

PRIMEQUEST Virtual Machine Function

White Paper
August 2007
Fujitsu Limited

This document includes functions scheduled for future presentation.
The contents described in this document may be revised without prior notice.

Intel and Itanium are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Microsoft, Windows, and Windows Server are registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Linux is a registered trademark or trademark of Linus Torvalds in the United States and other countries.

Xen is a trademark of XenSource, Inc.

TRIOLE, PRIMEQUEST, ETERNUS, and Systemwalker are registered trademarks or trademarks of Fujitsu Limited.

Other company names and product names mentioned in this document are trademarks or registered trademarks of their respective companies.

Contents

1. Introduction	3
2. Server Virtualization.....	3
3. PRIMEQUEST Virtual Machine Function.....	5
3.1 Features of the PRIMEQUEST Virtual Machine Function.....	5
3.2 Realization of Integrated Operation Management	6
3.3 Building a High-Availability System.....	7
3.4 Specifications	8
4. Merits of Virtual Machine Use	8
4.1 Concurrent Operation of Multiple Business Application Systems	8
4.2 Development Environment Creation and Conversion to a New Operating System	9
4.3 Fast Acquisition of a New Server	11
4.4 Load Based Dynamic Resource Allocation	12
4.5 Integration of Standby Systems	14
4.6 Service Continuity during Machine Downtime	15
5. Virtual Machine Function Implementation Technology.....	16
5.1 Overall Structure	16
5.2 The Virtual Machine Function as Partitioning Technology	17
5.3 Full Virtualization and Paravirtualization	18
5.4 CPU Virtualization.....	19
5.5 Memory Virtualization.....	20
5.6 I/O Virtualization.....	21
5.6.1 Device Emulation Method	22
5.6.2 Virtual Device Method.....	23
5.6.3 Direct I/O Method (Planned).....	23
5.7 Disk Virtualization.....	24
5.8 Network Virtualization	25
6. Virtual Machine Duplication and Migration.....	26
6.1 Cloning (Virtual Machine Duplication).....	26
6.2 Static Migration (Static Moving of a Virtual Machine, Planned)	27
(1) Physical machine to virtual machine (P2V).....	27
(2) Virtual machine to physical machine (V2P).....	28
(3) Virtual machine to virtual machine	28
6.3 Dynamic Migration (Dynamic Moving of a Virtual Machine, Planned).....	28
7. Conclusion	28

1. Introduction

Key to success in business process reengineering is an ability to stay ahead of the competition by implementing change as simply and easily as possible. Business infrastructure therefore needs to have "agility" to quickly deal with changes in the business environment, "efficiency" to reduce unnecessary costs and "continuity" to provide essential business services 24 hours a day, 365 days a year. To support this Fujitsu has developed and advocates an information technology (IT) infrastructure framework called TRIOLE. TRIOLE is an ongoing development of leading-edge functions centered on three core technologies: virtualization, automation, and integration. This white paper describes how the virtual machine function of Fujitsu's mission-critical IA server PRIMEQUEST meets those aims.

2. Server Virtualization

Virtualization is an often-used concept in IT systems. Virtualization technology adds new degrees of freedom by severing fixed dependencies between separate layers of the computing model. For example, storage virtualization severs the previously fixed links between server and storage devices. It allows the user to change freely the relationships between server applications and the locations where data is stored; without having to change physical disks or cable connections. Similarly, network virtualization severs the fixed relationships between computing devices and the wiring to allow users to create new "virtual" networks without any change to the physical network.

The same idea is the basis of server virtualization, which is the subject of this document. Server virtualization breaks the rigid one-to-one link between the operating system(OS) layer and the physical machine layer (Figure 1). As a result, users gain freedom by the ability to operate multiple "virtual" machines on the same physical machine and move virtual machines across different physical machines.

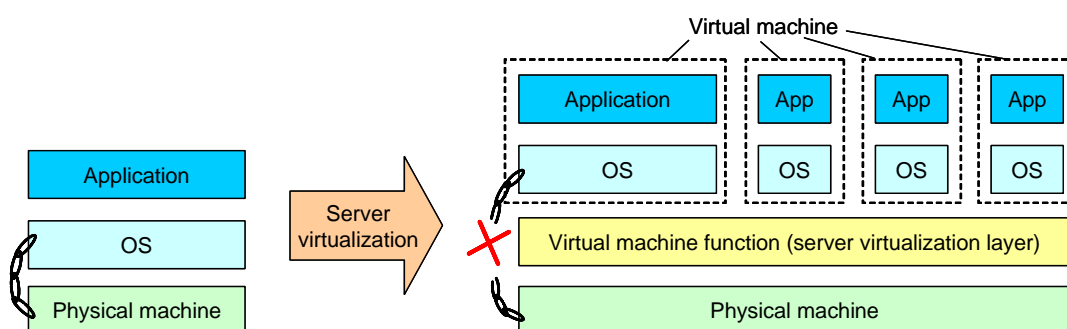


Figure 1. Server virtualization

Server virtualization is an attractive solution to current issues in business IT. Accelerated business change and increases in the cost of server management have seen a desire to consolidate scattered IT resources within companies. Technically, the elements necessary to allow server virtualization of open servers have also come together. This includes

- a) improved server performance resulting from multi-core CPU configurations;
- b) the introduction, by Intel and other chip manufacturers, of virtualization support hardware in CPUs;
- c) Plus Xen, emerging open source virtualization software, that works with such CPUs.

While open server virtualization has recently gained attention, the concept of the virtual machine has existed for quite some time. Fujitsu first offered its AVM virtual machine function on mainframes in 1980. Over the years, Fujitsu has continued to accumulate and develop virtualization technologies. An established technology, the resulting current AVM/EX, virtual machine environment for mainframes, is used by over 80 percent of Fujitsu's large-scale mainframe customers.

Fujitsu also participates in the Xen open source community and is a major contributor to its development (Figure 2). Fujitsu's plan is to continue applying its extensive experience in virtual machine technology in support of future Xen development as well.

April 2006 to March 2007

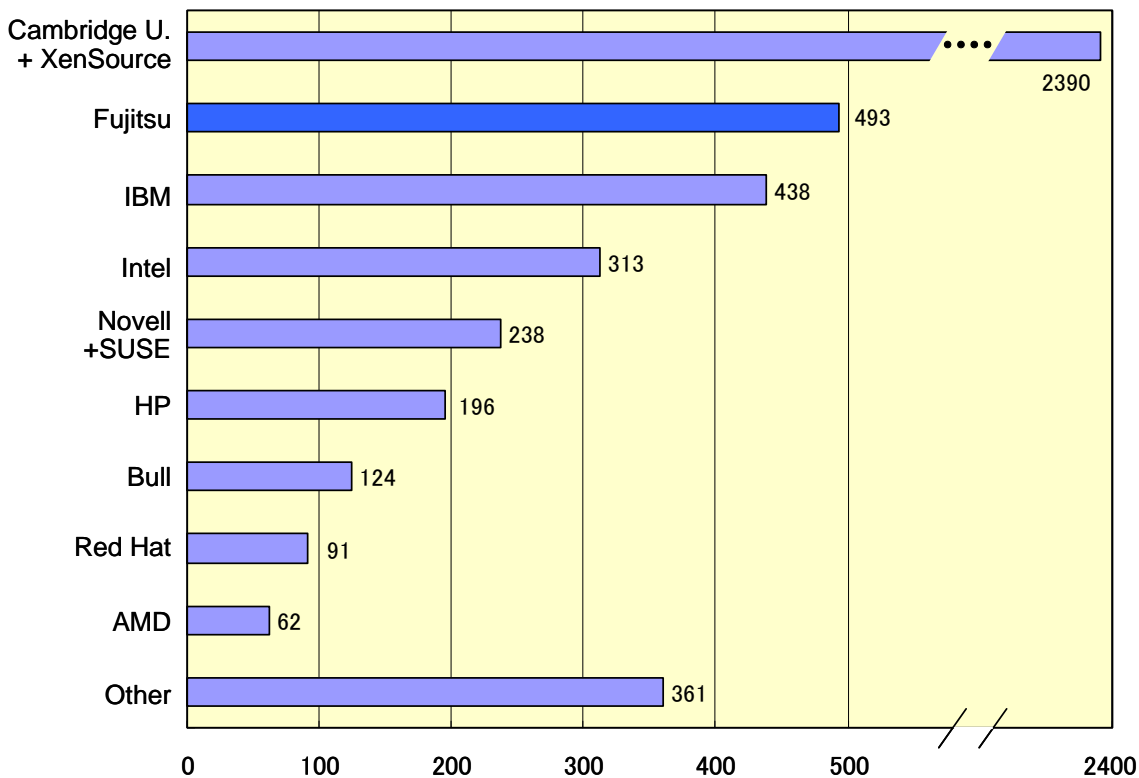


Figure 2. Contributions to the Xen community (Total number of expansions and revisions posted by Fujitsu and others)

