FUJITSU

FUJITSU Software
Technical Computing Suite V4.0L20

Job Operation Software Overview

J2UL-2533-01ENZ0(02)
August 2021

Preface

Purpose of This Manual

This manual provides a functional overview and explains the terminology of Job Operation Software in Technical Computing Suite.

Intended Readers

This manual is intended for all the users who use Job Operation Software.

Organization of This Manual

This manual is organized as follows.

Chapter 1 Overview

This chapter provides an overview of the Technical Computing Suite.

Chapter 2 Job Operation Software

This chapter describes the Job Operation Software.

Chapter 3 Related Software

This chapter describes the software related to the Job Operation Software.

Appendix A Manual List

This appendix lists the Job Operation Software manuals.

Appendix B FX Server-specific Management Structure

This appendix describes the hardware configuration of a computer (FX server) with a mounted Fujitsu CPU A64FX.

Notation Used in This Manual

Notation of model names

In this manual, the computer that based on Fujitsu A64FX CPU is abbreviated as "FX server", and FUJITSU server PRIMERGY as "PRIMERGY server" (or simply "PRIMERGY").
Also, specifications of some of the functions described in the manual are different depending on the target model. In the description of such a function, the target model is represented by its abbreviation as follows:
[FX]: The description applies to FX servers.
[PG]: The description applies to PRIMERGY servers.

Symbols in this manual

This manual uses the following symbols.

Note

The Note symbol indicates an item requiring special care. Be sure to read these items.

See

The See symbol indicates the written reference source of detailed information.

Information

The Information symbol indicates a reference note related to Job Operation Software.

Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Trademarks

- Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.
- All other trademarks are the property of their respective owners.

Date of publication and Version

Version	Manual Code
August 2021, Version 1.2	J2UL-2533-01ENZ0(02)
March 2021, Version 1.1	J2UL-2533-01ENZ0(01)
February 2020, First version	J2UL-2533-01ENZ0(00)

Copyright

Update history

Changes	Location	Version
Changed the URL for McKernel information.	2.1.2	1.2
Fixed errata.	-	1.1

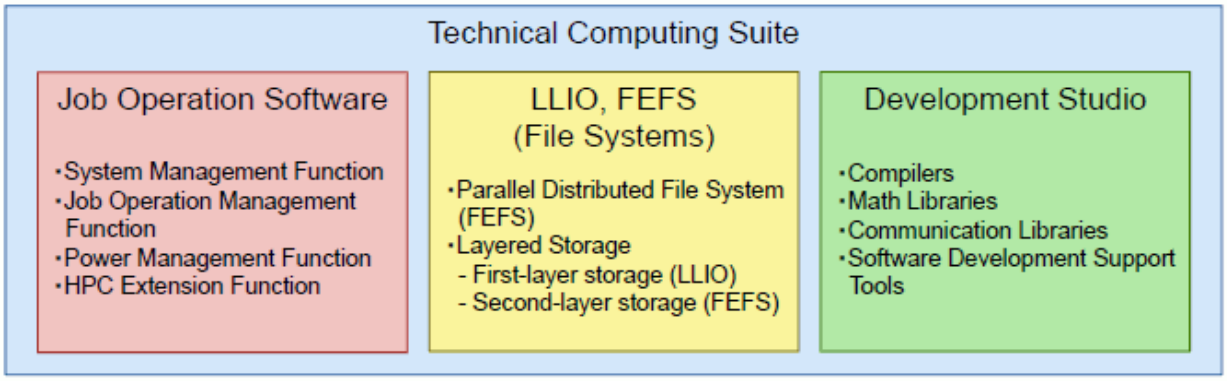
Contents

Chapter 1 Overview

Technical Computing Suite is an HPC middleware product that provides operational functions for large-scale computer systems, including supercomputers, and an environment to use applications.

Technical Computing Suite consists of the following software.

Figure 1.1 Software Configuration of Technical Computing Suite



Job Operation Software

Job Operation Software is a bundle of infrastructure software for managing a large-scale computer system and for managing and controlling application execution. The Job Operation Software has the following functions.

- System management function

  This function provides a form of layered management and a centralized operational view of the computers (node groups) in the system.

- Job operation management function

  This function manages applications and controls their execution in units called jobs.

- Power management function

  This function limits system power consumption, enabling power-saving operation that reduces unnecessary power consumption.

- HPC extensions

  This function provides drivers and libraries for individual Technical Computing Suite functions to use the FX server.

LLIO and FEFS

LLIO and FEFS provide the following two file systems.

- LLIO

  LLIO(Lightweight Layered IO-Accelerator) is a high-performance file system using high-speed Flash memory. LLIO is accessible from compute nodes executing applications.

- FEFS

  FEFS (Fujitsu Exabyte File System) is a scalable network file system that enables high-speed parallel distributed processing based on Lustre technology, an open source file system. FEFS is used as a shared file system in a computer system.

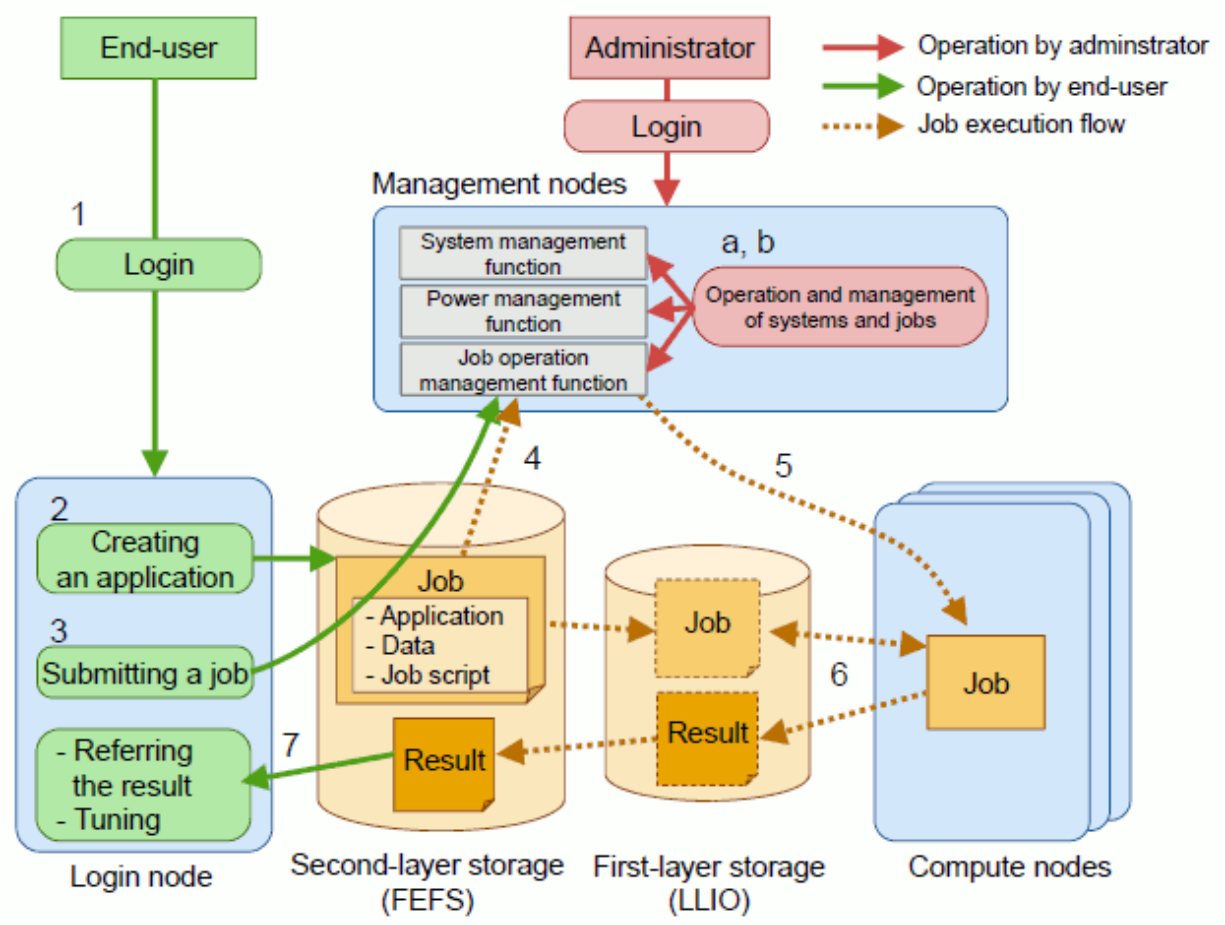
High-speed and high-capacity layered storage is realized from the combination of LLIO at the higher layer and FEFS at the lower layer as viewed from compute nodes. In the layered storage, LLIO is called first-layer storage, and FEFS is called second-layer storage.

