Flexible I/O Improves Flexibility and Reliability of Mission-Critical IA Server PRIMEQUEST

● Ohsai Hamada

(Manuscript received May 20, 2005)

The PRIMEQUEST mission-critical IA servers consist of a system board, which mounts the CPU and memory, and an I/O unit (IOU), which mounts I/O-related blocks such as hard disk drives and PCI slots. The system boards and IOUs are physically separate from each other and are interconnected by a crossbar. This feature is referred to as the Flexible I/O because it enables partitions to be configured from system boards and IOUs in any combination. This paper gives an overview of the Flexible I/O and discusses some of the benefits it provides.

1. Introduction

The PRIMEQUEST series of mission-critical IA servers provide various functions from the chipset level up to the unit and system levels to achieve the high reliability and flexibility required for mission-critical applications.

This paper describes the Flexible I/O, which is one of the features of PRIMEQUEST. First, it outlines the Flexible I/O and then gives examples of some of the benefits the Flexible I/O can provide in the PRIMEQUEST 480 model. Except for the number of system boards (SBs) and I/O units (IOUs) that can be installed in a cabinet, these examples also apply to the PRIMEQUEST 440 model.

2. Outline of Flexible I/O

As described in the paper, "High-Reliability Technology of Mission-Critical IA Server PRIMEQUEST," which appears elsewhere in this special issue, PRIMEQUEST uses a crossbar to configure flexible partitions from SBs, which contain the CPUs and memories, and IOUs, which contain the hard disk drives and PCI slots.

The PRIMEQUEST 480 model is configured from up to eight independent SBs (SB#0 to SB#7) and up to eight IOUs (IOU#0 to IOU#7). The SBs and IOUs are physically separate units.

The crossbar provides the Flexible I/O. A partition can be configured with any number of SBs chosen from SB#0 to SB#7 and any number of IOUs chosen from IOU#0 to IOU#7.

Figure 1 shows some partition configuration examples. Partition A has the same number of SBs and IOUs, whereas partitions B and C have different numbers of SBs and IOUs. The Flexible I/O makes it possible to define flexible partitions

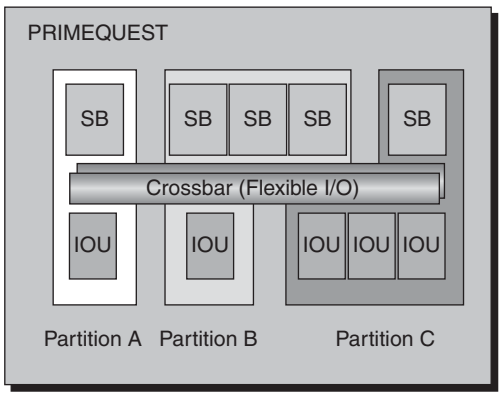

Figure 1
Partition configurations of PRIMEQUEST 400 series.

such as in these examples.

As shown in **Figure 2**, in a conventional server, the CPU, memory, and I/O interface are configured together as a single physical unit. However, this type of configuration causes the following problem when configuring partitions.

Because the CPU, memory, and I/O interfaces are configured as a single unit, for a partition that requires large amounts of CPU and memory resources but does not require a large number of I/O interfaces, unnecessary I/O interfaces will be allocated (partition A on the left in Figure 2). Conversely, for a partition that does require a large number of I/O interfaces, unnecessary CPU and memory resources will be allocated (partition B on the left side in Figure 2). This type of configuration therefore wastes precious resources and is more expensive than necessary because of the unnecessary allocation of resources.

In contrast to this, PRIMEQUEST is provided with the Flexible I/O so that 1) multiple SBs can be allocated for partitions that require a large amount of CPU and memory resources or 2) multiple IOUs can be allocated for partitions that require a large amount of I/O resources (right side in Figure 2).

Using the Flexible I/O therefore enables effective use of hardware resources (cost optimization) and minimizes their wasteful use, which in turn enables the optimum partitions to be configured.

3. Benefits of Flexible I/O

The Flexible I/O enables the construction of flexible partition configurations and also enables a variety of floating SB operations. In this paper, SBs that are not included in partition configurations are referred to as floating SBs.

The following are some of the benefits that floating SBs provide.

1) Rapid reconfiguration at SB failure
2) Partition reconfiguration at workload change
3) Fault prediction and partition reconfiguration

This section discusses each of these benefits.

