RIKEN Super Combined Cluster (RSCC) System

● Kouichi Kumon ● Toshiyuki Kimura ● Kohichiro Hotta
● Takayuki Hoshiya

(Manuscript received June 14, 2004)

Recently, Linux cluster systems have been replacing conventional vector processors in the area of high-performance computing. A typical Linux cluster system consists of high-performance commodity CPUs such as the Intel Xeon processor and commodity networks such as Gigabit Ethernet and Myrinet. Therefore, Linux cluster systems can directly benefit from the drastic performance improvement of these commodity components. It seems easy to build a large-scale Linux cluster by connecting these components, and the theoretical peak performance may be high. However, gaining stable operation and the expected performance from a large-scale cluster needs dedicated technologies from the hardware level to the software level. In February 2004, Fujitsu shipped one of the world's largest clusters to a customer. This system is currently the fastest Linux cluster in Japan and the seventh fastest supercomputer in the world. In this paper, we describe the technologies for realizing such a high-performance cluster system and then describe the implementation of an example large-scale cluster and its performance.

1. Introduction

A Linux cluster (sometimes called a PC cluster) consists of Intel Architecture (IA) based servers, interconnect networks, and cluster middleware. Formerly, the PC cluster system was regarded as a low-budget supercomputer because it uses inexpensive PCs as computing engines and Ethernet for interconnections. Many PC boards now have Ethernet built in, so such a PC cluster can be built using just PCs and Ethernet hubs. Because of improvements in the performance of IA CPUs, supercomputers can now be replaced with PC clusters by adding extra functionality or small amounts of non-commodity components. Therefore, we use the term "Linux Cluster" to distinguish such a supercomputer from a PC cluster system.

The major components of a Linux cluster system are the IA servers and interconnect networks, and the number of CPUs can vary from several to more than a thousand, depending on the performance requirements. For a large-scale cluster, the network's performance becomes more important because, if the network's performance is poor, the scalability will be limited regardless of the number of nodes. Therefore, the network performance becomes a key issue in a large-scale system. Improving the network performance requires not only hardware enhancement, but also optimization of the driver software and network middleware. In addition to the performance issue, system management causes another problem. That is, when a Linux cluster system is used by a relatively small, experienced group or a single person, the system's stability is not a prime issue because stability is maintained by the users' programming skills. For example, a well-written program periodically dumps its internal state (checkpointing) so it can be restarted from the latest dump (recovered) when the system crashes. However, as the application domain is extended, clusters are being used as the central machines in laboratories and even companies. As a result, cluster systems must be able to perform checkpointing and recovery without the help of application programs. Therefore, the system software must strengthen the system stability as well as the performance.

SCore1), note 1) cluster middleware is a system software for solving these problems. It provides both high-performance communication interfaces and a checkpoint-restart functionnote 2) that can augment the scalability and reliability of a large-scale system without application modification. In February 2004, Fujitsu shipped a large-scale Linux cluster system to a customer. This system—the RIKEN Super Combined Cluster (RSCC)—is the fastest Linux cluster in Japan and the seventh fastest supercomputer in the world. It is currently operating as the main computer at the customer's supercomputer center. In this paper, we describe technologies for realizing a high-performance Linux cluster. Some examples of these technologies are 1) high-performance interconnection networks, 2) programming support technology to extend the cluster usage so users can migrate their code from a vector computer environment to a cluster environment, and 3) management features to gain system reliability. We then introduce the RSCC system as an example of our Linux cluster technologies and describe its performance.

2. Issues of Linux cluster systems for high-performance computing facility

To realize a high-performance, large-scale Linux cluster and widen the application area, at least three goals must be reached: 1) a high-performance interconnect network to gain the best application performance, 2) language support to widen the application area, and 3) management support for a large-scale cluster to realize stable operation. In the following sections, we describe the technologies we have developed to reach these goals.

2.1 InfiniBand: A high-performance cluster interconnect

On a Linux cluster system, a large-scale computation is divided into small blocks of tasks, the blocks are distributed to the nodes of a cluster, and then the nodes execute the tasks by communicating with other nodes through the interconnecting network. If application programs need frequent communication to execute, the overall system performance will mainly be determined by the network performance. Therefore, a high-performance interconnection network is needed to build a high-performance cluster.

