White paper
Advanced Software for the FUJITSU Supercomputer PRIMEHPC FX100

Next Generation Technical Computing Unit
Fujitsu Limited

Contents

System Software Overview

System software structure

The system software *FUJITSU Software Technical Computing Suite* (or simply *Technical Computing Suite*) developed by Fujitsu provides a petascale system operation and application environment for the FUJITSU Supercomputer PRIMEHPC FX100 (or simply PRIMEHPC FX100). Development of the software began with a supercomputer, the K computer[(*1)]. Based on our experience with large-scale operations on the K computer, we have improved the performance and enhanced the features of the system software. Initially implemented in the preceding FUJITSU Supercomputer PRIMEHPC FX10 (or simply PRIMEHPC FX10), *Technical Computing Suite* works with higher-density and power-saving CPUs, focusing on the exascale era to come. The structure of the PRIMEHPC FX100 system software is shown in the following figure.

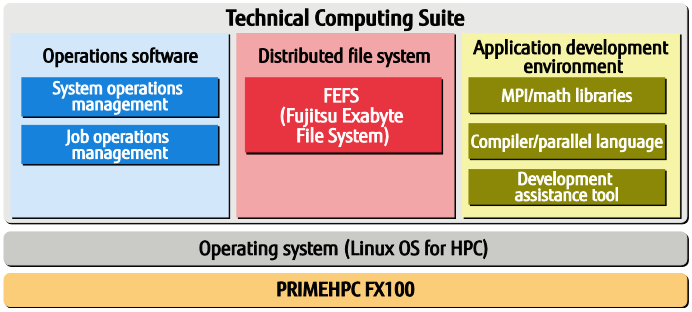


Figure 1 System software structure

Technical Computing Suite functions as HPC middleware providing a program execution environment for users and operational functions for administrators. The operating system provides an application programming interface for scientific computing programs using the PRIMEHPC FX100.

With the performance improvements and functional enhancements described below, the above software delivers higher application performance and simplifies development.

■ Operating system (Linux OS for HPC)

The operating system is based on a Linux kernel, with the OS libraries expanded to about 800 packages, which makes it easy to port from advanced programming languages and many applications. The operating system noise has been reduced to maximize the massively parallel performance of the PRIMEHPC FX100.

■ Operations software

Many users can execute their programs because the operations software allocates computer resources effectively. The software helps the whole system continue to operate even if a single point of failure occurs. The software also helps integrate and facilitate management of a large-scale system.

- System operations management

This software provides an integrated centralized system management view for a large system that has hundreds to tens of thousands of compute nodes. From the centralized management view, operators can easily control system start and stop, and enable automatic system operations for failure monitoring and isolation of faulty components. As an extension of the base technology for large-scale system management from our experience with the K computer, the software can also manage hybrid configurations of PRIMEHPC FX100 and PC clusters.

- Job operations management

This software makes it possible not only to execute a single job that uses tens of thousands of nodes but also to effectively execute a wide variation of lots of user jobs. The many user jobs can share large-scale system resources among themselves when sharing and job priorities have been set for the job scheduler. The PRIMEHPC FX100 has an added job allocation function for effective operation in small-scale configurations.

■ Distributed file system (FEFS)

Based on the K computer, the FEFS is a high-speed file system supporting large-scale systems. The K computer file system is capable of high-speed processing in a large-scale environment, as indicated by the following.

- The system supports large-scale systems. Hundreds of thousands of clients can stably access a file system consisting of thousands of storage devices (OSS/OST)[(*2)].
- The file system can support large volumes of data up to the petabyte level.
- The massive file I/O of one user does not affect the file I/O of other users.

With these features as a base, the PRIMEHPC FX100 improves the single process I/O performance for a particular process in a job that writes a large quantity of files.

■ Application development products

This integrated software suite is used to develop (compile, debug, tuning, etc.) and execute scientific computing programs written in Fortran, C, or C++. It supports parallelization technology, such as automatic parallelization, OpenMP, MPI, and the XPFortran language.

- Compiler, parallel language, MPI, math libraries

They support HPC-ACE2 and new language standards while maintaining upward compatibility for PRIMEHPC FX10 applications. Moreover, scientific computing programs can perform high-speed parallel processing by leveraging the PRIMEHPC FX100 hardware functions.

- Development assistance tool (application development environment)

This environment makes available the process cost information for the large page memory allocation function and deallocation function. It can graphically display the information collected from

the application layer by the operating system and effectively tune applications for a massively parallel computer.

This system software can easily support a university computing center with tens of thousands of computers, application development at research institutes, and the total process from execution to analysis of a scientific simulation.

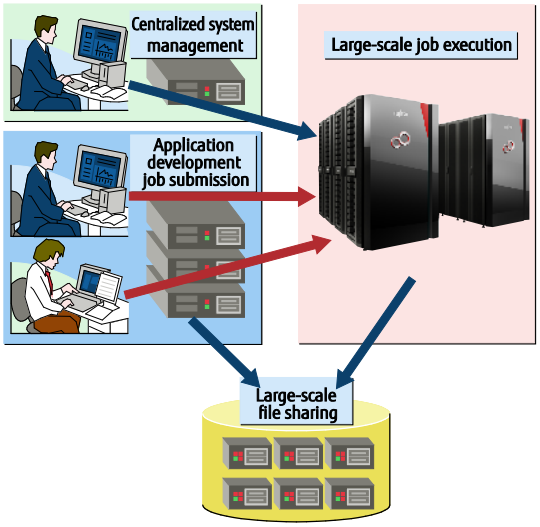


Figure 2 System software operation environment

Software challenges to reaching the 100-petabyte scale

The system software was designed and developed for the following purposes so that a large number of users can efficiently use a massively parallel system at the 100-petabyte scale:

- Efficient use of a large-scale system
- Easy application development
- Highly parallel simulation in actual practice

■ Efficient use of a large-scale system

Administrators and users of the PRIMEHPC FX100 can centrally manage hundreds to hundreds of thousands of compute nodes by using system management functions and job operations management functions. These functions bring about efficient use of large-scale systems as supercomputer processing power helps to bring about various degrees of convenience. Specifically, the assistant cores provide the following benefits.

- Assistant core utilization 1: Reducing system noise

System noise (OS jitter) interferes with job execution. For example, in a system with tens of thousands of nodes, the interference increases in proportion to the increase in the number of nodes and may degrade performance. The OS jitter index of the system is calculated from the average noise rate (indicating the frequency of noise occurrence) and maximum noise length (maximum runtime of the OS process generating the noise). The average noise rate is calculated from the logical processing time and actual processing time of the measurement program. In a typical Linux server, the average noise ratio is at the 10^{-3} level, whereas in the K computer and the PRIMEHPC FX10, the average noise ratio is at the 10^{-5} level. To improve the maximum noise length, a function developed for the PRIMEHPC FX100 separates the CPU bandwidth of the assistant cores used exclusively by the system daemons and I/O processing, so only the assistant cores process the OS processes that generate noise. The improved maximum noise length is 2/3 of that of the K computer and the PRIMEHPC FX10.

Table 1 Comparison of system noise

	PRIMEHPC FX100	PRIMEHPC FX10	PC cluster
Average noise rate	5.10×10^{-5}	5.20×10^{-5}	1.08×10^{-3}
Maximum noise length (μs)	38.0	59.4	931.7

- Assistant core utilization 2: Asynchronous communication processing

In conventional systems, the job computing process starts after control communication within MPI acknowledges a response from the communication partner. Another method (non-blocking communication function) has an assistant core process the response to the communication partner, and the start of job calculation overlaps with the wait for the response. In the PRIMEHPC FX100, an MPI asynchronous communication thread is allocated to an assistant core, and communication and computation are processed asynchronously. The MPI asynchronous communication thread is processed with high priority even if the CPU is overloaded. This processing is implemented in the Linux kernel as the guaranteed response function for MPI communication.

- Assistant core utilization 3: Promoting the separation of file I/O processes and jobs

I/O processes are routed to use the FEFS connected to an InfiniBand or other data transfer network from a compute node that is interconnected through an interconnect. In the K computer and the PRIMEHPC FX10, I/O processes are routed by I/O exclusive nodes (calculation node and relay node). But in the PRIMEHPC FX100, the processes are routed by the assistant cores, which eliminate the need for I/O exclusive nodes. Also, the assistant cores can also process the FEFS client function of compute nodes, so file I/O processes and job communication processes can be separated.

- High reliability and high availability

The number of system failures, including incidents of software faults, increases with hardware scale, so it is especially important in large-scale systems to detect faults early and isolate or recover faulty components. The system management software for the PRIMEHPC FX100 can detect faults in a timely manner through the inherited low-overhead hierarchy management capabilities already implemented in the PRIMEHPC FX10. Using these capabilities, the software re-executes jobs automatically. As a result, users obtain simulation results without awareness of system faults.