3. PRIMEQUEST Virtual Machine Function

The PRIMEQUEST virtual machine function enables similar high levels of server virtualization in Fujitsu's mission-critical IA server environment. It achieves flexible and efficient use of the high-performance and high-reliability resources of each PRIMEQUEST system. The functions are enabled by virtualization software included in Red Hat Enterprise Linux 5 and Fujitsu's support for that software. This includes Fujitsu proprietary value-added software for performance enhancement, improved reliability, and installation support. The PRIMEQUEST virtual machine function is available with all servers starting from PRIMEQUEST 500 series.

3.1 Features of the PRIMEQUEST Virtual Machine Function

The PRIMEQUEST virtual machine function has the following features:

- Concurrent operation of multiple operating systems (Linux and Windows)

Up to 60 virtual machines can be started, and independent operating systems

known as “guest OS” can operate on each virtual machine.

- Flexible and detailed allocation of system resources to each operating system

The physical partitioning capabilities of PRIMEQUEST include Physical PARTitioning (PPAR) and eXtended PARTitioning (XPAR). Both provided for the splitting of physical hardware at hardware component boundaries. However, with the PRIMEQUEST virtual machine function, allocation of system resources including CPU, memory, and I/O devices is possible in much smaller increments. This greater granularity makes it much easier to change resource allocations more accurately.

- Sharing of FC and LAN cards between multiple operating systems

Fibre Channel (FC) cards, LAN cards and cables can be shared between all operating systems enabling the building of more efficient systems without waste of resources.

3.2 Realization of Integrated Operation Management

Unified operation management of both physical and virtual machine environments is another advantage. This is achieved by combining the PRIMEQUEST virtual machine function with Fujitsu's Systemwalker integrated operation management software product set.

- Automatic operation

- By combining the virtual machine function with Systemwalker Operation Manager, users can automate startup, termination, and resource allocation changes to virtual machines and guest OS based on a defined schedule.

- Monitoring

- By combining with Systemwalker Centric Manager users can “visualize” operation of the virtual machines and guest OS on the physical machines. This enables highly accurate identification of failures, their location and the extent of their effect.
- By combining with Systemwalker Service Quality Coordinator, users can monitor the performance of virtual machines, guest OS, and middleware. This enables the visual tracking of resource usage status allocated to each virtual machine.

- High-speed backup

- By use of a Fujitsu ETERNUS disk array unit and the high-speed backup software ETERNUS SF AdvancedCopy Manager, users can easily perform high-speed disk-to-disk backup and integrated disk-to-disk-to-tape backup with

minimum load on the virtual and physical servers.

3.3 Building a High-Availability System

Fujitsu provides a range of redundancy functions to increase service availability, with systems using the PRIMEQUEST virtual machine function.

- PRIMEQUEST hardware redundancy
 - The System Mirror mechanism enables duplex of important hardware devices, such as memory and chipsets. This enables users to achieve mainframe-class levels of reliability and system availability.
 - The use of a floating (spare) system board likewise enables the user to recover quickly from a system board fault, simply by restarting the virtual machine function.
- Disk and network redundancy
 - Fujitsu's PRIMECLUSTER GDS enables the mirroring of disk units.
 - Fujitsu's ETERNUS Multipath driver enables the building of redundant FC connection paths between each operating system and the ETERNUS disk array units.
 - Fujitsu's PRIMECLUSTER GLS enables the building of redundant transmission paths between each operating system and key network destinations.

The above redundancy functions enable users to set up redundancy for each host OS and guest OS (Linux). By making shared services redundant on each host OS, redundancy of each guest OS is also ensured and operation is simplified.

- Server redundancy
 - Fujitsu's PRIMECLUSTER cluster software lets multiple servers operate as one system. Clusters can consist of a mix of servers (guest Linux) on virtual machines or servers (native Linux) on physical machines. Even if one physical or virtual server fails, business applications can continue on the remaining servers.

PRIMEQUEST Virtual Machine Function

3.4 Specifications

Supported hardware		PRIMEQUEST 500 series or later	
Maximum number of virtual machines (guest OSes)		60	
Maximum number of supported physical CPUs		4 CPUs (8 cores) Expansion to 32 CPUs (64 cores) scheduled for 2nd half of 2007	
Maximum supported memory size		Maximum mountable memory size on 1 system board PRIMEQUEST 520: 128 GB PRIMEQUEST 540/580: 256 GB Expansion to maximum mounted memory for each PRIMEQUEST model scheduled for 2nd half of 2007	
Memory size of host OS		1024 MB fixed (recommended)	
Memory size of hypervisor		64 MB	
Supported virtualization method		Full virtualization method	
Supported guest OSes		<ul style="list-style-type: none"> •Red Hat Enterprise Linux 5 (for Intel Itanium) or later •Red Hat Enterprise Linux AS (v.4 for Itanium) Update 4 or later •Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems SP1 or later •Microsoft Windows Server 2003, Datacenter Edition for Itanium-based Systems SP1 or later 	
Virtual network	Virtual network format	Virtual bridge connection	
	Maximum number of virtual bridges	60 (However, number of mounted physical NICs cannot be exceeded.)	
	Maximum number of VNIFs (*1) that can connect to virtual bridge	64	
Disk	Supported block devices	Disks, partitions, GDS logical volumes File format (*2) support scheduled for second half of 2007	
		Peripheral devices supported by RHEL5	
Connectable peripheral devices		Peripheral devices supported by RHEL5	
Availability	Cluster configuration		
	Realization of cluster configuration in guest OS by PRIMECLUSTER scheduled for 2nd half of 2007 (Linux guests only)		
	Redundancy (*3)	Disk path	Enabled by ETERNUS Multipath driver
		Disk unit	Enabled by PRIMECLUSTER GDS
LAN		Enabled by PRIMECLUSTER GLS	
Main restriction		Hyperthreading technology is not supported	

(*1) VNIF: Virtual network interface

(*2) File format: Format that stores virtual machine environment and data as files

(*3) Redundancy in host OS. Redundancy in Linux guest OS scheduled for 2nd half of 2007.

4. Merits of Virtual Machine Use

The PRIMEQUEST virtual machine function provides a number of TRIOLE declared benefits:

Efficiency: Reduce costs (reduce hardware, software, and operation management costs)

Agility: Reduce time to change (changes can be dealt with immediately)

Continuity: Reduce down time (fast error correction & software recovery, highly secure system, able to handle overload situations)

These benefits materialize in a number of ways depending on business use, PRIMEQUEST configuration, and use of the virtual machine function. This section provides specific usage examples and shows the types of benefit that can be gained in each scenario.

4.1 Concurrent Operation of Multiple Business Application Systems

Efficiency	Agility	Continuity
------------	---------	------------

Excellent	Good	--
-----------	------	----

Dividing one physical server so that it operates multiple operating systems is the most obvious use of the PRIMEQUEST virtual machine function. This function means just one server needs to be managed instead of the multiple servers used previously. Server hardware costs, operation management costs, and power consumption are all reduced. Improved server hardware performance and the PRIMEQUEST virtual machine function means better use of all available resources, more exact allocation of resources per service and the ability to create a pool of resources for immediate use at times of high load. The PRIMEQUEST virtual machine function, therefore enables users to operate multiple Windows and Linux operating systems, simultaneously, on the same machine.

