PRIMERGY 6000 Series: Business Servers Providing Strong Support for Enterprise Systems

● Toyoaki Nakamoto ● Kenichi Nakano ● Kouji Aratake

Following on from the top-selling K series of midrange computers, the PRIMERGY 6000 series offers a range of powerful business servers that can be used as the main server for medium-sized companies or individual departments. This series was developed to provide significant benefits in terms of increased reliability, high performance, and compatibility with legacy systems while being able to adapt flexibly to changes in the business environment such as the Internet and personal computers. Consequently, the hardware and operating system of the K series—whose architecture was developed independently by Fujitsu—have continued to provide new functions while evolving and fusing with open-source technology. This paper introduces the PRIMERGY 6000's dependable technology and operating system features that we have developed over the years and describes our approach to virtualization in the Cloud computing era.

1. Introduction

Fujitsu entered the midrange computer market in the 1970s with the V series and later produced the best-selling K series and GRANPOWER6000 series before the PRIMERGY 6000 series (abbreviated to PG6000 below) (**Figure 1**). Owing to our extensive experience in this field and use of the latest technologies, our systems have been used for many years as departmental servers or as mission-critical host systems for medium-sized enterprises by taking over the extensive legacy assets of our customers. By migrating the primary communication protocol from FNA (Fujitsu Network Architecture) to TCP/IP (transmission control protocol, Internet protocol) and providing support for open-source technology such as the Worldwide Web, we are continuing to provide solutions that are ideally matched to the Internet era.

The PG6000 is designed to be installed in offices, where it achieves a high degree of interoperability in integrating hardware and services. It offers enhanced reliability by incorporating monitoring functions and redundant configurations into various components, while still being designed to operate silently so as not to disturb office workers. It also includes, as standard, functions that enable it to operate automatically according to a schedule, allowing it to support efficient business operations.

Furthermore, it uses a proprietary microkernel that allows legacy COBOL (common business-oriented language) applications to run simultaneously with open-source C and Java software. These technologies make it possible to use COBOL applications from a Web browser and to link up with the database in a Windows Server.

This paper introduces the PG6000 hardware and the dependable high-performance and high-reliability technology that we have implemented over the years with the ASP (Advanced System Products) operating system (OS) used on this hardware. It also introduces our approach to

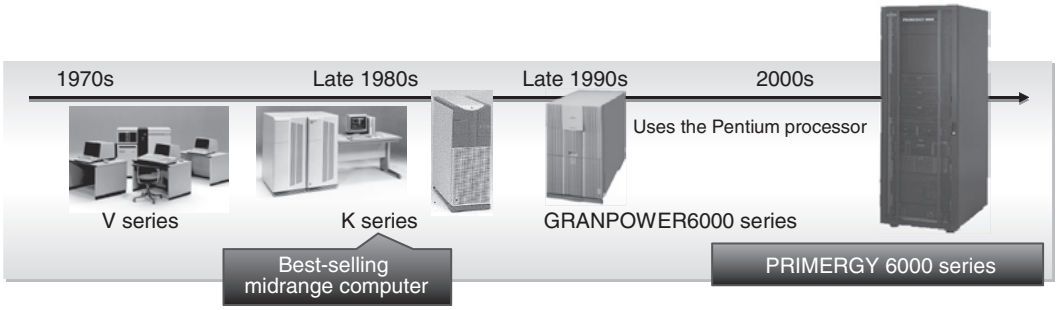


Figure 1
History of Fujitsu's midrange computers.

virtualization technology, which has recently attracted interest.

2. PG6000 features

The PG6000 features high reliability, high performance, and backwards compatibility with legacy assets in order to provide many years of trouble-free use. For the K series or the former servers, we produced dedicated hardware systems incorporating our proprietary processor (FSSP: Fujitsu Small System Processor). The operating system installed on this hardware was CSP/FX (CSP: Customer-oriented System Products), based on a proprietary architecture implemented with dedicated instruction code.

