

Classifying jobs towards power-aware HPC system operation through long-term log analysis

Yuichi Tsujita ^{a,*}, Atsuya Uno ^a, Ryuichi Sekizawa ^b, Keiji Yamamoto ^a, Fumichika Sueyasu ^b

^a RIKEN Center for Computational Science, Kobe, Hyogo, Japan

^b Fujitsu Limited, Tokyo, Japan

ARTICLE INFO

Keywords:

Classification
Machine learning
Electric power
FLOPS
Memory bandwidth
File I/O

ABSTRACT

The efficient utilization of high-performance computing (HPC) system resources under rigorous electric power budget or I/O workload constraints is among the most important goals set by system operators to deal with the demanding requirements of application users. In most cases, the effective utilization of CPU and memory devices, which is tightly linked to electric power consumption, is a counterpart metric of I/O activities in most HPC jobs. Towards higher utilization of HPC systems under strict electric power consumption and I/O activity management constraints, we must be careful to prevent hot-spots from developing in power consumption or I/O operations that could lead to unstable system operations by exceeding electric power supply or I/O subsystem capabilities. One of the feasible solutions is arranging compute node assignment not to have such hot-spots in electric power or I/O operations. To address this issue, we analyzed vast amounts of log data collected from the K computer and found strong positive correlations between CPU and memory device utilization rates and electric power consumption levels. On the one hand, we also observed strong negative correlations and reduced electric power consumption in relation to file I/O activities in a specific compute node-layout, thereby indicating unique characteristics in some I/O-intensive HPC jobs in the node-layout. Our investigation revealed that HPC jobs could be divided into two groups when classified in terms of required electric power — jobs consuming high electric power levels and I/O-intensive jobs with reduced electric power levels. Then, we achieved high levels of accuracy when classifying jobs in terms of electric power levels using `RandomForestClassifier` among multiple machine learning classification models provided from `scikit-learn`. The classification can prevent us from hot-spots in electric power consumption in compute node assignment in job scheduling. Thus we demonstrated efficient job classifications towards power-aware system operations in the supercomputer Fugaku, which is the successor to the K computer.

1. Introduction

In our current era, high-performance computing (HPC) systems with huge numbers of CPU cores are now approaching exascale performance levels. The primary mission of HPC systems is to provide stable computing resources with high levels of utilization and usability under specific constraints such as established power consumption and I/O workload levels. However, increases in computing power require advanced storage systems of a similar scale with equally high I/O performance levels. Additionally, current and future HPC systems must accommodate wide varieties of jobs, including compute-, memory-, and I/O-intensive applications. In such situations, we have found that high utilization levels of CPU and memory devices tend to lead to high electric power consumption levels, while I/O-intensive operations tend to result in reduced electric power usage.

Our current HPC system, the supercomputer Fugaku [1] (hereinafter, Fugaku), is the successor to the K computer. Compared with the K computer, Fugaku has advanced management functions for electric power consumption that ensure more stable and effective electric power usage [2,3]. Even though the K computer was decommissioned in August 2019, analyses of its power consumption during HPC job executions have already provided some useful hints for more power-aware operations using Fugaku. This is because, over its years of the K computer operation, we collected a wide variety of system logs and job stats, including performance counters related to CPU and memory devices, file I/O activities, and even environmental metrics such as system board airflow and CPU cooling water temperatures. However, since the K computer did not have any electric power measurement devices in its compute node modules, it was necessary to estimate

* Corresponding author.

E-mail address: yuichi.tsujita@riken.jp (Y. Tsujita).

