

Our team accomplished many different things during the course of our project. There is a GUI, gXenophilia for managing nodes that run Xen, a distribution, Xenophilia which installs Xen kernels by default and comes prepackaged with Linux based domU domains, documentation on setting up and using Xen that is more in depth than existing documentation, and several prepackaged domU images for several different Linux distributions. Work has also started on an XML based domain creation utility.

There are several things that we will need to do in order to finish the system. Our Xen distribution is currently in beta and needs to be thoroughly tested. In addition to that, hardware detection needs to be added to the installer and support for initrd images should also be added. Support for deploying images and changing network configuration needs to be integrated with xend, and xend needs to support SSL encryption. One option for adding SSL support would be to use stunnel, or a similar solution.

Support for more flexible networking also needs to be added to both Xenophilia and gXenophilia. We would like to support NAT, Bridging, Routing, and private lans for domains. Support for this improved networking in Xenophilia will likely be implemented in robust shell scripts, and a dhcp server will also be added for possible use with NAT support. Xend will need to be slightly modified in order to support the different networking options. A network configuration dialog will also be implemented in gXenophilia to support all of these options. The XML based domain creation and image deployment utility will need to be completed. Currently it is non-functional, but an XML schema has been written, along with some parsing code. Currently the installation module has support for downloading and decompressing files, and for creating file systems that the system is running on supports. Several different tasks need to be added to make this functional. The XML parser needs to be completed, file system creation needs to be thoroughly tested, and lvm2 support needs to be added, and tested so that all virtual block device types are supported.

Our entire group learned a great deal about how to manage a large software project and use the necessary tools. Everyone learned how to use and deploy subversion, the version control system that we selected. We had to select a bug server to track problems in our software, so we learned how to set up Mantis and configure the necessary access controls on Apache and MySQL. We also learned a great deal about Debian, one of the largest Linux distributions. We learned how to make a derivative of Debian and how to create proper Debian packages.

While learning these systems, we gained experience on how to use sed, and awk, and write robust shell scripts. These scripts were used to track package versions and to help configure and install the packages that we had created. In the past, when programming GUIs we've used GTK+. Every widget, every relevant attribute of every

widget, and the way each widget interacts with other widgets had to be specified. Gxenophilia was written with Glade, Python, and pyGTK. This forms a formidable development kit. Progress went much quicker than our experience with GUIs in the past. This has taught us the importance of rapid GUI development. Without these technologies we would not have accomplished as much as we did in the time frame. Also, while designing the look and layout of the GUI, we realized the importance of HCI (human computer interaction). Previously we had not put much thought into the placement of widgets, but in reality this has a huge impact on the usability of a product. We also learned docbook XML, after starting with very little experience using it or writing formal documentation. By the end of the project we knew how to write good documentation that can easily be outputted into many different formats.

We used several external tools for different reasons. The biggest external tools that we used were Debian and Debian's new installer for Sarge. We decided to use these tools because we already had some knowledge of Debian and Debian packaging, and the Debian installer is much easier to customize than most other Linux installation solutions, since, we did not have the time or manpower to write our own installer. Another reason that we chose Debian to derive our system is that most Debian packages will work unmodified with our distribution. This allows us to focus on our interaction with Xen instead of packaging thousands of applications that are necessary to include in most Linux distributions. The third reason that we derived from Debian is that is noted for its stability and security. This will allow us to have much higher up-times than if we used a different distribution to derive or created our own distribution from scratch.

Another tool that we used Python and PyGTK because they are cross platform technologies and allow for rapid application development. Given our short time frame we needed tools that would allow for us to rapidly develop our application and give us time to thoroughly test it. Another reason that we chose python is because it is the language that most of the Xen configuration tools are written in and it will help us get up the learning curve of a new language when we start integrating more of our work with xend.

For our documentation we decided to write all of it in Docbook XML. This is because it can easily be transformed into many different formats including html, xhtml, pdf, postscript, and plain text. It also has many nice features in its tagset including support for examples, and screen output.

Once completed our project will bring several major benefits to enterprise users. Administrators will be able to easily deploy new systems at the touch of a button, and without any downtime. For a domain running a complex web domain downtimes from a live migration of a domain to a different computer can be as low as 210ms, which is short enough for all of the client connections to be maintained

according to the Live Migration of Virtual Machines paper published by the University of Cambridge Computer laboratory. Our custom installation CD, with support of Debian pre-seeding, will allow for rapid deployments of large numbers of Xen systems. Administrators will no longer have to create their own Xen packages and have them installed after the system has been installed. Gxenophilia our graphical tool will allow system administrators to easily administer Xen nodes without having to use SSH or a custom tool. This will lower the deployment and administrators time and allow administrators to be more productive, from the desktop environment of their choice.