Development Studio

Development Studio is an integrated software bundle supporting development (compile, debug, tuning, etc.) and execution of scientific computation programs written in Fortran, C, or C++.
The software supports parallelization technologies such as automatic parallelization, OpenMP, and MPI (Message Passing Interface).

The following figure is a use case diagram of Technical Computing Suite.

Figure 1.2 Use Case Diagram



[End User]

The end user creates an application and executes it on the system.

1. The end user logs in to a node (login node) to create and execute an application. The node is an entry point to using the system.
2. The end user creates an application in the development environment with a compiler, debugger, etc. provided by Development Studio.
   The end user places the created application, the necessary data for its execution, and the shell script (job script) with a written procedure to execute the application on second-layer storage (FEFS) as an execution unit called a job.
3. The end user requests job execution by using a job operation management function command. This is called job submission. An end user who has logged in directly to a compute node cannot execute the job.
4. Information on the job submitted by the job operation management function command is sent to the job operation management function for batch processing.
5. Based on the quantity of computer resources allocated to jobs (node, memory in a node, CPU time, etc.), execution priority, and other factors, the job operation management function schedules the execution order of multiple jobs to execute the jobs.
6. The compute node accesses files (job script, application, and files required for job execution) on second-layer storage (FEFS) via first-layer storage (LLIO).
   The job outputs the results (file), which are output to second-layer storage via first-layer storage. The login node can reference the results.
7. When the job ends, files containing the standard output contents and standard error output contents of the job are created on the login node. If an error occurs during job execution, details are provided by e-mail notification.
   As needed, the end user references the job execution results to tune the application.

Information

The end user can also develop programs, submit jobs, and check job execution results through a GUI by using the programming development support software provided by Development Studio.

[Administrator]

The administrator logs in to a node (system management node or compute cluster management node) to manage the system. This node can centralize the following work related to system operation:

a. System operation and management

- Build nodes (install software)
- Start or stop nodes
- Monitor the system operating status
- Perform software maintenance (backup/restore, apply fixes)
- Collect investigation materials for troubleshooting

b. Job operation and management

- Configure job operation
- Monitor and manipulate job operation
- Control job execution

Chapter 2 Job Operation Software

This chapter describes the Job Operation Software of Technical Computing Suite.

In the scientific computation area, parallel processing is a means used with a large number of computers to achieve high performance.

Such a high-performance system has the following requirements.

- The system responds to requests from numerous users by efficiently allocating an enormous number of computer resources and raising the utilization rate.
- The system as a whole can continue operation even when a partial failure occurs.
- The administrator can easily manage the entire system even if the number of computers increases and the configuration becomes complicated.
- A user can check to see the power consumption status of the entire system to save energy.
- Numerous end users can easily use the system to execute programs.

The Job Operation Software is infrastructure software for managing the large-scale computer system described above and for managing and controlling program execution on this system.

2.1 Job Operation Software Functions

The Job Operation Software consists of the following functions:

- System management function
- Job operation management function
- Power management function
- HPC extension function

The following sections introduce the respective functions.

2.1.1 System Management Function

The system management function is a function for administrators. For efficient operation in a large-scale system, this function provides a form of layered management and a centralized operational view of the computers (node groups) in the system.

Configuration management function

The configuration management function is a function for managing the nodes and networks in the system. The nodes composing the system are grouped into units called clusters and node groups (see "2.2 Job Operation Software Management Structure"). The function can also manage equipment like disk drives and network switches as system components.

System control function

The system control function is a function for node power control (start and stop). The function can collectively control node power in units of groups, such as for each cluster or the whole system, instead of in units of nodes. The function can also control power by taking into account the node starting/stopping order.

System monitoring function

The system monitoring function is a function for monitoring the hardware and software operating status. Upon detecting an abnormality, the function notifies the administrator and automatically isolates the abnormal node from job operation. Besides the Job Operation Software services, services specified by the administrator can also be managed with the monitoring of the software operating status.

System maintenance function

The system maintenance function is a support function for hardware and software maintenance. By using this function, the administrator can isolate the maintenance target node from operation so that no job is allocated to it during maintenance. If the node is in a redundant configuration, maintenance work can be done using failover to switch between the active and standby nodes so that operation is not affected. Working together with the system monitoring function, this function also enables automatic node failover.

Operational support function

The operational support function is a function supporting the operation and management of multiple nodes. By using this function, the administrator can execute a command, batch distribute a file, and batch collect files on multiple nodes, and establish a connection from one node to each node console in the system. The function can also manage the dump files on each node. The files are required in troubleshooting.

Log management function

The log management function is a function for batch collecting logs and other materials required in troubleshooting, and for monitoring the contents of logs. Although the administrator may be creating a configuration file on each node for this operation, a support function is also provided to make it easy to create the file.

Software environment check function

The software environment check function is a function for checking the Job Operation Software settings and software package application status on multiple nodes. By using this function when applying a software package to newly added nodes or during maintenance, the administrator can check whether the expected settings and the software package have been applied to multiple nodes.

Install function

The install function is a function for efficiently installing an operating system on the nodes composing the system. This function manages the operating systems and packages in a repository. The install function also works together with ServerView Suite, which is the system integration management tool for the PRIMERGY server. The administrator can install an operating system on the PRIMERGY server without taking ServerView Suite into account.

Backup/Restore function

The backup/restore function is a function for backing up the disk drive contents of a node as a disk image and for restoring a disk image. The function can replicate and build a disk image from one node to another node. The function can also restore a node to a previous state by restoring a backup disk image when a problem occurs, such as when a hard disk fails or when a fix package is applied.

See

For details on the system management function, see "Job Operation Software Administrator's Guide for System Management." For details on installation and maintenance, see "Job Operation Software Setup Guide" and "Job Operation Software Administrator's Guide for Maintenance."

2.1.2 Job Operation Management Function

The job operation management function uses computer resources efficiently in application execution to bring out the maximum system performance.

Job manager function

The Job Operation Software manages applications in units called jobs and controls their execution. Instead of directly executing a job on a compute node, the end user requests the Job Operation Software to execute the job. The job manager function controls job reception, job status management, and job execution.

Job scheduler function

The jobs received by the job manager function are not executed immediately but temporarily queued instead. Then, the job scheduler function schedules the order of execution. The job scheduler function determines the job execution order based on job priority, available computer resources (node and memory and CPU in a node), and other factors in order to efficiently execute multiple jobs with limited computer resources.

