FUJITSU

Fujitsu Software
Technical Computing Suite V4.0L20

Job Operation Software Administrator's Guide for Power Management

J2UL-2457-02ENZ0(03)
March 2023

Preface

Purpose of This Manual

This manual describes the power management function settings and operation methods provided by the Job Operation Software of Technical Computing Suite.

Intended Readers

This manual is intended for administrators who operate and manage the power of the system where the Job Operation Software is installed.

The manual assumes that readers have the following knowledge:

- Basic Linux knowledge
- Understanding of an overview of the Job Operation Software based on "Job Operation Software Overview"

Administrators who operate the system are requested to read "Job Operation Software Administrator's Guide for System Management." Administrators who perform job operation are requested to read "Job Operation Software Administrator's Guide for Job Management."

For details on maintenance and troubleshooting, see "Job Operation Software Administrator's Guide for Maintenance" and "Job Operation Software Troubleshooting."

Organization of This Manual

This manual is organized as follows.

Chapter 1 Overview of the Power Management Function

This chapter provides an overview of the power management function.

Chapter 2 Details of the Power Management Function

This chapter describes the functions provided by the power management function in detail.

Chapter 3 Power Management Function Settings

This chapter describes the necessary setting items for using the power management function.

Chapter 4 Operation with the Power Management Function

This chapter specifically describes how to operate the power management function.

Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function

This appendix describes settings for the exit functions (hooks) of the power cap scheduling (job power estimate) and power knob operation functions. The exit functions are functions that acquire estimated job power consumption values and job power consumption information and use power knob operations.

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.

Notation of Cluster Names

In this manual, "cluster" refers to a compute cluster unless otherwise noted.

Notation of Node Names

In this manual, "system management node" refers to the active system management node unless otherwise noted, and "compute cluster management node" refers to the active compute cluster management node.

Administrators

The Job Operation Software has different types of administrators: system administrator, cluster administrator, and job operation administrator. Unless otherwise noted, the descriptions in this manual apply to functions for system administrators and cluster administrators. Therefore, the term "administrator" in the text usually means an administrator with cluster administrator privileges or higher.

Prompts in Command Input Examples

means that OS administrator (super user) privileges are required.

Path Names of Commands

In the examples of the operations, the path names of the commands in the directory /bin, /usr/bin, /sbin or /usr/sbin might not be represented by absolute path.

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

When exporting this document or providing it to a third party, check the export control-related laws and regulations of your country and the U.S., and take necessary procedures.

Trademarks

- Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Intel is a trademark of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- All other trademarks are the property of their respective owners.

Date of Publication and Version

Version	Manual code
March 2023, Version 2.3	J2UL-2457-02ENZ0(03)
March 2022, Version 2.2	J2UL-2457-02ENZ0(02)
June 2020, Version 2.1	J2UL-2457-02ENZ0(01)
March 2020, Second version	J2UL-2457-02ENZ0(00)
January 2020, First version	J2UL-2457-01ENZ0(00)

Copyright

Update History

Changes	Location	Version
Fixed the descriptin about Job ACL.	3.6.3	2.3
Changed apply settings of papwrm.conf.	3.7.1	2.2
Added a note about the visualization support function.	2.1.3	2.1
Added an article about the apply settings.	3.7.1	
Fixed the description about a sum of the power consumption of the entire system.	4.1.1	
Changed sample CPU Frequency.	2.2.2 2.3.2 3.2 3.6.1 3.6.4	2
Added disabling for the System Power Collecting/Visualization Support Function.	3.3.6	
Added disabling for the Automatic Compute Node Power Control Function.	3.4.2	
Removed settings for job power consumption measurement.	3.5	
Added disabling for the Job Power Estimate Function.	3.5.3	
Added disabling for the Power Knob Operation Function.	3.6.5	
Changed the look according to product upgrades.	-	

Contents

Chapter 1 Overview of the Power Management Function

This chapter describes the purpose of the power management function and provides a functional overview.

In a computer system that performs scientific computations, multiple computers called nodes execute parallel distributed processing to improve computational performance. The number of nodes has been continuously increasing to improve performance through distribution, and the increase of power consumption is becoming a big problem. Therefore, the following requirements must be met to reduce the power problem in a large-scale system:

- The system can operate with its power consumption limited to an arbitrary value.
- The system can operate in power-saving mode by reducing its unnecessary power consumption.

The Job Operation Software provides the power management function with the purpose of meeting these requirements.

The power management function achieves the following functions in cooperation with the job management function of the Job Operation Software.

Table 1.1 Features of the Power Management Function

Function	Target (*)	Description
System power collecting/visualization support function	System	Collects the power consumption information of the system, compute nodes, jobs, or system-related equipment, and supports visualization of the information ("2.1 System Power Collecting/Visualization Support Function").
Power-saving function	System	[System] Provides 2 types of power-saving functions for the power consumption of compute nodes to which jobs are not allocated. - Saves system power consumption by automatically controlling power supply to compute nodes where no jobs are scheduled to run for a certain consecutive period of time or longer. This is achieved in cooperation with the job scheduler of the job management function. ("2.2.1 Automatic Compute Node Power Control Function [PG]") - Enables the administrator to save the power consumption of each device in compute nodes in cooperation with the job scheduler of the job management function. ("2.2.2 Power Knob Operation Function with the Job Operation Software [FX]")
Power API function	Job	End users can perform power measurement by using the Sandia Power API in a job. End users can also control power in the FX server. For more information, refer "Job Operation Software API user's Guide for Power API." The system administrator can limit the range of power that can be controlled by end users. ("2.3.2 Power Knob Operation Restriction Function")
Capping function	System	Prevents the allowable power of the system from being exceeded through job scheduling that takes into account the total power based on the estimated power consumption of a submitted job. This is achieved in cooperation with the job scheduler of the job management function. ("2.4.1 Power Cap Scheduling Function" and "2.4.1.1 Job Power Estimate Function")

(*) "Target" indicates the following:
System : The ability to manage power for all compute nodes and associated equipment.
Job : The ability to manage power for a specific job and the compute nodes on which it runs.

Note

In this manual, "compute node" refers to a PRIMERGY or FX server. Models for PRIMERGY server must support the RAPL (Running Average Power Limit) and the IPMI (Intelligent Platform Management Interface).

See

For details on the job operation management function, see "Job Operation Software Administrator's Guide for Job Management."

Chapter 2 Details of the Power Management Function

This chapter describes the power management function in detail.

The power management function provides the following functions:

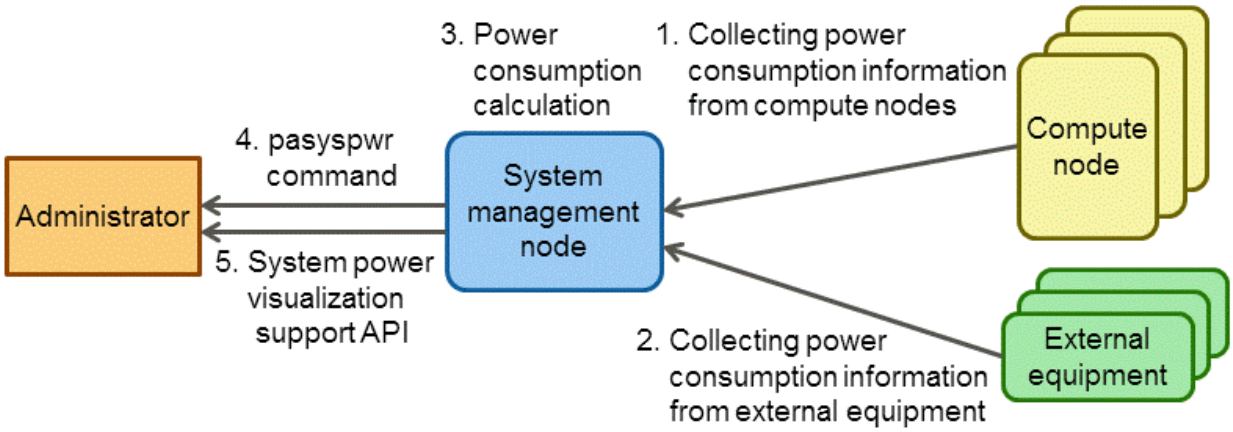
- System power collecting/visualization support function
  - Power collecting function
  - Power calculation function
  - Visualization support function
- Power-saving function
  - Automatic compute node power control function
  - Power knob operation function
- Power API function
  - Sandia Power API
  - Power knob operation restriction function
- Capping function
  - Power cap scheduling function

2.1 System Power Collecting/Visualization Support Function

The system power collecting/visualization support function provides three functions: the "power collecting function," "power calculation function," and "visualization support function." The power collecting function regularly (at one-minute intervals) collects power consumption information from the compute nodes constituting the system and external equipment. The power calculation function calculates average power consumption, etc. based on the collected information. The visualization support function outputs those pieces of information. These functions enable the administrator to understand the power consumption of the system so that they can plan a power budget and analyze trends in power consumption.

The following shows the operation of the system power collecting/visualization support function.

Figure 2.1 Operation of the System Power Collecting/Visualization Support Function



1. Collecting power consumption information from compute nodes [Power Collecting Function]
   The function regularly collects power consumption information from all the compute nodes.
2. Collecting power consumption information from external equipment [Power Collecting Function]
   The function regularly collects power consumption information from non-compute node devices required for system operation. The non-compute node devices required for system operation that are described here include not only devices directly required for job

execution, such as disk drives and network switches, but also devices not directly involved in job execution, such as cooling devices for computers. In this manual, these devices are called "external equipment."

3. Power consumption calculation [Power calculation function]
   The function regularly calculates average power consumption and total power consumption based on the power consumption information collected in 1 and 2.

4. pasyspwr command [Visualization Support Function]
   This command can output the power consumption information of the entire system, compute nodes/external equipment, or each power group at any time.

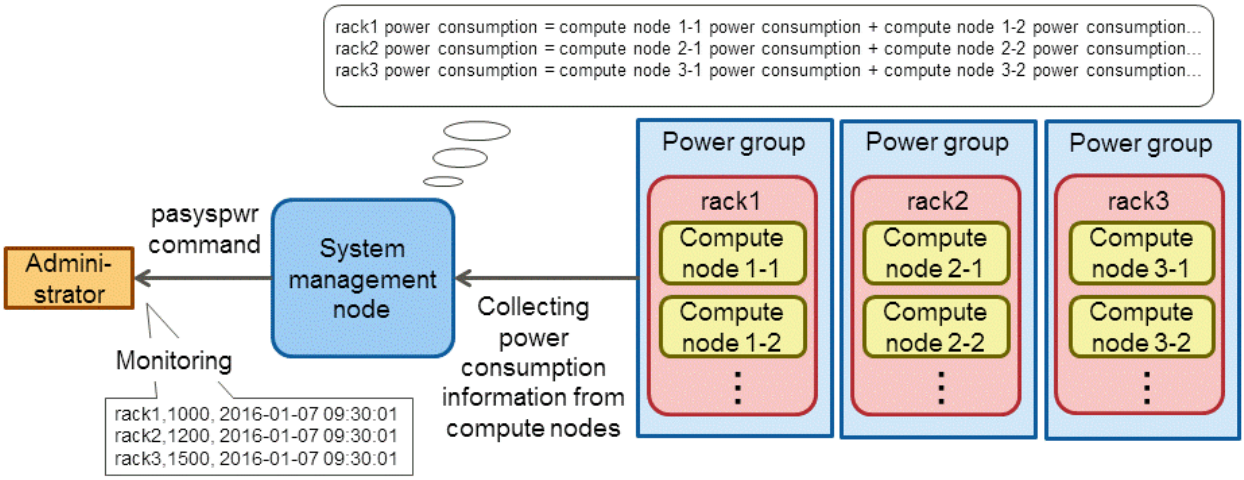
5. System power visualization support API [Visualization Support Function]
   The system power visualization support API enables the administrator to view the power consumption information of compute nodes/external equipment, or each power group from a command or load module created in C/C++ language at any time.

To use these functions, it is necessary to make settings for the system power collecting/visualization support function. For details on the setting method, see "3.3 Settings for the System Power Collecting/Visualization Support Function."

The system power collecting/visualization support function can handle power consumption information by considering an aggregation of compute nodes/external equipment as one unit and calculate the total power consumption of the unit. In this manual, this aggregation is called a "power group." For details on how to configure a power group, see "3.3.4 Settings for a Power Group."

The following is a conceptual diagram of use when power groups are configured in units of racks.

Figure 2.2 Conceptual Diagram of Use When Power Groups are Configured in Units of Racks



For example, if a group of equipment pieces constituting a rack is defined as a power group, the function outputs power consumption information by summing the average or momentary power consumption of the equipment pieces. This enables the administrator to monitor the state of power consumption in units of racks.

The following describes the power collecting, power calculation, and visualization support functions one by one.

2.1.1 Power Collecting Function

The method to collect power consumption information depends on the device from which the information is collected.

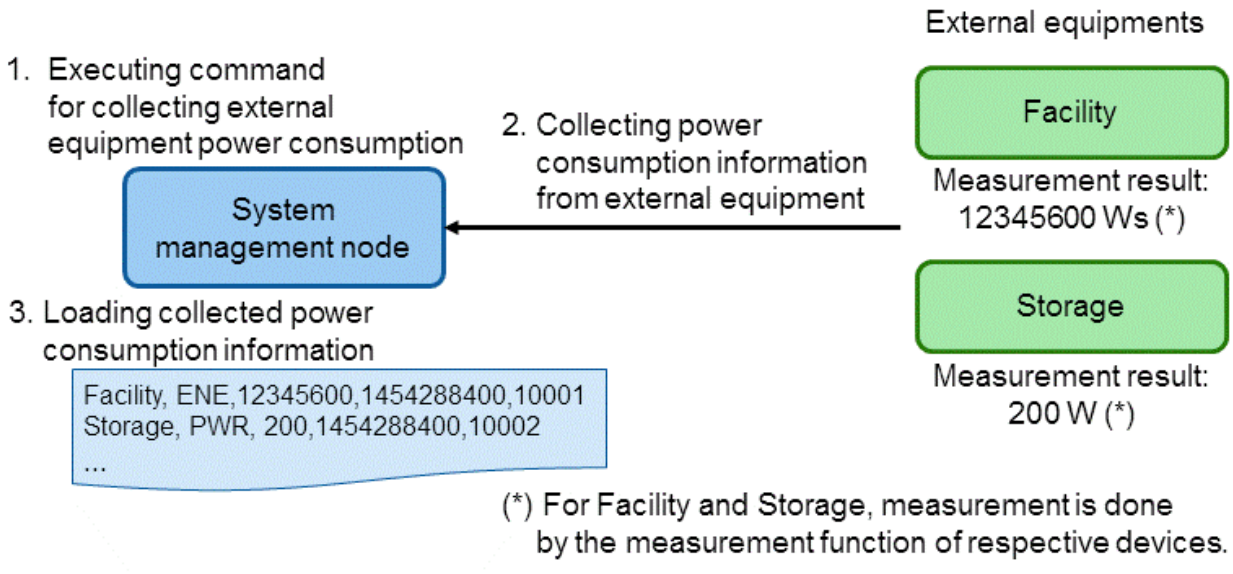
- Compute nodes
  Momentarily measured momentary power consumption (unit: W (watt)) or integral power consumption (unit: Ws (watt-second)) and the measurement time of the power consumption are regularly collected as power consumption information. This occurs at the timing set in "3.3.1 Settings for System Power Collecting."
  Integral power consumption is collected from FX servers and momentary power consumption are collected from PRIMERGY servers.

- External equipment
  External equipment varies depending on the system configuration, and collectable power consumption information and how to collect it also vary. Using a power information collection tool (command), this function can load power consumption information measured by the internal measurement functions of external equipment. The power collecting function calls the command a "command for

collecting external equipment power consumption." The command for collecting external equipment power consumption needs to convert power consumption information measured for each external equipment into the format defined by this function. The command for collecting external equipment power consumption is regularly executed at the timing set in "3.3.1 Settings for System Power Collecting" (same as the collection timing for compute nodes).

The following shows the operation when a command for collecting external equipment power consumption is executed.

Figure 2.3 Operation When a Command for Collecting External Equipment Power Consumption is Executed



In this example, two external equipment pieces (Facility and Storage are the names registered in the external equipment registration file /etc/opt/FJSVtcs/pwrm/extdev) exist in the system.

1. Executing the command for collecting external equipment power consumption
   The system power collecting/visualization support function regularly executes the command for collecting external equipment power consumption (which is created and registered by the administrator in advance).
2. Collecting power consumption information from external equipment
   The command for collecting external equipment power consumption collects and outputs power consumption information from the external equipment.
3. Loading collected power consumption information
   The system power collecting/visualization support function loads the results output by the command for collecting external equipment power consumption.

Note

- External equipment needs to be registered in advance in the equipment registration file /etc/opt/FJSVtcs/pwrm/extdev. In addition, the administrator needs to create a command for collecting external equipment power consumption and register it in advance in the configuration file papwrm.conf. For details on how to register external equipment and how to create and register a command for collecting external equipment power consumption, see "3.3.3 Settings for External Equipment."
- In external multiple equipment, create a command for collecting external equipment power consumption in order to collect and output the information by a single instance of the command.
- During a redundantly configured system management node's failover due to a system error or maintenance work, the power collecting function is stopped. Therefore, power consumption information is not collected.
- While the system management node is failing over or cannot collect power due to a communication error, the power information is not obtained. However, since FX servers internally contain up to 10-minute power consumption, the information is automatically collected after communication recovery with the system management node.

- Depending on the state of the compute nodes when the power collecting function collects the power consumption information, some compute nodes may not be able to collect.

2.1.2 Power Calculation Function

The power calculation function performs the following two calculations:

- Calculating average power consumption
- Summing power consumption

The following describes the respective calculations in detail.

- Calculating average power consumption

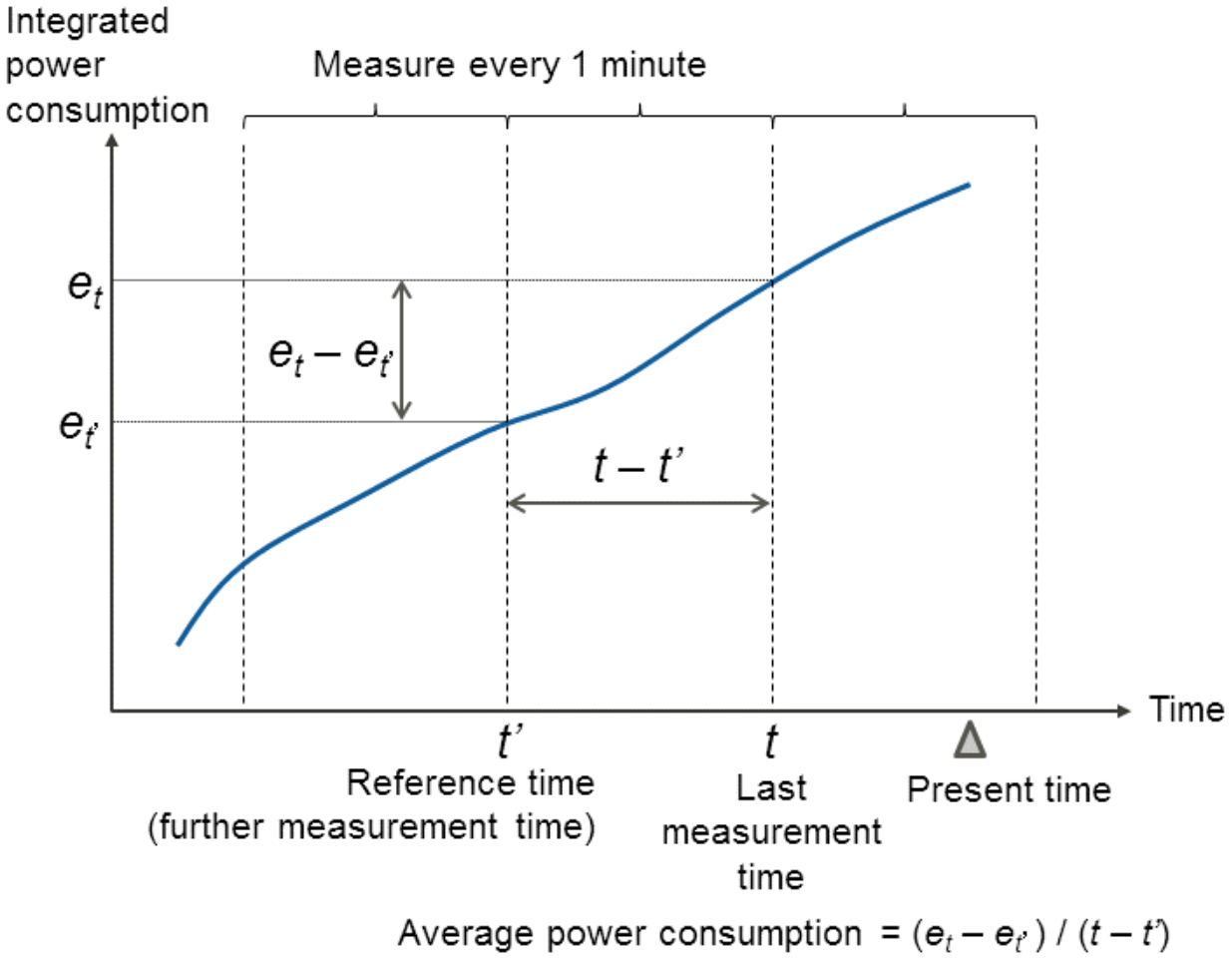
  In the case of devices (compute nodes and external equipment) from which integral power consumption can be obtained, power consumption information is output as average power consumption. When the information is output, average power consumption (W) is calculated from integral power consumption (Ws).

  Average power consumption is calculated by the following formula. The formula assumes that the last measurement time collected is t, the integral power consumption at the time is e_t, the past measurement time closest to the time t is t', and the integral power consumption at the time is $e_{t'}$. Note that the measurement time t' is called the reference time.

$$\text{Average power consumption (W)} = (e_t - e_{t'}) / (t - t')$$

  The following shows the measurement timings used for calculating average power consumption from collected integral power consumption.

