Thin Client Technology in the Academic Environment

Jesús Martínez, Iván Pérez, Susana Muñoz, Daniel Lancho,
TEDECO Group (TEchnology for DEvelopment and COoperation)
Technical University of Madrid (UPM), Spain.
jmartinez@fi.upm.es iperez@babel.ls.fi.upm.es susana@fi.upm.es
dlancho@fi.upm.es

Abstract

From research groups at the universities of developed countries there is a growing interest in providing solutions to problems of developing countries. In this context we have studied typical problems in many (educational) institutions, such as the lack of technicians who repair the computers, the administration of the machines, and also the difficulty to maintain and configure the old hardware available due to the variety of characteristics of the different machines and the amount of hardware breakdowns and software issues (viruses, administration issues) that the local staff has to face up to with their equipments.

We propose a thin client approach that takes into account the human, hardware and software characteristics of developing institutions to provide a complete service for a computer network. The network administration is reduced to the administration of one server only. The maintenance of the machines is simplified and old computers can simulate the running of a powerful computer. Our proposal results in a cheap, simple (from the support point of view) and powerful (in terms of achieved functionalities) design.

Index Terms

Thin client, terminal, network boot, remote desktop, computer classroom, hardware recycling.

1. Introduction

During centuries, the universities in developed countries have been the birthplace of scientific research and later of technical research. The goal of research groups have traditionally been to innovate for providing solutions to problems of the local society and improving the quality of life. In fact, it is what they have done during years: developing the quality of the occidental society.

We can realize just by looking around that in our current world the needs of developed and developing societies are completely different. The research has never being oriented to solve the problems of the poorest communities because they were first far, second unknown and third non-profitable for the researchers.

Since years ago the universities are changing their priorities and there are many important universities that are widening their interest fields to include also the research for solving problems in developing countries with a completely different context with respect to our classical interests. In this line, many cooperation groups have appeared inside the universities and their aim is studying the challenge of countries, institutions and population of developing countries and innovating proposals to improve their problems.

This is the context where we present the work that is included in this paper. We describe a specific technical problem that many (educational) institutions are suffering in developing countries, after that we present our proposal in detail and provide some data about the pilot experience that we have already completed successfully[1]. But also, as a Computer Science School we have focused our support in the design and development of technical solutions, taking advantage of the wealth of the University: knowledge, research, innovation and human resources. In an attempt to make the most out of every developed project we have started to propose and supervise final dissertations, in which students tackle some problems such as those addressed in this paper. We have organized this proposal within a University Development Cooperation (UDC) program, where students are our most valued heritage. They are the best funding source for our cooperation group when they decide that their work will be applied to developing countries. Our major goal is to involve students in these programs, not just to achieve the project results, but also to train and raise conscience on students about cooperation.

A. Motivation: Technical overview

The goal of cooperation groups is not being present at the affected area forever. They are expected to leave definitely at some point. For this reason, minimizing and lightening maintenance of the infrastructure that they have developed should be a priority.

Institutions of developing countries in general and educational institutions (universities, secondary and primary schools) in particular have a number of problems for maintaining their computer infrastructures.

Adapting an underdeveloped environment to the use of computers and the internet can be a daunting task. Four key problems have been identified on this matter. First, the stability and reliability of the power network use to be too low. Problems of this kind often cause permanent damage to the hardware. Second, transportation costs are expensive. Using the smallest device available for a task is essential to keep costs low. Next, developing countries often get low-tech hardware. Europe donates old computers through NPOs (non-profit organizations), which obviously cannot run the latest versions of most software applications. Last, but not least, local personnel is not qualified to maintain the system, and qualified personnel is not constantly in the place.

B. Thin client concept

A **thin client** (figure 1), also referred to as terminal, thin client terminal or boot diskless device, is a low-performance computer, usually small and sometimes assembled with recycled hardware. Due to its low performance a thin client works always connected to a server which provides one o more services through terminal (deprecated nowadays), web or remote desktop access. But the main difference found between a thin client-based system and other network-based solutions is that in the first the user actually works on the server. In the client-server network paradigm, it's usual that a client remains connected to one or more servers in order to use some resources, but this dependence is much higher within a thin client environment, because of that a thin client could also be defined as a computer with extreme network dependency. It's a full centralized system, where every user has to be always connected with its server. The server makes the processing and it sends the resulting information through the network to the clients, whose only task is to present it properly to the user. Today the communication between

[1] At the University of Ngozi, Burundi

thin client and server simulates the transport of the screen, and it's implemented by a protocol called **remote desktop**. Through this desktop shifting the user can get the feeling that his work is being done locally, as if he had a whole computer for himself.