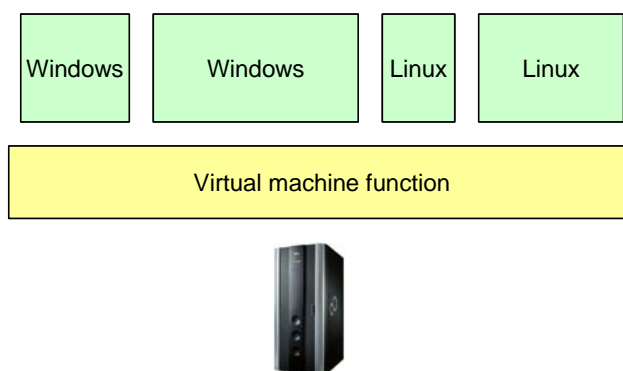


Figure 3. Run multiple operating systems in one physical server

4.2 Development Environment Creation and Conversion to a New Operating System

Efficiency	Agility	Continuity
Excellent	Good	Good

When a new application is developed, multiple development system environments are necessary for a variety of reasons during the testing and implementation period.

- Operating system settings may need to be changed for each development team.
- Development may be necessary on different operating system versions and at different patch levels.
- Multiple independent tests may need to be executed concurrently.
- Each development team may want to be able to restart their operating system

environment at any time.

Using the PRIMEQUEST virtual machine function, each user can prepare multiple development environment systems as necessary. The right resources can quickly be configured at low cost (Figure 4). Users can also have the business application system and the development environment reside on the same physical machine. Throughout they can adjust the amount of resources allocated to virtual machines and optimize resource usage efficiency based on progress of the development phase.

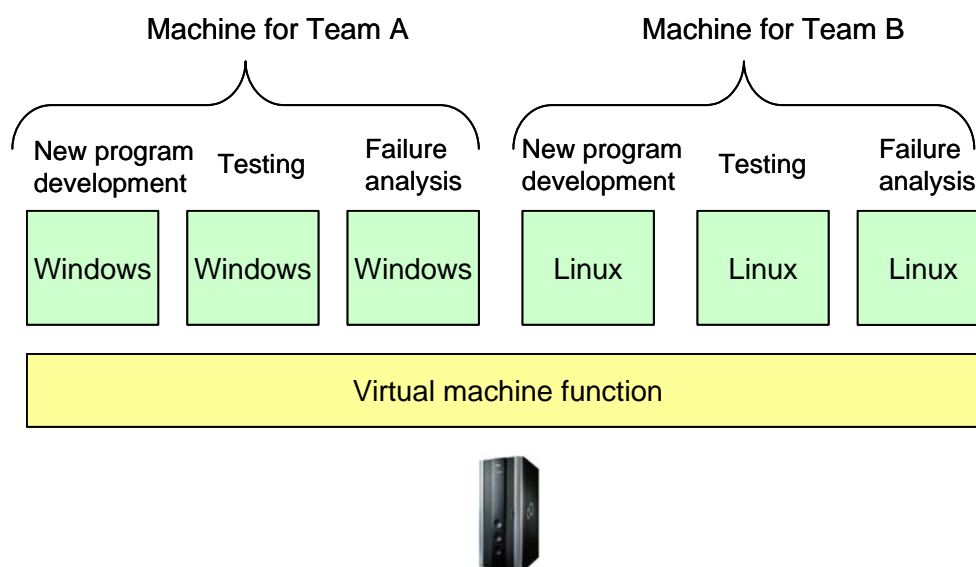


Figure 4. Development environment presentation

Even where there is a need to convert smoothly from an old operating system version to a new version, users can operate both the new and old systems on the same physical machine. In addition while each business application is verified for switch over to the new operating system, those not yet supported on the new version can continue to operate on the old version. As more business applications are migrated to the new operating environment, system resources can continue to be used efficiently by simply increasing the resources allocated to the virtual machines running applications on the new operating system.

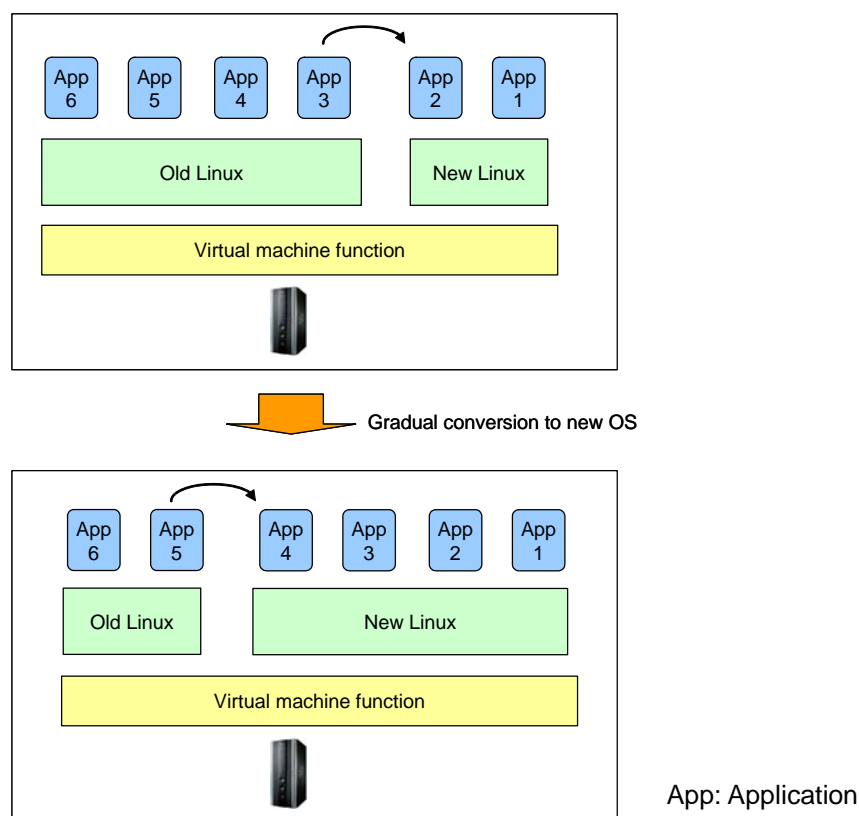


Figure 5. Conversion to a new operating system

4.3 Fast Acquisition of a New Server

Efficiency	Agility	Continuity
Excellent	Excellent	--

From time to time organizations have unexpected reasons for acquisition of a new server to handle shifts in business or organizational changes. Drawn out processes for securing hardware purchase budgets, the purchasing procedures themselves, plus hardware delivery and physical implementation times, can mean several months before the new service becomes available. As a result, the potential business opportunity can be lost or significantly reduced. However, use of the PRIMEQUEST virtual machine function allows the new server (virtual machine) to be quickly prepared with software. As no hardware purchase is involved, the organization can start up the server and receive benefits in a very short time. Although the actual time varies depending on conditions, the server can be prepared, in short order, from several minutes to under a few hours. Compared to times typically measured in months such preparation times are tiny fractions of the alternatives.

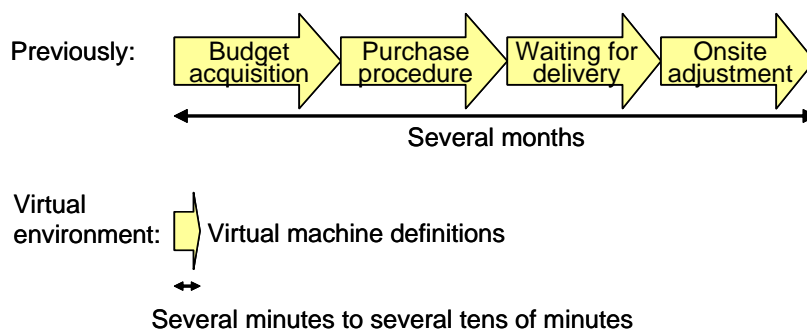


Figure 6. Physical and virtual server acquisition

The PRIMEQUEST virtual machine function also contributes to substantial shortening of the installation times for operating systems and applications. Since a virtual machine can be prepared easily, unlike a physical machine, the user can use a virtual machine to carry out the installation work well in advance. Saving this installation result as a virtual machine image enables very fast server operation start by restarting the virtual machine as necessary. If the virtual machine must run on a different physical machine from the one used to create it, a virtual machine-to-virtual machine conversion tool is used.