Figure 2.4 Measurement Timing Used for Calculating Average Power Consumption From Collected Integral Power Consumption



- Summing power consumption
  Handling the power consumption information of the entire system, any compute nodes/equipment, or for each power group, the function sums momentary or average power consumption collected from individual devices. To find the total based on momentary or average power consumption, see "4.1.1 pasyspwr Command." The most recently collected values are used as momentary or average power consumption values. The following rules apply to the calculation of summing power consumption:

  - Calculating a sum by using the power consumption at the measurement time closest to the execution time of the pasyspwr command or system power visualization support API
    The measurement time of power consumption varies depending on the device to be measured. Therefore, an acceptable range can be set as the variation of measurement time that is allowed in calculation of a sum (using the definition item AcceptableRange in the configuration file papwrm.conf). For details on the setting method, see "3.3.2 Settings for the Function to Calculate a Sum of Power Consumption." Power consumption outside the acceptable range is not summed. However, if the --last option is specified at the time of command execution, power consumption outside the acceptable range is also used for calculating a sum.

  - Calculating a sum of power consumption of the specified type, that is, momentary or average power consumption
    In the case of the pasyspwr command, either of the types of power consumption to be summed can be specified by specifying an option.

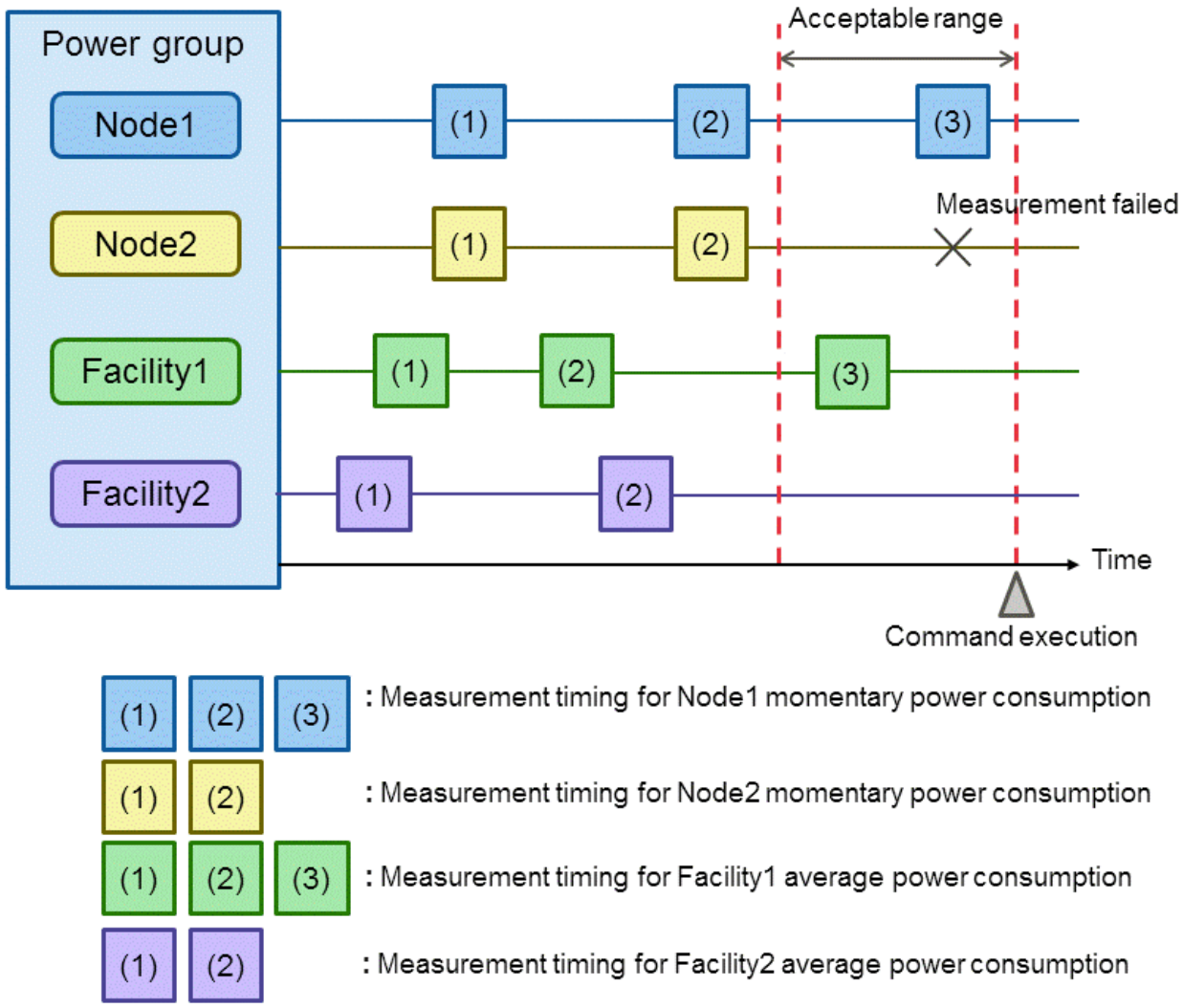
    - When the --momentary option is specified, momentary power consumption is preferentially used to calculate a sum.

    - When the --average option is specified, average power consumption is preferentially used to calculate a sum.

    If there is no power consumption of the specified type (power consumption outside the acceptable range is also considered as non-existing), the existing type of power consumption (momentary or average power consumption) is summed.

The following shows an example of calculating a sum of power consumption of a power group.

Figure 2.5 Sum of Power Consumption of a Power Group



The above figure shows the measurement timings for the momentary power consumption of the compute nodes Node1 and Node2 and for the average power consumption of external equipment Facility1 and Facility2. An acceptable range is set as the variation of measurement time that is allowed in calculation of a sum. Values measured within the acceptable range are summed.
The value of Node1 that is measured at the measurement timing (3) can be used for calculating a sum. However, Node2 cannot be used for calculating a sum because no value is measured within the acceptable range.
The value of Facility1 that is measured at the measurement timing (3) can be used for calculating a sum. However, Facility2 is not used for calculating a sum because no value is measured within the acceptable range.
In the case of this example, calculation results in an error because not all power consumption to be used for calculating a sum is there.

2.1.3 Visualization Support Function

The pasyspwr command and the system power visualization support API are provided as functions to output the power consumption information of a compute node, external equipment, or a power group.

- pasyspwr command

  The pasyspwr command outputs the following power consumption information:

Table 2.1 Output information of the pasyspwr command

	Any compute nodes	Any external equipment	Any power group	Entire system
Most recent	- Each momentary or average power consumption - Total value	- Each momentary or average power consumption - Total value	- Each momentary or average power consumption - Total value	- Each momentary or average power consumption - Total value
any given time or any given period	- Each momentary or average power consumption	- Each momentary or average power consumption	- Each momentary or average power consumption	-

- Most recently measured momentary or average power consumption (unit: W) in any compute nodes/external equipment
- Most recently measured momentary or average power consumption (unit: W) of the individual devices constituting any power group
- Sum of momentary or average power consumption (unit: W) of the entire system, any compute nodes/external equipment, or a power group.
- Momentary or average power consumption (unit: W) measured at any given time or for any given period in any compute nodes/external equipment or a power group

Power consumption information is output in text format. For details on the output method and contents of power consumption information, see "4.1.1 pasyspwr Command."

- System power visualization support API

An API for the C and C++ languages is provided so that power consumption information can be referenced from an application.

Table 2.2 Output information of the system power visualization support API

	Any compute nodes	Any external equipment	Any power group	Entire system
Most recent	- Each momentary or average power consumption	- Each momentary or average power consumption	- Each momentary or average power consumption - Total value	-
any given time or any given period	-	-	-	-

- Most recently measured momentary or average power consumption (unit: W) in any compute nodes/external equipment
- Most recently measured momentary or average power consumption and a sum of the power consumption (unit: W) of the individual devices constituting any power group

For details on this API, see "4.1.2 System Power Visualization Support API."

Note

- The pasyspwr commands and the system power visualization support API output the average power consumption as the power consumption of FX servers and the momentary power consumption as that of PRIMERGY servers.
- They output the average power consumption if type of external equipment is ENE (Integral power consumption), and the momentary power consumption if type of external equipment is PWR (momentary power consumption).
- When outputting power information of any compute nodes for any given period (pasyspwr command --trace option), power information at some time may be missing or duplicated. If it is missing, use the power information before and after it. If they are duplicated, use the power information with the higher serial number. "2.1.2 Power Calculation Function" does not sum the missing or duplicated power consumption.

2.2 Power-Saving Function

It may occur that some compute nodes have no jobs allocated in job operation. Such compute nodes consume power even while they are waiting for allocation.
The power-saving function provides the "automatic compute node power control function," which automatically stops and starts power supply to the compute nodes to which no jobs are allocated. This function reduces unnecessary power consumption. It also provides a function that switches compute nodes between performance priority mode and power-saving mode in line with the start and stop of jobs.

2.2.1 Automatic Compute Node Power Control Function [PG]

The automatic compute node power control function automatically stops the power supply to the compute node which are expected that no jobs are allocated for at least an hour. To reduce unnecessary power consumption efficiently, the automatic compute node power control function does not restart the stopped compute node until it has stopped for at least one hour. When a job is allocated the node which elapsed one hour after stopped, the function restarts it at the execution start time of the job.

This function becomes available when the administrator enables it. For details on the setting method, see "3.4 Settings for the Automatic Compute Node Power Control Function."

Information

The administrator can set a range of compute node IDs that should be targets of the automatic compute node power control function. For details on the setting method, see"3.4 Settings for the Automatic Compute Node Power Control Function."

Note

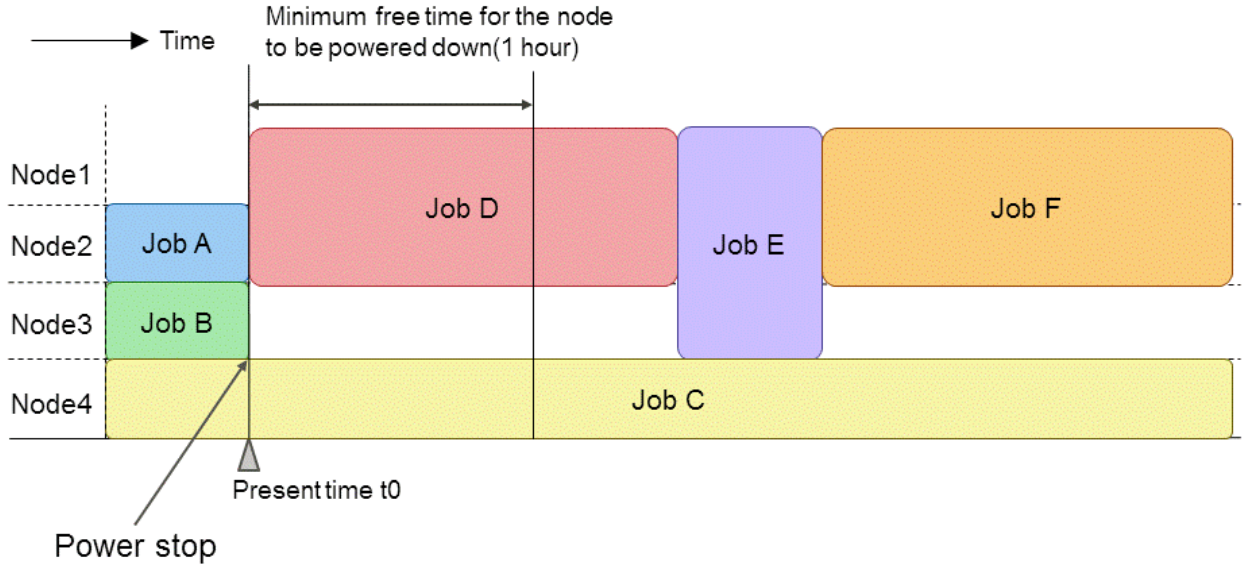
The automatic compute node power control function automatically excludes the following nodes from targets to which power supply is to be stopped:

- Compute nodes that are isolated from operation due to maintenance work, etc.

When an isolated node is incorporated into operation again, the node is automatically judged as a target of automatic power supply stop.

The following shows power control by the automatic compute node power control function.

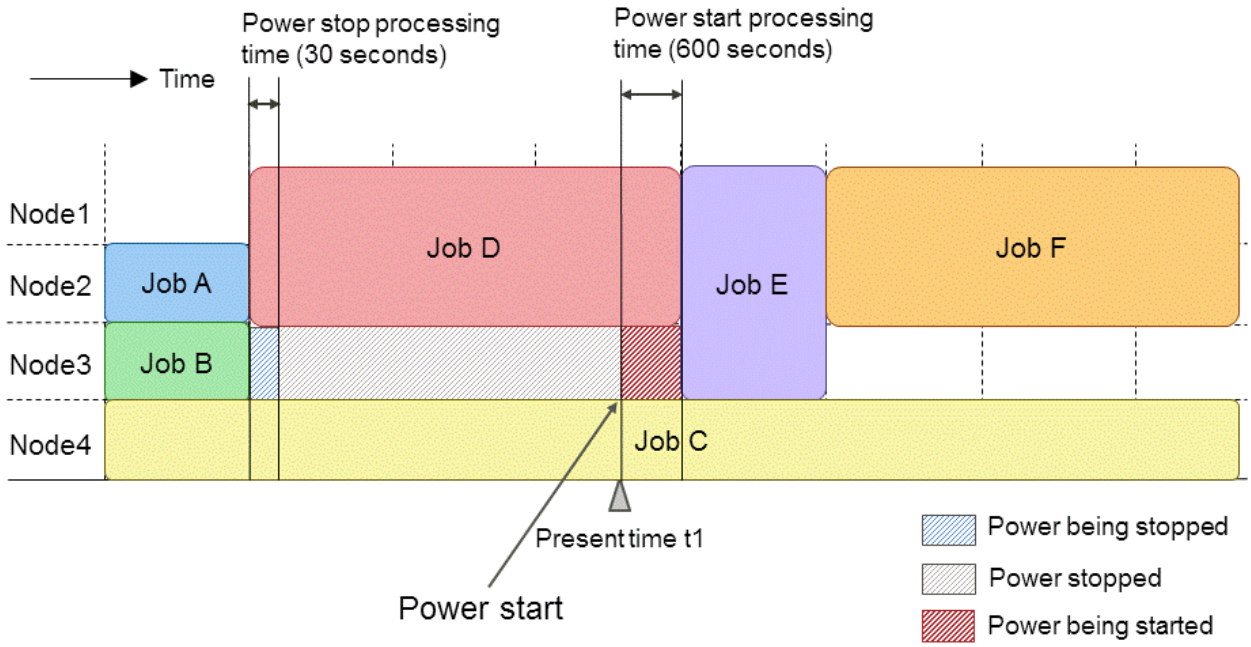
Figure 2.6 Power Control by the Automatic Compute Node Power Control Function (1)



The automatic compute node power control function powers down compute nodes that will not run a job for more than an hour.
Since Node3 continues to be in a free state for more than one hour after ending job B, the automatic compute node power control function stops power supply to it.

To reduce unnecessary power consumption efficiently, the stopped node is excluded from the allocation target until one hour has elapsed after stopping.

Figure 2.7 Power Control by the Automatic Compute Node Power Control Function (2)



The automatic compute node power control function starts power supply to the stopped node in line with the execution start time of the next job.
For Node3, the automatic compute node power control function starts power start processing in such a way that the start processing will be completed when job E is executed.

Note

When setting the deadline scheduling function to stop the operation such as system maintenance, depending on the timing, the automatic compute node power control function may power up the compute node during the deadline scheduling.

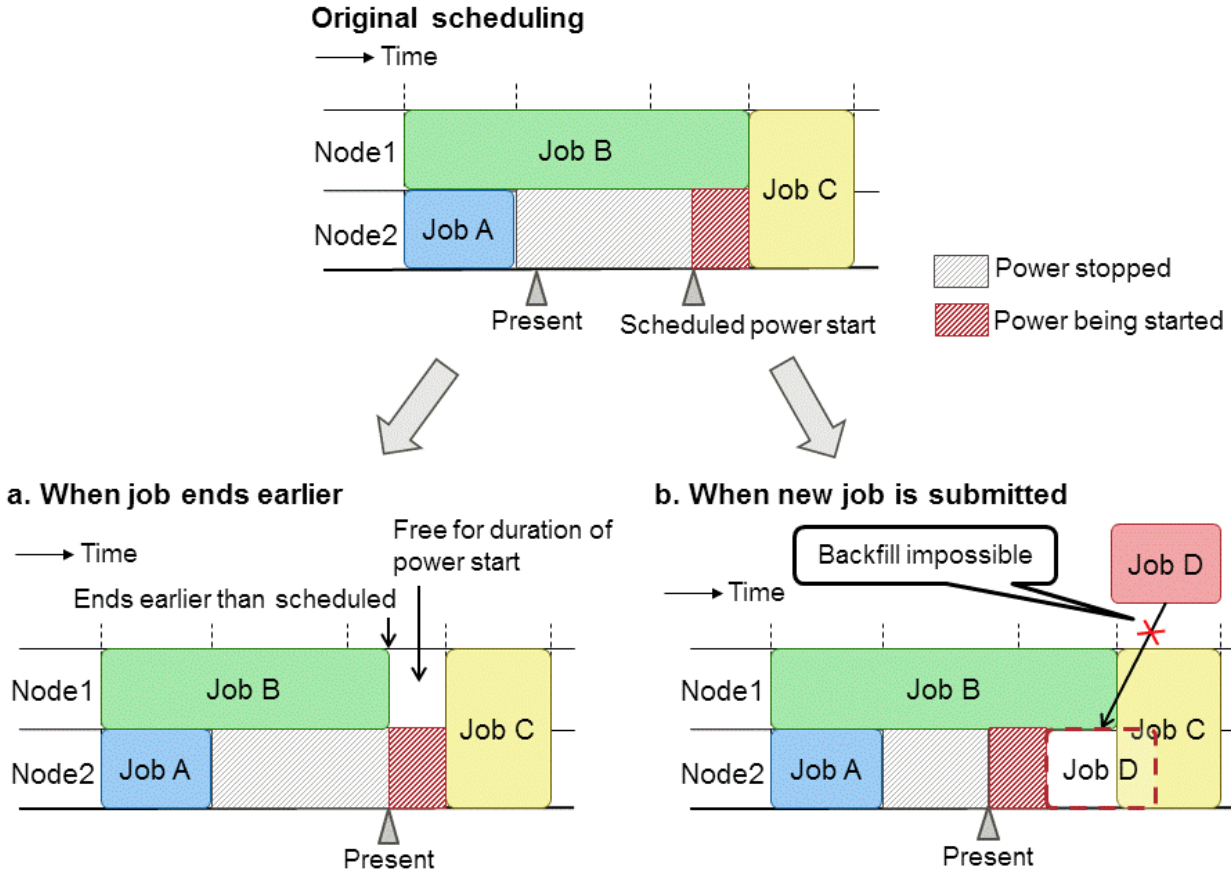
Therefore, for the range of nodes that are subject to deadline scheduling, disable the automatic compute node power control function before setting the deadline scheduling feature.

Details of the Job Operation Management Function" and also "Cluster deadline scheduling management" in "Chapter 4 Operation with the Job Operation Management Function" in "Job Operation Software Administrator's Guide for Job Management."

Effect of power-up processing on jobs

If a job is assigned to compute nodes that have been powered down by the automatic compute node power control function, the job execution scheduled start time may be later than if it were assigned to compute nodes that have not been powered down. Here are some examples:

Figure 2.8 Effect on Job Throughput



a. When job ends earlier

Even if job B running on Node1 ends earlier than the scheduled execution start time, it takes time to power up Node2 which is stopped by the automatic power control function. Therefore, job C cannot be started immediately after job B ends.

b. When new job is submitted

When a new job D is submitted, it may not be possible to backfill because of insufficient time for the power up process even though Node 2, which is powered down, has enough time to run the job. If new jobs are submitted frequently, you can reduce the effect by narrowing the scope of compute nodes that are subject to the automatic compute node power control function. However, power consumption increases. Set the range of compute nodes that are to be targets of the automatic compute node power control function by taking this into account.

2.2.2 Power Knob Operation Function with the Job Operation Software [FX]

The power knob operation function with the Job Operation Software is a function to automatically perform power knob operations at the start and end of a job so that job execution performance is ensured and power consumption by compute nodes is reduced.

The power knob operation function transitions compute nodes with no running jobs to power-saving mode and to performance priority mode from power-saving mode in line with the start of job execution. This way of control reduces unnecessary power consumption in the system. The power knob operation function enables the administrator to set each mode.

- Power-saving mode (IdleState)
- Performance priority mode (RunningState)

Information

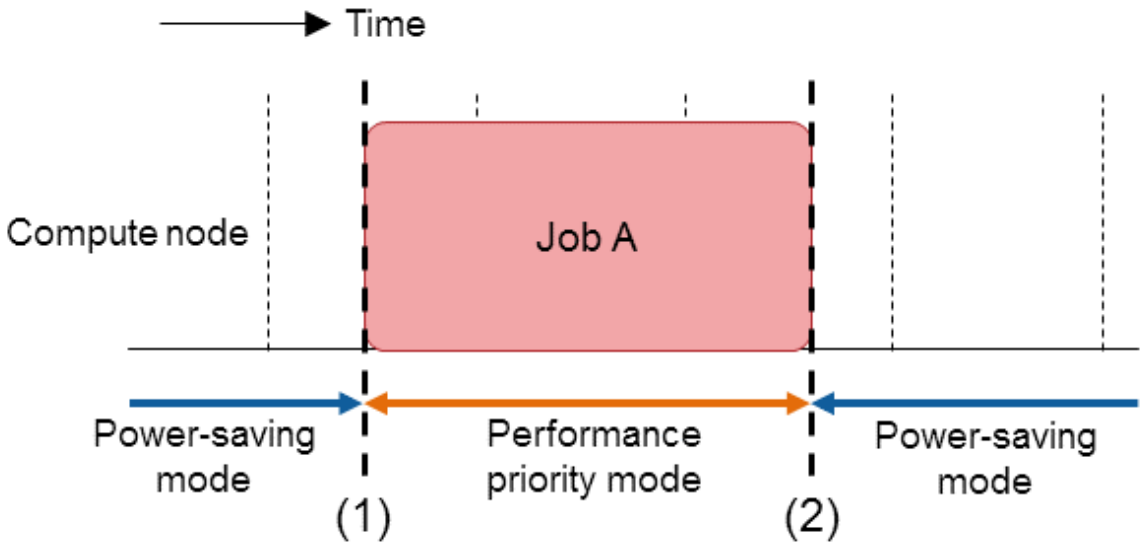
A power knob is an individual performance limiting function for hardware that can be dynamically controlled by the software. Use this function to control node power consumption. For details on controllable power knobs, see "3.6 Settings for the Power Knob Operation Function [FX]."

The following provides examples of power knob operations.

