Deployment of Grid gateways using virtual machines

Stephen Childs, Brian Coghlan, David O'Callaghan, Geoff Quigley, John Walsh
{childss, coghlan, ocalladw, gquigle, walshj1}@cs.tcd.ie

Department of Computer Science
Trinity College Dublin, Ireland

Abstract. Grid-Ireland, the national computational grid for Ireland, has a centrally managed core infrastructure: the installation and configuration of grid gateways at constituent sites is controlled from an Operations Centre based at Trinity College Dublin. This structure has necessitated the development of tools to automate and simplify the deployment of middleware to the sites. Virtual machine (VM) technology has performed an important role, allowing us to maximise the utilisation of server hardware and to simplify installation and management procedures. In this paper, we present an evaluation of competing VM technologies and relate our experience with virtual machines to date. We have used VM technology to reduce the hardware requirements for access to the Grid: by running multiple OS instances a full Grid gateway can be hosted on a single computer. This has significantly reduced the hardware, installation and management investment needed to deploy a new site. We describe a single-computer Grid gateway based on the Xen VM system which we plan to deploy to eight new sites in early 2005.

1 Introduction

1.1 Context

The Grid-Ireland project provides grid services above the Irish research network, allowing researchers to share computing and storage resources using a common interface. It also provides for extra-national collaborations by linking Irish sites into the international computing grid. The national infrastructure is based on middleware from the LHC Computing Grid (LCG) project [1]. LCG provides a common software distribution and site configuration to ensure interoperability between widely distributed sites. LCFGng [2] allows network installation of nodes according to configuration profiles stored on an install server.

Grid-Ireland currently comprises an Operations Centre based at Trinity College Dublin (TCD) and six nationwide sites. The Operations Centre provides top-level services (resource broker, replica management, virtual organisation management, etc.) to all sites. Each site hosts a Grid access gateway and a number of worker nodes that provide compute resources. We aim to make Grid services accessible to a far higher proportion of Irish research institutions in the

near future. To achieve this goal we must ensure that the hardware and personnel costs necessary to connect a new site to the Grid are not prohibitive. A standard LCG gateway configuration makes significant hardware demands of a site. A minimum of four dedicated machines are normally required: an install server, providing a configuration and software repository for all nodes; a computing element (CE), providing scheduled access to compute nodes; a storage element (SE), providing data management, and a user interface (UI) providing for job submissions from users.

We have already deployed VM technology at six sites: all UIs are running as User-Mode Linux (UML) VMs on the SE server. We now wish to deploy to eight new sites, and based on positive experience with the use of VMs, we are currently developing a software distribution that will allow us to run all gateway services for a site on a single computer. This machine will host a number of VMs acting as logical servers, with each VM running its own OS instance. As each VM would appear to be a real machine (both to the server software and to users), there will be no need for special configuration relative to the existing gateways.

1.2 Aims

We aim to make Grid deployment more cost-effective by using VM technology to reduce the number of machines that need to be dedicated to a Grid gateway. We will also develop tools that allow administrators to automate the initialisation and configuration of the VMs, thus simplifying the installation process. We also aim to limit the divergence from a standard Grid site configuration so that the same configuration data (LCFG profiles, software package lists) can be used for all gateways, whether they use VMs or not. We also provide for central management of remote sites: each of the servers must be accessible via both console redirection and Secure Shell (`ssh`). Finally we must provide a simple installation process that can be performed remotely.

1.3 Outline

In the remainder of this paper we describe the advantages of using VM technology to build and deploy Grid gateway services on remote sites. In Section 2 we discuss the factors determining the choice of a good VM system, in Section 3 we briefly describe the architecture of a Grid gateway built on VMs, and in Section 4 we describe the tools we have developed to aid deployment. In Section 5 we outline the installation procedure used to roll out new sites, and in 6 we make some observations based on our experience of deploying VM technology. Section 7 discusses related work, and finally Section 8 summarises our findings.

2 Choosing a VM system

Making a good choice of VM technology is crucial to building a secure, fast system that is easy to manage. We briefly describe a range of currently available

VM systems, and choose representative VM systems for evaluation. We also outline the technical and administrative requirements demanded by the task of Grid deployment.

