Building and securing large scale storage systems for students

Alan Berg, Frank Pinxt, Frank Benneker

Central Computing Services, Universiteit van Amsterdam, The Netherlands.

A.M.Berg@uva.nl, F.M.J.Pinxt@uva.nl, W.F.M.Benneker@uva.nl

Abstract

Storage space within any modern University is set to continue to expand exponentially and faster than Moores law [3]. Without aggressively proactive consolidation projects and centralization of fragmented infrastructure the cost and security implications will have a serious and negative impact.

This paper describes a relevant bread and butter consolidation system which is in development at the University of Amsterdam for 22,000 students and later 6500 members of staff [14]. The emphasis of this paper is two fold. The first emphasis is on the practical impact on the end-user the second is that of a technical backgrounder for interested parties.

Keywords: Consolidation, Storage, Samba.

1 Introduction

Storage space explosion is driven by two main factors. The first factor is that the density of information stored on a hard disk is increasing over time. The second normally more neglected factor is that sales of hard drives are also rapidly increasing. The practical effect for infrastructure management is that storage space is becoming larger, cheaper and prevalent. When the main author was still studying in the 1990's a 20 MB hard drive was considered a luxury. These days if you buy a machine with less than 200 GB then you are buying a very cheap machine indeed. The University of Amsterdam's (UvA) infrastructure includes servers, personal computers and not so many thin clients. Thin clients also represent consolidation by centralization, but these services are outside the bounds of this paper. Backing up and maintaining of storage if it occurs at all is very selective. Any central project can only improve this situation at significantly lower costs. One should note especially the hidden costs like the loss of a thesis or months work of a member of staff, hence the motivation for the UvA U-drive project.

The U-drive project is an initiative from the Central Computing Services, [11] at UvA to help with the containing and controlling of storage space explosion and the enhancement of reliability and safety of stored content. The main idea is to give each student an easily accessible storage area from anywhere that will act as a potential repository of there cherished work that is stable, fast and restorable. A noticeable issue for students is the geographic dispersal of the UvA infrastructure. UvA has grown over many centuries in an ad-hoc way, buying property when and where it can and this produces a scatological map. Students have a tendency to store data using the so called "sneaker net", floppies in pockets. We want therefore to significantly lower the student effort to use a central repository. The basic requirements state that the system must be easy to use within and without the campus network and highly available and just as importantly secure and measurably secure. The technologies used should follow open standards and the source code open to review and thus open source [4,5,6]. As with many projects 90% of the end product is achievable with 10% of the work. In the current incarnation building the file storage system based on a central Storage Area Network (SAN) and a farm of file servers, using SAMBA [12] a popular and open source project, in this case required relatively little effort. A universal view of the storage space is generated by a Distributed File System (DFS) [13] server also available as part of the SAMBA solution.

There was temptation within the project to place the link to the storage under the main UvA learning system, Blackboard 6 [9]. However this temptation was resisted due to the extra complexities involved, potential cost in performance on the learning system and the context that the end user finds themselves in. If this context was used it would be hard to give out ad-hoc storage space later to new groups of users without giving direct access to the learning environment. It is true that 12,000 students regularly use the system, but that leaves 10,000 that did not.

2 End User Expectations

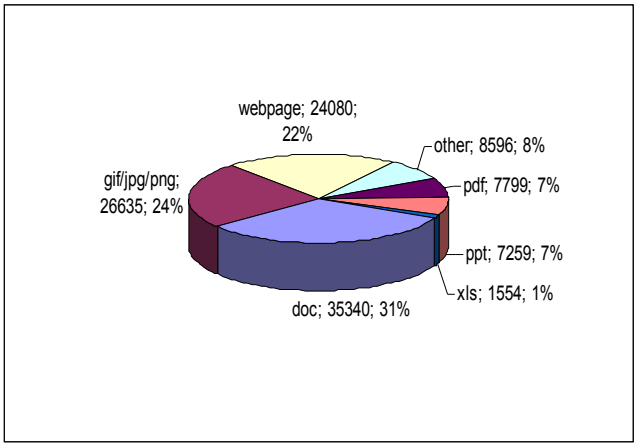
2.1 Overview

After consultations with the interest groups; student, management and administration the following demands were enumerated: The production system is required to be:

- **Accessible** from in and outside the University Network. This is not so simple a mater as packet filtering occurs at the parameter so a number of well known ports are shut, including the ports related to file sharing.
- **Easier** to use than with an FTP client. Preferably with a drag and drop enabled file management support with a severe requirement that no extra installation of software should be needed on the target clients.
- **Consistent and predictable:** the behaviour should be consistent no matter which computer OS you are using and where you are using it from. Interoperability issues should not have any first or second order effects.
- **Simple to administrate**: This is another way of stating that the provisioning of user accounts and quota's needs to be automatic or the sheer number of potential users will overwhelm administrative capacities. If needed extra infrastructure within the UvA directory services may be built.
- **Relatively cheap**: The cost per user needs to be within the reach of a middle sized project and expand linearly with demand.
- **Responsive**: The system needs to have very little latency. The delay in time between dragging and dropping a file from one directory to another and the occurrence of the related event in real life on the file server. Luckily bandwidth within the Universities network is ample and outside the network the trend towards adoption of broadband connections I is improving the average users experience with time.
- **Redundant**: If a failure occurs within any critical section of the system, the system should remain running and no disturbance should be felt by the end user. And in a worst case scenario a failure should be quickly spotted and easily replaceable. A so called "poor mans solution".
- **Scalable:** If another 10,000 or 100,000 users are added it should just be a question of buying in components and not that of changing the overall design to keep things smoothly running.

2.2 Hints from File Usage Statistics

Another method to gauge the behaviour and desires of users is to look at usage patterns on other similar systems. One example of a similar system is the Blackboard learning environment. At the University of Amsterdam 12,000 students regularly use this online environment. The corpus comprises of 112,000 files contained within a mere 28 GB of storage space. Figure 1 shows the absolute number of files per type. This clearly states that word documents are the primary type. Taking this analysis further it can be seen that around 60% of all files are native to the Microsoft Office suite. In terms of storage space usage PowerPoint presentations are dominant and take up 46% of the used storage. PDF files are just as popular as PowerPoint, if not slightly more so, but are more efficiently compact. All in all office formats dominate.

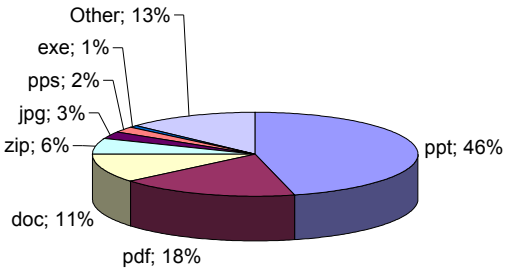


Type	Number of files
Doc	**35340**
gif/jpg/png	26635
Webpage	24080
Other	8596
Pdf	7799
Ppt	7259
Xls	1554

Figure 1: Content distribution for the BlackBoard learning environment.

Native Office	68233 (61%)
Other	43030 (39%)

Tabel 1: Microsoft Office files types vs. the rest.



Type	Size (%)
Ppt	**46%**
Pdf	18%
Doc	11%
Zip	6%
Jpg	3%
Pps	2%
Exe	1%
Other	13%

Figure 2 and included table: File type vs. Storage space allocation for the Blackboard learning environment.

The authors recognize that the file storage system for a learning environment is somewhat different to the general storage of a student/staff system. The learning system is about delivering relevant didactic content through the Internet and therefore less specifically orientated. However the statistics gives significant hints. The dominance of the Office format has the potential to make the system architects life easier for coping with file management outside the UvA perimeter. Office and Internet explorer are especially coded to enable what Microsoft marketing term "web folders". Web folders allow you to connect to a website that can talk a particular language over http (the language of web browsers talking to web servers) named WebDav [2] and drag and drop files from and to that website from within Office or the web browser. The advantage of WebDav is that is an open standard, has many clients that support it, works well with Microsoft Office and is a Industrial Standard. WebDAV itself being an open standard is also used by many other applications and clients. The main area of concern is that interoperability between different WebDav implementations. Even within the proprietary Windows family there are issues and more than a few security patches. We therefore balanced the risk benefit ratio by frequent testing of this particular methodology and left the choice of its inclusion to the latest possible moment. In the end the call for the drag and drop functionality was overwhelming. It was too simply to us to ignore.

