Open Source Report

November 7, 2005

ERAC

Educational Resource Acquisition Consortium

Table of Contents

Executive Summary

Open Source software has existed for about 20 years but recently, interest in Open Source has mushroomed in certain sectors. In British Columbia's K-12 education sector, Open Source has been used to varying degrees: some solutions appear in a single classroom in a school district, while other school districts have adopted it for specific district-wide uses. A number of school districts expressed an interest in ERAC producing a paper exploring the options and opportunities related to the use of Open Source in school districts.

Open Source software is developed with the source code freely available to the public. Anyone may download and use the software and make changes to it as necessary, in the hope that any improvements made by individuals will be committed back to the main source tree, so that everyone can benefit from the modifications. Schools and school districts are attracted to it for two main reasons: potential cost savings and the ability to create solutions specific to classroom, school or school district requirements.

This report reflects both the research done by a university co-op student ERAC hired to study applications and issues surrounding the use of Open Source in BC, Canada and in several other countries, and ERAC's interpretation of his findings. It is intended to serve as a guide to help districts in their deliberations around Open Source solutions. It is not intended to be a definitive resource evaluating everything districts should consider as they weigh the benefits and disadvantages of adopting an Open Source solution. This report provides an overview of Open Source issues, along with examples illustrating its current use, and is intended to aid technology leaders in their discussions around Open Source.

The report includes a summary of the issues and opportunities that school districts could consider in weighing the pros and cons of adopting Open Source. It introduces some background to some of the educational implications of adopting Open Source, its requirement for expert staff, the alternative methods of support available for those requiring help in using Open Source solutions, the issues around Open Source remote administration, the security and virus issues surrounding Open Source, parental concerns about the use of Open Source applications and the total cost of ownership.

The report also reviews Operating Systems; Open Source schools predominately use Linux and FreeBSD, with Linux being, by far, the most dominant. It also considers system software packages and single application server software. It presents models of the different kinds of labs using Open Source, including fat, thin and hybrid clients, and presents some of the benefits and disadvantages of using each solution. It also includes case studies showing how a classroom in Vancouver and the Kamloops/Thompson and Salmon Arm School Districts use Open Source. It also describes Open Source applications in specific schools in England and the United States and a broader-based application in Spain.

In the final section, the report outlines various resources and Open Source applications that may be used in the classroom. The list of applications is included for informational purposes only; none have been tested for the purposes of this report and are included so teachers see what kinds of resources are available.

Finally, this report is intended to serve as an introductory consideration of how Open Source could be used in K-12 education. The report does not advise districts on whether or not they should move to an

Open Source solution and it does not express a preference for any particular Open Source or Windows environment. It is simply intended to help districts in their deliberations and to offer some guidelines for those discussions. Districts and individual schools that are entertaining the idea of exploring Open Source options for their computing environments are encouraged to take the information in this report and use it as a starting point in their discussions. ERAC would also encourage districts to share their findings with each other, in an effort to continue an open dialogue on the subject.

Open Source and ERAC's Process

Open Source software is software that was developed with the source code freely available to the public. Anyone may download and use the software and make changes to it as necessary, in the hope that any improvements made by individuals will be committed back to the main source tree, so that everyone can benefit from the modifications.

Open Source has existed for about 20 years. Recently, interest in Open Source has mushroomed in the private sector, government and in the K-12 education sector as businesses and institutions consider the pros and cons of using these applications. Some organizations and institutions have embraced Open Source. Others use some Open Source solutions for specific tasks.

In the spring of 2005, ERAC decided to go forward with a report on Open Source, as it relates to school districts. It was responding to expressions of interest, by a number of school districts, in exploring options and opportunities related to Open Source.

ERAC hired an SFU co-op student, Greg Farrell to spend about two months researching Open Source applications and issues in the K-12 school system in BC, across Canada and internationally. He collected information by phone and then visited Kamloops and Salmon Arm School Districts in person to study firsthand their implementations of Open Source. Greg worked closely with ERAC's Janet Gregory, who helped him to direct his research and then compile the report.

This project was funded by ERAC with support from SD 34 Abbotsford which coordinated the co-op student's services.

ERAC's Role

The information contained in this report reflects our interpretation of the information collected by ERAC. ERAC is not endorsing Open Source or any of the examples included in this report. We are simply presenting a report, as requested by a number of member districts that explains some of the Open Source options, describes some of the pros and cons of Open Source solutions and gives readers some specific examples of how Open Source is being used in BC and internationally. The report is intended only as a guide to help technology decision-makers, not as a definitive evaluation of Open Source solutions.

Questions related to this report should be directed to:

ERAC
B.C. Educational Resource Acquisition Consortium
2nd Floor, 1580 West Boradway
Vancouver, BC V6J 5K8
Phone: 604.713.5920 Fax: 604.713.5921
E-mail: bcerac@bcerac.ca
Web Site: www.bcerac.ca

Summary of Issues and Opportunities

In weighing the potential for Open Source, schools and school districts should consider a number of issues related to its use. Open Source can represent opportunities for schools and school districts but, as with any technology, there are pros and cons. This section of the report outlines some of the issues, the opportunities and the pros and cons associated with Open Source.

a) The Educational Implications of Adopting Open Source

This is probably the area of Open Source K-12 development with the most obvious shortcomings, as well as the most development. It has been well documented that Open Source has only a small fraction of the educational applications available to users of Windows. As well, many of the applications that are available are largely untested in schools or too limited in their functionality to be of much use. (For more information, see Susan B. Shor's 2004 article titled "Linux Making Headway in Primary Education" (found at http://www.linuxinsider.com/story/Linux-Gaining-K-12-Ground-but-More-Software-Needed-37585.html).)

Many schools that have switched to Open Source feel that educational applications don't necessarily produce the results they would like and prefer to use computers as research and office tools to support the teacher's curriculum. Some schools, however, believe that the amount of educational software becoming available on the Web will make up for this discrepancy in software availability.

The potential for developing Open Source Software (OSS) for schools is undeniable. Many projects already in existence try to work closely with schools and teachers to produce effective solutions that provide valuable functionality and are easy to use. One example is School Forge, which states that its goal is "to unify independent organizations that advocate, use and develop open resources for primary and secondary education"

Along with School forge, K12LTSP and Edubuntu (described under "Target Areas") are both trying to bring the best resources available to schools in an easy-to-use, "turn-key" package.

b) The Requirement for Experts

All schools mentioned that the most significant cost with an Open Source system was technical support. As most of the hardware and software were free, or cost only a nominal amount, this isn't surprising. There seem to be two main considerations in weighing the need for an expert: the size of the Open Source implementation and, if the support is going to come from a teacher, how long that teacher will be staying at the school.

Trained professionals are used in all cases where districts have switched to a largely Open Source solution of any kind. The amount of phone call traffic for day-to-day issues --such as forgotten passwords or problems opening files-- would be far too much for teacher/administrators to deal with in their free time. Good administration can bring high cost-savings to a district. This has led those districts with years of experience with Open Source to stress that a good administrator is worth a higher-than-average salary.

There are few examples of a single school converting entirely to Open Source; in these cases the administrator usually has a background in UNIX or Linux administration. These administrators most

often recommended that, at the very least, a trained professional be hired to set up the labs in the school, in order to optimize performance and reliability. The drawback to hiring someone only for set-up is that the teachers left to handle daily administration could find themselves with a system they don't fully understand. In all cases, proper documentation of the set-up is essential but, in a case like this, its absence could prove fatal. Without proper documentation it can take someone unfamiliar with a system hours, or even days, to fully familiarize themselves with the code.

In particular, there can be problems with the building-block design of a Linux server's implementation. To somebody without sufficient knowledge or experience, the interrelation of the different components of the system can appear to be a vast maze.

c) Getting Help

The most common form of support for Open Source Software (OSS) is through list serves. These are discussion boards with free subscriptions that allow visitors to get help for a problem by simply posting the issue or question to the list. Usually, within an hour or two, there will be a response from someone who has encountered the problem before and can lead the questioner through its solution. People belonging to list serves are often very passionate about OSS and ensure that those people using it get the most of what it has to offer. Many of the people on the list serve will have written OSS applications so a request for help may be answered by the very person who wrote the program.

d) Open Source Remote Administration

There are benefits and potential problems with Open Source remote administration. Here is a brief overview of both sides of the issue.

The benefits of remote administration include:

1. Central Administration

With the ability to log on to any server connected to the network or the Internet, all of the Open Source administrators for one district could share a central office and telephone. This would give all district school staff one known point of contact for any question or support requirements. It would also enable the administrators to confer easily and look after a large number of servers.

2. Virtual Administration

VNC (Virtual Network Computing) allows an administrator at one internet-connected computer to remotely log in to any other internet-connected computer and manage its functions. The administrator could even operate servers across a district, without being restricted to that district office. As long as he has an Internet connection, the VNC would allow the administrator to install software, add or remove groups from labs or manage any other non-hardware issue. Of course the administrator must have access to the necessary passwords and firewalls to access the server’s systems.

3. Security

In order to protect critical functions of the servers, administrators set a password that restricts access to what is called the system’s “root directory,” the directory where the system’s most important functions originate. Open Source servers automatically come with a default password for this directory; it must be changed immediately upon set-up by the administrator because these passwords tend to be broadly known. Once the new password is set it must be kept safe, thereby limiting access to critical systems to those few people who can manage them properly.

Along with the security provided by root passwords and firewalls, there is encryption. All VNC Internet communication can be tunneled through a "Secure Shell", which is always encrypted. This means that an administrator may log into a server across the network or Internet and feel confident that sensitive passwords will remain private, even in transit.

4. Remote Screen Viewing
When handling phone-in support, VNC allows the administrator to view the same screen the caller is viewing. With both parties looking at the same screen, it becomes much easier for the technical support person to walk the caller through the solution. VNC can be run on Linux, MS Windows (but not DOS), and Macs, as long as the user has a network card and GUI.

5. Small Bandwidth Requirements
In the past, remote administration was challenged by the potential of slow Internet connections. Now, however, within the LAN of the schools, the connection speed is a minimum of 10 Megabits per second and usually 100 Megabits per second.

The potential challenges with remote administration include:
1. Limited Personnel Resources
Every school we contacted described the long-term administration options available to them as the most important factor in deciding on the scale, or possibility, of a switch to Open Source. For example, if only one teacher does the technical support for the Open Source labs there is a real chance that the project will come to an end if that teacher leaves. Remote districts are concerned about the challenge of attracting qualified technicians.

In fact, several districts using Open Source reported that they pay their technicians more than the average rate because, they said, they consider the technician to be the system's greatest asset. They also reasoned that with low hardware and software costs, and the higher level of system efficiency with a talented technician, they could well afford to spend more money on staffing.

ERAC's research found that, in some instances where a district tried to save money on administration costs or didn't have access to properly qualified people, the resulting vacuum created various problems with the system. One district even experimented with out-of-province (and even overseas) administration but found that the level of commitment from these people was too low.

2. The Potential Need for Command-line Administration
The user interface can be a Graphic User Interface (GUI) but is most commonly implemented as command-line interface, meaning text commands are used to configure the set-up. (The command-line interface is also the most secure and functional way to configure an Open Source server.) Most list-serve help offers an administrator the proper command for their task, along with an explanation of why the command works. With the dominance of command-line administration in Open Source servers, this means that some schools or school districts will have to dedicate resources to the re-education of IT staff only used to GUI administration. (Salmon Arm School District offered this kind of training to its network administrators.)

e) Security and Viruses

In any Open Source application, the biggest security threat involves the root password. As the initial set-up uses a weak default password, the new password should be created quickly after set-up, so that

the system isn't hacked into or the password doesn't become widely known. Secondly, even if the root password is changed, it can be shared too broadly throughout the district, meaning that a school or school district may lose track of who has access to critical systems. That could result in too many people making alterations to the system, leading to a situation where administrators would face difficulties in locating files and the structure of the entire system would be weakened. This could decrease the system's reliability and increase support costs. It could also allow unwanted guests to compromise the system.