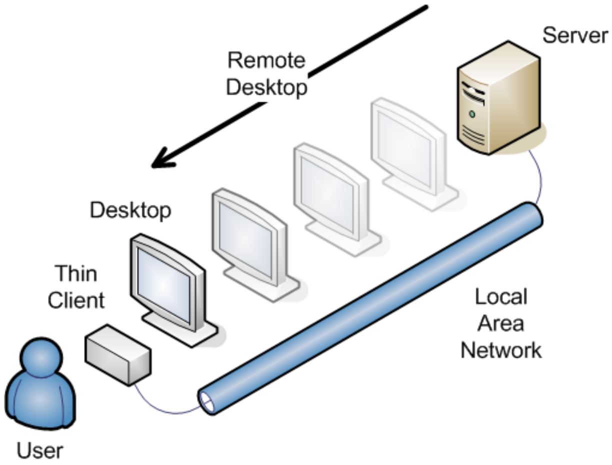

Figure 1. Remote desktop as a screen translation.

C. *Brief state of the art*

Back in the 80's, *graphical terminals* or simply *terminals*[2], were quite common devices mainly due to the computer characteristics in those days, where a mainframe could fill an entire room, that added to its high price suggested the shared use of the computer through several workplaces. Then a workplace was only composed of a monitor and a keyboard. But in the next decade, when personal computers (PC in advance) came and graphical user interface-based operating systems appeared, terminals lost their position —back then in the company and other institutions such as the University— to leave room for what most of us use now in our daily work. Only one step from single purpose to multifunction machines that radically changed our lives. Since then, every advance has been made in this way, using the PC as an essential device. However, today we find again terminals in some companies and universities as a new and quite changed device: thin client. Thin client devices have also advanced, and every day there is less restrictions to the kind of software that can be used with these. Moreover, thin client technology has been integrated with the workstation to provide a "user friendly" solution: most users get the feeling that they have a whole PC just for them, even though they do not.

D. *Content organization*

This paper has been structured as follows. At the beginning, in section 1, we have seen an overview where it is analyzed the detected problems that we have found when we tried to build a computer classroom in a developing country, problems which motivated our work. This overview was followed by the definition of a possibly new concept, thin

[2] The term *thin client* was coined during the mid 90's.

client, to finish with a brief description of the thin client history and the summary that you are reading right now about this document structure.

In the following two sections, 2 and 3, we will show our approach to the problem described above and the solution provided by means of a thin client-based technology. In the second section we try to justify our approach, showing how the purposed solution solves some of the problems pointed out before. The third section outlines our proposal. We also describe how a pilot experience of this solution was applied successfully to the University of Ngozi in Burundi.

Finally, in section 4 we expose our conclusions, advantages and disadvantages, providing a detailed comparison of our solution against other common approaches; to conclude with section 5 where we will show a glance of some open problems that we have detected and we are still working on.

2. Thin client-based solution

Although initially it was designed with a different goal, terminal-based technology has evolved strongly over the years and current implementations of thin clients possess new characteristics that we find interesting for our purpose. Thin client computing can be seen as sharing a computer, with all the devices connected to it (i.e., peripherals, hard disks, etc.), via remote desktops. This shows one of the advantages, which is also the main advantage of this technology: **resource sharing**. Moreover, building a network of thin clients can be seen as a cloning process, which yields **scalability**. This property expresses the fact that new clients can be added to the network with low-cost in terms of time and hardware. This attribute is quite desirable in networks, especially in those environments where new requirements can arise often and in those where hardware is expected to fail —due to the reasons we pointed out at section 1—. Finally, what may be considered as one of the most crucial properties: **reliability**. This cannot be defined accurately for systems in general, and it may not be so important in some environments. In this scenario, reliability may be defined as the quality that ensures the minimum operational guarantees for the correct performance of the entire system. Additionally most points in which failures can take place will not cause jeopardy in the whole system. The proposed solution achieves this goal as described later in this section.

