Virtual Platform Technologies

● Hiro Kishimoto ● Keisuke Fukui ● Riichiro Take

(Manuscript received April 27, 2009)

Cloud computing—providing information technology (IT) capability as a service over the network—has been attracting attention as a result of advances in server performance, storage capacity, and virtualization technology and the deep penetration of broadband networks. It lets users access services from anywhere and share data and even services with other users as the need arises. In Cloud computing, the user has no need to own an IT infrastructure or purchase equipment or bother with system configuration or management. Furthermore, infrastructure service providers can centralize resources, which enables them to lower prices and provide only the services that are needed when they are needed.
In this paper, we outline a Cloud platform for providing an application system infrastructure in a flexible and rapid manner according to the needs of system developers and service providers. We also describe the advanced technologies that make this platform a reality.

1. Introduction

Cloud computing—providing information technology (IT) capability as a service over the network—has been attracting attention as a result of advances in server performance, storage capacity, and virtualization technology and the deep penetration of broadband networks.[1)] It lets users access services from anywhere and share data and even services with other users as the need arises. In Cloud computing, the user has no need to own an IT infrastructure or purchase equipment or bother with system configuration or management. Furthermore, infrastructure service providers can centralize resources, which enables them to lower prices and provide only the services that are needed when they are needed.

In this paper, we outline a Cloud platform for providing an application system infrastructure in a flexible and rapid manner according to the needs of system developers and service providers. We also describe the advanced technologies that make this platform a reality.

2. Outline of architecture

The architecture of a Cloud platform providing dynamic and flexible support for many applications is shown in **Figure 1**. This platform, which corresponds to Fujitsu's Trusted Service Platform supporting Cloud services, is called a "virtual platform" in this paper.

2.1 Development environment and service groups

In the following discussion, service developers and service providers who develop, construct, and operate application systems are called "users", which we differentiate from end users who actually use the provided services. A user can develop and operate an application system relatively quickly by using the development environment provided by the virtual platform and combining services provided

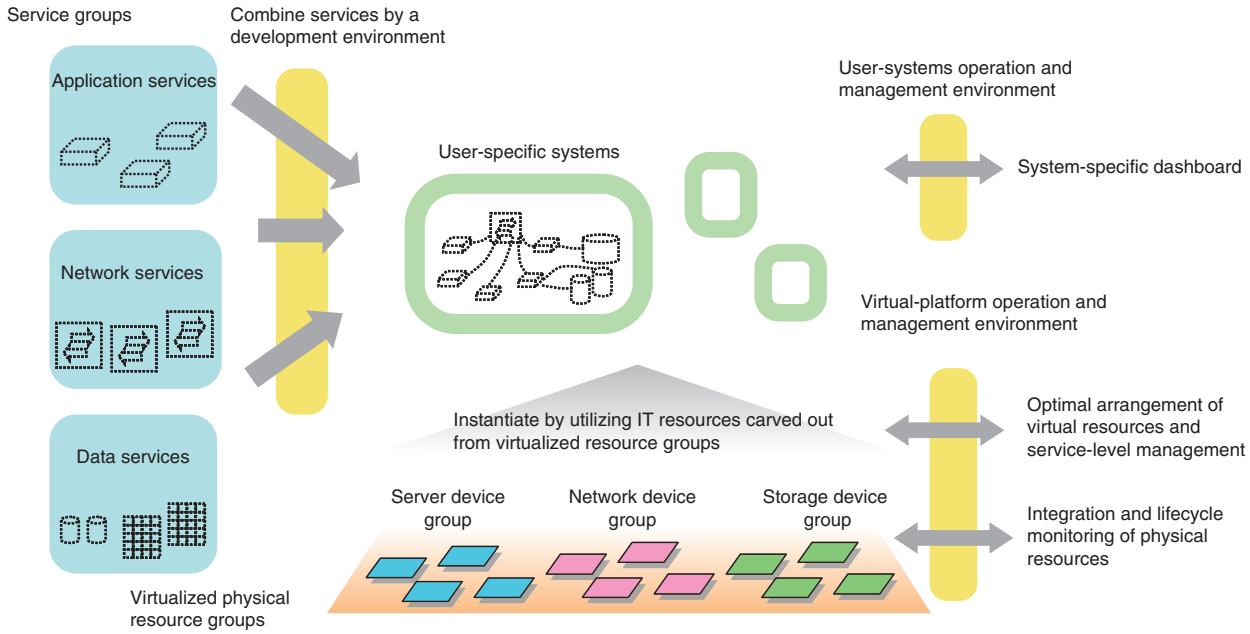

Figure 1
Virtual platform architecture.

by service groups. There are three service groups: application services, network services, and data services. They respectively provide programmable server systems, network functions such as firewalls and account management, and data storage and management functions such as databases. A user can quickly construct a desired application system by combining these service functions appropriately using the development environment provided by the virtual platform.

