Grid Middleware for Realizing Autonomous Resource Sharing: Grid Service Platform

● Soichi Shigeta ● Haruyasu Ueda ● Nobutaka Imamura
(Manuscript received April 19, 2007)

These days, many enterprises are interested in consolidating systems to achieve more efficient operation of their IT resources. In particular, Grid techniques are attracting keen attention because distributed systems can be used collectively as a single system. The changes in load that occur during business operations cause fluctuations in the number of servers needed, and Grid systems must flexibly allocate server resources to respond to these fluctuations. Therefore, in conjunction with France Télécom, we have developed a middleware called Grid Service Platform (GSP) for realizing autonomous resource sharing based on the priority of business operations. We implemented a batch service and a real-time online service and then evaluated GSP on a testbed consisting of sites from Japan and France. We confirmed that GSP enhanced the online service capacity by 30% or more, while simultaneously reducing the runtime of the batch service by 12%.

1. Introduction

Grid computing has developed as a technology to provide a computational environment for large-scale distributed arithmetic processing required for science and engineering, for example, molecular dynamics simulation and parameter searching for optimization problems. However, it has recently come to the forefront as a solution that employs various business systems distributed at each branch office as a virtual single system.

Telecom carriers are also very interested in sharing servers among multiple business systems. These carriers offer a variety of business operation services, for example, Web content servers for cellular phones and video conferencing on the Internet. Currently, the carriers construct a dedicated system for each business operation service. As a result, they cannot dynamically change the allocation of a server from a service with a light load to one with a heavy load. Existing dedicated systems are consequently designed with a large excess of performance and capacity to accommodate peak demand. By diverting surplus servers to other business operation services, the operational efficiency of a system can be improved and the total cost of ownership (TCO) reduced. Therefore, we have developed a middleware called Grid Service Platform (GSP)[1),2)] through cooperative research conducted with France Télécom to realize a Grid system for telecom carriers. GSP can 1) achieve not only batch processing but also online operation as a business operation service on a Grid system and 2) share servers autonomously among business operation services.

This paper discusses the challenges in developing GSP, gives a functional overview of GSP and describes its architecture, and summarizes the results of a demonstrational study conducted using actual business applications.

2. Challenges in developing GSP

To effectively share heterogeneous servers among multiple business operation services, the following problems had to be addressed.

1) Virtualization of servers and realization of optimum brokering

Existing business systems are designed for a particular business operation service. Therefore, the performance and properties of servers differ depending on the specific aims of systems. The physical differences must be hidden to use servers accommodated from other systems in the same way as servers in the original system. However, simply hiding such differences is often inadequate because of the different requirements (e.g., CPU performance, memory size, OS) of servers needed to run business applications according to the business operation services involved. This is why a brokering (intermediate) function is required to search for physical servers that satisfy specific requirements in the Grid system and use them as virtualized (logical) servers.

2) Autonomous server sharing

When a Grid system is used for scientific and engineering calculations, the servers are usually reserved for exclusive use before starting calculations. However, such a method cannot accommodate fluctuations in the workload of business operations. Moreover, while a fair allocation of servers is emphasized when using a Grid system for scientific and engineering calculations, in general business applications, servers should be allocated to important business operation services on a preferential basis. This requires a function that autonomously adjusts the server allocation according to the importance of each business operation service.

3) Simplification of porting existing applications to Grid environment

A unified approach is needed to transform applications that are running on a dedicated system into business operation services running on GSP without making major changes to their source code.

3. Functional overview of GSP in addressing problems

The problems described above were resolved as follows:

1) Virtualization of servers and realization of optimum brokering

We developed Physical Resource Broker (PRB), which virtualizes physical servers and brokers logical servers. PRB receives heartbeats from the node monitors set up in each physical server and maintains a list of alive servers. It also conducts a search of the database, which exists in the Grid information subsystem, to obtain a list of physical servers that meet the server requirements of a particular business operation. PRB crosschecks both lists and selects appropriate physical servers for a specific business operation. It considers these selected physical servers as being virtually equivalent and brokers them as logical servers.