When multiple servers of the same type are required, it is simple for users to use an installed image as a template and clone more of the same. Using the cloning tool not only shortens installation time compared to manual setup of each server, but also reduces the number of installation errors. This is especially effective in the following situations:

- Using multiple servers of the same type for load distribution
- Using multiple environments of the same type for development and testing
- Distributing the software environment of a centralized information system to servers in each department
- Distributing software stacks for demonstrations, trials, and training

4.4 Load Based Dynamic Resource Allocation

Efficiency	Agility	Continuity
Excellent	Excellent	Good

Keeping server costs under control requires organizations to keep server system resource allocations to a minimum. In a physical machine environment, resources are

often difficult to change due to the high level of skill required for system resource reallocation, such as a CPU change. However, with the PRIMEQUEST virtual machine function, system resource allocation between virtual machines is dynamic and easy to change. This means users can minimize the overall number of required resources by switching them between virtual machines as required.

For example, if there are two operations, one online processing and one batch based, the online operation typically has high loads during the day, while the batch operation has high loads at night. To minimize costs, the user could setup two physical servers, one small and one large, and switch the online and batch operations between them each night and morning. However, this type of frequent exchange is unrealistic using physical servers. As a result, users tend to prepare two large-scale servers, one for each service (Top of Figure 7). Using the PRIMEQUEST virtual machine function users would be able to use just one physical server and reallocate resources between the online operation (guest) during the day and the batch operation (guest) at night (Bottom of Figure 7). Where the peak load times for each operation differ, as shown, overall costs can be reduced by only acquiring physical resources for the heaviest load, not for the total of each load peak.

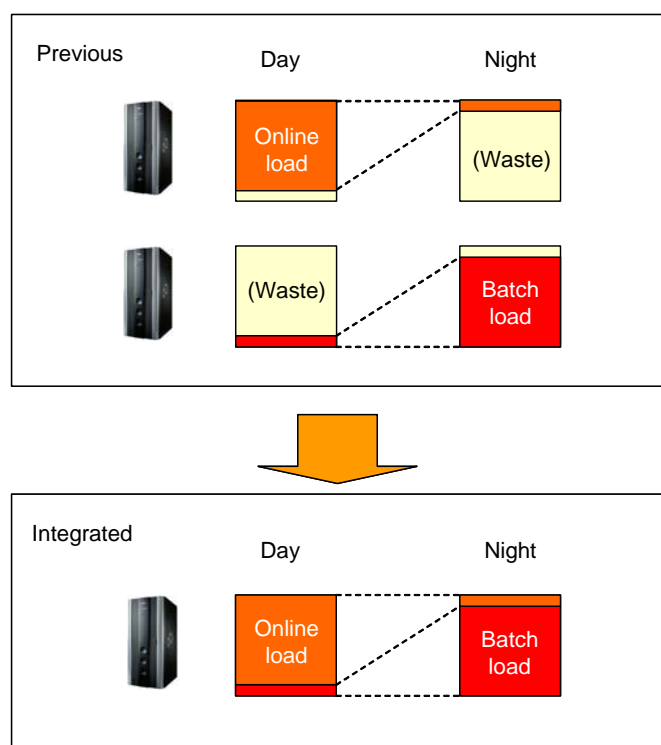


Figure 7. Dynamic resource allocation

4.5 Integration of Standby Systems

Efficiency	Agility	Continuity
Excellent	--	Excellent

For high levels of business continuity, users need to set up redundant servers in case one server fails. As a result high-reliability clustering also needs to be setup, plus a method of switching to the standby system if a failure occurs in the active system. Previously, if there were multiple active servers users would have to setup the same number of physical standby servers. However, with the probability of a problem occurring in any one active system quite low, the cost of so many standby servers was often a thorny management issue.

With the probability of simultaneously switching of multiple servers to the corresponding standby system extremely small, it is possible to combine the standby systems into one server using the PRIMEQUEST virtual machine function. Typically, only the minimum number of system resources is allocated to each standby virtual machine. Later following the switch to the standby system, allocation of sufficient system resources on the virtual machine can occur. By operating the system in this way, users can achieve high availability with a smaller investment.

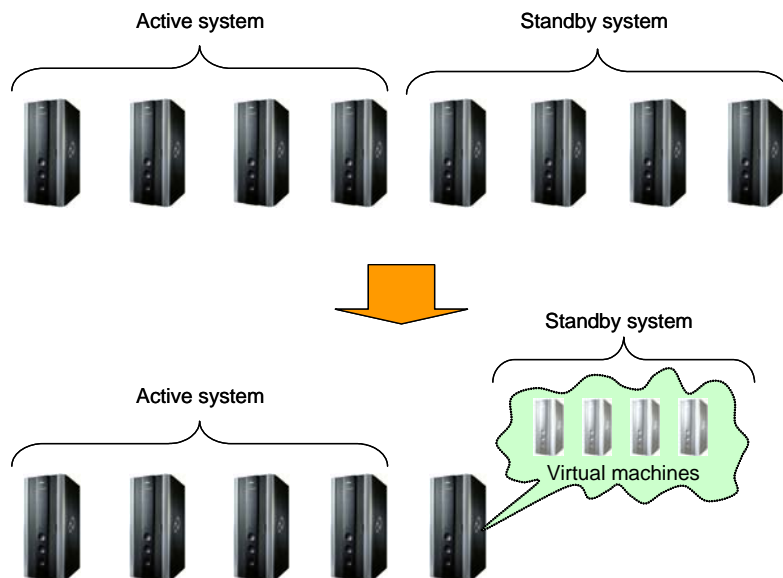


Figure 8. Sharing of the standby system machine

4.6 Service Continuity during Machine Downtime

Efficiency	Agility	Continuity
--	Good	Excellent

Regular maintenance of physical machines probably requires turning off the server from time to time. Using the PRIMEQUEST virtual machine function, organizations can ensure business continuity by moving important virtual machines to another physical machine.

If short-term service stoppage is possible, users can temporarily stop the operating systems of the virtual machines, use a migration tool to move the virtual machines to another physical machine, and then restarts the operating systems (Static Migration). Even if short-term service stoppage is not possible, user will soon be able to move the virtual machines to another physical machine while the operating systems are still running (Dynamic Migration, **Planned**).

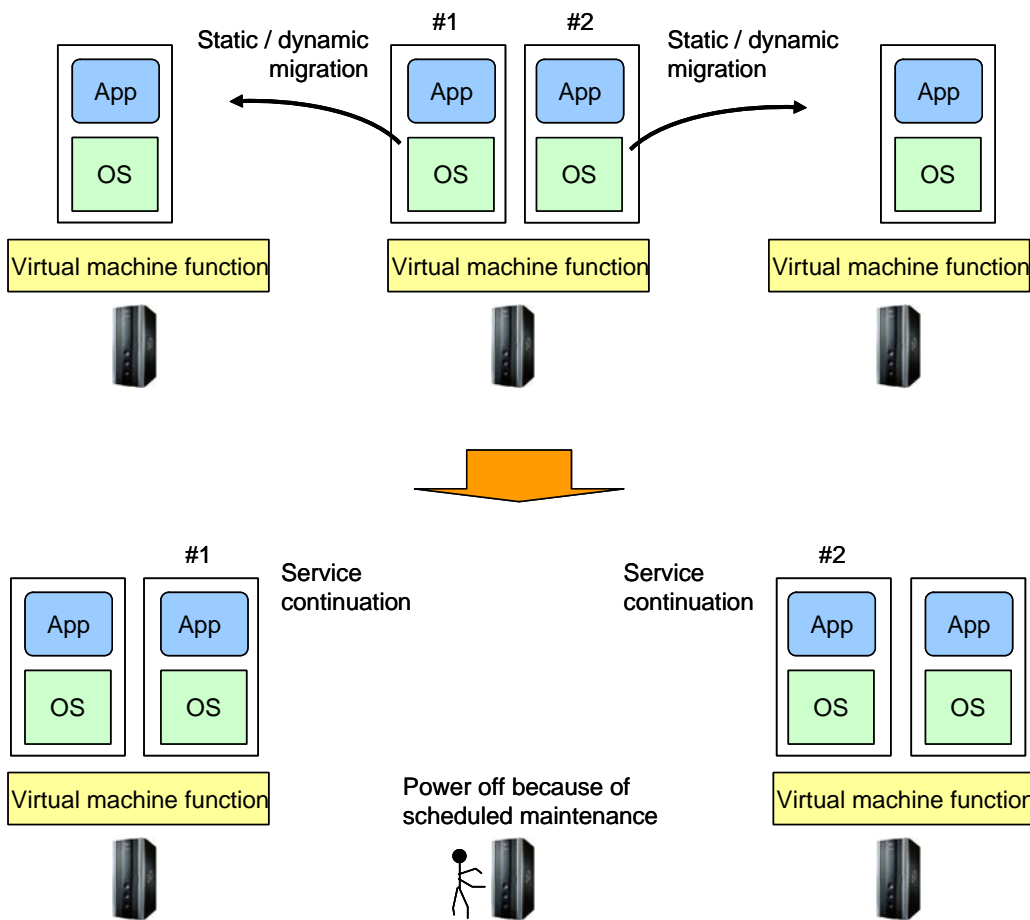


Figure 9. Service continuity during scheduled maintenance

5. Virtual Machine Function Implementation Technology

This section describes the implementation technologies users need to understand before using the PRIMEQUEST virtual machine function.