- Sharing by numerous users

Using the job operations management system, users in a computer center can process a large-scale simulation using many compute nodes. They can likewise process many small-scale simulations that handle a wide variety of parameters and input data. The management system allocates the necessary resources and I/O bandwidth for that processing to the users. Resource allocation and I/O bandwidth allocation methods for user applications can be configured with *Technical Computing Suite* so that the computer center operates according to the operation policy that is set by the administrator. Two examples of rules in such a policy are "prioritize system usage efficiency" and "prioritize job execution performance."

The job scheduler selects the jobs to be executed according to computer center operation rules so users share the system fairly. The job scheduler also assigns compute nodes to jobs such that system resources are fully utilized. Specific to I/O resources, the FEFS has a fair share function to prevent particular users from occupying I/O resources.

- **Easy application development**

  Some supercomputers provide a proprietary OS and compilers with limited functionality compared to the standard UNIX functions, so porting (source code modification) and verification of existing applications is required. That is not necessary with the PRIMEHPC FX100, which inherits the OS and the industry standard API support for compilers. It also supports the new standards for Fortran, C, and C++ compilers and MPI libraries to be compliant with industry standards.

- **Highly parallel simulation in actual practice**

  - Parallel programming model

  Inter-core performance of the automatic parallel compiler in the hybrid model has been improved upon in this model to support exascale many-core architectures. Also, the job execution environment was developed to provide a means to set the number of threads per application, because the optimal number of processes and threads differs depending on the hybrid program.

  - Parallel file access

  High-speed access to input/output data is important to a high-speed simulation using many compute nodes. The PRIMEHPC FX100 accomplishes input/output processing of massive amounts of data in a short time by distributing file data to multiple servers and using the MPI-I/O parallel access interface.

*1 The **K** computer has been jointly developed by RIKEN and Fujitsu. **K** computer is the registered trademark of RIKEN.

*2 The OSS (object storage server) is a server for controlling file data. The OST (object storage target) is a storage device connected to the OSS.

System Operations Management

Effective management of system operations

System operations management has grown ever more important as systems grow larger in scale to increase system performance. The latest supercomputer systems, which are the largest-scale systems, tend to consist of thousands to tens of thousands of compute nodes. Due to that size, they need a function facilitating status management and operation control to operate efficiently. The system operations management function of the PRIMEHPC FX100 was developed by Fujitsu based on our operations management experience with the K computer, which has over 80,000 compute nodes. This capability to efficiently manage large-scale systems has the following features.

- Scalability supporting massive-scale systems
  - Monitoring process load distribution through hierarchical management
  - Frequent communication reduced through coordination of the notification process
- Flexible and efficient system operations
  - Centralized management of multiple clusters
  - Integrated management from installation to maintenance and routine task monitoring
  - Flexible system configuration changes appropriate to operation
- High availability and continuous operation
  - Automatic job resubmission upon fault detection
  - Redundancy of important nodes, and automatic node switching

The features of the system operations management function and how they work are described below.

Scalability supporting massive-scale systems

■ Monitoring process load distribution through hierarchical management

System operations management is processed with a hierarchical structure, such as shown in Figure 3, to distribute the monitoring load, etc. in a massive-scale configuration. The hierarchical structure divides the entire system into logical units called node groups. To distribute the system monitoring load and system control load, each job operations management subnode handles the system operations management processing that falls within the scope of its node group. System scalability is higher in this structure because administrators can easily manage any additional nodes by simply adding it to a node group.

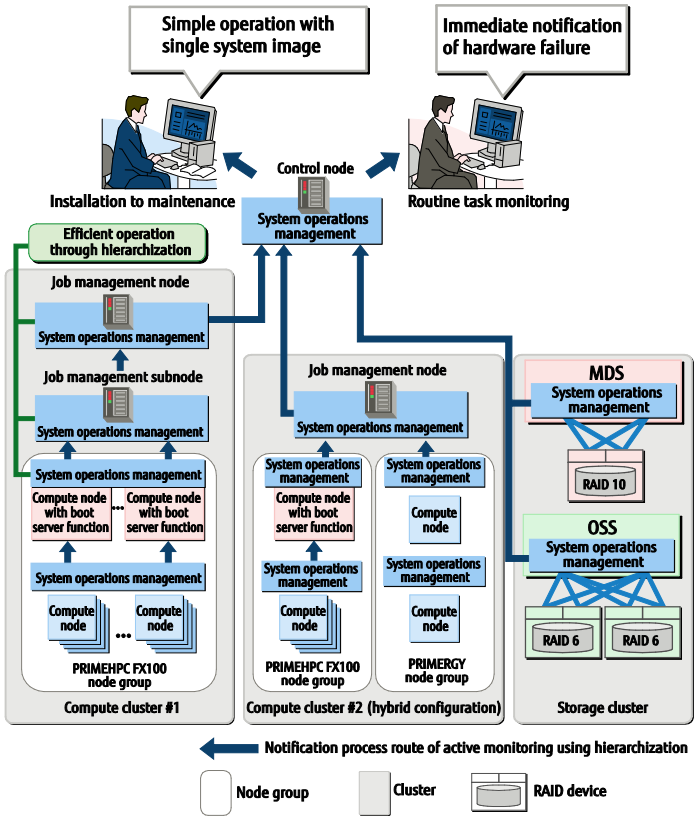


Figure 3 Grouping and hierarchization in massive-scale system management

■ Frequent communication reduced through coordination of the notification process

In a system with a single node taking care of alive monitoring of thousands to tens of thousands of compute nodes and service state monitoring, an extremely heavy load is placed on this one node. In the PRIMEHPC FX100, the operations management software increases the efficiency of that monitoring by using functions unique to PRIMEHPC and structuring nodes in a hierarchy. In the hierarchy, lower-order compute nodes do not notify higher-order nodes of faults until a certain number of notifications are accumulated and combined into a single report, which is then sent to the higher-order nodes. So even if the states of many compute nodes suddenly change, this function can ensure efficient notification of higher-order nodes by preventing large amounts of information from being frequently transmitted to them.

Flexible and efficient system operations

■ Centralized management of multiple clusters

Operations are managed per cluster from the control node as shown in Figure 3. Node groups with the PRIMEHPC FX100, node groups with PC servers, and storage groups with disk devices are considered to be clusters. In a hybrid configuration, one cluster may also consist of a small-scale node group of the PRIMEHPC FX100 and a PC server node group.

■ **Integrated management from installation to maintenance and routine task monitoring**

From the control node, the administrator can centrally manage tasks from installation to maintenance. Here are a few examples.

- Software installation using a distributed processing installer

Generally, installation of the OS and various packages on a large number of node groups is extremely time-consuming. The subsequent management of the installed software is also difficult. The PRIMEHPC FX100 provides an installer specially designed for large-scale installation. This installer supports the above-described hierarchical structure of node groups to centrally manage the application status of packages and configuration files, so the administrator of a large-scale system can install software and manage their packages and settings with less effort and time. In the hierarchy shown in Figure 3, the control node acts as an installation server, and the job management subnodes and the compute nodes with the boot server function act as intermediate installation servers. The resulting three-tier installation structure distributes and speeds up processing. Furthermore, by synchronizing with the intermediate installation servers, the installer on the control node centrally manages installation information and configuration file contents to keep all that data unified within the system.

- Display of summaries

A large-scale system is considered to have more than 100,000 nodes. If their status is displayed simply with one node per line from the command-line interface of a terminal, the displayed information extends over more than 100,000 lines. When displayed in this way, the information does not present an overall picture of the system so something important may be overlooked, which is a potential problem. The standard PRIMEHPC display format shows only the quantities of nodes by operational status in units of clusters of the PRIMEHPC FX100, PC servers, and storage systems. As shown in Figure 4, users can get a step-by-step understanding of a failure by:
(1) identifying the cluster with the failure from the summarized view,
(2) filtering the faulty node by specifying the failure option, and
(3) specifying the node option.