Without the root password, however, an intruder would find it difficult to damage a properly configured system. They could damage peripheral directories or files, but would find it hard to hurt properly protected critical systems. This makes recovery from viruses much faster and easier than in systems that do not have a root password.

As a further security precaution, some school districts disable the parts of the server they know they will not be using; if a part of the system is not running it cannot be compromised in any way. This leads to better security and faster performance.

Many in the Open Source community say that there is virtually no virus threat to their systems. This is not exactly true but may not be too far from the truth. According to most virus-tracking websites, there are over 100,000 known viruses (McAfee). According to http://www.viruslibrary.com, there are a total of seven viruses for the Linux operating system. Updates and patches are written for Linux, just as they are for any other system; having to adjust to seven viruses instead of thousands means that, if the version of Linux is recent, it may have no current virus threat at all.

In its research, ERAC found that there are usually two reasons given for the discrepancy between Microsoft and Open Source virus attention. One is that Open Source is not used by as many people or for as critical systems as Microsoft and therefore does not give the virus writer the same notoriety as hitting the "big guy". If this is the case then, as Open Source gains ground, it will get more attention and become more affected by virus attacks. A big risk would only come from a virus able to get the Open Source application's root password and attack critical systems. Otherwise most effects can be well contained.

The second reason given for the lack of Open Source software virus vulnerability is that the code is written and tested by a larger number of people, as compared with proprietary software. This is said to produce tighter code with less vulnerability, meaning it is actually harder to write viruses for Linux than for proprietary programs. Security is still a concern, however, for Open Source systems. Here are the most common Open Source security applications:

SquidGuard (http://www.squidguard.org) is a combined filter, access controller and redirector for the Squid full-featured Web proxy cache. It is used for managing blacklists and other protective measures and is popular with most schools. It can block and redirect unwanted or unknown URLs and IP addresses and can have different rules for different users or times of day.

FreeBSD (http://www.freebsd.org) is a version of UNIX developed at the University of California that is considered ideal for Internet servers. It can be used to support many types of applications but is particularly popular for running Firewalls.

ClamAV (http://www.clamav.net/) E-mail servers often use this as their anti-virus software for scanning attachments. This package includes automatic updating over the Internet; the database of viruses it searches out is said to be up to date.

SpamAssassin (http://spamassassin.apache.org) filters spam out of e-mails. It is available for Open Source and Windows operating systems. It uses a wide range of tests to identify spam, making it harder for spam writers to work around it. Commercial versions are also available.

An article comparing the security of Windows and Open Source servers is at
http://www.geek.com/news/geeknews/2003Jun/gee20030605020295.htm

f) Parental Concerns

Schools in the initial phase of Open source application integration have noted some parental concern. In some school districts that have implemented Open Source applications, occasionally parents have expressed a concern that their children are being taught a technology that may have limited use, putting the children at a disadvantage when they enter the workforce. The level of parent opposition to Open Source seems to depend on parental familiarity with Linux before a school switches.

There are a few ways that schools try to alleviate such concerns. Most often, schools show parents that OpenOffice and StarOffice, the Open Source office suites used by most schools, can open and save files in all Office formats. Therefore, students can work on OpenOffice at school and MS Word or Excel at home, all on the same document. It should be noted that there are limitations to the reproduction of formatting between the Office suites, which become more significant as the file's formatting gets more complex. This means that, even if a student saves his school document as a .doc file, when he takes it home and opens it in MS Word, the formatting of the file will look different.

Another response to parents' concerns is in news from Europe. The Organization for the Advancement of Structured Information Standards (OASIS) is a global consortium focusing on standardizing web-service and e-business formats. They adopted an OpenDocument standard in May 2005, which is based on the format developed by OpenOffice. The push for OpenOffice as the standard was headed by the European Union and the OpenDocument format is expected to become an International Organization for Standardization (ISO) standard.

Microsoft and IBM, as well as many other North American companies, are OASIS sponsors and say they will be shifting to OASIS standards from their current W3C formats. This should mean that the formatting problems experienced by people who switch between OpenOffice and other suites should disappear in the future and could mean that the differences between Office suites, from a user's standpoint, could diminish. Koffice 1.4 already meets the OpenDocument standards and OpenOffice 2.0 will also conform.

g) Total Cost of Ownership

As with any large scale technology initiative, it is important to look at the Total Cost of Ownership (TCO) when determining the best route to take for any particular environment. While Open Source software may appear to have less of a significant financial impact for schools, there are costs associated with implementation and sustainability that should be investigated. For example, hardware and software costs may be diminished but the cost for a skilled technician and the development of district specific systems should be looked at realistically. There are a number of resources that can be accessed

when undertaking such a study.

School District 73, Kamloops employing Open Source technicians report that these technicians are currently making wages only slightly above average wages when compared with other technicians in B.C.

BECTA has reports available on TCO studies here:
http://www.becta.org.uk/corporate/publications/documents/BEC5606_Full_report18.pdf
http://www.becta.org.uk/corporate/publications/documents/BEC5606_Case_Study_16.pdf
http://www.becta.org.uk/corporate/publications/documents/BEC5606_Information_Sheetrev.pdf
CoSN TCO pages can be found here:
http://classroomtco.cosn.org/

Target Areas

This section reviews Operating Systems, core software packages and single application server software. There is also an in-depth discussion of Open Source thin client technology packages.

This section is presented with three core areas, starting with Open Source Operating Systems, followed by considerations of system software packages and single application server software.

Open Source Operating Systems

An operating system is a core application on a computer that manages and allocates the computer's resources. There are a number of different Open Source operating systems available, but schools predominantly use two. By far, the most dominant is Linux, in its variations. The other is FreeBSD which, as described earlier, is a version of UNIX developed at the University of California that is considered ideal for Internet servers. It can be used to support many types of applications but is particularly popular for running Firewalls.

This section focuses on Linux. Linux is an operating system much like Windows or Mac OSX and is based off of the popular and practical UNIX operating system. The main difference between Linux and the other operating systems is that Linux has been written by a large number of programmers in different countries, with the idea of making a powerful and stable operating system available to anyone who wants it. It also allows any interested person to see the code of the operating system and alter it as they wish. This gives the user the ability to customize the system to their particular needs.

Most versions of Linux allow a user to operate their computer in either a command line or Windows-like (GUI) environment. As Linux has matured, many of its distributions have focused on making it as user friendly as possible. Most versions seem to be modeled on Microsoft Windows and many Windows users would find it easier to learn one of the Linux variations than they would a Mac operating system.

Linux comes with full language support so the operating system (and many other applications) can be translated into Chinese, Japanese, Arabic and many other languages. Even though all the clients in a lab may run off of one server, each client can be configured separately, meaning one class can have five

clients running English, two running Spanish, four running Chinese and so on, from the same server.

There are some points of caution regarding Linux and its variations. One is that there are cases of incompatibility involving hardware and a Linux system. Fewer and fewer incompatibilities exist over time and, as long as proper research is carried out before hardware choices are made, this is not a problem. It is important, however, to pay close attention particularly to the compatibility peripheral devices such as modems, printers, scanners, etc.

The customized configurations and installations of the core Linux package are the variations listed below. Each variation may have its own hardware requirements, which will be listed on their websites if they exist. All provide GUIs for the type of Operating System navigation that most people will recognize.

The following are all variations of Linux:
RED HAT - The first release of Red Hat was more than 10 years ago. After version nine, it split into two versions: a proprietary version called Red Hat Enterprise Linux and a continued free version now called Fedora.

FEDORA - The on-going, free version of Red Hat. Fedora's support comes in the form of open communities instead of formal support. Red Hat created, manages and participates in these communities but support also comes from others who use Fedora. The company describes this support as "opportunistic" meaning there are no guarantees, warranties, or Service Level Agreements.

DEBIAN - Debian is known as the most stable of all Linux variations. It releases few new versions and concentrates instead on creating the most stable versions of what it already offers.

UBUNTU - Ubuntu is based off of Debian, which is generally considered to be the most stable of all Linux versions. They are committed to releasing a new version every six months. Paid technical support, if desired, is provided by Canonical Limited. They also offer 18 months of free security and other critical updates for each new release. This is suitable for desktop and server use.

EDUBUNTU - Edubuntu is a default setup of Ubuntu that is specifically tailored for classroom use. It is not considered to be separate from Ubuntu and they are handled by the same development team. Its projected release date is October 2005. The stated goal of Edubuntu is to provide the easiest OSS classroom option available, and it is geared at the educator with "mid-range" technical abilities and little time to tinker. Administration of the system is meant to be as easy as possible. Its aim is to provide collaborative classroom environments that can be configured under centralized management. It is dedicated to collecting the best of all available OSS educational software and making it available in one environment. Currently, Edubuntu contains the K12LTSP package. It also includes DrPython. PythonCard and PyGame, which are used to teach programming.

There are many different GUI window managers for Linux. The three most popular ones are:
GNOME: a popular and full-featured windows desktop environment that is a part of the GNU project, used in UNIX and Linux.

KDE: another very popular, full-featured windows desktop environment for Linux and UNIX.

ICE: a windows desktop environment for Linux that seems to be gaining in popularity. It mimics a Windows XP look and some estimate that is uses about one-third of the memory required by either GNOME or KDE.

Open Source Server Software Options

All Linux variations include server software. Many schools, however, prefer to use a software package to run their labs. The following section considers two options: server software packages and non-packaged server software.

Open Source Server Software Packages

Many of the Open Source server software packages available now have auto-detection for hardware, meaning that there is little effort needed to get the package installed and running in a lab. The best way to guarantee an easy installation is to fill the lab with the same make and model of a particular computer. Generally, regardless of the package, all students get their own login names and passwords, which allow the students to customize their desktops and keep these unique settings for their whole time at the school. One of the obvious drawbacks to this is that the students have control over their desktops and can change them as they like. Newer releases of K12LTSP (the most popular Open Source server software package) allow administrators to set a single default background for all clients that can not be changed.

LTSP (Linux Terminal Server Project) is an add-on package designed for making thin client servers out of Linux servers.

K12LTSP (http://k12ltsp.org/) is an LTSP package aimed specifically at K-12 schools. It is designed to be educational and as easy and turn-key as possible. In fact, many businesses and universities also use K12LTSP because they find it very practical. K12LTSP is mostly linked with Fedora but Ubuntu is planning on incorporating K12LTSP into its first release of Edubuntu.

K12LTSP designers began their project by asking teachers in Portland schools to name the tools they found most valuable in a computer lab; K12LTSP then tried to provide those tools in a convenient and easy-to-use package. This package is aimed specifically at making thin client lab set-up as easy and uncomplicated as possible.

The features K12LTSP provides include:

- A support list aimed specifically at people who don't know much about Linux.
- A default installation that runs everything off of the server, making it easy to run thin client labs.
- A default system which automatically checks terminals to identify hardware (including video cards, network cards, etc).
- Automatic creation of persistent desktop environments. This means that, if a school has all of its labs connected through K12LTSP, a student can login to any computer in the school and have all of their desktop settings and documents automatically appear. If the lab is run in isolation, then these properties are automatically loaded for the student on any computer within that lab.
- Website instructions detailing how to get a floppy disk network boot to work.
- Website manuals covering such topics as getting started, security, system administration and others.

- RHEL 4, or CentOS, is included in the newest version of K12LTSP. This is an enterprise grade Operating System, meaning that it is made for corporations and must be as stable and up to date as possible. This also means that its hardware detection is very thorough and accurate and is available to schools for free.