Job resource manager function

The job resource manager function ensures that a job can exclusively use a computer resource (memory or CPU) for a certain period. This capability brings out the maximum computer performance by eliminating contention of computer resources used by jobs and processes other than jobs (OS daemon, etc.).

Parallel execution environment

The parallel execution environment is a mechanism to control the processes of a parallel program that uses multiple nodes.

Job execution environment

The job operation management function can switch the job execution environment. This supports job execution on a Docker container in addition to the host Linux environment. The job execution environment on a Docker container can execute a job in a software environment appropriate to the job (such as a specific OS version) without relying on the system software environment to execute the job.
The job operation management function enables job execution using McKernel (*), a lightweight OS aimed at improving HPC application performance.
(*) https://ihkmckernel.readthedocs.io

API for customizing the job operation management function

The job operation management function provides the command API so that the administrator and end user can create job operation management function commands with their own command interfaces. The administrator provides interfaces (hook function, etc.) to control job execution according to the job operation policy or acquire information.

See

For details on how to execute a job and how to use the job operation management function, see "Job Operation Software End-user's Guide" and "Job Operation Software Administrator's Guide for Job Management." For details on each API, see the user's guide prepared for the API. For the manual list, see "Appendix A Manual List."

2.1.3 Power Management Function

The power management function limits system power consumption, enabling power-saving operation that reduces unnecessary power consumption.

System power data collection function

This function collects information on the power consumption of compute nodes and other equipment (external equipment) required for system operation. The function also provides the administrator with commands for displaying the power consumption information and an interface (system power data collection support API) for acquiring this information from applications.

Power saving function

This function works together with the job operation management function to automatically power off nodes that have no job execution scheduled for a long time. Also, some nodes may for a short time not be executing any jobs. These nodes enter a low power consumption mode for hardware. The nodes recover from the power-off state or low power consumption mode when job execution begins. Such control can reduce wasteful power consumption in the system.

Power API

Sandia National Laboratories provides the prescribed and promoted Power API (http://powerapi.sandia.gov/). From an application created by the administrator or end user, the Power API can measure and control power consumption in units of CPUs or memory.

Capping function

This function schedules jobs so that the system power consumption does not exceed the upper limit. The FX server also controls the CPU frequency and power down of the compute nodes in units of BoB (Bunch of Blades, See "B.1 FX Server Hardware Components") to prevent sudden increases in system power consumption.

See

For details on the power management function, see "Job Operation Software Administrator's Guide for Power Management." For details on the Power API function, see "Job Operation Software API User's Guide for Power API."

2.1.4 HPC Extension Function

The HPC extension function provides various extended drivers/libraries for the FX server to use standard Linux functions.

TofuD driver

The TofuD driver is a driver provided to enable use of the Tofu interconnect D (called "Tofu interconnect" below), which is hardware for interconnect connections to the FX server.
Jobs can use the Tofu interconnect through the job operation management function without taking into account the TofuD driver.

HPC tag address override control function

This function controls the processor-specific HPC tag address override function of the FX server.
This driver supports not only performance tuning of applications created with the related software Development Studio but also acquisition of hardware events by a profiler. For details, see "Job Operation Software End-user's Guide for HPC Extensions."

Power control driver/library

This driver/library for the FX server conforms to the Power API, which is an API for power measurement and control. The unique driver and library provided make it possible to use the Power API.
Not just for use by the power control function of the Job Operation Software, this driver/library is also publicly available for users who create applications.
For details on how to use the Power API with the FX server, see "Job Operation Software API user's Guide for Power API."

Inter-core hardware barrier driver/library

This driver/library provides a driver and library to support the inter-core hardware barrier function on the FX server.
The inter-core hardware barrier function is a function for high-speed synchronization between the threads of a thread-parallelized application program. This function can be used by applications created with Development Studio. For details, read the Development Studio manuals.

Sector cache driver/library

This driver/library provides a driver and library to support the sector cache function on the FX server.
The sector cache function is a function that improves the operating speed of applications by constantly keeping as much as possible highly reusable data in cache. This function can be used by applications created with Development Studio. For details, read the Development Studio manuals.

Large page library

The FX server uses Huge Pages, a standard feature in Linux. The two types of huge pages in Linux are THP (Transparent Huge Page) and HugeTLBfs. The FX server adopts the use of HugeTLBfs due to its memory usage efficiency, the range of memory area covered by enabled Huge Pages, guaranteed acquisition of huge pages, etc.
The large page library provided by the HPC extension function extends the functionality for the FX server so that HugeTLBfs can be used more efficiently and also with higher extensibility. Furthermore, a tool is provided to collect the status of Huge Pages memory usage.
Users can create applications using huge pages by linking this library in Technical Computing Suite Development Studio. Users can also change the operation of the large page library by using an environment variable in a job script when executing a job. For details, see "Job Operation Software End-user's Guide for HPC Extensions."

Information

The Job Operation Software refers to a huge page in Linux as a "large page" because it has a larger size than a normal page. The reference manuals use the term "large page" without any special notice.

The HPC extension function plays the following roles in the Job Operation Software, enabling efficient operation of a large-scale computing system without the administrator paying attention to the function.

Fast restart

Reducing the FX server restart time improves the system utilization rate.

Dump generation management

Proper control of the number of resources (memory dumps) for maintenance of a large-scale computing system like the FX server makes it possible to efficiently use compute node resources (disk capacity).

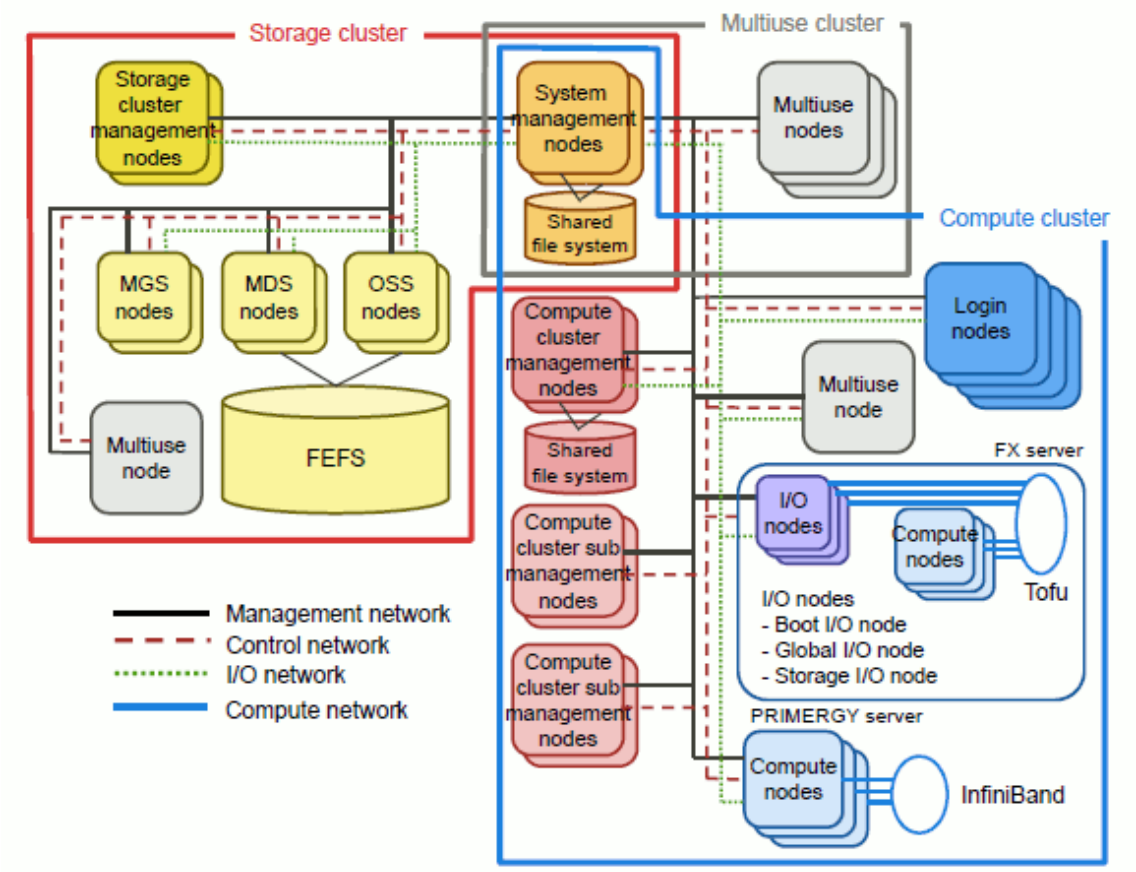
2.2 Job Operation Software Management Structure

This section describes the structure of system management by the Job Operation Software. The administrator, in particular, needs to understand it for system control, system management, and job operation.