In the PG6000, a proprietary hybrid configuration is implemented in both hardware and software by combining the K series architecture with open-systems technology. This provides the following merits:

1) Integrated system-wide monitoring of hardware and software

Dedicated hardware called the System Monitoring Mechanism is incorporated into the Intel Architecture (IA) server developed by Fujitsu. This hardware is closely bound to the operating system to provide the following functions for enhanced reliability and operability.

- Monitoring for failures in hardware components such as power supplies and fans
- Automatic operation control based on an internal calendar

2) Optimized dynamic object translation

Object programs written in the instruction code of our proprietary processor are automatically converted into Intel processor instructions and executed directly at high speed. Instruction code sequences that are executed frequently are optimized and cached for even greater speed enhancement.

3) Hybrid OS based on microkernel

For better portability of open-source software while running COBOL applications such as batch processes at high speed, the system uses a hybrid structure that combines an OS environment running in the conventional mode with an operating environment that supports a portable operating system for Unix (POSIX) application programming interface (API) and a hierarchical file system.

The next section discusses the internal operation of these technologies.

3. Hardware

The PG6000 series comprises four different models and eleven different types, and covers a performance range of 40 times from the model 900, which is designed to act as the core of large-scale mission-critical server systems, to the model 300, which is the space-saving lower-end model. All of these models and types provide high reliability for mission-critical services and

facilitate the installation and operation of office servers. They have the same architecture and include the following basic features.

3.1 Advanced system operation management/monitoring

All systems in the PG6000 series are fitted with an Integrated Service Facility (ISF), which is a set of system monitoring mechanisms on dedicated processors operating separately from the central processing unit (CPU). ISF achieves high reliability and operability across the PG6000 series by providing the following functions.

1) Automatic operation function

This function switches the system on and off and controls OS interrupts according to a calendar schedule.

2) System monitoring mechanism

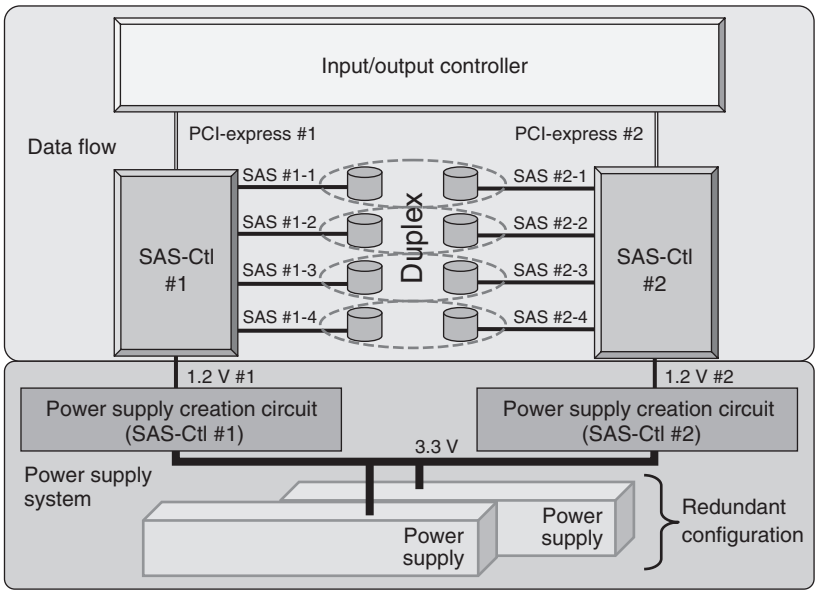
This mechanism continually performs centralized monitoring of the environmental conditions (temperature, voltage, cooling fan rotation speed, etc.) and checks for failures in the hardware, firmware, and software. When it detects a failure, it notifies the system administrator by logging the failure and generating error messages; in some cases, it can even initiate an emergency system shutdown.

3) Remote maintenance function

A remote control card installed in the PG6000 enables remote maintenance operations to be performed from Fujitsu's online maintenance service center.

3.2 High reliability

All products in the PG6000 series are fitted with an uninterruptible power supply (UPS) and duplex disk systems. These are highly reliable mechanisms used in the mission-critical systems and they completely prevent system stoppage and system data loss owing to disk failures or power failures (**Figure 2**). Furthermore, the latest models that started shipping in August 2009 are not only provided with duplex disk devices but also have redundant disk interfaces and power supplies, which improve the disk availability even further.