The programs included in K12LTSP are:

- Gcompris – Educational software targeted to children. Topics covered include math, science, geography, computer discovery, reading and learning to tell time. Screenshots and more information of Gcompris are available at http://www.ofset.org/gcompris/.
- Ggradebook – Keeps track of student grades for teachers.
- Tuxtype - This is billed as a typing teacher but many teachers view it as more of a game than a teaching tool. Teachers using it say it gets too challenging too quickly. (Gtypist or Ktype are commonly recommended for teaching.)
- Tuxpaint – A simple drawing program for young children.
- Tuxmath – A game where math problems must be solved in order to defend cities from missile attacks. Teachers say that it gets too difficult too quickly and needs to have adjustable speeds and levels.
- Kalzium – An introduction to the periodic table of the elements.
- Khangman – A version of the classic hangman game.
- Kmessedwords – A game to unscramble the words given.
- Kpercentage – Improves percentage calculating abilities.
- Kstars – A planetarium program showing the night sky from any location or time on Earth.
- Ktouch – Touch-typing instruction.
- Ktuberling – A drag-and-drop drawing game for young children, sometimes called a "potato editor" because it lets the child attach eyes, nose and so on to a head, much like the child's popular plastic potato head game.

Recent talk suggests that the edutainment package Childsplay may be included in upcoming releases of K12LTSP. Childsplay is a suite of games for young children, featuring Fallingletters, Findsound, Memory, Numbers, SoundNpic, and Soundmemory. It distinguishes itself from Gcompris in that it does not use the GNOMe desktop environment. It can be downloaded at http://childsplay.sourceforge.net/. It requires Python 2.1 or higher and pygame 1.6 or higher.

Please note that K12LTSP has reported troubles using USB Keychain devices on its Linux workstations. The next version of K12LTSP is supposed to take care of this problem.

Hardware for LTSP labs (including network cards and boot ROMs) is available at http://www.disklessworkstations.com. As well, there's a large archive of troubleshooting questions at https://listman.redhat.com/mailman/listinfo/k12osn.

Open Source Non-Packaged Server Software

There are a number of non-packaged server software options. Free online server comparisons are available at: http://www.serverwatch.com/stypes/compare/.

There are several server options that are most popular. Apache is Open Source, free to purchase and considered to be the most commonly used Web server on the Internet since 1995. In 2005 it was

reportedly being used by 68 per cent of Internet websites (http://httpd.apache.org/). (These numbers have been validated by non-Microsoft sponsored studies).

It is important to note that, like so many server solutions, the initial setup for Apache uses default passwords that are well known and obviously easy to hack into. It is typically expected that, after a server is opened for the first time, it will be hacked within 20 minutes, if it is left with the default passwords. The technician must also know which parts of the server he or she wants to disable, to allow for better security and faster performance.

As with most Open Source options, there is no official support for Apache. Help can be found at Apache HTTP Server Users List, comp.infosystems.www.servers.unix, or comp.infosystems.www.servers.ms-windows. There is also a General Technical Questions section at http://httpd.apache.org/docs/misc/FAQ.html#compare.

In terms of Linux servers, all variations of Linux mentioned in this paper come with server software in the package. This includes SAMBA, a program that allows files on the server to be run on a Microsoft or Mac computer with an MS or Mac look and feel.

Red Hat Enterprise Edition Linux servers are available at about $50 each, for educational institutions. Novell Linux servers are an option for those with a regular Novell Netware server arrangements or a license already in place that includes access the Novell Linux Desktop license as an option.

Possible Models

This section offers a review of the different types of labs using Open Source. First, however, here are the definitions of fat, thin and hybrid clients as used in this report.

Fat client – a server/client set-up where the client takes on all of the processing work. The server only stores data and some applications while the fat client does all of the work of running the applications for the user. For the purposes of this report, all fat clients have and use a hard drive of their own. This means that the client can store and run its own programs without involving the server. Some fat client advantages are lower networking and server requirements and increased flexibility.

Thin client – a server/client set-up where the server takes on all of the processing and storing work for the applications the user runs. The thin client often has no hard drive and is little more than a monitor and keyboard with the necessary hardware to tap into the server's resources. In thin client set ups, the server is normally a more powerful computer than a stand-alone unit. Some thin client advantages are lower administration and hardware costs and security is easier to establish and maintain.

Hybrid client – this server/client set-up is some form of the combination of fat and thin. The clients have hard drives and can store and run their own programs and files like a fat client. They also tap into a powerful server and can retrieve and run programs from the server like a thin client.

Most Common Implementation of Open Source Lab Servers

Virtually every school involved with Open Source is using it to run their servers. Whether it's for labs, Internet filtering or for running websites, schools are using Open Source servers. The schools report savings in the areas of maintenance costs, licensing fees, reduced hardware requirements and purchase prices for software.

The majority of the best-trained and most experienced administrators interviewed for this report said they felt that there was little difference in the amount of effort required to manage Windows, Novell or Linux servers. Patches and other security fixes are said to be much less frequent on either Open Source or Novell servers, but an overworked or under-trained administrator of any brand will produce a system with security risks. When managed properly, all systems are considered to be equally secure; when managed improperly, all are equally vulnerable to security risks.

The Benefits of Open Source Servers

The savings a school or district could expect to find by switching to Open Source servers include the following:

(1) Lower Hardware Requirements

One of the most easily quantifiable areas of cost savings for Open Source solutions is in hardware. The hardware requirements for Open Source are generally quite a bit lower than the requirements for proprietary applications and servers. Consequently, many schools have kept computers that they would have discarded and re-deployed them as file servers or client terminals. Numerous schools have mentioned that they have resurrected or extended the lives of many of their computers by switching to Open Source. Many schools also find that this allows them to run more servers than they would otherwise be able to afford with proprietary server software.

A fat client lab can use a recycled Pentium II or III computer as a file server while thin client labs can easily use these computers as clients. Most schools find computers of these specifications easy to obtain in off-lease purchases or through Computers for Schools (www.cfs.bc.ca). Numerous schools report that they have saved thousands to tens of thousands of dollars as a result of these lower hardware requirements.

(2) **Reduced Maintenance Cost and Time**

By far, the most commonly stated benefit mentioned by the schools who have switched to Open Source servers is the reduction in maintenance time and cost. Most schools claim that their system’s down time drops significantly once they switch to OSS, with many schools --and even entire districts-- saying that all of their servers and labs have gone for a year or longer without any interruption.

As well, fewer reboots of the servers are necessary. The two main reasons for reboots are power failures or the need to update the kernel of the operating system. Upgrades are released once or twice a year for most versions of Linux, but this doesn't mean a system needs the upgrade. If there are no problems with the system (for example, no security fixes are required) and the changes to the kernel don't affect the systems they are using, schools will often not bother with an upgrade.

Another often cited reason for long up times with Linux or Apache administration is that once something works it generally keeps working; it is very hard to break the system. The only real damage that can be done is by someone with ill intent who has the root password. Without the root password, as long as the administrator has set up the system properly, nothing critical can be accessed.

Open Source administrators also comment that there is a greater amount of control available in Linux, especially when using the command line interface. While this makes it more time consuming to set up Open Source servers, as compared with Windows servers, this extra time is justified by the overall security and reliability acquired in the process. Administrators have the option of shutting down anything they don't need. If an unnecessary part of a server is not running, then there is no way it can be hacked or fail.

With the right skills, a technician can write scripted programs that are robust and handle the most common maintenance issues the system experiences. This will eliminate many of the routine maintenance issues most administrators face and reduce delays staff experience from these issues. It should be stressed that these scripted solutions require knowledge of MySQL, PHP, Python, or other appropriate languages and are not a skill that every Open Source administrator may possess.

(3) Reduced Security and Virus Concerns
All Open Source servers discussed in this paper have a root password security feature. All critical systems of the server can be protected by this password. This feature makes it harder for anyone to write malicious software for an Open Source system.

(4) Lower Licensing Costs
Monitoring software and web services such as Web login, account administration and database population can be set up without licensing costs.

(5) Easy Remote Administration, Enabling Administrators to Look After More Servers and Labs
Remote administration is possible in either text mode (console mode) or GUI mode. The GUI mode may not be encrypted for Virtual Network Computing and is also slower than text mode but several options, including free NX technology, allows secure and encrypted GUI mode remote administration.

Liabilities for All Types of Open Source Servers

The two largest expenses in making a school Open Source are the transition costs and the technical support needed to run the Open Source solution. The transitional costs are unique to each school but typically involve: the research needed to evaluate whether or not the solution will be feasible and cost effective for the school; the question of whether technically qualified people will be available; the time needed to train staff on a new system.

The costs a school or district could expect to find by switching to Open Source servers include:
1) The cost of researching the new system's viability. Detailed assessments of all factors involved in switching to Open Source must be formally listed and analyzed, to see if a school has the necessary resources to implement a switch and to determine if a switch will, in fact, save the school or district money.

2) The cost of finding qualified people. The technical support issue of greatest concern can be in finding a qualified administrator to look after a school’s labs. Qualified Open Source administrators are not as common as other administrators, especially if a school is in a remote area. Technical support staff, according to an evaluation of costs in existing examples in British Columbia, earn slightly more than the average hourly rate of non-Open Source technicians. It’s important to note that most schools that use Open Source feel that a single Open Source technician can manage more servers than a single technician managing other types of servers; the potential cost adjustment of this factor has not be taken

into account in the above salary figure.

3) The cost of retraining staff. For the best use of an Open Source server, understanding of the command line interface is needed. This can mean districts must invest in retraining for a school or district if their administrators are only used to GUI interfaces. This has been done in BC: most administrators from SD83 in Salmon Arm were originally Windows administrators and were retrained by a Linux administrator working for the district.

4) Most free Open Source applications do not have any dial-up support or official support teams working for them on any level. For a school or district used to or wanting this style of support, this can be a big hurdle. Most school districts contacted for this report said however, that the list serve style of support was just as good, if not better, and is also free of charge.

5) The potential cost if code is not well documented. A major concern facing any school or district that switches to Open Source is ensuring that any custom code is very well documented. This means that there are comments in the code explaining what changes have been made, what the code is supposed to do and any other information that will help other administrators new to the code to understand it. The best code writers write code that is simple, clear and well documented, so that any other code writer will be able to understand it quickly. Without clearly documented code, a school or district may find itself having to rebuild its whole system from scratch if its administrator leaves.

One-school Labs and District-wide Implementation

There are challenges and concerns with setting up single labs or district-wide systems.

One School or Lab Only

It will be hard for someone unfamiliar with the technology, and especially the command line interfaces, to set up the server. Even if the person in charge of the project is familiar with OSS systems (but not an experienced professional), it is strongly recommended that an initial test lab be set up.

If a single interested teacher has set up a lab, then there is a good chance that, if that teacher leaves, the school may not find a replacement administrator to look after the Open Source lab. As with any technology, it is wise to have more than one person familiar with the system.

District Wide

This implementation will require trained professionals to configure and maintain the system. District administration staff should have strong skills in UNIX and/or Linux, as well as good FreeBSD skills. Districts have mentioned that it can be hard to find administrators with the skills they need.

Some districts ask applicants to perform skill assessment tests, to ensure they hire only qualified people. If a district doesn't know which questions to ask, however, it can be hard to determine which candidates are truly qualified for the position.

In an Open Source environment, there is typically a teacher in each school who helps with day-to-day issues. This is usually not a problem as most schools already have a teacher handling various technical administration issues. This is an issue a district will have to consider, however, when assessing the feasibility of their solution.