A. A special approach: thin clients in the classroom

Our work in cooperation has led us to focus our attention on the **academic community** and the computer classroom environments. Both are good scenarios to use thin client technology for several reasons. First of all in both environments we find an innate resource sharing: in a classroom the teacher tries to teach at once the same concept to all his students, and this also happens in a computer classroom where teacher and students share and use common software. Besides, the architecture of a thin client solution follows the hierarchy of an education system, centralized in one computer which serves to all students, and it's only managed by one administrator: the teacher.

Moreover, in a **computer classroom** based on thin clients connected to only one server, every student has a single and private environment: the desktop. But under those desktops all they share common resources (from up to down): applications, drivers, operating system and hardware. This means that every user works on top of the same basic configuration at any moment.

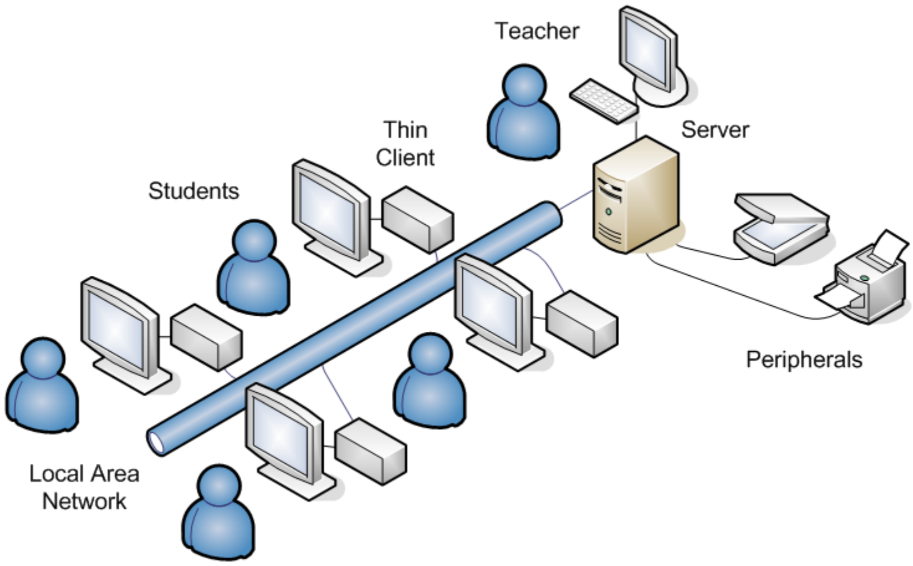

Figure 2. Computer classroom based on thin clients.

A new teaching way: A graphic representation can be seen in figure 2.

Although the content of the lessons and the classroom organization is basically the same as in other computer classrooms, teacher and students should be aware of some changes. This changes come from the fact that now they are working in a highly centralized system with a strong network dependency, what means that they are all working in the same workstation, using the same storage system, the same operating system and same peripherals (apart from screen, keyboard and mouse). This raises some security issues that should be taken into account by the operating system and the network administrator.

This kind of centralization also forces users to change their point of view, for example when they work with their peripherals (i.e. thin clients' USB ports won't connect devices to the workstation unless the thin client management software implements this feature; and this problem is also extensible for CD drives, printers and other peripherals).

Integration with a learning management system (i.e. Moodle, a b-learning platform) could be considered a perfect match. This kind of systems is centralized so its performance is not affected by our proposed structure, but instead it can be improved (i.e. reinforcing its strength against power losses).

B. A particular situation: thin clients in a developing country

When we study the benefits of thin client technology we must know that this kind of computing is not only a way to get a lower cost. We can save money on hardware and maintenance, although for us remote **support** and **hardware recycling** are even more important. We have tried to develop a solution compatible with the requirements of an educational system but also available with the resources of a developing country.

Remote support: Thin client solutions offer a remarkable advantage on support: only the server needs to be updated or repaired (as the following section explains). This is basic when running a new educational project, the support (and teaching to the future local maintainers) are usually provided from other places (countries). We have adopted this solution, based on thin client technology, in order to centralize all processing in only one computer: the server. At the beginning of our pilot experience we kept a copy of this computer in our school at UPM, so we could try to reproduce and solve locally each failure in a handy way. Today, every time we have less and less support requests —less and less critical also— and we can deal with them without the local server.

