Shun Pillay and Alexander Holt
School of Computer Science
University of the Witwatersrand, Johannesburg
10 May 2006

# Teaching Mathematical Sciences with a Linux-based Laboratory

**Abstract**

*Six years ago, part of our university created a large computer laboratory based on diskless PCs running Linux. At the time, it adequately met our needs. However, evolving requirements from lecturers and students, along with growing technical obstacles, have led us to make major changes in its organization. We now maintain an undergraduate computer laboratory of 200 Linux client machines, servicing up to 2200 students. The current laboratory is robust, scalable, easily administered and cost-effective while meeting our contemporary educational needs. As well as demonstrating the maturity of Linux on the desktop, our experience offers lessons in dealing with technical problems, in choice of Linux distribution, and in overall management of the laboratory.*

## 1 Introduction

Our computer laboratory supports the schools which make up Mathematical Sciences at our university, namely Computational and Applied Mathematics, Computer Science, Mathematics, and Statistics and Actuarial Science. It offers 200 desktops, used by up to 2200 undergraduate students (years one, two and three).

McGlashan [McGlashan, 2000] describes this laboratory as it was configured and maintained six years ago. What made the laboratory interesting then was that in order to satisfy our teaching, learning, and computing requirements we had to do things somewhat differently to trends in the Linux world at the time.

Since then, requirements within the laboratory have become more numerous and varied. The situation now, six years later, is that we find ourselves increasingly adhering to common practice in Linux-based laboratories, in order to best meet our requirements. One of the reasons for this is clearly the maturity of Linux as a computing platform for both servers and desktops.

By exploiting the new strengths of modern Linux distributions, taking advantage of reducing hardware costs, and managing our technical staff in new ways, we have been able to satisfy our requirements with, we believe, sufficient headroom for future growth.

This article explains how the teaching and computing environment has changed in our laboratory since 2000. We summarize the old setup, describe the problems that emerged, and show how our new arrangement deals with them. We aim to provide sufficient technical detail to enable others to replicate the key aspects of our configuration. We also briefly discuss how what we do contrasts with a Microsoft-only laboratory. Finally, we note currently unresolved problems, and look at where we expect our laboratory to be in the near future.

## 2 The old laboratory

The approach we took in 2000 involved three main servers and a laboratory of diskless clients. A desktop-class PC running Red Hat Linux served as the server which handled diskless booting of Linux clients, the Linux filespace, authentication, and applications. A server-class PC running Novell NetWare was used for the MS-DOS equivalent. A third Red Hat server was used to host Linux versions of non-standard and proprietary applications such as MATLAB, Mathematica and SAS.

## 3 Technical problems

Initially, things worked reasonably well. The laboratory, however, eventually became constrained by the following problems:

T1. *Increasing demand for Windows 2000.* A portion of the laboratory dual-booted into MS-DOS, but as use of this diminished, the demand for a more recent Microsoft environment—in the form of Windows 2000—increased.

T2. *KDE- or GNOME-based desktops not feasible*. Linux offers both KDE and GNOME as rich, graphical desktop environments. But our diskless approach meant that as much as 300MB of programs and data needed to be sent over the network to each client every time a student logged in. It transpired that this led to unacceptable response times, especially at the start of a lab session.

The immediate solution was to adopt the lightweight window manager FVWM in place of KDE or GNOME. But not supporting a more user-friendly interface damaged the laboratory's reputation among students.

T3. *Increasingly complex server arrangement.* We ended up using five separate servers, four of which distributed the server load during boot-ups or periods of great demand for applications across the network. This made boot-up times manageable, and the use of a modern desktop environment such as KDE started to become feasible. However, these additional servers caused a maintenance overhead.

T4. *Unacceptable load times for large applications*. Non-standard or proprietary Linux software was stored on a remote server which had relevant directories mounted on the clients via NFS. The program executables on this server were large. Having the software in just one place made maintaining these packages easier, but start-up times during busy classes became prohibitively slow due to excessive network traffic and load on the server.