3.1 Rapid reconfiguration at SB failure

If an SB fails, PRIMEQUEST automatically removes the faulty SB and replaces it with one that has been set as a floating SB and reconfigures the partition. If there is no floating SB, the partition to which the faulty SB belongs is stopped

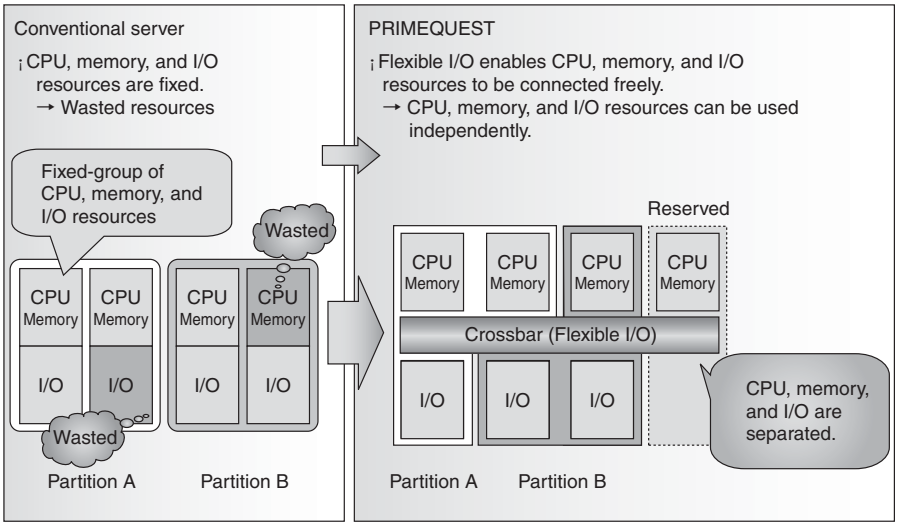

Figure 2
Partition configurations of PRIMEQUEST and conventional servers.

and rebooted. During the reboot, the faulty SB is removed from the partition so tasks can continue to be executed. However, performance will be degraded because of the reduced number of SBs.

When a floating SB has been set, it is automatically assigned when the partition is reconfigured and the partition rebooted. This ensures that the number of SBs in the partition is maintained. That is, any degradation in performance can be prevented because the number of CPUs and memory capacity are maintained.

Figure 3 shows an example of rapid reconfiguration at SB failure. Partition A is configured with SB#0 and SB#1 and IOU#0 and IOU#1. Partition B is configured with SB#2, SB#3, IOU#2, and IOU#3, and SB#4 is set as the floating SB. If SB#3 fails, partition B is stopped and reconfigured by the firmware of the server management unit (MMB). During this reconfiguration, the faulty SB#3 is removed from partition B and replaced with floating SB#4 to reconfigure partition B. This process is implemented by the MMB firmware. Therefore, even if SB#3 fails, the CPU and memory resources that configure partition B can be maintained. Preparing a floating SB in advance enables a faulty SB to be quickly removed from the partition and the floating SB incorporated. As a result, the system downtime can be kept to a minimum.

3.2 Partition reconfiguration at workload change

If the workload of a task being executed in a partition becomes too high, the floating SB can be incorporated into the partition to handle the increased workload.

Figure 4 shows an example of partition reconfiguration at workload change. Partition A is configured with SB#0, SB#1, IOU#0, and IOU#1. Partition B is configured with SB#2, SB#3, IOU#2, and IOU#3, and SB#4 is set as the floating SB.

As an example, assume that the workload of partition B becomes too high. At this time, the incorporation of floating SB#4 into partition B increases the amount of CPU and memory resources available to partition B. As a result, partition B can handle the increased workload.

The Flexible I/O therefore makes it possible to handle workload changes.

3.3 Fault prediction and partition reconfiguration

Hardware failures that occur on servers can be roughly divided into two types: correctable and uncorrectable. Two examples of the first type are a single-bit error in data transferred over a bus and a single-bit error in memory data. The hardware automatically corrects this type of error using the error correcting code (ECC) circuits installed in the hardware, and this type does not

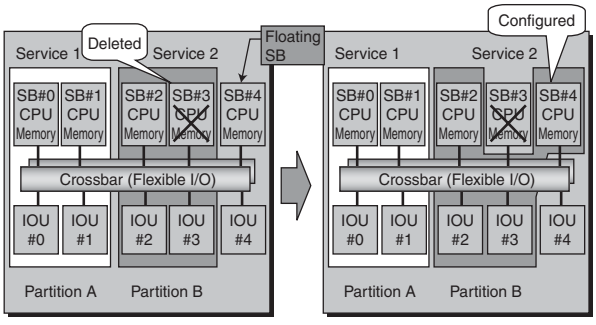

Figure 3
Rapid reconfiguration at SB failure.