2.2.1 System Configuration

The system with the Job Operation Software installed has the following configuration.

Figure 2.1 System Configuration Diagram



The following sections describe each component.

2.2.2 Nodes

The following table shows the node types determined for nodes according to their roles in the system with the Job Operation Software installed.

Table 2.1 Node Types

Node Type	Abbreviation	Role
Login node	LN	Node used by the end user to create applications and request the Job Operation Software to execute jobs. (LN: Login node)
System management node	SMM	Node for controlling power to start/stop a cluster and monitoring the nodes and services in the cluster. (SMM: System Management node) Normally, a compute cluster, storage cluster, and multiuse cluster are interdependent, and they share the same system management node. The Job Operation Software supports the redundant configuration of the system

Node Type	Abbreviation	Role
		management node. The redundant configuration requires a shared file system for takeover of files between the active and standby system management nodes.
Compute cluster management node	CCM	Node that manages information relating to job operation in a compute cluster. (CCM: Compute Cluster Management node) After the end user submits a job from the login node, the compute cluster management node receives the job and schedules its execution. The Job Operation Software supports the redundant configuration of the compute cluster management node. The redundant configuration requires a shared file system for takeover of job operation information between the active and standby compute cluster management nodes.
Compute cluster sub management node	CCS	Node for reducing the load of service monitoring by the compute cluster management node. (CCS: Compute Cluster Sub management node) Rather than service monitoring directly by the compute cluster management node, the compute cluster sub management node monitors services. The Job Operation Software supports the redundant configuration of the compute cluster sub management node.
Boot I/O node [FX]	BIO	I/O node that acts as the boot server of nodes in the FX server. (BIO: Boot I/O node) Some compute nodes also serve the role of boot I/O node in the FX server. These nodes are referred to as a compute node that also serves as a boot I/O node (CN/BIO).
Global I/O node [FX]	GIO	Node for relaying input/output for second-layer storage in the FX server. (GIO: Global I/O node) Some compute nodes also serve the role of global I/O node in the FX server. These nodes are referred to as a compute node that also serves as a global I/O node (CN/GIO). If a global I/O node fails, another global I/O node in the same rack handles the degraded operation.
Storage I/O node [FX]	SIO	I/O node responsible for input/output for first-layer storage in the FX server. (SIO: Storage I/O node) The storage I/O node is connected to a disk drive (SSD) composing first-layer storage. Some compute nodes also serve the role of storage I/O node in the FX server. These nodes are referred to as a compute node that also serves as a storage I/O node (CN/SIO).
Compute node	CN	Node that runs jobs. (CN: Compute node)
Storage cluster management node	SCM	Node that manages the configuration and monitors the services in a storage cluster. (SCM: Storage Cluster Management node) The Job Operation Software supports the redundant configuration of the storage cluster management node. The storage cluster management node can also serve as the system management node. These nodes are referred to as a system management node that also serves as a storage cluster management node (SMM/SCM).
MGS node	MGS	Node for managing the file system configuration information. (MGS: Management Server) The Job Operation Software supports the redundant configuration of the MGS node.
MDS node	MDS	Node for storing and managing the FEFS metadata provided by a storage cluster. (MDS: Meta Data Server) The Job Operation Software supports the redundant configuration of the MDS node.
OSS node	OSS	Node for storing and managing file data for FEFS. (OSS: Object Storage Server)

Node Type	Abbreviation	Role
		The Job Operation Software supports the redundant configuration of the OSS node.
Multiuse node	Any	Node used for any purpose other than the above. The node is subject to status monitoring and power control by the Job Operation Software. The administrator can define an abbreviation when building a multiuse node.

For the system management node or compute cluster management node in a redundant configuration, a shared file system mounted only on the respective active node is required for using the information in logs and dump files.

Information

- There are combinations of node types that can be used together. For more information, see "Node Types for Multiple Purposes" in "Job Operation Software Setup Guide."
  The node type of any single node that serves as multiple node types may be written like "CN/BIO" in this manual.
- The FX server (boot I/O node, global I/O node, storage I/O node, and compute node) has a hardware-specific configuration. For details, see "Appendix B FX Server-specific Management Structure."

2.2.3 Clusters

The types of clusters are compute cluster, storage cluster, and multiuse cluster, which split the system into units based on the operation function.

2.2.3.1 Compute Cluster

A compute cluster consists of node groups for creating applications and executing the created applications in units called jobs.

The following units of a compute cluster manage the nodes in the cluster:

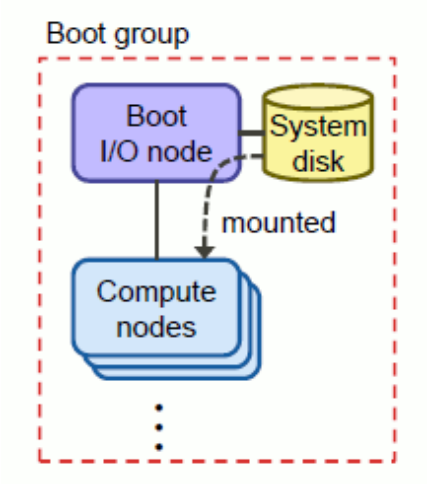
- Boot group [FX]
- Node group
- SIO group [FX]
- GIO group [FX]
- Resource unit
- Resource group

This section describes each of them.

Boot group [FX]

A boot group is equivalent to a BoB (Bunch of Blades) (see "B.1 FX Server Hardware Components"). It is the start unit for FX server nodes. The nodes in the boot group use the same boot I/O node as the boot server and mount the system disk.