2.1 Overview of VMMs

A virtual machine system provides each user with a complete OS environment tailored to his applications and isolated from other users of the computer. The virtual machines are controlled by a monitor (VMM), which enforces protection and provides communication channels. In the past, VM technology was most widely applied in mainframe computing, for example in IBM's VM/370 system [3]), where it was used to allow many users to share the resources of a single large computer.

Recently, interest has grown in implementing VMMs on commodity hardware and the past few years have seen a stream of commercial and open-source VMMs which provide varying levels of virtualisation. Full virtualisation virtualises a complete instruction-set architecture: any OS that will run on the underlying hardware will run on the VM. Examples include VMWare [4], a commercial product that provides full x86 virtualisation on both Windows and Linux. Para-virtualisation presents a modified interface to guest OSes, which must be ported to the new VM "architecture". Xen [5] is a para-virtualised VMM which supports Linux and BSD-based guest OSes. Finally, system call virtualisation provides an application binary interface that enables guest OSes to run as user-space processes. User Mode Linux (UML) [6] is a port of the Linux kernel to run in user-space; it can be run on an unmodified host OS although kernel modifications are available that improve performance.

Our evaluation has focussed on Xen and UML. These are both open-source projects, allowing us to customise the code if we need to. They are also stable projects with active user communities to provide support. Both Xen and UML are compatible with the LCG software: LCFG profiles can be modified to selectively install custom kernels on the correct machines. UML VMs can be run on an unmodified Linux kernel, although specially patched kernels may be used to improve performance. We have excluded commercial VMMs from our experiments due to cost considerations and licensing restrictions.

2.2 Requirements

We aim to run all gateway services quickly, securely and reliably on a single machine and so require a VMM to provide the following features:

Isolation: In a single-box solution, it is important for the VMM to provide isolation between VMs, so that even a catastrophic OS failure in one VM will not affect the others. (A hardware failure will inevitably affect all hosted OSes, but this risk could be mitigated by providing a backup machine to act as a failover.)

Storage: The logical servers each have different storage requirements but must share a limited set of local disks: the VMM should provide a flexible means of

sharing the available disk space between hosted nodes to reduce the need for tricky repartitioning in the case of file systems filling up.

Resource control: The various servers also have different CPU requirements: the VMM should provide a means for controlling CPU utilisation. For example, to preserve interactive performance on the UI it may be necessary to throttle back the CPU utilisation of the other nodes. It would also be useful to be able to partition other resources such as disk and network bandwidth and physical memory.

Low overhead: The VMM should not impose a high performance overhead or significantly reduce system reliability. This is particularly an issue during I/O intensive operations such as installation/upgrade: while almost all VMMs can run compute-bound code without much of a performance hit, few can efficiently run code that makes intensive use of OS services. As the gateway will host the User Interface, the VMs must provide good interactive response times.

The VMM should also provide features to facilitate management of VM nodes. Such features typically include access to consoles for each VM, a facility for storing VM configurations, and tools for displaying and controlling VMs' resource usage.

2.3 Performance evaluation

Detailed performance measurements of Xen, VMWare and UML can be found in the main paper describing Xen [5]. The results show that Xen consistently out-performs the other VMs: by a small factor for computation-intensive applications, and by a very large factor for applications using I/O and other OS services.

We have also performed our own measurements of Grid applications; full results may be found here [7]. Figure 1 shows the outcome of tests recording the duration of a first-phase LCFG installation of a UI node — a procedure including the creation of file-systems and the installation of the Red Hat 7.3 OS and LCG middleware (over 700 RPM packages). This procedure takes around seven minutes on a Xen VM, but over an hour on UML. The results of this performance evaluation have led us to migrate our existing UML-based VM setup over to Xen.

3 System structure

Figure 2 shows the structure of our main VM-based application: a single-computer Grid gateway. Each server runs in its own OS on a separate virtual machine: the LCFG install server runs on the first VM and the other servers (CE, SE, UI and WN) run in VMs with filesystems hosted in loopback files on the install server's file system. Xen provides a virtual network interface for each of the VMs. These are bridged onto the real Ethernet card using standard Linux utilities, providing direct network connections. Further details may be found in [7].