3 System Definition

3.1 Overview

This section will describe the underlying workings of the storage system.

The system is built out of a number of components that when placed together generate a consistent and reliable and relatively cheap infrastructure. The initial design is shown in diagram three next. The two situations that the student may find themselves in are; working within the University network and working outside in the big bad and uncensored cloud that is the Internet in general. These two stories will be expanded on in this section via collaboration diagrams and descriptions of the players involved. One should also note the obvious that the inner network is quite large and also not particularly secure.

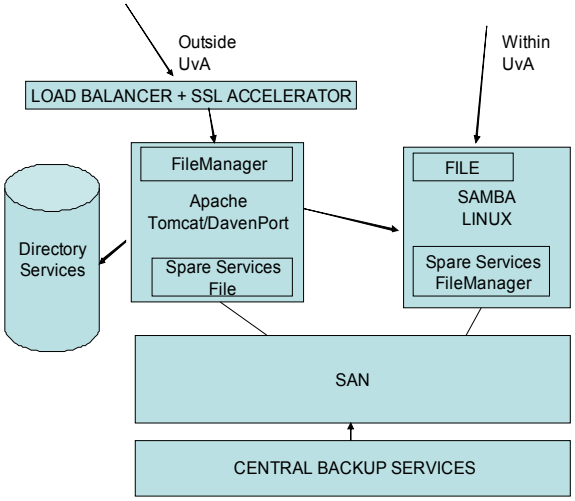


Figure 3: A simplified overview of the Consolidation infrastructure.

At the beginning during the proof of concept stage of the design process a serious philosophical question occurred. Should we use open source software or proprietary software? After performing a detailed analysis, the conclusion was that both software development models could fulfil our expectations. This is an important point. You could argue validly in terms of functionality, ease of use or reliability for a Windows 2003 service or SAMBA/LINUX. It therefore comes down to detail and preferences. The Central Computing Services has vast experience with both types of systems. But since the economic down turn in Holland the climate is orientated towards total cost of ownership. The perceived negative for open source is support, but our development group decided that due to the large and active open source community this was a minimal risk. The perceived negative to proprietary software is vendor locking. The Total Cost of Ownership (TCO) arguments can go both ways. The main assumption about the hidden costs in open source installations is that of training, since we already had in

house talent. The second argument against was the risk of IP infringement. The authors believe that IP infringement is a ghost issue for mainstream and well known software. Therefore one would expect that TCO is lower for open source and in this time of economic down turn this was a strong driving factor.

The next subsections will zoom in on the details of design.

3.2 Samba and DFS

3.2.1 Description

DFS stands for Distributed File System and is a method of creating a logical view of file structure out of a series of file systems. The exact naming conventions are not important, but what can be achieved with this approach is. Figure 4 displays an example. Before DFS you have a series of file servers that exist without relationship and after you no longer see the machines hostname only the location under a logically inverted tree. Students are no longer bothered by too much infrastructure detail. They, the student, only need to know there position in the organization.

A further benefit is that under the SAMBA implementation of DFS each node may contain one or more servers allowing for allocation per client of separate servers, a primitive load balancing event.

A considerable advantage of this setup can be seen during migration of servers to or away from the structure. It is just a case of changing the configuration. The end user does not have to be interrupted in the use of the namespace. What is also importantly for the back office is the potential size of the tree. For the SAMBA implementation the tree structure is stored as links in the file system and not within a database. This scales to many millions of nodes if required, more than enough for each user and all there personal gadgets and IP addresses. You never know this type of complexity and detail to the tree structure may become a requirement in the future.

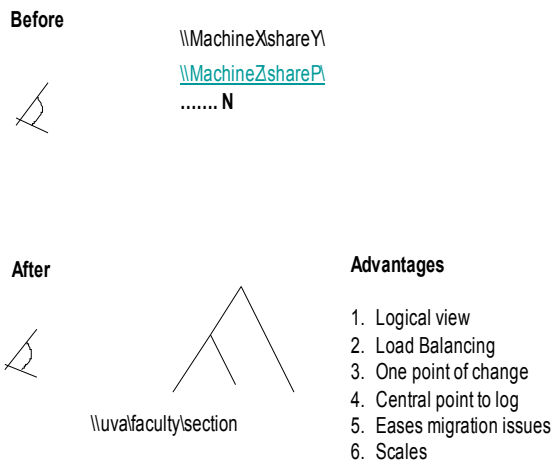


Figure 4: Describes how DFS brings a logical structure to a potentially disjointed physical infrastructure.