Two common measures are used to evaluate the interconnect network: the network throughput and the roundtrip latency time (RTT). For a cluster interconnect, Gigabit Ethernet and Myrinet are typically used. Gigabit Ethernet is cheap and commonly used in systems ranging from a desktop computer to a large server system. Its theoretical maximum throughput is 1 Gbps (125 MB/s), and the roundtrip latency, which depends on the network interface controller (NIC) and the communication protocol, is usually from 30 to 100 µs. Myricom's Myrinet interconnect is widely used as a high-performance interconnect and has a 2 Gbps link speed and a roundtrip latency of less than 10 µs. As the cluster scale increases, a much higher performance interconnect is needed. To meet this need, we developed

note 1) SCore was developed by the Real World Computing Project, which was funded by the Japanese Government. It is currently supported by the Linux Cluster Consortium.

note 2) A checkpoint restart function dumps the status of a running process so it can be used to restore the status at a later time. It is mainly used to recover the status after a system has crashed.

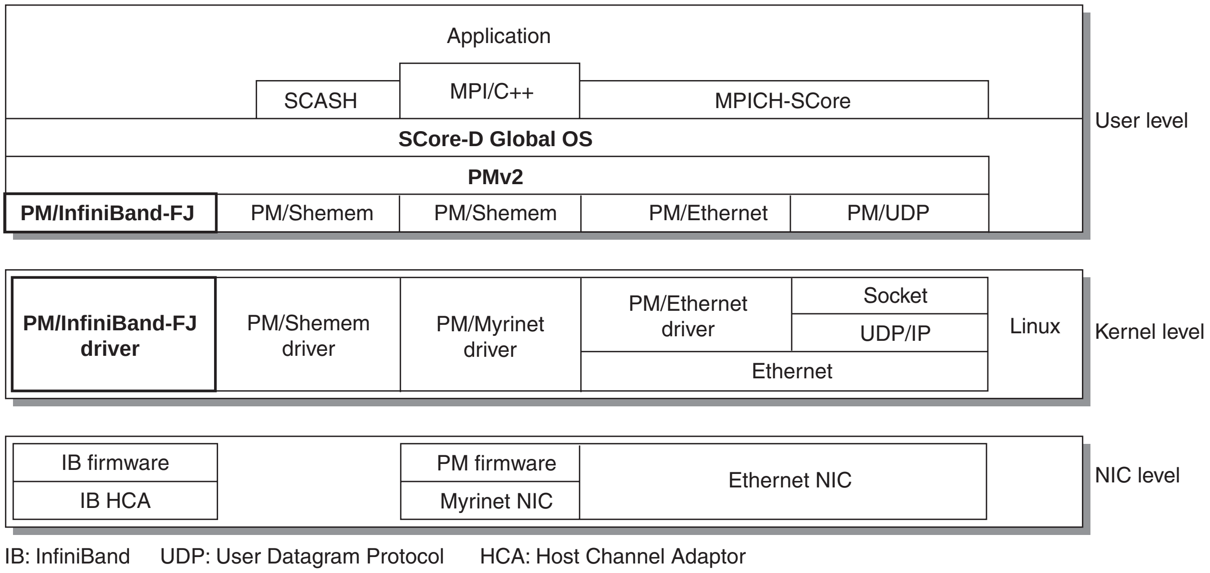


Figure 1
SCore cluster middleware architecture.

the InfiniBand interconnect.

The InfiniBand is a general-purpose interconnect network specified by the InfiniBand Trade Association (IBTA). This specification defines three data transfer speeds: 2 Gbps, 8 Gbps, and 24 Gbps—called the 1x, 4x, and 12x link, respectively. To gain a superior performance, the 4x and 12x InfiniBand links are candidates. Fujitsu already ships a 4x InfiniBand Host Channel Adaptor (HCA) for its PRIMEPOWER server, so we decided to develop a high-performance driver for use with a Linux cluster system. To provide good network performance, the interface between a user program and the hardware is also important. We therefore chose the SCore cluster middleware to realize the required performance and stability. The main components of the SCore middleware are the PMv2[2] high-performance, low-level communication interface and the SCore-D[3] cluster operating system (**Figure 1**).

PMv2 provides a short latency and a high-throughput network infrastructure to the MPI (Message Passing Interface) library and abstracts the implementation of underlining hardware such as the Ethernet, Myrinet, shared memory, and SCI (Scalable Coherent Interface). Therefore, by adding the PM/InfiniBand-FJ driver, application programs can run on these different kinds of interconnect hardware without the need for modification or recompilation.

