Deploying and Managing Xen Sites with XSM *

Felipe Franciosi[1]   Jean Paulo Orengo[1]   Mauro Storch[1]   Felipe Grazziotin[1]
Tiago Ferreto[2]   César De Rose[2]

[1] PUCRS/HP Research Center
Porto Alegre, Brazil
[2] Catholic University of Rio Grande do Sul
Computer Science Department
Porto Alegre, Brazil

Abstract. Xen is becoming a de facto solution for virtualization with low performance overhead. Virtualization features include optimization in resources utilization, server consolidation, improved security and fault tolerance. Server consolidation is currently one of the main attractions of virtualization for enterprises. It enhances the scalability of a regular IT infrastructure, enabling the utilization of an amount of servers larger than available physical machines. However, deploying and managing virtual machines in such environment can be a complex task. It's first necessary to install the virtualization layer, represented by the VMM (Virtual Machine Monitor), in each physical machine. After that, the virtual machines can be deployed. Due to the possibility of having several virtual machines inside a single physical machine, it's required an efficient management of the environment in order to enhance overall system performance and resources utilization. Considering these issues, we present in this paper the Xen Site Manager (XSM), an open-source tool to facilitate the deployment and management of virtual sites based on Xen. It interacts with standard services and tools, such as SystemImager, DHCP, PXE and Ganglia, in order to provide high flexibility. We provide a detailed description of XSM architecture and present a performance evaluation of its deployment feature.

1   Introduction

Virtualization is one of the latest trends in the IT market. It's becoming a common standard due to the increase of commercial and open-source software products (e.g., Xen [1], VMWare [2], OpenVZ [3], Virtuozzo [4]) and investments of major hardware companies in providing mechanisms in their newest products to give better support to virtualization (e.g., Intel VT [5], AMD Virtualization [6]). Despite being an old concept (used initially by IBM 370 mainframes [7]), virtualization is proving to be an efficient solution to current problems faced by enterprises. Some of these problems are: exponential increase in the number of resources owned by organizations, high management costs, inefficient utilization of resources, etc.

One of the sectors presenting the most increase in the utilization of virtualization techniques is data centers. Typical data center problems fit well in the types of problems that virtualization intends to solve. Usually, there is a huge number of services being executed on sub-utilized resources, which results in high management costs. In order to obtain an efficient utilization of all computing infrastructure available and provide good performance to all services, virtualization techniques must be applied.

In spite of virtualization seeming to be a de facto solution to data center problems, it's utilization isn't straightforward, specially in environments with a huge number of resources. Deploying and managing a virtualization solution is currently a problematic task due to the nonexistence of flexible tools that provide a complete creation of a manageable virtualization environment. Current tools, such as Oscar [8], XenMan [9], Enomalism Virtualized Management Console (VMC) [10] don't provide all the necessary features, from the installation of the virtualization layer in each resource, til the creation, removal, migration and monitoring of virtual machines in the environment.

Due to the lack of a flexible tool to deploy virtualization sites, we developed the Xen Site Manager (XSM). XSM is an open-source tool to deploy and manage virtual environments. It is based on the Xen Virtual Machine Monitor [1], an open-source solution for virtualization. Xen is becoming a standard solution for virtualization due it's low performance overhead to execute virtual machines. XSM is highly flexible in deploying virtual machines with different configurations in a regular site. It uses standard services and tools such as DHCP (Dynamic Host Configuration Protocol), PXE (Preboot Execution Environment), SystemImager [11] and Ganglia [12].

The remainder of this paper is organized as follows. Section 2 describes the XSM tool presenting all of its features. Section 3 presents a performance evaluation of XSM showing the time it takes to deploy variable configurations of Xen Sites. Section 4 presents some related work. Finally Section 5 concludes the paper and presents some future works.

2 Xen Site Manager

When required to install a set of Xen servers in a data center, it is natural that a site administrator will look for tools to assist him with the job. Besides the help needed to install the machines, assistance to manage the virtual hosts is also required. That was the main motivation to develop XSM, an open-source tool for deploying and managing Xen virtual machines in a site.

