## Operations Management Software of Supercomputer Fugaku

Atsuya Uno

Fumichika Sueyasu

Ryuichi Sekizawa

Recent years have seen dramatic progress in the parallelization and scaling up of high performance computing (HPC) systems consisting of supercomputers and clusters of ordinary computers. On the other hand, ensuring that such a large-scale, high-performance computing environment operates in a stable manner for the benefit of users has become a major issue. Operations management software plays an important role in managing and operating a system. The development of operations management software for the supercomputer Fugaku (hereafter, Fugaku) was carried out with a view to achieving stable operation based on continuity with K computer know-how and on the solutions found to operation problems after eight years of operating the K computer and using its software technology. A user-friendly system was also a key pursuit of this development initiative. This article describes enhanced and added functions in operations management software for Fugaku.

## 1. Introduction

The design of operations management software for supercomputer Fugaku (hereafter, Fugaku) [1] was conducted with a view to achieving stable operation based on continuity with K computer know-how [2, 3] and on the solutions found to operation problems after years of operating the K computer and using its software technology. Achieving a user-friendly system was also an important objective. Fugaku operations management software can be broadly divided into system management software, job management software, and user management software, all of which have been enhanced in this development initiative.

In this article, we describe enhanced and added functions in operations management software for Fugaku.

## 2. System management software

System management software manages and operates many units of hardware and software as a single system. In addition to enabling the system to continue overall operation even if part of the system should fail, this software is capable of dividing the system into multiple partitions and to operate each partition as an independent environment as needed.

In general, the larger is the scale of a system the

higher is the possibility that a failure will occur. As a consequence, to achieve high system availability and minimize system downtime, it is essential that the occurrence of anomalies be detected and dealt with rapidly. In Fugaku, the following enhancements have been made based on the analysis of maintenance operations in the K computer.

# 2.1 Enhancement to the system management function

In the event of an anomaly in a compute node, for example, the role of operations management software is to detect that anomaly and remove the node from operation. As a result, taking a long time to detect such an anomaly can have a major impact on system operation. In the K computer, there was only periodic monitoring for anomalies, and as a consequence, there were cases in which some time elapsed before detecting a particular anomaly. In Fugaku, this scheme was changed to one in which a service notifies the monitoring system whenever it experiences an anomaly thereby shortening the time to anomaly detection.

## 2.2 Shortened maintenance time

Improving system availability requires that system

downtime due to maintenance be shortened. In the K computer, it took some time to reboot the system, so there were cases in which software maintenance required as much as two days. In Fugaku, on the other hand, the reboot process has been optimized with the result that reboot time has been shortened to about half that of the K computer. In the boot process of a compute node group, for example, multiple booting that takes into account inrush current, power fluctuations, etc. has been optimized. In addition, the reboot process has been improved by selecting optimal reboot methods, such as node reboot (cold reboot [4]), warm reboot [5], service reboot [6], etc., to meet maintenance needs.

Next, in the case of hardware maintenance, enhancements were made to the maintenance of storage systems that usually requires some time to complete. Reliability, availability, and serviceability (RAS) of Fugaku follows the policy set with the K computer. The management nodes, which can significantly affect system operation, have a redundant configuration. If compute nodes in a non-redundant configuration fail, they are removed from operation and replaced with new units that are then placed in operation. Additionally, if a storage system fails, the software stored on that system may likewise be deleted, in which case the software as well will have to be restored. In the K computer, the failure of a compute node's system disk would require reinstallation and resetting of that software, which was one factor in the lengthening of maintenance work. To deal with this issue, Fugaku adopted a method that stores a system disk as a disk image so that restoration in the event of a failure can be achieved by simply copying that image. This enhancement shortened maintenance work time to about half that of the K computer.

## 2.3 Efficient log analysis

As system scale becomes larger, so does the volume of logs output from servers and other equipment. In the K computer, the format of the logs differed for each function. This made it difficult to check on system operation and look for faults or failures, which made log analysis inefficient and increased the time needed for analysis. In Fugaku, it was decided to unify the log format of operations management software, unify log management, and form links between related logs thereby creating an environment conducive to speedy fault surveys and operations analysis. The shortening of analysis time in this way can contribute to the shortening of maintenance time. In addition, the use of open-source software (OSS) in log collection and analysis makes this work more efficient. Using OSS has other features as well, such as enabling swift introduction of new technologies and flexible response to a variety of phenomena.