In the development of a Web-based three-tier system, for example, the user would use application services to create a Web server tier, application server tier, and database-server tier. The user would also use network services to obtain 1) functions for creating a virtual network to connect the servers making up the system, obtaining global Internet protocol (IP) addresses, or constructing a demilitarized zone consisting of firewalls and the like and 2) a mechanism for authenticating end-user accounts in the developed system. A more detailed description of these three types of service groups and the development environment is provided in section 3.

2.2 Virtual resource layer

Services provided by these service groups are implemented by allocating resources from a pool of virtualized resources; that is, the physical server, network, and storage resources are all virtualized. Here, all of the networks that interconnect these physical infrastructure resources are IP networks (Ethernet), including those for connecting storage devices. This approach reduces the number of types of physical network devices and the number of cables to be used, which results in a significant reduction in the network construction, configuration, and management costs. Mechanisms have also been developed to incorporate many physical resources into a virtual platform in units of racks, to configure hardware and basic input/output system (BIOS), and to install virtualization software.

The Xen virtualization software is installed on each physical server, and a group of guest virtual servers runs on each physical server. Although most virtualization software implementations handle only the server

equipment virtualization, the virtualization software used in our virtual platform also handles network and storage virtualization. In other words, separate virtualization functions are prepared for servers, networks, and storage. The point here is not that these functions are to be combined by the user, but rather that the virtual platform can uniformly manage all virtualization including server, network, and storage virtualization. The virtual resource layer is described in more detail in section 4.

2.3 Operation and management environment

The virtual platform provides two types of operation & management environments: the user-systems operation & management environment and the virtual-platform operation & management environment. The former provides users with functions for monitoring and controlling the user's application systems while the latter provides platform managers with functions for monitoring and controlling physical and virtual resource pools across the whole data center.

The monitoring and control of an application system is performed through the use of a dashboard function that is automatically provided when the target system starts running. Dashboard operations include system startup, termination, and backup. In system monitoring, the user can observe resource-usage conditions such as CPU load as well as the response time of user's application on virtual servers and services currently in use.

The monitoring and control of resource pools, on the other hand, is performed through the use of a cockpit display designed for the platform administrator. This display supports the management of a physical resource pool and the optimal arrangement of virtual resources on physical resources. Monitoring functions include the lifecycle management of physical resources, the measurement of resource-usage and response time statuses of virtual resources and the visualization of those statuses on the cockpit display, and the provision of information to the dashboard for each application system.[2] These operation and management environments are described in more detail in section 5.

3. Service groups and system development environment

In this section, we describe the service groups used by users in more detail, the system development environment for constructing application systems using those service groups, and the lifecycle of an application system developed in this way.

3.1 Service groups

Users can make use of three service groups: application, network, and data services. These are described below.

1) Application services

This group provides virtual server functions for implementing Web servers and application servers and network functions for interconnecting those servers. Users can create their own applications—using Java, Ruby, or other programming languages—for installation on virtual servers and they can construct application systems by interconnecting multiple virtual servers.

2) Network services

This group provides access to the user's corporate in-house network and the Internet and provides firewall, encryption, account management, and other functions as services. Although it is possible for users themselves to provide such functions on their virtual servers, using the services provided by this group and entrusting the development and management of such functions to proven Fujitsu technologies lets users concentrate their efforts on application development.

3) Data services

This group provides storage and database

functions as services. Here, as well, users themselves may provide such functions on their virtual servers, but using the provided services eliminates the need to design and develop backup and recovery functions, data migration procedures, etc. The functions provided by this group include virtual storage[3], file servers, relational databases, and key-value data services. A key-value database is a data service applicable to Cloud applications that handle sets of names (keys) and values. While not capable of strict atomicity, consistency, isolation, and durability (ACID) properties and complex queries characteristic of relational databases, a key-value database can be used to process the large volume of data required by Cloud applications in a scalable manner through the use of the eventual consistency model.