Ctl: Controller
PCI: Peripheral component interconnect
SAS: Serial attached SCSI (small computer system interface)

Figure 2
Duplex disk configuration.

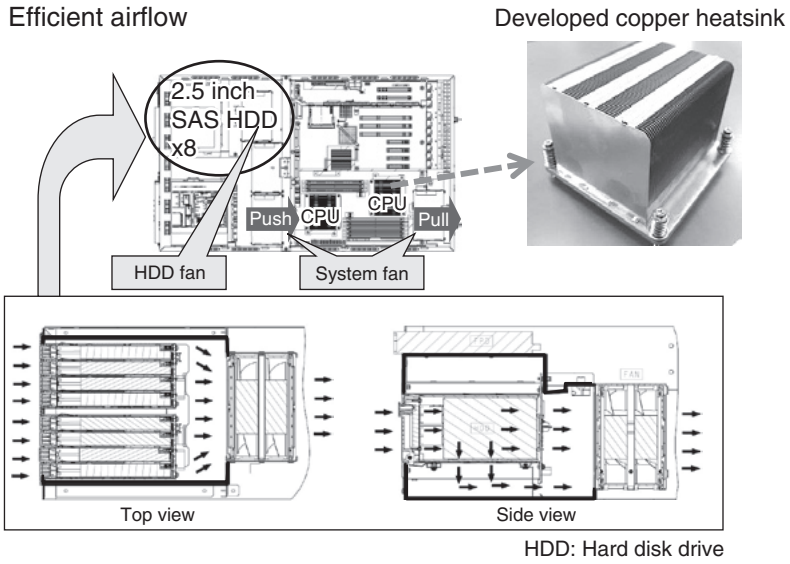


Figure 3
Duct configuration.

3.3. Quietness

Since PG6000s are often installed in ordinary offices, they are subject to strict noise level requirements. An ongoing theme in the evolution of our product line-up is that newer equipment should always be quieter than the equipment it replaces.

In our most recent models, we have completely redesigned the airflow to achieve a noise level of 42 dB, which is 3.5 dB lower than the original noise level of the same equipment. A 3-dB noise reduction means that the perceived noise level is approximately halved, so this is a significant achievement. In addition, by eliminating the CPU fan, developing a copper heatsink with better cooling efficiency, and choosing to use a duplex system-fan configuration, we have made it possible to prevent outages due to CPU fan failures (**Figure 3**).

4. Software

The switch from our proprietary processors to Intel processors presented us with the major challenge of guaranteeing that the OS (ASP) and the software assets of our customers remained compatible at the object level (to maintain backward compatibility with legacy resources).

We were able to overcome this challenge by implementing an interpreter in the microkernel that sequentially converts ASP instruction code into Intel processor instruction code (Intel code) and executes it. We have also developed a dynamic object translator that operates in tandem with the interpreter to achieve faster instruction execution.

To provide products that match the needs of our customers, we have also developed a unified POSIX function environment (UXF) operating mechanism that makes it easy to incorporate open-source technology.

4.1 Dynamic object translator (accelerated)

The dynamic object translator (OOCT: optimized object code translation) that uses optimization techniques for Intel code generation achieves speed increases of up to tenfold compared with the execution speed of a conventional interpreter. OOCT helps to increase the execution speed by generating optimized Intel code from ASP objects according to the following procedure.

1) When a branch instruction is executed, the

interpreter acquires branching information necessary for the conversion process (branch address, number of iterations, etc.). If the instruction at the branch destination has already been translated, then the translated code is executed. Otherwise, the instructions are executed by the interpreter. While these instructions are being executed, frequently called processes (hot paths) are detected from the branching information and submitted for translation by the object translation process.

2) The object translation process generates a flow graph from the branching information to discover sequences of processing units that are executed with high frequency and translates them into an intermediate language that facilitates optimization analysis.

3) The intermediate language is optimized by processes such as flow analysis to remove unnecessary labels and code, eliminate common sub-expressions, and convolute constants and instructions.