## 3. Job management software

The main functions of job management software are a job manager function and job scheduler function, which perform job management, resource management, job scheduling, and other tasks. The following describes enhancements made to these functions.

## 3.1 Job management function

The following enhancements were made to the job manager function in Fugaku.

1) Support of virtual machines

The K computer supported only a job execution environment for physical machines, but Fugaku supports a virtual-machine job execution environment such as kernel-based virtual machine (KVM). A virtualmachine execution environment may be one prepared by the system administrator or one independently prepared by a user. Virtual machines can be booted up in units of jobs and executed without affecting other jobs.

2) Support of custom resources

Fugaku is equipped with a function for managing arbitrary resources (custom resources) on a specific compute node. With custom resources, virtual resources such as power and software licenses can be managed in addition to hardware functions. The user requests what custom resources are needed when submitting a job and the scheduler automatically allocates compute nodes with those resources to the job. Since the software license for a commercial application can also be defined as a custom resource, the user may use commercial applications without having to worry about the number of software licenses needed.

## 3.2 Job scheduler function

The following enhancements were made to the job scheduler function in Fugaku.

1) Enhanced user interface

Providing easy-to-use commands for all users is difficult. Fugaku has an application programming interface (API) for customizing operation commands featuring an enhanced option system for various types of commands and the ability to implement commands independently. This API enables users themselves to customize commands.

2) Customization function for scheduling algorithm

In addition to the scheduling algorithm provided as standard, Fugaku is equipped with a scheduler API that enables system operation for each computer center to be optimized (**Figure 1**). Among the various job scheduler functions, this API enables the system administrator to customize the (a) job selection section and (b) resource selection section. Here, (a) is a function for controlling priority at job scheduling and (b) is a function for controlling the allocation of compute nodes. The scheduler API makes it possible to replace this functional section with original algorithms as needed.

3) Enhanced node usage rate

In the K computer, a job could only be executed up to a designated elapsed time at most. Consequently, if an interval opened up before executing the following job, that compute node would enter an idling state thereby lowering the node usage rate. To improve upon this situation, Fugaku adopted a new job execution method that enables a job to continue execution as long as it does not impede the execution of the following job (**Figure 2**). This method dynamically modifies the elapsed time limit after job execution reaches the user-designated elapsed time according to the scheduling conditions of the following job. This scheme can improve node usage rate.

Also installed is a scheduling function that aims to improve node usage rate while maintaining job I/O performance. As shown in Figure 3, Fugaku takes on a configuration that shares storage I/O (SIO) nodes, which manages 1st layer storage, in 48-node units (shelf unit). To therefore enable a job to occupy the I/O bandwidth in 1st layer storage, scheduling must be performed while taking into account the Tofu network [7] that interconnects SIO nodes with compute nodes. In this case, a shelf unit (2×3×8) becomes a scheduling unit, so a scheduling function that takes this format into account has been installed. However, if small jobs come to be scheduled in a dispersed position, the I/O of large jobs may be hindered. To prevent this, a function that groups together small-scale jobs on a specific shelf as much as possible has also been installed. This function is expected to improve the node usage rate while



Figure 1 Customizable functions.

Job execution exceeds the designated elapsed time and terminates immediately before the following job



Job execution exceeds the designated elapsed time and terminates immediately before the following job

Forced termination when designated elapsed time is reached even if no following job exists



K computer



#### Figure 2

Dynamic modification of elapsed time limit.



Figure 3 Internal configuration of compute rack.

maintaining I/O performance even in an environment with a mixture of large and small jobs.

4) Approaches to power problems

Fugaku consumes more power than the K computer, so it must be operated in a way that the power consumed by the entire system is kept within an allowable range for the facility housing that system. In the K computer, there were cases in which power consumption would exceed the upper limit when executing a large-scale job using the entire system. To prevent this upper limit from being exceeded, a control scheme was adopted that estimated the power that would be consumed by a large-scale job based on the results of measuring the power consumed by small-scale jobs beforehand. In Fugaku, on the other hand, the upper limit may be exceeded even during times of normal operation. In response to this problem, Fugaku installs a function that estimates the power consumed by each job based on past execution results and schedules jobs so that the power consumed by the entire system does not exceed the upper limit.