3.2 System development environment

The user develops an application system using software called "a virtual system" (VSYS). VSYS is intended to be a unit of software distribution when virtualization technology becomes popular. It can be used for sharing convenient and proven open source software and for selling commercial software. With VSYS, there is no need to install individual software. VSYS can be provided with pre-verified software stack compatibility and completed application tuning.

VSYS can include multiple virtual machines (VMs). The VSYS package consists of an extensible markup language (XML) document called the "VSYS description file" that describes the infrastructure elements, linkage among VMs, configuration parameters, license information, etc. required for deployment and a disk-image file that has the software stack already installed. Use of the Open Virtualization Format[4] developed by the Distributed Management Task Force (DMTF) is being studied to ensure interoperability among multiple Clouds. Multiple application systems can be deployed and run from a single VSYS package (**Figure 2**).

The system development environment provides two tools: System Builder and Service Builder. The first, System Builder, is used

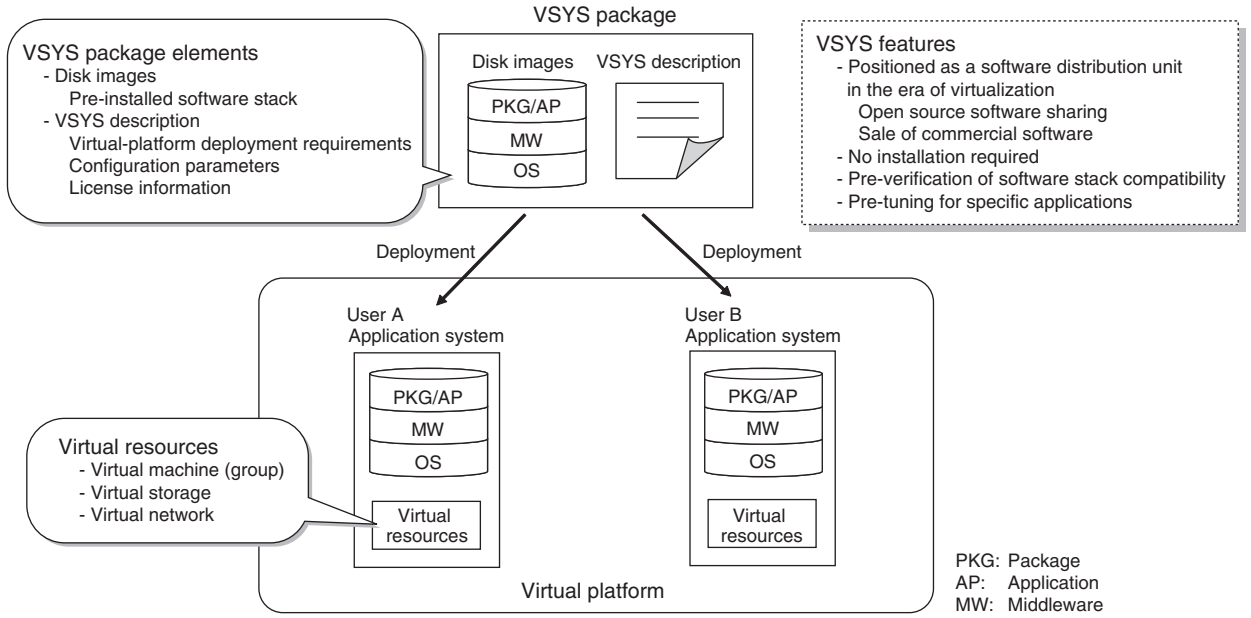

Figure 2
Virtual system (VSYS) and application systems.

to construct an application system out of application services. It allows the user to design virtual servers, the operating systems (OSs) running on those servers, a virtual network for interconnecting virtual servers, etc. The other tool, Service Builder, allows the user to interconnect the application system with the network and data services used by that system in a seamless manner.

When constructing a single system on the platform, the user can design the entire system using Service Builder. A new system can be constructed completely from scratch or can be based on existing off-the-shelf VSYS design templates for more efficient development. Constructing a system using Service Builder lets the user create an advanced system such as a Web-based three-tier system without having to get involved with the details of firewall settings, connections between application servers and databases, etc. In addition, VSYS templates also provide operational procedures such as for backing up data and responding to faults and security policies related to maintaining operating logs and encrypting data, which means that using existing VSYS templates for system construction can help reduce the time and labor involved in system operation and supervision.