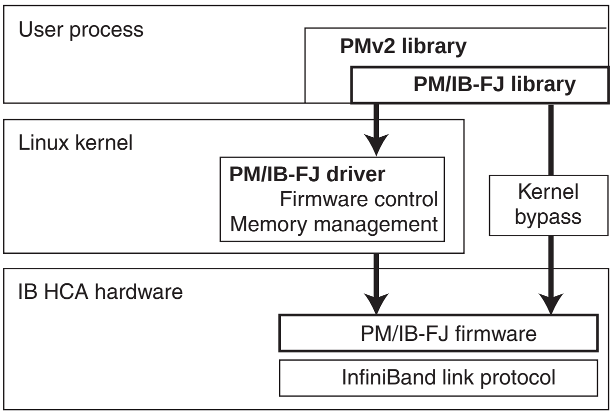


Figure 2
Architecture of PM/InfiniBand-FJ.

The PM/InfiniBand-FJ library directly communicates with the InfiniBand HCA; therefore, the kernel overhead is not added to the communication overhead (**Figure 2**). Conventional networks need a driver to prevent malicious access to the hardware and also prevent interference between user processes. InfiniBand HCA has a hardware mechanism to implement this protec-

Table 1
PM-level roundtrip latency.

	RTT (µs)	Ratio
PM/InfiniBand-FJ	16.8	2.02
PM/MyrinetXP	8.3	1
PM/Ethernet	29.6	3.37

tion; therefore, the user-level library can safely access the hardware directly.

Table 1 and **Figure 3** show, respectively, the PM-level roundtrip latency and PM-level network throughput of the developed driver as measured on a Fujitsu PRIMERGY RX200 IA server.

The maximum Remote Direct Memory Access (RDMA) throughput via InfiniBand is 820.9 MB/s, which is 82% of the theoretical maximum speed. The roundtrip latency of PM/InfiniBand-FJ is relatively longer than that of PM/MyrinetXP, and this difference is mainly due to the protocol processing complexity on the IB HCA. Note that the maximum throughput depends on the chipset located between the main memory and PCI-X on the server system. If the ServerWorks chipset is used, the maximum throughput becomes 917 MB/s or 91% of the theoretical maximum speed. Therefore, the driver effectively uses the full bandwidth of InfiniBand.

2.2 Language support for high-performance Linux clusters

To support programming for high-performance Linux clusters, Fujitsu provides the "Fujitsu FORTRAN & C Package for Linux." This package contains a FORTRAN compiler and a C compiler and also includes frequently used mathematical libraries. Although the peak computational performance of Intel architecture (IA) CPUs becomes high as the CPU core frequency increases, extracting the maximum performance is not an easy task. The Xeon processor performs two floating-point operation per CPU cycle; therefore, a single 3.06 GHz Xeon can execute up to 6.12 G floating-point operations per second (FLOPS) and a dual CPU can execute 12.24 GFLOPS.

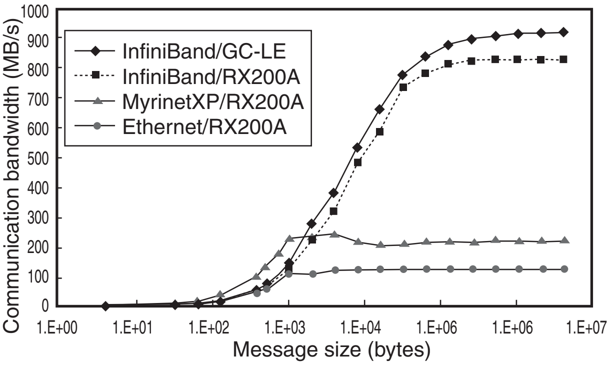
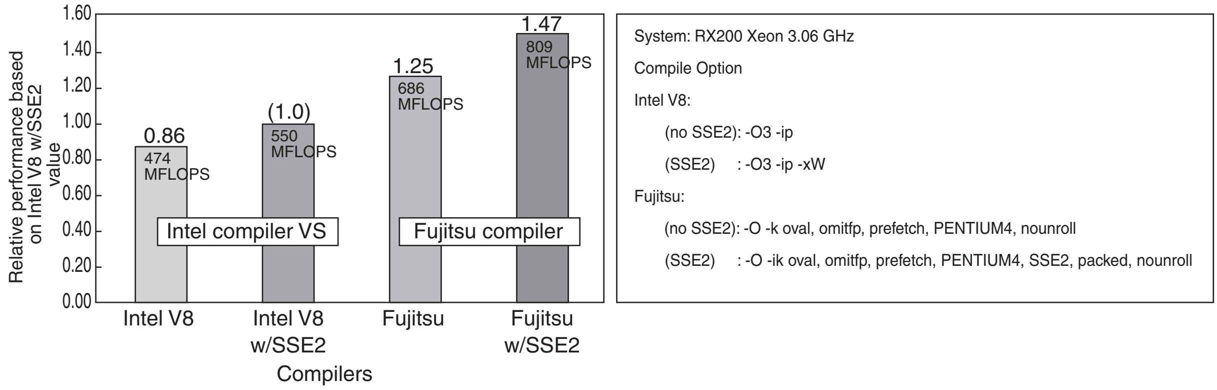