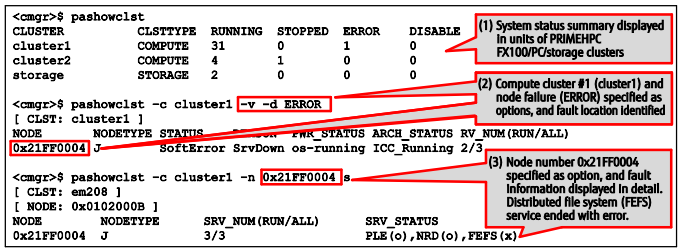


Figure 4 Step-by-step comprehension of information

- Standardized system configuration information and status display

Users and administrators want to get increasingly diverse information. The PRIMEHPC FX100 offers a great variety of information about each node in the system configuration, including hardware and software configuration information. The hardware information includes the number of installed CPUs and amount of installed memory on the node, and the software configuration information includes the assigned IP address and role of the node. There is also node state-related information, such as whether the node power is on and whether hardware has failed or software has a defect. The system configuration information is handled by a variety of commands, depending on the user and use scenario, which have a standardized command display and specification formats to prevent confusion among users. All the various software designed for the PRIMEHPC FX100 have standardized forms of expression for the system configuration information.

■ **Flexible system configuration changes appropriate to operation**

A joint research center operates with resource units defined to support various job operation policies. Resource units can start the job scheduler function. They are the logical divisions of nodes within a cluster. In the PRIMEHPC FX100, multiple resource units can be defined in a compute cluster, and configurations can be changed dynamically, like with an expansion, reduction, division, or merger of resource units, even during operation. It is also possible to lend a compute node in a joint research center to a specific user. The PRIMEHPC FX100 uses the partitioning function to confine interconnect communication by cluster, so the lending and other operations are secure.

High availability and continuous operation

■ **Automatic job resubmission upon fault detection**

The PRIMEHPC FX100 can detect node failures and remove the failed nodes from operation. The PRIMEHPC FX100 detects the failures in two ways. The first method is system monitoring by software. Using the hierarchical structure of node groups, the software efficiently collects the status of nodes and services to detect node failures while distributing the monitoring load. The second method is linking with failure notification by hardware. Through the internal PRIMEHPC FX100 mechanism for notification of node- and interconnect-related hardware failures, node failures can be instantly detected.

To continue operation after a node fails, the PRIMEHPC FX100 terminates the processing of all the jobs running on the node, and the jobs are automatically restarted on available nodes.

■ **Redundancy of important nodes, and automatic node switching**

The important nodes essential to operations management in the PRIMEHPC FX100 include the control node, job management node, and job management subnode. If any these nodes cannot start due to failure, all or part of system operations are suspended, and operation cannot continue. To prevent that situation and continue operation, the PRIMEHPC FX100 can configure all of these nodes in a redundant configuration as active and standby systems. The detection of a failure in the active node will result in automatically switching to the system to the standby system to continue operation.

Job Operations Management

Job schedulers

Users of the PRIMEHPC FX100 can run many jobs without concern, even in a large-scale system with 100,000 nodes, because of Fujitsu's continuous efforts to develop batch job execution environments (job schedulers). The PRIMEHPC FX100 has a large-scale job scheduler based on the performance-proven K computer, which has approximately 80,000 nodes. The job scheduler also has enhanced functions. The job scheduler functions of the K computer and the features of PRIMEHPC FX100 functional enhancements are discussed below.

Job scheduler functions of the K computer

As system sizes have increased, the number of jobs handled by a job scheduler has increased significantly. The K computer handles more than one million jobs in the system, using the following job operation functions developed for massive-scale systems:

- Function for reliable control of many jobs (large-scale support)
- Support of myriad requirements for a joint research center (operability)

■ Function for reliable control of many jobs (large-scale support)

The system for job management in a large-scale system is fast and reliable--characteristics made a reality by the following mechanisms.

- The resource management function reserves the computing resources used per job in advance and operates at high speeds with the threads/processes attached (bound) to CPU cores working together with the language system (runtime).
- To take full advantage of the characteristics of the Tofu (Torus fusion) interconnect, the job scheduler groups compute nodes (with the same temporal distance between the nodes) and assigns each job to compute nodes belonging to the same group. The job scheduler also controls node allocation to maximize the amount of space with consecutive empty nodes.
- The job scheduler function shortens the selection process by parallelizing the process for selecting the optimal resources for job execution from vast computing resources.
- The backfill function improves the system operating ratio by executing small-scale jobs before a large-scale job provided that the small-scale jobs finish before the scheduled execution time of the large-scale job. So any computing resources used by the small-scale jobs are available to the large-scale job when executed, since the small-scale jobs release their computing resources before the execution of the large-scale job begins.
- The distributed parallel execution function starts jobs at a high speed by using the hierarchical node group structure of job management nodes and compute nodes to efficiently generate and manage processes.
- For greater responsiveness, multi-processing and multi-threading of the process flow from job reception to job end improves the operational response.

■ Support of myriad requirements for a joint research center (operability)

A number of joint research centers run supercomputer systems, with the operation policy on job execution varying slightly from one center to another. From our dealings with customers, Fujitsu has implemented functional enhancements reflecting the customers' operating requirements into our products. Here are a few examples.

- The job ACL function was developed to flexibly support the operation policies differing between joint research centers. In the K computer, the items subject to management are consolidated into the following two categories to reduce the management work by administrators:
  - Restrictions (upper and lower limits of the quantity of resources required for job execution, default value assumed when a specification is omitted, etc.) that apply to a job when the job is executed
  - Policies on overall job operations, such as which jobs have priority for execution
- To support computing methods like parameter study, the bulk job function can submit a massive number of jobs in the same process with only the parameters changed.
- With the display interface, users can sort out the detailed information they need from a job summary view and thus easily comprehend the status even with a massive number of jobs.

Features of the PRIMEHPC FX100 job scheduler

Incorporating the job scheduler functions of the K computer, the PRIMEHPC FX100 job scheduler has other built-in functions to improve the system operating ratio of small-scale centers and make full use of the numerous cores in the system. In particular, an added operation function works together with PC server jobs and emergency jobs highly requested by customers in the field of climatology.

■ High operating ratio function - Expanded job allocation methods

The job allocation method can be selected to meet a guarantee of job communication performance and user requirements, such as early execution.

- Exclusive node allocation using torus mode (K computer, PRIMEHPC FX10, PRIMEHPC FX100)
  In this conventional allocation method, the minimum allocation unit is a Tofu unit of 3 x 2 x 2 nodes. Communication can be optimized to minimize unstable performance so that the same physical shape on the six-dimensional mesh/torus can be allocated at any time.

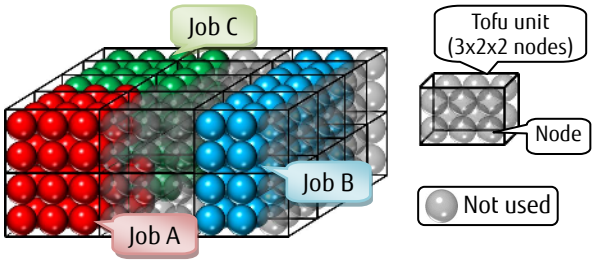


Figure 5 Job allocation using torus mode

– Exclusive node allocation using mesh mode (PRIMEHPC FX100)
In this allocation method, the minimum unit is 1 node. The lack of a guarantee of the physical shape on the six-dimensional mesh/torus may cause some unstable communication performance. Unlike torus mode, however, excessive numbers of nodes are not required, which leads to an improved node utilization rate.

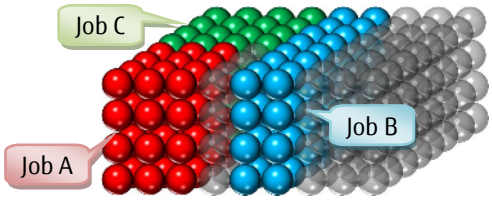


Figure 6 Job allocation using mesh mode

– Node utilization rate improvement using mesh mode
In mesh mode, system operations in a center where various jobs are executed concurrently can be expected to have a better node utilization rate and job turnaround performance. The improvement of the node utilization rate in mesh mode was simulated and assessed using actual job operation data. The data was provided by JAXA. The improvement in mesh mode was assessed with a simulation of the job scheduling status where approximately one month of jobs was submitted at the same time. Figures 7 and 8 and Table 2 show the simulation results for the node utilization rate and job wait time per job size. The results confirm that the node utilization rate and system throughput improved without greatly affecting the execution distribution of medium- and large-scale jobs. Looking at small-scale jobs, you can see a significant improvement of the node utilization rate from 86.2% to 91.9% when there is a wait for job execution. The execution wait time of these jobs also improves from 638.220 seconds to 415.560 seconds. This means that small-scale jobs running in mesh mode effectively use the nodes remaining from the execution of large-scale jobs.