3.3 Application-system lifecycle

An example of an application-system lifecycle is shown in **Figure 3**. In this example, the lifecycle begins when users extract existing system definitions (called "VSYS master") from the marketplace. It then continues when users perform a "check-in" of these definitions with respect to their individual repositories and begin development. Alternatively, a new application system can be constructed from scratch.

The user personalizes, deploys, and starts up the VSYS to obtain a personal application system. Functions and system performance may also be tested by using a load-generating service. In this step, the application system is shielded by a firewall and cannot be accessed or used from the Internet. After testing, modifications

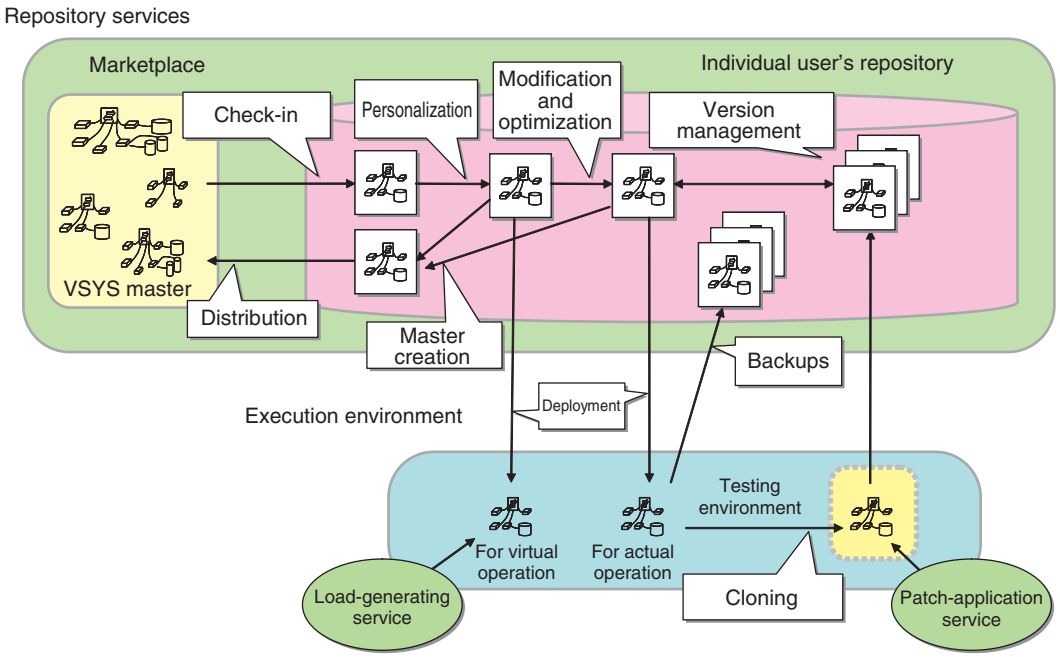

Figure 3
Lifecycle of VSYS.

can be performed if needed and actual operation can commence after it has been connected to the Internet.

A running application system may use a backup and snapshot service as needed. The images of multiple systems may also be managed within a repository. The operation flow for backups and snapshots and related automatic execution functions are included in the VSYS master. The user may customize this flow if desired.

VSYS also provides patch processing for an application system as a service. In this process, the service first takes a snapshot of the target application system. It then deploys this snapshot in a closed environment for testing purposes and applies any patches that are deemed necessary. At this point, the user can test the patched system using a load-generating service if necessary, and after testing, the user can update the actual operating system.

Such user modifications and patches result in the existence of multiple versions of a system in the user's VSYS. These various versions are managed by the repository version-management function. A particular version may also be distributed back into the marketplace, and for this purpose, VSYS provides a function for creating a "master" version from a system image in which all IP addresses, business data, and other user-specific items have been blanked out.

4. Physical resource layer and its virtualization technology

4.1 Management and virtualization technology of server

There has been active research on developing ways to construct a large-scale distributed system in an efficient manner and deploying, operating, and managing multiple application systems.[5),6)] In the case of a virtual platform that is called upon to operate and manage from several hundred thousand to a million physical servers, the batch introduction of a large quantity of physical devices (servers, network switches, etc.) and their efficient operation and management are crucial issues. Efficient and simple operation/management technologies are needed not only to reduce the cost of labor and other operating costs but also to maintain a uniform level of maintenance quality across a large number of devices. Fujitsu has developed and provided infrastructure construction technology for significantly improving the efficiency of operating and managing such a huge number of physical devices.