Figure 3
PM-level network throughput (RDMA-WRITE).

Memory access is not fast enough to keep up with the increased CPU performance: the access latency within the system is 150 ns or more, depending on memory traffic. Therefore, data prefetching must be performed to maintain the level of performance. The Xeon processor has built-in hardware for simultaneously prefetching eight data streams, but this stream count is not enough for many HPC applications. Therefore, instructions to explicitly prefetch data must be included to sustain the CPU performance. To solve this issue, Fujitsu developed a high-performance FORTRAN compiler in Fujitsu FORTRAN & C Package for Linux. The compiler is fully compliant with the ANSI FORTRAN 77/90 and FORTRAN95 standards and accepts some FORTRAN2000 extensions. The compiler can provide good compatibility with the compilers of Fujitsu's other computer systems, so it is easy to transport code to and share code on a Linux cluster system. The complier produces optimized object code suitable for running HPC applications on recent Intel architecture processors, especially Xeon processors. **Figure 4** shows the performance difference between FJ-FORTRAN and other major compilers.

Another tool is XP-FORTRAN. This tool is used for parallel programming on a Linux cluster and was originally a language extension designed for Fujitsu vector processors such as the VPP5000. Usually, a program running on a Linux cluster



System: RX200 Xeon 3.06 GHz
Compile Option
Intel V8:
(no SSE2): -O3 -ip
(SSE2) : -O3 -ip -xW
Fujitsu:
(no SSE2): -O -k oval, omitfp, prefetch, PENTIUM4, nounroll
(SSE2) : -O -ik oval, omitfp, prefetch, PENTIUM4, SSE2, packed, nounroll

Figure 4
Fujitsu FORTRAN compiler performance.

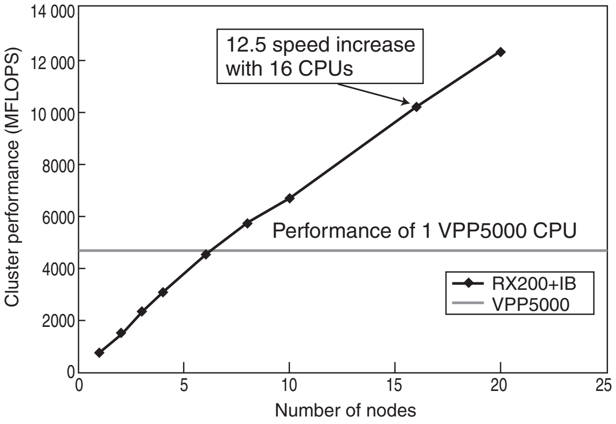


Figure 5
XP FORTRAN performance using HIMENO benchmark program.

uses a library that conforms to the MPI standard or XP-FORTRAN. By using XP-FORTRAN for Linux clusters, programs on a VPP system can easily be migrated to a cluster environment. Also, users familiar with XP-FORTRAN can easily write a new application for clusters. XP FORTRAN directly uses PMv2 RDMA API; therefore, a low overhead and asynchronous remote memory access can be achieved while the user's programs are running. **Figure 5** shows the performance of XP FORTRAN using the HIMENO Benchmark program.[note 3)]

This package also includes the high-performance SSL-II BLAS (Basic Linear Algebra Subprograms)/LAPACK library of frequently used mathematical routines.

DGEMM (Double precision GEneral Matrix-Matrix multiply) in BLAS, which is the core matrix multiplication routine, has been redesigned to maximize the performance on the Xeon Processor. The performance of this routine is superior to that of GOTO's BLAS,[4)] which has been regarded as the fastest DGEMM implementation (**Figure 6**).