In order to fulfill those tasks, XSM uses available tools such as: SystemImager [11], Ganglia [12] and XM (Xen Management User Interface). While Sys-

temImager and Ganglia are standard solutions for deploying and monitoring systems respectively, XM is a tool that comes along with Xen itself and is used to manage virtual machines. Figure 1 illustrates the typical lifecycle of XSM utilization, showing the concept of deploying the site, managing virtual machines and monitoring the environment. At any time, management of the previously installed hosts or deployment of new machines can be performed.
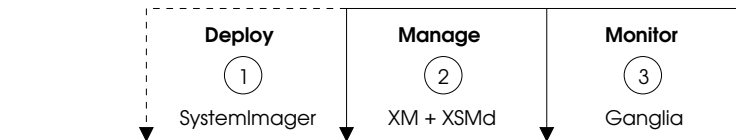
Deploy	Manage	Monitor
①	②	③
SystemImager	XM + XSMd	Ganglia

Fig. 1. XSM lifecycle

Jointly with the tools, XSM interacts also with another component, called Xen Site Manager Daemon (XSMd). This component was developed in order to perform management tasks in all hosts. It's based on a distributed management strategy. Each host contains a description of the whole site configuration. All XSMd communicate with each other in order to maintain a common vision of the site. The site can be managed from any host. When the site configuration changes, the configuration is modified in all other hosts. When the host receives the new configuration, it verifies if there is any local modification to be performed, and acts accordingly. This model excuses the need of a centralized management server, once any host may assume this role. Besides that, if any host goes down, the site will be manageable through any other host in the site.

2.1 Deployment

When installing more than one computer with the same system, automatic deployment becomes an attractive choice if compared to the manual installation of each host. XSM facilitates this process by using SystemImager and the PXE facility available in most computers. Figure 2 illustrates the boot process of a new computer involving the installation of the base system.

As the figure suggests, at least one auxiliary server is required for the initial deployment process. This server must run DHCP, TFTP (Trivial File Transfer Protocol) and SystemImager services. The first step consists of the new computer booting and, using PXE, fetching its configuration from the DHCP server. This configuration specifies, besides basic network configuration, the IP address of the TFTP server and the filename of a PXELinux image, which is downloaded in the second step. Next, this image is executed and the SystemImager client is
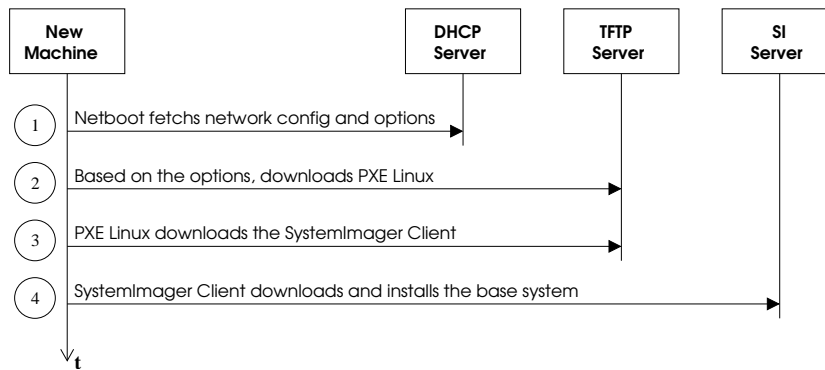
Fig. 2. Boot process using XSM deployment tools

downloaded. Step four consists of SystemImager client connecting to the server and fetching the base system, which is copied to the disk using partitioning information also specified by the server.

The base image supplied by XSM consists of a Linux Debian Etch as Domain 0. Naturally, this installation already ships a pre-configured Xen, Ganglia, XSMd and other Linux standard tools. As for the host configuration, SystemImager is also capable of automatically setting network interfaces (using DHCP or static setup) and hostname parameters.

As for the installation methods, SystemImager can use three different methods: unicast, multicast and broadcast. The choice depends on the site situation. With unicast, the SystemImager client will use the rsync tool to fetch the base image from the server. This method is well suited for a small number of hosts. When installing a larger number of hosts, the other methods, multicast and broadcast, are preferred due to the increase in network overhead with unicast.

