detect abnormal situations before major failures occur. The sheer scale of the system also tends to generate several occurrences of low-probability errors; monitoring them from a central location can help in understanding and solving the problem. Automatic filtering tools will be needed to cope with the volume of messages.

Automatic Software Update and Integrity Check. As noted above, updating workstation software is a major undertaking, since it currently requires a visit to each workstation. It is necessary to develop a reliable means to automatically update software on a workstation. Reliability is the key issue; a software update should not leave a trail of broken workstations. Part of this effort is a software integrity check system that verifies that the configuration and software are correct.

Shift to Vendor Base. As DEC and IBM produce new hardware, it will become more and more necessary for Athena to work with the vendors’ software as well. This is driven partly by device support, and partly by the fact that their diverging systems make it harder to build software from nearly identical source code across different platforms. New applications also require some of the new functionality of the vendor versions. To this end, the Athena developments (e.g., *Kerberos*) must either be absorbed by the vendors or engineered as a layer above the vendor systems. As a side effect, this engineering will enable non-Athena sites to import Athena software easily.

Dynamic Configuration. One last consequence of scale is that the configuration of a workstation should be as dynamic as possible. The names and addresses of nameservers, for example, should not be wired into a configuration file on the workstation; they should be available from a server on the local network when a machine boots. This is one exception to the use of broadcast packets above; a workstation may be permitted to broadcast an initial request for information.

In a similar vein, the Internet address of a workstation could be assigned dynamically. Already one of the most common and most annoying problems that Athena faces is the misconfigured network address. Experiments with assigning the address at boot time are already underway. One immediate application of this is for student-owned workstations: if a student moves a workstation to a different dormitory with a different subnetwork, the workstation requires a different address. The average student, however, cannot be expected to understand how to change the address or how to get a new address assigned by MIT Telecommunications. Maintaining correct workstation name-to-address mappings in a dynamic environment would become much more difficult; the management issues need to be resolved as well.

Scale to 10,000 workstations/1000 servers. Athena's work at MIT has generated a demand for the workstation environment to be used elsewhere on campus. This, coupled with one of Athena's original goals to eventually provide a workstation system for students to own, means that the next few years will see a dramatic increase in the number of workstations in use (and the number of servers required to support them). Scaling up another order of magnitude will require further work as we reach the limits of the work done so far. As one example, a *Hesiod* nameserver currently has an in-core image of over ten megabytes, stretching the limits of our current system configuration. This problem, of course, has a straightforward solution, but it indicates that solutions suitable for the current scale may be insufficient for the next stage.

11. Conclusion

The Athena environment represents a first step in re-engineering UNIX for a distributed workstation system. As large networks become more common, the issues of scale and management will become more and more important. One wizard must suffice for perhaps a thousand workstations.

12. Acknowledgements

The author would like to thank Dan Geer, Kathy Lieben, Jerry Saltzer, and Jennifer Steiner for valuable comments on earlier drafts of this paper. Thanks also to the staff and users of Project Athena, who have supported and endured the construction of this system.

13. References

1. R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, ‘‘Design and Implementation of the Sun Network Filesystem,’’ in *Usenix Conference Proceedings* (Summer, 1985).

2. R. W. Scheifler and J. Gettys, ‘‘The X Window System,’’ *ACM Transactions On Graphics* **5**(2), pp. 79-109 (April, 1987).

3. M. A. Rosenstein, D. E. Geer, and P. J. Levine, ‘‘The Athena Service Management System,’’ in *Usenix Conference Proceedings*, M.I.T. Project Athena (Winter, 1988).

4. J. M. Bloom and K. J. Dunlap, ‘‘A Distributed Name Server for the DARPA Internet,’’ pp. 172-181 in *Usenix Conference Proceedings* (Summer, 1986).

5. Steve Dyer, ‘‘Hesiod,’’ in *Usenix Conference Proceedings*, M.I.T. Project Athena (Winter, 1988).

6. J. G. Steiner, C. Neuman, and J. I. Schiller, ‘‘Kerberos: An Authentication Service for Open Network Systems,’’ in *Usenix Conference Proceedings*, M.I.T. Project Athena (Winter, 1988).

7. M. T. Rose, *Post Office Protocol (revised)*, University of Delaware (1985). (MH internal)

8. Rand Corp., *The Rand Message Handling System: User's Manual*, U.C.I. Dept. of Information & Computer Science, Irvine, California (November, 1985).

9. C. A. DellaFera, M. W. Eichin, R. S. French, D. C. Jedlinsky, J. T. Kohl, and W. E. Sommerfeld, ‘‘The Zephyr Notification System,’’ in *Usenix Conference Proceedings*, M.I.T. Project Athena (Winter, 1988).