Nevertheless, what we propose is not keeping forever this remote support but using it for the preliminary stage of the project. Later, the support will be provided by the local staff with the additional advantage of the centralized administration.

Ancient hardware recycling: Designing a device that fulfils our constraints —particularly with respect to the price— raises the question of what the requirements to build a thin client computer are. A minimal computer just requires one mainboard with a compatible microprocessor. User interaction requires input and output peripherals: monitor, VGA card, keyboard and optionally a mouse. Finally, by adding a network card to this minimal computer we obtain a thin client. Then, what are the benefits with a thin client? It seems that we have not made a great advance, but actually we have removed mechanical components —hardest hits for failures— like hard disks, CD and floppy units. So, we have also improved the reliability. But above all, a thin client does not require a high processing —note that it will not run anything additional to the hardware and software needed to present properly the graphical user interface—, so we can use low-performance hardware to build a thin client, i.e. "ancient hardware".

Available machines in developing countries use to be quite old fashion. Sometimes they come from donations of second hand hardware. Sometimes they were new but the budget does not allow renewing the machines frequently.

Thin client yields a new use for the outdated machines in developing countries.

C. Limits of the proposed solution

So far we have explained the benefits of the thin client technology. This solution has, however, some remarkable limits inherent to its structure —mainly network limitations—, as well as upper boundaries regarding reliability and performance.

In general, we could state that this technology is not useful when the clients, or only one of them, need to compute a high-performance process, a long time execution or high-resolution graphics. In more detail we could talk about:

- Multimedia: video and audio applications. As we stated above, the thin client is just a terminal for the applications, while all the processing is done at the main server. With multimedia, the information presented to the user is higher than with other kind of software, what in this technology means a lot of network traffic. Although maybe the server could support all the required processing, the network traffic generated will raise serious scalability problems.
- Computer design, animations, and games in general are similar examples to the multimedia affair: frequent changes on the screen increase the network bandwidth taken by the communication between client and server.

- High-performance applications. Obviously, the processing time available on the server must be shared by all the users. If this processing required is high the system cannot take advantage of this characteristic.

Notice that general educational purposes are not included in these limitations.

3. Proposal

As engineers we have worked in the design and development of the infrastructure needed to build an academic classroom environment with computers. Additionally, we have made some teaching work for the local staff. This environment is based on thin client technology that we have already explained, with an additional recovery system currently waiting to be improved.

All developed software has been developed by us and licensed under GPL as open source to be used freely for educational project.

A. Thin client software

Three basic modules were developed to build a full network solution using thin clients: a boot server, a thin client operating system (thin client OS in advance) and a centralized management application. The first two are required in any thin client solution, and the third was developed as complement to make the system more "user friendly".

Network boot server: In general the boot process in a thin client will be different from a usual boot. A thin client can be provided with an internal storage disk[3], then the client may boot using a local copy of its operating system, the thin client OS, but this is unusual. A thin client can take advantage of the existing network (because a thin client solution always demands a local network), so it may boot through the network connected to a server by using a special protocol (since in this scenario is not mandatory the use of internal storage in the client, often we refer to thin clients as diskless devices). There are several network booting protocols, although most of these protocols tend to work in a similar way. The boot process starts when the thin client ask for a network identification (an IP address, remember that it will be always connected), after that the operating system is downloaded by the client using a file transfer protocol[4]. Finally the thin client is ready to work with a local copy of its system. Two known boot protocols which work in this way are the bootstrap protocol, or BOOTP, and the PXE[5] protocol. The first and oldest uses a static addressing, manually assigned, while the second offers dynamic or different addresses allowing the automatic connection to new clients. We decided to use the newest version of existing boot protocols, meeting the following requirements for the server and clients:

- The server must be provided by two services: the Dynamic Host Configuration Protocol, or DHCP, and the Trivial File Transfer Protocol, or TFTP. The first service will be the IP-address provider, while the second will send the operating system and other configuration parameters to the clients.

[3] Hard disk, Disk on Module (DoM) or Compact Flash are the more usual storage systems used in thin client devices.

[4] Here we find one of the main characteristics of thin clients, since in each boot process the thin client OS is downloaded from a server, the operating system will be always an unchanged state, so that we can be sure that the system works except for hardware failures (this greatly reduces the time spent on maintenance).