5.1 Overall Structure

The PRIMEQUEST virtual machine function employs a hypervisor method of control. The PRIMEQUEST virtual machine function is composed of a hypervisor and a host OS. The host OS includes software for managing the virtual machine function. The host OS is not used to run general business applications.

General business applications run on guest OS. A guest OS operates in a virtual machine on top of the hypervisor (Figure 10). Multiple guest OS can run simultaneously.

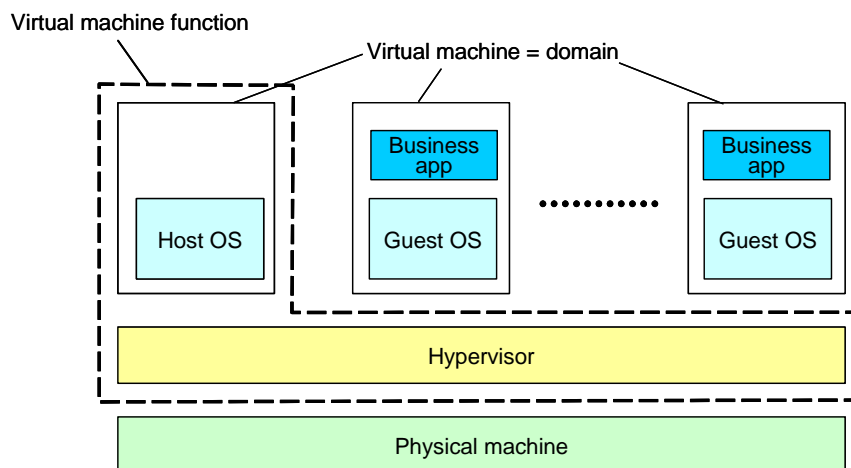


Figure 10. Configuration of the virtual machine function

The virtual machine function provides a virtualized, self-contained computing environment for each guest OS. Viewed from the guest OS, there is no difference in operation between a physical or virtual machine environment. Within the PRIMEQUEST virtual machine function, each virtual machine is called a domain. Each guest OS is isolated by domain walls. This means that even if a particular operating system becomes unstable or hangs, all other guest operating systems continue operating as normal. “Native environment execution” is used to distinguish direct use of an operating system on a physical machine, while the operating system operating directly on the physical machine is called a “native OS”.

5.2 The Virtual Machine Function as Partitioning Technology

The use of the virtual machine function to operate multiple operating systems on one physical machine can be viewed as software partitioning of the physical machine. This provides additional functionality to the hardware partitioning technology of PRIMEQUEST also known as PPAR and XPAR. Table 1 compares these methods.

As shown in the table, PPAR and XPAR are best suited for use where even a slight deterioration in reliability or performance is unacceptable. The virtual machine function however suits cases where fine division of resources (granularity) and operational management flexibility are required. It is possible to combine the two. This enables the operation of virtual machines within PPAR or XPAR hardware partitions. Figure 11 shows an example wherein a PRIMEQUEST system with four system boards (SB) is divided into three PPARs, with one PPAR further divided using the virtual machine function.

Table 1. Comparison of hardware and software partitions

	Implementation by	Granularity	Key Features
PPAR, XPAR	Hardware	Large (PPAR: 1SB*, XPAR: ½ a SB)	Highest reliability, robust hardware isolated partitioning No performance degradation.
Virtual machine	Software	Small and flexible (one CPU core can be divided in multiple resource units)	Flexibility in software management. Dynamic modification of resource allocations Resources can be shared among guest OS.

* SB: System board. In the PRIMEQUEST500 series, one SB is equivalent to four CPU sockets.

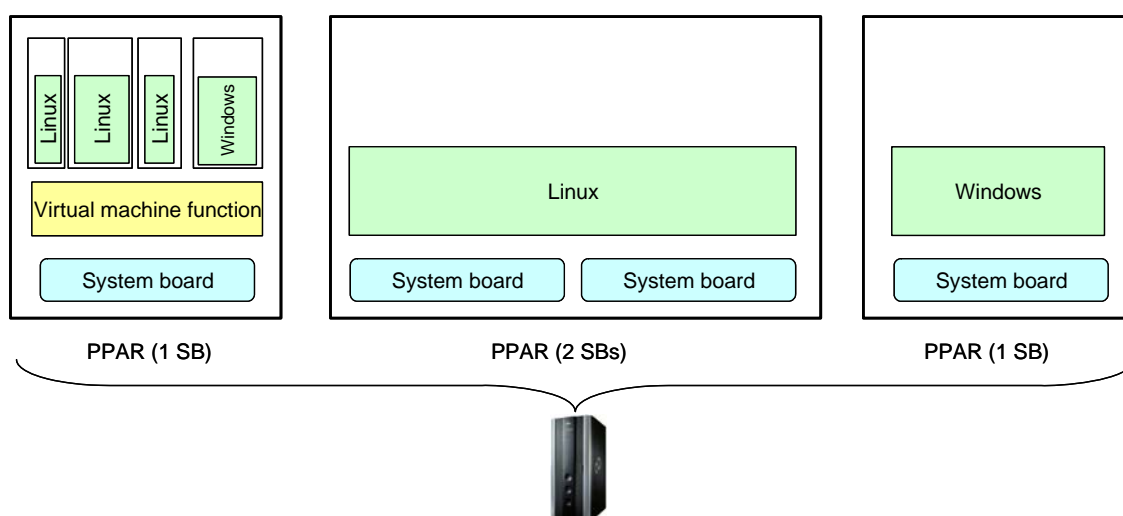


Figure 11. Example of combining PPAR and virtual machine functions

5.3 Full Virtualization and Paravirtualization

There are two machine virtualization methods: paravirtualization and full virtualization. Paravirtualization partially changes the operating system kernel (core of the operating system) on the physical machine for use in virtual machines. Full virtualization makes no changes to the operating system kernel. Domains using the full virtualization method of the PRIMEQUEST virtual machine function are called hardware-assisted virtual machine domains (HVM domain) because they use the virtualization support function (Intel VT-i) in the CPU hardware. Domains that use paravirtualization are called paravirtualized domains (PV domain). Both HVM and PV domains can co-exist on the same hypervisor.

Table 2. PV domain (paravirtualization) and HVM domain (full virtualization)

Domain type	Operating system kernel	Note
PV domain (paravirtualization)	Kernel revised for PV	A PV kernel must be setup. The host OS is always PV.
HVM domain (full virtualization)	Same as native kernel	The CPU must feature a virtualization support function (Intel VT-i). A PV device driver for the HVM domain is needed to obtain good I/O performance.

The PRIMEQUEST virtual machine function uses full virtualization for guests to operate Windows Server 2003, which does not support paravirtualization, and to secure consistency among operating system kernels when physical and virtual machines are clustered.

5.4 CPU Virtualization

For server virtualization, the key components, CPU, memory, and I/O devices are separately virtualized. This section starts by describing CPU virtualization.

CPU virtualization means creating the equivalent of a physical CPU (pCPU) for each domain. The CPU resource as seen from a domain is called a virtual CPU (vCPU).