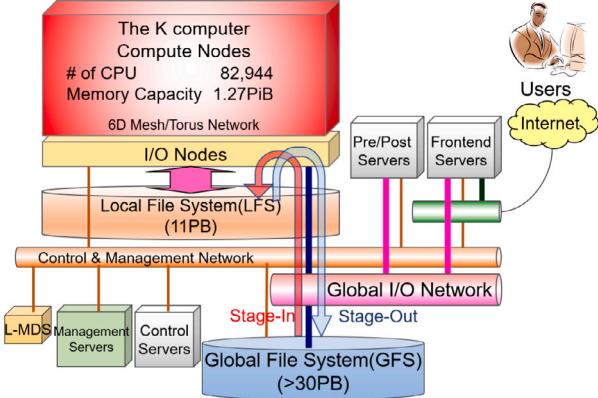


Fig. 1. K computer system overview.

the electric power of each compute node using the abovementioned temperature information and statistical information from our previous study [4].

Our log analyses regarding jobs executed in the K computer revealed unique correlations among CPU, memory, file I/O activities, and the compute node-layouts used. After further analysis, we determined that jobs could be split into two groups based on electric power levels — jobs requiring higher electric power levels due to high CPU and memory device utilization and jobs consuming relatively lower levels of electric power such as I/O-intensive applications. These preliminary results were reported in our conference paper [5]. Additional analyses were made by using an additional classification model to examine the usefulness of our approach. We also modified evaluation scheme for the classification models by introducing Matthews Correlation Coefficients (MCC) [6]. Furthermore, we conducted factor loading analyses of the metrics used to examine contribution rates. Our approach, which we discuss in this paper, is based on the following two-step method. First, we analyze correlation coefficients among the metrics used in job stats. Then, based on the results obtained in the first step, we perform classifications using those metrics and a machine learning (ML) scheme. The classification process leverages useful information towards more effective power-aware compute node allocation and job scheduling that is expected to be useful not only for Fugaku but for other HPC platforms as well.

The rest of this paper is organized as follows. In Section 2, we present an overview of the K computer, including the file I/O subsystems and electric power-related metrics. Correlation analysis using job stats log is then reported in Section 3, followed by a prediction model discussion based on an ML approach in Section 4. We then discuss related work in Section 5 before finally concluding the paper and discussing our future work plans in Section 6.

2. K computer and its electric power log collection

2.1. K computer overview

After seven years of system operations, the K computer was decommissioned in August 2019. During its operational period, vast amounts of log data on system operations, facility operations, and job stats log were collected from the K computer, and were analyzed not only for root-cause information concerning system failures or performance degradation but also by researchers searching for ways to improve system operation quality. Fig. 1 shows an overview of the K computer. The K computer consisted of 82,944 compute nodes and 5,184 I/O nodes. In operation, the compute nodes and I/O nodes were connected via high speed and low latency interconnects named Tofu [7] developed by Fujitsu. In the two-level file system that was introduced for the K

computer, the local file system (LFS) was a scratch high-performance storage space dedicated to file I/O operations during computations, while the global file system (GFS) was used to store programs and data with high redundancy. An enhanced Lustre file system named Fujitsu Exabyte File System (FEFS) [8] that was based on Lustre version 1.8 was used to build both of the K computer file systems. When combined with asynchronous file staging scheme [9], this two-level file system improved job scheduling efficiency and mitigated I/O interference.

Each compute node in the K computer had about 150 GB of local disk space on the LFS, and every HPC job was assigned to compute nodes with guaranteed amounts of available disk space. Free disk space up to 100 GB per node was set for job scripts using the *node-quota* option. However, when the *node-quota* description was not given in job scripts, a minimum size (14 GB per node) was set as the default. Specified programs or datasets in a job script were copied from the GFS to the assigned disk space on the LFS in the stage-in phase. After job execution, the specified output data in the script were copied back from the LFS to the GFS in stage-out phase. When asynchronous file staging was used in the K computer, successive HPC jobs could jump to the stage-in phase once the K computer job scheduler system found enough disk space for the jobs on target compute nodes. For other queued jobs, the job scheduler system conducted periodic checks of the compute nodes until the required disk space was available and then matched those jobs to the appropriate compute nodes.