The conventional approach to deploying devices in a data center required much manual labor for mounting devices in racks, laying cable and visually checking connections, configuring the server BIOS, checking and registering media access control addresses, etc. This placed a considerable burden on on-site customer engineers and resulted in poor efficiency. In response to this situation, Fujitsu has developed a function for automatically executing a series of on-site tasks when deploying new devices and a system for incorporating physical equipment into a virtual platform in units of racks. This minimizes human error and reduces costs when new equipment is added to a data center.

A virtual platform is also equipped with functions for enabling the simultaneous operation of a lot of application systems. These functions automatically deploy the virtual servers that make up an application system and make connections between the virtual servers over a virtual network. The design map of each user's application system is described by an XML document referred to as the VSYS description file. Thus, when the user indicates an application deployment or removal operation from the system dashboard, the required operation will be automatically executed from the virtual platform side in accordance with that VSYS description. This mechanism provides a form of self-service for an application system to user.

Various policies can be considered when arranging virtual servers on a virtual platform. In a user-developed system, virtual servers will be arranged dynamically on the most appropriate physical servers according to the policy specified by the platform manager. Two key arrangement policies are the energy-saving and high-availability policies. The energy-saving policy dictates that virtual servers shall be placed on as few physical servers as possible to free up other physical servers and that the power switches of free servers shall be turned off. This has the effect of reducing the amount of power consumed by the entire data center. The high-availability policy, on the other hand, dictates that virtual servers shall be placed on as many different physical servers as possible so that the number of virtual servers affected by the failure of a physical server can be minimized.

4.2 Network virtualization technology

In the process of accommodating multiple application systems on a single physical network within a data center, it is vital that the networks for these application systems be strictly separated. Layer-2 network separation technology based on virtual local area networks (VLANs) has been conventionally used to create virtual networks. The VLAN scheme has proved useful in simplifying deployment in the case of a small-scale network. In a large-scale network, however, the VLAN scheme is known to suffer from various problems such as a limited number of VLAN tags (max.: 1024), increase in the number of switches that must be set, easy occurrence of faults having far-reaching effects caused by trivial setting errors, and difficulty of testing overall network separation.

For its virtual platform, Fujitsu has chosen to use IP tunneling—a layered network connection technology—to make connections between virtual servers. The tunneling mechanism is implemented on the host VM side as opposed to the virtual-server (guest VM) side to prevent problems caused by application-system faults and to defend against attacks from malicious users (**Figure 4**). The virtual platform provides a mechanism for establishing IP tunnels when the user starts up virtual servers and deploys the application system. This has two benefits.

- The complex task of connection-path graph design in a VLAN and the setting of all

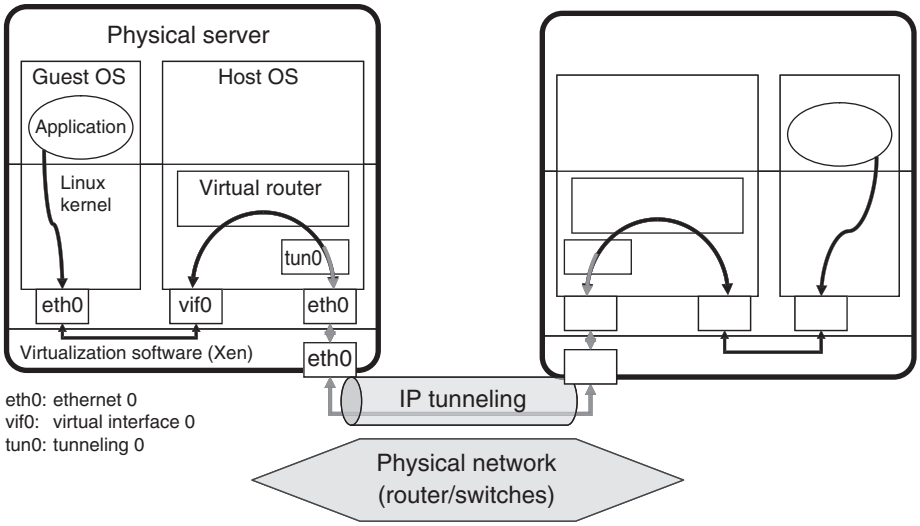

Figure 4
Network virtualization technology through IP tunneling.