When a Single Job is Executed on Nodes

1. Before the execution of job A starts, the power knobs of compute nodes, which will execute the job, are changed to performance priority mode. (1)
2. After job A ends, the power knobs of Node1 and Node2 are changed to power-saving mode. (2)

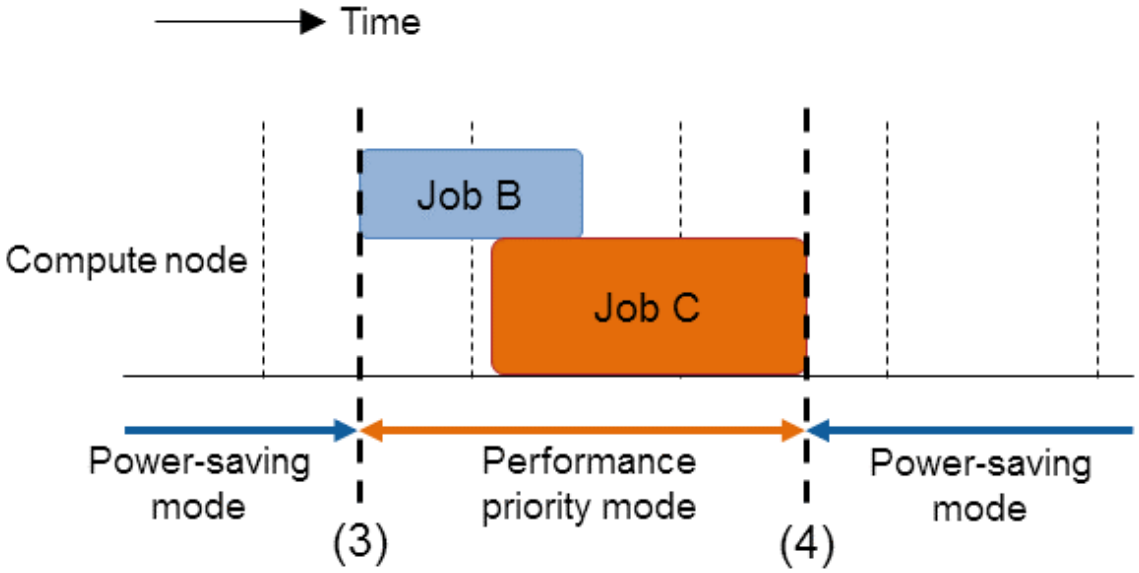
Figure 2.9 Power Knob Operations When a Job Runs/Ends (Node-exclusive Job)



When Multiple Jobs are Executed on Nodes

1. Before the execution of the first job (job B) starts, the mode is changed to performance priority mode. (3)
2. When job B ends, no power knob operation is performed due to a subsequent job (job C).
3. When job C starts, no power job operation is performed due to a preceding job (job B).
4. After the last job (job C) ends, the mode is changed to power-saving mode. (4)

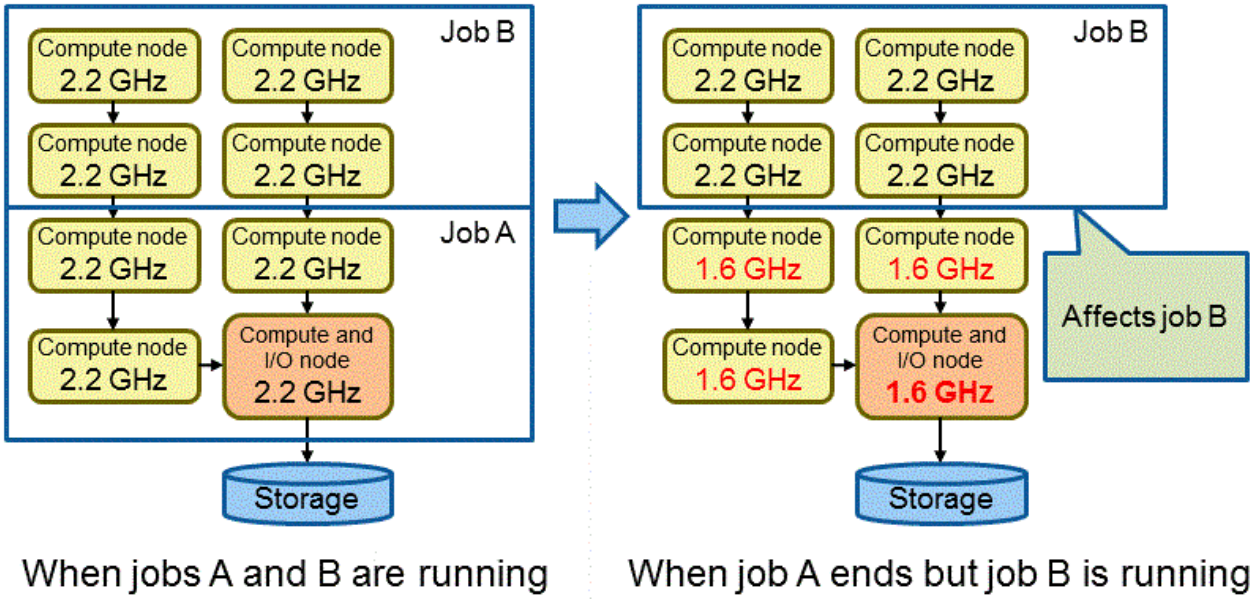
Figure 2.10 Power Knob Operations When a Job Runs/Ends (Node-Sharing Job)



Setting Function for Each Node Type

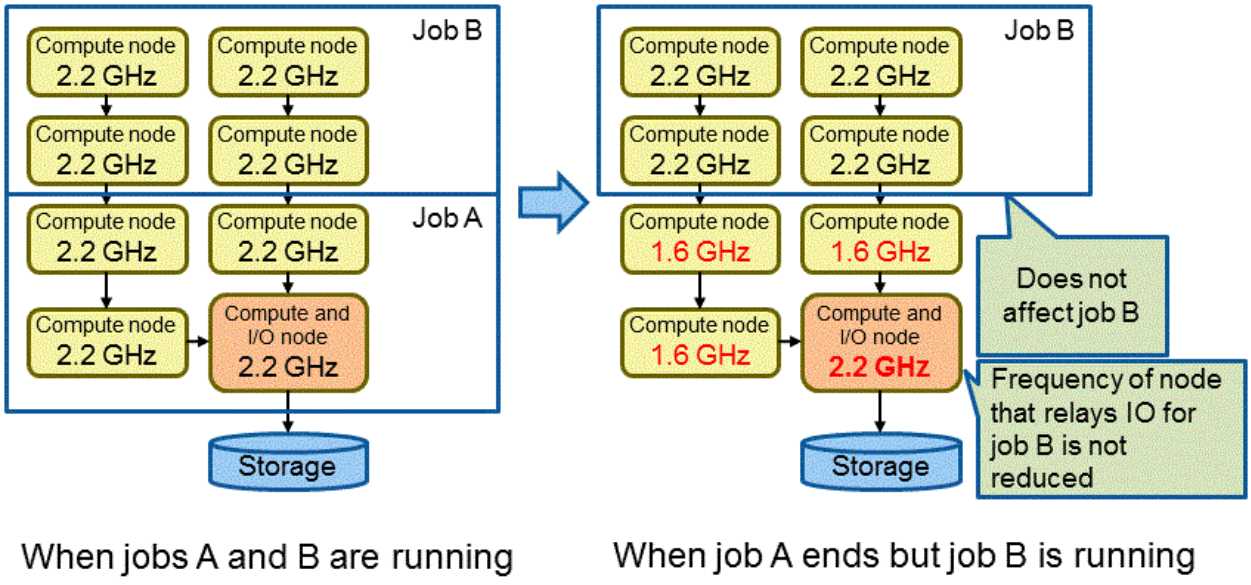
In the case of the FX server, the node serving as both compute and I/O node is used to process I/O for jobs that are running on other compute nodes, even if no jobs are running on them. Therefore, if the power knob operation is performed on the node serving as both compute and I/O node, I/O performance will be affected. In the configuration shown in the Figure 2.11 Effect of the Node Serving as Both Compute and I/O Node on a Job, there are jobs A and B, which use the same compute and I/O node. When job A ends first and the nodes, including the node serving as both compute and I/O node, are transitioned to power-saving mode (1.6 GHz), file I/O performance for job B is affected.

Figure 2.11 Effect of the Node Serving as Both Compute and I/O Node on a Job



To solve this problem, the administrator can set different power knob operations for the compute nodes and the node serving as both compute and I/O node. Making this setting reduces the effect of the decrease in the performance of the node serving as both compute and I/O node that relays I/O for a job ("Figure 2.12 Reducing Effect on the Node Serving as Both Compute and I/O Node by Making a Different Setting"). Make this setting with knob_io in the job ACL. For details, see "3.6.4.3 Setting Examples for Power Knob Operations at Job Submission Time."

Figure 2.12 Reducing Effect on the Node Serving as Both Compute and I/O Node by Making a Different Setting



2.3 Power API Function

The Power API function is the feature that operates on the Power API supported by the hardware. The Power API function has two features:

a. The end-user sets the value of the power knob : "Power knob operation function for end users"

b. Administrator limits end-user power knob operations : "Power knob operation restriction function"

2.3.1 Power Knob Operation Function for End Users

End users can operate power knobs.

Operation at the Start of Job Execution

End users can set a power knob value by using a job ACL when submitting a job.

PowerAPI

The Power API advocated by Sandia can be used during job execution.
For FX server and PRIMERGY server, end users can measure power by using the Power API. For FX server, they can also control power by using the Sandia Power API within a job.
The administrator can set a range of power knob values that can be changed by end users when they control power ("2.3.2 Power Knob Operation Restriction Function").
For details on the Power API, see "Job Operation Software API user's Guide for Power API."

2.3.2 Power Knob Operation Restriction Function

The power knob operation restriction function enables the administrator to restrict power knob operations by end users at the time of job submission and job execution. The administrator can restrict them through the following two settings:

- Configuration file of the power management function (papwrm.conf)
- Custom resources in the job ACL

The administrator uses the job ACL function to configure custom resources for power knobs that allows end users to configure it when the job starts.
The administrator can set the upper and lower limits of power knob values to be operated by end users by using the Power API within a job.
If setting items of the same power knob are set in the configuration file of the power management function and as custom resources in the job ACL, the settings in the job ACL are used.

Table 2.3 Settings for the Power Knob Operation Function and Whether They are Operable or Inoperable

Item No.	Administrator Setting		Operable/Inoperable for End Users	
	Configuration File of Power Management Function	Custom Resource Setting in Job ACL	Specification at Job Submission	Operation Within Job (FX server Only)
1	Upper and lower limits set	Set	Specifiable	Operable
2	Upper and lower limits set	Not set	Not specifiable	Operable
3	Upper and lower limits not set	Set	Specifiable	Operable
4	Upper and lower limits not set	Not set	Not specifiable	Inoperable

The upper and lower limits of power knob values used for the system can be written in the configuration file (papwrm.conf) of the power management function. If setting items of the same power knob are set as custom resources in the job ACL, they take priority over those written in the configuration file of the power management function.

If custom resources are set in the job ACL, the user who submits a job can set a power knob value for compute nodes at job start when submitting the job (item numbers 1 and 3).

For FX server, power knob operations can be performed by using Power API within a job. The range within which operations can be performed is as set in the job ACL (item numbers 1 and 3). If one is not set in the job ACL, the range set in the power management configuration file is used (item number 2).

If the upper and lower limits are not set in the configuration file of the power management function or as custom resource settings in the job ACL, neither operations at the time of job submission nor operations within a job can be performed (item number 4).

Settable values are limited to those that can be set for the hardware. For details, see "Job Operation Software API user's Guide for Power API."

Note

A node-sharing job that shares a one-node hardware resource with others affects jobs of other users, from operations on power knob values for compute resources of other users or resources shared with other users. Therefore, the job ACL set by default does not allow users to perform power knob operations on node-sharing jobs. This setting can be changed through the value of AllowSharedKnob in the configuration file. Before making a setting so that power knob operations on node-sharing jobs are allowed, understand that jobs of other users with whom nodes are shared may be affected.

The power knob function uses environment variables that begin with "PK_." Users should not use environment variables that begin with "PK_."

See

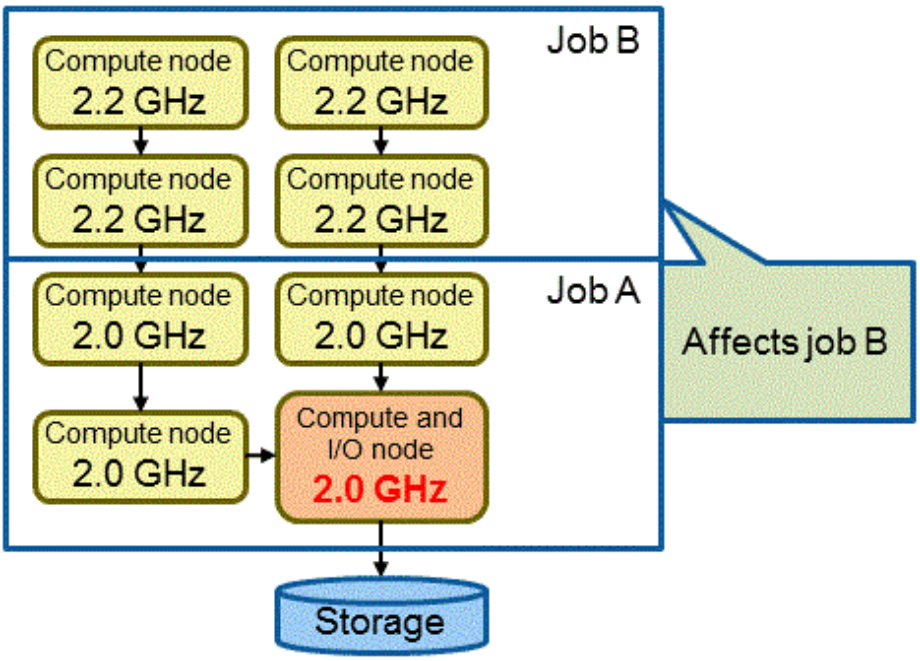
For details on the job ACL and custom resources, see "Chapter 3 Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management."

2.3.2.1 Operation Settings for the Compute and I/O Node [FX server]

When using the job ACL set, power knob values are set in units of jobs. Therefore, the same power knob values are used for both compute nodes and the compute and I/O node. As a result, making power knob settings with the job ACL affects jobs that share I/O, similarly to what is described in "2.2.2 Power Knob Operation Function with the Job Operation Software [FX]" in 2.2.2.

In "Figure 2.13 Example of Effect of Using the Job ACL on Other Jobs," the CPU frequency of compute nodes is set to 2.2 GHz by the power management function. While 2.2 GHz is set for job B in accordance with the power management setting, a frequency of 2.0 GHz is set for CPUs for job A by using the job ACL at the time of its submission. The job ACL specifies that both the frequencies of the compute nodes and compute and I/O node should be changed to the same value (2.0 GHz). As a result, file I/O performance for job B, which uses the same compute and I/O node, is affected.

Figure 2.13 Example of Effect of Using the Job ACL on Other Jobs



To solve this problem, you can use the job ACL knob_io setting to set the job submission power knob value for the node serving as both compute and I/O node in two ways:

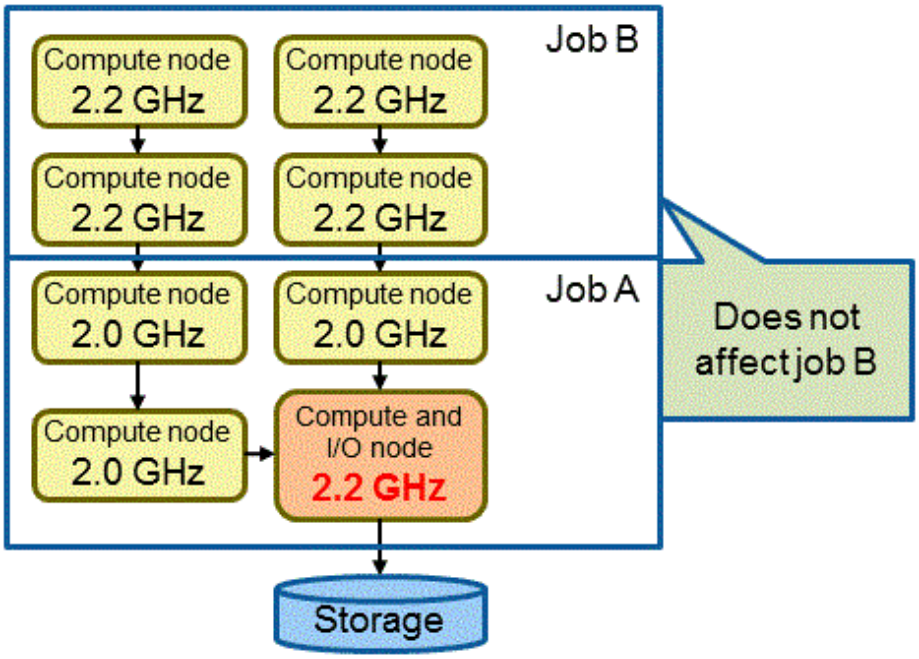
a. Set to the same setting as the compute node.

b. Set to the value specified in the power management configuration file.

In "Figure 2.14 Example of Avoiding Effect on Other Jobs by Excluding Settings for the Compute and I/O Node From the Job ACL Configuration," a frequency of 2.0 GHz is specified as a setting in the job ACL. However, selecting "Set to the value specified in the power management configuration file" will cause it to run at 2.2 GHz using the power management function settings regardless of the job ACL settings.

In this way, the effect on I/O performance for job B is prevented. For details on the settings, see "3.6.3 Configuration File of the Power Management Function and Job ACL Configuration for the Power Knob Operation Function."

Figure 2.14 Example of Avoiding Effect on Other Jobs by Excluding Settings for the Compute and I/O Node From the Job ACL Configuration



2.4 Capping Function

The capping function is a function to keep system power consumption below a certain value. The capping function provides the following function:

- The power cap scheduling function (job power estimate function) that schedules jobs to prevent the power cap for the entire system from being exceeded by estimating the power consumption of the jobs

2.4.1 Power Cap Scheduling Function

In the power cap scheduling function, the job power estimate function implements a limitation on the power consumption of the entire system in cooperation with the job operation management function (job manager function, job scheduler function, and job resource manager).

See

Determination of resource allocation to jobs and an execution order based on estimated values of the power consumption of jobs is supported by the job operation management function. For details on roles that the job operation management function plays for the power cap scheduling function, see "Power cap scheduling function" in "Chapter 2 Details of the Job Operation Management Function" in "Job Operation Software Administrator's Guide for Job Management."

2.4.1.1 Job Power Estimate Function

The following describes the operation of the power cap scheduling function (job power estimate function):

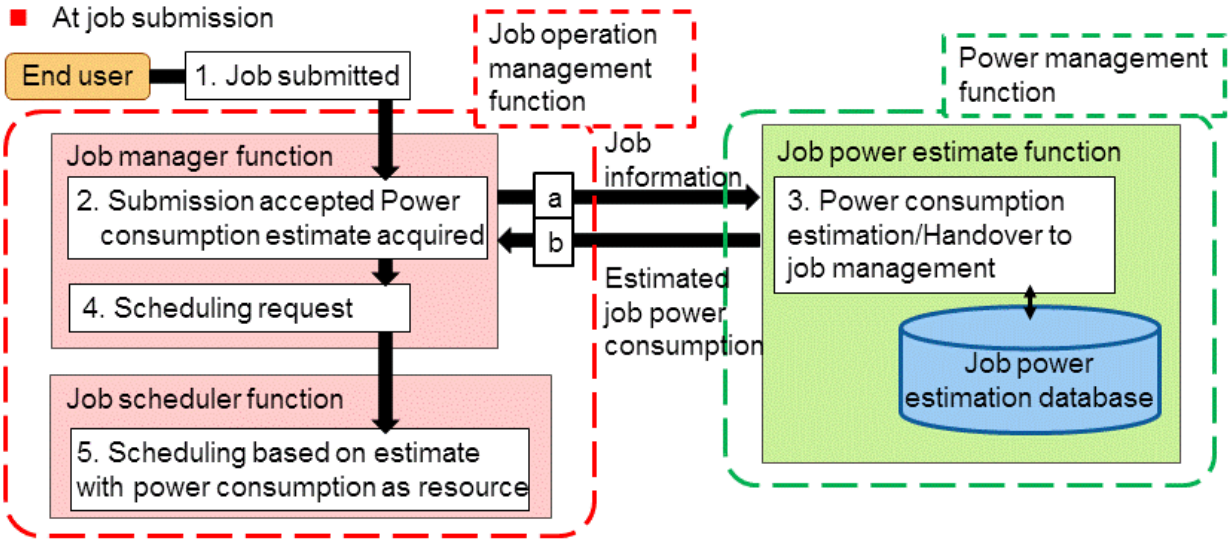
- At job submission

  1. A user submits a job.
  2. The job manager function accepts the submission of the job.
  3. Information on the submitted job is handed over to the job power estimate function through an exit function provided by the job operation management function. (An exit function is an API for the job information acquisition and job information setting functions that can be used in exit function a. In this manual, the exit function is called a "hook.")

The job power estimate function estimates the power consumption of the job based on the handed-over job information and hands over the estimated value to the job manager function through the hook (b).

4. The job manager function requests the job scheduler function to schedule the job.
5. The job scheduler function schedules the job based on the estimated value of the power consumption of the job, without exceeding the preset number of custom resources (see the note below).