Fig. 2 depicts the configuration of compute nodes and I/O nodes in a single cabinet of the K computer associated with the LFS. Every cabinet consisted of two system racks, and each system rack consisted of 96 compute nodes and six I/O nodes. Compute nodes and I/O nodes were connected through the Tofu interconnect in a six-dimensional (6D) mesh/torus network represented by *X*, *Y*, *Z*, *A*, *B*, and *C*. Tofu links of *X*, *Z* and *B* were connected in a torus configuration, while those of *Y*, *A*, and *C* were connected in a mesh configuration.

Compute node allocation for each job was configured based on the number of compute nodes or the node-layout specified in a job script in a three-dimensional (3D) rather than 6D manner, in which the first dimension represented *X* × *A*, the second represented *Y* × *B*, and the third represented *Z* × *C*. The minimum unit of compute node allocation in the K computer consisted of 12 compute nodes configured by the $2 \times 3 \times 2$ in a 3D way. This layout was also represented by $1 \times 1 \times 1 \times 2 \times 3 \times 2$ in a 6D expression. Therefore, if a specific node-layout was desired, the users needed to give multiple numbers of 2, 3, and 2 in the first, second, and third dimensions in a 3D manner. The job scheduler searched for free adjacent compute nodes in a block-wise manner using a variety of 3D patterns in which the number of nodes in each axis was an integral multiple of the number of nodes in the corresponding axis in the minimum unit. If a job script specified the number of nodes or node-layout that did not fit the above condition, the job scheduler attempted to meet the number of required compute nodes by adding additional compute nodes in a way that allowed the node-layout to fit the above conditions. The job scheduler also allocated compute nodes among 3D candidates by rearranging the number of nodes in each axis if there were no node-layout restrictions.

Note that the *Z*-link torus configuration depicted by the vertical connections in this figure was only available in I/O accesses through I/O nodes. Once the compute node-layout was fixed, I/O nodes on the same *Z*-links were assigned to handle the I/O operations, and I/O paths were routed to the corresponding I/O nodes with the help of I/O zoning scheme [10]. This I/O zoning scheme was introduced to mitigate I/O interference happened at Object Storage Targets (OSTs) and I/O nodes by assigning I/O nodes and OSTs to the same *Z*-link used by the compute nodes. In other cases, application jobs used the *Z*-link in the mesh configuration for inter-node communications. Since I/O node-layout of the K computer was dependent on the compute node-layout facilitated by the I/O zoning scheme, some I/O-intensive jobs tended to specify fixed 3D node-layouts to guarantee I/O performance. In such cases, the most promising way to avoid I/O interference was

