INTEGRATION OF COMPUTERS AND TERMINALS BY INTRODUCTION OF THE VIRTUALIZATION TECHNOLOGY AND THIN CLIENT IN SPRING-8

Masahiko Kodera#, Taichi Shimizu, Ko Mayama, Masahiko Hanada, Shigeru Yokota
JASRI/SPring-8, Hyogo, Japan

Abstract

We applied virtualization technology to server computers to form a high-availability redundant server system. At the same time, we replaced general-purpose PCs with thin-client terminals. The introduction of these technologies reduced the number of computers substantially and gave us an opportunity to develop high-availability computing systems with inexpensive management. To ensure high availability, the server computer has to be built using reliable components with redundant architecture instead of reducing the number of computers. The application-processing performance of the client OS on the host OS was greater than or equal to that a standalone server. The combination of the recent multicore architecture server and Xen hypervisor showed good performance as a result of appropriately allocating system resources to Xen Hypervisor. The thin-client system is useful for integrating widely scattered terminals into a small number of systems, which will reduce maintenance effort. The integrated virtual machine system and thin-client system use a network-attached storage system that runs under the redundant configuration.

SELECTION OF VIRTUALIZATION SOFTWARE

The computers that we planned to integrate using a virtualization technique consisted of many http and ftp server computers and a few Windows Server OS computers. We adopted the hypervisor “Xen” that acts on Linux OS in virtualization software to it. The main reason for adopting Xen is because the transaction speed of “paravirtualization” of Xen is well suited to operate many Linux servers as a virtual guest. Very few Windows servers were included in the computers we wanted to unify.

Xen can operate various OSs by a “full-virtualization” mode by using a virtualization support function of the CPU. When testing virtualization on a Windows server in Xen 3.0.4, full-virtualization was stabilized and carried out. However, a full-virtualization guest needs attention not to boot lots of numbers on the server computer since a utilization rate of a CPU is always high.

We began virtualizing an SPring-8 informational system in May, 2008 by taking advantage of Xen 3.2.0 of SUSE Linux Enterprise Server 10 SP2 and this bundled edition. We are currently conducting transference work to SUSE Linux Enterprise Server 11 and Xen 3.3.1.

Novell expresses “one subscription to SUSE Linux Enterprise Server covers all virtual images on the same physical server” as a unique subscription policy.[1]

PERFORMANCE REQUIREMENTS OF VIRTUALIZATION SERVER

Performance requirements of the virtualization server are as follows.

Capacity of Main Memory

The virtual guest cannot start up after having used up the physical memory. Therefore, the memory capacity of the physical machine is decided on the basis of the total of memory that a server OS and all virtual guests require.

CPU Performance

In a physical machine for virtualization, the large number of core CPUs is an advantage. If this core CPU number on a physical machine is large, the tasks of many virtual guests can be performed at the same time. If the CPU number is small, the performance speed can reduce, unlike the memory capacity, which is determined by a virtual guest.

When our information system computer is integrated, the time for which the CPU is busy must be as short as possible; then, it would not matter even if we would boot virtual guests equal in number to more than twice the number of CPU cores.

Attention is necessary in that the guest using a Full-virtualization mode occupies a CPU for emulation for a street that was written in an item in front however and Windows OS.

A method for achieving a high CPU processing rate and carrying out non-multi-thread application with virtual machines has been proposed. In this method, the ability of the multicore CPU of the mainstream is utilized effectively for the non-multi-thread application. The number of CPU cores should be more than the number of virtual guests in order to avoid competition between the various processes.

HIGH-AVAILABILITY VIRTUALIZATION INFRASTRUCTURE

High-availability computers for virtualization integration are shown in Figure 1.