Next, we developed Lifecycle Manager (LM) for requesting and returning additional logical servers according to the workload. A single LM supports one business operation service. We also developed Arbitrator (ARB) for receiving requests from all LMs and adjusting the allocation of logical servers. PRB virtualizes physical servers and brokers them as logical servers to ARB, while ARB arbitrates the requests from LMs and determines the allocation of logical servers.

2) Autonomous server sharing

We focused on the importance of business operation services to realize a function that autonomously optimizes the effectiveness of system operations. ARB optimizes the effectiveness by allocating servers to important business operation services on a priority basis. In other words, ARB diverts servers from low-priority business operation services and allocates them to high-priority business operation services that are requesting servers. On the other hand, a policy to dynamically change the priority can be set on LM to minimize the loss incurred by server deprivation. LM always notifies the latest priorities to ARB. For example, greater loss occurs when aborting a job being run for a long time in batch processing services. Therefore, we set a policy for increasing the priority of a running job in incremental steps according to the elapsed execution time and resetting to the default priority once the job is completed. This policy can raise a job's priority so that a server allocated to the job is not diverted to other services as its elapsed execution time increases. As a result, the loss caused by such a server deprivation can be minimized.

3) Simplification of porting existing applications to Grid environment

We designed a framework that transforms existing business applications into business operation services on GSP. With this framework, any business application that is adaptable to dynamic server allocation can be adapted to GSP through the procedure described below.

- Step 1: Preparation of LM

Write a Java program of about 100 lines using the library developed by the authors to request and return logical servers according to the workload.

- Step 2: Preparation of Application Wrapper (AW)

Write a Java program of about several 10s of lines using the library developed by the authors to control the start and stop of the business application on a physical server.

- Step 3: Preparation of a provisioning image

Create a provisioning image in which the business applications are installed.

- Step 4: Preparation of a service description file

Describe the requirements for the physical servers needed to run the business application and the policy parameters to determine the priority of the business operation service. Multiple service description files can be prepared for an implementation of LM. In this case, the behavior of LM depends on the specified service description file.

4. Design of GSP

Figure 1 shows the architecture of GSP, which was designed by reflecting the solutions described above. GSP realizes business operation services in the service layer by using the physical servers in the resource layer. GSP consists of the following four subsystems:

1) Grid service subsystem

This subsystem adapts existing business applications to the Grid environment in order to realize business operation services on GSP. It also requests and returns logical servers according to the workload.

2) Resource brokering subsystem

This subsystem receives the requests and returns of logical servers from the Grid service subsystem and determines the server allocation by considering the priority of business operation services.

3) Grid information subsystem

This subsystem collects information about the properties of each physical server and the status of each business operation service. Additionally, it compiles the collected information into a database and provides this information to the other subsystems.

4) Administration subsystem

This subsystem provides administrative functions to operate GSP, for example, monitoring of the priority of business operation services and setting of accounting information.

5. LM for adapting existing applications to GSP

Most existing business applications were written to use physical servers that were already configured and were not designed to use the GSP function that dynamically allocates logical servers. One major feature of GSP is that an existing business application can be ported as a business operation service on GSP without needing major changes to its source code. The mechanism of LM that provides a function for adapting existing business applications to GSP is described below.

LM requests ARB to add or return logical servers according to the workload fluctuations of the business operation services. The requirements for physical servers and the priority of service depend on the properties of each business operation service. Therefore, LM consists of a common part that is common to all business operation services and a specific part that can be customized to each business operation service. Each function for controlling a business operation service — including processes for measuring the workload based on a service-specific criteria and determining whether logical servers should be added or returned — must be implemented in the specific part in order to adapt an existing business application to GSP.