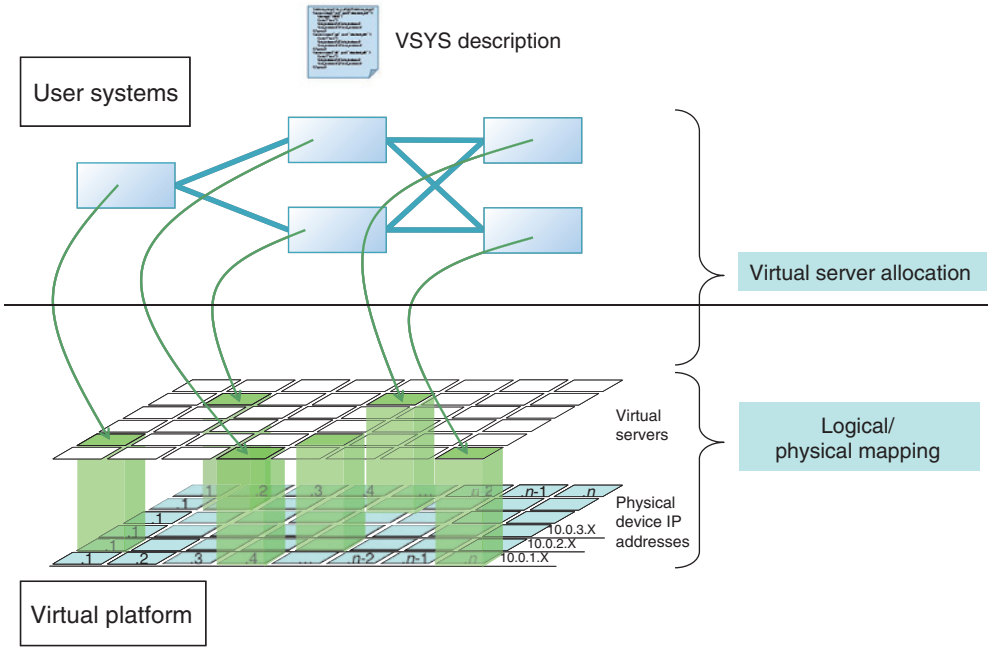

Figure 5
Automatic deployment of application systems.

network switches can be simplified by simply specifying the set of nodes for the IP-tunneling virtual network.

- Isolating the IP-tunneling mechanism from the user's virtual servers provides a mechanism for separating the network from errors in the application system and from malicious attacks.

In this way, the complexity of making physical network settings can be mitigated and the virtual resource pool can be safely used by multiple application systems. As a result, resources can be flexibly and promptly allocated or removed in accordance with the user's changing business environment. At the same time, the overall operation cost of the data center can be reduced.

The live migration of a guest server may be necessary to adjust the processing load of virtual servers, to perform preventive maintenance on physical servers, and to perform maintenance on the host VM's OS. At this time, IP tunneling must be dynamically redone in conjunction with this guest-server migration, but such virtual network configuration changes can be accomplished without having to change the physical network settings.

5. User system operation environment

Once a user has given instructions to deploy and start up an application system, the virtual platform will allocate from the resource pool only the requested number of virtual servers equipped with the necessary specifications. The virtual platform will use IP-tunneling to create a virtual network that includes only the virtual servers so allocated, and after connecting to virtual storage and deploying and initiating the user's guest VMs, it will boot the application system. This processing sequence is carried out automatically without the intervention of a data center operator (**Figure 5**).

If the operator detects signs of a physical fault in a fan, hard disk drive, or other component, he or she can specify a live migration for the guest VM so that the application system can continue normal operations. In this case, the virtual

platform will look for a free physical server, move the guest VM targeted for migration to that physical server, and automatically construct a network in accordance with the new deployment conditions.

The virtual platform provides the user with functions for measuring and visualizing basic system response times in addition to virtual-server lifecycles and CPU usage rates.[7)]

If the system load should increase and the response time deteriorate, the user can add virtual servers to increase the system's processing power. Conversely, if the processing load decreases, some of the virtual servers can be terminated and returned to the server pool to reduce the system scale.