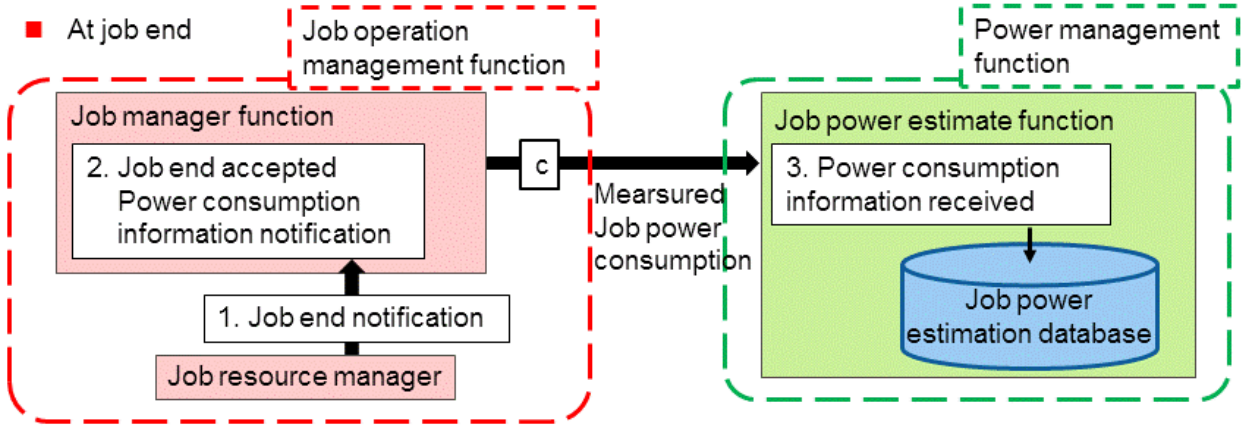
Figure 2.15 Operation of the Power Cap Scheduling Function (Job Power Estimate Function) at Job Submission



- At job end (power estimate)

1. When a job ends, the job resource manager measures the power consumption of the executed job and hands over the measured power consumption value (power consumption information on the job) to the job manager function.
2. When accepting the end of the job, the job manager function hands over the power consumption information on the job to the job power estimate function through the hook (c).
3. The job power estimate function stores the power consumption information on the job in the job power estimation database.

Figure 2.16 Operation of the Power Cap Scheduling Function (Job Power Estimate Function) at Job End



See

- For details on the setting methods for hooks related to the power cap scheduling function, see "Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function."

- For details on the execution timings of hooks, see "Appendix A Functional Comparison of Hooks" in "Job Operation Software Administrator's Guide for Job Operation Manager Hook."

Note

- Since job power is measured in units of nodes, virtual node allocated jobs are not subject to power estimation. For details on virtual node allocated jobs, see "Chapter 1 Job Mechanism" in "Job Operation Software End-user's Guide." In the case of a virtual node allocated job, the value written in DefaultNodePower is used as the estimated value.
- The estimation of the power consumption of jobs may be off the mark during job execution because it is estimation, and the set upper limit may be exceeded. To handle such a case, see "Power cap scheduling function" in "Chapter 2 Details of the Job Operation Management Function" in "Job Operation Software Administrator's Guide for Job Management."

Chapter 3 Power Management Function Settings

This chapter describes power management function settings.

The following configuration files are used to make power management function settings.

Table 3.1 Power Management Function and Configuration Files

Function	Configuration File Name
System power collecting/visualization support function	papwrm.conf extdev Command for collecting external equipment power consumption (any file name) PowerGroup node list file (any file name) PowerGroup external equipment list file (any file name) syspwr_db.conf
Power-saving function	papwrm.conf pmpjm.conf pmrsc.conf Job ACL configuration (any file name)
Power API function	
Capping function	papwrm.conf pmpjm.conf pmrsc.conf

Table 3.2 Setting tools for configuration files

Configuration File and Command Name	Path	Setting Tool
papwrm.conf	/etc/opt/FJSVtcs/papwrm.conf	papwrmgradm command
extdev	/etc/opt/FJSVtcs/pwrm/extdev	papwrmgradm command
Command for collecting external equipment power consumption Any file name	Any path	papwrmgradm command
PowerGroup node list file Any file name	Directly under /etc/opt/FJSVtcs/pwrm/	papwrmgradm command
PowerGroup external equipment list file Any file name	Directly under /etc/opt/FJSVtcs/pwrm/	papwrmgradm command
syspwr_db.conf	/etc/opt/FJSVtcs/pwrm/syspwr_db.conf	Edit the file.
pmpjm.conf	/etc/opt/FJSVtcs/Rscunit.d/*resource unit name*/pmpjm.conf	pmpjmadm command
pmrsc.conf	/etc/opt/FJSVtcs/Rscunit.d/*resource unit name*/pmrsc.conf	pmrscadm command
Job ACL configuration Any file name	Any path	pmjacladm command

The node on which you create the configuration file is a system-managed node. Create a configuration file and execute configuration commands with administrator privileges.

In addition, it is necessary to configure the database (MariaDB) to use the system power collecting/visualization support function and the job power estimate function. For details, see "3.3.5 Settings for the System Power Database" and "3.5.2 Settings for the job power estimate database."

Note

- After editing papwrm.conf, extdev, PowerGroup node list file, PowerGroup external equipment list file, or syspwr_db.conf, confirm that the file is owned by the user root and the group root and that the access permission is 0600. This is necessary to prevent the configuration file from being manipulated by malicious users.
- Allow the root privileges to execute a command for collecting external equipment power consumption.
- For details on pmpjm.conf, pmrsc.conf, and job ACL configuration, see "Job Operation Software Administrator's Guide for Job Management."

Information

Checking the system configuration

When setting the power management function, the administrator needs to know the configuration of the system, including cluster names and resource unit names. The configuration of the system can be checked by using the pashowclst command (with the --rscunit option specified). For details on how to check the system configuration by using the pashowclst command, see "Checking the System Configuration" in "Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management" and "Displaying System Configuration Information" in "Chapter 3 Details of the System Management Function" in "Job Operation Software Administrator's Guide for System Management."

3.1 How to Code Configuration Files

The administrator codes configuration files (papwrm.conf, pmpjm.conf, pmrsc.conf, and job ACL configuration) in the following format.

```
section name {
    item 1 = setting value 1
    item 2 = setting value 2
    subsection name {
        item 3 = setting value 3
        item 4 = setting value 4
    }
}
```

The section name is a keyword that indicates what structure definition the part enclosed in curly brackets {} is. The name is determined according to the definition contents. Some definition items contain subsections in their sections.

Note the following at the time of coding:

- A curly bracket "{" that indicates the start of a section and the section or subsection name needs to be separated by one or more space or tab characters. They need to be written on the same line. A curly bracket "}" that indicates the end of a section needs to be written on a separate line.
- The same definition item cannot be written multiple times in the same section.
- The configuration files can contain only single-byte alphanumeric characters and signs.
- A zero-character string ("") cannot be written as a setting value.
- A line that begins with a pound sign "#" or the part of a line after a pound sign "#" is considered a comment.
- A set value is enclosed in double quotes (").

3.2 Setting Example for the papwrm.conf File

Make settings for the following functions in the papwrm.conf file:

Settings for the system power collecting/visualization support function

SystemPower section

2.1 System Power Collecting/Visualization Support Function

Settings for the automatic compute node power control function

PowerSave section

2.2.1 Automatic Compute Node Power Control Function [PG]

Settings for the job power estimate function

JobPowerEstimation section

2.4.1 Power Cap Scheduling Function

Settings for the power knob operation function

PowerKnob section

2.2.2 Power Knob Operation Function with the Job Operation Software [FX]

2.3 Power API Function

The following is a setting example for the papwrm.conf file.

```
SystemPower {
    StartTime = "50"
    LogLevel = "1"
    CommandLine = "/usr/sbin/cmd arg"
    AcceptableRange = "600"
    PowerGroup {
        PowerGroupName = "pwrgrp1"
        ClusterName = "cluster"
        NodeList = "sysnodelist1.txt"
        ExternalDeviceList = "extlist1.txt"
    }
}
PowerSave {
    Cluster {
        ClusterName = "cluster"
        LogLevel = "1"
        ResourceUnit {
            ResourceUnitName = "unit1"
            NodeList = "savenodelist1.txt"
        }
    }
}
JobPowerEstimation {
    Cluster {
        ClusterName = "cluster"
        LogLevel = "1"
        ResourceUnit {
            ResourceUnitName = "unit1"
            DefaultNodePower = "160"
            ErrorNoticeMode = "0"
        }
        DbAuth = "pwr:password"
    }
}
PowerKnob {
    Cluster {
        ClusterName = "cluster"
        LogLevel = "1"
        ResourceUnit {
            ResourceUnitName = "unit1"
```

```
AllowSharedKnob = "0"
ComputeNode {
    IdleState {
        freq = "1600"
        throttling_state = "9"
        issue_state = "1"
        ex_pipe_state = "1"
        eco_state = "2"
        retention_state = "1"
        retention_state_acores = "1"
    }
    RunningState {
        freq = "2000"
        throttling_state = "0"
        issue_state = "0"
        ex_pipe_state = "0"
        eco_state = "0"
        retention_state = "0"
        retention_state_acores = "0"
        freq_min = "1600"
        freq_max = "2000"
        throttling_state_min = "0"
        throttling_state_max = "1"
        issue_state_min = "0"
        issue_state_max = "1"
        ex_pipe_state_min = "0"
        ex_pipe_state_max = "1"
        eco_state_min = "0"
        eco_state_max = "2"
        retention_state_min = "0"
        retention_state_max = "1"
    }
}
IONode {
    IdleState {
        freq = "2000"
        throttling_state = "0"
        issue_state = "0"
        ex_pipe_state = "0"
        eco_state = "0"
        retention_state = "0"
        retention_state_acores = "0"
    }
    RunningState {
        freq = "2000"
        throttling_state = "0"
        issue_state = "0"
        ex_pipe_state = "0"
        eco_state = "0"
        retention_state = "0"
        retention_state_acores = "0"
        freq_min = "1600"
        freq_max = "2000"
        throttling_state_min = "0"
        throttling_state_max = "1"
        issue_state_min = "0"
        issue_state_max = "1"
        ex_pipe_state_min = "0"
        ex_pipe_state_max = "1"
        eco_state_min = "0"
        eco_state_max = "2"
        retention_state_min = "0"
        retention_state_max = "1"
```

```
                }
            }
        }
    }
}
```

For details on the setting items, see the following configuration method for each function or the man page of the papwrm.conf file.

Information

Edit the papwrm.conf file installed in the directory /etc/opt/FJSVtcs/ on the system management node.

The following describes the configuration method for each function. Unless otherwise noted, the administrator shall make the settings from the system management node.

3.3 Settings for the System Power Collecting/Visualization Support Function

This section describes settings for the system power collecting/visualization support function.

The system power collecting/visualization support function provides the following functions:

a. Function to collect power consumption from compute nodes and external equipment

b. Function to calculate average power consumption and total power consumption

c. Visualization support function (pasyspwr command and system power visualization support API)

To use the system power collecting/visualization support function, make the following settings:

- Settings for the system power collecting/visualization support function
  Make these settings in a SystemPower section in the papwrm.conf file.
  - Settings for system power collecting
  - Settings for the function to calculate a sum of power consumption
  - Settings for external equipment
  - Settings for a power group

  If the SystemPower section is omitted, system power collecting is stopped. When the power group has been set, apply the settings which is omitted PowerGroup subsection from SystemPower section. Next, check that the pasyspwr command does not output the power group information. After that, apply the settings which is omitted SystemPower section.
- Settings for the system power database
  MariaDB is used for internal processing.

Note

Only one SystemPower section can be set in the papwrm.conf file. If two or more are set, an error occurs when the settings are attempted to be applied, and they are not applied (see "3.7 Applying and Viewing the papwrm.conf File").

3.3.1 Settings for System Power Collecting

In the SystemPower section, make settings related to power consumption information collection from compute nodes and external equipment and the calculation function.

Table 3.3 Setting Items of System Power Collecting

Section Name	Item Name	Definition Contents	Specifiable Value	Default Value
SystemPower	StartTime	Number of seconds at which power consumption information collection and computing should start. Power consumption information collection starts at the specified time (second) of every minute. For example, if this value is 40 seconds, the collection starts when the specified second value of the time is reached (at the 40th second of the 00th minute, the 40th second of the 01st minute, and so on).	Integer from 0 to 59	30
	LogLevel	Log level 1: Outputs only information messages 2: Debug-level messages Debug information for which output is reduced as much as possible in consideration of system load 3: Debug-level messages Most detailed debug information output without consideration of system load	1, 2, or 3	1

3.3.2 Settings for the Function to Calculate a Sum of Power Consumption

In the SystemPower section, make settings for the function to calculate a sum of power consumption.

Table 3.4 Setting item for the function to calculate a sum of power consumption

Section Name	Item Name	Definition Contents	Specifiable Value	Default Value
SystemPower	AcceptableRange	Acceptable range (in seconds) of variation in the measurement time of power consumption used for calculating a sum	Integer from 0 to 3600	600

Note

As a guide, set a value that is equal to or greater than 60 seconds (i.e., the operation interval of system power collecting) and also the smallest number of seconds among power measurement intervals for external equipment.

3.3.3 Settings for External Equipment

As described in "2.1.1 Power Collecting Function," the system power collecting/visualization support function does not have a function to directly access external equipment and collect power consumption information. Therefore, it is necessary to register external equipment and create and register a command for collecting external equipment power consumption corresponding to the external equipment.
This section describes how to register external equipment and how to create and register a command for collecting external equipment power consumption.

3.3.3.1 How to register external equipment

Write the external equipment name to be registered in the following file on the system management node:

```
/etc/opt/FJSVtcs/pwrm/extdev
```

The following is a setting example for the extdev file. facility1, facility2, and facility3 are examples of external equipment names set by the administrator.

```
facility1
facility2
facility3
```

Edit the installed extdev file in the /etc/opt/FJSVtcs/pwrm/ on the system management node. Do not delete the extdev file.

An external equipment name is a character string consisting of 1 to 63 alphanumeric characters (upper/lower case characters) and hyphens (-) or underscores (_) (However, a hyphen cannot be specified at the beginning of the string). Write the name in the extdev file. Multiple external equipment names can be written by separating them with line breaks. Up to 10,000 external equipment names can be registered. If an invalid external equipment name is specified, an external equipment name is duplicated, or the number of written external equipment names exceeds 10,000, an error message appears when the settings are applied by papwrmgradm. In this case, the settings are not applied.

The pasyspwr command with the --trace option (without the --data option) can output external equipment names of only up to 10 characters. Set external equipment names to 10 characters or less, or specify with the --data option.

3.3.3.2 How to Create a Command for Collecting External Equipment Power Consumption

The command for collecting power consumption of external equipment must satisfy the following conditions:

- Execution condition
  Allow the root privileges to execute the command from the active system management node. Any name can be given to the command. For details on how to register the command, see "3.3.3.3 How to Register a Command for Collecting External Equipment Power Consumption."

- Functional specifications
  Output the following information in the text format defined by the system power collecting/visualization support function:

  a. Integral power consumption from when the watt-hour meter of the measurement target external equipment is reset to when measurement is performed

  b. Momentarily measured momentary power consumption of the measurement target external equipment

  c. Measured time

  If both information a, and information b of the target external equipment can be measured, output two values. If only one of them can be measured, output a measurable value. If multiple external equipment pieces are registered, output their respective power consumption information.

- Output format
  Output power consumption information in text in the following format to the standard output.

```
extdev_name, type, value, m_time[, ser_no]
```

  If you output both integral power consumption and momentary power consumption, output them to two separate lines. If multiple external equipment pieces are registered, use as many lines as the number of the external equipment pieces to output their information.

  The following shows details of the output items of a command for collecting external equipment power consumption.

Table 3.5 Output items of a command for collecting external equipment power consumption

Item Name	Description	Format
extdev_name	External equipment name Output the external equipment name registered in "3.3.3.1 How to register external equipment."	Character string
type	Output type ENE: Integral power consumption PWR: Momentary power consumption For 1 output line, write either ENE or PWR.	Character string ENE or PWR

Item Name	Description	Format
value	Value When type is ENE, integral power consumption (Ws) is output. When type is PWR, momentary power consumption (W) is output.	Integer from 0 to 10^{18}
m_time	Measurement time (seconds) It must be synchronized with the time on the system management node.	UNIX time
ser_no	Serial number (optional) Serial numbers by output type are assigned to even the same external equipment name. This item is optional, and does not affect collecting.	Integer from 0 to 18446744073709551615 (2^{64}-1)

In the case of output across multiple lines, it is handled as power consumption information according to the output order. If the external equipment name, output type, and measurement time of a power consumption information line to be output duplicate those of another line, the later output is collected.

For example, suppose that the following information is output.

```
facility1,ENE,12345600,1454288400,10001
facility1,ENE,12346000,1454288400,10001
facility2,ENE,55555000,1454288430,10001
facility1,ENE,12346600,1454288460,10003
```

The following information is collected:

- External equipment facility1
  Integral power consumption at measurement time 1454288400: 12346000 Ws (Note)
  Integral power consumption at measurement time 1454288460: 12346600 Ws

  (Note) The information in the first line (facility1,ENE,12345600,1454288400,10001) is the same as the information in the second line in terms of the time and output type. Therefore, the information in the first line is overwritten by the information in the second line, which is output later.

- External equipment facility2
  Integral power consumption at measurement time 1454288430: 55555000 Ws

The following is a sample script, "sample.sh", for a command to collect the momentary power consumption of an external equipment "facility1."

Momentary power consumption is obtained from an external equipment "facility1" (IP: 172.17.1.154) using the ipmitool command, and the output is adjusted to the output format of the command for collecting power consumption of the external equipment.

```
#!/bin/sh
# Command Sample Script for Collecting Power Consumption of External Equipment
#   Obtain momentary power consumption of external equipment "facility1" (IP: 172.17.1.154) with
#  the ipmitool command
#    Use the power value of the item name "Total Power Out" in the information output by ipmitool
#  as the momentary power consumption of the external equipment "facility1".
# Measurement time UNIX time
m_time=`date +%s`

# external equipment name: facility1
extdev_name="facility1"

# Output type: PWR: Momentary power consumption
type="PWR"

# value: Value: momentary power consumption (W)
# Extract momentary power consumption value output to "Total Power Out" item from output result
# of ipmitool
```

```
value=`ipmitool sdr -H 172.17.1.154 -U admin -P admin | grep "^Total Power Out.*Watts" |
 awk -F" " '{print $5}'`

# Momentary power consumption of the external equipment "facility1" is output in accordance with
# the output format of the command for collecting power consumption of the external equipment
echo "$extdev_name,$type,$value,$m_time"
```

The result of executing Sample Script "sample.sh" above is as follows:.

```
# ./sample.sh
facility1,PWR,108,1565088897
```

The momentary power consumption of the external equipment "facility1" at Unixtime 1565088897 is 108 W.

- Create a command for collecting external equipment power consumption in such a way that the power consumption information of the multiple external equipment pieces written in /etc/opt/FJSVtcs/pwrm/extdev is output at one time. This command can be created with sh script, python script, etc.
- A single execution of a command for collecting external equipment power consumption can collect the results of up to 10 measurements (each performed at different times). If 11 or more measurement results are output, the results of the latest 10 measurements are collected in order of newest measurement time.
- Only one command for collecting external equipment power consumption can be registered in accordance with "3.3.3.3 How to Register a Command for Collecting External Equipment Power Consumption." If you need to execute multiple commands, register a wrapper script that executes the commands together.
- Arguments that can be given to a command for collecting external equipment power consumption can be specified as fixed values. The external equipment name, measurement time, serial number, etc. cannot be given as arguments. For details, see "3.3.3.3 How to Register a Command for Collecting External Equipment Power Consumption."
- Ensure that a command for collecting external equipment power consumption outputs the difference from the power consumption information that is output last time (one minute earlier). If there are no changes from the last output information, do not output the new information. (Do not output the same information as the last output information.)

3.3.3.3 How to Register a Command for Collecting External Equipment Power Consumption

Register a command for collecting external equipment power consumption in the SystemPower section in the papwrm.conf file.

Table 3.6 Setting Items of a Command for Collecting External Equipment Power Consumption

Section Name	Item Name	Definition Contents	Specifiable Value	Default Value
SystemPower	CommandLine	Command for collecting external equipment power consumption (full path and arguments of the command) (Note 1) (Note 2) This item can be omitted. However, if external equipment is registered in the extdev file, this item is required.	Full path of the command: Character string valid as a file full path name, consisting of up to 255 characters. Arguments for the command can also be specified. If any arguments are specified, the total number of characters for the arguments and command full path is up to 1023. Only one command can be specified.	None

(Note 1)
Register the command created by the procedure in "3.3.3.2 How to Create a Command for Collecting External Equipment Power Consumption."

(Note 2)
When the system management node is redundantly configured, place the same command for collecting external equipment power consumption on the standby system management node. If the node has no command for collecting external equipment power consumption or the contents of the command are different, the power management function does not operate properly.

3.3.4 Settings for a Power Group

Make settings for a power group in a PowerGroup subsection in the SystemPower section.
To set multiple power groups, register multiple PowerGroup subsections. Up to 10,000 PowerGroup subsections can be registered.

Table 3.7 Setting Items of a Power Group

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
PowerGroup	PowerGroupName	Power group name The power group name cannot duplicate power group names specified in other PowerGroup subsections. Such duplication causes an error. This item cannot be omitted.	Character string consisting of 1 to 63 alphanumeric characters (upper/lower case characters) and hyphens (-) or underscores (_) However, a hyphen (-) cannot be specified at the beginning of the string.	Not omissible
	ClusterName	Compute cluster name to be registered in the power group This item can be omitted, but is required when registering compute nodes in a power group.	Compute cluster name	None
	NodeList	Name of the file containing the node IDs of the compute nodes to be registered in the power group Specify the file name excluding the directory name (/etc/opt/FJSVtcs/pwrm/). If this item is specified, only the specified compute nodes are registered in the power group. If this item is omitted and the ClusterName item is specified, all the compute nodes belonging to the compute cluster specified by ClusterName are registered in the power group. If this item is specified, the ClusterName item cannot be omitted.	File name consisting of 1 to 255 characters. However, the file name cannot contain commas (,) or slashes (/).	None
	ExternalDeviceList	Name of the file containing external equipment names to be registered in the power group Specify the file name excluding the directory names (/etc/opt/FJSVtcs/pwrm/). If no external equipment is to be specified for the power group, this item can be omitted.	File name consisting of 1 to 255 characters However, the file name cannot contain commas (,) or slashes (/).	None

The following is a setting example for the file set in the item NodeList. The node IDs of the compute nodes specified here are registered in the power group.

```
0x01010001
0x01020001-0x0102000F
```

Place the file directly under /etc/opt/FJSVtcs/pwrm/ on the system management node. Confirm that the file owner is the user root, the group root, and that the access permission is 0600.

Node IDs of multiple compute nodes can be written in the file by separating them with line breaks. A hyphen (-) can be used to specify a range (e.g., 0x01020001-0x0102000F).If the node ID of a compute node does not belong to the compute cluster specified by ClusterName or it duplicates within the same power group, an error occurs. However, if a range of node IDs of compute nodes is specified and a node ID that does not belong to the compute cluster specified by ClusterName is within the specified range, the error does not occur.

The node ID of the same compute node can be specified for multiple power groups.
The number of compute nodes that can be specified for a single power group is equal to the maximum number of compute nodes in the compute cluster.