The common part provides the following functions:

1) Interfaces for requesting ARB to add or return logical servers.
2) A function for calling the procedure defined in the specific part periodically and at the initialization/finalization of the business operation services.
3) A function for receiving events concerning

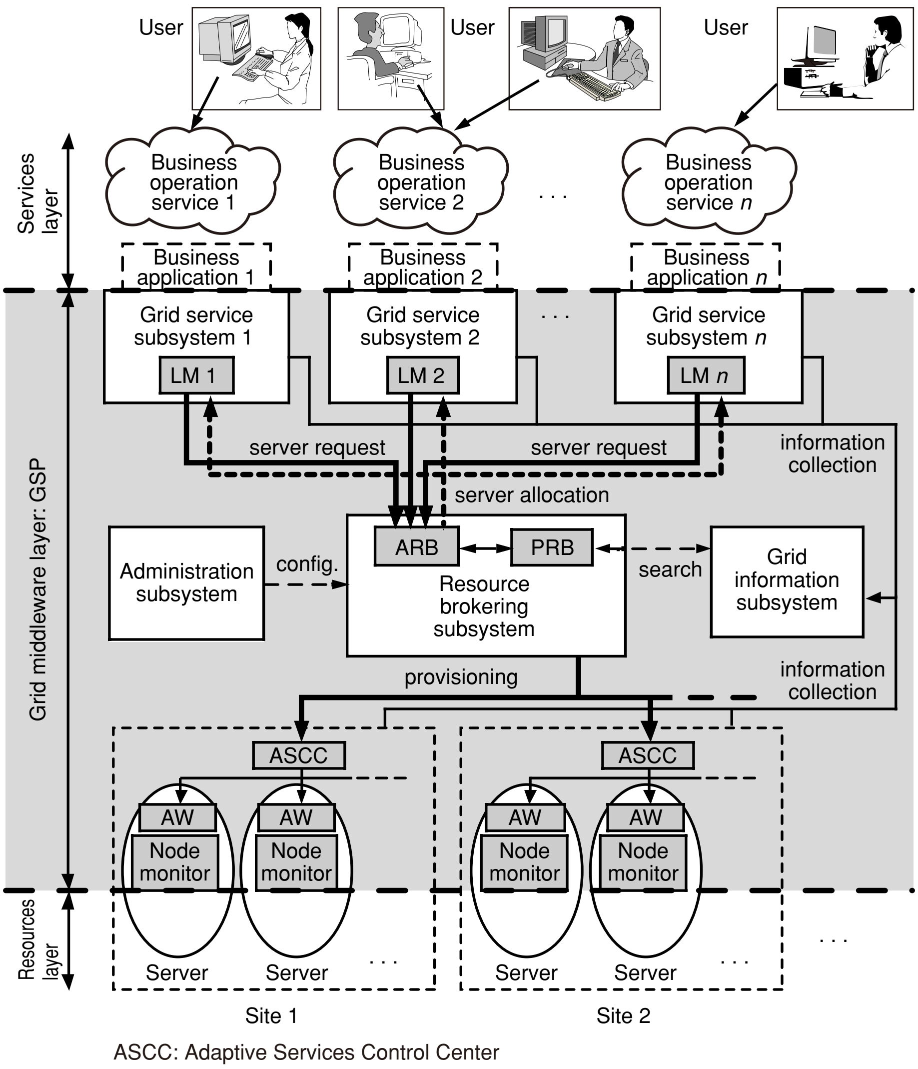


Figure 1
Architecture of GSP.

the start/stop of the business applications on logical servers from an AW. The received events can be dispatched to appropriate procedures defined in the specific part.

In the specific part, functions for controlling the business operation services are implemented using functions provided in the common part. For example, the common part provides a function for catching events notified from an AW. With these notifications, a function can be implemented that passes command arguments and environmental variables to be set for the business application to an AW when it notifies an event concerning a business application start. Moreover, a function can be implemented that adds logical servers to business operation services after confirming the success of a start and initialization of business applications on the additional servers allocated by ARB.

Existing business applications can therefore be adapted to GSP without major changes to their source code thanks to LM dynamically adding and returning logical servers in cooperation with ARB and AW.

6. Demonstration and evaluation

We realized an online-processed video conferencing service and a data analysis service on GSP. The former service was implemented using e-Conf,[3] which is a product of France Télécom. The latter service was implemented using CyberGRIP,[4),5)] which was developed by Fujitsu Laboratories as a Grid middleware for batch processing.