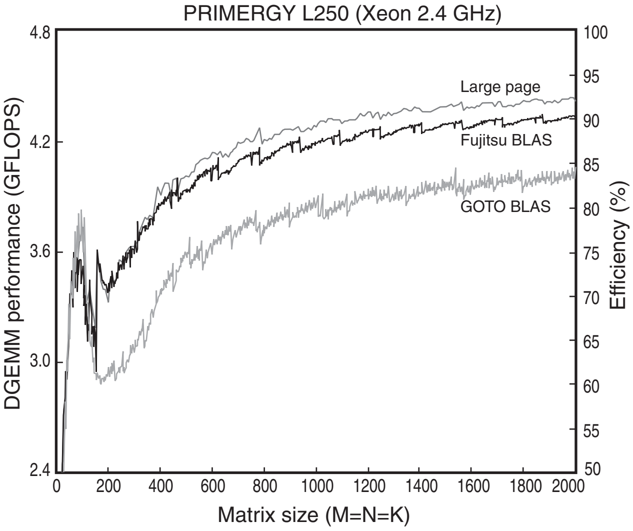
This performance improvement is achieved by tuning memory access by utilizing SSE and SSE2 SIMD instructions and by optimizing memory access using data prefetch instructions and instruction scheduling. The DGEMM routine is used in most of the level-two BLAS routines,[5)] and the optimization improves most of the BLAS level-two routines. Additionally, the DGEMM is heavily used in the Linpack benchmark program;[note 4)] therefore, Fujitsu's BLAS also accelerates the performance.

2.3 Cluster management tools to support large-scale clusters

The system management of clusters is a big issue for practical operation of large-scale systems.

note 3) The HIMENO benchmark program measures the processing speed of the main loop for solving Poisson's equation by using Jacobi iteration.

note 4) The Linpack benchmark is a program for solving a dense system of linear equations.[6)]



	Average performance		Maximum performance	
1CPU	GFLOPS	Efficiency	GFLOPS	Efficiency
GOTO BLAS	3.69	76.8%	4.06	84.5%
Fujitsu BLAS	4.04	84.2%	4.34	90.4%
Large Page	4.12	85.8%	4.43	92.4%

Large page: IA32 Linux system uses 4 KB as the page size. By using a 4 MB page, the new DGEMM shows more than 92% efficiency. Unfortunately, large pages are not usable in conventional Linux, so this is only a reference value.

Figure 6
DGEMM in BLAS performance comparison.

The main system management functions are:

- Job invocation and data transfer library functions. These are provided by SCore and MPICH.
- Batch job scheduling: SCore provides PBS (Portable Batch System) for this function.
- Monitoring of system hardware with sensors and/or hardware error detection.
- System deployment support.

SCore can provide the job invocation, data transfer, and batch job scheduling functions, and we additionally provide the NQS (Network Queueing System) as a batch job scheduler. NQS is used for Fujitsu's supercomputer systems and makes transitions from previous operations smooth. To satisfy the needs for center operation, we have added a checkpoint/restart management feature to NQS. We have also added a job freeze facility to the SCore-D operating system so the job operation policy can be changed for weekends: for example, to stop workday jobs on Friday night and restart them on Monday morning.

Cluster hardware management is one of the key issues in cluster management for stable operation, especially for a large-scale Linux cluster. Large-scale cluster systems (e.g., systems containing hundreds of IA servers) require tools for monitoring the hardware status and sensor data and also for checking the hardware error logs so that system malfunctions can be discovered before they cause system crashes. However, collecting the hardware status of hundreds of servers and controlling them usually degrades the system performance. To overcome this problem, we developed a remote management tool using the Intelligent Platform Management Interface (IPMI) over Ethernet. IPMI is a de facto standard specification defined by Intel and other server vendors. Fujitsu's new PRIMERGY server conforms to the IPMI specification standard version 1.5. This standard enables sensing and control of the server hardware through an onboard LAN interface without additional hardware. This tool directly communicates with a baseboard management controller (BMC) on the server main board using an on-board Ethernet interface. The BMC continuously monitors and controls the server, even when the main DC power is off. Therefore, the server status such as the board temperature, fan rotation, and hardware error logs stored inside the BMC can be collected anytime and also the server power-on/off and reset can be controlled if needed. The IPMI tool is designed to be applicable to a large-scale cluster and can execute