3.2.2 Players

In figure 5 you will see the collaboration between the various players that are mentioned here for DFS to function.

Windows Client: This is the network driver that lives within the users OS and translates file actions into requests to the SAMBA server. The client can also be an application like Office or Internet explorer or Goliath for Macs. The authors name it a Windows client as most students will be interacting via Windows.

Samba DFS server: Is responsible for translating requests for a logical part of a tree to a physical location of a relevant file server.
E.G: \\uva\faculty\department\uid=> \\hostX\areaY\uid

Disk: The local disc of the DFS server that contains the links between the logical and physical view. This may seem a small point, but has important implications that will be discussed later.

Samba File Server: The server that is responsible for communicating via the correct protocol with the client. It is a separate entity to the DFS server to ensure reliability

SAN: Storage Area Network., the point of consolidation. Please note in the future we may be not just using one SAN but a series in combination with other types of storage, which depends on how storage technologies evolve in the market place.

3.2.3 Collaborations

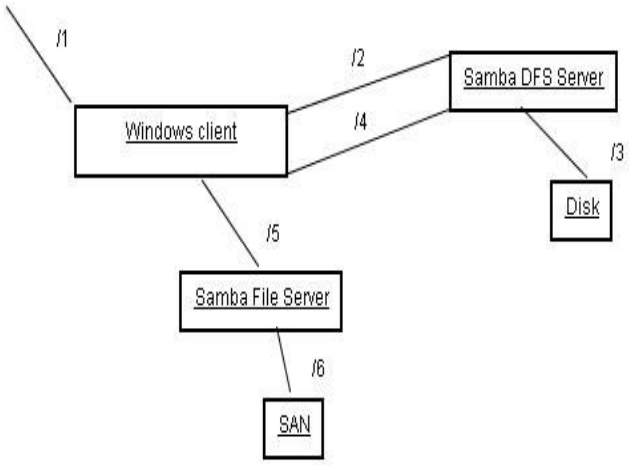


Figure 5: A collaboration diagram giving a simplified idea of the interactions involved between the various actors.

/1 the user asks for a resource within a tree structure via their standard interface be it the Windows GUI or the command line (e.g. the net command in Windows).

/2 The file driver passes the request to the DFS server.

/3 The DFS server looks at the file structure on the local disc.

/4 Sends a redirect request to the windows client.

/5 The client asks the file server for a resource.

/6 The resource is retrieved from the SAN when required.

3.3 File managers and WebDav

3.3.1 Description

The most difficult part of the design was exposing the file system to the Internet with secure and scalable technologies. The requirements were achieved via two methods: First is the use of the WebDAV protocol. The protocol allows for basic manipulation of files and revision control. The Web part of WebDAV explains that files are reachable over the internet via the HTTP protocol. The DAV part stands for Distributed Authoring and Versioning and implies that groups of people may work on the same document with a lockable infrastructure. An example of a client is that of Microsoft Office 2000, XP, 2003 ™. Office on installation in a windows environment enables the use of so called web folders that allow drag and drop functionality for uploading and downloading files. This is a highly attractive feature that requires much less skill from the average student than using an FTP client and with its potential security issues. The second method is the encryption of sessions so that passwords are never exposed in plaintext on the Internet. This is achieved by the enabling of the SSL protocol. This issue with this is that encryption costs cpu effort and thus impacts in a negative way on the resources of the server that delivers the session. The cost of exposing the File system to the Internet in a secure and student usable way requires the greatest degree of effort and concentration from the design team. In terms of system resources the use of encrypted sessions via the SSL protocol can be debilitating. Only the use of extra hardware such as hardware SSL accelerators can bridge the gap with expectations. The use of hardware SSL accelerators and distribution of requests via load balancing over a number of relatively cheap Apache web servers allows for the required scalability at the right price. In fact the hardware acceleration takes place on the Load Balancer which can act to change an HTTPS stream to an HTTP stream and vice versa.