The PRIMEQUEST virtual machine function allows one physical CPU to be shared as the vCPUs of multiple domains. When sharing a pCPU, administrators can control each domains share by either guaranteeing a particular CPU performance or by limiting the CPU performance of each vCPU.

The CPU scheduler built into the hypervisor implements CPU virtualization. It determines which vCPU can use each pCPU (Figure 12). At any given moment, when seen from the Domain side its vCPU is either associated with a pCPU or disassociated (in a standby state) with it. The operating system has its own internal CPU scheduler and since this scheduler determines the process for which a virtual CPU is used, an overall two-phase scheduling process is used.

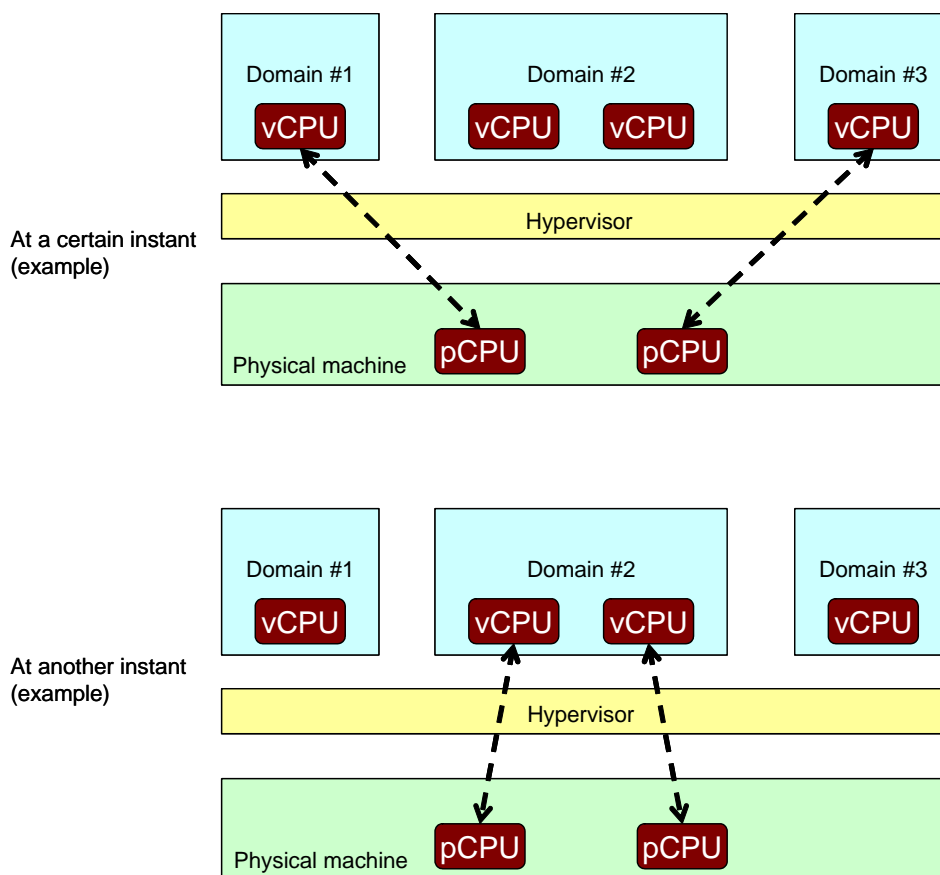


Figure 12. Example of CPU scheduling using the hypervisor

5.5 Memory Virtualization

For memory virtualization, the PRIMEQUEST virtual machine function divides the physical memory of the machine and allocates it to the domains. For any guest OS, it appears that a contiguous physical memory area (guest physical memory) exists. However, importantly, each guest OS cannot view the memory of other domains. The physical memory size allocated to each domain is specified from the host OS.

It is the hypervisor layer that executes address conversion, in page units, from guest physical addresses to machine physical addresses. Since the operating system also performs memory virtualization, two-phase address conversion takes place overall, as shown in Figure 13.

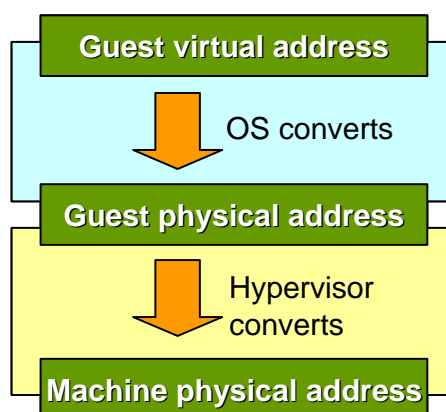


Figure 13. Two-phase address conversion via the operating system and hypervisor

Currently, memory virtualization of the PRIMEQUEST virtual machine function only performs address conversion. It does not save data to secondary storage (disk). Saving data to secondary storage is performed by the operating system layer.

5.6 I/O Virtualization

For I/O virtualization the PRIMEQUEST virtual machine function implements virtual I/O devices seen by each guest OS from the I/O devices of the physical machine. The mapping of virtual devices and physical devices takes place within the host OS.

Although a physical I/O device can be mapped exclusively to one guest OS, a Fibre Channel card, network card, or disk volume can also be shared by multiple guest OS. However, even when shared, the virtual machine function never shares data. For example, the virtual machine function does have a function for sharing one file among multiple guest OS. So the distributed file system functions of the operating system layer are used.

There is also no longer a physical maximum to the number of I/O devices. There is a maximum number of physical cards, determined by the number of server card slots, but the system is no longer constrained by such physical numbers as cards can be shared by guest OS.

In addition through I/O virtualization, a guest OS can use the latest physical devices even if the guest OS itself does not support the latest format devices.

The PRIMEQUEST virtual machine function I/O virtualization supports the following methods:

- Device emulation

- Virtual device drivers
- Direct I/O (Planned)

5.6.1 Device Emulation Method

The device emulation method uses software to emulate I/O device hardware seen from the CPU. Each guest OS uses the same device drivers used by the native operating system. Each time a CPU of a guest domain accesses a device control register or memory, at the machine instruction level, control is passed to the device emulator in the host OS. The device emulator then executes the same processing as the actual I/O device. This method enables emulation of older specification using device of the latest format. It also enables emulation of devices of Company A specifications using a device of Company B specifications.

Although the device emulation method has the advantage of not requiring the user to change the device drivers of the guest OS, it can have a large execution overhead. This method should therefore only be used where the next “virtual device driver method” cannot be used for a low-speed device or guest OS. It should not be used with disks containing business data nor with network connections.

The device emulator of the PRIMEQUEST virtual machine function cannot be used from a guest domain that uses the paravirtualization method.

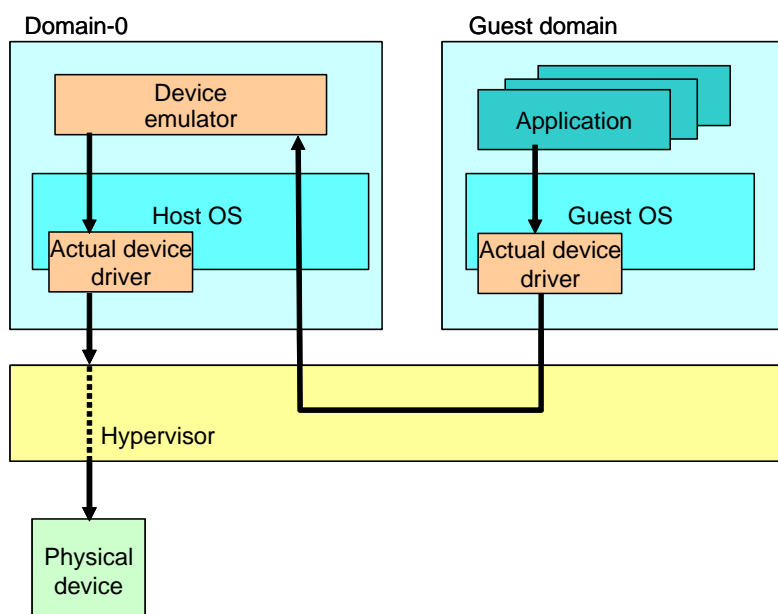


Figure 14. Device emulation method

5.6.2 Virtual Device Method

The virtual device method defines abstract virtual devices, installs virtual device-specific drivers in the guest OS and uses those device drivers. Since these drivers operate in the virtual environment, they are called PV drivers. There are PV drivers for PV domains, and PV drivers for HVM domains. Both types of PV drivers are operated by linking them with a backend device driver in the host OS via the hypervisor.