Figure 2.2 Boot group

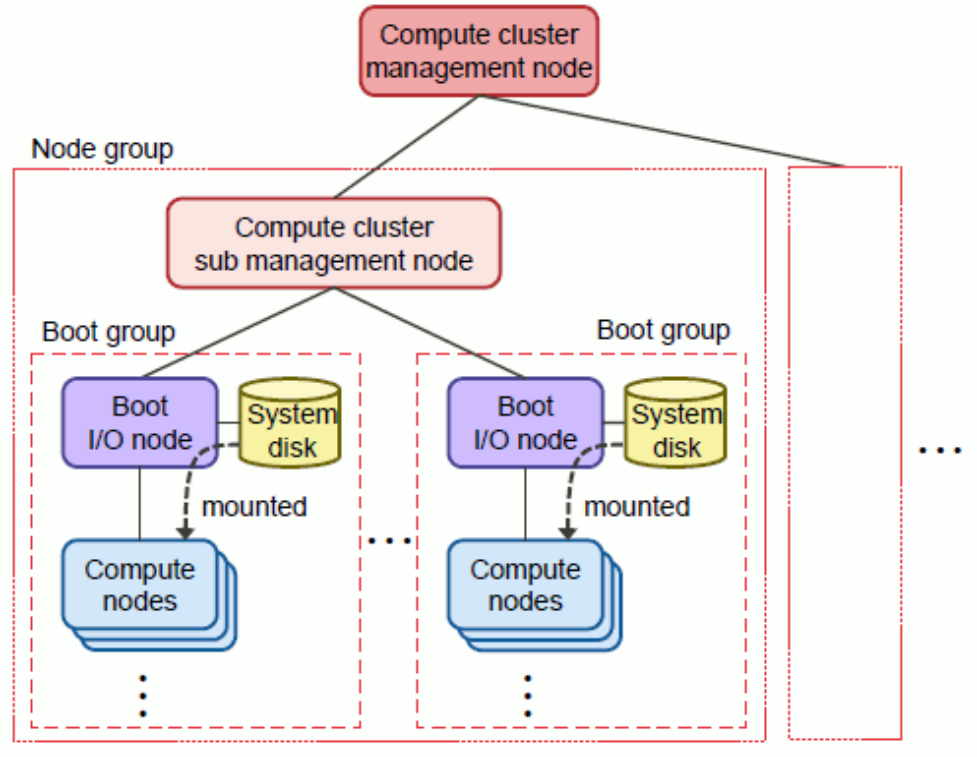


Node group

A large-scale system with many compute nodes has compute cluster sub management nodes placed under a compute cluster management node. Monitoring of the compute nodes is distributed to the compute cluster sub management nodes to reduce the load on the compute cluster management node.
The nodes monitored by one compute cluster sub management node are called a node group.

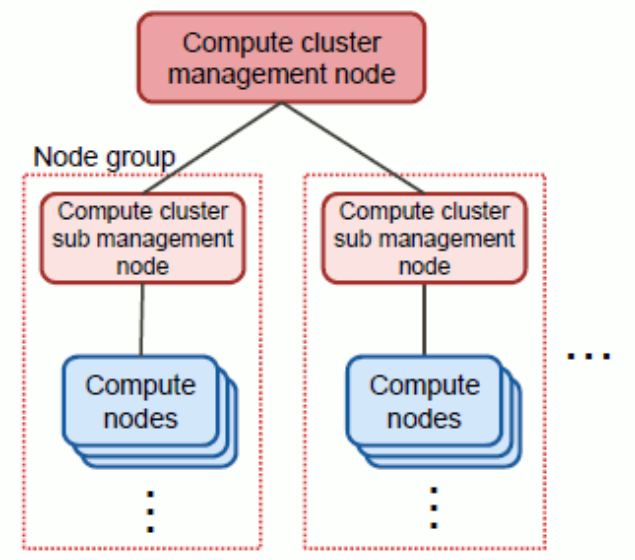
With the FX server, a node group is configured in the unit of a boot group.

Figure 2.3 Node Group Configuration (Where a Compute Node is the FX Server)



With the PRIMERGY server, a node group is in a configuration with a compute cluster sub management node and compute node.

Figure 2.4 Node Group Configuration (Where a Compute Node is the PRIMERGY Server)



Information

Guidelines for installation of compute cluster sub management nodes are listed below.

- With the FX server
  If the number of boot groups in a cluster exceeds 252, compute cluster sub management nodes are required. Also, the number of boot groups per node group should not exceed 252 (equivalent to 4,032 nodes).
- With the PRIMERGY server
  If the number of PRIMERGY servers in a cluster exceeds 1,024, compute cluster sub management nodes are required. Also, the number of PRIMERGY servers per node group should not exceed 1,024.

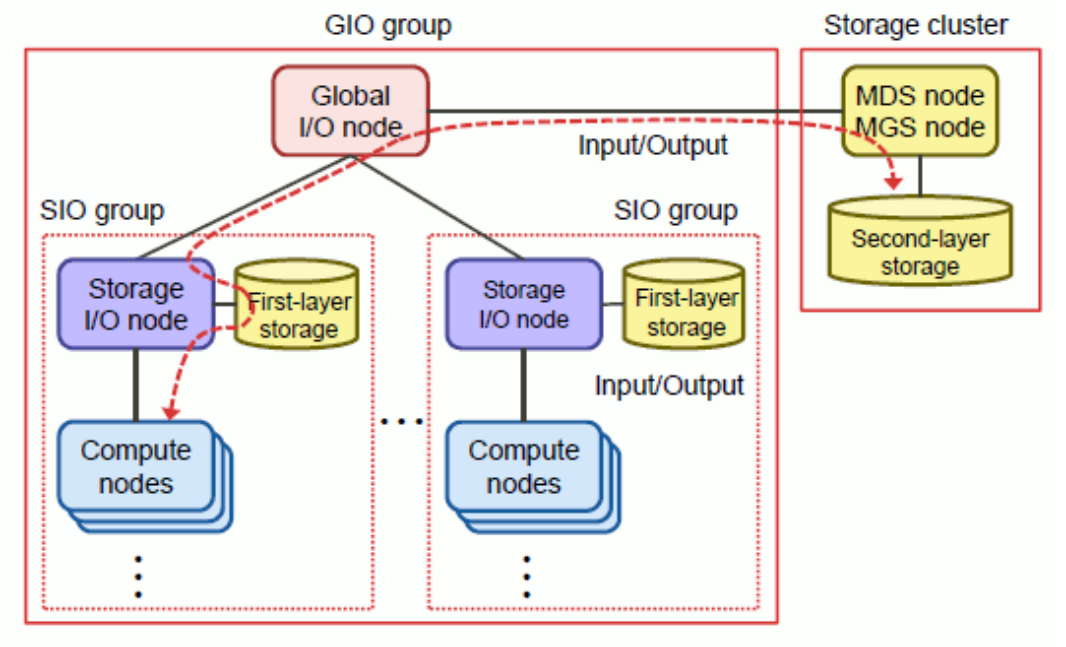
For details on how to obtain concrete estimates, see "Criteria for Cluster Configuration Estimates" in "Job Operation Software Setup Guide."

SIO group and GIO group [FX]

In the FX server, a compute node group for input/output on one storage I/O node and on first-layer storage using the storage I/O node as a relay node is called an SIO group.

Input/Output from a compute node to second-layer storage (FEFS) in the FX server passes through first-layer storage and a global I/O node. The global I/O nodes in one main unit rack together with the storage I/O nodes and compute node group that use the global I/O nodes to input/output data are called a GIO group. For details on the main unit rack, see "B.1 FX Server Hardware Components."

Figure 2.5 SIO Group and GIO Group Configuration

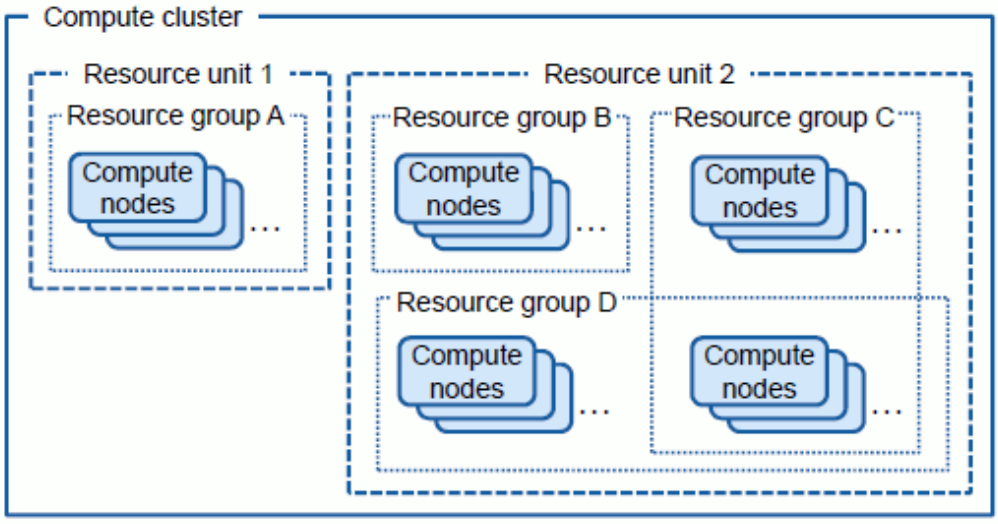


Resource unit and resource group

A resource unit is a unit of job operation. For example, a resource unit can be divided for use in individual activity classes with different operation policies for the job operation management function. The administrator can define any range of compute nodes as a resource unit provided that they are in a compute cluster. However, the compute nodes composing one resource unit must be of the same type.