T5. *The labs became increasingly heterogeneous.* Incrementally expanding our laboratory every year resulted in a large variety of hardware. We needed a separate boot-image on the server for each hardware combination; eventually the need to easily support network-booting dominated procurement decisions, at the expense of optimum value for money.

T6. *No performance gain from new hardware.* When older PCs broke and were replaced with new models, problems with boot-up times and memory limitations persisted because the maximum throughput of the servers, the capacity of the network, and excessive RAM usage on the diskless clients (leading to swapping over the network when physical RAM was exhausted) remained the key bottlenecks.

T7. *Limited software support.* The software that we could support easily or on short notice was largely limited to software pre-packaged for Red Hat Linux. Very often we would be required to support other packages. These had to be custom-compiled from source to satisfy the idiosyncrasies of our system.

T8. *Server redundancy expensive.* Keeping spare servers on stand-by or adding devices for redundancy would have been prohibitively expensive. This was particularly true of the server-class PCs required to manage MS-DOS via NetWare.

T9. *SCSI drives required.* The diskless environment generated high loads on the disk subsystems of our servers, making it necessary to equip them with fast SCSI (rather than IDE) drives. These drives (and controller cards) were expensive and making the controllers work well with Linux could also require substantial effort. A further consequence was limited disk space, which led to disk quotas for students.

T10. *Security patches, bug fixes and upgrades largely ignored.* Continuously rebuilding the system from scratch to keep up with newer Linux distributions involved more work than we could spare. As a result, our operating system software became outdated, meaning that patches and bug fixes were either hard to obtain, or required special work to integrate into our non-standard configuration. Requests for new versions of software often could not be satisfied because of software dependencies incompatible with our system.

## 4 Changing educational requirements

Since 2000, a number of changes in the needs of both lecturers and students added further pressure to reorganize the way our laboratory functioned. On top of the specific issues noted here, student numbers increased from 1200 to 2200 between 2000 and 2005.

E1. *Software deployment at short notice.* We noticed increased demand for the rapid installation of some new packages. Often the use of such packages is short-lived, making investment in proprietary alternatives infeasible. Searching for and preparing such software consumed a large portion of our technical staff’s time.

E2. *Increased demand for hands-on assistance.* It seems this arises partly from a growing awareness among students that they are paying for the use of our laboratory, and therefore expect adequate service for their money. An additional factor is students who are highly focused on completing the educational tasks asked of them, and are not interested in spending much time on developing fundamental computer skills. These students prefer having someone help them get up and running (particularly with shorter courses).

E3. *More students with PC/Linux expertise.* Such students come to the laboratory with higher expectations. It's important for us not to disappoint them because their perceptions of quality of service spread quickly in our environment.

E4. *Many first year students have never used a computer.* This is a continuing trend. Special attention needs to be given to these students as early as possible to avoid them falling behind, and being discouraged from computing-oriented studies.

E5. *Increased demand for local language support.* While English is the language of instruction at Wits, many students interact using other languages. We would like to support this diversity in our computer laboratory.

E6. *Increased use of home PCs.* Many students choose to complement their time in the laboratory with additional study at home using their own computers. This often requires that the student have access to the same software set at home. If only proprietary software is used in the laboratory, this scenario becomes expensive for the student.

E7. *Increased demand for software not requested by lecturers.* We attribute this to the increased number of computer-aware students entering the laboratory. It is particularly evident among students who enjoy experimenting with the system. Since these activities are secondary to our main educational goals, technical staff cannot afford to spend much time on these requests.

E8. *Familiarity and ease-of-use critically important.* To encourage students to focus on the specific educational tasks at hand, lecturers now strongly prefer having as few as possible unrelated technical hurdles that the student needs to overcome.

## 5 The new laboratory environment

The key change was to exploit the much reduced price of hard disks by doing away with the diskless approach entirely. (We now also do backups using hard disks rather than tape drives.)

We now have only one main server, which handles both the majority Linux environment and the subset of machines that also dual-boot into Windows XP.