Till date interoperability issues with the WebDAV protocol and various clients and server combinations give the system developers cause for concern. The difference between the hype and the reality is sadly still quite large. However, with vigorous testing and a few clever tricks and open publishing of issues the authors expects this 'interoperability concern' to be alleviated. The authors expectations is that due to severe competition in the commercial market place that many of the interoperability issues will soon be cleaned up and hopefully by the time you have read this paper the reality has considerably improved.

3.3.2 Players

Web browser: The user GUI, acting as a rendering point for the drawing of the structure of the file system and a means to communicate with the underlying infrastructure.

Load Balancer: Responsible for choosing one of a number of web servers from a farm and passing on of the request and if needed changing an HTTPS stream to an HTTP one.

Apache Server: Responsible for the direct delivery of static web pages and the passing on of request to the file manager.

Fie Manager: the file manager which is written in Java and is conforming to the servlet specification. This relies on an Apache/Tomcat infrastructure. Apache being way the most dominant web server used on the Internet one would expect this to be a good choice.

File Server: The SAMBA infrastructure as mentioned previously.

3.3.3 Collaborations

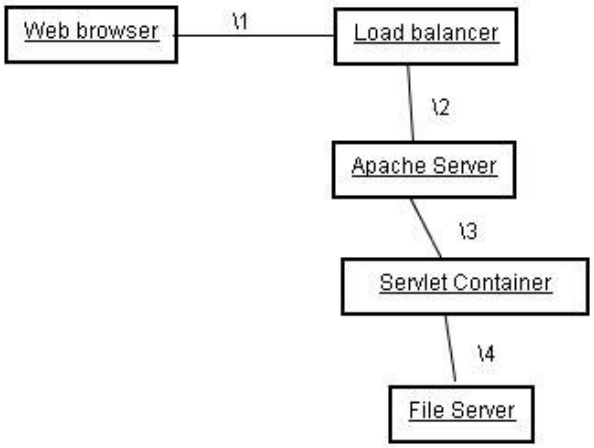


Figure 6: A collaboration diagram giving a simplified idea of the interactions involved between the various actors.

\1 The user drags and drops a file from one folder to another. The web browser sends a request off to a relevant host, if required communicating via HTTPS.

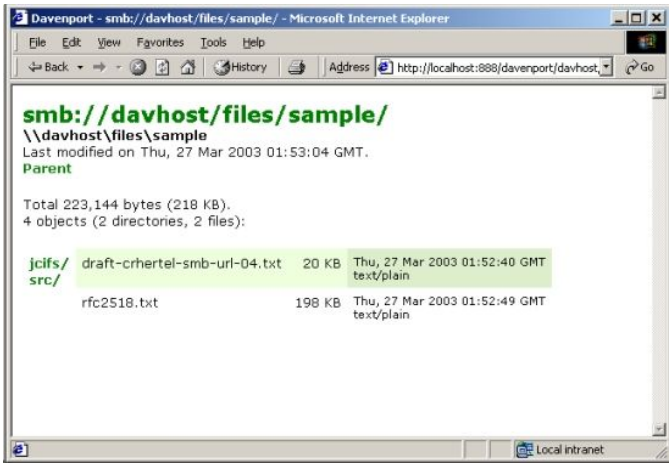
\2 The Load balancer chooses the relevant load balanced web server, decrypting on the fly an HTTPS stream to plain HTTP if required,

\3 A web server decides if the request is for static web pages or for the file manager. If it is for the file manager then the request is passed on through a private connector to a container.

\4 A servlet translates the WebDAV instructions into Windows native CIF language and communicated directly with the SAMBA server as mentioned in the DFS sub section. Returning any required resources via WebDAV back to the Apache server, load balancer and last the browser via https.

3.3.4 The File Manager

From the end users perspective there is an intimate relationship between usability and the look and feel of the file manager. Students are used to the cut and paste, and the drag and drop paradigm. It becomes therefore crucial to overall acceptability that this part of the infrastructure is fluent and this does not require additional installation of software. After extensive market the Davenport servlet [10] was chosen. This is a prime example of Open source software. All code is viewable written in Java and highly structured and understandable. The look and feel can be modified via an XML document and uses software libraries written by the SAMBA team themselves. The next screen grabs the standard view as soon as you first navigate the file structure. Once you click on a file the view transforms itself into the one shown in the second screen grab. Drag and drop is now enabled. Behind the screens the file manager is translating between WebDAV and the CIFS protocol that the Samba server speaks. This costs system effort, but the end result is worthy of this.