[5] Stands for Preboot Execution Environment.

- Otherwise, in the client, today is usual to find a network boot protocol within the BIOS firmware (named as PXE or LAN boot). But it is not a limitation because we can find other ways to boot using the network, for example from floppy, CD or a common hard disk with a small boot loader [10].

Thin client OS: The thin client OS was developed from scratch, based on a Linux system. It was developed as an embedded system[6] to reduce the bandwidth used in the boot process —remember that the thin client OS is downloaded from a server in each boot—. Besides, because of the reduced size of an embedded system, it can also be stored in main memory (also known as RAM[7] memory) avoiding the use of a local or internal storage disk.

Management application: Server and clients' configuration can be modified directly from a file, but this is not a friendly configuration method. Thinking mainly in non-technically trained administrators, as may be a teacher, we have developed a graphical user interface to change every possible configuration parameter. Some of these parameters are, referfing to the clients: monitor frequencies, screen resolution or keyboard layout; and referfing to the server: network parameters for DHCP and TFTP services. In addition to the configuration, from the management application the administrator is allowed to: poweroff or reboot a client device, and close a user session.

B. Recovery software

During the life of an installed operating system it losses performance due to its use, and usually there is only one solution: a reinstallation process. This process is slow —it can take hours—, and sometimes it can also be a hard and tedious work —we must do a driver installations, user accounts management, a tuning of the operating system, and other remaining tasks after this new installation process—. To avoid this overload we have developed a software application which stores a local copy of the compressed image of a previously installed system. That copy is created in each computer, keeping a frozen image with the current state of the operating system properly configured. Afterwards the system can be recovered during the boot process, taking just a few minutes.

This software not only recovers a system from a compressed image stored in the local hard disk. It can also download the image from a server and apply it to the disk, allowing computers to be installed completely from scratch. This also allows the administrator to modify remotely some of these images in order to change the configuration of the thin clients.

There is an alternative hardware solution, recovery cards: not expensive but hardware-dependent and sometimes also operating-system dependent. We decided to reject this alternative because it is an exclusively commercial option, and it adds new critical hardware to the system that has to be taken into account to guarantee reliability.

C. A practical example

The work presented in this paper is not based on hypothesis, but rather on our practical experience. As mentioned before the developed software has been released as open

[6] An embedded system is a custom-built system, designed with a specific purpose, usually small and integrated within a hardware device.

[7] Stands for Read-Only Memory.

source and the activities described in this paper were put in practice by the TEDECO group from the Technical University of Madrid (UPM in advance). Since its inception back in 2006, TEDECO has been linked to the University of Ngozi (UNG in advance), in Burundi, and the cooperation group has been working to provide UNG with the means to support a Computer Science and Engineering degree from the technological and teaching point of view. From the beginning, we knew that our presence in Ngozi was not expected to last forever, and although we still keep supporting UNG actively, the university has made an enormous breakthrough by maintaining the computer classrooms completely —including hardware replacing—, and the computer centre mostly by themselves.

The infrastructure deployed at UNG consists of two computer classrooms, a computer centre and an Internet access point. One computer classroom was built by using thin client technology, and the other one in the usual way: with workstations. A total of 34 computers were divided between the two classrooms: 16 thin clients with two servers and 16 workstations. Both servers and workstations can be easily recovered using our aforementioned recovery software. On the other hand, we also designed a new computer centre which is now composed of three servers. These servers are connected through a local area network, and although some of them are critical, most functionalities are replicated so that a failure in one of them will not affect the whole network. All servers establish reverse SSH tunnel connections to servers located at UPM, in order to be accessed independently to perform maintenance tasks remotely. Finally, an internet access point is opened for people from outside UNG, with eight thin clients and a server. This service is provided by UNG for low fee. The software used to manage this cybercafe has been released as free software, and although includes a whole set of complete features, is under constant development to improve it and add new features. The cybercafe allows UNG to pay for the internet connection. So this provides economical sostenibility to the whole infrastructure.

This whole system is built on top of an internal network, which also gives service to other computers connected from different offices and departments of the UNG. The university has a 512 kbps satellite connection to the internet.