Districts wanting to get an idea of the hardware requirements and options available for a Linux-based system or computer could visit sites like Linuxhardware.net at http://k12linux.org/build.html or the

K12LTSP hardware page http://www.k12ltsp.org/terminal_guide.html. Another option is to post hardware questions on a Linux discussion group, such as K12OSN at https://listman.redhat.com/mailman/listinfo/k12osn or the Schoolforge discussion group "school-discuss" at http://www.schoolforge.net/sfdiscuss.php

Specific Implementations of Open Source Labs

Fat Client Labs with File Servers

File servers are machines that are typically used only for storing and sending files. They provide none of the processing resources needed to run software, leaving that to the fat client terminals entirely.
A file server's purpose is to free up hard drive space on fat clients in a lab. Similar to thin client labs, all possible applications are stored on the file server. It then sends these programs to a fat client across the LAN when they are requested. The main difference between these and thin client servers is that the server does not send any display information or take on any of the load of running the programs. Because of this, these servers do not have the same hardware requirements as they would if they were required to run the programs.

Many schools that use this system can get the necessary hardware for the servers from Computers for Schools; often a Pentium II for elementary schools and a Pentium IV for secondary schools is all that is needed for a lab. The individual fat client computers keep their hard drives and typically have at least 64 MB of RAM. Any individually licensed programs can be kept on the hard drives of the client computers and run locally, just as they would for any stand-alone desktop computer.

The amount of network traffic generated in this kind of lab is much lower than in a lab where the server handles the running of the applications, because only files are sent and not video display. As well, there isn't the constant back and forth traffic of a server that handles processing. As a result, the network connections can be as small as 10 megabits for the fat clients and 100 megabits for the backbone to the server. A 100 megabit connection virtually eliminates performance drags and is strongly recommended for roaming.

This means that the price of the network hardware for a fat client with a file server LAN is the cost of a 100 megabit LAN Switch, enough Ethernet cable to connect the clients to the server and 10 and/or 100 megabit network cards.

The advantages to a fat client/file server lab are:

1) Lower Server Requirements
File servers don't take on any of the memory or processing requirements for running software, so they don't need the large amounts of RAM or powerful processors that thin or hybrid servers need. This allows schools to use donated or stored hardware as servers for their labs.

2) Lower LAN Requirements
LANs in these labs are only used for sending a program to a client or for storing files, not for sending screen information or continual updates for user/program interaction to the client. The amount of traffic experienced by these LANs, therefore, is far less than what is experienced in thin or hybrid labs.

3) Potential Reduction in Software Maintenance Compared With Stand-Alone Labs
By having the option of running a number of applications from the server instead of from the terminals, a lab can save itself some of the time of configuring individual client machines (such as installing

software).

The disadvantages of a fat client/file server lab are:

1) Fat Clients Need More Resources than Thin or Hybrid Clients

Because the server takes on none of the work of running the programs, all fat clients need to meet all of the computing power required by the applications they are using. This results in the need for more powerful and expensive computers.

2) Increased Maintenance Time, Compared With Thin Client Labs

The remote administration benefits of thin client labs will extend to the maintenance of the servers of these labs. Administration of the individual client computers, however, will resemble that of a typical, stand-alone terminal lab.

Some of the maintenance benefits found in a thin client lab that are forfeited in a fat client lab include the following:

a) Increased risk of hardware maintenance: Fat clients always house hard drives and, typically, CD ROMs and floppy drives are left in as well. Because of their moving parts, these components are more susceptible to failure than most other parts of a computer. Leaving them in all the clients in a lab means there is an increased risk of hardware maintenance in the lab.
b) Increased threat of viruses: If the proprietary software operating systems are connected to the Internet, the maintenance time for fixing virus problems and installing patches will be similar to the time needed for stand-alone labs of the same type of operating system.
c) Time-consuming installation of software: Open Source software installed on the server will still be fast and easy, but proprietary software will take more effort. Because proprietary software is stored on the individual client computers, the time taken to install these programs expands from one machine (the server) in a thin client lab to numerous machines for fat client labs.

3) No Prepackaged, “Turn-key” Options

Packages such as K12LTSP, designed specifically to ease the installation and administration of thin client labs, must be rewritten for use with fat client labs. These packages have built-in settings that tell clients to boot off of the server instead of their hard drives. If a school wanted to use this kind of package for a fat client lab, custom code would have to be written to switch this procedure, so that the clients boot from their own hard drives as the default.

4) Administrator Adjustment Time

If the IT support for a school has never included Open Source servers or command line administration, there will have to be a period of adjustment so that staff can learn these new tools.

5) Potential Client Hard Drive Costs

Potentially, all client computer hard drives in the lab may have to be replaced. If the client machines are old enough, the hard drives are usually replaced, to minimize the costs and damages that would result from their failure.

Other Fat-Client/File Server Lab Considerations

No matter how much of the overall lab workload is given to the fat clients, all Internet connections are still typically run through the server. Therefore, as with all typical Open Source server options, the teacher can easily control whether the Internet is on or off for the lab. Some schools find that using only Open Source browsers, such as Mozilla or Firefox, reduces the number of virus attacks their labs

receive.

All the capabilities of remote administration described further on for thin client labs are possible for these servers as well, but not for their clients.

Complete Thin-Client Labs

Perhaps the most dominant trend in the use of Open Source in schools is to run a lab of thin clients off of a central server. One of the reasons for this trend is that the hardware, software and maintenance cost savings for these labs are said to be the highest, especially if a school needs to upgrade a lab or start up a new one from scratch. The big savings described by schools refer to the fact that an old, out-of-date lab can be transformed into a new, high-performance lab without the cost of 20 or 30 new stand-alone units. The biggest hardware cost is generally about $2,000, for the server and the necessary network hardware.

Here's how a thin client lab works: all programs are stored and run off of one powerful computer, called the server. A LAN connects a group of “simple” computer stations to the server. These individual computer stations are called thin clients and are simply a processor, network card and usually 64 MB of RAM (along with the necessary mouse, keyboard, and monitor that all terminals require). They do not have hard drives or software of their own and usually do not have floppy or CD ROM drives.

In a thin-client lab, an individual logs onto the system and then chooses, from among the programs available, the one he wishes to use. Even though all of these available programs are running off of the server and not off the thin client, the individual sees no visual difference between using a thin client and a powerful, stand-alone desktop computer. Performance is also comparable. Most schools claim that, in many cases, the thin client performance seems faster.

Thin Client Servers

In these labs, the server doesn't just store the programs, it runs them as well. For this reason, these servers must be more powerful machines than file servers in fat-client labs. The general rule of thumb is that a server has 100 MB of RAM per client it serves. This means that, with 4GB of RAM on your server, you can support 40 clients, all of whom could run OpenOffice and Firefox simultaneously.

It is important to note that, along with these RAM requirements, 100MHz of processor power is needed for each client hooked up to the server. For example, if a school purchased a dual core Pentium 2.8 GHz processor, it would have sufficient processor power for (2x28) 56 clients. If the number of students logged on goes above these thresholds, then performance will degrade and the system will seem sluggish.

In summary, a thin-client server typically needs:

- 100MHz of processor power per client;
- 100 MB (megabytes) of RAM per client;
- the necessary LAN cards to connect to the Switch and the Internet; and
- varying amounts of hard-drive space (these days usually over 100 GB), typically of type RAID 5 or SCSI for faster performance.

Thin Clients

All thin clients require a monitor, keyboard, mouse and power supply. Along with these, a thin client is most commonly said to require 64 Megs of RAM, a 100 megabit Ethernet card, a good video card, disconnection of the CD ROM and the floppy disk and boot instructions.

Most schools prefer to convert Pentium II computers, or better (some schools have mentioned that a Pentium I processor doesn't give them enough power and produces a slower station).

The cost of converting these computers to thin clients for a lab is hard to determine, because it is dependent on the costs of technical labour.

The best functioning thin client labs ensure all client computers are the same make and model, thus reducing maintenance and set-up costs. In fact, set-up and maintenance are considered to be the two largest costs a school that chooses an Open Source solution will face. In any case, the NICS (network interface cards) should all be the same.

Another maintenance-saving practice is to remove the floppy and CD ROM drives from all thin client computers. Both of these components have a lot of moving parts and can be prone to mechanical failures; their removal eliminates any possible costs of maintaining them. Student files can be saved on an FTP site, a drive on the server or by having students e-mail files to themselves.

The floppy disk can be left in the thin clients, if this is where the network administrators wish to store the boot instructions for the individual stations. To make this form of booting as mechanically secure as possible, some suggest the floppy drive can have its door sealed shut, to prevent students from removing the boot disk. A problem with overheating drives can result, however, and the disks can wear out in about six months. Whether or not the disk drive door is sealed, copies of the boot disk must be kept, in case the disks are ever lost or damaged.

Most schools feel that it’s better to have fewer moving parts in the thin clients. As a result, they reset the Basic Input Output System (BIOS) instructions, telling the computer where to find its boot instructions (they retrieve it from the network), thereby eliminating the need for a floppy drive. This means one less piece of hardware that can potentially fail and makes the labs that much more efficient. Unfortunately, this will not be possible with every thin client: the newer the client, however, the greater the likelihood that this will be an option. There are many resources available to make this resetting of the BIOS simple. For a list of up to date sources, it is best to check with a list serve such as the K12OSN.

A common observation is that a good video card is very important (4 MB RAM is the minimum recommendation). If it is not fast enough, then the lab will seem slow. Therefore it is important that those setting up a lab research and factor in the cost of video cards to meet the lab's needs.

In summary, the steps needed to set up a thin client are:
1) Find a PC that is not being used (including monitor).
2) Remove the hard drive, CD ROM and floppy drives.
3) Ensure there are 64 megabytes of RAM.
4) Install the 100 megabit network card to connect to the LAN.
5) Enable the computer to boot off the network by changing the BIOS.
6) Hook the client up to the LAN and see if it boots. (If it doesn't boot all the way, this usually means

that the video card needs to be updated. The video card needs at least 4 MB of RAM; otherwise, full colour won't show up on the monitors.)

A technically minded person should easily become proficient at the steps outlined above. Many schools remark that the ease with which they have become able to convert unused or donated computers to thin clients was an important factor in their staying with an LTSP set-up. Spare thin clients can be held in reserve, ready to be swapped into the lab as required.

LAN Requirements

Thin clients are connected to the server through a LAN. The dominant trend is to use a "backbone" of 1,000 megabits (1 gigabit), to connect the server to all of the clients. This backbone is a Switch, to which all of the individual thin clients connect. (A Hub is generally considered too slow for servicing the LAN and, therefore, most schools opt for a Switch.) If there are fewer than 20 thin clients running off of the server, the backbone going into the server can be as low as 100 megabits, because of the small amount of network traffic. If there are more than 20 clients, the backbone should be increased to 1 gigabit.

Each individual thin client is connected to this backbone with 100 megabit connections. A smaller connection, of 10 megabits, for example, typically results in bottlenecks in the network pipeline and slows performance. More bandwidth is required for thin client labs than fat client labs because video display information is sent across the network to each of the clients.

Therefore, the price of the network hardware for a thin client LAN is the cost of:

- 100 megabit LAN Switch for labs under 20 clients,
- 1 gigabit (1,000 megabits) LAN Switch for labs over 20 clients,
- enough Ethernet cable to connect the clients to the server, and
- the 100 and 1,000 megabit network cards as described previously.

The potential benefits to setting up a thin client lab are:

(1) Greatest Hardware Savings

Because thin clients have such low hardware needs, the bulk of their hardware can usually be obtained for low cost or with computers a school has put into storage. Often the only hardware costs for the thin clients are for bootable Ethernet cards and, possibly, for upgrading video cards.

Thin client labs also make better use of the CPU, another hardware savings. In a conventional stand-alone desktop lab each terminal has its own CPU, which spends most of the time sitting idle. In a thin client lab, all of the terminals share a single powerful CPU, thus greatly reducing the number of idle cycles the CPU experiences. This means a much better return on CPU investment.

(2) Greatest Software Savings

Typically, schools save the most on Open Source programs because they do not have to pay license and upgrade costs. Most software is free both to obtain and to run on any number of terminals by any number of users. All Open Source software is distributed under the GNU Public License. This license guarantees not only that everyone has legal permission to copy and/or distribute the software for free, but also that everyone has access to the source code of the program so that they can modify it to their own specifications. Schools that have chosen free Open Source operating systems and programs for their labs report that they save thousands of dollars each year. Many schools report that all programs

used in their labs were obtained for no charge.