2.2 Management

After completing the deployment process, it is possible to manage the Xen site. XSM is capable of starting, stopping and migrating Xen VMs via a site configuration described in an XML file. An example of the site configuration file is presented in Figure 3. The example shows a site composed by 4 hosts, with one virtual machine per host (vm1, vm2, vm3, vm4). Virtual machines vm1, vm2 and vm3 are composed by Apache servers, and vm4 is a MySQL server. Hosts are identified by the IP address in the *DOM0* parameter. The parameters to configure a virtual machine are: virtual machine name, virtual machine configuration (kernel and image details), location (Domain 0 IP address), number of virtual CPUs, MAC address, method to obtain network configuration (static or

DHCP) and physical memory. The virtual machine configuration represents the name of a regular Xen virtual machine configuration file which contains details to execute the virtual machine, such as: kernel, image, disks configuration, etc. Figure 4 presents the environment after the processing of the configuration file by the XSMd in each host.

```
<XSMConfig>
 <Set name="My_site">
  <Owner>root@domain.com</Owner>
  <VM name="vm1" config="apache_svr">
   <DOM0>192.168.5.101</DOM0>
   <VCPUS>4</VCPUS>
   <HWADDR>AA:00:00:00:00:01</HWADDR>
   <IP>DHCP</IP>
   <MEM>96</MEM>
  </VM>
  <VM name="vm2" config="apache_svr">
   <DOM0>192.168.5.102</DOM0>
   <VCPUS>4</VCPUS>
   <HWADDR>AA:00:00:00:00:02</HWADDR>
   <IP>DHCP</IP>
   <MEM>96</MEM>
  </VM>
  <VM name="vm3" config="apache_svr">
   <DOM0>192.168.5.103</DOM0>
   <VCPUS>4</VCPUS>
   <HWADDR>AA:00:00:00:00:03</HWADDR>
   <IP>DHCP</IP>
   <MEM>96</MEM>
  </VM>
  <VM name="vm4" config="mysql_svr">
   <DOM0>192.168.5.104</DOM0>
   <VCPUS>4</VCPUS>
   <HWADDR>AA:00:00:00:00:04</HWADDR>
   <IP>DHCP</IP>
   <MEM>96</MEM>
  </VM>
 </Set>
</XSMConfig>
```

Fig. 3. XSM site configuration file

When the site configuration file is modified, the XSMd reads the new configuration and forwards to other hosts. Each host reads the new configuration
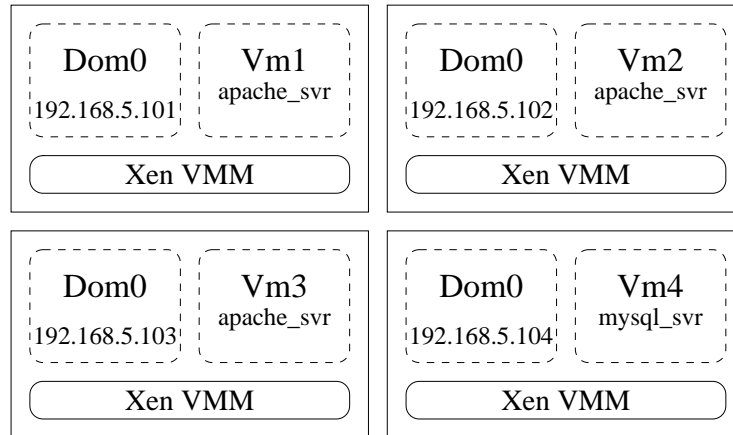
Fig. 4. Environment configuration

and performs the necessary modifications. XSMd interacts with XM in order to create, remove, reconfigure and migrate virtual machines. Virtual machines creation is performed when inserting a new virtual machine entry in the file. The host that will contain the new virtual machine reads the file, creates a new Xen virtual machine configuration file with the generic configuration (virtual machine configuration) and the virtual machine parameters (VCPUs, HWADDR, IP and MEM), and starts the virtual machine. To remove a virtual machine, it's just necessary to delete the virtual machine entry in the file. Reconfiguration is performed modifying the values of the parameters, and migration changing the host (Domain 0) IP address of a virtual machine.

This distributed approach was chosen prior to a centralized management server in order to avoid a central point of failure and to simplify the development of the system according to an already existent infrastructure in our data center. Another advantage of such method is that no additional servers are needed for this purpose.