Figure 2 shows the testbed used for evaluation. **Table 1** lists the server models and the number of servers at each site. This distributed heterogeneous environment consisted of various machines, including servers made by Fujitsu and IBM. GSP switches business applications on servers by using the function of ASCC[6] manufactured by Fujitsu Siemens Computers (FSC).

We conducted an experiment, in which a day was shortened to just two hours, as follows. First, jobs were input into the data analysis service and the number of requests for connection to the video conferencing service was gradually increased. Then, the number of requests was held at a maximum level for about 10 minutes and then gradually reduced to zero. The experiment simulated a situation where the number

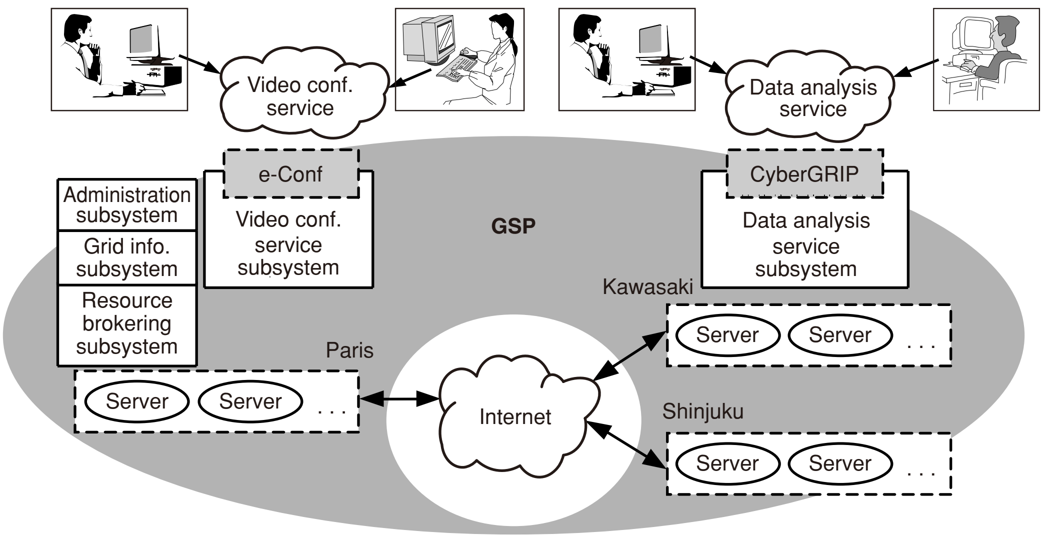


Figure 2
Testbed for evaluation.

of video conferencing users increases during the morning and decreases from the afternoon to late at night. A higher priority was set for the online-processed video conferencing service.

Figure 3 shows the change in server allocation over time for the data analysis service (gray area) and video conferencing service (black area). In the dedicated system, server allocation did not change because the business operation services could not share servers [Figure 3 (a)]. In GSP, on the other hand, the servers for the data analysis service having a lower priority were diverted and allocated to the video conferencing service in response to the increase in session requests for video conferencing. Conversely, the servers returned from the video conferencing service were reallocated to the data analysis service when the number of sessions for the video conferencing service decreased [Figure 3 (b)]. In this way, autonomous server sharing was realized among business operation services.

Figure 4 shows the change in the number

Table 1
Server models and numbers at each site.