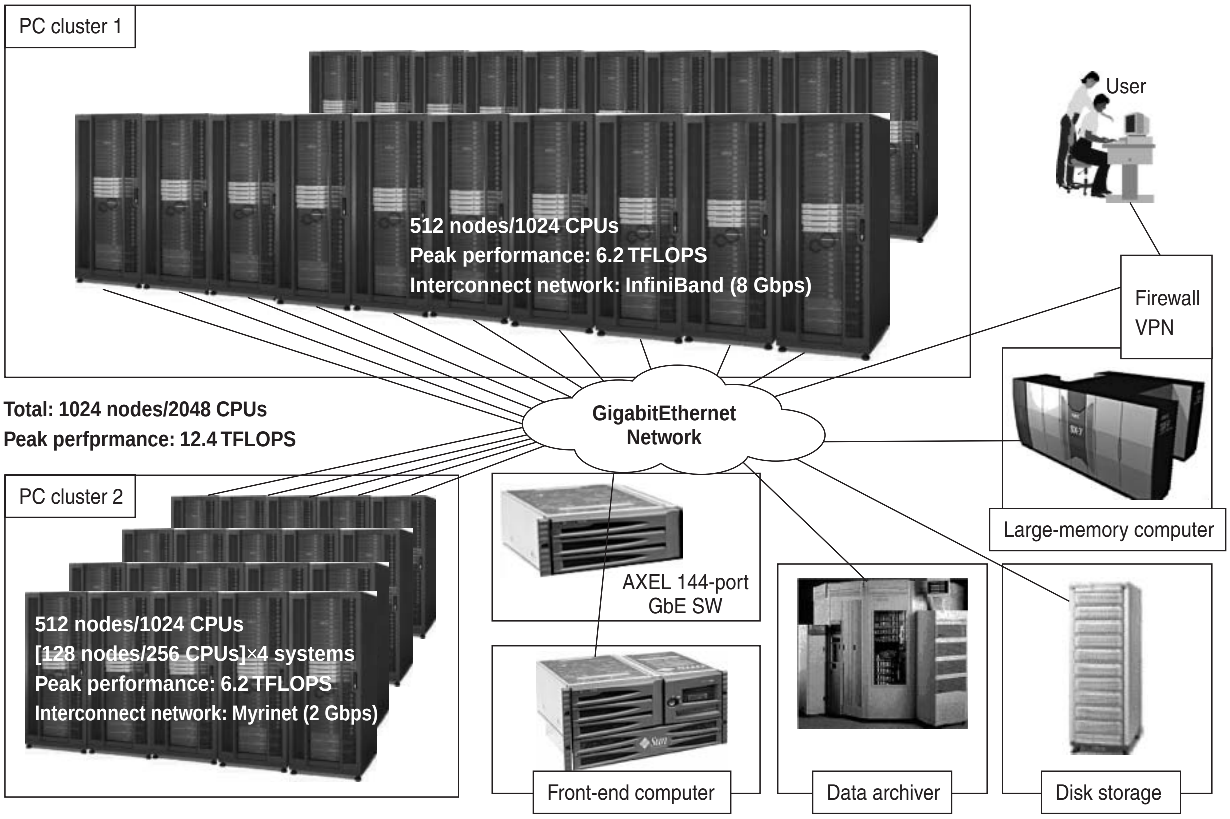

PC cluster 1
512 nodes/1024 CPUs
Peak performance: 6.2 TFLOPS
Interconnect network: InfiniBand (8 Gbps)
User
Firewall
VPN
Total: 1024 nodes/2048 CPUs
Peak perfprmance: 12.4 TFLOPS
GigabitEthernet
Network
PC cluster 2
512 nodes/1024 CPUs
[128 nodes/256 CPUs]×4 systems
Peak performance: 6.2 TFLOPS
Interconnect network: Myrinet (2 Gbps)
AXEL 144-port
GbE SW
Large-memory computer
Front-end computer
Data archiver
Disk storage

Figure 7
Overview of RSCC system.

most of its operations within a few seconds to at most a minute, even when the system contains thousands of servers. Collected server data such as the main-board temperature can be used to maintain the operation environment, for example, the room airflow. Also, non-fatal hardware error status data such as correctable memory errors can indicate the need to replace a unit before it fails. Therefore, this tool can enhance system reliability.

3. RIKEN Super Combined Cluster (RSCC) system

By applying the technologies described above, a large-scale cluster system can be used as a main machine in a computer center. On March 2004, Fujitsu shipped one of the largest Linux cluster systems in Japan to RIKEN[7), note 5)] as their main computer system. The system is called the RIKEN Super Combined Cluster (RSCC) (**Figure 7**).