(3) Added Maintenance Savings
There are also maintenance savings unique to thin client labs. For example, packages such as K12LTSP have greatly simplified the set-up and maintenance of these labs. Also, because the hard drive is one of the most likely parts of the computer to fail, having stations without hard drives means there is less risk of maintenance due to mechanical failure in the labs.

Other maintenance benefits of this type of lab include:
a) Re-imaging – A class in an Open Source server/thin client classroom takes seconds to re-image. This ensures that no student will find another student's work left on her computer from the previous period.

b) Control of Internet Access - For labs that aren't interconnected with the servers from other labs, it is possible for a teacher to shut off the Internet. This is done with a single command that disables the outgoing network card. Every other part of the lab will continue to run as usual, leaving all other applications available for use.

c) Assignment Drop Boxes - Labs can set up an assignment drop box for each class, where students can submit assignments but cannot read or copy from it. This makes it easy for teachers to collect assignments securely, knowing that no student can look at other students' work in the folder. In case a student wishes to change his assignment before the deadline, the folders will accept multiple submissions; they'll stop accepting files, however, once the deadline has passed.

d) Unique Lists of Applications Available for Different Classes – Each Open Source window management system allows users to create unique menus for each class that uses the lab. When instructors direct students to log on with an ID particular to that class, only the software relevant to that class is made available to the students.

e) Different Levels of System Access - The administrator can set up different levels of access for the system's different users. Students can be restricted to using the specific programs they need for a class or to being able to save files for certain applications only. At most, a student could damage his or her own personal account, and even then the administrator could easily reset this account to its previous state. Teachers can set rights or file space or whatever else a school wishes, but can also be kept out of critical systems they might not understand. This way, if a teacher knows he cannot inadvertently hurt a system, he may be more confident to try things he wouldn't otherwise.

f) Less time recovering downed labs – Thin clients never need to be rebooted. If a lab goes down for any reason, only the server must be brought up again.

(4) In-Lab Time Savings
Running all the programs in a lab off of a central server means that only one command is needed to disable the Internet or any other application for a whole lab. As well, compiling is a fraction of the time in a lab of stand-alone desktop machines. In a server/thin-client lab, students don't launch programs independently: because the server runs all applications, once an application is launched by one student, all students in that lab (or connected to that server) have access to the running version of the application.

The potential difficulties in setting up a thin client lab are:

(1) Learning a New System

One of the first concerns most schools express about switching to a complete Open Source environment surrounds the time and effort it will take to learn a new system. Time is often a rare commodity in schools and everyone is already putting in a lot of effort to get proficient at the current computer environment. There is no adjustment needed if the change is limited to the server, but if it involves a new Operating System and applications, staff may feel intimidated.

Most schools mention at least some level of staff resistance to an Open Source switch. For this reason, many schools run a test lab or a couple of test terminals of Open Source applications, so that teachers can get used to the differences. How much effort will be required for teachers to learn the new system is a cost that must be factored into a school's Total Cost of Ownership (TCO) assessment. Many schools mention two to three weeks are needed for staff to get to know the system and, generally, two to three months pass before they're confident with it. Generally, student adjustment is said to be rarely more than a few minutes. This is thought to be because the office suites and Web browsers are more similar than different to what they already know.

(2) Switching Curriculum Software

Complete thin client labs are often run with 100 per cent Open Source software, in order to avoid proprietary software licensing costs or legal issues. This includes all curriculum-supporting applications.

If a school is looking for more than the basics of Internet access, e-mail, office suites, file servers and printing, Open Source cannot offer as much as a Windows system. There are a great many applications written for OSS, but many of them are untested. If a school feels it needs certain educational applications in order to teach its classes, it should take the time to research and test whether Open Source alternatives to these applications exist and are right for the school.

There are ways to run Windows and Mac software in a thin client lab, but not without licensing issues. Exactly how much of a legacy system a school wishes to keep varies greatly from school to school and depends a lot on how much curriculum software specific to the old system it retains. The transitioning costs for these legacy programs and files must be factored in to any TCO assessment.

(3) Greater LAN Requirements

Because the server is used more and provides more for the client stations, a greater amount of information is sent across the LAN connections. Because of this higher amount of traffic, the LAN's requirements are higher than in other labs.

(4) Most Costly Server

Because the server is required to do more than in other types of labs, the hardware requirements are higher.

(5) Increased Hardware Set-Up Time

As mentioned above, thin client labs have their hard drives removed. This results, potentially, in fewer mechanical breakdowns in the labs, but it also means that time must be spent during set-up removing the hard drives.

Further Considerations
A number of schools have mentioned that using Open Source allows them to be more confident that all students have the computer resources they need at home as well as at school. They can send students home with disks or CD ROMs that contain all the applications the students need to do their homework. In schools with proprietary software, the cost has meant that many schools can afford only a few licenses for the software they use, meaning that the application is generally not available to the students outside of the classroom.

If a school switches to Open Office (which comes with Open Office Presentation, an equivalent program to PowerPoint), students can put this application onto their home computers and learn to use the program there. Schools trying this method report that it allows teachers to focus more on content and less on application training. It also ensures that most students have the applications needed to finish their homework. Also, because most students will have at least the basic office programs on their computers at home, this means that they are exposed to more than one suite of office software and more than one Operating System. They have found that students with this kind of exposure are more likely to understand general skills relating to computers, beyond limited application knowledge.

Hybrid Servers and Labs
These labs use a powerful server, much like what would be used for a thin client lab, but take advantage of the extra resource of a hard drive in each of the client stations. These servers are used to store and run applications that are licensed for multiple users. As with the thin client labs, administration and maintenance are easiest if all computers are the same model. If this is not possible, the default boot of the individual computers can be overridden.

Hybrid lab servers have large amounts of RAM and fast processors. There aren't the same hard guidelines for hybrid servers because the labs aren't solely dependent on them, as they are in thin client labs. A good deal of the performance of these labs depends on the power of the client computers. The low end for client computers would be around the Pentium II level, but there is a lot of variation in the specifications of fat clients. The more powerful the clients, the better performance a school will get from its lab.

These servers can use the same software that thin client servers use, but configured differently.

The normal steps needed to set up a hybrid client are:

- find a PC (including monitor) that is not being used;
- research how much RAM your clients will need and ensure it is installed;
- check that the hard drive has the space required and is in good order (if clients are old, hard drives are often replaced to increase reliability);
- check that any CD ROMs or floppy drives wanted are installed and in good working order;
- install the 100 megabit network card to connect to the LAN (if it's not already there);
- hook the client up to the LAN and see if it connects; and
- ensure that, if the server is used to run applications, the video cards are strong enough to efficiently handle display information.

Therefore, the price of the LAN hardware for a hybrid lab server is the cost of:

- a 100 megabit LAN Switch for labs under 20 clients;
- a 1 gigabit (1,000 megabits) LAN Switch for labs over 20 clients;
- enough Ethernet cable to connect the clients to the server; and
- the 100 and 1,000 megabit network cards previously described in the server and hybrid client sections.

The benefits particular to hybrid servers are:

(1) More Clients per Server than in Complete Thin Client Labs

The combination of a powerful server and Pentium II (or better grade), clients and fast network connections means that one server can reportedly serve over 100 clients. The server can pass off the work of running many applications to individual hybrid clients, instead of taking the entire processing load itself. This also means that administrators only have to maintain and configure one server for a potential 100 clients.

The hard drives in the terminals are also used for extra swap space.

(2) Fat Clients Can Ease the Resource Load from the Servers

When running the lab from the server (like a thin client lab) the extra hard drive space in the clients can be used as swap space, in case the server's memory is getting overloaded. This will improve the performance of the lab.

(3) Lower Client Requirements than Fat Client Labs with File Servers

Because the servers for these labs have so much RAM and processing power, clients don't need to be as powerful as they would in file server labs. If a program requires a lot of resources, the client can defer running the program to the server. In file server labs, the client computer must run all programs locally, so using a more powerful server can save the school money on client computers.

(4) Allow a School to Run Open Source from the Server and Individually Licensed Proprietary Software from the Client Hard Drive in the Same Lab

All types of labs discussed (thin, hybrid, and fat) can run both Open Source and proprietary software in the same lab. Hybrid labs, however, are the only labs that can take full advantage of the benefits that both thin and fat client labs have to offer.

Because all client computers in a hybrid lab have their own hard drives, each client can store its own individually licensed software, such as operating systems or other programs. Having a powerful server means that each client also has the option of running Open Source software (globally or locally) off the server. Setting up each client with a dual boot option provides the user with the opportunity to choose which of these two ways they wish to use the client. Dual booting means that, at start-up, the user at the client will be asked to choose which operating system she wishes to use.

If the user chooses Windows or MacOS, then that client computer will run like a typical Windows or Mac stand-alone terminal. If the user chooses Linux, then the client will run like a typical thin client terminal.

Many schools use the dual boot option because it allows them to run the best applications available to them at their convenience. For example, if it wants to keep programs on the server for maintenance and cost savings, it can use the Open Source option. This is often done for browsers and office suites. If,

however, it wants to run a curriculum-supporting piece of software that can only run on Windows or MacOS, it can store that program on the client's hard drive and run it on the individual client.

The Windows or Mac operating system will be stored on a partitioned section of the client hard drive, in order to comply with licensing restrictions. All proprietary software for these operating systems will also be stored in this partition. The Linux operating system can be stored on a separate hard drive partition or on the server. This choice will depend on hardware requirements and the administrator's preference.

It should be noted that this is not the only way to get Open Source and proprietary programs to run on the same computer. Other options include running Open Source programs from the server, while in Windows or MacOS or running proprietary software from Linux stations.

(5) Potential Reduction in Software Maintenance Compared to Stand-Alone Labs
With the option of running a number of applications from the server instead of from the terminals, a lab can save itself some of the time of configuring individual client machines (such as installing software).

The potential down sides to this type of lab set-up are comparable to those found in a fat-client lab set-up.

Combining Open Source with Windows and Macs
There are several ways to combine Open Source with proprietary software. One of the most common is to use a type of emulation software that will allow the Open Source and proprietary software to be run on the same terminal but in separate windows. WINE, SAMBA, and Win4Lin are examples of this type of software.

WINE is used by many schools to run Windows applications in a Linux environment. One or two schools mentioned that older versions of WINE were too problematic and they had to look for a more reliable solution, but none of the schools running current versions of WINE mentioned any problems.

Some say it does not support sound, microphone and USB ports, limiting its use for special needs students. This should be researched further before a school makes a decision about WINE.

SAMBA can be used if a school wants to have its Windows and Mac computers to be included on the network with the Linux computers. SAMBA is free Open Source software that allows a host operating system, such as Linux, to interact with a Windows or Mac server or client. This means that a Linux-based terminal can access applications from a Windows server, allowing MS or Mac applications to still be available to the lab if needed. It also allows the MS-based applications to be controlled in their distribution, saving licensing costs. SAMBA also mimics a Windows or Mac server, allowing Windows or Mac terminals to access Open Source servers as though they were native servers. Windows NT/2000 file server users can even connect to shares on a Linux server using SAMBA. Many schools use SAMBA to phase out legacy Windows or Mac applications gracefully.

All applications used by a lab can be run off of just a couple of servers, which many schools cite as a way of reducing administration costs. Windows and Linux screens, for example, can be run in separate windows on the same workstation. An additional Windows terminal server could be connected to the lab to provide native Windows support using Remote Desktop.

KNOPPIX CDs are one of the most popular ways for people to experience Open Source programs on a Windows or Mac computer without having to install anything permanent. It boots from a CD drive and leaves no permanent files on the booting computer. When the computer is shut off, it is as though the Open Source programs never ran on it.