4) Intel code is generated from the optimized intermediate language.

4.2 UXF operating mechanism

The UXF operating mechanism is an execution environment equipped with a POSIX compliant function interface and a tree-structure file system that is equivalent to an open-standard file system. This execution environment is based on a BSD (Berkley Software Distribution) UNIX system and is built on the microkernel (**Figure 4**).

With the aim of incorporating open-source technology, we ported various open-source products such as Samba (file sharing), Apache (Web server), Tomcat (servlet container), and JavaVM (Java virtual machine) to our UXF operating mechanism.

The UXF operating mechanism is designed so that these open-source functions can be operated and managed from the conventional ASP environment, allowing customers to draw on their existing ASP skill base. For this purpose, we have developed a UXF cooperation mechanism

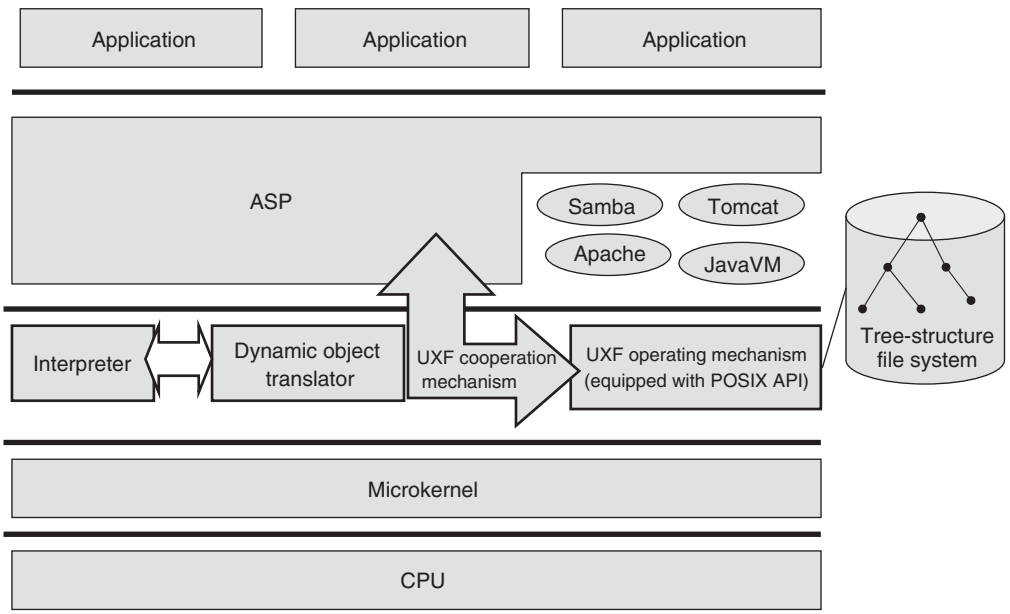


Figure 4
System configuration.

that links together the operating mechanisms of ASP and UXF.

4.2.1 Tree-structure file system

The conventional file system of ASP has a two-level configuration comprising storage devices and the actual files. The names of storage devices and actual files are limited to eight characters in the EBCDIC+JEF code system. This made it difficult to incorporate open technologies and circulate assets together with open systems. To address this problem, we developed a tree-structure file system (using the JIS 8-bit and Shift_JIS code systems) that is equivalent to an open-standard file system. This made it possible to share resources easily with open-standard systems by using file-sharing (Samba) and file transfer functions based on FTP (file transfer protocol).

4.2.2. UXF cooperation mechanism

The UXF cooperation mechanism provides the following functions that connect the ASP and UXF operating mechanisms so that operation and management of the UXF operating mechanisms can be performed from the ASP environment:

1) Starting, stopping, and canceling programs in the UXF operating mechanism from ASP
2) Inheriting the ASP job environment (user-authorized information) by processes running on the UXF operating mechanism
3) Reporting execution results and operation errors of programs running on the UXF operation mechanism to ASP
4) Accessing the tree-structure file system from ASP
5) Relaying standard output data generated on the UXF operating mechanism and outputting it to the ASP spool queue

5. Virtualization technology

Owing to the increased threats and intensified competition with global markets resulting from the spread of information, our customers are finding it necessary to adapt even faster to changes in the business environment. The users of midrange computers are no exception, and they face the need to rapidly construct optimal information and communications technology (ICT) infrastructures that are ideally suited to their business needs while maintaining their existing business.