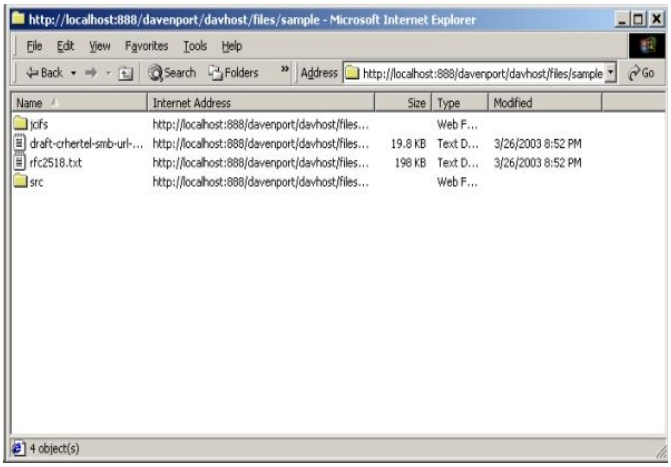


Figure 7, 8: Shows screen grabs of the Davenport WebDAV CIF's gateway. The first shows a standard tree structure via links. The second shows that on clicking of a link the drag and drop environment is rendered by Internet Explorer version 6.

Other contenders for the client throne were taken into account and excluded later. Secure file transfer was a serious contender but failed due to the incontrovertible problem of needing to install a client. WebDav client software in windows is installed by default as a file redirector in Windows XP or for Windows in general when Internet Explorer is installed or Office.

We also explored the option of a plain old file manager as a servlet. Test Java code was written that formed the basis of such an interface. However though it fulfilled the functionality it was not as pleasant as WebDAV.

The authors do not expect the file manager to be applied. However, one can consider a situation were WebDAV is proven to be unreliable for a number of target clients and the file manager can at that moment act as a backup method.

3.4 Authentication

The University of Amsterdam's directory services [8] are the binding force behind all significant large scale systems that are open to the student, within its administrative authority. Over time the infrastructure will change taking over single sign on functionality. At present the directory services store a limited amount of representational information for each student and valid member of staff. The information is exported via scripts from the ISIS user database to an LDAP infrastructure. Further scripts synchronize this information with a Microsoft Active Directory forest. Passwords are synchronized two ways between LDAP and AD via a Netscape extension API for the LDAP servers and password filter plugin for AD. The directory services are shown pictorially in figure 9.

To the end user this detail is irrelevant and no doubt will evolve within the foreseeable future. It is already common practice in the general marketplace for directory services to be driven by Metadirectories of other such products.

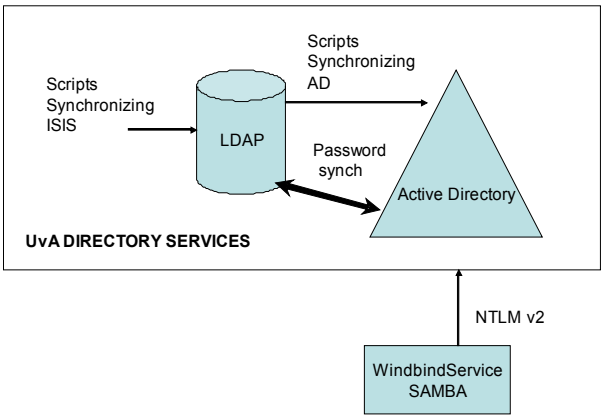


Figure 9: An idealized view of the UvA Directory services.

The SAMBA server takes advantage of the directory infrastructure to verify the password of the user. The included Windbind service communicates via the NTLM version 2 protocols with Active Directory.

Windbind is not the only possible solution. Pluggable Authentication Modules [7] using LDAP, Kerberos or even NIS (with minor changes to the infrastructure) is viable. Windbind was chosen for its relative simplicity and minimal impact on the UvA directory services way of doing business.

3.5 Scalability

In short, scalability within the current design is via a combination of factors.