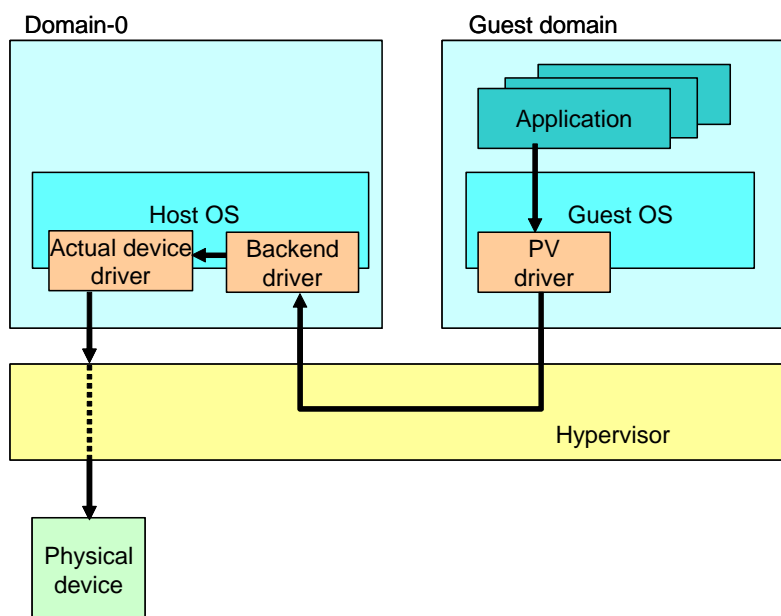


Figure 15. Virtual device method

The virtual device types include virtual block device (VBD), virtual network interface (VNIF), virtual SCSI (VSCSI), and virtual frame buffer (VFB).

Compared with the device emulation method, the virtual device method has fewer exchanges with the host OS and therefore achieves I/O throughput near to that of a native environment. By collecting all access requests from multiple guest OS to the host OS, the system can share I/O devices among the guest OSes. Sufficient CPU performance required for the I/O volume must be allocated to the domain of the host OS.

5.6.3 Direct I/O Method (Planned)

The direct access method is for accessing I/O devices directly from a guest domain. Processing takes place via the host OS only during device allocation and release. The guest OS uses the same (or similar) device drivers as the native environment. With the direct I/O method, the guest domain achieves nearly identical I/O performance as

the native environment because it does not go through the host OS for data exchange. The direct I/O method requires hardware that supports direct I/O. Generally, the direct I/O method does not support device sharing among guests. For device sharing to take place, the physical device must support a sharing function.

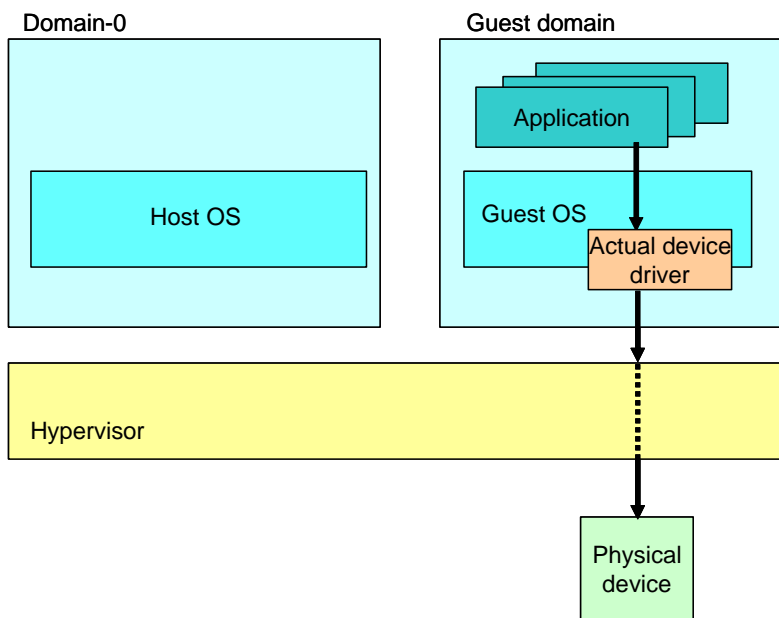


Figure 16. Direct I/O method

5.7 Disk Virtualization

Virtualization of a disk unit as an I/O connected device, is an example of I/O virtualization. Using the device emulation and virtual device methods, the guest disks and the physical disks are associated in the host OS as shown in the next figure.

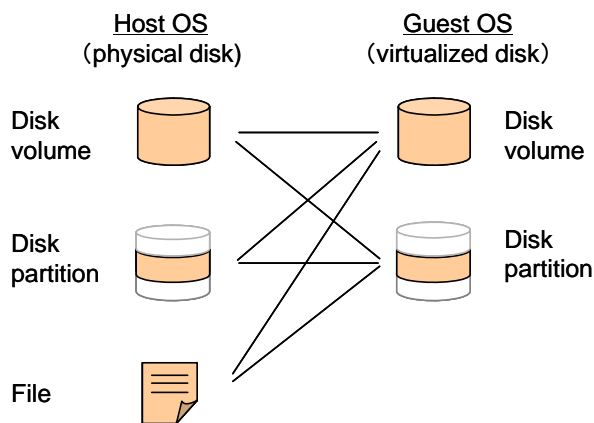


Figure 17. Disk virtualization

The user can share one disk volume from multiple guests by dividing the disk volume on the host OS side into disk partitions and associating each partition with a disk volume of a different guest OS.

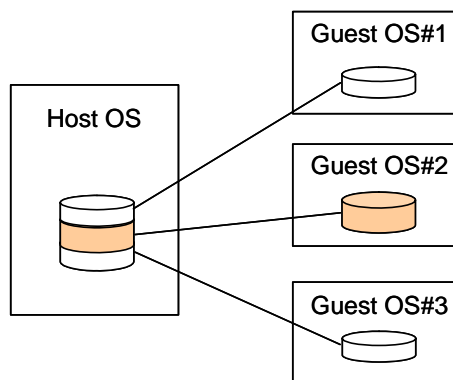


Figure 18. Sharing a disk among guests

When both performance and high reliability are required, users can associate the host OS side with a disk or partition. As a rule, any entity can be used as a disk or partition on the host OS side as long as it can be seen as a disk or partition by the host OS. For example, a virtualized disk in a disk array, such as a Fujitsu ETERNUS storage system can be used.

Associating the host OS side with a file has the advantages of easier management. When a guest requires a disk file, it can easily be provided by creating a file on the host OS side. However, in comparison with associating the host OS side to a disk or partition, associating the host OS side to a file has a performance disadvantage.

5.8 Network Virtualization

By network interface (NIF) virtualization an administrator can provide a virtual network interface (VNIF) to guest OS without being constrained by the number of physical LAN cards. One physical NIF can be shared by multiple guest VNIFs. To do this, an Ethernet switch (virtual bridge) implemented by software is set up on the host OS side (Figure 19). The host OS sets up the associations between the VNIFs, the virtual bridges, and the physical NIFs.