3.1 RSCC configuration

The RSCC consists of five coarsely coupled clusters, the largest of which consists of 512 Fujitsu PRIMERGY RX200 dual 3.06 GHz Intel Xeon processor servers with a Fujitsu 4x InfiniBand host channel adaptor (HCA) on a PCI-X bus and 4 GB of main memory. To maximize the usage of the high-performance interconnection network, the PCI-X bus frequency of the RX200s has been enhanced from the original 100 MHz to 133 MHz by reducing the bus loads. The forty-eight 32-port InfiniBand 4x switches make up a full bisection-bandwidth multi-stage network.

Three of the other four clusters consist of 128 of the same RX200 dual-CPU servers, but

note 5) An Independent Administrative Institution under the Ministry of Education. It performs high-level experimental and research work from basic research to practical applications, including physics, chemistry, medical science, biology, and engineering.

the memory size in these servers is 2 GB and Myricom's Mryinet-XP is used for interconnection. These servers have a theoretical maximum bi-directional data transfer speed of 250 MB/s. The last cluster consists of 128 servers and uses InfiniBand as the interconnect.

All five clusters are connected by Gigabit Ethernet, each cluster has 16 nodes, and each node is connected to a Fujitsu SH4432G 24-port Gigabit Ethernet hub; therefore, a total of 1024 nodes are connected by 64 SH4432Gs. The two up-links of each hub terminate at root switches that consist of two CISCO 4507 Ethernet routers. The two routers are connected by eight aggregated Gigabit Ethernet lines. Usually, the five clusters are independently operated, but they can be used as a single 1024-node cluster when required. For this purpose, 144 high-speed PFU XG-144G Gigabit Ethernet switches are used to directly connect all 64 SH4432G hubs to improve the inter-cluster communication bandwidth via Gigabit Ethernet. An application binary can run on these clusters without recompilation by using the PMv2 network abstraction feature, or it can run on the single 1024-node cluster. The cluster hardware is controlled from a single management console and can be powered on/off, reset, and monitored by issuing simple commands or by using a GUI.

3.2 RSCC system performance

To measure the system performance, we ran the Linpack benchmark program. We used High Performance Linpack (HPL)[6] as the framework and Fujitsu's BLAS for basic matrix operations. Before running it on the fully configured combined cluster, we measured the performance of the largest cluster (512 nodes) connected by InfiniBand. This cluster operated at 4565 GFLOPS or an efficiency (actual performance/peak performance) of 72.8%. The RSCC system has the highest efficiency among the Xeon systems in Professor Dongarra's top 100 Linpack performance list (**Table 2**).

The fully configured RSCC operated at 8729 GFLOPS or an efficiency of 69.64%. This performance makes the RSCC the seventh fastest supercomputer in the world (**Table 3**) and the fastest Linux cluster in Japan.[8] Although the cluster network consists of three kinds of networks (InfiniBand, Myrinet, and Gigabit Ethernet), SCore hides the underlining network differences so HPL applications can do high-performance data communication simply by using single-level MPI calls.[6]

Table 2
Top 10 efficient Xeon clusters in top 100 Linpack list.

System	Interconnect	CPU freq.	No. of CPUs	Rmax	Efficiency
Fujitsu PRIMERGY RX200	InfiniBand	3.06	1024	4.56	72.84
IBM/Quadrics	QsNet	2.4	1920	6.59	71.46
IBM eServer 1350	QsNet	3.06	1456	6.23	69.94
RIKEN Super Combined Cluster	IB+Myrinet+GbE	3.06	2048	8.73	69.63
PowerEdge HPC Cluster	GigE	2.4	600	2	69.58
LinuxNetworX	Quadrics QsNet	2.4	2304	7.63	69.03
IBM eServer 1350	Myrinet2K	3.06	768	3.23	68.74
Intel dual Pentium Xeon	Myrinet	3.06	598	2.46	67.08
Dell PowerEdge 1750	Myrinet	3.06	2500	9.82	64.18
Self-made	GigE	3.06	512	2	63.74

Table 3
Top 10 high-performance supercomputers in the world.