- **Load Balancer:** Scaling out, under the current configuration WebDav traffic is passed through a load balancer; the load balancer can choose between any numbers of web servers and remember which session belongs to which client and server. In load balancer terminology sticky sessions are enabled. Therefore if a particular file manager is under too much load we can simply add extra Apache/Tomcat servers and change the configuration of the balancer.
- **DFS Server:** The server allows for poor man load balancing on the tree nodes, but more importantly hides the physical infrastructure behind a logical presentation. Hence infrastructure migration disturbance can be hidden.
- **SAN:** Storage Area Networks are extremely reliable and highly scalable. If we need more storage we can add more sectors or extra SAN's.
- **Provisioning:** When the user first logs onto the file system, the relevant SAMBA server runs a set of provisioning scripts that set up file structure, quota etc for the user. This removes administration burden. The scripts have been deliberately kept to a minimum to ensure smooth and reliable functioning. However in the future it is the author's expectation that the scripts will increase in complexity and add such functionality as local administrator notification and acknowledgement e-mails.

3.6 Security

No ship is unsinkable, no security is perfect. The only perfect system is the one that is not turned on. However there are some serious advantages to consolidation within this arena.

- **Organizational boundaries**: From the authors experiences it is clear that many security issues occur when a system is the responsibility of more than one set of administrators. In theory this cannot and does not happen. In practice systems are not attended when one group of administrators believes mistakenly that another group is in control. In a large diffuse structure as UvA there are many chances for this to happen before consolidation.
- **Patch management:** Security patches come out very regularly indeed. A centralized service is much easier to patch than a decentred and thus better secured against known issues.
- **Monitoring:** Specific tools, for example tripwire, work better for monitoring if central. The number of nodes is less and more effort can be concentrated.
- **Lowering of the use of sneaker net:** Floppies are bad, and a good method of spreading viruses and Trojans and the rest of the mangier of system violators. If data can be transferred via drag and drop between the systems that students used with little learning curve then sneaker net usage is no longer necessary.
- **Physical security:** All large scale central services are physically secured. Doors, locks, security guards, fences. No unauthorized person can walk into the relevant room and stick a floppy into a drive.
- **Fewer Servers:** This translates into less ports and IP addresses to attack.
- **Highest protocol levels:** File sharing uses NTLM version 2 protocol passwords are sent over SSL and not plain HTTP. These details make it more difficult for crackers and script children from getting their dirty little fingers on the underlying systems.
- **Specialization:** Central services focus on the task at hand. Have well defined organization and ways of doing business that are specialized for this type of service. Experience accumulates over time improving quality.

Two arguments that are Performa applied against centralization are: the classic “all eggs in one basket” scenario. If the central store is compromised then all that is in that store is also. The second is that of local knowledge. A central service will personalize to the generic needs of an average student. This definition may be subtly different. Of the two arguments the second has the least basis. File storage is a common factor for all students.

When calculating the end result of the various pressures mentioned one can only come to the conclusion that central, if sensitively exposed, is much more secure and verifiably secure than that which was previously found. This project can only improve the historic structure.

3.7 Virus Checkers

There are a reasonable number of virus checkers available for Linux and even a few specially made to take advantage of the file system abstractions that SAMBA provides [1], so there is no limitation of choice. The generic question is one of system load. Careful consideration is required with the load which is placed on the system from the virus checkers in question. We can perform virus scans either immediately on request or as a scheduled task at a low load period. This is shown in Figure 8. Two issues that play a part in the decision are: (1) the load at peak time may push the system into an unresponsive phase transition. (2) Latency due to an extra delta t on response time may diminish the user experience. It is also possible to use more than one virus checker and then enable one in real time and the other scheduled this gives a greater coverage of potential threats, but this increases complexity and load. Time and experience of the live system will decide the final configuration details for the development team.