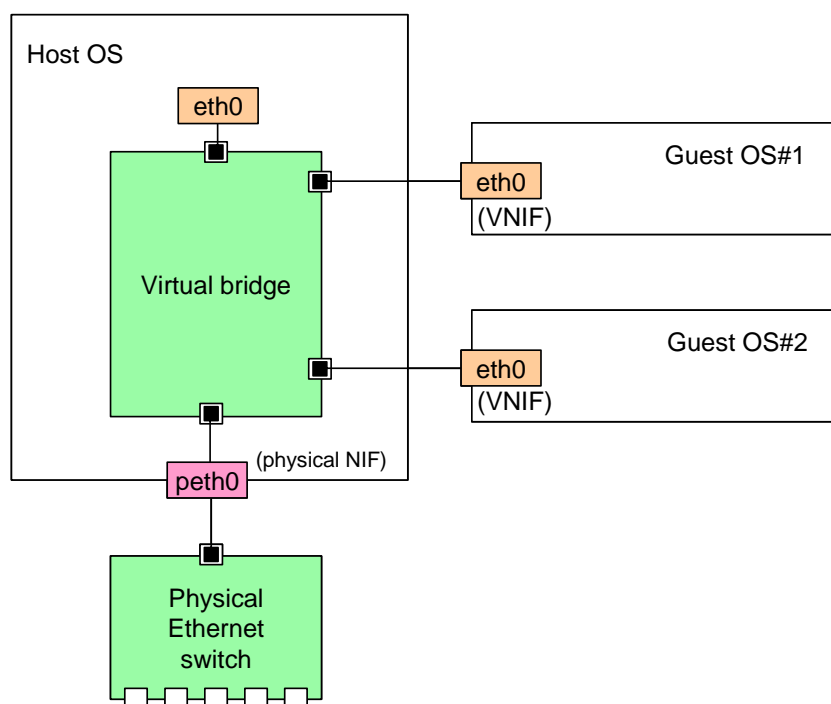


Figure 19. Sharing a network card among guests

To associate two VNIFs to separate physical NIFs for the purpose of redundancy or performance separation, two virtual bridges associated with two physical NIFs and VNIF connections to each bridge must be set up.

If performance is a priority, one VNIF, as seen in a guest domain, should be associated to one physical NIF. The guest therefore has exclusive possession of the physical NIF.

A media access control (MAC) address, not duplicated on the LAN, must also be assigned to the VNIF.

6. Virtual Machine Duplication and Migration

6.1 Cloning (Virtual Machine Duplication)

A software stack (same combination of operating system to application) of an existing virtual machine can be copied to a separate virtual machine using the cloning function of the operation management tools. This enables fast and simple preparation of the same execution environment just by carrying out a machine-specific setup.

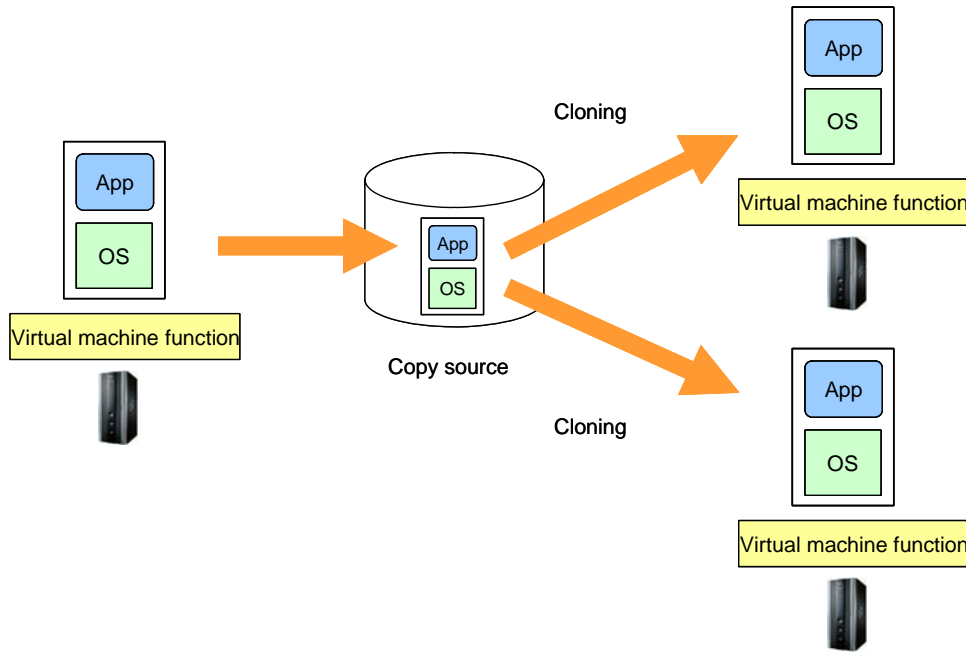


Figure 20. Cloning

6.2 Static Migration (Static Moving of a Virtual Machine, Planned)

Users may want to move an operating system and its applications (software stack) between a virtual machine and a physical machine, or between two virtual machines on different physical machines. There is a method that temporarily stops the operating system during the move. Called “static migration”, Fujitsu plans to release a migration tool that supports this method as shown below.

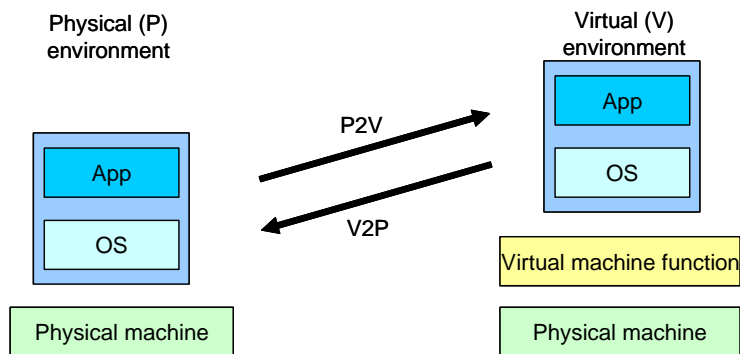


Figure 21. P2V and V2P

(1) Physical machine to virtual machine (P2V)

The P2V tool moves a software stack running on a physical machine (P) to a virtual

machine (V). This can be used, for example, to virtualize and then integrate jobs that were previously performed on multiple servers onto multiple virtual servers on a single physical machine.

(2) Virtual machine to physical machine (V2P)

The V2P tool moves a software stack operating as a virtual machine (V) to a physical machine (P). This tool is used when development is performed on a virtual machine but actual operation is performed on a physical machine.

(3) Virtual machine to virtual machine

This tool temporarily stops the operational system before moving the software stack between two physical machines that use the same PRIMEQUEST virtual machine function. This tool is used, for example, for long-term load distribution and for continuing service during scheduled maintenance.

6.3 Dynamic Migration (Dynamic Moving of a Virtual Machine, Planned)

Moving an operating system and its applications from a virtual machine to a virtual machine while the operational system remains running is called dynamic migration or live migration. Dynamic migration enables the following types of operation:

- Dynamic load distribution: The ability to move a guest domain from a high-load server to a low-load server and optimize the load without operational halt.
- Service continuation during hardware maintenance: If a physical machine must be stopped for scheduled maintenance, guest domains can be moved to another physical machine, and services continued.

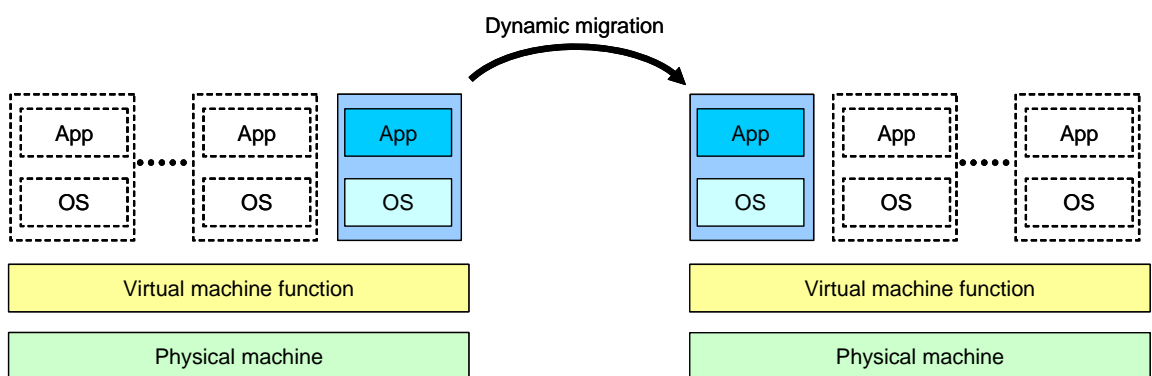


Figure 22. Dynamic migration

7. Conclusion

This white paper describes the PRIMEQUEST virtual machine function provided by

Red Hat Enterprise Linux 5.

Fujitsu provides its customers with support services that include Fujitsu's proprietary value-added software for installation support, performance enhancement, and reliability enhancement for this virtual machine function. Fujitsu is also continuing its efforts to improve further the reliability and usability of virtual machine functions in cooperation with Red Hat.