2.3 Monitoring

To keep track of all site state we use Ganglia, a scalable distributed monitoring system that was designed to monitor clusters and grids state [12]. On these environments, scalability, reliability and heterogeneity are common issues. To work with scalability, Ganglia was structured as a distributed system, with decentralized control. To detect a component failure, Ganglia uses a heartbeat protocol. It was also implemented and deployed over many different operating systems as Linux, FreeBSD, Solaris and IBM AIX.

Ganglia encompasses three main components: gmond, gmetad and gmetric. Each host of a XSM site runs gmond, which is responsible for collecting information about CPU usage, free memory, and other monitoring metrics. It also sends this information to a well-known multicast address every time significant updates occur. All hosts listen for metrics on the multicast address and collect and maintain monitoring data for all other hosts. Therefore, in case of failure, the view of the entire site state can be reconstructed easily because every host has an approximated view of the environment. In addition, Ganglia allows the user to easily expand the core metrics adding any arbitrary host metric using the gmetric component. On this work, we use gmetric to add information collected from the XM command about every VM including their current state and resource usage.

Besides the common metrics included in Ganglia, the metrics collected for each virtual machine and Domain 0 are: CPU time and weight, maximum amount of memory and used memory, number of VCPUs (online and available), and status (blocked, ready, etc). Every metric is obtained through XM. Ganglia performs the periodic capture of these metrics and presentation with graphs showing a historical perspective.

3   Performance Evaluation

In order to evaluate the deployment performance, we built a testbed composed of sixteen machines and one server. The machines share a common Fast-Ethernet Network (100Mbps). The server is responsible for executing basic services that provide support to the deployment system. This server is a Dual Pentium III 1 GHz, 512 MBytes RAM running Debian Sarge 2.6 and the following services: DHCP, TFTP, rsync, and SystemImager. The machines used as installation targets are all Intel Pentium 4 1.6 Ghz with 256 MBytes RAM and 20 GBytes HDD.

The experiment intends to analyze the performance of deploying a variable number of machines with our base image using two deployment techniques: unicast and multicast. The broadcast method was not measured because, to what is network concerned, it's results would be the same as the multicast scenario. The size of our base image is 920 MBytes and, as described before, it consists of a Debian Etch Linux with Xen 3.0.1 and other software packages. It is also important to report that every mesurement was executed 10 times, discarding the slowest and the fastest executions.

Figure 5 presents the results achieved. As expected, we obtained better results with the unicast method when using a small number of machines, once the installation is straightforward using TCP packets. The multicast installation uses
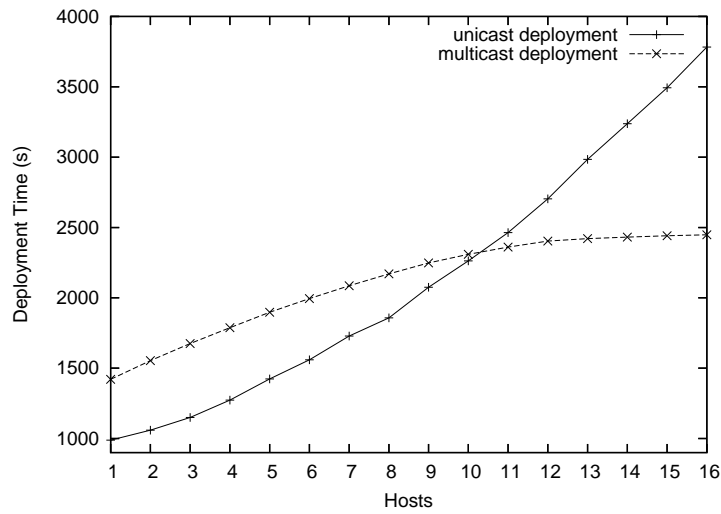
Fig. 5. XSM deployment performance evaluation

UDP packets, which is unreliable and requires synchronization stages and error correction with retransmission, causing overhead for a single host installation. Nevertheless, this overhead is softened when installing multiple computers, because there's only one packet sent no matter the number of hosts involved in the process (considering no retransmissions).

This explanation helps understanding why the multicast curve looks logarithmic. That's because some computers may perform the installation faster than others, lagging the synchronization stages. With several similar computers in the deployment process, these stages are likely to be reduced, once the speed of the fastest and slowest computers is not very different.