We moved from Red Hat Linux to Ubuntu Linux, and we maintain our own mirror of the Ubuntu software repository, giving us access on demand to a vast collection of software. (We also gain immunity from any difficulty in accessing the official Ubuntu archives, such as problems with international internet connectivity.)

We install and maintain software using the *apt* package manager, a standard feature of Ubuntu. This includes the proprietary software that we own licences for (it was fairly easy to repackage Mathematica, MATLAB and SAS in the right format for *apt*). The two strongest features of *apt* are its ability to automatically satisfy software dependencies, and to communicate with a software repository via HTTP.

We have also increased the use of student assistants, using timetables that are synchronised with students that use our laboratory. These assistants help students with basic computer competency, and also assist technical staff with hardware maintenance.

Today we feel our laboratory both resolves previous technical constraints, and meets our central educational requirements, as the following sections discuss.

## 6 Overcoming the technical issues

We've dealt with the technical obstacles as follows:

T1. *Increasing demand for Windows 2000.* By using Samba on our Linux server, we were able to make the transition to Windows 2000 (and later to Windows XP) without an investment in expensive Windows Server licences.

T2. *KDE- or GNOME-based desktops not feasible.* Using local hard disks eliminated the network load problem and allowed us to adopt Ubuntu's default GNOME desktop as standard.

T3. *Increasingly complex server arrangement.* Samba helped us eliminate the need for a server dedicated to MS-DOS or Windows. The use of local hard disks on our client machines removed the need for boot-servers, and for a separate server for proprietary executables. Our server count was reduced from five to one.

T4. *Unacceptable load times for large applications.* Problem eliminated by installing all software on local disk.

T5. *The labs became increasingly heterogeneous.* Hardware diversity has now become the problem of the Ubuntu installer—which copes with it admirably.

T6. *No performance gain from new hardware.* The causes of this were the maximum throughput of our servers, the capacity of our network and excessive RAM usage on the client machines. All of these bottlenecks became insignificant through the use of local hard disks. Our network traffic has decreased, and students are enjoying a more comfortable computing experience by not having to worry about overusing memory resources.

T7. *Limited software support.* Today, we are still limited by the Ubuntu software repositories, but these are very much larger and more comprehensive than what was once available for Red Hat (and still considerably larger than the main Fedora repositories, the free successor to Red Hat). We now need to compile software from scratch very rarely; when we do, it involves little customization because all of our machines have standard Ubuntu installations.

T8. *Server redundancy expensive.* We have entirely moved away from expensive server-class machines since modern desktop PCs adequately serve our needs, even as servers. The reduced costs have allowed us to instead budget for redundancy at various levels. For example, we have a second server which is a clone of our main server and is kept on standby. This level of redundancy was not possible six years ago; we could not afford to invest in a high performance server and just have it lying around waiting for something to go wrong.

T9. *SCSI drives required.* Today, standard SATA hard disks perform adequately. They don't require any additional controller cards. SATA disks typically work out-of-the-box with Linux. With the low cost of disk storage, we no longer find it necessary to use disk quotas; even with 2200 users, we find that within a year, we are unlikely to exceed the 250GB disk space on our server.

T10. *Security patches, bug fixes and upgrades largely ignored.* Security patches and fixes are automatically installed from our Ubuntu software archive. It is now possible to upgrade to entirely new software versions without adversely disrupting the underlying installation. In fact, it is also possible to seamlessly upgrade from one version of Ubuntu to a later version without the need for a re-install. Our policy has therefore become to move to the latest version of Ubuntu at the end of every academic year, which is easily realised.

It is worth noting that in moving away from our diskless approach we did lose one benefit, namely being able to easily distribute software to all clients and ensure that the set of software on each client was identical. Today we achieve these goals with simple scripts which automate installation and configuration tasks throughout the lab.

## 7 Why Ubuntu?

Many of the benefits that our laboratory has inherited from the advances in the Linux operating system are not characteristics only of the Ubuntu Linux distribution. Many other distributions could have been used, but Ubuntu was chosen for the following reasons:

- Good hardware detection saves us time.
- Good hardware detection makes Ubuntu ideal for passing on to students.
- Security patches provided for a long time after initial release; up to five years on the server version.