The following is a setting example for the file set in the item ExternalDeviceList. The external equipment names specified here are registered in the power group.

```
facility1
facility2
```

Place the file directly under /etc/opt/FJSVtcs/pwrm/ on the system management node. Confirm that the file owner is the user root, the group root, and that the access permission is 0600.

Multiple external equipment names can be written in the file by separating them with line breaks. Only the external equipment names registered in /etc/opt/FJSVtcs/pwrm/extdev can be written. If a written external equipment name is not written in the external equipment registration file (/etc/opt/FJSVtcs/pwrm/extdev) or it duplicates within the same power group, an error occurs.

The same external equipment name can be specified for multiple power groups.
The maximum number of external equipment pieces that can be specified for a single power group is 10,000.

Note

- In NodeList and ExternalDeviceList, specify a file name alone excluding directory names. For example, to specify /etc/opt/FJSVtcs/pwrm/sysnodelist1.txt in NodeList, write NodeList = "sysnodelist1.txt" in papwrm.conf.
- Devices to be registered in a power group must be capable of collecting power consumption. For example, if you want to register a compute cluster with compute nodes that have never collected power consumption in the power group, in addition to setting ClusterName, you should set NodeList that specifies only compute nodes that can collect power consumption.
  If you register a device that cannot collect power consumption in the power group, you cannot output past power information by pasyspwr command with --trace option. If so, delete the power group. Then, use the pasyspwr command with -v option to determine which devices can collect power consumption, and then register them in the power group.
- The pasyspwr command with the --trace option (without the --data option) can output power group names of only up to 10 characters. Set power group names to 10 characters or less, or specify with the --data option.

3.3.5 Settings for the System Power Database

The system power collecting/visualization support function has the system power database to store power information. MariaDB is used for internal processing. Perform the following work only once at the time of installation.

See

MariaDB must be installed in advance on the compute cluster management node. For the procedure for installing MariaDB, see "Performing MariaDB-related Work for Job Operations" in "Chapter 2 New System Installation" in "Job Operation Software Setup Guide."

First, register an account for the system power database.
The following is an example where the user name is **syspwr** and the password is **password** for the system power database. The items to be entered are underlined.

```
[Compute cluster management node]
# mysql -u root -p
Enter password: password of root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.50-MariaDB MariaDB Server

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database syspwr;
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> grant all on syspwr.* to syspwr identified by 'password';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit;
Bye
```

Note

Use a user name that is different from user names set for other databases that use MariaDB. The database name is syspwr. This database name cannot be changed.

Next, write the user name and password for the system power database, and the representative network for the compute cluster management node in the configuration file on the system management node. The following file is already installed. Edit relevant parts.

```
[/etc/opt/FJSVtcs/pwrm/syspwr_db.conf]

$username = "syspwr";
$password = "password";
$server = "The representative network for the compute cluster management node"
```

You can determine the representative network of a compute cluster management node by executing "pashowclst -l -v --nodetype CCM".

Note

When the system management node is redundantly configured, place the same file on the standby system management node after editing the file. If the node does not have the file or the contents of the file are different, the system power collecting/visualization support function does not operate properly.

3.3.6 Disabling for the System Power Collecting/Visualization Support Function

This section describes disabling for the System Power Collecting/Visualization Support Function.

1. Delete external equipment and power group names in configuration files

   If you have registered external device names, delete all external device names from the external device registration file /etc/opt/FJSVtcs/pwrm/extdev and delete CommandLine in the SystemPower section. If you are configuring power groups, delete the PowerGroup subsection from the SystemPower section in the Power Management Facility configuration file named papwrm.conf.

2. Apply changes effective on the system

   Apply your changes effective. See "3.7.1 Applying Settings" for the procedure to apply the settings.

   After the completion of these changes, run "pasypwr" command on the system managed node to verify that the registered external device names and power group names are no longer shown on the screen as follow:

   ```
   # pasyspwr --extdev all -v
   # pasyspwr --pwrgrp all
   ```

3. Delete SystemPower section

   Delete SystemPower section in the configure file papwrm.conf.

4. Apply the setting for deleting SystemPower section

   Apply the deleted SystemPower section configuration file. See "3.7.1 Applying Settings."

3.4 Settings for the Automatic Compute Node Power Control Function

3.4.1 Settings for the Automatic Compute Node Power Control Function

This section describes settings for the automatic compute node power control function.

The automatic compute node power control function provides the following two functions:

a. Function to stop power nodes to which no jobs are allocated for a long time

b. Function to start stopped power nodes in such a way that they are in time for the start of job execution

To use the automatic compute node power control function, settings need to be made in the PowerSave section in papwrm.conf.

In the PowerSave section, make settings for a compute cluster (Cluster subsection) and a resource unit under each compute cluster (ResourceUnit subsection). The following settings can be made in the PowerSave section:

- Enabling/Disabling the automatic compute node power control function
- Setting the conditions under which the compute node is powered down or started

If the PowerSave section is omitted, the automatic compute node power control function is disabled. If you are configuring multiple clusters, configure the Cluster subsection separately for each cluster. If you are configuring multiple resource units, configure them separately for each ResourceUnit subsection resource unit as well.

Note

Only one PowerSave section can be set in the papwrm.conf file. If two or more are set, an error occurs when the settings are attempted to be applied, and they are not applied (see "3.7 Applying and Viewing the papwrm.conf File").

Table 3.8 Setting Items in the Cluster Subsection

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
Cluster	ClusterName	Compute cluster name This item is a required setting. If a non-existing compute cluster name is specified, an error occurs.	Compute cluster name	Not omissible
	LogLevel	Log level 1: Outputs only information messages 2: Debug-level messages Debug information for which output is reduced as much as possible in consideration of system load 3: Debug-level messages Most detailed debug information output without consideration of system load	1, 2, or 3	1

Table 3.9 Setting Items in the ResourceUnit Subsection in the Cluster Section

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
ResourceUnit	ResourceUnitName	Resource unit name	Resource unit name	Not omissible

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
		This item is a required setting. If a non-existing resource unit is specified, an error occurs.		
	NodeList	Name of the file containing the node IDs of compute nodes that should be targets of the automatic compute node power control function Specify the file name excluding the directory names (/etc/opt/FJSVtcs/pwrm/). Place the file directly under /etc/opt/FJSVtcs/pwrm/ on the system management node. Node IDs of multiple compute nodes can be written in the file by separating them with line breaks. A hyphen (-) can be used to specify a range (e.g., 0x01FF0001-0x01FF0010). If an invalid node ID of a compute node is specified or node IDs of compute nodes duplicate, an error occurs. However, if a range of node IDs of compute nodes is specified, the error does not occur. If this item is omitted, all the compute nodes in the resource unit become targets of the automatic power control function.	File name consisting of 1 to 255 characters However, the file name cannot contain commas (,) or slashes (/).	None

In NodeList, specify a file name alone excluding directory names. For example, to specify /etc/opt/FJSVtcs/pwrm/savenodelist1.txt in NodeList, write NodeList = "savenodelist1.txt" in papwrm.conf.

3.4.2 Disabling for the Automatic Compute Node Power Control Function

This section describes how to disable the Automatic Compute Node Power Control Function.

1. Delete PowerSave section

   Delete PowerSave section in papwrm.conf.

2. Apply the papwrm.conf you modified to the compute cluster management nodes

   Apply the papwrm.conf you modified to the compute cluster management node using papwrmgradm command. When you run the command, specify the file that contains the node IDs of the compute cluster management nodes in the -f option. If the compute cluster contains redundant nodes, type this command for each node including a standby-node.

   ```
   [System management node]
   # papwrmgradm --set -c ClusterName -f NodeIDListOfComputeClusterManagementNodes
   ```

3. Start stopped compute nodes

   Start compute nodes which were stopped by the automatic compute node power control function. To identify compute nodes with automatic power control disabled, see "4.3 Checking the Operation Status of the Compute Node Automatic Power Control Function."

4. Delete the deadline schedules

   Delete the deadline schedules set by the automatic compute node power control function. To check and delete the deadline schedule which was set by automatic compute node power control function, use the following command.

   ```
   [Compute cluster management node]
   # padeadline --show
   ```

```
NO     TYPE START               END                 TARGET
231    f    2019-10-29 17:12:07 2019-10-29 18:12:59 AUTOPWRCTL
# padeadline -c clst1 --cancel 231
[WARNING]
Do you really want to continue (y/n)?
y
[INFO] PJM 6200 padeadline Deadline-schedule 231 canceled.
```

The deadline schedule number with AUTOPWRCTL as TARGET is the one set by the automatic compute node power control function. In the above example, the deadline schedule number 231 set by the automatic compute node power control function is deleted.

5. Apply the setting for deleting PowerSave section

   Apply the deleted PowerSave section configuration file. See "3.7.1 Applying Settings."

3.5 Settings for the Job Power Estimate Function

This section describes settings for the job power estimate function of the power cap scheduling function.

- "Power cap scheduling function (job power estimate function)" that schedules jobs so they do not exceed the system-wide power limit by predicting their power consumption

To use the job power estimate function, the following settings need to be made:

- Settings for the job power estimate library
  To use the job power estimate function, make settings for the job power estimate library. Make these settings in the papwrm.conf file. For more information, see "3.5.1 Settings for the Job Power Estimate Library."

- Settings for the job power estimate database
  Make settings for the database that stores power information. Make these settings from the compute cluster management node by using a command supplied with MariaDB. For more information, see "3.5.2 Settings for the job power estimate database."

- Settings for custom resources
  Set power as a resource for job execution (custom resource "sys-power"). Make this setting in the configuration file pmpjm.conf of the job operation management function in a resource unit of the job operation management function. The administrator needs to set a custom resource as the power consumption of the system in advance.
  The following is an example of writing a custom resource in pmpjm.conf. In this example, 1000 W is set for the target resource unit (unit1) as power to be allocated to all the jobs that simultaneously run within the resource unit.

```
Change the code in /etc/opt/FJSVtcs/Rscunit.d/unit1/pmpjm.conf
for the target resource unit on the system management node.

ResourceUnit {
    ResourceUnitName = unit1
    CustomResource {          <- Add
        Name = sys-power      <- Add
        ValueType = numeric   <- Add
        Value = 1000          <- Add
    }                         <- Add
}
```

  A sys-power is a custom resource for the power of the node where the job runs. This resource does not include the power used by nodes that are not running jobs. For example, if the number of nodes is 1000 nodes, and you want the power per 1 compute node that is not running a job to be 10 W, and you want the power of the entire compute node to be 1,000,000W, set sys-power to 990,000 W (= 1000000 W -1000 nodes x 10 W), excluding the power consumed by nodes that are not running jobs.

  The power value of the compute node where the job is not running reflects papwrm.conf, then the power value of the /etc/opt/FJSVtcs/pwrm/base.*Resource_unit_name*.

- Settings for the job scheduler exit function
  Make settings for the job scheduler exit function. For details on the setting method, see "Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function."
  Before starting operation, submit a test job and check if the hook setting is correctly reflected.

- Reflect the settings
  Reflects the configured pmpjm.conf. For more information, see "Chapter 3 Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management."

```
# pmpjmadm --set -c ClusterName --ru ResourceUnitName
```

  To verify the settings as follows.

```
# pjshowrsc -c ClusterName --ru ResourceUnitName -C
```

Information

The job operation management function enables the job operation administrator to define any resource as a job resource to generically schedule and run jobs according to various uses. This resource is called a "custom resource." The power consumption of a system is one of these custom resources.

- For details on custom resources, see "Job scheduling function using custom resources" in "Chapter 2 Details of the Job Operation Management Function" in "Job Operation Software Administrator's Guide for Job Management."
- For details on the custom resource setting method, see "Custom resource settings" in "Chapter 3 Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management."

3.5.1 Settings for the Job Power Estimate Library

In the setting section JobPowerEstimation for the power estimate library, make settings for a compute cluster (Cluster subsection) and a resource unit under each compute cluster (ResourceUnit subsection). If the JobPowerEstimation section is omitted, the job power estimate function is disabled. If you are setting up multiple clusters, set the Cluster subsection for each cluster. If you are configuring multiple resource units, set the ResourceUnit subsection for each resource unit as well.

Note

Only one JobPowerEstimation section can be set in the papwrm.conf file. If two or more are set, an error occurs when the settings are attempted to be applied, and they are not applied. For more information about configuring the papwrm.conf file, see "3.7 Applying and Viewing the papwrm.conf File."

Table 3.10 Setting Items in the Cluster Subsection

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
Cluster	ClusterName	Compute cluster name This item cannot be omitted. If a non-existing cluster name is specified, an error occurs.	Compute cluster name	Not omissible
	LogLevel	Log level 1: Outputs only information messages 2: Debug-level messages Debug information for which output is reduced as much as possible in consideration of system load 3: Debug-level messages Most detailed debug information output without consideration of system load	1, 2, or 3	1
	DbAuth	User name and password to connect to the job power estimate database Set the values set in "3.5.2 Settings for the job power estimate database" by connecting them with a colon (e.g., pwr:password) as the user	Character string consisting of 1 or more characters, including a colon (:).	Not omissible

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
		name and password. This item cannot be omitted.		

Table 3.11 Setting Items in the ResourceUnit Subsection in the Cluster Subsection

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
ResourceUnit	ResourceUnitName	Resource unit name This item cannot be omitted. If a non-existing resource unit name is specified, an error occurs.	Resource unit name	Not omissible
	DefaultNodePower	Job power (W) per node (Note) If the job power estimate function cannot estimate job power, the value specified by this item is used. The value specified by this item is also used as the estimated power value for interactive or virtual node allocated jobs. This item cannot be omitted.	Integer from 1 to 100000	Not omissible
	ErrorNoticeMode	Return value when a function of the job power estimate library ends abnormally 0: An abnormal end is reported as the return value. In this case, jobs cannot be accepted. 1: An abnormal end is not reported as the return value. In this case, jobs are accepted.	0 or 1	0

(Note)
We recommend setting the maximum power consumption stated in the specifications in the item DefaultNodePower or actually measuring the power consumption of a typical job submitted to the system and setting the measured value.

The power of nodes where no jobs are running (base power) (W) is automatically acquired and stored in the following file on the system management node when settings are applied by the papwrmgradm command:

```
/etc/opt/FJSVtcs/pwrm/base.ResourceUnitName
```

To measure base power, run the first papwrmgradm command after the system is built with the compute node up and no jobs running on the compute node. If the system managed nodes are redundant, copy the base power file to /etc/opt/FJSVtcs/pwrm/ on the standby system management node.

3.5.2 Settings for the job power estimate database

The job power estimate function uses MariaDB for internal processing.

See

MariaDB must be installed in advance on the compute cluster management node. For the procedure for installing MariaDB, see "MariaDB-related work for job operation" in "Chapter 2 Installing a New System" in "Job Operation Software Setup Guide."

Register an account for the job power estimate database from the compute cluster management node. The following is an example where the user is **pwr** and the password is **password** for the job power estimate database. The items to be entered are underlined.

```
[Compute cluster management node]
# mysql -u root -p
Enter password: password of root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.50-MariaDB MariaDB Server

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database pwr;
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> grant all on pwr.* to pwr@localhost identified by 'password';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit;
Bye
```

Note

Use a user name that is different from user names set for other databases that use MariaDB. The database name is pwr. This database name cannot be changed.

3.5.3 Disabling for the Job Power Estimate Function

This section describes disabling for the Job Power Estimate function.

1. Stop the Job Operation Software service

   Isolate compute nodes with the operation described in "Job Operation Software Administrator's Guide for Maintenance."

2. Delete custom resource sys-power

   Delete custom resource "sys-power" in the configuration file named pmpjm.conf of the job operation management function in a resource unit of the job operation management function.

3. Unset exit functions provided by the job scheduler function

   Delete exit functions provided by the job scheduler function set in "Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function."

   After deleting, you need to execute the pmpjmadm command on system management node so that the system reflects the setting contents.

   ```
   [Compute cluster management node]
   # pmpjmadm --set -c ClusterName --rscunit ResourceUNitName
   ```

   For details on the unsetting method, see "Job manager exit function and Job scheduler exit function " in "Chapter 2 Creating and Incorporating Hooks" in " Job Operation Software Administrator's Guide for Job Operation Manager Hook."

4. Delete JobPowerEstimation section

   Delete JobPowerEstimation section in the configuration file papwrm.conf of the power management function.

5. Apply the setting for deleting JobPowerEstimation section

   Apply the deleted JobPowerEstimation section configuration file. See "3.7.1 Applying Settings."

6. Start the Job Operation Software service

   Incorporate compute nodes with the operation described in "Job Operation Software Administrator's Guide for Maintenance."

3.6 Settings for the Power Knob Operation Function [FX]

This section describes settings for the power knob operation function.

The power knob operation function provides the following two functions for each resource unit:

a. Power knob operation function with the Job Operation Software that sets each administrator-specified power knob value when a job runs or ends to support power saving and performance at the same time ("2.2.2 Power Knob Operation Function with the Job Operation Software [FX]")

b. Function to set ranges within which the Power API (power knob operations) can be used by users during job execution ("2.3.2 Power Knob Operation Restriction Function")

To use the power knob operation function, the following settings need to be made in addition to the settings in the PowerKnob section in papwrm.conf:

- Settings for the job scheduler exit function
  Make settings for the job scheduler exit function (pmpjm.conf). For details on the setting method, see "Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function."

- Settings for custom resources
  Set power knobs that can be used by users during job execution as resources for job execution (custom resources). Make these settings in the configuration file pmpjm.conf of the job operation management function in the resource unit of the job operation management function and the job ACL function.
  For details on the setting items and setting method for custom resources, see "Job operation management function settings in a resource unit (pmpjm.conf file)" in "Chapter 3 Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management."

3.6.1 Rules for setting power knobs

There are two types of rules for setting power knobs: "3.6.1.1 Setting Rules for the Power Knob Operation Function with the Job Operation Software" and "3.6.1.2 Setting Rules for the Power Knob Operation Restriction Function."

3.6.1.1 Setting Rules for the Power Knob Operation Function with the Job Operation Software

This section describes the setting rules for knob operations associated with the start and end of a job.

Make settings in papwrm.conf for each resource unit. Settings for each resource unit include ones for compute nodes (ComputeNode subsection) and for the compute and I/O node (IONode subsection). Settings for compute nodes cannot be omitted.

Settings for compute nodes and for the compute and I/O node include ones for power-saving mode (IdleState subsection) and for performance priority mode (RunningState subsection). Settings for power-saving mode and performance priority mode of compute nodes cannot be omitted.

a. Making the same setting for the compute and I/O node as for compute nodes

   To make the same settings for the compute and I/O node as for compute nodes, omit settings for the compute and I/O node. In this case, settings for compute nodes are used for the compute and I/O node.

b. Making settings for the compute and I/O node that are different from those for compute nodes

   To make settings for the compute and I/O node that are different from those for compute nodes, make settings for the compute and I/O node (IONode subsection). For items that are not written as settings for the compute and I/O node, settings for compute nodes are used.

c. Changing settings for compute nodes for each resource group or user

   To change settings for compute nodes for each resource group or user, set a custom resource in the power knob item to be changed in pmpjm.conf (see "Table 3.18 Resources for the Power Knob Operation Function That Can be Set as Custom Resources in the Job ACL (Power Knob Operation Function With the Job Operation Software)"). Then, make settings for each resource group or user by using the job ACL function.

   For the compute and I/O node, settings in papwrm.conf are used, instead of those in job ACL.

d. Changing common settings among compute nodes and the compute and I/O node for each resource group or user

   To change settings for compute nodes and the compute and I/O node for each resource group or user, set a custom resource in the power knob item to be changed and make a custom resource (knob_io) setting in pmpjm.conf. Then, using the job ACL function, set the power knob item to be changed and specify COMPUTE as a knob_io resource to make custom resource settings for each resource group or user.

e. Setting to allow users to change power knob values for compute nodes when submitting a job

   To change power knob values for compute nodes for each resource group or user at the time of job submission, set custom resources in pmpjm.conf. Then, using the job ACL function, allow each resource group or user to make settings.

When submitting a job, users add custom resource settings (e.g., pjsub **-L freq=2000** run,sh) to the submission.

Power knob values for the compute and I/O node are set in papwrm.conf, and cannot be changed by users.

f. Setting to allow users to change power knob values for compute nodes and the compute and I/O node when submitting a job

To change power knob values for compute nodes for each resource group or user at the time of job submission, set custom resources and make a custom resource (knob_io) setting in pmpjm.conf. Then, using the job ACL function, set the power knob item to be changed and specify COMPUTE as a knob_io resource to make custom resource settings for each resource group or user. When submitting a job, add custom resource settings (e.g., pjsub **-L freq=2000** run.sh) to the submission.

If you use the job ACL to allow settings on a per-resource group or per-user basis, make sure that you are aware of the custom resources that you have allowed and how they are submitted.

3.6.1.2 Setting Rules for the Power Knob Operation Restriction Function

This section describes the setting rules for ranges within which users can perform power knob operations, during execution of a job they submit, by using the Power API within the job. The setting items for a setting range have "_max" (upper limit) or "_min" (lower limit) appended. If the Power API requests a change to a power knob value that is out of range, an error occurs.

For frequency, for example, the upper bound is freq_max and the lower bound is freq_min.

If you set freq_min = 1600 (MHz) and freq_max = 2000 (MHz), and then use the Power API to request that the frequency be changed to 2200 (MHz), an error occurs.

If upper and lower limits are not set, the same values as power knob setting values are used as the upper and lower limits. For example, if freq=2000 is set and no values set in freq_min and freq_max, freq_min=2000 and freq_max=2000 are used.

a. Enabling all users to perform power knob operations within a job by using settings common to compute nodes and the compute and I/O node

To enable users to perform power knob operation within a job, set the upper and lower limits for power knobs in the RunningState subsection in the ComputeNode subsection in papwrm.conf. Since these are common settings, they are not defined in the IONode subsection.

b. Enabling all users to perform power knob operations within a job by using different settings in compute nodes and the compute and I/O node

To make settings for the compute and I/O node different from those for compute nodes, set the upper and lower limits for power knobs in the RunningState subsection in the IONode subsection.

c. Changing settings for compute nodes for each resource group or user

To change settings for compute nodes for each resource group or user, set custom resources for upper and lower limits in pmpjm.conf (see "Table 3.19 Resources for the Power Knob Operation Function That Can be Set as Custom Resources in the Job ACL (Power Knob Operation Restriction Function)"). Then, make settings for each resource group or user by using the job ACL function. Settings for the compute and I/O node cannot be changed from papwrm.conf, regardless of the knob_io setting.