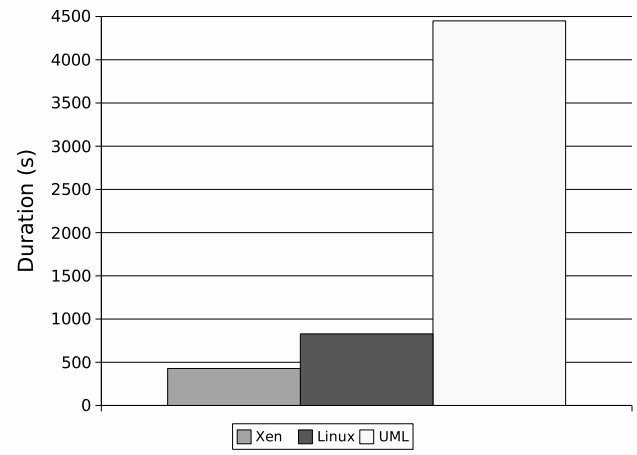

Fig. 1. LCFG installation of UI node

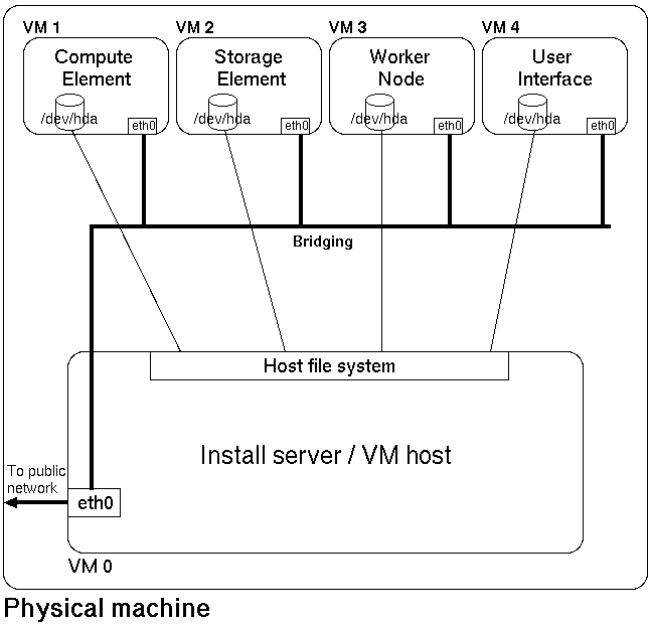

Fig. 2. Architecture of grid gateway

4 Tool support

4.1 Control tools

If a VM system is to be deployed across multiple system, it should be possible for administrators to control its operation without needing detailed knowledge of the parameters to the VM system. For this reason, is important to integrate the VM system with standard system service control tools. This is most easily done by writing scripts to wrap the tools provided by the VM software.

We have already developed control tools for our UML-based VMs which have been deployed for some time now on the national infrastructure. There are two main components: a service control script used for integrating the VM system with standard boot service configuration, and a command-line tool designed for run-time control of VMs.

The service control script allows for all confgured VMs to be automatically started or stopped in a particular run-level. A list of VMs is stored in a file and for each VM, a configuration file contains network settings stored as simple shell variables. The control scripts read the configuration variables and creates an appropriate command line for the VM control system.

The principle behind the development of these tools is to provide a level of abstraction that allows administrators to control the gateway services in the same manner as they would other standard services. Configuration settings can be set using standard shell variables: the administrator does not need detailed knowledge of the VM command line parameters.

4.2 Remote management

There are typically two levels at which remote management must be provided for VMs: access to the machine itself at a low-level, and access to each of the VMs. The first is typically provided by dedicated hardware in the server machines which enables management features such as remote console access over a dedicated serial or Ethernet connection and console redirection allowing access to BIOS settings. The second must be provided by the VM system itself. UML VMs are in fact standard Linux processes, output appears directly on the console where the VM was created. However, when multiple VMs are running, it is convenient to be able to attach to the console later on. We start each VM within an instance of the `screen` utility and give them an unique name that allows us to reconnect later on as necessary. Xen provides its own tools for console-level access: the complete boot sequence for each VM can be observed via a command-line application which can be attached or re-attached at any time.

5 Installation procedure

In order to deploy software to eight new sites around the country, it was important to develop a simple installation procedure. We took a three-phase approach:

firstly we installed a base system using a combination of manual installation and the LCFG install tool, then we took an image of that system and copied it to the target machines' disks, and finally we performed reconfiguration of the new servers to prepare them for installation at a specific site.