Company name and site	Server model	No. of servers
FUJITSU Laboratories Ltd.		
Kawasaki	FUJITSU PRIMERGY BX600	10
France Télécom R&D		
Paris	FUJITSU PRIMERGY RX200S Ⅱ	3
	IBM xSeries 336	5
Shinjuku	FUJITSU PRIMERGY BX620	5

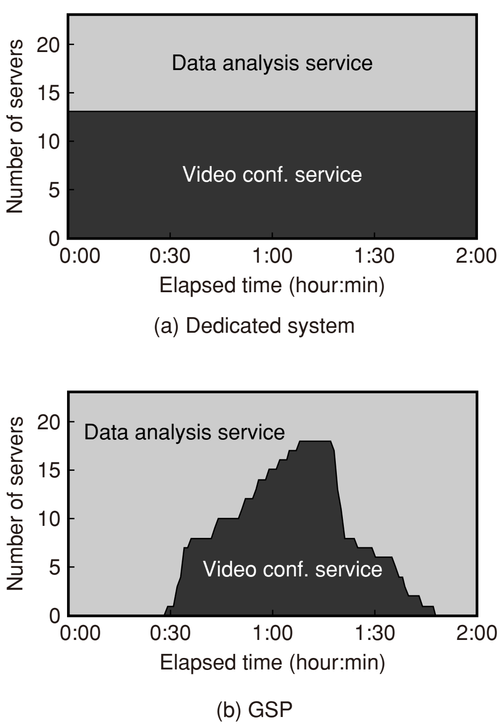


Figure 3
Temporal changes of server allocation.

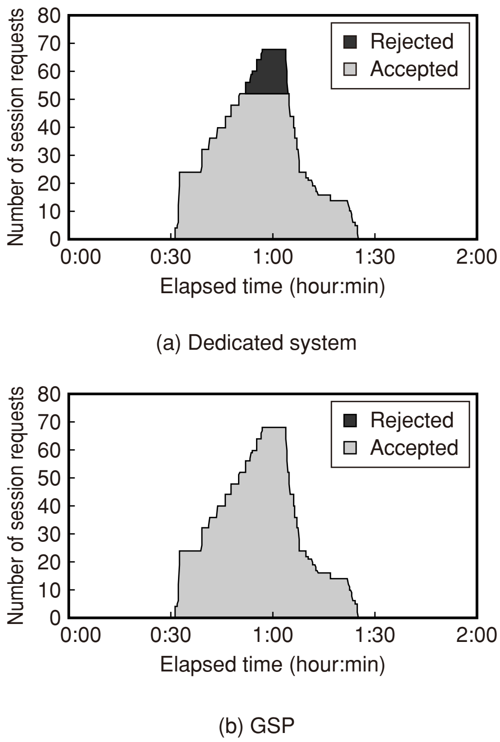


Figure 4
Temporal changes of number of sessions and status.

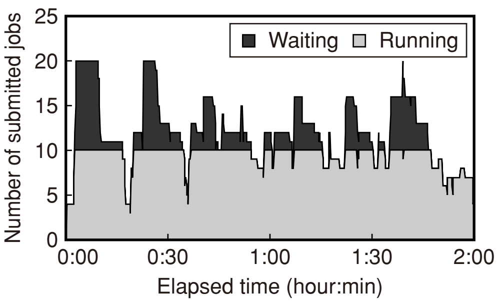
of requests for a video conferencing session and the request status. The gray area indicates the session requests that were accepted, and the black area indicates the session requests that were rejected due to an insufficient number of servers. In the dedicated system [Figure 4 (a)], some of the requests were rejected. However, in GSP [Figure 4 (b)], all the requests were accepted because the number of servers increased in line with the number of session requests.

Figure 5 shows the change in the number of jobs of the data analysis service and the job status. The gray area indicates running jobs, and the black area indicates jobs that are waiting due to an insufficient number of servers. The dedicated system had a large number of waiting jobs [Figure 5 (a)], which shows that not all of the jobs were completed in the system during the two-hour experiment. In contrast, the jobs ran in GSP for most of the time [Figure 5 (b)] because the servers for video conferencing were switched over to the data analysis service. As a result, all jobs were completed on GSP during the experiment.

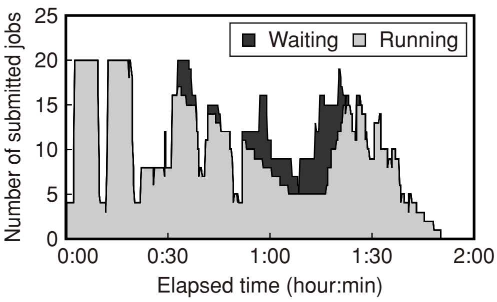
Table 2 compares the results of the dedicated system and GSP. This experiment confirmed that the number of session requests accepted for video conferencing increased by 30% and the processing time of the data analysis service was simultaneously reduced by 12% due to the sharing of servers on GSP.