A list of Open Source applications that can run in a Windows environment can be found at http://osswin.sourceforge.net/. This includes OpenOffice, mail and news, firewall, FTP, databases, compression software and much more. It also has several math and biology pages that are geared for post-secondary education.Win4Lin

Win4Lin is short for Windows for Linux, and the schools that use it say it works very well. It is not free, and prices can be found at http://www.win4lin.com/content/view/68/115.

An Open Source Windows package Server bundle is available from http://sourceforge.net/project/showfiles.php?group_id=93507 (many other bundles are available at this site) This bundle includes: Apache2 , MySQL , PostgreSQL , OpenSSL , Xmail , SlimFTPd and many other features.
XAMPP is an Apache Distribution for Linux, Solaris, Mac OS X and Windows. It includes the Apache web server, MySQL, PHP, SQLite, and more. It is available at http://www.apachefriends.org/en/xampp-linux.html#374

There is also an option to “dual boot”, meaning that you can install Linux on the free space on your hard disk and choose to load either Windows or Linux (or another OS) moments after you turn on your computer.

Most Common Introductions to Open Source

Small, Isolated Thin-Client Lab

Any school wanting to convert its labs to Open Source should run a small test lab first. This allows a less experienced Linux administrator the opportunity to learn the system and experiment with it, without having to worry about how their experiments will impact the school. The other benefit is that an isolated trial lab will give the faculty the opportunity to see front-end software, such as OpenOffice, and become familiar with it before using it in a classroom or lab environment.

Commonly, the servers for these labs will be run off one recycled computer with the thin (or hybrid or fat) clients made from a few other recycled computers. It is not difficult to set up these labs in such a way that they do not interfere with the current IT set-up of the rest of the school. The only part of the lab that will extend beyond its doors is a single Internet protocol (IP) address for connecting the lab to the Internet.

The trial lab's Terminal Server will have two network cards. One is to connect to the school network (for the Internet) and the other is to service all the thin clients within the lab. As long as the lab has its own dedicated Switch or VLAN (virtual LAN on a managed Switch) the LTSP LAN does not affect the rest of the school.

Numerous Linux thin client labs such as these have been set up by teachers inexperienced in Linux. These teachers, however, were usually already doing technical support for a school and were curious

about trying a Linux system, so some level of experience and training is important for the project's success.

Often, these experiments resulted in the introduction of more Open Source labs to the schools; there are also several cases of these labs shutting down. Usually, the labs that closed were discontinued because the teacher heading up the project moved on to another school. The odds of the new tech support teacher also being interested in Open Source has been very small so the new tech support teacher would usually implement lab set-ups that are most familiar to them.

A note of caution: schools often remarked that if a person does not have adequate knowledge or understanding of the OSS system, that person can easily get in over his head. Trying to do too much too soon greatly increases the chance that the experiment will end in failure.

Finally, all schools recommended testing everything, including all updates and changes, in a controlled environment before deploying them into the school.

Web Server in the Library
This is a common way to introduce Open Source to a school. It typically involves setting up a single terminal running Firefox or Mozilla for surfing the Web. The aim is to let staff and students see how much there is in common among different versions of Web browsers and how easy it is to learn a new browsing application. The disadvantage of this method is that the terminal will often go unused and sit idle in a corner of the library. With no reason to learn a new Web browser, few people will make the time to try it. The same problem exists for the few schools that set up a single Open Office terminal.

Knoppix
This is a CD (or can come from a download to a CD) available for people who want to try Linux without installing it on their hard drives. The Linux distribution used is Debian, using the KDE desktop environment. It typically includes Koffice and the Konqueror Web-browser, as well as GIMP and OpenOffice. Systems vary, but it is normally quite easy to get a computer to boot itself off the CD. Instructions are available at http://www.knoppix.org/. Nothing is saved to the hard drive, meaning that as soon as the computer is switched off and the CD ROM removed, all files that have been created cease to exist and your computer is returned to the state it was in before the CDs were installed.

Regardless of the style of introduction chosen by a school, several common traits to successful trials are the following:

- Start small and work up gradually;
- understand what you require and research how to get it;
- know the costs, pros and cons of your plan;
- know where and how to get help (e.g. ask a lot of questions on listserves);
- test everything before putting it into a working lab; and
- inform all teachers who will be affected about the potential benefits of using Open Source and train all those who will be affected in the proper use of the new system.

For more information, see the article about John Hansknecht, the director of technology at the University of Detroit Jesuit High School and Academy.

http://software.newsforge.com/article.pl?sid=05/05/18/1944227&from=rss

Case Studies

A number of BC schools and school districts have, to varying degrees, adopted Open Source solutions. This section outlines a few case studies, in BC and internationally, to give readers a taste of how Open Source solutions have been applied.

Kamloops/Thompson

The district of Kamloops/Thompson has about 60 sites overall, in 38 elementary schools, 10 secondary schools and seven school administration sites and provides a comprehensive example of an Open Source solution. The district reports savings of $750,000 during the past two years attributable to the switch to Open Source.

Among their sites, there are 3,200 Windows stations and 1,700 Linux stations. The total for the district is more than 100 servers, 90 per cent of which are Linux or Free BSD. All firewalls run Free BSD.

Their servers are distributed as follows:

- One server per elementary school;
- four servers per secondary school (three are Linux, and one is a Novell that will soon be Linux);
- 22 servers at the board office, of which 20 are Open Source; and
- six servers at administration sites (two Linux, four Novell).

The district has seven software support technicians, one systems analyst, one network support/help desk and one network support/programmer looking after the servers. It has also hired an I.T. Manager, a hardware technician, a hardware/video conferencing technician and a network wiring technician.

The district's Open Source administrators look after 98 servers:

- One Linux server in each of the district's 38 elementary schools;
- Three Linux servers in each of the district's 12 secondary schools;
- 20 Linux servers at the board office; and
- Four Linux servers at administration sites.

All of Kamloops/Thompson's elementary school labs run Open Source software; the majority are Linux-only, 100-per-cent Open Source machines. About one in six clients has a dual boot option with Windows.

Most of the district's Open Source servers run a no-cost version of Red Hat, though some Open Source servers at the board office run Red Hat Enterprise Edition servers to support some proprietary programs they run.

The district started with an LTSP package for its labs, but decided to adapt it to run applications off of the client computers instead of the server. Lab terminals can run programs both locally (off of the client resources), or globally (off the server). They have found that running programs globally will make a Pentium computer with only 32 MB RAM run like a new, powerful machine.

The terminal hard drives can also be used for extra swap space when the lab resources are experiencing high usage loads.

The district has found that using the same model for all computers in labs saves administration time; therefore, they do this wherever possible. If it's not possible, the default boot can be overridden.

A benefit for students is that all students get to customize and keep their "own" desktop for the entire time they are at a school. This is done by giving each student a unique username and password, which they keep for the full year. The district has also found that if a student shuts off a client improperly, no damage is done to the software or the operating system. Neither the client nor the server will have to go through a "scandisk" or similar procedure once started again.

Typically, elementary school lab servers use a dual processor AMD 1.8 GHz with 4 GB RAM, which cost about $2,000 each. The hardware requirements used by Kamloops' elementary school labs are:
Servers:

- 100 MHz processing power per client;
- 128 MB RAM per client, but they can get by with 64 MB per client;
- 40-80 GB IDE hard drives for the older servers, new servers have 100+ GB hard drive space; and
- 100 Megabit LAN.

Clients:

- Pentium II or better client computers with 64 MB RAM;
- one supported network card (such as 3COMS or Epros) per client; and
- one supported video card per client.

The amount of hard-drive space on these clients is not of much concern because the entire operating system is downloaded over the LAN and all student files are stored on the server. The client computers get their initial boot instructions from a PXE network card boot or, where the age of the client makes this impossible, a small file stored on the client hard drive. These boots require less than 10 KB of instructions. The initial boot is done in Red Hat but the user environment is the Debian operating system.

Kamloops Secondary's set-up includes one proxy server that uses Apache for the school website, one firewall server, Novell servers for labs and student accounts and Novell servers for school administrators.

With the use of DHCP the Novell servers are connected to the Internet through the proxy server. This means that everything coming into the labs from the Internet is filtered through Squid and DansGuardian, which checks all traffic against inappropriate domains, URLs, text etc.

All the in-town schools will have Web services run off of three central servers: one for Web and Proxy, one for Webmail, and one for backup.

The firewalls for each school are run with FreeBSD. VPN tunnels are set up to each firewall, which allows centralized maintenance.

One Red Hat server runs SquirrelMail for e-mail and webmail access for each student. The SquirrelMail default install is very basic, but can be expanded with more than 300 plug-ins, to add features.

The district has had great success at all its sites, performing remote administration through VNC. It encrypts its VNC connections by tunneling through the servers' secure shell.

Most problems are handled over the phone in minutes. Staff report that it's easy to implement quotas, such as available disk space for a class, and that the MS machines have no problems recognizing the settings. They report that the same is true for read, write and execute access to files. These quotas, as well as changes to blacklists, passwords, groups and so on, are done on the Web through Webmin. Those with the necessary accesses --such as teachers, administrators or others-- can change these settings in a GUI environment.

A bigger administrator concern is the hardware problems that require a visit to a school or site. To minimize this challenge, teachers have been shown how to perform simple hardware maintenance. The district has found that electrical storms can shut down a server, but staff have been able to get their Open Source servers going again with only phone-call support.

The district has experienced no administrative problems running Novell servers but it is in the process of phasing them out. The district made that decision because it found the Linux servers to be just as reliable and easy to look after and they do not have any licensing fees. The district did have some Windows servers that had reasonable licensing fees (they were only used as Web servers). These were phased out because the district was looking for something more reliable. Linux Apache are now being used for Web servers.

Salmon Arm

This school district is an example of fat client/file server labs. It has a total of 38 sites overall, broken down as follows:

- 25 elementary schools (K-5);
- two middle schools (6-8);
- two high schools (8-12);
- two secondary schools (9-12);
- two store-front schools;
- two alternates;
- one continuing education school; and
- two administrator sites (board office and works).
- The schools have a total of 1,200 client computers with an average of 30 computers per lab. All labs are Windows 98 or XP and have no roaming profiles.
- All servers are Open Source, including the one Apache server that runs their Web services.

The servers in all 38 sites (except the board office), are broken down as follows:

- one file server running SAMBA;
- one proxy server running Squid; and
- one LDAP slave server for redundancy if the web goes down.

The board office has an Apache server, DNS server, Samba file server, qmail server, LDAP master server and a back-up server. (The district decided against Windows servers because of licensing costs.) The schools run SAMBA servers. There are 117 servers, with three servers at each of 37 sites, plus the six at the board office. To look after these servers, the district has one Manager of Information Services and five network technicians. The terminals run Windows XP with roaming profiles. School secretaries also use XP with roaming profiles.

Maintenance is district-wide, with the six technicians looking after the entire district. All of their support is done through the command line interface; no GUI is used. With the exception of the lead manager, all technicians were originally trained to support Microsoft servers. The lead manager trained all the others in Open Source server support. Maintenance is largely handled from one central location with each technician assigned his or her own six or seven sites, a combination of administration and school labs.

The district reports that most help desk calls involve: problems opening software, which is usually resolved in about 30 seconds over the phone; printing problems, which are quickly fixed by clearing the printing cue; or problems with e-mail, which usually involve forgotten passwords.

All labs are composed of fat clients that log into a SAMBA file server. The file server distributes the applications through a launcher, allowing the servers to be used as an extension of each client's hard drive. This means that the Windows clients do not have to reboot when they pull an application off of the server. As with all schools using Open Source servers, installing software on the server is immediate and requires no rebooting.