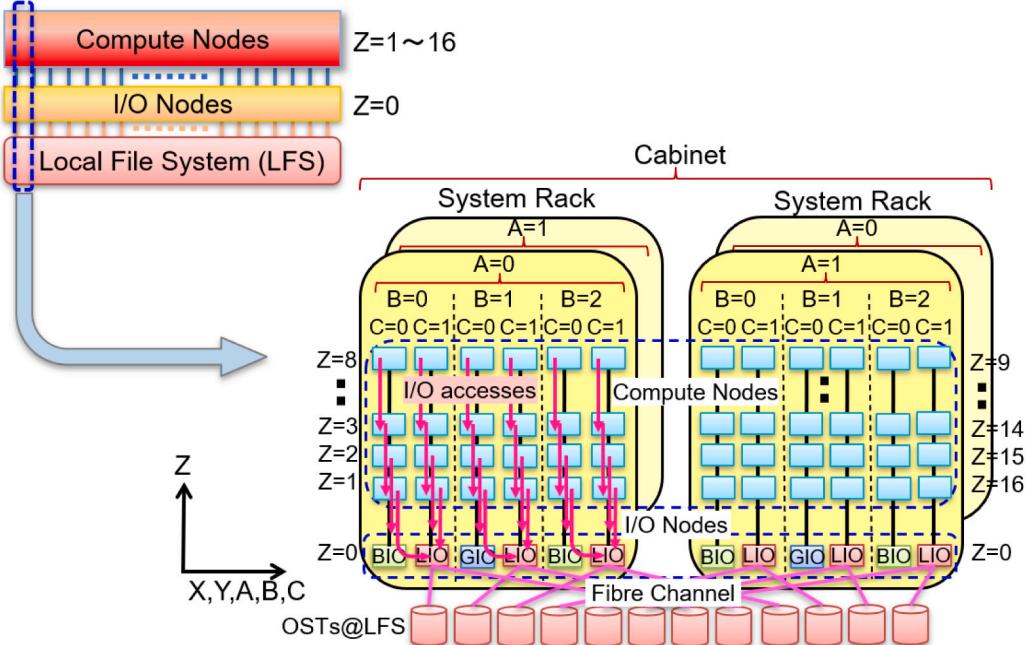


Fig. 2. LFS I/O subsystems with compute nodes and I/O nodes connected through Tofu interconnects.

found to be placing all the compute nodes on the same Z -link by giving 32 ($= 16 \times 2$) in the last dimension of the 3D node-layout in the job submission. This specification resulted in compute node layouts ranging from $Z = 1$ to 16 at $C = 0$ and 1 with the same values in A , B , X , and Y within the same cabinet.

2.2. Electric power log collection

As mentioned above, since the K computer did not have any electric power measurement devices, we established a prediction model [4] using the following two averaged temperature differences among the following temperature sensors:

- Temperature sensors for CPU and inlet cooling water at each compute node module ($\overline{\Delta T}_{CPU}$)
- Temperature sensors for inlet and outlet of air-flow at each system board consisting of four compute node modules ($\overline{\Delta T}_{air}$)

We observed that the consumed electric power had strong correlation with $\overline{\Delta T}_{CPU}$ and $\overline{\Delta T}_{air}$. Therefore we assumed that the electric power was proportional to the two temperature values. We obtained each coefficient as a mean value using measured data about 33,500 jobs from April 2014 to November 2014. The following formula was obtained to predict the electric power used for the K computer (P_K) in our previous study [4]:

$$\begin{aligned} P_K[\text{MW}] &= 0.802393382361262 \times \overline{\Delta T}_{CPU} \\ &+ 0.345223838880426 \times \overline{\Delta T}_{air} \\ &+ 7.67202252302052 \end{aligned}$$

In order to evaluate correctness of the obtained formula, we made regression analysis between measured electric power and estimated one at the special rack which equipped with electric power meter for this analysis. Consequently, we confirmed that the formula achieved enough correctness through about 13,000 jobs from December 2014 to February 2015 [4].

Note that the 10 MW value referred to electric power without any computation and I/O loads, and it was included in the P_K [4]. In order to characterize each job through electric power fluctuations, electric power values per compute node without the 10 MW (P_{node})

were obtained by $P_{node}[\text{MW}] = (P_K[\text{MW}] - 10[\text{MW}])/82944$, where 82,944 is the total number of compute nodes in the K computer. Note that the power measurement was carried out for the prediction study, and we did not have any electric power data in the usual system operation. Therefore, we have utilized the above formula to predict electric power during the usual system operation.

The disk utilization ratio relative to the allocated disk space provides another insight into the file I/O status of each job and its associated activities. In general, CPU and memory device utilization, both of which are tightly associated with electric power consumption, tend to decrease with an increase in I/O operation workload and vice versa. Additionally, the same or similar job scripts are likely to be used repeatedly in HPC jobs [11]. This indicates that investigating I/O activities from past job stats log data can be expected to yield insights about the electric power levels of each job. The results of such analyses, when combined with job stats log data collected from the K computer, can be expected to provide useful information towards ensuring more power-aware HPC system operations in Fugaku.

For system operation analysis, we collected many metrics including predicted electric power in the PostgreSQL database. Stored information was accessible through our dashboard service built by Redash [12]. Some of principal metrics are listed in Table 1.

Note that P_{node} values were predicted at five-minute increments during each job execution, and the log information kept the maximum, minimum, and mean values of P_{node} in P_{node}^{\max} , P_{node}^{\min} , and P_{node}^{mean} , respectively. We could have various offline analysis by extracting target metrics from the redash service.