The fee that the user pays for using a virtual server is determined by metered usage from server startup to termination. The virtual platform records the startup and termination of each virtual server, adds this usage to that of storage and network usage, and charges the user accordingly.

6. Conclusion

In this paper, we outlined technologies for achieving a large-scale virtual platform that can be shared by multiple application systems. This virtual platform enables the prompt construction, deployment, and secure separation and operation of VSYSs each of which consists not of a simple virtual server pool but of multiple servers, interconnecting networks, and storage facilities. The virtual platform provides application services, network services, and data services. The proactive use of these services, which are supported by Fujitsu's proven operation technologies, enables the user to construct a highly reliable system in a relatively short time. The fact that this virtual platform is shared may generate concerns about performance, but the dashboard function provided for each application system to monitor and analyze performance enables a change from the "best-effort" level of quality to "good enough" at a reasonable price. The benefits of using this virtual platform lie not only in minimizing physical resources through server integration. Development, test runs, actual operation, modifications, and testing can all be performed on this single platform, and sufficient resources can be quickly allocated even for applications for which resources for development, testing, and other purposes could not be easily prepared in the past. As such, we can expect this virtual platform to enable efficient lifecycle management of application systems.

Our plans for the future are to 1) raise resource usage efficiency, 2) expand the application scope of the virtual platform by enhancing the development environment and increasing the number and richness of services offered within service groups, and 3) further improve the behavior visibility, reliability, and security of application systems.

References

1) N. Carr: The Big Switch. W. W. Norton & Co. (January 17, 2008), ISBN-10: 0393062287.
2) R. Take: Virtualized IT Platform for Cloud Computing Era. (in Japanese), *FUJITSU*, Vol. 60, No. 3, pp. 266–273 (2009).
3) R. Take: Organic Storage System. (in Japanese), *FUJITSU*, Vol. 55, No. 4, pp. 364–367 (2004).
4) Open Virtualization Format Specification, v1.0.0e, 15 Jan. 2009, DMTF, DSP0243.
5) A. Grimshaw et al.: An Open Grid Services Architecture Primer. IEEE Computer, Feb 2009, pp. 27–34.
6) A. Savva, T. Suzuki, H. Kishimoto: Business Grid Computing Project Activities. *Fujitsu Sci. Tech. J.,* Vol. 40, No. 2, pp. 252–260 (2004). *http://www.fujitsu.com/downloads/MAG/vol40-2/paper09.pdf*
7) R. Take: Real-time System Behavior Visualization Technology for Managing Complex Distributed Application Systems. (in Japanese), *FUJITSU*, Vol. 59, No. 1, pp. 33–38 (2008).

Hiro Kishimoto
Fujitsu Laboratories Ltd. and Fujitsu Ltd.
Dr. Kishimoto received a Ph.D. degree in Computer Science from Tohoku University, Sendai, Japan. He joined Fujitsu Ltd. in 1983. He has led several software projects on distributed systems development and data center optimization. He is also a visiting professor at the National Institute of Informatics, Vice President of the Grid Consortium in Japan, and a board member of DMTF and OGF. He received the IEEE Gordon Bell Award in 1994, the InfiniBand Contribution Award in 2000, and the inaugural GGF Leadership Award in 2005.

Keisuke Fukui
Fujitsu Ltd. and Fujitsu Laboratories Ltd.
Mr. Fukui received a B.S. degree in Applied Physics from Waseda University, Japan in 1987. He joined Fujitsu Ltd. in 1987 and is currently a Senior Researcher at Fujitsu Laboratories Ltd. in Japan. He has worked on many software development projects in Fujitsu Ltd. as a leader and developer, including the OPEN LOOK toolkit I18N and VIPL user-level low-latency networking. He is now leading the ITcell project. He also has experience in developing international standards, having co-chaired the ACS Working Group in the Open Grid Forum.

Riichiro Take
Fujitsu Ltd. and Fujitsu Laboratories Ltd.
Mr. Take joined Fujitsu Ltd. in 1982 and has been working mainly in the software algorithms and system architecture domain. He has been interested in several research areas such as database processing algorithms, interconnection architectures for parallel database processing, data mining algorithms, and complex event-data processing systems. He is currently leading a research activity called “Organic Computing” and working to give IT systems the intelligence to perceive their own condition and react to it appropriately.