A resource group is a unit of resources available to jobs. For example, the administrator can prepare multiple resource groups that differ in the maximum number of compute nodes available to jobs and the maximum job execution time. Then, by using the appropriate resource group according the job size and type, jobs can operate efficiently. The administrator can define any range of compute nodes as a resource group provided that they are nodes in a resource unit. A single compute node can also be configured to belong to multiple resource groups.

Figure 2.6 Resource Unit and Resource Group

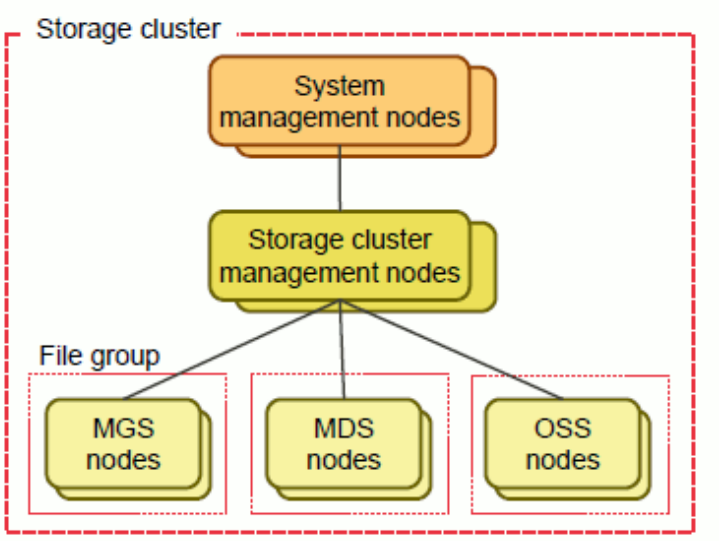


2.2.3.2 Storage Cluster

A storage cluster is a node group providing shared file system to compute clusters. The related software FEFS can use as the shared file system. In the layered storage, it is used as the second-layer storage.

In the storage cluster, the paired nodes in a redundant configuration of the MGS node, MDS node, or OSS node are called a file group.

Figure 2.7 Storage Cluster Management Structure



2.2.3.3 Multiuse Cluster

A multiuse cluster is a node group where power control and status monitoring should be independent of compute and storage clusters. The multiuse cluster can be used for any purpose: for example, to handle a node group acting as the LDAP server and NFS server.

2.2.3.4 Cluster and Node Types

The following table uses "Yes" to indicate which nodes can compose each of the clusters.

Table 2.2 Nodes that can compose a cluster

The node which can compose the cluster	Compute cluster	Storage cluster	Multiuse cluster
System management node	Yes (*)	Yes (*)	Yes (*)
Compute cluster management node	Yes		
Compute cluster sub management node	Yes		
Boot I/O node [FX]	Yes		
Global I/O node [FX]	Yes		
Storage I/O node [FX]	Yes		
Compute node	Yes		
Login node	Yes		
Multiuse node	Yes	Yes	Yes
Storage cluster management node		Yes	
MGS node		Yes	
MDS node		Yes	
OSS node		Yes	

(*) The system management node can be shared by each cluster.

2.2.4 Networks

The networks in the system are categorized by use in the same way as nodes. Categorization of the networks by use prevents system operation processing, such as status monitoring, from interfering with job execution performance.

The following table lists the four types of networks for the Job Operation Software.

Table 2.3 Types of networks

Name	Description
Control network	Network used to control the hardware of nodes (such as power control, notification of abnormalities). The system management node must be connected via this network to the control devices of each node in the cluster.
Management network	Network used to control services and communicate information related to the operations of this product. This network must be able to communicate with the OS of each node.
I/O network	High-speed network for using the shared file system provided by a storage cluster. The network is built with an interconnect that has a small transmission delay. The FEFS provided by the storage cluster and the nodes that input/output files must be able to communicate with each other.
Compute network	High-speed network used by parallel programs like MPI programs to communicate between nodes. Compute nodes must be able to communicate with one another over this network. The PRIMERGY server uses the management network. The FX server uses the Tofu network.

2.2.5 Structure Identifiers

The job operation software manages the aforementioned structures by attaching names or numerical values as their identifiers. Administrators need these identifiers in order to perform operations related to job operation management or system power control. For details on how to find out the identifier values required for various operations, see "Job Operation Software Administrator's Guide for System Management" and "Job Operation Software Administrator's Guide for Job Management."

Table 2.4 Structure identifiers in the job operation software

Identifier	Type	Description
Cluster name	Character string	Name attached to a cluster (compute cluster, storage cluster, and multiuse cluster). The administrator who designs the cluster decides the name.
Node ID	Numerical value	Numerical value automatically assigned to a node. The value is unique in the cluster. The job operation software identifies each node by node ID, not by host name.
Node group ID	Numerical value	Numerical value automatically assigned to a node group. The value is unique in the cluster.
Boot group ID [FX]	Numerical value	Numerical value automatically assigned to a boot group in the FX server. The value is unique in the cluster.
Resource unit name	Character string	Name attached to a resource unit. The name is unique in the range managed by one control node. The administrator who designs the resource unit decides the name.
Resource group name	Character string	Name attached to a resource group. The name is unique in the resource unit. The administrator who designs the resource group decides the name.

2.3 Administrator Types

In the Job Operation Software, administrators are users who have OS root privileges.

As the system scale increases, the system is split into units of activity classes, considering the form of operation used.

In these circumstances, administrator work may likewise be divided by role, like into operation management for the entire system and operation management for individual work units. Such work includes management of the system configuration and hardware.

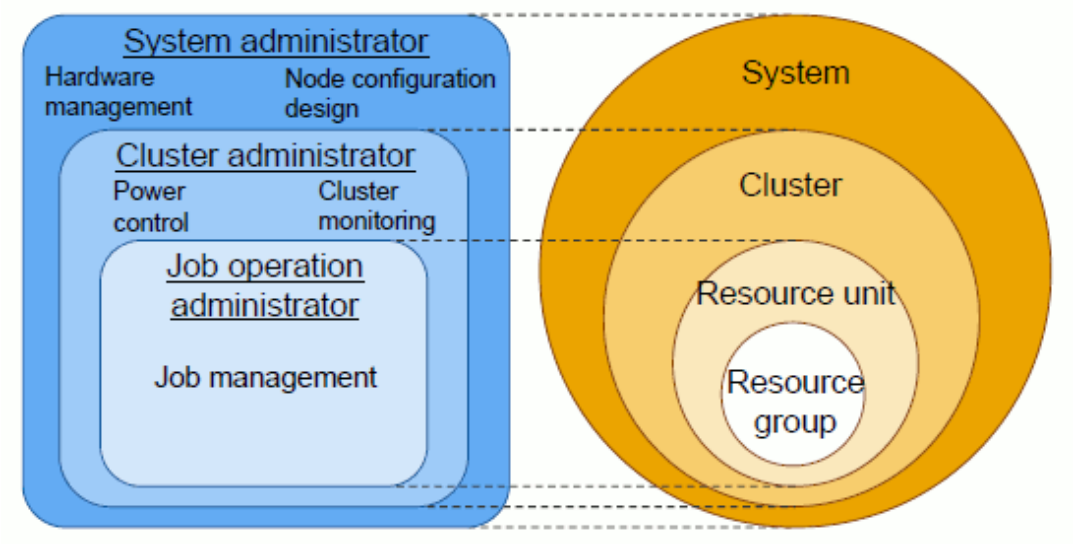
The Job Operation Software manuals refer to administrators as follows, taking into account the differences in operational roles.