- Separate server and desktop streams, but with seamless integration between the two.
- Support of both GNOME and KDE.
- The *apt* software management system.
- The vast Ubuntu software archive.
- Currently very popular, which means that problems tend to be detected and fixed quickly.
- Predictable release cycles, so we can plan our upgrades.
- Emphasis on language localisation.
- The name Ubuntu sparks immediate curiosity in our students!

We decided to use the same Linux distribution on both server and desktop machines, reasoning that learning the quirks of Ubuntu in one environment would pay off in the other. This has indeed turned out to be the case.

## 8 Meeting current educational needs

We now discuss how the changes implemented in our laboratory since 2000 have allowed us to meet our new educational requirements.

E1. *Software deployment at short notice.* The combination of our local Ubuntu software archive and the fact that we use a standard Ubuntu installation makes it easy and quick to deploy additional open source software on the lab desktops. Effectively, we no longer have to worry about spending money on software that will not be used extensively.

E2. *Increased demand for hands-on assistance.* This has been resolved through our student assistants. Timetable synchronization means that students quickly learn who to approach with queries; many of them also appear to be more comfortable asking help of fellow students.

E3. *More students with PC/Linux expertise.* By moving to the latest version of Ubuntu at the end of every academic year we can offer more technically aware students a computing experience that's likely to be on a par with anything that they may have at home.

E4. *Many first year students have never used a computer.* These students need a simple non-intimidating interface, which we feel Ubuntu's GNOME environment succeeds in providing. It offers a clean and simple set of choices, with (by default) no redundancy in functionality. Navigation of the desktop is intuitive due to sensible default behaviors. Again, our student assistants are critical in providing these students with a positive experience in our laboratory.

E5. *Increased demand for local language support.* Ubuntu takes language localisation very seriously; we do not have to do any extra work to provide this extended service.

E6. *Increased use of home PCs.* We freely distribute Ubuntu to interested students, and encourage them to use our Ubuntu software archive. With the exception of a few proprietary applications such as MATLAB and Mathematica, students can completely replicate our laboratory's functionality at home.

E7. *Increased demand for software not requested by lecturers.* Our Ubuntu software archive often makes it very easy to satisfy these requests; we find ourselves fulfilling them more frequently.

E8. *Familiarity and ease-of-use critically important.* Familiarity and ease-of-use are no longer issues, because Ubuntu is configured with commonly accepted interface behaviours as default. Windows users, in particular, feel right at home with Ubuntu.

## 9 Comparison with a Microsoft-only approach

We support Windows XP on 50 of our 200 client machines via dual-booting. This is done only because of a few specific software requirements that can't currently be adequately met on the Linux platform. These are SAS Enterprise Guide, Microsoft Excel with VB Scripting and Equation Solver, and Microsoft Access. That only a quarter of our laboratory needs to support Windows XP is an indication of the large extent to which Linux satisfies our remaining requirements.

It is informative, however, to consider what limitations would need to be overcome before an entirely Windows-based laboratory would become feasible:

1. *Software set not openly available to students.* In limiting ourselves to Windows applications, we would be restricted by a plethora of licencing agreements, many of which would constrain redistribution to students (this includes Windows XP itself).
2. *No common archive of Windows software.* We would have to manually search for software and maintain our own archive, paying careful attention to licence restrictions for each piece of software. We would also have to plan—or guess—our future software needs; with the Ubuntu archive in place, we find our needs are often anticipated by other Linux users contributing to the Ubuntu software archive.
3. *High costs of Microsoft Server and client licences.* Despite educational pricing, licencing a laboratory the size of ours would still be very expensive. Our laboratory is growing, and restricting ourselves to Microsoft products would make it difficult to anticipate future costs, given how the Microsoft licencing regime has changed over the years. (We would also need to maintain at least two servers for some while, in order to retain the Linux user base.)
4. *No common interface for software management.* There is no universally convenient method for mass installation, configuration and maintenance of Windows and Windows applications. While there exist scripting tools which allow automation of Windows operations, using these

tools would imply having to write custom maintenance procedures for every supported application. Each installation has peculiarities making this difficult at best.