## 4. User management software

User management software performs user account management and resource management. This section describes enhancements to user management software in Fugaku.

## 4.1 Extensions to charging management

In the K computer, computing resources were managed in units of research project, the same as in ordinary computer systems. Thus, if a research project were to consist of multiple groups, the user could not freely adjust the amount of resources allocated to the groups within the project. To solve this problem, Fugaku adopts a hierarchical structure for managing computing resources so that the user of each project can freely move the amount of resources between groups. The hierarchical structure of resource management is shown in **Figure 4**. In the K computer, projects were allocated to groups, but in Fugaku, they are allocated to subthemes. Once the operator has completed theme setting, each user can manipulate the amount of resources at levels lower than the subtheme group.

Resource management in the K computer was performed solely on the basis of "node hours" defined as the product of job execution time and the number of nodes. In Fugaku, a functional extension has been made so that the amount of resources can be managed in any manner other than "node hours." In this way, any resource from which the information needed for calculating charges can be acquired can be targeted for management such as "amount of power used."



Figure 4 Hierarchical structure of resource management.

# 4.2 Enhanced operations during a file system fault

In Fugaku, as in the K computer, the 2nd layer file system that stores user data is made up of multiple volumes. In the K computer, if a fault occurred in the file system, it was not possible to detach only the failed volume and continue operation. As a result, there were cases in which operation had to be halted because of fault recovery work. To solve this problem, Fugaku is equipped with a function that links with the system management function to avoid use of the failed volume so that operation can continue. This function enables operation to continue without halting the entire system even if a particular volume should fail.

## 5. Conclusion

This article introduced enhanced and added functions in operations management software for the supercomputer Fugaku. In actual operation, a variety of measures will be needed while using these functions according to operating conditions. For example, new technologies for achieving Society 5.0 [8] such as cloud-oriented use and the use of big data and AI will need to be supported. The development initiative described here placed importance on extendibility and flexibility based on operating experience with the K computer, but supporting new and emerging technologies is an important issue going forward.

All company and product names mentioned herein are trademarks or registered trademarks

## **References and Notes**

- [1] RIKEN Center for Computational Science: About the Supercomputer Fugaku.
  - https://www.r-ccs.riken.jp/en/fugaku/project
- [2] Information Processing Society of Japan: "Special Issue: The K Computer." IPSJ MAGAZINE. Vol. 53, No. 8, pp. 752–807 (2012). (in Japanese) https://ipsj.ixsq.nii.ac.jp/ej/index.php?action=pages\_ view\_main&active\_action=repository\_view\_main\_ item\_snippet&index\_id=6578&pn=1&count= 20&order=7&lang=japanese&page\_id=13&block\_id=8
- [3] RIKEN Center for Computational Science: What is K? https://www.r-ccs.riken.jp/en/k-computer/about/
- [4] Reboot from a power-off state.
- [5] Reboot from a non-power-off state.
- [6] Reboot of only related programs.

[7] Y. Ajima: High-dimensional Interconnect Technology for the K Computer and the Supercomputer Fugaku. Fujitsu Technical Review. https://www.fujitsu.com/global/about/resources/

publications/technicalreview/topics/article005.html[8] Cabinet Office: What is Society 5.0?

https://www8.cao.go.jp/cstp/english/society5\_0/index.html



#### Atsuya Uno

RIKEN Center for Computational Science, System Operations and Development Unit and Computer Technologies Division, Unit Leader Dr. Uno is engaged in research and development related to the operation and enhancement of supercomputer systems.



#### Fumichika Sueyasu

Fujitsu Limited, Technical Computing Business Unit Mr. Sueyasu is engaged in the deployment and operation of the supercomputer Fugaku.



**Ryuichi Sekizawa** Fujitsu Limited, Technical Computing Business Unit Mr. Sekizawa is engaged in the deployment and operation of the supercomputer Fugaku.

This article first appeared in Fujitsu Technical Review, one of Fujitsu's technical information media. Please check out the other articles.

Fujitsu Technical Review

https://www.fujitsu.com/global/technicalreview/