#kodera@spring8.or.jp

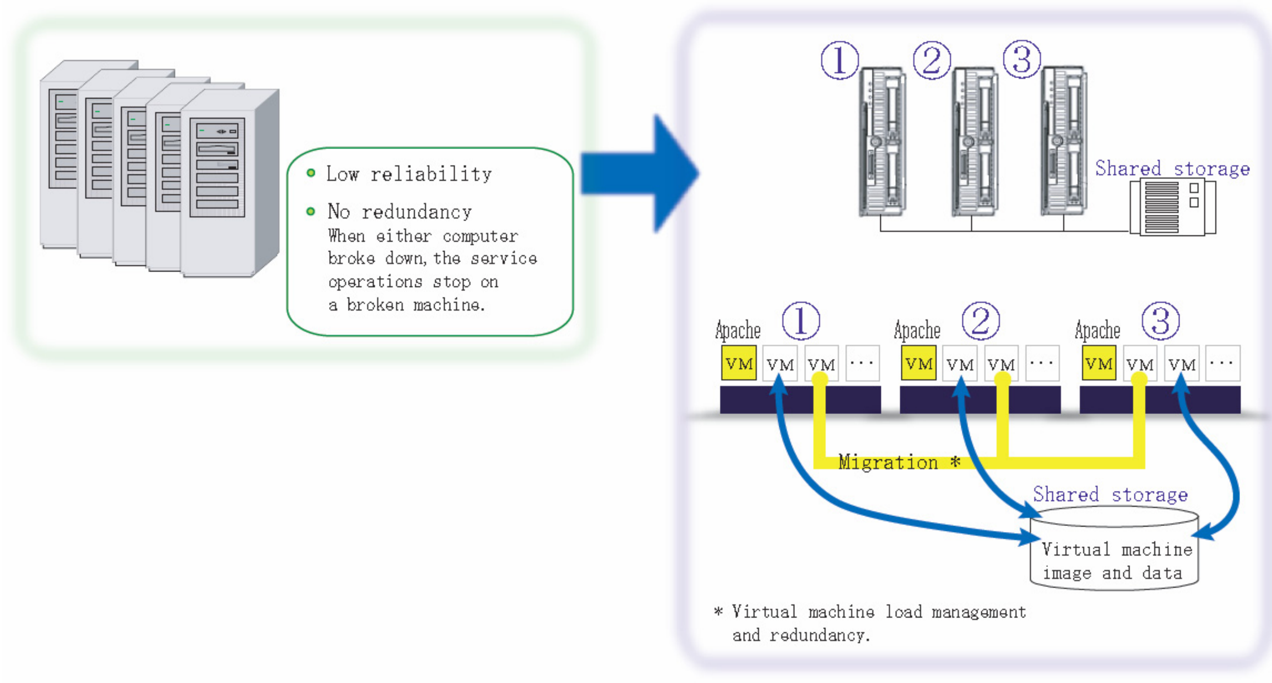

Figure 1: High-availability computers for virtualization integration.

A virtualization infrastructure created using many server computers (physical machines) and a network-attached storage (NAS) system. The physical machine takes charge of several virtual guests.

When a physical machine breaks down, we can migrate the virtual guests to another physical machine and can resume functioning immediately.

Further, it is better to have more than three physical machines for virtualization, because two physical machines are required for ensuring optimal performance. When building virtualization infrastructure having three physical machines with two-socket quad-core CPU and 16 GB memory, we can integrate approximately 30 low-end servers having single-core CPUs having less than 1 GB memory. We can make all virtual guests work with the remaining two machines even when one machine has broken down.

If the virtualization infrastructure has only two physical machines, we must use a four-socket CPU with 32 GB memory, which is very expensive. Therefore, for reasons of economic efficiency and high availability, it is preferable to use a two-socket (total eight cores) CPU multitudinously.

We are ensuring cost-effectiveness that provides sufficient reliability improvement of the virtualization server, assuming that the number of servers decreases by 1 in 10. For performing a hot swap, a disk, a fan, and a power supply for our all servers become redundant.

SPECIFICATION OF VIRTUALIZATION INFRASTRUCTURE FOR INFORMATION SYSTEM

The abovementioned specifications, based on the physical servers that are to be used with the virtualization infrastructure for the information system, are shown in Table 1.

Table 1: Specifications of Physical Servers

Components	**Specifications**	**Memo**
CPU	L5335 2.00 GHz × 2 E5410 2.33 GHz × 2 L5420 2.50 GHz × 2	TDP: L5335,5420 = 50w E5410 = 80w
Memory Capacity	16 GB	
Chases	Blade or Rack Mount 2U	
Redundancy	Power Supply, Cooling Fan, System Disk	Hot Swap

When placing an order for the CPUs, it is important to ensure that several are ordered.