3. Correlation analysis of HPC job activities

Through our I/O activity analyses using job stats log data collected from the K computer, the following questions arose:

- How much of the allocated disk space specified by *node-quota* did the users utilize?
- Which metrics were most tightly related to I/O activities and electric power?
- How can we include electric power awareness when classifying HPC jobs for compute node allocations?

Table 1
List of log information collected for system operation analysis.

Metric	Description
JOB-ID	Unique JOB-ID given for each job
User-ID	User-ID of the submitted job
QUE_CLS	Queue class used for the job
REQ_X	Requested number of compute nodes in Tofu X-link direction
REQ_Y	Requested number of compute nodes in Tofu Y-link direction
REQ_Z	Requested number of compute nodes in Tofu Z-link direction
ND_X	Allocated number of compute nodes in Tofu X-link direction
ND_Y	Allocated number of compute nodes in Tofu Y-link direction
ND_Z	Allocated number of compute nodes in Tofu Z-link direction
$T_{\text{elapse}}^{\text{limit}}$	Elapsed time limit for the job
T_{elapse}	Elapsed time of the executed job
NQ	The allocated disk space per compute node specified by <i>node-quota</i> option
$S_{I/O}$	Total bytes of file I/O
R_F	The ratio of sustained FLOPS relative to the peak FLOPS
R_M	The ratio of sustained memory bandwidth utilization relative to the theoretical bandwidth
$P_{\text{node}}^{\text{max}}$	The maximum value of P_{node}
$P_{\text{node}}^{\text{min}}$	The minimum value of P_{node}
$P_{\text{node}}^{\text{mean}}$	The mean value of P_{node}

In our search for answers, using job stats log data from the second half term of 2016 to the first half term of 2018 of the Japanese fiscal year, we selected the following metrics for each job that ran equal to or more than 10 min without any errors in the largest queue class named “large”. There were two reasons for using log data during this two-year term; the second half term of 2016 was chosen as the start time because of also marked the establishment of the abovementioned electric power prediction process using temperature log data, while the first half term of 2018 was selected as the end time because we began conducting cooling system examinations (in which cooling water temperatures were intentionally changed) from the second term of 2018. This meant we could not use the estimation model to predict electric power during that period.

The following metrics were used from the log data:

- NQ : The allocated disk space per node by *node-quota*
- ND_Z : The number of allocated compute nodes in Tofu Z-link direction
- R_D : The ratio of used disk space relative to the assigned disk space specified by the *node-quota* option
- $R_{I/O}$: The ratio of used disk space relative to the maximum achievable amount of I/O size
- R_F : The ratio of sustained FLOPS relative to the peak FLOPS
- R_M : The ratio of sustained memory bandwidth utilization relative to the theoretical bandwidth
- $P_{\text{node}}^{\text{max}}$: The maximum electric power per compute node in situations where the electric power basement has been removed

Note that the “large” queue class covered about 90% of the K computer compute nodes and accepted jobs using from 385 to 36,864 compute nodes. Hereinafter, we describe jobs executed in this queue class as large jobs.

While four of the abovementioned metrics are related to file I/O activities, the rest are related to electric power. The two file I/O-related metrics, R_D and $R_{I/O}$, are clarified as follows:

$$D_{I/O} = S_{I/O}/ND_X/ND_Y/ND_Z$$

$$R_D = D_{I/O}/NQ$$

$$R_{I/O} = D_{I/O}/\min(NQ, BW_{I/O}^{\text{max}} \times T_{\text{elapse}}^{\text{limit}})$$

$D_{I/O}$, $BW_{I/O}^{\text{max}}$, and $T_{\text{elapse}}^{\text{limit}}$ are the actual file I/O size per node, maximum I/O bandwidth, and elapsed time limit, respectively. Note that $BW_{I/O}^{\text{max}} \times T_{\text{elapse}}^{\text{limit}}$ is an achievable maximum I/O size amount if a job utilizes 100% of the I/O bandwidth and is within the elapsed time

limit. However, it is limited by the *node-quota*. In this study, we assume 100 MB/s for $BW_{I/O}^{\text{max}}$.