4. Conclusions

A. Advantages and disadvantages

Reliability: A thin client-based solution allows us to avoid operating system degradation in clients by using only one recovery system in the server which rebuilds an entire network in few minutes. This solution can also be improved with an easier and cheaper uninterruptible power supply, UPS: only one unit required for the thin client desktop server (after a power outage all clients will recover their working sessions). Finally, the solution may reach its highest robustness by adding an automatic backup system, which needs to be configured in only server.

A simple failure in the server hardware left the entire network without service. To avoid this, the recommendation is to have an additional computer with enough capabilities to be used as a temporal server.

Economy and ecology: The first installation of a thin client solution will be cheaper, with the disadvantage that a full network has to be connected and configuration could be somewhat more complex —depending on installer training—. The final price of this first installation can also be reduced with hardware recycling, by using less and oldest hardware.

On the other hand, this solution will decrease the power consumption and the expenses of adapting a new environment to the use computers. It also requires fewer personnel to maintain the whole system, most maintenance tasks can be performed remotely in the first stage and by one only local person in the following stage. It lessens the consequences of power failures and takes advantage.

Scalability: New thin clients can be connected without a required configuration change. Although there is a limit to the number of thin client simultaneously connected. This limitation depends on the server hardware characteristics[8], mainly: available memory or processor performance.

Performance: The main disadvantage of a thin client solution is the particular or single performance of each client. A thin client network has to share common resources, so every client must be generous with other potential clients. Generally this is not a problem in an academic environment[9].

On the other hand, we are taking advantage of every shared computer. Using thin client technology we are cloning each computer to serve more/many people, exploiting the full productive capability of one computer (new computers as servers and old ones as clients).

5. Future and related work

Our work on this field is still on process. First, we want to improve the system so that some peripherals can be connected directly to thin clients. Some solutions already exist for sound devices [8], [1]; and USB hard drives [3] mainly. We intend to support these features as well as modify some of them so that more devices are available.

The built solution based on thin client technology was slightly modified and adapted to be used in an open internet access point (cybercafe). Our daily work follows the development of an improved version of this cyber management software, freely available at SourceForge web page [9]. New features include accounting, reports and automatically transferring these reports by mail.

As pointed out in the introduction, power failures are common in these environments, and transportation costs are extremely high. This has led us to start designing a computer that fulfils more strict financial and technological requirements. The objective is to create a cheap small-factor solar-powered passively-cooled computer that runs in hostile environments with an extremely low failure rate and requiring no maintenance.

Finally, this system is not restricted to educational environments. We believe that similar designs can be set up in other cooperation facilities, providing excellent advantages especially in hospitals and other medical centres.

Acknowledgments

We would like to thank the University of Ngozi, the Bishopric of Ngozi, and in particular, three people: Apollinaire Bangayimbaga, Bonaventure Bangurambona and Trésor Tshisekedi Mutapay.

[8] Sometimes the allowed number of simultaneous clients depends on the software licenses hired under private software.

[9] We may find exceptions, i.e. an architecture course will be surely limited by the server and network performance.

REFERENCES

[1] Jack. audio connection kit. http://jackaudio.org/.

[2] Douglas Brinkley. Thin-clients in the classroom; software compatibility and a survey of systems. In Thomas Reeves and Shirley Yamashita, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006*, pages 383–390, Honolulu, Hawaii, USA, October 2006. AACE.

[3] Lyle Frost. Thinking thin. http://www.linuxjournal.com/article/9108, 2006.

[4] Jim Fulton and Ed Kramer. Can you ever be too thin? *netWorker*, 1(2):19–23, 1997.

[5] Jerry Golick. Network computing in the new thin-client age. *netWorker*, 3(1):30–40, 1999.

[6] Albert Lai and Jason Nieh. Limits of wide-area thin-client computing. In *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 228–239, New York, NY, USA, 2002. ACM.

[7] Albert M. Lai and Jason Nieh. On the performance of wide-area thin-client computing. *ACM Trans. Comput. Syst.*, 24(2):175–209, 2006.

[8] Pierre Ossman & Shahms E. King Lennart Poettering. Pulseaudio. http://pulseaudio.org.

[9] Jesús Martínez & Iván Pérez. Lan core: thin-client server, operating system and management application. http://lancore.sourceforge.net/, 2007.

[10] Etherboot/gPXE Project Team. Etherboot/gpxe rom images. http://rom-o-matic.net/.