Table 2.5 Administrator types in the job operation software

Administrator type	Range of responsibility	Role
System administrator	Whole system	Administrator who has the highest authority in using the job operation software. The administrator is permitted to use all functions, including those involving the work of other administrators. Mainly, the system

Administrator type	Range of responsibility	Role
		administrator is responsible for configuration design and hardware management of the entire system.
Cluster administrator	Within cluster	Administrator of a cluster, the largest operation unit in the job operation software. For example, suppose separate clusters are organized for business units. Each of these clusters would have a cluster administrator. The cluster administrator is in charge of operations in a cluster. The administrator's tasks include starting and stopping the cluster and nodes, and configuring the status monitoring of nodes and services. The administrator also configures job operations for the entire cluster.
Job operation administrator	Within resource unit	Administrator who manages a resource unit, which is the unit of a job operation. The administrator manages the job operation policy and computer resources required by the job.

Figure 2.8 Administrators' ranges of responsibilities



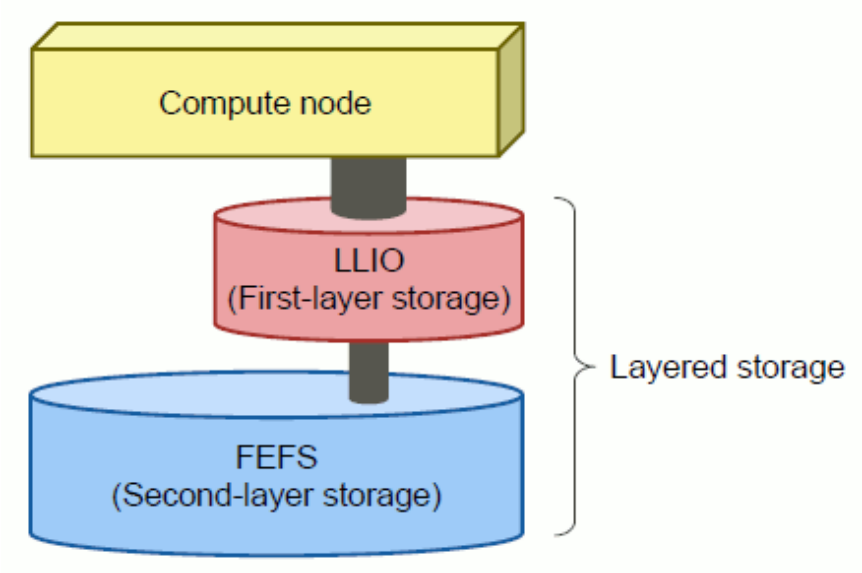
Chapter 3 Related Software

This chapter presents the software related to Job Operation Software.

3.1 LLIO, FEFS

Technical Computing Suite provides two file systems: LLIO and FEFS.
They are set in layered storage with LLIO as first-layer storage and FEFS as second-layer storage. Taking advantage of their individual characteristics, a high-speed and large-capacity file system is realized with the layered storage.

Figure 3.1 LLIO and FEFS



3.1.1 LLIO

LLIO is a file system achieving high performance. It sets a storage layer (first-layer storage) using high-speed Flash memory between FEFS and compute nodes and uses the layer as a cache area of FEFS and a temporary area for jobs. LLIO has the following features.

Constructing optimum LLIO areas for jobs

LLIO has three types of areas: cache area of second-layer storage, shared temporary area, and node temporary area. The end user can construct areas of the optimum size when submitting a job.

High-speed file access

LLIO implements the following functions to achieve high-speed file access:

- Striping function
- Shared file distribution function
- Compute node cache function

Statistical information

LLIO collects a lot of statistical information and provides it to the users executing jobs and the system administrator. This information is helpful in I/O tuning of jobs and investigation of system problems.

For details on use of LLIO, see the LLIO manual.

3.1.2 FEFS

FEFS is a large-scale, high-performance parallel distributed file system based on the technology of Lustre, an open-source file system. FEFS has the following features.

Large scale

FEFS supports 100,000 node clients and a file system size of 8 EiB ($8x2^{60}$).

High performance

FEFS improves I/O performance through distributed storage of file data, on storage using striping and round-robin techniques.

Easy to use

FEFS reduces the effect of large amounts of input/output by other users, through I/O priority control among clients, fair sharing among users (QoS function), etc.

High reliability

FEFS has MGS (management server), MDS (metadata server), and OSS (object storage server) failover functions.

Extensibility

FEFS enables dynamic extension of the metadata area/data storage area. With multi-MDS function support, FEFS also enables scalable performance improvement according to the MDS/MDT quantity.

For details on use of FEFS, see the FEFS manual.

3.2 Development Studio

Development Studio is a software that supports high-performance parallel programs, from development to execution, in Fortran, C, and C ++.

Development Studio has the following features.

- Supporting development of high-performance parallel programs
- Supporting efficient development of large-scale application programs
- Supporting development of highly portable programs

For more information, see the Development Studio manual.

3.2.1 Compilers

The Fortran compiler, C compiler, and C++ compiler can translate programs written in each language to create highly optimized executable programs that fully extract the CPU execution performance for the targeted compute nodes. These compilers can create thread-level parallelizable executables through automatic parallelization and the OpenMP specification.

Main functions

Each compiler features an automatic parallelization function. By just specifying a compile-time option for this function, the compiler automatically creates a program that performs thread-level parallel processing. In addition, each compiler supports the "OpenMP API specifications" in thread level parallel processing using directive lines specified in programs. The automatic parallelization function and the parallel processing function compliant with OpenMP specifications assume a shared memory system. These functions are effective on a compute node. When used in combination with an MPI library, the functions support a high-efficiency hybrid parallel programming model (thread parallelism + MPI process parallelism).

Optimization functions

Featuring the following optimization functions, each compiler can create object programs that can be executed at high speed on compute nodes. Each compiler also provides various optimization control lines to promote optimization in programs.

- Optimization by changing the configuration of nested loops
- Instruction scheduling function suited to the CPU processor characteristics
- Efficient use of cache using prefetch instructions
- Improve the parallelism level by SIMD utilizing SVE
- Reduce the number of times to save or restore register contents
- Optimization features that effectively utilize the HPC tag address override function of the A64FX processor

3.2.2 Math Libraries

Development Studio provides not only Fujitsu's original math libraries (SSL II and C-SSL II) that are widely used by R&D users in Japan but also linear algebra libraries (BLAS, LAPACK, and ScaLAPACK) developed in the U.S. It also provides a fast basic operations library for quadruple precision that represents quadruple precision values in double-double form and performs operations.

These math libraries are tuned for optimum execution performance on each and every compute node. For details on the math libraries, see the Development Studio manual.