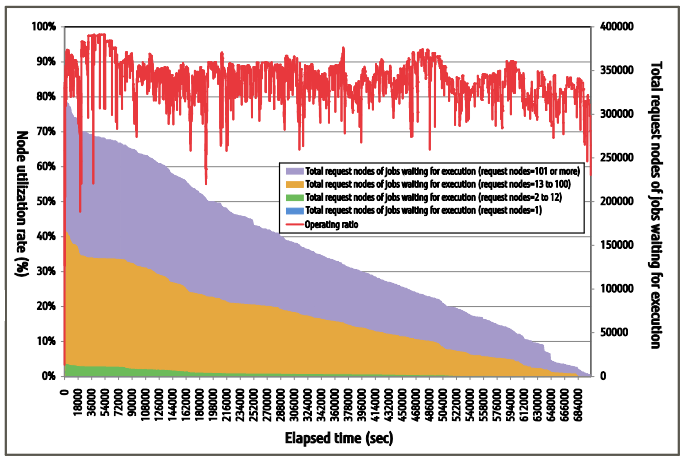


Figure 7 Simulated job execution in torus mode (conventional)

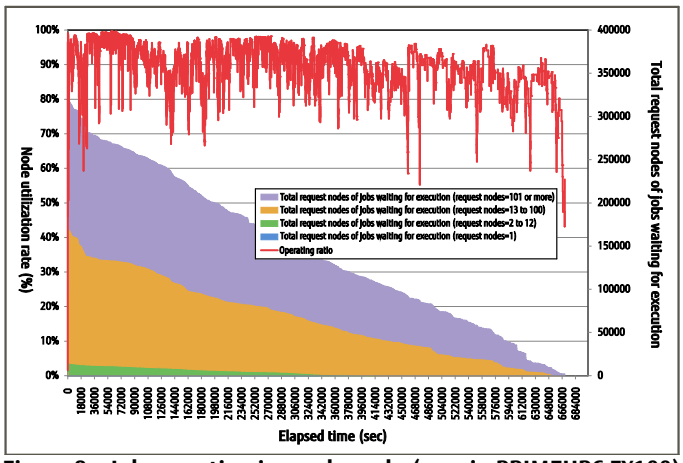


Figure 8 Job execution in mesh mode (new in PRIMEHPC FX100)

Table 2 Job wait times and node utilization rates in each mode

	Torus mode	Mesh mode
Time until all waiting jobs are finished (Node utilization rate)	700,800 sec (85.5%)	669,420 sec (89.7%)
Time until waiting small-scale jobs (under 12 nodes) are finished (Node utilization rate)	638,220 sec (86.2%)	415,560 sec (91.9%)

■ Enhanced hybrid parallel function for making full use of many cores

In anticipation of growing widespread use of many-core CPU architectures for exascale computers, the hybrid parallel function of the PRIMEHPC FX100 has been enhanced to make full use of the many cores. The hybrid parallel processing supported by the PRIMEHPC FX10 combines MPI parallel processing and thread parallel processing.

- Multiscale and multiphysics support

Multiscale and multiphysics analysis combining various events was once impossible but not anymore because of advances in computing capabilities. The PRIMEHPC FX100 can link and compute multiple MPI programs by running the multiple programs in a job script and using the "Establishing Communication" function for communication between the programs. The optimal number of threads in an MPI

program varies between programs, so the PRIMEHPC FX100 provides a function to specify the number of threads per MPI program.

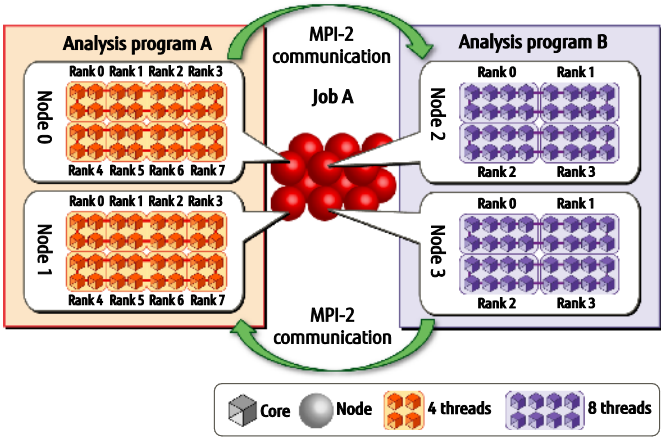


Figure 9 Multiscale and multiphysics

- Preventing load imbalances

Meteorological analysis and impact analysis are examples of applications that concentrate load on a specific process, causing a bottleneck and possibly degrading the overall performance (load imbalance). To prevent load imbalances, a lot of CPU resources need to be allocated to the process with the high computing load. The PRIMEHPC FX100 has an enhanced function for flexibly allocating CPU resources according to the computing load.

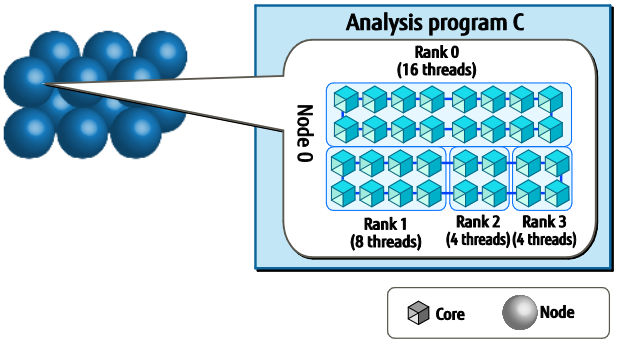


Figure 10 Preventing load imbalances

■ Node sharing jobs

Another enhanced function in the PRIMEHPC FX100 effectively utilizes cores by running multiple jobs in a node.

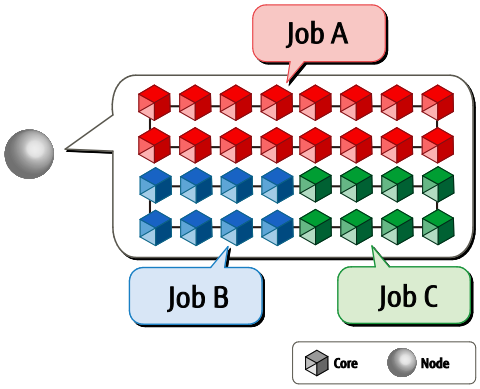


Figure 11 Node sharing jobs

■ Emergency jobs

The emergency job function added to the PRIMEHPC FX100 urgently simulates the process of tsunami propagation, such as after an earthquake occurs. The job scheduler tries to allocate resources with the highest priority to a submitted emergency job. If there are no free resources, this function selects and suspends a running job in accordance with an assessment policy, releases the resources of the job, and allocates the released resources to the emergency job. With the following principles in mind, the administrator can flexibly set the assessment policy:

- Select a job to suspend so that the emergency job has a minimal wait time for execution
- Select a job to suspend to obtain a high system operating ratio
- Select a job to suspend from the low-impact jobs

- How resources are released by a job swap

Emergency jobs have a short wait time for execution because resource management of the emergency job function automatically selects a resource release method from the three methods described below.

– Logical swap
Logical swap is selected when the memory resources of the emergency job have free space. The logical swap releases only the CPU from the target job. If there are still insufficient memory resources, the partial swap or physical swap method is selected.

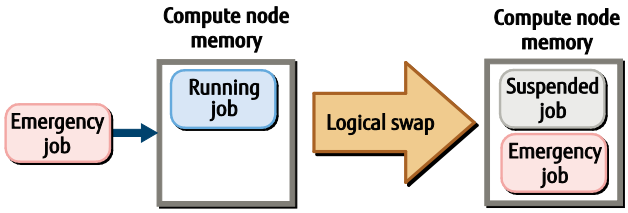


Figure 12 How resources are released by logical swap

– Partial swap
Partial swap is selected when memory resources can be allocated to the emergency job from a partial release of the memory resources of the target job. The partial swap writes part of the memory contents of the target job to a swap-out file. If there are still insufficient memory resources, the physical swap method is selected.

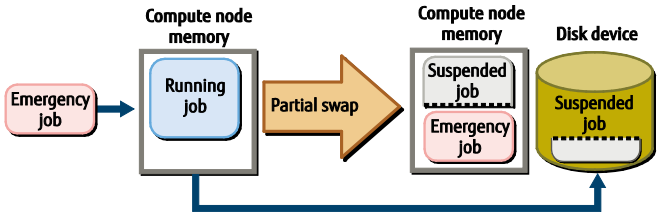


Figure 13 How resources are released by partial swap

– Physical swap
A physical swap releases both the CPU and memory of the target job. The memory contents are written to a swap-out file.