5.1 Installation of base image

The base OS (Red Hat Linux 7.3) was installed from CD, and then the LCFG server software was installed according to the standard procedure. At this stage, we upgraded a number of packages to provide newer versions of software required by the Xen software. We then installed a custom RPM package which included Xen kernels compiled with support for the hardware on the target machine. Changes were made to the LCFG configuration to support the Grid-Ireland layout, and profiles for the various nodes to be hosted were retrieved from a CVS repository.

The next step was to install the various nodes using LCFG. This involved various modifications to the standard LCFG install procedure due to the fact that these were virtual machines rather than real physical machines. Normally, LCFG-installed machines network boot using PXE: this isn't necessary with VMs, as the VMs are started from the command line. Also, network settings can be specified directly, eliminating the need for DHCP. Once the necessary modifications were made to the LCFG install script, the VMs were booted with a root file-system set to the LCFG installation directory, and executed this script as `init`.

The LCFG installation process installs the Red Hat Linux 7.3 operating system and the packages making up the LCG middleware: a total of 700-800 packages depending on the configuration. The first phase of the installation process takes around 7 minutes on a Dell PowerEdge 1750 2.4 GHz machine running Xen. Once all nodes (CE, SE, UI and WN) had been successfully installed, we ran tests to confirm that they were operating correctly. We then shut down the virtual machines and dumped the complete filesystem to a compressed image file, which was approximately 6.6 GB in size.

5.2 Imaging target machines

This image file was then stored on a portable hard drive, which we then used to install the target machines. We used the standard `dump` format so the file system could be restored to any disk of a sufficient size.

5.3 Configuration of new servers

The system image transferred to the target machine contains many settings that refer to the original installation, and these have to be updated to reflect the desired configuration. Ideally, this would be simply a case of pointing each

individual virtual machine to its new profile, and then allowing LCFG to reconfigure the system. In practice, we have found it simpler to perform some pre-configuration on the new file system before starting the boot process.

The basic steps needed are to update the network settings, to copy the new LCFG profile and to compile it. In the host OS, we mount the filesystem and edit the network configuration files to reflect the new identity of the machine. We then copy the XML version of the profile to the correct location on the VM filesystem. We use the `chroot` program to run the profile compiler within the VM filesystem so that the compiled profile ends up in the right place. All these steps are scripted so that they can be included in an automated process.

When the virtual machine is booted, the LCFG client reads the new profile and performs any reconfiguration necessary. Extensive changes should not be necessary as the main differences between server installations at different sites are the network settings, which we have already modified at this stage. Once the new VM is up and running, a small number of manual steps still need to be performed: this is because LCFG objects have not been written for all system components.

5.4 Deployment to sites

All previous steps can be carried out at the Operations Centre (subject to site managers providing the necessary information about their site: network address settings, etc.). The actual procedure of installation at the site should be straightforward: the main task is to provide network and power connections for the gateway machine.

6 Experience with virtual machines

We have been using UML for the past year to share a single machine between an SE and a UI; this configuration currently runs on five sites nationwide. As a result of performance evaluations [7], we are switching to Xen for the next phase of gateway rollout. The performance overhead due to UML, while just about acceptable for use on a single node, is too high for our target of five VMs per computer. UML is around ten times slower than Xen for OS-intensive tasks, making node installations and upgrades lengthy processes. Xen is also more responsive than UML during interactive use.

There are also management benefits: as the VMs' file systems are stored as regular files on the host, we can easily back up an entire site gateway by dumping the host file system. VMs also ease site installation as only a single machine needs to be provided with network connection and power. Installation of individual servers is also more manageable. Even with network installation, unforeseen issues often arise that require physical access to the machines. With VMs this doesn't arise: once the host is up and running, all servers can be easily installed and accessed from the command line.

Full remote console access is more easily arranged with VMs than with real machines. With real machines, access to BIOS settings and boot menus is only possible on motherboards with extra remote management hardware. Even when these features are available, they often require extra network or serial connections to the machine, and can provide a somewhat limited interface. With virtual machines, BIOS settings are simply not an issue (there is no BIOS!), and both Xen and UML provide full console redirection that allows the entire boot process to be monitored.