3.2.3 Communication Libraries

MPI library

The MPI library conforms to the standards defined in the MPI forum. It supports a 6-dimensional mesh/torus interconnect called Tofu and realizes high performance and memory savings.

uTofu

uTofu is a low-level application programming interface (API) for communication through the Tofu interconnect by software in the user space. uTofu supports one-sided communication and barrier communication on the Tofu interconnect.

3.2.4 Software Development Support Tools

Profiler

The Profiler is a performance analysis tool that allows application programs written in Fortran, C, or C++ to obtain information needed for performance analysis. The profiler can also obtain profiler information for programs that support thread parallelism and MPI process parallelism.

The Profiler consists of the following features:.

- Instant Performance Profiler
- Advanced Performance Profiler
- CPU Performance Analysis Report

Debugger for Parallel Applications

The Debugger for Parallel Applications is a debugging tool for application programs that call MPI libraries written in Fortran, C, or C++ languages.

The Debugger for Parallel Applications consists of the following functions:

- Abnormal Termination Investigative Function
- Deadlock Investigative Function
- Duplication Removal Function
- Debugging Control Function with Command Files

IDE (Integrated Development Environment)

Eclipse, the most major and proven open source integrated development environment, is adopted as the IDE. Parallel Tools Platform, a plug-in for parallel program development, works with the job scheduler to submit jobs, check job status, and so on.

Appendix A Manual List

This appendix lists the Job Operation Software manuals.

Table A.1 Manual List

Manual Name	Outline	Intended Reader
Job Operation Software Overview	This document. This manual provides an overview of Job Operation Software and related software.	End-user and Administrator
Job Operation Software End-user's Guide	This manual describes how to execute applications (jobs) with Job Operation Software.	End-user
Job Operation Software End-user's Guide for HPC Extensions	This manual for the end user describes how to use the HPC extension function.	End-user
Job Operation Software End-user's Guide for Master-Worker Job	This manual describes how to create master-worker jobs.	End-user
Job Operation Software Setup Guide	This manual describes how to install Job Operation Software.	Administrator
Job Operation Software Administrator's Guide for System Management	This manual describes how to operate and manage a system with Job Operation Software installed.	Administrator
Job Operation Software Administrator's Guide for Job Management	This manual describes how to operate and manage jobs in a system with Job Operation Software installed.	Administrator
Job Operation Software Administrator's Guide for Job Operation Manager Hook	This manual describes how to use the hooks of the job operation management function.	Administrator
Job Operation Software Administrator's Guide for Power Management	This manual describes power management of a system with Job Operation Software installed.	Administrator
Job Operation Software Administrator's Guide for HPC Extensions	This manual for the administrator describes how to use the HPC extension function.	Administrator
Job Operation Software Administrator's Guide for Maintenance	This manual describes how to perform maintenance on a system with Job Operation Software installed.	Administrator
Job Operation Software API user's Guide for Command API	This manual describes how to use the command API of the job operation management function.	End-user and Administrator
Job Operation Software API user's Guide for Power API	This manual describes how to use the Power API of the power management function.	End-user
Job Operation Software API user's Guide for Job Information Notification API	This manual describes how to use the job information notification API of the job operation management function.	Administrator
Job Operation Software API user's Guide for Scheduler API	This manual describes how to use the scheduler API of the job operation management function.	Administrator
Job Operation Software Troubleshooting	This manual is a troubleshooting reference for the administrator during system installation and operation.	Administrator

Manual Name	Outline	Intended Reader
Job Operation Software Command Reference	This manual is a command reference manual and message reference for Job Operation Software.	End-user and Administrator
Job Operation Software Glossary	This manual explains terms related to Job Operation Software.	End-user and Administrator

Appendix B FX Server-specific Management Structure

This appendix provides an overview of the FX server hardware-specific management structure.

B.1 FX Server Hardware Components

The following diagram shows the hardware components of the FX server.

Figure B.1 FX Server Hardware Configuration (Outline)

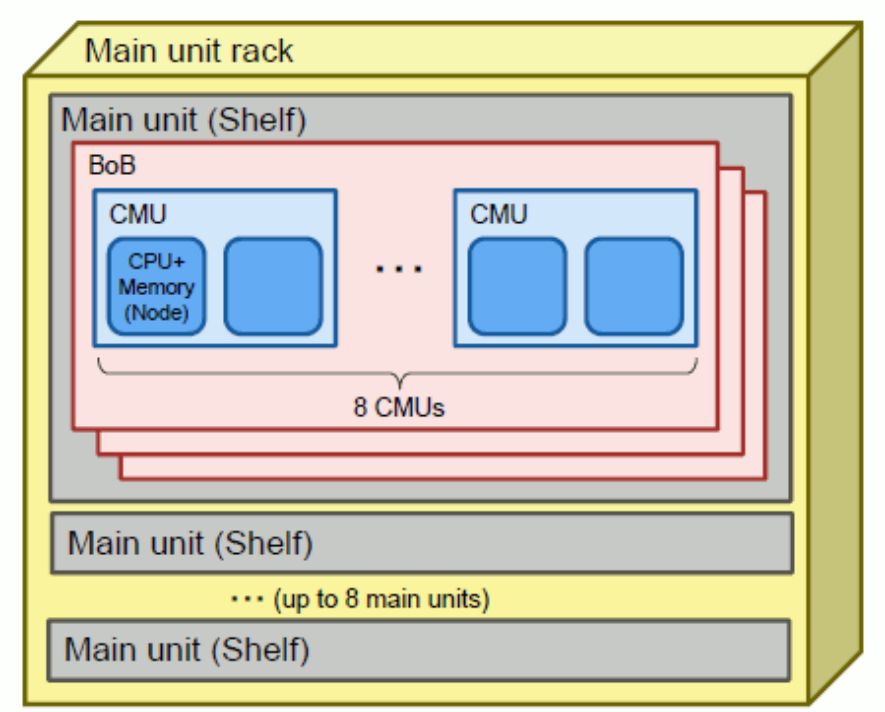


Table B.1 FX Server Hardware Components

Component	Description
CMU (CPU Memory Unit)	The CMU has a mounted CPU and memory. The unit is equivalent to 2 compute nodes.
BoB (Bunch of Blades)	The BoB is a unit of control for the FX server. The BoB consists of 8 CMUs. That is, 1 BoB contains 16 compute nodes. Among the compute nodes in a BoB, 3 nodes have input/output functions. These compute nodes also respectively serve as a boot I/O node, storage I/O node, and global I/O node. The number of I/O nodes in a BoB varies depending on the system. The boot I/O node is connected to the system disk to start each node. The storage I/O node is connected to a disk drive (SSD) that can be used as a high-speed temporary area during job execution. The global I/O node is connected to second-layer storage via an input/output interface.
Main unit	The main unit configuration has 3 BoBs. The main unit is also called a shelf.
Main unit rack	The main unit rack is the chassis housing the main unit. Up to 8 main units can be mounted in the main unit rack.

B.2 Tofu Unit and Tofu Coordinates

The FX server nodes are connected with one another via a high-speed network called the Tofu interconnect. The Tofu interconnect is a compute network used as an inter-node communication channel in a parallel job.

The FX server handles 12 nodes connected by the Tofu interconnect as a single unit, which is called "Tofu unit." The Tofu unit is regarded as a 2x3x2 cuboid. The axes are called the A, B, and C axes. The Tofu unit is managed as an object placed on three-dimensional coordinates, that is, along the X, Y, and Z axes. Therefore, each FX server node is positioned on the X, Y, Z, A, B, and C axes, at a location defined using six-dimensional coordinates. The six-dimensional coordinates are called Tofu coordinates. The upper limits of the Tofu X, Y, and Z coordinates vary depending on the system size and configuration.

Figure B.2 Tofu Unit and Tofu Coordinates