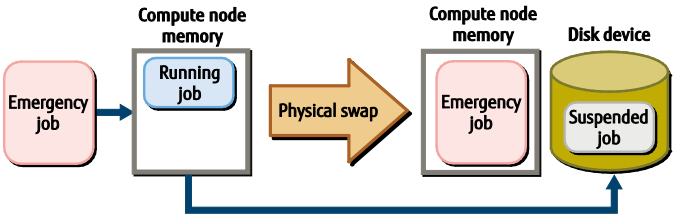


Figure 14 How resources are released by physical swap

Hybrid operation with a PC server

A single job management node centrally manages the PRIMEHPC FX100 and PC servers as compute nodes and can schedule jobs accordingly. Since jobs are submitted to the respective PRIMEHPC FX100 and PC servers, step execution between jobs (where the execution results from one job are used as the input for the next job) becomes possible.

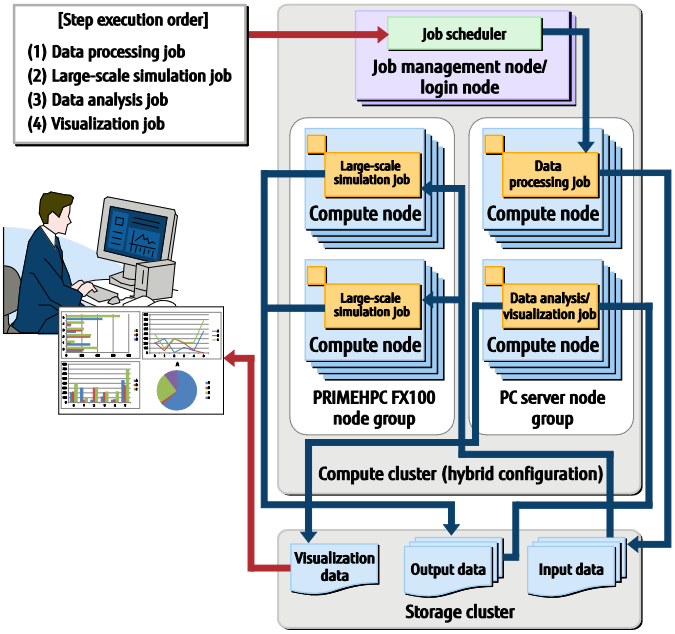


Figure 15 Step execution of job in hybrid configuration

- Step execution is enabled when the user assets (applications) of PC servers are linked with applications optimized for the PRIMEHPC FX100.
- Step execution is enabled for a series of processes, such as pre-processing (data processing, etc.) on a PC server, followed by application execution on the PRIMEHPC FX100 for a large-scale simulation, and ending with post-processing (data analysis/visualization, etc.) of the results on the PC server.

Distributed File System

"FEFS" is the name of the distributed file system that provides the high reliability required for supercomputer-level processing power. The FEFS is also stable, since file system stability is directly related to the stability of a supercomputer. The implemented file system also delivers high I/O parallel performance as it is important to minimize the file I/O time of tens of thousands of nodes. Otherwise, the supercomputer cannot make full use of its computing capabilities.

As discussed below, the FEFS achieves both high performance and high reliability.

Cluster-type file system

FEFS stands for Fujitsu Exabyte File System, which is a cluster-type distributed file system. Originally based on the open-source Lustre, which is the de facto standard for HPC file systems, the FEFS has not just inherited the excellent performance and functions of Lustre, such as high parallelization and scalability but gone further with enhancements featuring high performance, stability, and usability. It can be widely adapted to massive-scale systems, such as the K computer, and to medium- and small-scale center operations.

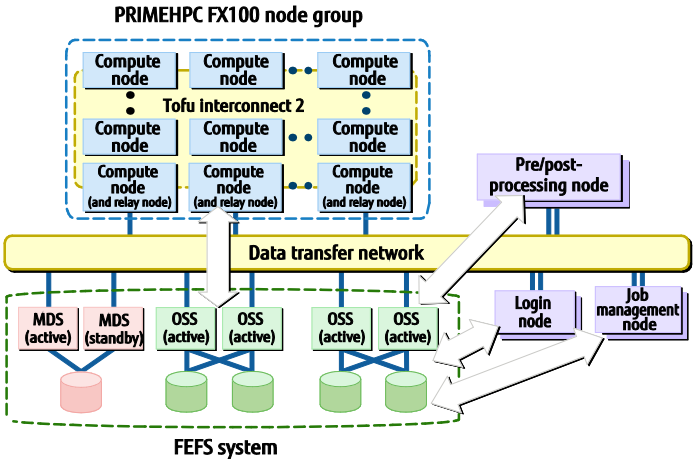


Figure 16 PRIMEHPC FX100 + FEFS system configuration

Many successes with large-scale systems

Inheriting the technology gained through the development and operation of the K computer, the FEFS proved itself to be a success with the K computer and then achieved further success with Fujitsu's PC cluster systems and PRIMEHPC FX10, helping promote the stable operation of users' systems.

■ High scalability of over a terabyte per second

The K computer with the FEFS achieved throughput of 1.5 TB/s (parallel read performance on the local file system), which was the best in the world. The FEFS is a cluster-type file system that can scale out total throughput in proportion to the number of object storage servers used. Total throughput is several gigabytes per seconds per object storage server, so to obtain the required throughput, the number of units appropriate to the required performance have to be prepared. The PRIMEHPC FX100 has one compute node with the IO function for every chassis, and it has a mounted InfiniBand adapter for access to the FEFS file servers. The number of nodes equipped with InfiniBand can be selected flexibly according to performance requirements, and throughput performance can be scaled out in proportion to the number of nodes.

■ Elimination of interference on parallel applications

An important factor to getting the best super-parallel MPI application performance with tens of thousands of nodes is to eliminate interference from the system daemons. The start of a system daemon delays the synchronization process between parallel processes and extends the application runtime. The FEFS has completely eliminated file system daemon processes that run periodically, reducing the impact on the MPI application runtime with 80,000 nodes to 50 µs or less.

Table 3 File system daemon runtime (estimate for 80,000 nodes)

File system daemon	Execution cycle	Lustre	FEFS
Node alive monitoring	25 sec	15 sec	4 µs
Distributed lock management	1 sec	16 ms	21 µs

The PRIMEHPC FX100 has 32 compute cores and 2 assistant cores for the OS. The system daemons for I/O processing run on the assistant cores. By making applications exclusively use the compute cores, the system has successfully eliminated interference completely.

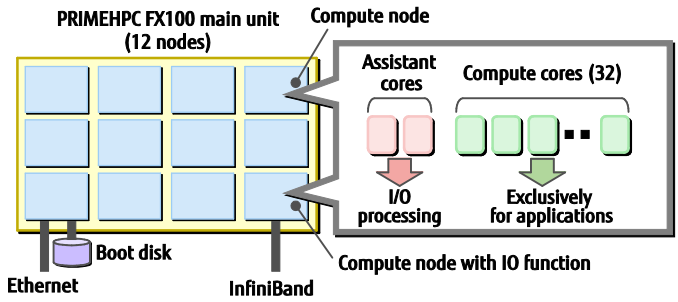


Figure 17 I/O processing and computing processing separated utilizing assistant cores

Reliability of continuous operation when hardware fails

High reliability as well as performance is essential to stable operation of a large-scale system. In a cluster-type file system consisting of many file servers, storage devices, and network devices, system operation must continue even when part of the system fails or stops. So it needs to be fault tolerant. The FEFS improves fault tolerance through hardware redundancy. Also, by using node monitoring and automatic switching in link with system management software, the file system

can continue in service even during maintenance or when a single point of failure occurs.

■ Fault tolerance

The ability to automatically detect a failure and bypass the fault location to continue file system services is a critical feature for a large-scale file system consisting of over hundreds of file servers and storage devices. The FEFS can provide continuous service as a file system by duplicating hardware components and using software-controlled switching of servers and I/O communication paths, even if a single point of failure occurs. In the PRIMEHPC FX100, if InfiniBand on a compute node with the IO function is faulty and cannot communicate, the communication path is automatically switched to use InfiniBand on another compute node with the IO function. The result is continued access to file servers and improved fault tolerance.