5. *Malicious software aims at Microsoft platforms.* Our laboratory is largely immune to malicious software because even our Windows desktops are hidden behind a Linux server. This is not likely to be the case in a Windows-only environment, while the problem of malicious software for Linux desktops is not yet a serious issue.
6. *No real equivalent of Linux community for Windows troubleshooting.* While information on resolving Windows problems can be found on the web, there isn't the vast cooperative community of users and developers that the Linux world can boast—not to mention the lack of access to source code. All that one requires to solve most Linux-related problems is Google and some reading time.

## 10 Unresolved problems and the immediate future

The following issues remain:

1. *Limitations of disk-images.* In setting up a client machine, we currently use Norton Ghost to propagate a disk image file across the network. Various scripts are then run on the client to complete the configuration. This suffers from the usual disadvantages of the disk-image approach, notably the need to maintain different images for different hardware combinations. As our laboratory becomes increasingly heterogeneous we plan to move to *FAI* (Fully Automatic Installation) as our main tool for Linux installations: this should remove the need for any human presence at the client desktops during installation.

   Until the release of better tools for remotely automating a Windows installation, however, disk-images still seem to be the best option for installing the Windows portion of the dual-boot machines in our laboratory.
2. *RAID on the server.* Despite multiple levels of backups, a main disk crash would still result in loss of up to an hour's worth of data. In an examination situation, for example, this is unacceptable. Software RAID on Linux now makes a RAID solution possible without the need for specialized controller cards; consequently, we intend using RAID-1 on our main server in due course.
3. *Corrupted Windows installations.* Occasionally, Windows installations become corrupted after extended use, despite users not having any administrator privileges. We are looking into tools such as Deep Freeze to reduce the frequency of these occurrences.
4. *Samba caching problems.* A default Samba installation writes a student's files from the server to the client computer every time the student logs on. The more data a student has, the longer

that students log-on process will take. The solution is a network mounted file system that continuously reads and writes to the server, which we are investigating.

5. *Anticipated increase in storage requirements.* Historically, we did not retain student data from one year to the next. As of this year, we will keep this data until the student is no longer registered with the university. The additional strain on our storage resources may force us to adopt a new strategy in this area.
6. *Software removal using apt problematic.* Removing software using *apt* is occasionally problematic. There have been cases where we wished to remove a single software package, but *apt* forced us to remove all of that package's dependencies as well, meaning that other software with similar dependencies became unusable.
7. *Redundancy of the software archive.* The functioning of our laboratory is critically dependent on our Ubuntu software archive. We plan to use one of our existing servers as a redundant archive.
8. *Hands-on training for new students.* A crash course for first year students will be offered in 2007 during our orientation week. This is intended to aid students who are using computers for the first time; it will also better advertise what services are available in our laboratory. Our student assistants will be used as tutors during this course.

## 11 Conclusion

We believe that our laboratory provides a good example of how schools with common computing interests can reduce their financial expenditure and yet provide a high quality and highly reliable facility that is tailored to their needs.

With the move to disk-based clients, the exploitation of Ubuntu's strengths and our new organization of staff and assistants, we are providing a better service using solutions that are far less technically involved than was necessary six years ago. Most of our tools work out-of-the-box. Our system now requires less of our time and less specialised knowledge to maintain. We therefore have more time to focus on our students' computing experience and to cater for lecturers' changing needs.

Our laboratory offers evidence that the Linux platform is not difficult to configure or maintain, either as a server or a client desktop.

Finally, we observe that our laboratory is fairly large, and our educational requirements relatively demanding. It seems to us that a Linux-based system is therefore likely to meet—or exceed—the needs of most university computer laboratories.

**References**

[McGlashan, 2000] McGlashan, S. A. (2000). Linux in a diskless laboratory. In

*Proceedings of the SACLA 2000 Conference, Strand, South Africa, 28–30 June 2000*, pages 145–152.