Server virtualization[1)] is a technique that solves these issues. It not only allows customers to inherit the operating environments of applications that they have already developed, but also has the following additional merits:

1) Businesses can be rapidly developed and reformed by flexibly reconfiguring the development environment and server configuration.
2) The total cost of ownership (TCO) is reduced by using virtualization technology to achieve optimal placement of the ICT infrastructure.
3) Space and power are conserved by integrating multiple PG6000 systems into a single server.

For this purpose, we are planning to offer an ASP Execution Kit (virtual machine function) based on Fujitsu's Mission-Critical IA Server PRIMEQUEST as a system that can be adapted to in-house Cloud computing. This mechanism enables an ASP program running on a PG6000 to be executed on a highly reliable open platform in dozens of cases.

Our approach to virtualization technology is discussed below from the viewpoint of the ASP Execution Kit, which is currently under development (as of April 2011).

5.1 Overview of ASP Execution Kit

The ASP Execution Kit is a PRIMEQUEST add-on that includes components such as a CPU, memory, and internal hard disk drive. The hard disk drive stores the firmware for the ASP Execution Kit and a model system image of the OS. Two Intel processors for use by

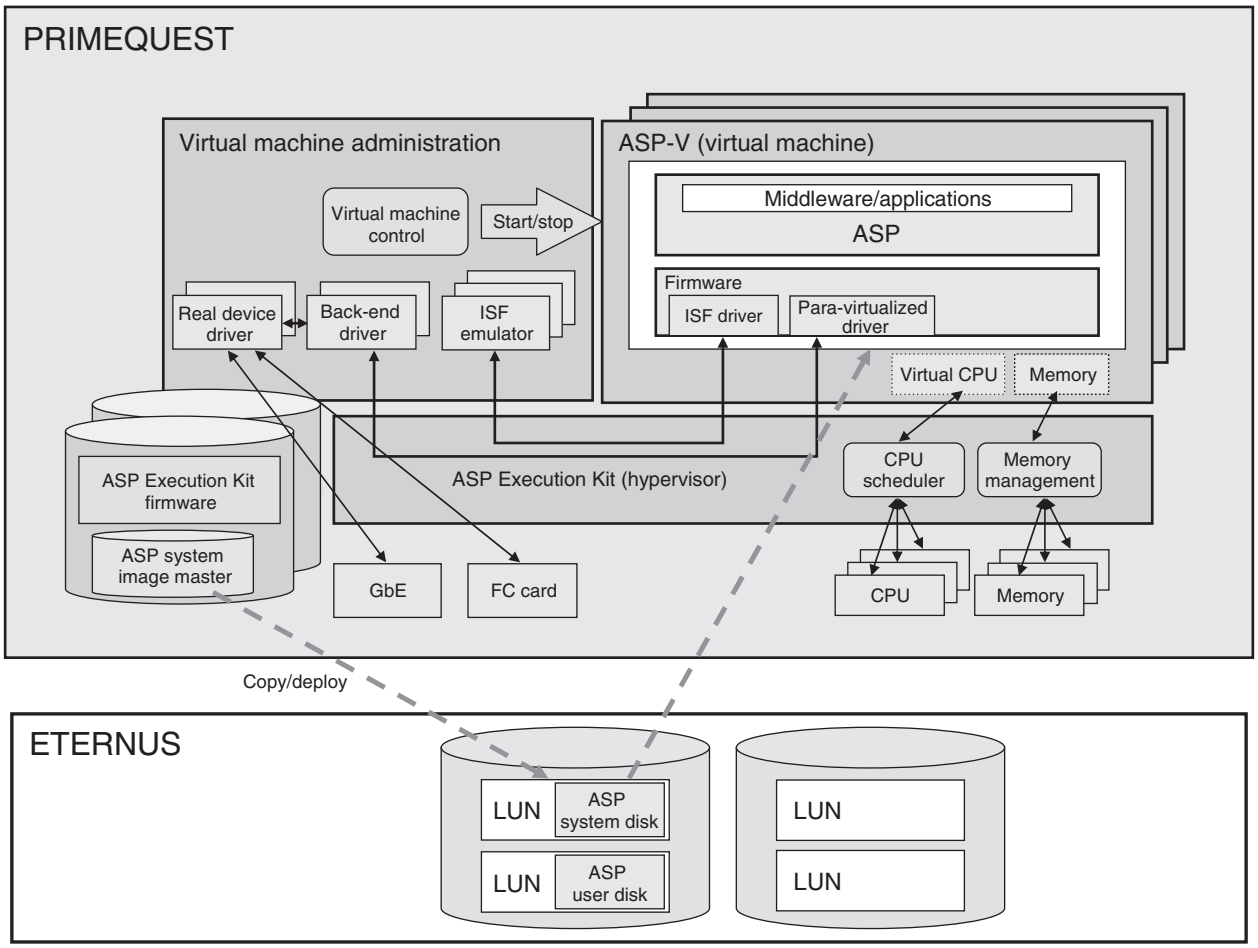
PRIMEQUEST and up to 256 GB of memory can be mounted on a single system board.