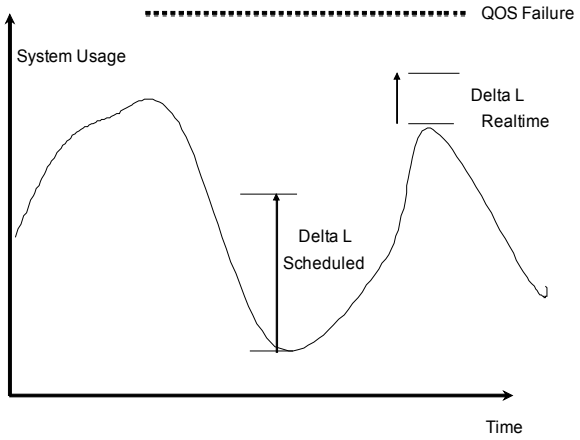


Figure 8: System usage against time showing the effect of load due to real time and scheduled virus checking.

4 End User Experience

A small number of end user tests were performed to date and a collection of surveys filled and analysed. The main reoccurring theme from the trail was the happiness of the end user with the functionality. The end user did not notice any meaningful quantitative difference in the SAMBA/Linux services than that that was expected from a Microsoft Windows only solution. In fact small details of improvement were noted. One example comes to mind that of quota reporting. On the mildly negative side it was found that the primary issue was the interoperability of WebDAV as instanced by the DavenPort servlet. WebDAV mostly worked, but on repeatable occasions WebDAV had certain issues with certain clients. It was noticeable though that this situation was improving over time, the reason for the improvement trend, being market demand and volume.

5 Summary and Conclusions

This paper describes a campus file system for all students and later potentially for all staff that is securely exposed to the Internet. The system is achievable, but requires detailed inspection.

The authors expect that the system to be relatively cheap, reliable and secure when compared with the more chaotic and distributed form one finds at present.

Minor issues were found in the use of web folders as a means of doing work. The authors hope for a time when the interoperability issues that exist with WebDav, the underlying protocol used by web folders, to be defined out of existence by the Request for Comment process of standardization and via commonsense reining in in the marketplace.

In summary UvA recognizes the risk of run away costs due to distributed storage capacity and is now proactively running an example consolidation project that can easily be plug and played into later by other file dominated systems. Without such projects storage costs cannot be contained. This project will improve the end experience for the student and hopefully diminish the use of the sneaker net.

On the sometimes heated debate over proprietary or open source software the authors would like to clearly state that Open source is a viable approach to building stable large scale systems. In particular the use of SAMBA/Linux eases many design issues.

Acknowledgements

The authors would like to acknowledge the Open Source Community for such excellent products as SAMBA and DavenPort and Jim Mintha who is the energy of construction behind the building of this system.

References

[1] Alexander Bokovoy. “Virtual File systems”. CIFS 2003. conference.
”http://www.cifs2003.org/conference/program/bokrev1.pdf”

[2] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, “HTTP Extensions for Distributed Authoring – WEBDAV” Request for Comments: 2518
”http://www.ietf.org/rfc/rfc2518.txt”

[3] Gordon E. Moore. “The experts look ahead”. *Electronics*, Volume 38, Number 8, April 19, 1965
”ftp://download.intel.com/research/silicon/moorespaper.pdf”

[4] Bruce Perens. “Open Standards Definition”.
”http://perens.com/OpenStandards/Definition.html”

[5] Brue Perens. “Open Source Definition“.
”http://perens.com/OSD.html”

[6] Alex Reid, David Glance. “Issues in the use of Open Source Software”. EUNIS 2003 Proceedings Book page 474-479

[7] The V. Samar and R. Schemers (SunSoft), ``UNIFIED LOGIN WITH PLUGGABLE AUTHENTICATION MODULES", Open Software Foundation Request For Comments 86.0, October 1995.
“http://www.kernel.org/pub/linux/libs/pam/pre/doc/rfc86.0.txt.gz”

[8] J. van Zuijlen and M. Vandecappelle. “Central Authentication service Universiteit van Amsterdam”. EUNIS 2003 Proceedings Book, page 54-62

[9] Blackboard homepage. “http://www.blackboard.com”

[10] Davenport Open source project.
“http://sourceforge.net/projects/davenport/”

[11] Informatersingscentrum at the University of Amsterdam
http://www.ic.uva.nl/organisatie/

[12] Samba homepage. “http://www.samba.org”

[13] Microsoft Corporation “Simplifying Infrastructure Complexity with the Windows Distributed File System”
”http://www.microsoft.com/windowsserver2003/techinfo/overview/dfs.mspx”

[14] University of Amsterdam “http://www.uva.nl”