3.6.2 Settings in the Power Management Configuration File of the Power Knob Operation Function

In the setting section PowerKnob for the power knob operation function, make settings for a compute cluster (Cluster subsection), a resource unit under each compute cluster (ResourceUnit subsection), a compute node under the resource unit (ComputeNode and IONode subsections), and a state section under the compute node (IdleState and RunningState subsections). If you are setting up multiple clusters, set the Cluster subsection for each cluster. If you are configuring multiple resource units, set the ResourceUnit subsection for each resource unit as well.

If the PowerKnob section is omitted, power knobs are not used to control power at the start and end of jobs, and ranges within which the Power API can be used by users are not set.

In the case of FX server, power knob settings can be made in each of compute nodes and the compute and I/O node. In FX server, the compute and I/O node may relay I/O for other jobs. Changing power knobs for the compute and I/O node to power-saving mode affects I/O for other jobs that is relayed by such the compute and I/O node. For this reason, the power knob settings can be made in compute nodes, and the compute and I/O node, respectively.

Note

Only one PowerKnob section can be set in the papwrm.conf file. If two or more are set, an error occurs when the settings are attempted to be applied, and they are not applied (see "3.7 Applying and Viewing the papwrm.conf File").

Table 3.12 Setting Items in the Cluster Subsection

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
Cluster	ClusterName	Compute cluster name This item cannot be omitted. If a non-existing compute cluster is specified, an error occurs.	Compute cluster name	Not omissible
	LogLevel	Log level 1: Outputs only information messages 2: Debug-level messages Debug information for which output is reduced as much as possible in consideration of system load 3: Debug-level messages Most detailed debug information output without consideration of system load	1, 2, or 3	1

Table 3.13 Setting Items in the ResourceUnit Subsection in the Cluster Subsection

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
ResourceUnit	ResourceUnitName	Resource unit name This item cannot be omitted. If a non-existing resource unit name is specified, an error occurs.	Resource unit name	Not omissible
	AllowSharedKnob	Whether to allow power knob operations for node-sharing jobs This item is valid only for FX server. 0: Not allow power knob operations 1: Allow power knob operations	0 or 1	0

Note

If the setting value of AllowSharedKnob is set to 1, power knob values for other users' jobs that are executed on the same node are possibly affected in the case of node-sharing jobs. Be aware of this risk before making a setting. For details, see "2.3.2 Power Knob Operation Restriction Function."

Subsection Items in the ComputeNode and IONode Subsections

The ComputeNode and IONode subsections have IdleState and RunningState subsections. In the RunningState subsection, set power knob values during job execution. In the IdleState subsection, set power knob values while there are no jobs. The IdleState and RunningState subsections in the ComputeNode subsection cannot be omitted. The IdleState and RunningState subsections in IONode subsection can be omitted. If omitted, setting values in the ComputeNode subsection are used.

Table 3.14 Section Names in the ComputeNode and IONode Subsections in the ResourceUnit Subsection

Subsection Name	Subsection name	Definition Contents	Default Value
ComputeNode	IdleState	Subsection to set power knob values when compute nodes are idle	Not omissible

Subsection Name	Subsection name	Definition Contents	Default Value
	RunningState	Subsection to set power knob values when compute nodes is executing jobs and to set power knob operation restrictions	Not omissible
IONode	IdleState	Subsection to set power knob values when the compute and I/O node is idle	Omissible If omitted, values in the IdleState subsection in the ComputeNode subsection are used.
	RunningState	Subsection to set power knob values when the compute and I/O node is executing jobs and to set power knob operation restrictions	Omissible If omitted, values in the RunningState subsection in the ComputeNode subsection are used.

Items in the IdleState Subsection

Make settings for items in the IdleState subsection.

For details on each item, see "Job Operation Software API user's Guide for Power API."

Table 3.15 Setting Items in the IdleState Subsection in the ComputeNode and IONode Subsections (Power Knob Operation Function With the Job Operation Software)

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
IdleState	freq	CPU frequency (MHz) A frequency that is allowed by the hardware can be set. This item corresponds to PWR_ATTR_FREQ of the Power API.	Integer from 0 to 10000	Items in the IdleState subsection in the ComputeNode subsection cannot be omitted. If items in the IdleState subsection in the IONode subsection are omitted, values defined in the IdleState subsection in the ComputeNode subsection are used.
	throttling_state	HBM access restriction 0: No restriction (maximum performance) 1: 90% of the number of requests 2: 80% of the number of requests 3: 70% of the number of requests 4: 60% of the number of requests 5: 50% of the number of requests 6: 40% of the number of requests 7: 30% of the number of requests 8: 20% of the number of requests 9: Maximum restriction (10% of the number of requests) This item corresponds to PWR_ATTR_THROTTLING_STATE of the Power API.	Integer from 0 to 9	
	issue_state	Instruction issue restriction for compute cores 0: 4 instructions (maximum performance) 1: 2 instructions (minimum performance) This item corresponds to PWR_ATTR_ISSUE_STATE of the Power API.	0 or 1	
	ex_pipe_state	Number of compute core EXes 0: Use AB (maximum performance) 1: Use only A (minimum performance) This item corresponds to PWR_EX_PIPE_STATE of the Power API.	0 or 1	

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
	eco_state	Eco mode state of compute cores 0: Off, use FLAB (maximum performance) 1: Off, use only FLA 2: On, use only FLA (minimum performance) This item corresponds to PWR_ATTR_ECO_STATE of the Power API.	0, 1, or 2	
	retention_state	Whether to allow compute cores to transition to Retention state 0: Non-Retention mode 1: Retention mode This item corresponds to PWR_ATTR_RETENTION_STATE of the Power API.	0 or 1	
	retention_state_acores	Whether to allow assistant cores other than core 0 (Core 0) to transition to Retention state 0: Non-Retention mode 1: Retention mode retention_state for assistant cores This item corresponds to PWR_ATTR_RETENTION_STATE the Power API.	0 or 1	

Items in the RunningState Subsection

Make settings for items in the RunningState subsection

Table 3.16 Setting Items in the RunningState Subsection in the ComputeNode and IONode Subsections (Power Knob Operation Function With the Job Operation Software)

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
RunningState	freq	CPU frequency (MHz) A frequency that is allowed by the hardware can be set. This item corresponds to PWR_ATTR_FREQ of the Power API.	Integer from 0 to 10000	Items in the RunningState subsection in the ComputeNode subsection cannot be omitted. If items in the RunningState subsection in the IONode subsection are omitted, values defined in the RunningState subsection in the ComputeNode subsection are used.
	throttling_state	HBM access restriction 0: No restriction (maximum performance) 1: 90% of the number of requests 2: 80% of the number of requests 3: 70% of the number of requests 4: 60% of the number of requests 5: 50% of the number of requests 6: 40% of the number of requests 7: 30% of the number of requests 8: 20% of the number of requests 9: Maximum restriction (10% of the number of requests) This item corresponds to PWR_ATTR_THROTTLING_STATE of the Power API.	Integer from 0 to 9	

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
	issue_state	Instruction issue restriction for compute cores 0: 4 instructions (maximum performance) 1: 2 instructions (minimum performance) This item corresponds to PWR_ATTR_ISSUE_STATE of the Power API.	0 or 1	
	ex_pipe_state	Number of compute core EXes 0: Use AB (maximum performance) 1: Use only A (minimum performance) This item corresponds to PWR_EX_PIPE _STATE of the Power API.	0 or 1	
	eco_state	Eco mode state of compute nodes 0: Off, use FLAB (maximum performance) 1: Off, use only FLA 2: On, use only FLA (minimum performance) This item corresponds to PWR_ATTR_ECO_STATE of the Power API.	0, 1, or 2	
	retention_state	Whether to allow compute cores to transition to Retention state 0: Non-Retention mode 1: Retention mode This item corresponds to PWR_ATTR_RETENTION_STATE of the Power API.	0 or 1	
	retention_state_acores	Whether to allow assistant cores other than core 0 (Core 0) to transition to Retention state 0: Non-Retention mode 1: Retention mode retention_state for assistant cores This item corresponds to PWR_ATTR_RETENTION_STATE of the Power API	0 or 1	

Table 3.17 Setting Items in the RunningState Subsection in the ComputeNode and IONode Subsections (Power Knob Operation Restriction Function)

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
RunningState	freq_min	Lower limit on CPU frequency (MHz) Specifiable values are the same as for freq.	Integer from 0 to 10000	Same as for freq
	freq_max	Upper limit on CPU frequency (MHz) Specifiable values are the same as for freq.	Integer from 0 to 10000	Same as for freq
	throttling_state_min	Lower limit on HBM access restriction Specifiable values are the same as for throttling_state.	Integer from 0 to 9	Same as for throttling_state

Subsection Name	Item Name	Definition Contents	Specifiable Value	Default Value
	throttling_state_max	Upper limit on HBM access restriction Specifiable values are the same as for throttling_state.	Integer from 0 to 9	Same as for throttling_state
	issue_state_min	Lower limit on instruction issue restriction for compute cores Specifiable values are the same as for issue_state.	0 or 1	Same as for issue_state
	issue_state_max	Upper limit on instruction issue restriction for compute cores Specifiable values are the same as for issue_state.	0 or 1	Same as for issue_state
	ex_pipe_state_min	Lower limit on the number of compute core EX Settable values are the same as for ex_pipe_state.	0 or 1	Same as for ex_pipe_state
	ex_pipe_state_max	Upper limit on the number of compute core EX Settable values are the same as for ex_pipe_state.	0 or 1	Same as for ex_pipe_state
	eco_state_min	Lower limit on the eco mode state of compute cores Settable values are the same as for eco_state.	0, 1, or 2	Same as for eco_state
	eco_state_max	Upper limit on the eco mode state of compute cores Settable values are the same as for eco_state.	0, 1, or 2	Same as for eco_state
	retention_state_min	Lower limit on whether to allow compute cores to transition to Retention state Settable values are the same as for retention_state.	0 or 1	Same as for retention_state
	retention_state_max	Upper limit on whether to allow compute cores to transition to Retention state Settable values are the same as for retention_state.	0 or 1	Same as for retention_state

3.6.3 Configuration File of the Power Management Function and Job ACL Configuration for the Power Knob Operation Function

By defining the power knob as a custom resource, the value specified for RunningState in the power management configuration file (papwrm.conf) can be changed at job submission. A setting range within which a job's power knob operation restriction can be performed by users can be changed by using the job ACL. Make custom resource settings in pmpjm.conf.

This section describes in detail how to specify power knob values and setting ranges that restrict power knob operations. For settings in the configuration file papwrm.conf of the power management function and ones in the job ACL, values different between them can be specified for the same power knobs. In this case, settings in the job ACL take priority over values written in the configuration file (papwrm.conf) of the power management function.

Table 3.18 Resources for the Power Knob Operation Function That Can be Set as Custom Resources in the Job ACL (Power Knob Operation Function With the Job Operation Software)

Custom Resource Name	Definition Contents	Settable Value
freq	CPU frequency (MHz) A frequency that is allowed by the hardware can be set.	Value allowed by the hardware

Custom Resource Name	Definition Contents	Settable Value
throttling_state	HBM access restriction 0: No restriction (maximum performance) 1: 90% of the number of requests 2: 80% of the number of requests 3: 70% of the number of requests 4: 60% of the number of requests 5: 50% of the number of requests 6: 40% of the number of requests 7: 30% of the number of requests 8: 20% of the number of requests 9: Maximum restriction (10% of the number of requests)	Integer from 0 to 9
issue_state	Instruction issue restriction for compute cores 0: 4 instructions (maximum performance) 1: 2 instructions (minimum performance)	0 or 1
ex_pipe_state	Number of compute core EXes 0: Use AB (maximum performance) 1: Use only A (minimum performance)	0 or 1
eco_state	Eco mode state of compute cores 0: Off, use FLAB (maximum performance) 1: Off, use only FLA 2: On, use only FLA (minimum performance)	0, 1, or 2
retention_state	Whether to allow compute cores to transition to Retention state 0: Non-Retention mode 1: Retention mode	0 or 1
knob_io	Power knob value selection for compute and I/O node SYSTEM: Use values in the configuration file of power management COMPUTE: Use values specified in the job ACL	SYSTEM or COMPUTE If no custom resources are set, operation as SYSTEM is performed.

Use the custom resource "knob_io" to select whether to use values specified in the job ACL or ones in the configuration file of power management as power knob values for the compute and I/O node. If SYSTEM is set, settings are as shown in Figure 2.12 Reducing Effect on the Node Serving as Both Compute and I/O Node by Making a Different Setting.

Table 3.19 Resources for the Power Knob Operation Function That Can be Set as Custom Resources in the Job ACL (Power Knob Operation Restriction Function)

Custom Resource Name	Definition Contents	Settable Value
freq_min	Lower limit on CPU frequency (MHz)	Value allowed by the hardware
freq_max	Upper limit on CPU frequency (MHz)	Value allowed by the hardware
throttling_state_min	Lower limit on HBM access restriction Specifiable values are the same as for throttling_state.	0, 1, or 2
throttling_state_max	Upper limit on HBM access restriction Specifiable values are the same as for throttling_state.	0, 1, or 2
issue_state_min	Lower limit on instruction issue restriction for compute cores Specifiable values are the same as for issue_state.	0 or 1
issue_state_max	Upper limit on instruction issue restriction for compute cores Specifiable values are the same as for issue_state.	0 or 1

Custom Resource Name	Definition Contents	Settable Value
ex_pipe_state_min	Lower limit on the number of compute core EXes Settable values are the same as for for ex_pipe_state.	0 or 1
ex_pipe_state_max	Upper limit on the number of compute core EXes Settable values are the same as for ex_pipe_state.	0 or 1
eco_state_min	Lower limit on the eco mode state of compute cores Settable values are the same as for eco_state.	Integral from 0 to 2
eco_state_max	Upper limit on the eco mode state of compute cores Settable values are the same as for eco_state.	Integral from 0 to 2
retention_state_min	Lower limit on whether to allow compute cores to transition to Retention state Settable values are the same as for retention_state.	0 or 1
retention_state_max	Upper limit on whether to allow compute cores to transition to Retention state Settable values are the same as for retention_state.	0 or 1

If the setting for a custom resource becomes no longer necessary, delete it from the configuration file pmpjm.conf, and make a setting again.

For details on custom resource settings, see "Custom resource settings" in "Chapter 3 Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management."

For details on the job ACL configuration, see "Job ACL function settings in a cluster" in "Chapter 3 Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Job Management."

3.6.4 Setting Examples for Power Knob Operations

This section provides examples of the settings described in "3.6.1 Rules for setting power knobs."

3.6.4.1 Inheritance of Setting Values to the Compute and I/O Node

This section describes how to omit values when power knob values for the compute and I/O node are the same as for compute nodes.

All the power knob values for compute nodes (ComputeNode subsection) need to be written in the configuration file (papwrm.conf), excluding items that are omissible. If settings for the compute and I/O node (IONode subsection) are omitted, its power knob values are the same as set in the ComputeNode subsection.

Specification Example 1

This specification example specifies respective frequencies for compute nodes and the compute and I/O node when they are idle and executing a job without omission (see b in 3.6.1.1 Setting Rules for the Power Knob Operation Function with the Job Operation Software).

- Compute nodes
  - CPU frequency when the nodes are idle : 1.6 GHz
  - CPU frequency at job execution time : 2.0 GHz
- Compute and I/O node
  - CPU frequency when the node is idle : 2.2 GHz
  - CPU frequency at job execution time : 2.2 GHz

Compute cluster name : cluster

ResourceUnit name : unit1

Example of Code in the Configuration File of the Power Management Function

```
PowerKnob{
    Cluster{
        ClusterName = "cluster"
        ResourceUnit{
            ResourceUnitName = "unit1"
            ComputeNode{
                IdleState{
                    freq = "1600"
                    throttling_state = "9"
                    issue_state = "1"
                    ex_pipe_state = "1"
                    eco_state = "2"
                    retention_state = "1"
                    retention_state_acores = "1"
                }
                RunningState{
                    freq = "2000"
                    throttling_state = "0"
                    issue_state = "0"
                    ex_pipe_state = "0"
                    eco_state = "0"
                    retention_state = "0"
                    retention_state_acores = "0"
                    freq_min = "1600"
                    freq_max = "2200"
                    throttling_state_min = "0"
                    throttling_state_max = "1"
                    issue_state_min = "0"
                    issue_state_max = "1"
                    ex_pipe_state_min = "0"
                    ex_pipe_state_max = "1"
                    eco_state_min = "0"
                    eco_state_max = "2"
                    retention_state_min = "0"
                    retention_state_max = "1"
                }
            }
            IONode{
                IdleState{
                    freq = "2200" # Only specify the CPU frequency
                }
                RunningState {
                    freq = "2200" # Only specify the CPU frequency
                }
            }
        }
    }
}
```

Specification Example 2

The following example specifies the idle frequency for the compute and I/O node, and the same frequency as compute node when the job is run.

- Compute nodes
  - CPU frequency when the nodes are idle : 1.6 GHz
  - CPU frequency at job execution time : 2.0 GHz
- Compute and I/O node
  - CPU frequency when the node is idle : 2.0 GHz
  - CPU frequency at job execution time : 2.0 GHz (Value inherited from compute nodes)

Compute cluster name : cluster

ResourceUnit name : unit1

Example of Code in the Configuration File of the Power Management Function

```
PowerKnob{
    Cluster{
        ClusterName = "cluster"
        ResourceUnit{
            ResourceUnitName = "unit1"
            ComputeNode{
                IdleState{
                    freq = "1600"
                    throttling_state = "9"
                    issue_state = "1"
                    ex_pipe_state = "1"
                    eco_state = "2"
                    retention_state = "1"
                    retention_state_acores = "1"
                }
                RunningState{
                    freq = "2000"
                    throttling_state = "0"
                    issue_state = "0"
                    ex_pipe_state = "0"
                    eco_state = "0"
                    retention_state = "0"
                    retention_state_acores = "0"
                    freq_min = "1600"
                    freq_max = "2000"
                    throttling_state_min = "0"
                    throttling_state_max = "1"
                    issue_state_min = "0"
                    issue_state_max = "1"
                    ex_pipe_state_min = "0"
                    ex_pipe_state_max = "1"
                    eco_state_min = "0"
                    eco_state_max = "2"
                    retention_state_min = "0"
                    retention_state_max = "1"
                }
            }
            IONode{
                IdleState{
                    freq = "2000" # Only specify the CPU frequency
                }
            }
        }
    }
}
```

3.6.4.2 Specifying Default Values Dependent on the Job ACL of the Job or User

The administrator can change settings for job execution time (RunningState) that are written in the configuration file (papwrm.conf) of the power management function, by making definitions in the job ACL. In the configuration file (pmpjm.conf) of the job operation management function on the system management node, the administrator writes default values and settable ranges, as custom resources, that depend on the job ACL of the job or user. They enable them by using the configuration command (pmpjmadm) of the job operation management function. Then, they set the default values and settable ranges in the job ACL by using the job ACL configuration command (pmjacladm).

Items to be Specified

Write power knob values as custom resources in the configuration file (pmpjm.conf) of the job operation management function. Not all items need to be written. Write only power knobs for which settings different from the values written in the configuration file of the power management function need to be made. Power knob items that are not written in the job ACL cannot be used as options to be changed by users at the time of job submission.

Synopsis

- Synopsis for registering values as custom resources in pmpjm.conf

```
ResourceUnit{
    ResourceUnitName=ResourceUnitName
    ResourceGroup{
        ResourceGroupName=ResourceGroupName
        CustomResource {
            Name=PowerKnobName
            ValueType = string
            Value=ListOfSpecifiableValues(comma-separated)
        }
    }
}
```

- Synopsis for temporary file for specifying default values and settable ranges in the job ACL
  - Synopsis for specifying default values and settable ranges for specific users

```
USER: CL, RU={resource unit name}, RG={resource group name} {
    user=<def>{        # General user
        select  custom-PowerKnobName ListOfSpecifiableValues(comma-separated) DefaultValue
    }
    user=UserName {   # Specific user
        select  custom-PowerKnobName ListOfSpecifiableValues(comma-separated) DefaultValue
    }
}
```

- Configuration commands
  - Custom resource settings

    Make settings by using the configuration command of the job management function on the system management node.

```
# pmpjmadm --set -c ComputeClusterName --rscunit ResourceUnitName
```

  - Job ACL settings

    Make settings by using the job ACL management function on the system management node. The following is a command example of specifying settings by writing them in a temporary file.

```
# pmjacladm  -c ComputeClusterName --set -f FileContainingSettings
```

- Confirmation commands
  - Custom resource setting confirmation

    Use the job ACL management command on the system management node. The following is a command example of outputting settings to the standard output.

```
# papjmadm --show -c ComputeClusterName
```

  - Job ACL setting confirmation

    Use the job ACL management command on the system management node. The following is a command example of outputting settings to the standard output.

```
# pmjacladm -c ComputeClusterName --show '*'
```

3.6.4.3 Setting Examples for Power Knob Operations at Job Submission Time

This section shows examples of settings to allow users to perform power knob operations at the time of job submission.

Specification Example 1

This example shows settings to allow users to change, at job submission time, the frequency for compute nodes and the compute and I/O node when they are executing a job. The following settings allow users to submit a job to compute nodes and the compute and I/O node at a frequency of 1.6 GHz. They can also submit a job by specifying a frequency of 2.0 GHz as a custom resource.

- Configuration file
  - Configuration file of the power management function

```
PowerKnob{
    Cluster{
        ClusterName = "cluster"
        ResourceUnit{
            ResourceUnitName = "unit1"
            ComputeNode{
                IdleState{
                    freq = "1600"
                    throttling_state = "9"
                    issue_state = "1"
                    ex_pipe_state = "1"
                    eco_state = "2"
                    retention_state = "1"
                    retention_state_acores = "1"
                }
                RunningState{
                    freq = "2000"
                    throttling_state = "0"
                    issue_state = "0"
                    ex_pipe_state = "0"
                    eco_state = "0"
                    retention_state = "0"
                    retention_state_acores = "0"
                    freq_min = "1600"
                    freq_max = "2200"
                    throttling_state_min = "0"
                    throttling_state_max = "1"
                    issue_state_min = "0"
                    issue_state_max = "1"
                    ex_pipe_state_min = "0"
                    ex_pipe_state_max = "0"
                    eco_state_min = "0"
                    eco_state_max = "2"
                    retention_state_min = "0"
                    retention_state_max = "1"
                }
            }
            IONode{
                IdleState{
                    freq = "2200" # Only specify the CPU frequency
                }
                RunningState {
                    freq = "2200" # Only specify the CPU frequency
                }
            }
        }
    }
}
```

- Custom resource settings

```
ResourceUnit{
    ResourceUnitName=unit1
    ResourceGroup{
        ResourceGroupName=group1
        CustomResource{
            Name=freq
            ValueType=string
            Value=1600,2000,2200  # List of specifiable CPU frequencies
        }
        CustomResource{
            Name=knob_io
            ValueType=string
            Value=SYSTEM,COMPUTE  # List of specifiable power knob options for the compute and
I/O node
        }
    }
}
```

- Job ACL configuration

```
USER: CL, RU=unit1, RG=group1{
    user=<def> {
        select  custom-freq  1600,2000  1600 (*1)
        select  custom-knob_io  COMPUTE  COMPUTE    (*2)
    }
}
```

(*1) Set the default value at job execution time to **1.6 GHz**.
(*2) CPU frequency for the compute and I/O node is the same as for **compute nodes**.