Most of the old information system servers that we integrated by virtualization were low-end servers with a single-core CPU having a memory capacity of under 1 GB. Three physical machines, shown in Table 1, can integrate approximately 30 old low-end servers.

The specifications of the NAS system used in this study are listed in Table 2.

Table 2: Specifications of NAS System

Components	Specifications	Memo
NAS Head	Two-Node Cluster	High Availability
RAID	RAID-DP (RAID4 + Dual Parity)	NetApp Filer
Disks	SAS for VM Images SATA for Data	

This storage system is comprises a high-availability cluster with two NAS controllers for the SAS disks and SATA disks.

Next, the security and operation of this storage system are described. The image files of the virtual machines are stored in the high-speed SAS disks. The data files of the virtual machines are stored in the large-capacity SATA disk.

We permit that only Domain 0 of a virtualization server does a mount as for a virtual machine image file area of this storage. Further, a virtual guest can mount only its assigned data file area. Therefore, a virtual guest cannot access the territories of other virtual guests.

In order that a specific virtual guest does not monopolize the capacity of the common storage, we set disk quotas in the territories of the guests.

TERMINAL INTEGRATION BY THIN CLIENT

We integrated the terminal computers by utilizing a thin-client system. We adopted Sun Ray in a thin-client system.

Many Windows PCs were available at SPring-8 for visitors. The application software used with these Windows PCs was Web browser and Office Viewer.

A Solaris-based Sun Ray thin-client system is suitable for developing a kiosk terminal that restricts applications in browser and Open Office. This thin-client system is such a simple structure that the terminal side does not have the main memory, CPU, and a fan as the OS completely works on the server. Therefore, maintenance of the terminals scattered far and wide can possibly be avoided.

The Sun Ray thin-Client system is used as an X11 base terminal to maintain the SPring-8 accelerator and a development terminal of the X11 base "MADOCA" application.

We are now attempting virtualization of the Sun Ray server numbering order to reduce the number of computers. Because the memory capacity of a Sun Ray system for X11 applications is large, we decided to use a separate SUSE Enterprise based Xen system for other information systems.

The virtualization technique "container" in the OS layer is a standard in Solaris OS, but the Sun Ray server does not support this technique. Therefore, we adopted the Citrix XenServer.

XenServer is a product code whose stability was tested by Citrix. This product became freeware other than a higher product including a high management capability recently.[2] When it was not required to include application software in Enterprise Linux, XenServer that hypervisor was made free became a good choice.

CONCLUSION

We reported the reduction of maintenance cost and labor saving as a result of the introduction of virtualization technology and thin client into SPring-8. When a virtual guest uses an application of Linux, such as Apache, Python, and Plone, Xen can still carry out on the Enterprise Linux OS.

In the virtualization of the Sun Ray server, we introduced Citrix XenServer but could not yet determine the advantage of the open-source version, particularly of the Solaris OS guests. We examined the moving of the Windows imagination guest to the Citrix XenServer from Xen of SUSE as the next step. The para-virtual driver for Windows offered to Xen as the structure that does not overload the CPU by the I/O processing of the Windows guest. We tested this driver and found it to be unstable. Further, we shelved the use of this driver as it was sold as a very expensive add-on to the SUSE Linux Enterprise package. Because it is considered that the development of Citrix goes ahead of this driver, we performed tests in the Citrix version. In the future, Xen with SUSE Linux Enterprise will support only Linux guests, and it is thought that the others can be supported in Citrix XenServer.

KVM merged into the Linux kernel is expected to be used in the future to monitor the performance of Xen and the possibility of virtualizing the Linux host OS.

REFERENCES

[1] Virtualization simplified with SUSE Linux Enterprise Server, Novell, Inc.; http://www.novell.com/products/server/virtualization.html.

[2] New Citrix XenServer Release Makes Enterprise-Class, Cloud-Proven Virtualization Free for Everyone; http://www.citrix.com/English/ne/news/news.asp?newsID=1687130.