We have also found that the use of VMs greatly eases the task of reconfiguring existing sites. For example, we recently decided to deploy a test worker node at each site to allow administrators to run tests without disturbing existing cluster nodes. By running this new system as a VM, the procedure proved very straightforward. We simply took an file system image (actually of a UI system — the process would be even easier using a WN image), copied it to a new machine, booted a new VM from this image, and then reconfigured the VM with a new profile and network settings. These operations were all performed remotely over SSH without any need for physical access to the site.

7 Related work

The work of Figueirdo *et al* [8] is complementary: they propose the use of virtual machines for Grid worker nodes whereas we use VMs for the gateway servers. They aim to support a variety of guest operating systems and so choose a VMM that supports full virtualisation. Other sites within the LCG collaboration have explored the use of VMs: the London e-Science Centre have used UML to provide an LCG-compatible environment on existing cluster machines [9], and Forschungszentrum Karlsruhe have used UML to host their install server [10]. To our knowledge, no-one else has implemented a complete site gateway using VMs. Outside the Grid community, the XenoServer [11] and Denali [12] projects both use VM techniques to support dynamically instantiated application environments for remote users.

8 Conclusion

The use of virtual machines is central to the speedy deployment of new sites in the Grid-Ireland architecture. We have found that using VMs reduces hardware costs, enabling more sites to be deployed. VMs also speed up deployment as an entire gateway configuration can be imaged onto a single computer. Remote management is also easier with VMs: full console access is available without the need for specialised hardware and software.

We have demonstrated that it is feasible to construct a single-machine Grid gateway using virtual machines. However, our experiments show that the choice of VM technology is crucial. User-Mode Linux, while in widespread use, is impractical for our purposes due to its extremely high overhead for OS-intensive tasks. Xen, in contrast, performs well across a range of applications. Because

Xen provides an OS environment that is indistinguishable from a regular OS instance, software servers can be run with the same configuration as on dedicated machines.

We have already experienced benefits by using VM technology in our site installations. The solution described here will allow rapid deployment of new gateways, enabling a significant increase in Grid participation. We believe that this approach will be of interest to many sites wishing to connect to the Grid for the first time.

Acknowledgements

We would like to thank the Xen team at the University of Cambridge for developing the Xen VMM and for providing useful support. Dell Ireland kindly donated the hardware. The original UML work drew on a configuration by David Coulson (available at `http://uml.openconsultancy.com`).

References

1. LCG: LHC Computing Grid Project (LCG) home page. `http://lcg.web.cern.ch/LCG` (2004)
2. Anderson, P., Scobie, A.: LCFG — the Next Generation. In: UKUUG Winter Conference, UKUUG (2002)
3. Gum, P.H.: System/370 Extended Architecture: Facilities for Virtual Machines. IBM Journal of Research and Development **27** (1983) 530–544
4. Devine, S., Bugnion, E., Rosenblum, M.: Virtualization system including a virtual machine monitor for a computer with a segmented architecture. US Patent (1998)
5. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the Art of Virtualization. In: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, ACM (2003)
6. Dike, J.: A user-mode port of the Linux kernel. In: Proceedings of the 4th Annual Linux Showcase & Conference, Atlanta, USENIX (2000)
7. Childs, S., Coghlan, B., O'Callaghan, D., Walsh, J., Quigley, G.: A single-computer grid gateway using virtual machines. In: Proceedings of the Conference on Advanced Information Networking and Applications (to appear). (2005)
8. Figueiredo, R.J., Dinda, P.A., Fortes, J.A.B.: A Case for Grid Computing on Virtual Machines. In: Proceedings of the International Conference on Distributed Computing Systems. (2003)
9. McBride, D.: Deploying LCG in User Mode Linux. (`http://www.doc.ic.ac.uk/~dwm99/LCG/LCG-in-UML.html`)
10. Garcia, A., Hardt, M.: User Mode Linux LCFGng server. `http://gridportal.fzk.de/websites/crossgrid/site-fzk/UML-LCFG.txt` (2004)
11. Fraser, K.A., Hand, S.M., Harris, T.L., Leslie, I.M., Pratt, I.A.: The Xenoserver computing infrastructure. Technical Report UCAM-CL-TR-552, University of Cambridge Computer Laboratory (2003)
12. Whitaker, A., Shaw, M., Gribble, S.: Denali: Lightweight virtual machines for distributed and networked applications. In: Proceedings of the USENIX Annual Technical Conference. (2002)