7. Conclusion

This article described a Grid middleware called GSP that realizes effective server sharing among multiple business operation services. In a demonstration study of GSP using actual business applications, we simultaneously increased the capacity of a video conferencing service and the performance of a data analysis service. France Télécom and Fujitsu conducted cooperative research on GSP. We wish to express our deep appreciation to the researchers involved at France Télécom.



(a) Dedicated system



(b) GSP

Figure 5
Temporal changes of number of jobs and status.

Table 2
Comparison between dedicated system and GSP.

Evaluation items	Dedicated system	GSP
Video conferencing service		
Number of allocated servers	13 (fixed)	0 - 18
Number of session requests	68	68
Number of accepted sessions	52	68
Number of rejected sessions	16	0
Improvement ratio	—	30%
Data analysis service		
Number of allocated servers	10 (fixed)	5 - 23
Number of submitted jobs	128	128
Execution time	126 min.	111 min.
Reduced ratio of exec. time	—	12%

References

1) H. Ueda et al.: Grid Service Platform: A Prototype to Share Computational Resources for Business Operation Services. Proc. the Conference on Computational Engineering and Science, 11, 2, 2006, p.531-534.
2) S. Shigeta et al.: Grid Service Platform: Design and Implementation of a Grid Middleware for Telecom Carriers. Proc. the 2nd IEEE Intl. Conf. on e-Science and Grid Computing, 2006.
3) e-Conf. *http://www.francetelecom.com/en/group/rd/offer/brevets/e-conf/presentation.html*
4) H. Kobashi et al.: Integration of dispersed computational resources using Grid middleware "CyberGRIP". Proc. the 7th Problem Solving Environment Workshop & the 2nd Grid Seminar, 2004, p.63-68.
5) K. Miyazawa et al.: Development of Grid Middleware CyberGRIP and its Applications. Proc. the 1st IEEE Intl. Conf. on e-Science and Grid Computing, 2005, p.486-493.
6) Adaptive Services Control Center (ASCC). *http://www.fujitsu-siemens.com/it_trends/dynamic_data_center/ascc.htm*

Soichi Shigeta, *Fujitsu Laboratories Ltd.*
Dr. Shigeta received the B.S, M.S., and Ph.D. degrees in Engineering from the University of Electro-Communications, Tokyo, Japan in 1995, 1997, and 2002, respectively. From 2002 to 2004, he was an assistant professor in the Graduated School of Information Systems, the University of Electro-Communications. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 2004, where he has been engaged in research and development of distributed computing systems such as Grid computing. He is a member of the Information Processing Society of Japan (IPSJ) and the Institute of Electronics, Information and Communication Engineering (IEICE) of Japan.

Haruyasu Ueda, *Fujitsu Laboratories Ltd.*
Mr. Ueda received the B.S. and M.S. degrees in Information Science from Tokyo Institute of Technology, Tokyo, Japan in 1987 and 1989, respectively. He joined the International Institute of Advanced Study of Social Information Science, Fujitsu Ltd., Kamata, Japan in 1989, where he has been engaged in research and development of groupware, combinatorial optimization and distributed computing. He is a member of the Information Processing Society of Japan (IPSJ) and the Japanese Society for Artificial Intelligence (JSAI). He received the JSAI Annual Conference Award from the JSAI in 1998.

Nobutaka Imamura, *Fujitsu Laboratories Ltd.*
Mr. Imamura received the B.S. degree in Electronics Engineering in 1990 and the M.S. degree in Information Systems in 1992 from Kyushu University, Fukuoka, Japan, respectively. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1992, where he has been engaged in research and development of parallel computing systems. He is a member of the Information Processing Society of Japan (IPSJ).