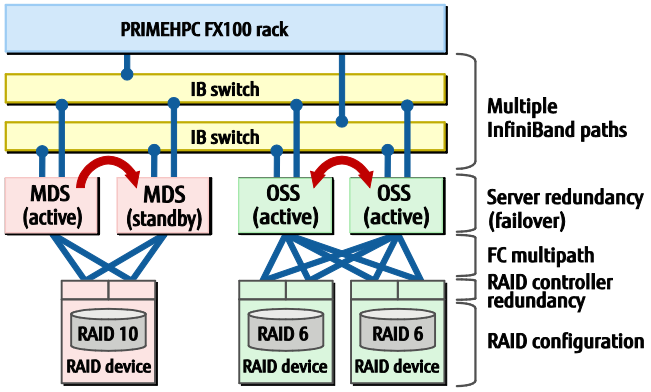


Figure 18 Fault tolerance

■ Hierarchical node monitoring and automatic switching

Large-scale systems require a scheme that can detect failures and automatically notify the affected nodes without human intervention. One scheme used so far is node state monitoring based on the monitoring of packet exchanges between compute nodes and file servers. However, one problem with this scheme is the very high number of generated monitoring packets. The number is exponentially proportional to the system scale. The resulting heavy packet transmissions hamper MPI communication between compute nodes and data communication between compute nodes and file servers. Working together with system management software, the FEFS minimizes communication loads through hierarchical node monitoring and control of switching between nodes. The FEFS monitors the nodes represented in a multi-tier tree with the following hierarchy: nodes inside the PRIMEHPC FX100 rack, node groups each consisting of multiple racks, and higher-order node groups other than the preceding node groups.

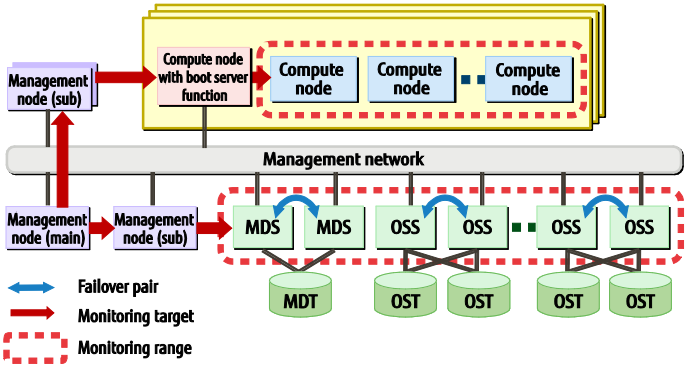


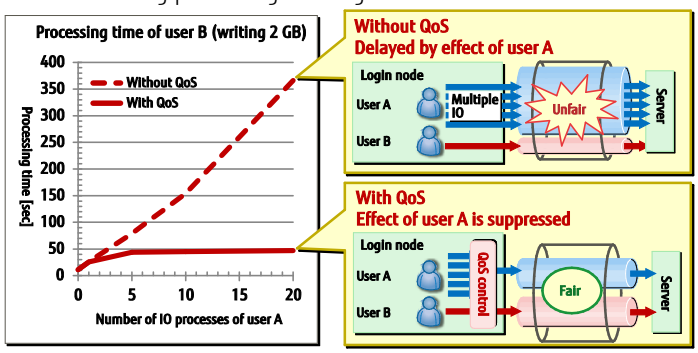
Figure 19 Node monitoring and automatic switching

Functions for improved operability

A large-scale system is used by many users, so the system must ensure that a tremendous amount of file I/O activity by any particular user does not affect other users. It must also ensure that file access by jobs on compute nodes does not affect responses to users on login nodes. The FEFS overcomes these challenges with the fair share QoS function for every user and the response-guaranteed QoS function for login nodes.

■ Fair share QoS function for every user

If many users are using a login node at the same time, a large amount of file I/O by any of the users may drastically slow down the file I/O of the other users. In the FEFS, to prevent massive I/O requests from a single user and I/O bandwidth occupation, the client side can limit the number of I/O requests that a user can issue. Figure 20 shows an example of the effect. Two users (users A and B) are using the login node, and the I/O processing time of user B is measured during processing of a large I/O volume from user A.



User A: Write by 0 to 20 processes
User B: Write by 1 process -> Measure time taken to write 2 GB

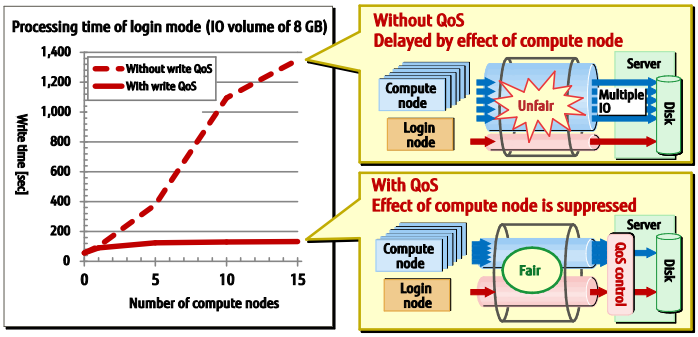
Figure 20 Fair access between multiple users using QoS

Without QoS, the delay of user B's process is proportionate to the increase in the number of user A's I/O processes (dotted line). However, QoS can suppress the effect on user B (solid line).

■ Response-guaranteed QoS function for login nodes

Access response to users is directly linked with usability, so it is more important than the access response to jobs. To ensure access response to the users on a login node, the FEFS has a function for allocating server threads that process I/O requests by node group. Thus, even during file I/O by jobs on a compute node, the system can still respond to a user who is accessing the file from the login node. Figure 21 shows an example of the effect. The write time for 8 GB of

data from the login node is measured during write processing by multiple compute nodes. Without QoS, login node processing is significantly delayed proportionate to the increase in the number of compute nodes accessing the file (dotted line). However, QoS suppresses the effect (solid line).



Compute node: Up to 15 nodes
Login node: 1 node -> Measure time taken to write 8-GB file

Figure 21 Response guarantee for login node

Performance improvement

Like a job with the I/O master method for file I/O by the representative process of the job, the I/O performance of a single process is important when writing a large amount of data from that one process. The FEFS has an enhanced Lustre request process that improves the write performance of a single process by issuing a request in parallel to the write request, according to the object storage server. Figure 22 shows the effect of the improvement. Performance with Lustre 2.5 is a little less than 1 GB/s, whereas performance with the FEFS exceeds 2 GB/s.

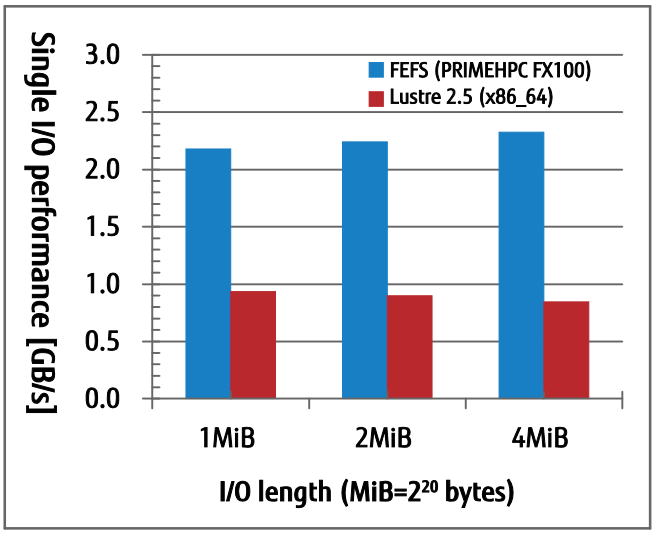


Figure 22 Single I/O performance improvement

Application Development

Language processing maximizing PRIMEHPC FX100 performance

■ Many cores leveraged by the VISIMPACT, and the effect on performance

In the PRIMEHPC FX100, VISIMPACT (Virtual Single Processor by Integrated Multicore Architecture) technology minimizes increases in communication time during parallel processing on a massive scale and maximizes execution efficiency by unit of time. The VISIMPACT shares the L2 cache between cores and implements barriers between cores in the hardware, which enables programs to handle multiple cores as one CPU using an automatic parallel compiler.

■ Hybrid programming model

Simply with a description of process parallelization between nodes, the compiler automatically performs thread parallel processing for multiple cores in a node. The ratio of thread parallel to process parallel can be changed flexibly to get the best core performance within the node. The following example shows that two processes of eight threads each are automatically parallelized and then a job consisting of four processes of four threads each is submitted.