It was noted that I/O-intensive jobs tended to require compute nodes in the fixed 3D node-layout in order to prevent I/O-intensive applications from encountering I/O interference from other jobs. Consequently, such jobs achieved high I/O bandwidth levels. However, the CPU and memory utilization ratios of such jobs were lower than those in other node-layout cases. Additionally, since jobs set in the fixed 3D node-layout tended to utilize larger amounts of disk space compared with other node-layout cases, we found some dependency in the required compute node-layout in terms of file I/O.

The job stats were analyzed separately based on the requested compute node-layout in one-dimensional (1D), two-dimensional (2D), and 3D shapes. In the log information, 1D jobs had the number of nodes only in REQ_X and zero in REQ_Y and REQ_Z . 2D jobs had the numbers in REQ_X and REQ_Y and zero in REQ_Z . Lastly, 3D jobs had the numbers in REQ_X , REQ_Y , and REQ_Z . Since the number of 2D jobs was negligibly small compared with 1D and 3D jobs, we analyzed 1D and 3D jobs from the large jobs. The 3D jobs were further separated into two groups, with and without changes in the compute node-layout, where we describe them as “3D(malleable)” and “3D(same)”, respectively. We separated 3D jobs into the 3D(same) if

$$(REQ_X = ND_X) \wedge (REQ_Y = ND_Y) \wedge (REQ_Z = ND_Z)$$

Otherwise we separated the job into the 3D(malleable) in this study. Correlation analyses were performed among R_D , R_F , and R_M , where R_D represents file I/O activities and R_F and R_M correspond to computing activities. As mentioned previously, higher file I/O activities in an HPC job tend to indicate lower electric power levels. In contrast, higher computing activities in an HPC job tend to result in increased electric power. For these correlation coefficient evaluations, we utilized a *Python* module; and to eliminate the dependency of individual correlation functions, we utilized the following three correlation functions provided by the *SciPy* [13] package:

- The Pearson product-moment correlation coefficient using `scipy.stats.pearsonr` with metrics converted in log-scale (hereinafter, Pearson(Log))
- The Spearman’s rank-order correlation coefficient using `scipy.stats.spearmanr` (hereinafter, Spearman)
- The Kendall rank-order correlation coefficient using `scipy.stats.kendalltau` (hereinafter, Kendall)

It should be noted that the converted metrics in log-scale are used for the Pearson product-moment correlation evaluation in order to mitigate the impact of outliers as has been addressed in other study [14].

3.1. Correlation between R_D and R_F

Since it is understood that I/O-intensive applications tend to lower CPU utilization rates, we next examined the correlation between R_D and R_F with colored plots describing NQ and $P_{\text{node}}^{\text{max}}$ in Fig. 3.

In Fig. 3(a), it can be seen that the higher R_D we have, the lower R_F we achieved, and vice versa. It is also noted that several red-colored plots near zero for R_D are jobs that showed low disk space utilization even though large amounts of disk space had been allocated by the higher NQ value. Separately, Fig. 3(b) indicates that some jobs achieved a higher ratio in R_F while they achieved a quite low ratio near zero in R_D , thus resulting in higher electric power consumption.

Figs. 3(c) and 3(d) show that the jobs in this group are localized with low values in both R_D and R_F in the case of 3D(malleable) jobs, and that the NQ values given in this case are also small. Meanwhile, 3D(same) jobs in Figs. 3(e) and 3(f) show two distinct groups. Here, one group consists of I/O oriented jobs spreading from zero to one in R_D and localizing around zero in R_F , while the other group consists of compute-intensive jobs achieving higher ratios in R_F , followed by achieving higher $P_{\text{node}}^{\text{max}}$, as shown in Fig. 3(f).