4 Related Work

The Xen project, although younger than its counterparts, was well received by the research community which elaborates many other projects to facilitate the deployment and management of Xen VMs as XenMan, Enomalism and Oscar.

The XenMan is an open source project that aims to provide an intuitive graphical user interface (GUI) for both administrators and new users of Xen. This interface, similar to the one provided by VMWare, can be used to create, start and stop VMs using buttons and fancy icons, making the management process easier. Although all the facilities supplied by XenMan, it was developed to work in one computer alone, thus this tool is not feasible to manage one entire site.

Oscar, on the other hand, can be used to deploy a cluster composed of Xen VMs in one or more computer hosts. Oscar is a well-know project in distributed computing [13], used to assist the installation and management of Beowulf clusters. Recently, the Oscar suite was improved to allow the creation and deployment of Xen VMs, aiming to turn effortless their suite testing procedure. Using this tool one can install Xen over a new machine, configure and create many VMs and also run them after the installation process. This tool can address the deployment step on the XSM lifecycle presented in Figure 1, however, cannot handle neither the management nor the monitor steps.

A more complete tool that can manage and monitor Xen sites is the Enomalism Virtualized Management Console (VMC). The VMC provides a web page interface to monitor the VMs, showing the CPU and memory usage among other information. With the same interface one can add new virtual machines or users and also manage them through a console emulator. To do these tasks the administrator need to be connected to the machine he wants to manage. In this way the VMC project is similar to XSM. However, the former is unable to deploy new machines and was built on a centralized point of view. The main advantages of XSM are that it assists the deployment steps and can handle failures because of the distributed approach used to manage the whole site. Moreover, the administrator can manage all the hosts by accessing only one of them rather than connecting to each host one by one, which is inadequate when there are many machines.

5 Conclusions and Future Work

In this paper we presented XSM, an open-source tool to deploy and manage Xen Sites. XSM provides high flexibility in deploying virtual machines with different configurations in a regular site. It uses standard services and tools such as DHCP, PXE, SystemImager and Ganglia. We evaluated the deployment functionality of XSM and observed that it can deploy a base Xen system efficiently in a site with a variable number of machines.

As future work, we intend to enhance the management functionalities of XSM, develop a website to facilitate the management of Xen Virtual Machines, and perform deployment tests with larger environments in order to identify spots for optimization.

References

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauery, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: Proceedings of SOSP'03. (2003)
2. VMware, I.: Vmware - virtualization software. http://www.vmware.com/ (2006)

3. SWsoft: Welcome to openvz - server virtualization open source project. http://openvz.org/ (2006)
4. Virtuozzo: Virtualization, virtual server, windows, linux, consolidation, physical to virtual, disaster recovery software. http://www.virtuozzo.com/ (2006)
5. Corporation, I.: Intel virtualization technology. http://www.intel.com/technology/computing/vptech/ (2006)
6. Advanced Micro Devices, I.: Amd's virtualization solutions. http://enterprise.amd.com/us-en/Solutions/Consolidation/virtualization.aspx (2006)
7. Creasy, R.J.: The origin of the vm/370 time-sharing system. IBM Journal of Research and Development **25**(5) (1981) 483–490
8. Vallee, G., Scott, S.L.: Oscar testing with xen. In: HPCS '06: Proceedings of the 20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment, Washington, DC, USA, IEEE Computer Society (2006) 43
9. XenMan: The xenman project. http://sourceforge.net/projects/xenman/ (2006)
10. Labs, E.: Enomalism virtualized management console. http://www.enomalism.com/ (2006)
11. Software, B.G.: Systemimager. http://systemimager.org (2006)
12. Massie, M.L., Chun, B.N., Culler, D.E.: The ganglia distributed monitoring system: Design, implementation, and experience. Parallel Computing **30**(7) (2004) 817–840
13. Mugler, J., Naughton, T., Scott, S.L., Barrett, B., Lumsdaine, A., Squyres, J.M., Beno t des Ligneris, Francis Giraldeau, Leangsuksun, C.: OSCAR Clusters. In: Proceedings of the Ottawa Linux Symposium (OLS'03), Ottawa, Canada (2003)