The district has found that there is no more effort maintaining its client computers than there was with the old Microsoft NT servers. Mac computers can connect to the server using Netatalk and the PC machines have no problems recognizing quotas that have been set through the Open Source servers. They can also recognize read, write and execute permissions for files without difficulty.

This type of set-up can also have a Webmin, meaning that teachers can go to a Website and, through a GUI, change their quotas (such as available disk space for a class). They use an application named DeepFreeze to keep students from inadvertently (or purposefully) messing up the desktop or computer. They have written custom scripts that allow students to save files in a "Favourites" section, which is then automatically saved to an H-drive section particular to that student.

The district's LANs are a mixture of 10 Megabit and 100 Megabit connections and it reports finding that roaming works better with 100 Megabits. All server hardware is obtained through Computers for Schools, either Celeron or Pentium II for elementary or Pentium IV for secondary. Only one server has had any down time in the past year. The district averages one client computer down for every two labs, usually due to vandalism.

The backup software the district uses is free and backup is done using external hard drives, which connect to the server through USB ports.

It's worth noting that when the district was using Novell servers, each school could set different rights and permissions for the servers themselves. Now everything is handled centrally. The district has noticed that this has sped up maintenance by providing consistency and predictability in file locations

and structures. Before rights and permissions were handled centrally, it would often take hours for a technician to find where the necessary files were located and before he could start to work on the problem. That was clearly a waste of time and money.

Vancouver, Chief Maquinna Elementary School

Alan Zisman (azisman@maquinna.vsb.bc.ca) reports that his school has adopted OpenOffice.org as its primary office software for student and staff use. He said the school has found that by configuring it to save to Microsoft Office formats by default, it is easy for users who work with MS Office at home or in other settings to share files back and forth, without needing to fuss with file formats.

Zisman reported that while the software is relatively slow to load, it has been easy for students in Grades 1-7 to adapt to using it, even if they have worked with other programs. He said a number of the user-interface features (such as adding and working with word processor headers and footers) are easier to use than the analogous features in MS Word. He said the school has also appreciated the export to HTML function of the OpenOffice.org presentation module, which creates standard web pages, viewable in any browser on any platform. PowerPoint, by comparison, creates web pages that often are not viewable unless using a Windows version of Internet Explorer. (He said Web-versions of Grade 6 and 7 presentations are available for viewing online at http://maquinna.vsb.bc.ca).

Zisman said he has found OpenOffice.org to be cost effective. Like many schools, Maquinna has received a large number of donated computers and does not have a budget that allows it to purchase even deeply-discounted commercial software for each computer. The free OpenOffice has fulfilled that need.

Zisman offered a link to a presentation he made at last October's BC Computer Using Educators Pro-D session on free software options for K-12 education. It's available online (exported from OpenOffice.org) at: http://www.zisman.ca/Free/img0.html.

There are also some international experiences with Open Source that are instructive for BC educators. Here are three examples:

Spain

Representatives of the Junta of Extremadura regional government in Spain announced that a sweeping initiative has put 80,000 computers in schools across that region. The announcement came at the 4th annual GNOME Users and Developers European Conference (GUADEC) in 2003 in Dublin, Ireland.

The program, which was instituted last year, has just completed installation in schools, with a ratio of one computer for every two students. All of the computers are using a customized version of Linux, called GNU/LinEx, that features the GNOME desktop. Other free software productivity programs are also installed on every computer. The total cost savings exceeded 18 million euros ($25.6 million Cdn), or nearly one third of the total budget of 67 million euros ($95.3 million Cdn).

"For us, software libre (open source software) was the only choice," said Francisco A. Huertas Mendez, technical coordinator of GNU/LinuEx of the Junta de Extremadura. "We were able to stretch our budget very far and provide a powerful and easy-to-use environment with Linux and GNOME. We are also able to give the students all of the productivity programs they need."

The Junta of Extremadura has also created 33 computing centres for the general population. The centres feature one-on-one computer assistance, so users who are unfamiliar with computers can learn computer and e-mail basics. The centres have drawn citizens of all ages and walks of life.

"Extremadura has launched a very significant initiative, and we are extremely pleased that they chose to use GNOME as their desktop environment for Linux," said Miguel de Icaza, CTO of Ximian, Inc. and GNOME Foundation president. "This initiative not only gives computing ability to all of its students, it also has the potential to grow a local IT industry in Extremadura. This is an excellent example of the control and flexibility that Linux and open source give governments and public sector institutions."

More information on the Junta de Extremadura's computer initiative can be found at www.linex.org.

England

Orwell High School, in Felixstowe on the East Coast of England, is a school with some 1,000 students ranging in age from 11 to 18. The school has just received Specialist School for Technology status through a government initiative.

Funding is never easy for schools in the UK public sector and John Osborne, the Deputy Head of the School responsible for the Specialist School initiative, found himself faced with a difficult situation in early 2004. Funding for hardware was very limited and he couldn't contemplate upgrading to Windows XP, since he would have to replace some 50 or so PCs with higher-end models, just to run the software.

A capital cost in the region of £25,000 ($51,680Cdn) was well outside the budget and when he took into account a software licensing spend in the region of £13,000 ($26,873Cdn) per year, Osborne became convinced that he had to find a better way of using the school's resources.

When Osborne contacted Andy Trevor to discuss his cabling and server requirements, an idea arose. Trevor had recently been working closely with Mike Banahan of GBdirect and Open Forum Europe, discussing Open Source implementations in education. Orwell High School was precisely the kind of establishment that would be able to benefit from the range and kind of ICT provision that Trevor and Banahan had been planning.

The school required four principal ICT classrooms with approximately 30 work stations in each one, distributed printing services and support for a number of smaller clusters of one to five workstations. All staff at the school now have laptops, and the school wanted to link these to the network wirelessly.

The school had specific software requirements for the teaching environment, nearly all of which are met and exceeded by standard Open Source software packages, such as OpenOffice.org, MySQL and The Gimp. These have a huge advantage over their proprietary counterparts because the students can also run them at home on their PCs, without needing to worry about software licensing. Some proprietary teaching packages have no direct equivalent in the Open Source world at the moment and some of the teaching packs in use were based on Microsoft software so support for this legacy software was also important.

The Linux-based desktop uses a range of standard applications, including OpenOffice.org which provides word processing, a presentation package and a spreadsheet; all of them are able to save and

import files in their native XML format while retaining compatibility with Microsoft formats. Quanta is used as the HTML editor, the KDE education package provides an assortment of educational software components, Scribus is the desktop publishing package and The Gimp is an excellent image manipulation tool with a wide range of capabilities.

Overall, the project has been a resounding success. "I can't believe how easy it has been to move to Linux. The systems were installed and working within a week and it has been a revelation how simple and painless the process has been. I have saved thousands of pounds per year and got a brand-new ICT infrastructure at the same time," John Osborne said. He added: "Without switching to Linux, I would have been forced to cut back on our ICT hardware and software provision. There simply wasn't the budget to upgrade to the latest versions of the software, nor to keep replacing suites of PCs on a three or four-year cycle. Now I have no licensing costs to worry about for the Open Source parts of the solution. We shall be moving to a complete Open Source basis as quickly as is practical and hope to start working with other schools interested in this type of development to share ideas and best practice."

Osborne can be contacted by e-mail at john.osborne@orwellhs.suffolk.sch.uk. The school website, www.orwellhs.suffolk.sch.uk/ will also be used to keep interested parties up to date with developments.

Portland, Oregon

The Riverdale School District in Portland, Oregon, began using Linux in 1996 as a simple file server for Windows and Macintosh work stations. Right away, moving to Linux from Novell saved the district more than $30US ($35 Cdn) per year in licensing fees for every computer in the district. Used initially in the server room, Linux proved to be easy to deploy as well as dependable in providing file, print, web, and e-mail services.

The next step was to put Linux in classrooms. When Riverdale High School opened in a new location in 2002, school officials chose to combine Linux with a thin-client solution: four powerful Linux servers to drive 100 Linux thin-client computers in computer labs and in classrooms.

Without hard drives, floppy drives, CD-ROM drives or any other moving parts, the thin-client computers cost the district just $472 ($554 Cdn) each to assemble from scratch, and most of that cost was in the flat-panel monitors. (Older 486 PCs and Macs may also be used as thin clients.) All the computer processing takes place on the servers; the thin-client Linux workstations simply plug into the network and boot up. As a result, the computers used by kids are immune from malicious tampering and viruses, and require no configuration or maintenance.

The server software used by Riverdale High is the free K12 Linux Terminal Server Project (K12LTSP). It installs in less than 30 minutes and is pre-configured with many useful applications.

Linux thin clients aren't the only computers at Riverdale High. Students and teachers use Macintosh computers and Windows-based PCs, too. The Linux servers provide a system in which any user can sit down at any machine --Windows, Mac, or Linux-- and have instant access to their files, regardless of operating system. More information on the Riverdale School District Open Source Project can be found at: http://business.newsforge.com/article.pl?sid=02/05/17/1319208&tid=35&tid=3&tid=30&tid=31

Resources

Applications

See Susan Shor's “Linux Making Headway in Primary Education” for more information. (http://www.linuxinsider.com/story/Linux-Gaining-K-12-Ground-but-More-Software-Needed-37585.html)

Most schools that have been using Open Source for a long time have adjusted to the limited educational Open Source software options and have focused on using software as a support, rather than as a teaching tool. For this reason, the most commonly used applications are a Web browser, like Mozilla or Firefox, and the office suite packages OpenOffice and StarOffice.

The following list of applications begins with the best known and most commonly used Open Source applications.

Worthy of special note is Schoolforge, an organization aimed at unifying the efforts of independent K-12 Open Source organizations. Its website, http://www.schoolforge.net/, lists a large number of applications and resources and it supports a list serve for discussions.

Open Source Desktop Applications

Mozilla - http://www.mozilla.org/products/mozilla1.x
Mozilla is a full-featured Web browser that includes mail, a news suite, pop-up ad blocking, e-mail spam filter and a simple Web page designer that allows a user to create and publish his own Web pages. It is currently quite safe from viruses.

Firefox - http://www.mozilla.org/products/firefox
Firefox is a lightweight, streamlined Web browser based off of Mozilla. It has the same pop-up blocking, tabbed browsing and lack of virus attention that Mozilla provides. It also allows the importing of favourites, passwords and more from other browsers, saving a person from having to re-enter all of this information from scratch.

OpenOffice - http://www.openoffice.org
OpenOffice.org is a productivity suite that includes a word processor called Writer, a presentation program called Impress and a spreadsheet program called Calc Spreadsheet. These programs provide all the functionality that a user of MS Office would expect from Word, Power Point and Excel, though tools such as Spell Check are not as advanced as they are for MS Word. It also includes a web page editor and database tools. It can open and save documents from a wide range of formats, including Microsoft Office.The free versions of Open Office come with PDF file conversion option built in for free, and some schools use Impress to make slide shows.

Star Office - http://www.sun.com/software/star/staroffice/index.xml
This is the enterprise version of OpenOffice and for educational institutions; the cost is only media (CDs) and shipping.

GIMP – http://www.gimp.org/

This is the GNU Image Manipulation Program that is said to be equivalent to Adobe Photoshop. It can be used for image authoring, composition and retouching, or as a paint program. It provides image format conversion and numerous plug-ins can be added for further functionality. More information can be found at http://www.gimp.org/about/introduction.html.

Moodle - http://www.moodle.org
Moodle is an online course management system. It supports the making of assignments, quizzes, chats, forums, and more. It is said to offer similar functionality to the proprietary program Blackboard. A highly detailed comparison between Moodle and Blackboard is at the following site. (Moodle wins) http://www.humboldt.edu/~jdv1/moodle/all.htm

TeacherTool is an application that allows a teacher to broadcast a window from the teacher station to all the students' desktops, allowing students to view teacher demonstrations. In addition it has the ability to allow a teacher to see which programs a student is currently running. If the teacher is having trouble getting the attention of all the students in the class, TeacherTool lets the teacher shut down all or any portion of the applications the students are running, making sure the students are not distracted when the teacher needs their attention. This can be done for individual students or the whole class and can also be done remotely, from anywhere in the school, as long as that computer is connected to the school network. TeacherTool is now a part of the K12LTSP packages.