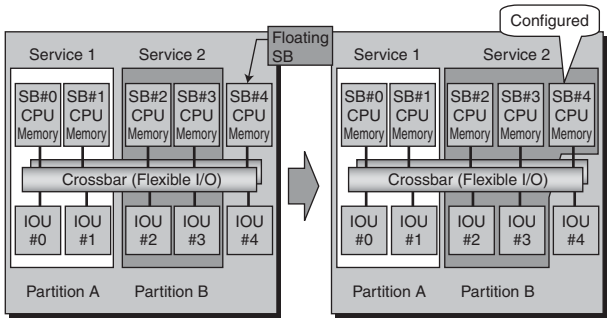

Figure 4
Partition reconfiguration at workload change.

affect the execution of applications. Uncorrectable errors can be detected by the ECC circuits but the hardware cannot correct them. An example of an uncorrectable error is a multi-bit error in the data. When a multi-bit error occurs, the hardware notifies the OS that an uncorrectable error has occurred, and the OS then performs an emergency stop to prevent data from being corrupted.

Because correctable errors do not affect applications, the tasks are not stopped. Usually, there is no problem if the frequency of correctable errors remains below a specific level. However, if the frequency of single-bit errors increases, the frequency of multi-bit errors also increases. Therefore, if the frequency of correctable errors such as single-bit errors in the data exceeds a specific level, the frequency of uncorrectable multi-bit errors can be predicted to increase. The processing required to make such a prediction is referred to as fault prediction.

System availability can therefore be improved by combining fault prediction and the floating SB.

Figure 5 shows an example of fault prediction and partition reconfiguration. Partition A is configured with SB#0, SB#1, IOU#0, and IOU#1. Partition B is configured with SB#2, SB#3, IOU#2, and IOU#3, and SB#4 is set as the floating SB.

If single-bit errors frequently occur in the data for SB#2 of partition B, the fault prediction function judges that single-bit errors will become multi-bit errors with high probability. Then, before the single-bit errors in the data become multi-bit errors, SB#2 is removed from the configuration of partition B and replaced with the floating SB#4. As a result, multi-bit errors and failure of the partition are prevented.

In this section, we described the three benefits of using a floating SB. In each case, the Flexible I/O enables an arbitrary SB to be set as a floating SB for flexible operation.

4. Conclusion

This paper described the Flexible I/O, which is one of the features of PRIMEQUEST. It discussed the Flexible I/O's ability to configure highly flexible partitions using the minimum number of hardware resources required for a task and the use of a floating SB to minimize the amount of downtime. Use of the Flexible I/O and floating SB reduces the total cost of ownership (TCO).

It is imperative that the information systems of the Internet age are reliable enough to operate 24 hours a day 365 days a year and flexible enough to handle rapid fluctuations in workload. The Flexible I/O is a key feature for realizing highly flexible systems. We therefore plan to further improve and expand its functions to provide our customers with systems that are easier to operate and manage.

This research has been partially funded by the Ministry of Economy, Trade and Industry (METI) and the New Energy and Industrial Technology Development Organization (NEDO).

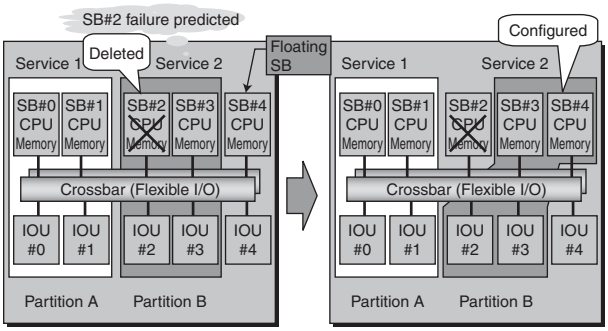

Figure 5
Fault prediction and partition reconfiguration.

Ohsai Hamada received the B.S. and M.S. degrees in Electronics Engineering from Hokkaido University, Sapporo, Japan in 1981 and 1983, respectively. He joined Fujitsu Ltd., Kawasaki, Japan in 1983, where he has been developing server systems, including mainframes and IA servers. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and the IEEE.