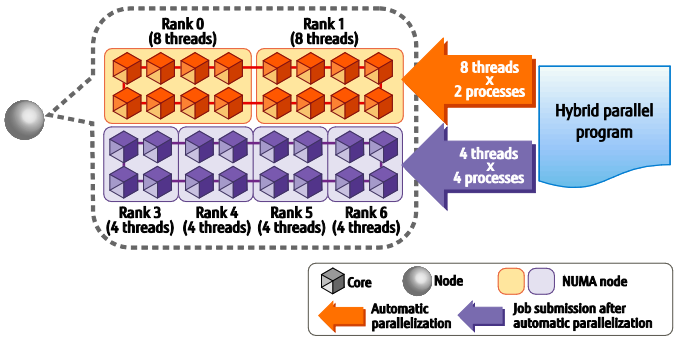


Figure 23 Compute node architecture

■ MPI communication using the Tofu interconnect 2

The Tofu interconnect 2 with improved bandwidth and latency in the PRIMEHPC FX100 is based on the Tofu interconnect technology first introduced in the K computer. Applications can achieve parallel processing of more than tens of thousands of processes by using MPI via the Tofu interconnect 2. To improve point-to-point communication performance, the PRIMEHPC FX100 uses a special type of low-latency path that bypasses the software layer. Moreover, additional consideration is given to the length and location of data being exchanged as well as the number of hops, resulting in an optimized transfer mode switch. Figures 24 and 25 show the improved performance in point-to-point communication latency and bandwidth. Congestion is controlled by a dedicated algorithm implemented for collective communication performance, drawing upon features of the Tofu interconnect 2. Frequently used functions such as MPI_Bcast, MPI_Allreduce, MPI_Allgather, and MPI_Alltoall use this special algorithm instead of point-to-point communication. For MPI_Barrier and MPI_Allreduce, processing is faster with the advanced barrier communication function (implemented in hardware) provided by the Tofu interconnect 2.

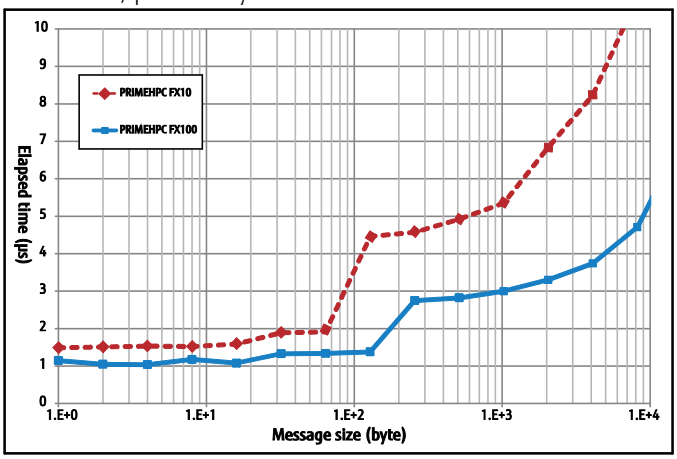


Figure 24 Point-to-point communication performance (latency)

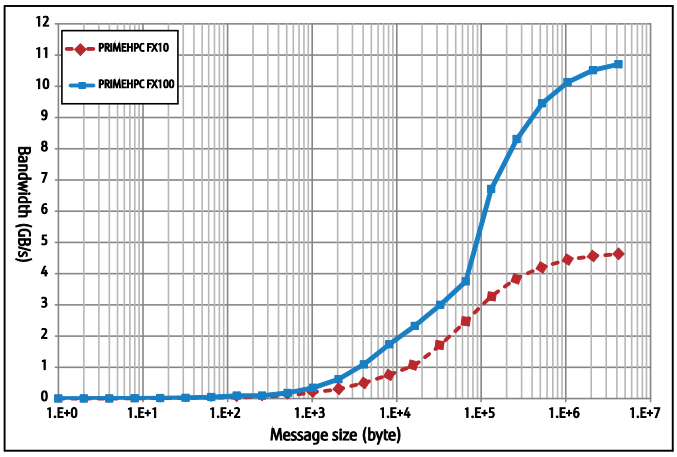


Figure 25 Point-to-point communication performance (bandwidth)

Effect on performance by HPC-ACE2

HPC-ACE2 (High Performance Computing - Arithmetic Computational Extensions 2) is an instruction set leading the way to HPC in the many-core era. The HPC-ACE2 added to the PRIMEHPC FX100 enhances the HPC-specific instructions implemented in the PRIMEHPC FX10. These new instructions are automatically generated by the compiler, enabling high-speed processing of applications. The other additions to the PRIMEHPC FX100 include an enhancement from 2-wide SIMD[(*3)] to 4-wide SIMD, a new instruction to convert the integer type to SIMD, and a new instruction to execute eight computations of a single-precision floating-point operation simultaneously. The compiler generates these new instructions automatically, enabling high-speed processing in the execution module. The following newly enhanced instructions in addition to the trigonometric function auxiliary instruction and reciprocal approximation instruction (division, SQRT) implemented in the HPC-ACE can improve parallelization at the

instruction level, enabling high-speed processing of applications whose performance used to be difficult to improve:

- Exponential function auxiliary instruction
- Indirect load store instruction (including those with a mask)
- Strided load/store instruction
- Broadcast load instruction

Figure 26 shows the performance improvement due to conversion of the compiler to SIMD. By implementing 4-wide SIMD, the system is now about 2.6 times faster. In Figure 27, with conversion of loops including an indirect load instruction to SIMD and use of the scheduling instruction, an indirect access program whose performance used to be difficult to improve is about 1.6 times faster.

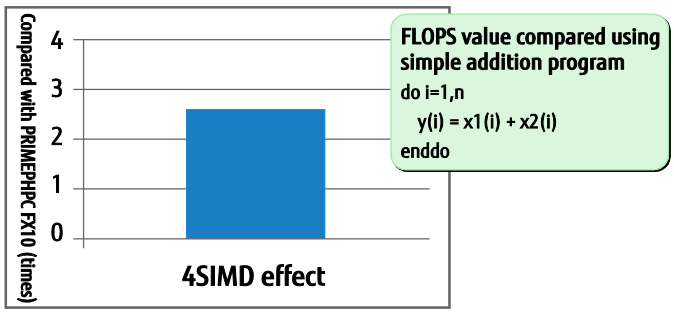


Figure 26 4SIMD effect for simple addition

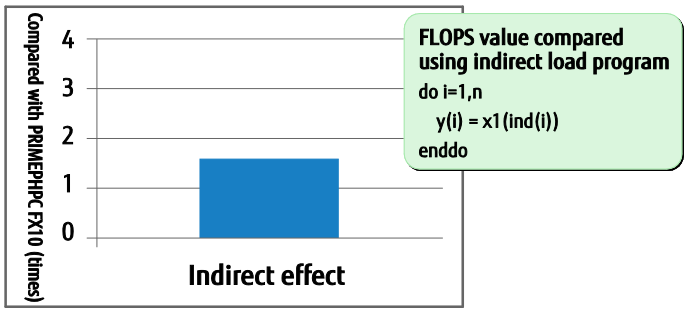


Figure 27 SIMD conversion effect by indirect instruction

Support technology that achieves super parallel processing

■ Enhanced application range of automatic parallelization

The automatic parallelization function has an enhanced code analysis capability, and parallelization can be applied automatically against a complex loop structure. As shown in Figure 28, the number of parallelization targets in ANL vectorization contest(*4) has increased significantly.

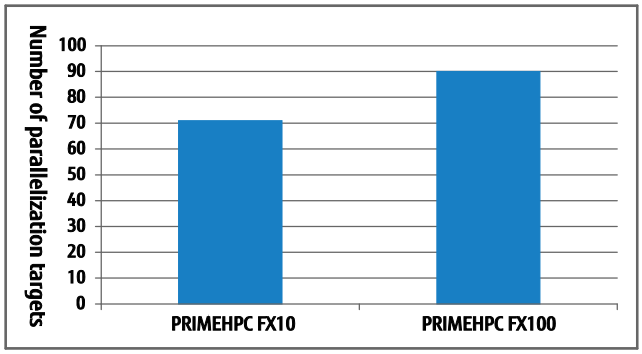


Figure 28 Enhancement of automatic parallelization

■ User-friendly application development product

- Visualization of massively parallel applications

In the tuning of highly parallel programs, sequential performance and high parallelization performance must be treated together. The PRIMEHPC FX100 provides tools to collect the appropriate tuning information, supporting industry-standard interfaces to work smoothly with the ISV tools familiar to many users. As shown in Figure 29, the provided set of tuning tools obtains information from all layers, including the hardware, OS, library, and application layers. Industry standard tools such as mpipP/TAU (tuning tool) will also be supported as ISV tools in stages.