Rank	System Name	Interconnect	No. of CPUs	Type of CPU	LinuxOS
1	Earth Simulator	CUSTOM	5120	Vector Processor	
2	Lawrence Livermore NL, Thunder	Quadrics	4096	Itanium2 1.4 GHz	yes
3	Los Alamos NL ASCI Q	Quadrics	8160	AlphaServer EV-68(1.25	
4	IBM Rochester BlueGene/L	CUSTOM	8192	PowerPC 440 0.5 GHz	
5	NCSA Tungsten	Myrinet	2500	Xeon 3.06 GHz	yes
6	ECMWF pSeries 690	CUSTOM	2112	Power4+ 1.9 GHz	
7	**RIKEN Super Combined Cluster**	**PM Combined**	**2048**	**Xeon 3.06 GHz**	**yes**
8	IBM TWRC BlueGene/L	Quadrics	1936	PowerPC440 0.7 GHz	
9	PNNL mpp2	Quadrics	1936	Itanium 2 1.5 GHz	yes
10	Shanghai SC Dawning 4000A	Myrinet2000	2560	Opteron 848 2.2 GHz	yes

4. Conclusion

We described technologies for realizing a reliable, high-performance Linux cluster system. These technologies included a high-performance interconnect network, basic programming software, and cluster management. We also presented an overview of the 1024-node, 2048-CPU RIKEN Super Combined Cluster system as an example implementation of these technologies. Thanks to these technologies, the RSCC system is the seventh fastest supercomputer in the world and the fastest Linux cluster in Japan. Also, the 128-node InfiniBand-connected cluster has the highest efficiency among the Intel Xeon based Linux clusters. Linux cluster applications are rapidly spreading, so large cluster systems will soon be in widespread use. We will continue to develop new technologies that support reliable, high-performance, large-scale Linux clusters that are also easy to use.

References

1) SCore Cluster System Software Version 5.8 Release Note.
http://www.pccluster.org/index.html.en

2) T. Takahashi, S. Sumimoto, A. Hori, H. Harada, and Y. Ishikawa: PM2: High Performance Communication Middleware for Heterogeneous Network Environment, SC2000, Nov. 2000.

3) A. Hori, H. Tezuka, F. O'Carroll, and Y. Ishikawa: Overhead Analysis of Preemptive Gang Scheduling. IPPS'98 Workshop on Job Scheduling Strategies for Parallel Processing, volume 1459 of Lecture Notes in Computer Science, Springer-Verlag, April 1998, p.217-230.

4) K. Goto and R. van de Geijn: On Reducing TLB Misses in Matrix Multiplication. FLAME Working Note #9, The University of Texas at Austin, Department of Computer Sciences. Technical Report TR-2002-55. Nov. 2002.
http://www.cs.utexas.edu/users/flame/pubs.html

5) A. Naruse, S. Sumimoto, and K. Kumon: Optimization and Evaluation of Linpack Benchmark for Xeon Processor. Proceedings of SACSIS 2004, p.287-294.

6) HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers.
http://www.netlib.org/benchmark/hpl/

7) RIKEN.
http://www.riken.jp

8) Top 500 List for June 2004, Top 500 supercomputer site,
http://www.top500.org/dlist/2004/06/

Kouichi Kumon received the M.S. degree in Electrical Engineering from the University of Tokyo, Tokyo, Japan in 1980 and then completed three years of the doctorial course in Electrical Engineering at the university's graduate school. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1984, where he has been engaged in development of hardware and software for parallel computer systems. He is a member of the Information Processing Society of Japan (IPSJ).

Toshiyuki Kimura received the B.S. degree in Computer Science from Kyusyu University, Fukuoka, Japan in 1977. He joined Fujitsu Ltd., Kawasaki, Japan in 1977, where he was engaged in development of peripheral equipment for computer and Unix systems. Since 1997, he has been engaged in development of IA server systems.

Kohichiro Hotta received the B.S. degree in Information Science from the University of Tokyo, Tokyo, Japan in 1980 and the Masters degree in Management from McGill University, Montreal, Canada in 2001. He joined Fujitsu Ltd., Numazu, Japan in 1980, where he has been engaged in development of compilers for high-performance computer systems. He was the leader of the group of Advanced Parallelizing Compiler Technology in the Advanced Parallelizing Compiler Japanese National Project from 2000 to 2003. He is a member of the Information Processing Society of Japan (IPSJ) and the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Takayuki Hoshiya received the B.S. degree in Information Physics from the University of Tokyo, Tokyo, Japan in 1985. He joined Fujitsu Laboratories Ltd., Atsugi, Japan in 1985, where he has been engaged in research and development of liquid crystal displays. Since 1993, he has been in charge of operating system development for supercomputer systems at Fujitsu Ltd.