5.2 Functional overview of ASP-V (virtual machine)

The virtualized ASP system disks and user disks are deployed on an ETERNUS disk array device. ASP virtualization is implemented by the hypervisor functions of the ASP Execution Kit. A virtual machine in which ASP has been deployed is called an ASP-V. A functional overview of an ASP-V is shown in **Figure 5**. Since conventional ASP operates normally in an ASP-V, the ASP middleware and applications run with binary compatibility, and multiple ASP-V systems can be deployed.

The hypervisor's CPU scheduler associates the time-shared PRIMEQUEST physical CPU with ASP-V virtual CPUs. Moreover, if from one to eight virtual CPUs are allocated to a single ASP-V, one can create virtual machines with a wide range of CPU performance from the model 300 class through to the model 900 class of the conventional PG6000 series. Memory mounted in the physical hardware is allocated to each ASP-V according to its requirements.

The PG6000 is also characterized by the ISF hardware, which is emulated by software to guarantee compatibility with ASP. This makes it possible to perform automatic operations such as scheduled power up and down and job schedule management in the same way as on conventional



FC: Fibre Channel
GbE: Gigabit Ethernet
LUN: Logical unit number

Figure 5
ASP-V operational overview.

equipment.

Access to the operation panel, error log information, and the like of PG6000 is provided by a Web admin console function.

Furthermore, with regard to the virtualization of input/output (I/O) resources such as the disks and network, para-virtualized drivers have been developed so that the I/O devices can be used via back-end drivers on the virtual machine administration. This method enables I/O devices to be shared from multiple ASP-V systems while achieving an adequate level of performance.

6. Conclusion

The ASP Execution Kit virtualizes the ASP operating environment and can be flexibly constructed to continue running the business logic of our customers' assets such as COBOL applications in the most suitable way. This dramatically reduces the required amount of system reconfiguration, which has hitherto taken a long time to achieve, and shortens the development/operation lifecycle, enabling customers to quickly transform their business processes.

This paper described the merits of the PG6000 hardware and software and our approach to virtualization technology. In future, we will continue to expand this virtualization technology into ASP Cloud services to provide our customers with services that allow them to focus on business strategy without having to consider its effects on their ICT infrastructure.

Reference

1) Y. Oguchi et al.: Server Virtualization Technology and Its Latest Trends, *Fujitsu Sci. Tech. J.*, Vol. 44, No. 1, pp. 46–52 (2008).

Toyoaki Nakamoto
Fujitsu Ltd.
Mr. Nakamoto is engaged in the development of the PRIMERGY 6000 systems.

Kenichi Nakano
Fujitsu Ltd.
Mr. Nakano is engaged in the development of the PRIMERGY 6000 virtualization technology.

Kouji Aratake
Fujitsu Ltd.
Mr. Aratake is engaged in the development of the PRIMERGY 6000 operating system.