- Submission example

In the case of default (1.6 GHz)

```
$ pjsub run.sh
```

When 2.0 GHz is specified in the job ACL

```
$ pjsub -L freq=2000 run.sh
```

In the case of this configuration, a frequency in the range from 1600 MHz to 2200 MHz can be set by using the Power API within a job.

throttling_state is not set in the job ACL. Therefore, users cannot specify throttling_state at the time of job submission. Since throttling_state_min and throttling_state_max are set to 0 and 1, respectively, throttling_state can be set to 0 or 1 by using the Power API within a job.

ex_pipe_state is not set in the job ACL. Therefore, users cannot specify ex_pipe_state at the time of job submission. Since both ex_pipe_state_min and ex_pipe_state_max are set to 0, ex_pipe_state cannot be changed to a value other than 0 by using the Power API within a job.

When the job ends, power knob values for compute nodes are changed to ones written in the IdleState subsection in the ComputeNode subsection. Power knob values for the compute and I/O node are changed to ones written in the IdleState subsection in the IONode subsection. Since items other than frequency are not written in the IdleState subsection in the IONode subsection in this configuration, power knob values other than frequency are changed to the values written in the IdleState subsection in the ComputeNode subsection.

Specification Example 2

This specification example is different from specification example 1 only in the setting for knob_io. In this configuration, users can manipulate power knob values for compute nodes similarly to specification example 1, but they cannot manipulate ones for the compute and I/O node at the time of job submission.

- Configuration file
  - Configuration file of the power management function (omitted as it is the same as in specification example 1)

- Custom resource settings (omitted as they are the same as in specification example 1)
- Job ACL configuration

```
USER: CL, RU=unit1, RG=group1{
    user=<def> {
        select  custom-freq  1600,2000  1600    (*1)
        select  custom-knob_io  SYSTEM  SYSTEM  (*2)
    }
}
```

(*1) Set the default value at job execution time to **1.6 GHz**.
(*2) CPU frequency for the compute and I/O node is a system setting.

- Submission example

In the case of default (1.6 GHz)

```
$ pjsub run.sh
```

When 2.0 GHz is specified in the job ACL

```
$ pjsub -L freq=2000 run.sh
```

With either specification, the compute and I/O node operates at 2.2 GHz as written in the configuration file of the power management function. Operations using the Power API within a job can be performed because range settings for compute nodes are inherited. If you do not want to allow even operations using the Power API, set both the minimum and maximum values of the relevant item to the same value.

3.6.5 Disabling for the Power Knob Operation Function

This section describes disabling for the Power Knob Operation Function.

The Power Knob Operation Function disables the following two functions for each resource unit:

1. Stop the Job Operation Software service

   Isolate compute nodes with the operation described in "Job Operation Software Administrator's Guide for Maintenance."

2. Delete Job ACL definitions

   Delete Job ACL definitions if Job ACL custom resources were set (See "3.6.4.2 Specifying Default Values Dependent on the Job ACL of the Job or User.") Detailed instruction to delete Job ACL custom resources is described at "Job Operation Management Function Settings" in "Job Operation Software Administrator's Guide for Maintenance." The following is an example to delete custom resource "freq."

   ```
   [Execution from system management node]
   # pmjacladm -c ClusterName --del 'USER: CL, RU=ResourceUnitName, RG=ResourceGroupName{user=<def>
   {select custom-freq}}'
   ```

3. Delete custom resource

   Delete custom resource in the configuration file pmpjm.conf of the job operation management function in a resource unit of the job operation management function if custom resource was set in "3.6.4.2 Specifying Default Values Dependent on the Job ACL of the Job or User."

4. Unset exit functions provided by the job scheduler function

   Delete exit functions provided by the job scheduler function set in "Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function."

   After deleting, you need to execute the pmpjmadm command on system management node so that the system reflects the setting contents.

   ```
    [Execution from system management node]
   # pmpjmadm --set -c ClusterName --rscunit ResourceUnitName
   ```

For details on the unsetting method, see "Job manager exit function and Job scheduler exit function " in "Chapter 2 Creating and Incorporating Hooks" in " Job Operation Software Administrator's Guide for Job Operation Manager Hook."

5. Delete PowerKnob section

   Delete PowerKnob section in the configuration file papwrm.conf of the power management function.

6. Apply the setting for deleting PowerKnob section

   Apply the deleted PowerKnob section configuration file. See "3.7.1 Applying Settings."

7. Restart compute nodes

   Restart compute nodes for initialize Power Knob.

8. Start the Job Operation Software service

   Add isolated compute nodes again with the operation described in "Job Operation Software Administrator's Guide for Maintenance".

3.7 Applying and Viewing the papwrm.conf File

You can apply settings or view applied settings by using the papwrmgradm command.

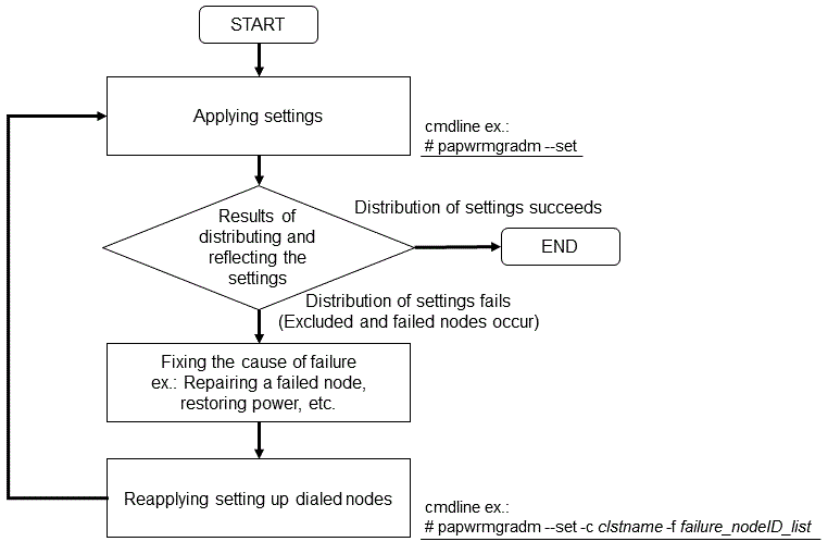
See

For details on the papwrmgradm command options, see the man page of the papwrmgradm command.

3.7.1 Applying Settings

Distribute and reflect the papwrm.conf file and the contents of the files located in the/etc/opt/FJSVtcs/pwrm directory as shown in "Figure 3.1 Flow of Power Management Function Settings"

Figure 3.1 Flow of Power Management Function Settings

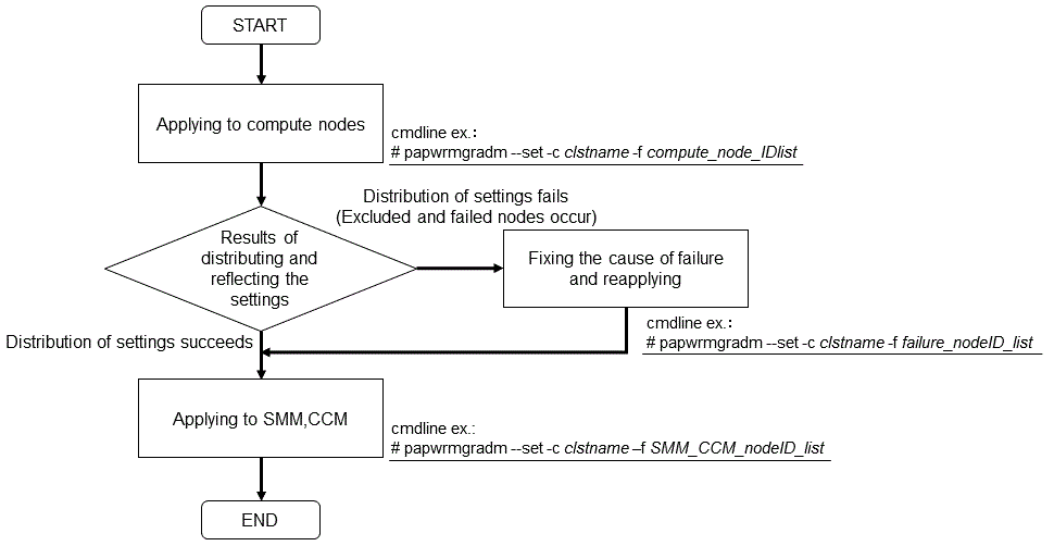


Specify the --set option with the papwrmgradm command to distribute and apply the contents of papwrm.conf file and the files placed in and under the /etc/opt/FJSVtcs/pwrm directory.

```
[System management node]
# papwrmgradm --set
[WARNING]
Do you really want to continue (y/n)y     <- Confirmation is required before registration.
pmscatter -c clst1
pmscatter command was completed.
[INFO] PWRM 0110 papwrmgradm The processing of the configuration file was completed
```

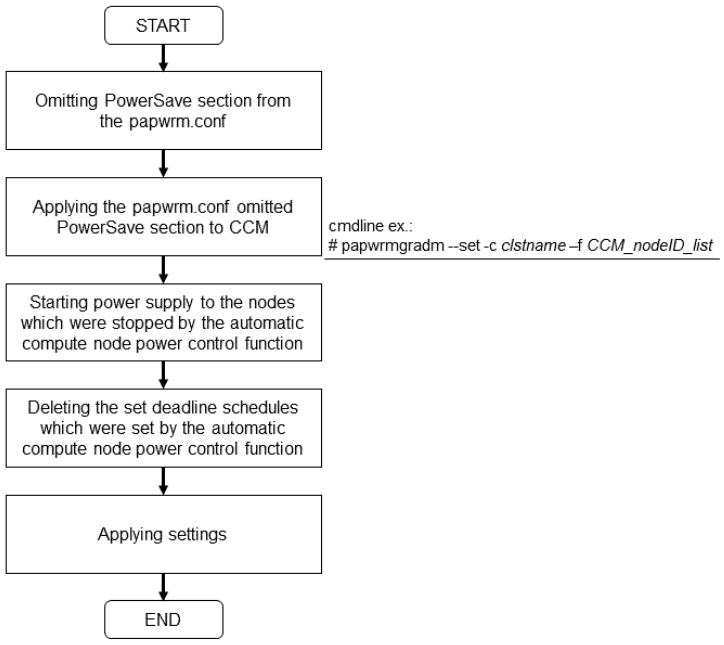
The information above shows that there is a compute cluster clst1 in the system, and distributes settings. It was successful because "pmscatter command was completed" is printed.

Figure 3.2 Flow of the Automatic Compute Node Power Control Function Settings



If the automatic compute node power control function becomes available, apply the settings to the compute nodes and then to the system management node and the compute cluster management node as shown in "Figure 3.2 Flow of the Automatic Compute Node Power Control Function Settings."

Figure 3.3 Flow of Any Power Management Function Settings When the Automatic Compute Node Power Control Function has enabled



"Figure 3.3 Flow of Any Power Management Function Settings When the Automatic Compute Node Power Control Function has enabled" shows the flow of applying the new power management settings when the automatic compute node power control function has been available. First, omit PowerSave section from the papwrm.conf and apply it to the compute cluster management node because of stopping the automatic compute node power control function temporarily. Next, restart the compute nodes which was stopped by the automatic compute node power control function and delete the deadline schedules. Finally, apply the settings.

To check and delete the deadline schedule which was set by automatic compute node power control function, use the following command.

```
[Compute cluster management node]
# padeadline --show
```

```
NO     TYPE START               END                 TARGET
231    f    2019-10-29 17:12:07 2019-10-29 18:12:59 AUTOPWRCTL
# padeadline -c clst1 --cancel 231
[WARNING]
Do you really want to continue (y/n)?
y
[INFO] PJM 6200 padeadline Deadline-schedule 231 canceled.
```

The deadline schedule number with AUTOPWRCTL as TARGET is the one set by the automatic compute node power control function. In the above example, the deadline schedule number 231 set by the automatic compute node power control function is deleted.

Information

It is recommended to apply the settings as part of software maintenance. For detail of the software maintenance, see "Job Operation Software Administrator's Guide for Maintenance."
Each type of settings is applied at the following timing:

- New settings for the system power collecting/visualization function (definition in the SystemPower section (see "3.3 Settings for the System Power Collecting/Visualization Support Function")) are applied when power collecting at one-minute intervals is performed.
- New settings for the automatic compute node power control function (definition in the PowerSave section (see "3.4 Settings for the Automatic Compute Node Power Control Function")) are applied at every 30 seconds.
- New settings for the job power estimate function (definition in the JobPowerEstimation section (see "3.5 Settings for the Job Power Estimate Function")) are applied when a new job is submitted.
- New settings for the power knob operation function (definition in the PowerKnob section (see "3.6 Settings for the Power Knob Operation Function [FX]")) are applied when a new job starts running on compute nodes.
  If the settings are applied at the start and end of a job that runs on multiple compute nodes, the compute nodes within the same job may run with different settings.

If distribution of settings fails on any nodes, the failed nodes and their detailed information are output to the /var/log/FJSVtcs/pwrm/papwrmgradm directory.

```
[System management node]
# papwrmgradm --set
[WARNING]
Do you really want to continue (y/n)y     <- Confirmation is required before registration.
pmscatter -c clst1
pmscatter command was completed.
pmscatter -c clst2
The execution failed file was output.(/var/log/FJSVtcs/pwrm/papwrmgradm/papwrmgradm_clst2_failed)
The execution excluded file was output.(/var/log/FJSVtcs/pwrm/papwrmgradm/papwrmgradm_clst2_excluded)
pmscatter -c clst3
The execution failed file was output,(/var/log/FJSVtcs/pwrm/papwrmgradm/papwrmgradm_clst3_failed)
The execution excluded file was output.(/var/log/FJSVtcs/pwrm/papwrmgradm/papwrmgradm_clst3_excluded)
[ERR.] PWRM 0144 papwrmgradm pmscatter command failed: clst2, clst3
```

The information above shows that there are three compute clusters clst1, clst2, and clst3 in the system, and distributes settings to each cluster. clst1 was successful because "pmscatter command was completed" is printed. clst2 and clst3 were failed.

The following two types of files may be output, depending on the reason of failure:

- papwrmgradm_<*compute cluster name*>_failed

  This file is output if distribution failed on any nodes when the files placed in and under the /etc/opt/FJSVtcs/pwrm directory are distributed to individual nodes by the pmscatter command. The node IDs of the nodes to which distribution failed and their detailed information are output to this file.

- papwrmgradm_<*compute cluster name*>_excluded

  This file is output if there are any nodes that are excluded from the destination of distribution when the files placed in and under the /etc/opt/FJSVtcs/pwrm directory are distributed to individual nodes by the pmscatter command. The node IDs of the node excluded from distribution and their detailed information are output to this file.

<*compute cluster name*> specifies the name of the compute cluster to which the node belongs.

After removing the cause of the distribution failure, resetting can be done for the node to which settings failed to apply, by specifying the failed compute cluster or node with the -c or -f option.

Use the -c option to specify the compute cluster name for which resetting is to be done. In the file to be given by the -f option, write one node ID per line in hexadecimal. If the --force option is specified, settings are forcibly applied without a prompt for execution confirmation (y/n).

- If power supply is turned off by the automatic compute node power control function, apply settings after turning it on again.
- If you change the cluster name or resource unit name, modify the configuration and redistribute.

3.7.2 Viewing Settings

Settings that have been made can be viewed by using the --show option of the papwrmgradm command.

```
[System management node]
# papwrmgradm --show
SystemPower {
    StartTime = "50"
    LogLevel = "1" # default
    CommandLine = "/usr/sbin/cmd arg"
    AcceptableRange = "600"
    PowerGroup {
        PowerGroupName = "pwrgrp1"
        ClusterName = "cluster"
        NodeList = "sysnodelist1.txt"
        ExternalDeviceList = "extlist1.txt"
    }
}
PowerSave {
...
Omitted
```

(Remarks) For items that are set to default values when settings are omitted, # default is appended at the end of the line.

Chapter 4 Operation with the Power Management Function

This section describes specific operation methods for using the power management function for operation.

Information

[Specifying a compute cluster]
Some commands described below allow you to specify the compute cluster for which the command is executed.
Specify the compute cluster name by using the -c option of the command or the environment variable PXMYCLST. Only one compute cluster name can be specified.
If both the -c option and the environment variable PXMYCLST are specified, the specification by the -c option is valid.

4.1 Checking the Power Consumption Information of the System

The power consumption information of the system that was collected and stored by the system power collecting/visualization support function can be output. As output functions, the pasyspwr command and the system power visualization support API are available.

4.1.1 pasyspwr Command

Use the pasyspwr command to output the power consumption information of the system.
The administrator can execute the pasyspwr command from the active system management node.

See

For details on the pasyspwr command option, see the man page of the pasyspwr command.

The following describes how to output the power consumption information of the system and provides output examples.

a. Outputting a sum of the power consumption of the entire system
Execute the pasyspwr command to output a sum of the power consumption of all the compute nodes and all the external equipment in the compute cluster specified by the environment variable PXMYCLST.

The method to calculate a sum depends on the acceptable range of variation in measurement time (definition item AcceptableRange in the configuration file papwrm.conf) and the specifications by the -momentary/--average option of the pasyspwr command and by the --last option.

- When the power consumption of all the devices is successfully measured within the acceptable range, a sum is output. If measurement fails with any one of them, "0" is output as the sum (PWR_TYPE is set to ERR). If the --last option is specified, power consumption outside the acceptable range is also used for calculating a sum.
- If the --momentary option is specified, momentary power consumption is preferentially used. When momentary power consumption is successfully used for all the devices, PWR_TYPE is set to MOM.
- If the --average option is specified, average power consumption is preferentially used. When average power consumption is successfully used for all the devices, PWR_TYPE is set to AVE.
- If neither the --momentary nor --average option is specified, average power consumption is preferentially used. (This is the same as when the --average option is specified.)
- If average power consumption and momentary power consumption are mixed as types of power consumption for a sum calculation, PWR_TYPE is set to MIX.

For details on calculation methods for sums, see "2.1.2 Power Calculation Function."

The following provides an execution example and describes displayed items and their meanings. A sum of the power consumption of all the compute nodes and all the external equipment in the compute cluster *clstname* is output to TOTAL_PWR. Since neither the --momentary nor --average option is specified, output is the same as when the --average option is specified. However, this example shows that average power consumption and momentary power consumption are mixed as types of power consumption for a sum calculation.

```
[System management node]
# PXMYCLST=clstname pasyspwr
TOTAL_PWR : 12345000
PWR_TYPE  : MIX
DATE      : 2016/02/01 09:55:00
```

clstname: Compute cluster name. This meaning is the same in the subsequent description.

TOTAL_PWR

Sum of power consumption of the specified compute nodes and external equipment (unit: W)

PWR_TYPE

Type of power consumption used for calculating the sum

MOM: All are momentary power consumption.
AVE: All are average power consumption.
MIX: Average power consumption and momentary power consumption are mixed.
ERR: Sum acquisition error

DATE

Output date and time

[Remarks]
Description of the same output items is omitted from the subsequent description.

b. Outputting a sum of momentary power consumption

When the --momentary option is specified, momentary power consumption is used to calculate a sum of power consumption. For a device without momentary power consumption, average power consumption is handled as the power consumption of the device and used for calculating a sum. Specify the -v option to output the power consumption of respective compute nodes and external equipment that are used for calculating the sum, along with the sum of power consumption.

- When all the compute nodes in the specified compute cluster should be targets of output (Specifying the -c option)

```
[System management node]
# pasyspwr -c clstname --momentary -v
TOTAL_PWR : 12300000
PWR_TYPE : MOM
DATE : 2016/02/01 09:55:00
NODE/DEV    TYPE  PWR        M_DATE                B_DATE              SER_NO
0x01010001  MOM   320        2016/02/01 09:55:00   -                   10555
0x01010002  MOM   400        2016/02/01 09:55:00   -                   10555
0x01010003  MOM   400        2016/02/01 09:53:00   -                   10553
...
```

- When all the external equipment should be a target of output (Specifying the --extdev option)

```
[System management node]
# pasyspwr --extdev all --momentary -v
TOTAL_PWR : 45000
PWR_TYPE : MOM
DATE : 2016/02/01 09:55:00
NODE/DEV    TYPE  PWR        M_DATE                B_DATE              SER_NO
Facility1   MOM   1200       2016/02/01 09:53:00   -                   10005
Facility2   MOM   300        2016/02/01 09:52:00   -                   10004
...
```

To target a specific compute node, specify the node ID with the -n option. To target multiple compute nodes, specify their node IDs with the -n option by separating them with commas (,), a range of node IDs using a hyphen (-), or the name of the file containing node IDs of the targets node with the --nodelist option.
To target a specific external equipment piece, specify the external equipment name with the --extdev option. To target multiple external equipment pieces, specify them with --extdev option by separating them with commas (,) or the file containing the external equipment names of the target external equipment with the --extdevlist option.

The following describes the displayed items and their meaning:

NODE/DEV

Node ID of compute node or external equipment name

TYPE

Type of power consumption

MOM: Momentary power consumption
AVE: Average power consumption

PWR

Power consumption (W)

M_DATE

Measurement time

B_DATE

Reference time

SER_NO

Serial number

For compute nodes, a value that increments by 1 every minute is output.