Squeak – http://www.squeak.org/
Squeak is designed to allow the creation of simple interactive educational programs. A Morphic programming environment is provided to allow easy customization of a Squeak program. This could allow graphic responses to a child's input or morphs to react to touching other morphs. An authoring tool called Alice is available for non-programmers who want to write their own interactive 3D worlds. Other features include playing and manipulating MIDI files, sampling (though this takes a lot of memory space), a full range of internet capabilities including an e-mail client and browser and several games.

XLogo - http://xlogo.free.fr/index-en.html
This is a simple application to help teach programming to kids. There are loops, tests (conditionals) and procedures to control a turtle as it walks across the screen. The turtle leaves a trail behind it, allowing students to draw pictures with their turtles. There is a free manual (written for kids) available at http://www.snee.com/logo/logo4kids.pdf, which covers many areas and even includes chapters that gracefully introduce operator precedence in programming. Another tutorial is available at http://www.cs.brown.edu/courses/bridge/1997/Resources/LogoTutorial.html

QCAD - http://www.ribbonsoft.com/qcad.html
This program is used instead of AutoCAD for drafting classes. QCAD does not support 3D design, but it does all of the same 2D functionality that AutoCAD provides. It is not free: the licensing cost is $260 for an entire school.

The following nine applications are used by SD83 Salmon Arm:
Picassa - http://picasa.google.com/index.html
A free photo editor and organizer available from Google.

Putty - http://www.putty.nl/download.html

Free encryption software.

TuxMath – http://www.newbreedsoftware.com/tuxmath/download/
The Linux penguin Tux tries to defend cities from missile attacks by solving math problems. This program is known to have some problems changing grade levels but this is supposed to be fixed on new versions. It is also said to progress in difficulty too quickly for many students. Teachers' main wish is that the speed and levels could be adjustable, so that a level could concentrate on addition, for example.

TuxType - http://tuxtype.sourceforge.net/download/
This typing program is considered more of a game than a teaching tool. Gtypist or Ktype are recommended for teaching typing skills.

TuxPaint - http://sourceforge.net/projects/tuxpaint/
This is a drawing program that can also make animations.

Childsplay – http://childsplay.sourceforge.net/
This is a suite of games for young children, featuring Fallingletters, Findsound, Memory, Numbers, SoundNpic, and Soundmemory. It distinguishes itself from Gcompris in that it does not use the gnome desktop environment. It requires Python 2.1 or higher and pygame 1.6 or higher.

CutePDF - http://www.cutepdf.com/Products/CutePDF/writer.asp
This is a free PDF conversion tool for Windows applications. Note the installation requirements easyHTML - http://www.geocities.com/ResearchTriangle/1500/easyhtml/
This is just what it sounds like; a simple HTML editor that has features such as free Javascripts.

FrontPage Express - http://www.123ecommerce.com/frontpage-express.htm
This is Microsoft's freeware HTML editor. A review is available at http://www.irt.org/software/sw001/

Equivalent software for Windows applications includes the following:
Scribus – Adobe In Design
Inkscape – Adobe Illustrator
Anjuta – MS Visual Studio
Qcad – Autocad (2D only)
Ktouch – typing tutor
Simply Accounting/Quickbooks – Gnucash

MyClassroom - http://www.visionlearning.com/library/myclassrooms.php.
This is a course authoring system with a library of learning modules.

OpenCourse - http://opencourse.net/home/
This application is aimed to provide a means of putting course and materials online. (The website has not been updated since 2002.)

Interact - http://www.interactlms.org/spaces/space.php?space_key=1
This is an online learning and collaboration platform that is an alternative to WebCT and Blackboard. It allows a lot of freedom in the building of sites.

AbiWord - http://www.abisource.com
This is a stand-alone word processor suited for use on older hardware. It can work well on Pentium I or even 486 processors with 16 MB RAM. It supports a number of document formats including MS Word, supports spell checking and includes an HTML exporter. Plug-ins to extend features include thesaurus, translation interface, and spelling in different languages. It is available for Windows, Linux, and Macintosh OS X.

PDFCreator – http://sector7g.wurzel6.de/pdfcreator/index_en.htm
This generates PDF files for any Windows program. It creates a selection on the "File -> Print" menu of the word processor or drawing program.

Notepad2 - http://www.flos-freeware.ch/notepad2.html
This is much like the Notepad program for Windows. It has the added functionality of color syntax highlighting.

Blender – http://www.blender3d.org/cms/Home.2.0.html
This is a sophisticated 3D animation program that supports polygon meshes, NURBS surfaces, Beizer and B-spline curves. It also features a fast ray tracer, sound editing that allows a user to sync a soundtrack, and a built-in editor for Python scripting. It requires only 20 megabits of free disk space.

Audacity – http://audacity.sourceforge.net
An audio editor and recorder that can be used to record live audio, edit various sound files, convert tapes and records to digital formats and cut, copy and mix sounds together. Various effects can be applied and multiple tracks can be mixed together. It has been recommended for teachers who want to record audio commentaries or lessons for future podcasts.

SchoolTool Calendar – http://www.schooltool.org/
Included in the latest K12LTSP, version 4.2.1, this allows the creation of Web-based calendars and timetables. Each teacher, class, student and schools team or group can have their own timetable or calendar. There in an option to “overlay” multiple calendars, allowing a person can check for conflicts or manage time appropriately among assignment dates, team practices, and personal events. It can also be used to coordinate school resources, such as overhead projectors. Each calendar can have its own level of access which will ensure student privacy. http://lwn.net/Articles/139866/

XMX - http://www.cs.brown.edu/software/xmx/
XMX is a program that allows a teacher to share a desktop with a student or group of students, meaning that all Linux X Windows system displays in the session will show the same screen, and all participants will be able to interact with that screen. The students can see what the teacher does on the screen and vice versa. People at different locations can interact with the same applications at the same time and all can see each other's work. No changes are needed to the applications to allow this to happen. With XMX, it appears that each user's input seems to be coming from only one user to the application being used. NOTE: this is old software (1998).

ATutor - http://www.atutor.ca
A Learning Content Management System that allows teachers to create, organize, and distribute Web-based educational content. It can be used with assistive technologies for learners with disabilities. Because it conforms with IMS/SCORM content packaging specifications, content developed on ATutor can be swapped into other learning systems.

K12Admin - http://k12admin.cmsd.bc.ca/pen Source Report
This system is designed to maintain Linux servers in individual schools while maintaining a homogeneous network throughout a school district. Staff in each school can keep their own student and staff accounts within this homogeneous system.

EDesk, or MyEDesk - http://www.myedesk.org/refdesk.aspx
This is a resource management tool that's designed to make the creation, management and distribution of website resources easy. This includes personal portal pages, lesson plans, etc. It also focuses on online discussion threading and quick uploading of files to the website.

Site@School - http://siteatschool.sourceforge.net
This is a website content management system geared for primary schools. Its features include safe chatpages, a template editor that allows webpage design without knowledge of HTML, an Intranet for teachers, project collaboration pages for students and simple site management. It has modules such as agenda, picture gallery, calendar, news, email, and more.

Additional information is available at http://sourceforge.net/projects/siteatschool.

.LRN - http://dotlrn.org/
This enterprise-class Open Source software is designed to support learning and research. It was originally developed at MIT and is now used by K12, government, and universities (approximately half a million users).

A KNOPPIX CD version can be downloaded from http://e-lane.org/pub/, which does not have to be installed and disappears as soon as you turn off your computer. A persistent version can be downloaded from the dotLRN homepage.

Educational Games

(Some of the following are repeated from above)

Anagramarama – A word game for Linux and Windows that asks the player to find as many words as possible before time runs out. Players advance if they get the longest word possible from the available letters.
Asymptopia Math Crossword Builder – Web based in Javascript, which can lead to browsers timing out, therefore use Mozilla Firefox and follow the instructions at http://asymptopia.org/staticpages/index.php?page=AsymptopiaCrosswordBuilder-README. This program automatically generates crossword puzzles that involve addition, subtraction, multiplication, algebra and fractions. Located at http://asymptopia.org/AsymptopiaXW_v22/xw.html it will automatically generate math problems or allow a player to write her own.

atnag – For French pre-schoolers, this game includes letters, numbers, enumeration, memory, sorting and more.

Childsplay – This is a suite of games for young children, featuring Fallingletters, Findsound, Memory, Numbers, SoundNpic, and Soundmemory. It distinguishes itself from Gcompris in that it does not use the GNOME desktop environment. It can be downloaded at http://childsplay.sourceforge.net/ and requires Python .21 or higher and pygame 1.6 or higher.

CubeTest – A particular cube must have its identical twin identified from a group of four other cubes. Children have to mentally orient the cubes to match the single one to see if they're the same.

CueCard – A study flashcard program that can adjust card frequency for problem cards and introduce new cards to replace learned ones.

DroidQuest – A reworking of "Robot Odyessy 1" which aims to teach basic digital logic programming.

Gcompris – This game teaches the use of keyboards and mice to preschoolers. It can also teach letters, numbers, analog clock reading, words and some algebra.

Gretools – This is a suite of programs that seek to improve a student's vocabulary using the Barron's word list (about 4000 words). It remembers words that were erred on. It uses word guessing and synonym quizzes.

Kalcul – Interactive math game for 8 to 15 year olds.

Kard – This is a memory game.

Kavlon Coloring Book – This includes pictures to colour.

KDE Edutainment Project – This is a project aiming to provide educational software for the K Desktop Environment. They currently offer applications for mathematics such as Kbrunch, which generates questions involving fractions, Kig, which is for learning about geometry and KmPlot, which plots functions. It also offers language programs such as KVocTrain for vocabulary training with flashcards,

KLettres for learning letters and syllables in different languages and KHangMan.

KHangMan – Hangman in French or English.

Klettres – For ages two through six, this game teaches alphabet and sounds in different languages.

LeoCAD – A game of making things with bricks.

Lexter – In this game, letters fall down the screen and the student has to arrange them into words in English or French.

Linux Letters and Numbers – This game uses pictures to teach letters, numbers and spelling.

Math Volcano – This game aims to help students to memorize multiplication tables, up to the 12 times table, using flashcards.

Pavlov – In this game, students can study large question databases using operant conditioning methods.

Pizza Business – This teaches budgeting skills by allowing students to run a virtual pizza business.

Project LRNJ – This is a role-playing game that teaches Japanese.

QuizzShow – A Jeopardy-style class review game.

RealTimeBattle – A programming game involving battling robots.

Skill Literature – This game teaches about shapes, rewarding deduction over speed.

The Linux Memory Game – This offers different levels of difficulty for the card game “Memory”.

The Neverending Tale – A web-based, choose-your-own-adventure story written by the student.

TickleTux – Another hangman game.

Tux Math Scrabble – This game encourages multiple understandings of math equations; the game plays like Scrabble.

Tux Paint – This is a simple drawing program for younger students. It includes sound effects.

Xletters – This is a typing game.

Math

Gnumeric – A GNOME spreadsheet.

Gnuplot – A command-driven plotting program.

gonvert – This program converts units of measurement.

GPLconx – This is for displaying lines and conic sections of 2D hyperbolic geometry.

GraphThing – This program allows the exploration of the relationships between sets.

ISETL – This offers programming language to aid in learning calculus and other higher level math.

Kalcul – A math game for up to Grade 10.

Kbrunch – A game about solving problems through adding/subtracting or multiplying/dividing fractions.

Freeduc-primary is targeted to primary school education; with more than 40 free education software apps for any OS can be found at http://www.ofset.org/freeduc-ecole.