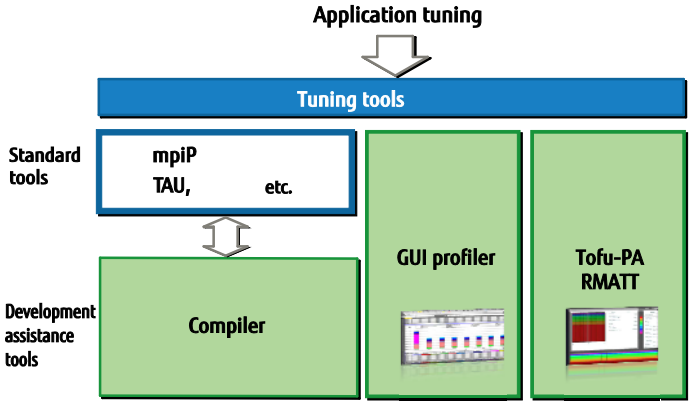


Figure 29 Stack configuration of tuning tools

To understand massive parallelism, it is important to understand the behavior of each process. The PRIMEHPC FX100 tuning tools are equipped with functions that can collect PA information (hardware monitoring information provided by the CPU) for each process and graphically display the collected information. Using these functions, you can easily understand the states of processes and take appropriate action. The following figure shows the elapsed runtime of 4,096 processes (16 x 16 x 16) of a program. The colors of the displayed processes depend on the time taken for execution: red, yellow, and blue indicate relatively long, moderate, and short times, respectively. The graph shows the relative length of time taken by individual processes, so you can see the performance balance between processes. One grid cell represents one process. To view information on a process, place the cursor on the corresponding cell.

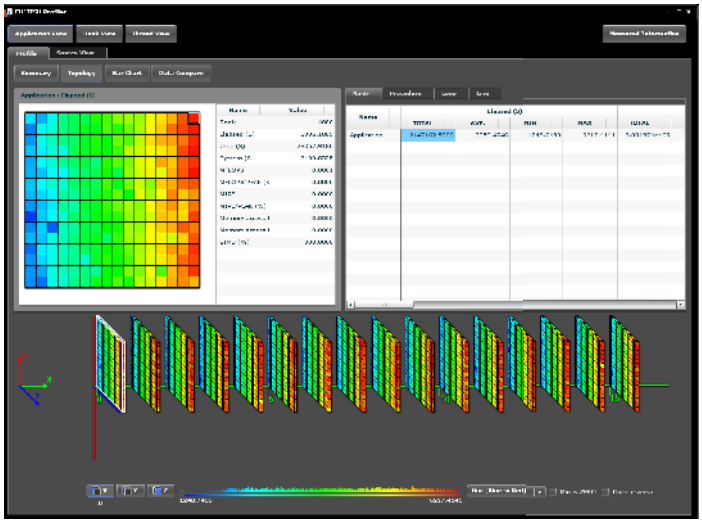


Figure 30 Visualization example with profiler GUI

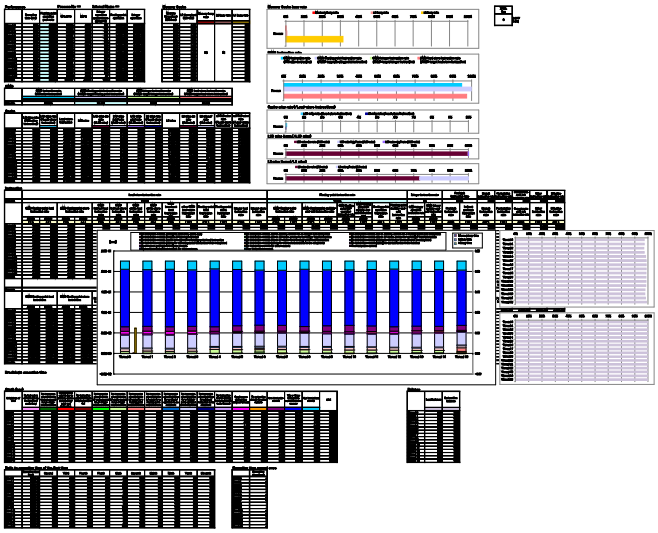


Figure 31 Detailed PA information by core

- Compliance with standards

The following table shows the compliance to the latest standards to support leading-edge applications.

Table 4 Supported standards

Language	Support standard specification
Fortran	ISO/IEC 1539-1:2010 (Fortran 2008 standard) ISO/IEC 1539-1:2004, JIS X 3001-1:2009 (Fortran 2003 standard) ISO/IEC 1539-1:1997, JIS X 3001-1:1998 (Fortran 95 standard) Fortran 90 standard and FORTRAN77 standard
C	ISO/IEC 9899:2011 (C11 standard) ISO/IEC 9899:1999 (C99 standard) ISO/IEC 9899:1990 (C89 standard) Enhanced specification of GNU compiler is also supported.
C++	ISO/IEC 14882:2011 (C++11 standard) ISO/IEC 14882:2003 (C++03 standard) Enhanced specification of GNU compiler is also supported.
OpenMP	OpenMP API Version 3.1
MPI	Message-Passing Interface Standard Version 3.0

- Parallel programming language, XPFortran

XPFortran (XPF) is a Fortran extension for distributed memory computers based on the data parallel model. XPF is defined by comment-style directives that support Fortran statements, providing data distribution arrangements, parallel execution of computing processes, and communication and synchronization functions. With XPF, step-by-step parallelization from a sequential program based on the data parallel model is possible, so parallel programs can be so developed and maintained more easily than with MPI.

*3 SIMD (Single Instruction Multiple Data) is a technology for multiple data computing by a single instruction.

*4 ANL vectorization contest is a benchmark developed by Argonne National Laboratory to assess the parallelization capability of parallel computers.

Application Fields

Supercomputer application fields

The application fields for supercomputers are growing wider because of improvements in hardware performance and advances in applications. Supercomputers have typically been used in fields that require massive computer processing power, such as for simulations.

Today, supercomputers are used in a variety of fields by government and university research institutes and private companies. Here are a few examples.

- **Basic research (pursuit of truth) and national strategic research**
  The central players in this category are the public agencies, universities, and other research institutes that use supercomputers in pursuit of the true nature of the universe, matter, life, etc. to understand universal laws in the natural world. Supercomputers are also used for research critical to national strategies, such as climate research (to address food problems and prevent disasters) and research on nuclear and fusion energy (to address energy problems). Many research groups develop and run their original simulation programs on supercomputers. Their work has produced great successes not achievable from theories and experiments alone.

- **Research for product development and economic forecasting**
  Private-sector businesses promote the use of supercomputers to increase the efficiency of product development (such as for industrial goods, medicines, and service products) and enhance competitiveness. For example, when supercomputers instead of numerous experiments are used to verify functionality and quality in the product design phase, they can greatly cut the time and cost of the verification, shortening the product development cycle.

The impressive new additions to the PRIMEHPC FX100 include HPC-ACE2 for ultra high-speed computing and an HMC with a high memory bandwidth. The PRIMEHPC FX100 is also equipped with the VISIMPACT, supporting the hybrid parallelization effective for massively parallel support, and the Tofu interconnect 2, which can be scaled up to 100,000 nodes. Simulation programs run at high speeds in massively parallel processing on the PRIMEHPC FX100, adopting highly parallelized algorithms and math calculation schemes. Improvements in both scientific technology and the competitive strength of companies are expected from the resulting breakthroughs.

Expected application fields of the PRIMEHPC FX100

The following table lists the major fields where we expect the PRIMEHPC FX100 to be applied.

Table 5 Expected PRIMEHPC FX100 application fields

Field	Representative application examples
Life science, medical care, and drug design	Elucidation of organ and cell functions, research on the interaction between proteins and drugs, virus molecular science, new medicine development, etc.
New materials and energy	New materials development, exploration of petroleum and other underground resources, nuclear safety analysis, fusion energy research, etc.
Disaster prevention and mitigation, and forecasting of global environmental changes	Weather forecasting, prediction of seismic wave propagation, forecasting of climate condition changes (disasters) and marine ecology changes (fishery resources) due to global warming, etc.
Next-generation manufacturing	Crash safety analysis of automobiles, aerodynamic characteristics analysis of airplanes and other aircraft, analysis of radio wave propagation of electronic equipment, etc.
Matter, and origin of universe	Research on elementary particles and atomic nuclei, exploration to reveal origins of astronomical bodies, etc.
Economics, finance, etc.	Economic model analysis, portfolio analysis, etc.

In the future, supercomputer applications will likely not be limited to individual fields but extend across various fields, like the following applications:

- Development of lightweight, high-strength materials for airplanes, automobile bodies, etc. (in the fields of new materials and manufacturing)
- New energy research and energy plant design focusing on safety and security (in the fields of energy, disaster prevention, and manufacturing)

Supercomputer simulations are becoming more and more important as a third research and development technique comparable to experiments and theories. By leveraging the power of supercomputers, the PRIMEHPC FX100 can strengthen the foundations of science and technology, be a source of innovation, and increase the power of companies to compete in fields of industry.

Reference
For more information about the PRIMEHPC FX100, contact our sales personnel or visit the following website:

http://www.fujitsu.com/global/products/computing/servers/supercomputer/primehpc-fx100/

Advanced Software for the FUJITSU Supercomputer PRIMEHPC FX100
Fujitsu Limited
November 17, 2014, Second Edition
2014-11-17-EN