For external equipment, the value of *ser_no* is output. This is a value output by the command for collecting external equipment power consumption that is created in "3.3.3.2 How to Create a Command for Collecting External Equipment Power Consumption."

c. Outputting a sum of average power consumption

When the --average option is specified, average power consumption is used to calculate a sum of power consumption. For a device without average power consumption, momentary power consumption is handled as the power consumption of the device and used for calculating a sum. The following targets all the external equipment.

```
[System management node]
# pasyspwr --extdev all --average -v
TOTAL_PWR : 45000
PWR_TYPE : AVE
DATE : 2016/02/01 09:55:00
NODE/DEV    TYPE  PWR        M_DATE                B_DATE                SER_NO
Facility1   AVE   1200       2016/02/01 09:53:00   2016/02/01 09:52:00   10005
Facility2   AVE   300        2016/02/01 09:52:00   2016/02/01 09:51:00   10004
...
```

d. Outputting a sum of the power consumption of a power group

If all is specified with the --pwrgrp option, a sum of power consumption is output for each and every power group. The method to calculate this sum is the same as described in "a. Outputting a sum of the power consumption of the entire system."
The following is an execution example.

```
[System management node]
# pasyspwr --pwrgrp all --momentary -v
PWR_GRP : grp1
TOTAL_PWR : 12000
PWR_TYPE : MOM
DATE : 2016/02/01 09:55:00
NODE/DEV    TYPE  PWR        M_DATE                B_DATE                SER_NO
0x01010001  MOM   320        2016/02/01 09:55:00   -                     10555
0x01010002  MOM   400        2016/02/01 09:55:00   -                     10555
0x01010003  MOM   400        2016/02/01 09:53:00   -                     10553
...
PWR_GRP : grp2
TOTAL_PWR : 10000
PWR_TYPE : MIX
```

```
DATE : 2016/02/01 09:55:00
NODE/DEV    TYPE  PWR        M_DATE               B_DATE               SER_NO
0x01010011  MOM   320        2016/02/01 09:55:00  -                    10555
0x01010012  MOM   400        2016/02/01 09:55:00  -                    10555
Facility1   AVE   1200       2016/02/01 09:53:00  2016/02/01 09:52:00  10005
Facility2   AVE   300        2016/02/01 09:52:00  2016/02/01 09:51:00  10004
...
```

To target a specific power group, specify the power group name with the --pwrgrp option. To target multiple power groups, specify the power group names with the --pwrgrp option by separating them with commas (,) or the name of the file containing the target power group name with the --pwrgrplist option.

The following describes the displayed items and their meaning:

PWR_GRP

Power group name

e. Outputting data in a delimiter-separated format (CSV format)
By specifying the --data option, data can be output in a delimiter (comma ",") separated format (CSV format). The following is an example of outputting data shown in "d. Outputting a sum of the power consumption of a power group" in a comma (,) separated format.

```
[System management node]
# pasyspwr --pwrgrp all --momentary -v --data
H,PWR_GRP,TOTAL_PWR,PWR_TYPE,DATE,NODE/DEV,TYPE,PWR,M_DATE,B_DATE,SER_NO
,grp1,12000,MOM,2016/02/01 09:55:00,0x01010001,MOM,320,2016/02/01 09:55:00,-,10555
,grp1,12000,MOM,2016/02/01 09:55:00,0x01010002,MOM,400,2016/02/01 09:55:00,-,10555
,grp1,12000,MOM,2016/02/01 09:55:00,0x01010003,MOM,400,2016/02/01 09:53:00,-,10553
...
,grp2,10000,MIX,2016/02/01 09:55:00,0x01010011,MOM,320,2016/02/01 09:55:00,-,10555
,grp2,10000,MIX,2016/02/01 09:55:00,0x01010012,MOM,400,2016/02/01 09:55:00,-,10555
,grp2,10000,MIX,2016/02/01 09:55:00,Facility1,AVE,1200,2016/02/01 09:53:00,2016/02/01
09:52:00,10005
,grp2,10000,MIX,2016/02/01 09:55:00,Facility2,AVE,300,2016/02/01 09:52:00,2016/02/01
09:51:00,10004
...
```

f. Outputting past power information
By specifying the --trace option, past power information (the latest 10 minutes, without the --time option) can be output. By specifying the --time option at the same time, power information at any time or for any period of time with in the last 10 days can be also output. If the --trace option is specified, no sums are output.

- When outputting the power information of a compute node (0x01010001) for the latest 10 minutes

```
[System management node]
# pasyspwr -c clstname -n 0x01010001 --trace
NODE/DEV    TYPE  PWR        M_DATE               B_DATE               SER_NO
0x01010001  MOM   320        2017/12/22 09:55:00  -                    10535
0x01010001  MOM   320        2017/12/22 09:54:00  -                    10534
0x01010001  MOM   320        2017/12/22 09:53:00  -                    10533
...
```

- When outputting the power information of a compute node (0x01010001) at a certain time (closest to 12:00:00 on December 21, 2017)

```
[System management node]
# pasyspwr -c clstname -n 0x01010001 --trace --time 20171221120000
NODE/DEV    TYPE  PWR        M_DATE               B_DATE               SER_NO
0x01010001  MOM   320        2017/12/21 12:00:00  -                    10450
```

- When outputting the power information of a compute node (0x01010001) for a certain period (from 23:00:00 on December 21, 2017 to 00:00:00 on December 22, 2017)

```
[System management node]
# pasyspwr -c clstname -n 0x01010001 --trace --time 20171221230000-20171222000000
NODE/DEV    TYPE  PWR        M_DATE               B_DATE               SER_NO
0x01010001  MOM   320        2017/12/22 00:00:00  -                    11840
0x01010001  MOM   320        2017/12/21 23:59:00  -                    11839
...
0x01010001  MOM   320        2017/12/21 23:01:00  -                    11781
0x01010001  MOM   320        2017/12/21 23:00:00  -                    11780
```

4.1.2 System Power Visualization Support API

An API library for viewing the power consumption information of the system from C/C++ language is provided.

The system power visualization support API has the following two structures and seven functions.

Table 4.1 Structures for the System Power Visualization Support API

Structure Name	Description
PwrmPwrInfo_t	Stores the power consumption information of compute nodes and external equipment.
PwrmPwrGrp_t	Stores the power consumption information of each power group.

Table 4.2 Functions of the System Power Visualization Support API

Function Name	Description
pwrm_init	Library initialization
pwrm_fini	Library termination
pwrm_free_PwrInfo	Releases the area where PwrmPwrInfo_t structures are stored.
pwrm_free_PwrGrp	Releases the area where PwrmPwrGrp_t structures are stored.
pwrm_get_pwrinfo_by_node	Returns the power consumption information of the specified compute node whose measurement time and reference time are the most recent.
pwrm_get_pwrinfo_by_extdev	Returns the power consumption information of the specified external equipment whose measurement time and reference time are the most recent.
pwrm_get_pwrinfo_by_pwrgrp	Returns the power consumption information of the specified power group whose measurement time is the most recent.

When using this API, use the header file /usr/include/FJSVtcs/pwrm.h and the library /usr/lib64/libpwrm.so. This API is not thread safe.

This API can be used by the administrator on the active system management node. Also perform compiling on the system management node.

4.1.2.1 Power Information Structure PwrmPwrInfo_t

The power information structure PwrmPwrInfo_t stores power consumption information acquired by the compute node power consumption acquisition function pwrm_get_pwrinfo_by_node() and the external equipment power consumption acquisition function pwrm_get_pwrinfo_by_extdev().

```
typedef struct PwrmPwrInfo_t {
    int dev_type;
    nid_t node_id;
    char *extdev_name;
    int pwr_type;
    uint64_t pwr;
    time_t m_time;
    time_t b_time;
    uint64_t ser_no;
} PwrmPwrInfo_t;
```

Table 4.3 Members of the Power Information Structure PwrmPwrInfo_t

Member	Type	Meaning
dev_type	int	Device type PWRM_NODE: Compute node PWRM_EXTDEV: External equipment
node_id	nid_t	Node ID When dev_type is PWRM_EXTDEV, 0 is stored.
extdev_name	char *	External equipment name When dev_type is PWRM_NODE, NULL is stored.
pwr_type	int	Power consumption type PWRM_AVERAGE: Average power consumption PWRM_MOMENT: Momentary power consumption PWRM_ERROR: Cannot acquire power consumption (Only when it is stored in PwrmPwrGrp_t)
pwr	uint64_t	Power consumption (W) When pwr_type is PWRM_AVERAGE, average power consumption is stored. When pwr_type is PWRM_MOMENT, momentary power consumption is stored.
m_time	time_t	Measurement time Time when power consumption was measured
b_time	time_t	Reference time When pwr_type is PWRM_AVERAGE, the past measurement time used for calculating average power consumption is stored. When pwr_type is PWRM_MOMENT, 0 is stored.
ser_no	uint64_t	Serial number For compute nodes, a value that increments by 1 every minute is stored. For external equipment, the value of *ser_no* is stored. This is a value output by the command for collecting external equipment power consumption that is created in "3.3.3.2 How to Create a Command for Collecting External Equipment Power Consumption."

4.1.2.2 Power Group Structure PwrmPwrGrp_t

The power group structure PwrmPwrGrp_t stores power consumption information acquired by the power group power consumption acquisition function pwrm_get_pwrinfo_by_pwrgrp().

```
typedef struct PwrmPwrGrp_t {
    char *pwrgrp_name;
    int pwr_type;
    uint64_t pwr;
    time_t c_time;
    int32_t num;
    PwrmPwrInfo_t *list_pwr_info;
} PwrmPwrGrp_t;
```

Table 4.4 Members of the Power Group Structure PwrmPwrGrp_t

Member	Type	Meaning
pwrgrp_name	char *	Power group name
pwr_type	int	Power consumption type PWRM_AVERAGE: Only average power consumption is used to calculate a sum. PWRM_MOMENT: Only momentary power consumption is used to calculate a sum. PWRM_MIX: Average power consumption and momentary power consumption are mixed to calculate a sum PWRM_ERROR: Cannot calculate a sum.
pwr	uint64_t	Total power consumption (W)

Member	Type	Meaning
c_time	time_t	Calculation time Timing at which the sum is calculated
num	int32_t	Number of compute nodes/external equipment pieces Total number of compute nodes and external equipment pieces included in the power group
list_pwr_info	PwrmPwrInfo_t *	Power consumption information num-piece array of the power information structure PwrmPwrInfo_t

4.1.2.3 Library Initialization Function pwrm_init()

The library initialization function pwrm_init() makes initial settings for this API. This function must be executed without fail before using this API. Do not execute it dually. Otherwise, the pwrm_init() function ends abnormally.

```
#include <FJSVtcs/pwrm.h>

int pwrm_init(void);
```

Table 4.5 End Status of pwrm_init()

End Status	Meaning
0	Normal end
1 or greater	Abnormal end. If the function ends abnormally, pwrm_fini() does not need to be executed.

4.1.2.4 Library Termination Function pwrm_fini()

The library termination function pwrm_fini() terminates this API. This function needs to be executed after using this API. Before executing this function, release the area secured by pwrm_get_pwrinfo_by_node()/pwrm_get_pwrinfo_by_extdev(), by using pwrm_free_PwrInfo(). Release the area secured by pwrm_get_pwrinfo_by_pwrgrp(), by using pwrm_free_PwrGrp().

```
#include <FJSVtcs/pwrm.h>

int pwrm_fini(void);
```

Table 4.6 End Status of pwrm_fini()

End Status	Meaning
0	Normal end
1 or greater	Abnormal end

4.1.2.5 PwrmPwrInfo_t Release Function pwrm_free_PwrInfo()

The PwrmPwrInfo_t release function pwrm_free_PwrInfo() releases the area secured by pwrm_get_pwrinfo_by_node() or pwrm_get_pwrinfo_by_extdev(). This function needs to be executed after using PwrmPwrInfo_t without fail. In addition, this function needs to be executed before executing pwrm_fini(). However, if pwrm_get_pwrinfo_by_node() or pwrm_get_pwrinfo_by_extdev() ends abnormally, do not execute this function.

```
#include <FJSVtcs/pwrm.h>

void pwrm_free_PwrInfo(PwrmPwrInfo_t *p);
```

Table 4.7 Argument of pwrm_free_PwrInfo()

Argument	Input/Output	Description
p	In	Area secured by pwrm_get_pwrinfo_by_node() or pwrm_get_pwrinfo_by_extdev()

4.1.2.6 PwrmPwrGrp_t Release Function pwrm_free_PwrGrp()

The PwrmPwrGrp_t release function pwrm_free_PwrGrp() releases the area secured by pwrm_get_pwrinfo_by_pwrgrp(). This function needs to be executed without fail after using PwrmPwrGrp_t. In addition, this function needs to be executed before executing pwrm_fini(). However, do not execute this function if pwrm_get_pwrinfo_by_pwrgrp() ends abnormally.

```
#include <FJSVtcs/pwrm.h>

void pwrm_free_PwrGrp(PwrmPwrGrp_t *p);
```

Table 4.8 Argument of pwrm_free_PwrGrp()

Argument	Input/Output	Description
p	In	Area secured by pwrm_get_pwrinfo_by_pwrgrp()

4.1.2.7 Compute Node Power Consumption Acquisition Function pwrm_get_pwrinfo_by_node()

The compute node power consumption acquisition function pwrm_get_pwrinfo_by_node() returns the most recent power consumption information of the specified compute node to the area secured internally in the function. The secured area needs to be released by pwrm_free_PwrInfo().

```
#include <FJSVtcs/pwrm.h>

int pwrm_get_pwrinfo_by_node(PwrmPwrInfo_t **pwrinfo, const char *clst_name, nid_t node_id, int
pwr_type);
```

Table 4.9 Arguments of pwrm_get_pwrinfo_by_node()

Argument	Input/Output	Description
pwrinfo	Out	Pointer to the area where power consumption information is stored
clst_name	In	Compute cluster name to be specified
node_id	In	Node ID to be specified
pwr_type	In	Power consumption type (PWRM_AVERAGE or PWRM_MOMENT) Which to specify depends on the device. Execute the pasyspwr command with the -v --last option, and choose a type according to the output TYPE.

Table 4.10 End Status of pwrm_get_pwrinfo_by_node()

End Status	Meaning
0	Normal end
1 or greater	Abnormal end. If the function ends abnormally, do not execute pwrm_free_PwrInfo().

4.1.2.8 External Equipment Power Consumption Acquisition Function pwrm_get_pwrinfo_by_extdev()

The external equipment power consumption acquisition function pwrm_get_pwrinfo_by_extdev() returns the most recent power consumption information of the specified external equipment to the area secured internally in the function. The secured area needs to be released by pwrm_free_PwrInfo().

```
#include <FJSVtcs/pwrm.h>

int pwrm_get_pwrinfo_by_extdev(PwrmPwrInfo_t **pwrinfo, const char *extdev_name, int pwr_type);
```

Table 4.11 Arguments of pwrm_get_pwrinfo_by_extdev()

Argument	Input/Output	Description
pwrinfo	Out	Pointer to the area where power consumption information is stored

Argument	Input/Output	Description
extdev_name	In	External equipment name to be specified
pwr_type	In	Power consumption type (PWRM_AVERAGE or PWRM_MOMENT) Which to specify depends on the device. Execute the pasyspwr command with the -v --last option, and choose a type according to the output TYPE.

Table 4.12 End Status of pwrm_get_pwrinfo_by_extdev()

End Status	Meaning
0	Normal end
1 or greater	Abnormal end. If the function ends abnormally, do not execute pwrm_free_PwrInfo().

4.1.2.9 Power Group Power Consumption Acquisition Function pwrm_get_pwrinfo_by_pwrgrp()

The power group power consumption acquisition function pwrm_get_pwrinfo_by_pwrgrp() returns the most recent power consumption information of a specified power group to the area secured internally in the function. The secured area needs to be released by pwrm_free_PwrGrp().

This function can calculate a sum of the power consumption of a power group. The method to calculate the sum is equivalent to the pasyspwr command. For details, see "4.1.1 pasyspwr Command."

- The --momentary option is equivalent to the pwr_type argument PWRM_MOMENT.
- The --average option is equivalent to the pwr_type argument PWRM_AVERAGE.
- There are no arguments equivalent to the --last option. If the measurement time of any device exceeds the acceptable range (definition item AcceptableRange in the configuration file papwrm.conf), the pwrgrp argument pwr_type is set to PWRM_ERROR and pwr_type of the device is set to PWRM_ERROR in list_pwr_info.

```
#include <FJSVtcs/pwrm.h>

int pwrm_get_pwrinfo_by_pwrgrp(PwrmPwrGrp_t **pwrgrp, const char *pwrgrp_name, int pwr_type);
```

Table 4.13 Arguments of pwrm_get_pwrinfo_by_pwrgrp()

Argument	Input/Output	Description
pwrgrp	Out	Pointer to the area where power consumption information is stored
pwrgrp_name	In	Power group name to be specified
pwr_type	In	Power consumption type (PWRM_AVERAGE or PWRM_MOMENT) Which to specify depends on the device. Execute the pasyspwr command with the -v --last option, and choose a type according to the output TYPE.

Table 4.14 End Status of pwrm_get_pwrinfo_by_pwrgrp()

End Status	Meaning
0	Normal end. Even when part or all of the power consumption of the power group cannot be acquired, the function ends normally as long as the power group exists. If acquisition failed with any device, the pwrgrp argument pwr_type is set to PWRM_ERROR and the pwr_type of the device is set to PWRM_ERROR in list_pwr_info.
1 or greater	Abnormal end. If the function ends abnormally, do not execute pwrm_free_PwrGrp().

4.1.2.10 Sample Code

The following is a sample code sample.c using this API to acquire the average node power of FX server (node ID 0x01010001) in the computer cluster cluster1 and display data on the standard output.

```
#include <stdio.h>
#include <inttypes.h>
#include <FJSVtcs/pwrm.h>

int main()
{
    PwrmPwrInfo_t* pwrinfo;
    nid_t nid;
    int r;

    r = pwrm_init();
    if (r != 0) {
        printf("pwrm_init error(%d)\n", r);
        return 1;
    }

    nid = 0x01010001;

    r = pwrm_get_pwrinfo_by_node(&pwrinfo, "cluster1", nid, PWRM_AVERAGE);
    if (r != 0) {
        printf("pwrm_get_pwrinfo_by_node error(%d)\n", r);
        return 1;
    }

    printf("%" PRIu64 "(W)\n", pwrinfo->pwr);

    pwrm_free_PwrInfo(pwrinfo);
    pwrm_fini();

    return 0;
}
```

If you compile sample.c, you must specify the library libpwrm.so.

```
[System management node]
# gcc sample.c -lpwrm
```

4.2 Backing Up the System Power Database

The system power database stores power consumption information for up to 10 days. Before viewing power consumption information older than 10 days or in case of database damage, back up the database as necessary. There are two methods.

Saving Power Consumption Information in Text

Regularly (for example, once a day) save power consumption information in text by using the pasyspwr command. For details, see "4.1.1 pasyspwr Command."

Saving Power Consumption Information in Database Format

Regularly (for example, once a day) save power consumption information by using the mysqldump command. The following is an execution example.

```
[Compute cluster management node]
# mysqldump --single-transaction -u syspwr -p syspwr > syspwr.backup
Enter password: password
```

For details on the mysqldump command, restore method, and repair method for a damaged database, see the MariaDB manual (https://mariadb.com).

4.3 Checking the Operation Status of the Compute Node Automatic Power Control Function

You can check the compute nodes stopped by the compute node automatic power control function by specifying the -v and --detail options with the pashowclst command.

The following example shows that the compute node automatic power control function has stopped the compute node with node ID 0xFFFF0004. For the compute node stopped by the compute node automatic power control function, the REASON field displays "DeadlineSchedule" and the DETAIL field displays "Power saving." When a job is allocated to the compute node, it automatically starts in line with the scheduled execution start time of the job. The administrator does not need to take action.

```
# pashowclst -c clstname --nodetype CN -v --detail
[ CLST: clstname ]
[ NODETYPE: CN ]
NODE       NODETYPE STATUS   REASON            PWR_STATUS ... SRV_STATUS             DETAIL
0xFFFF0004 CN       Disable  DeadlineSchedule  off        ... -                      Power saving
0xFFFF0005 CN       Running  -                 on         ... PLE(o),NRD(o),FEFS(o)  -
0xFFFF0006 CN       Running  -                 on         ... PLE(o),NRD(o),FEFS(o)  -
0xFFFF0007 CN       Running  -                 on         ... PLE(o),NRD(o),FEFS(o)  -
...
```

The following example shows that jobs cannot be allocated to the compute node with node ID 0xFFFF0004, which the automatic power control function instructed to stop. For the compute node to which jobs cannot be allocated, the REASON field displays other than "DeadlineSchedule" and DETAIL field displays "Power saving."

If the status does not change after 10 minutes, the system management node, compute cluster management node, or compute node has experienced a hardware or network anomaly. The administrator must restart the compute node after eliminating the cause of these anomalies.

```
# pashowclst -c clstname --nodetype CN -v --detail
[ CLST: clstname ]
[ NODETYPE: CN ]
NODE       NODETYPE STATUS   REASON            PWR_STATUS ... SRV_STATUS             DETAIL
0xFFFF0004 CN       Stopped  -                 on         ... -                      Power saving
0xFFFF0005 CN       Running  -                 on         ... PLE(o),NRD(o),FEFS(o)  -
0xFFFF0006 CN       Running  -                 on         ... PLE(o),NRD(o),FEFS(o)  -
0xFFFF0007 CN       Running  -                 on         ... PLE(o),NRD(o),FEFS(o)  -
...
```

See

For details on the pashowclst command, see "Displaying Operation Status of the System" in "Chapter 3 Details of the System Management Function" in "Job Operation Software Administrator's Guide for System Management."

Appendix A Hooks for the Power Cap Scheduling Function (Job Power Estimate Function) and Power Knob Operation Function

The job power estimate function and power knob operation function of the power cap scheduling function use hooks (exit functions provided by the job scheduler function). For details on incorporation of exit functions, see "Job Operation Software Administrator's Guide for Job Operation Manager Hook."

1. Settings for the exit function of the job power estimate function

   - Settings for the job manager exit function

   Define the ExitFunc subsection for the job manager exit function library in /etc/opt/FJSVtcs/Rscunit.d/*ResourceUnitName*/pmpjm.conf file in a resource unit on the system management node.

```
ResourceUnit {
   ...
    ExitFunc {
        ExitFuncLib = libpwrmjpepjm.so
        ExitFuncPri = 127
        ExitFuncType = pjm
    }
}
```

2. Settings for the exit function of the power knob operation function

   - Setting the job manager exit function

   Define the ExitFunc subsection for the job manager exit function library in the /etc/opt/FJSVtcs/Rscunit.d/*ResourceUnitName*/pmpjm.conf file in a resource unit on the system management node.

```
ResourceUnit {
   ...
    ExitFunc {
        ExitFuncLib = libpwrmknobutilpjm.so
        ExitFuncPri = 127
        ExitFuncType = pjm
    }
}
```