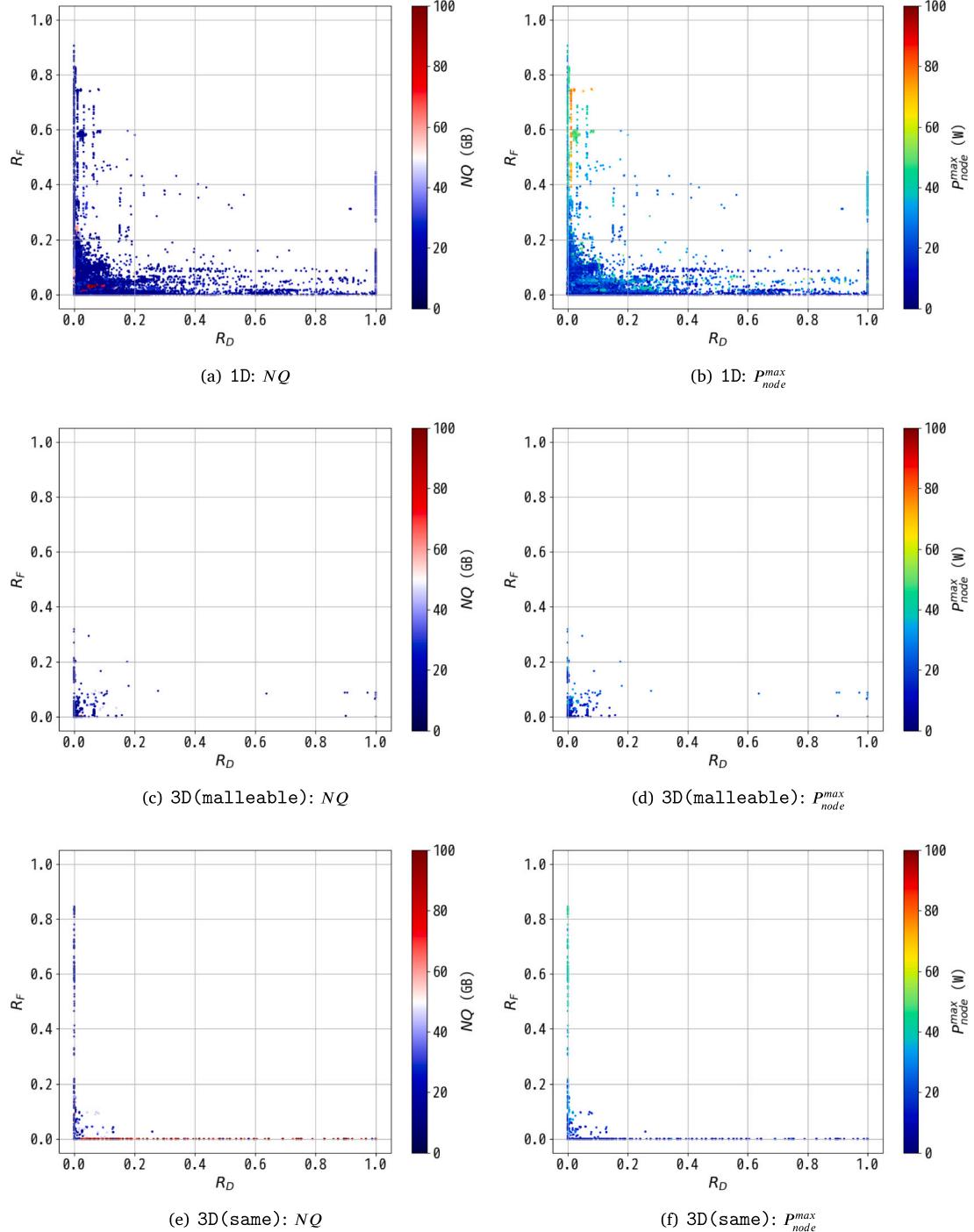


Fig. 3. Correlation between R_D and R_F with indicating NQ and P_{node}^{max} in each color-bar about jobs in the three node-layout cases, 1D, 3D(malleable), and 3D(same). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Correlation between R_D and R_M

Since memory bandwidth utilization is another key metric that tightly corresponds to electric power, we examined correlation between R_D and R_M using colored plots describing NQ and P_{node}^{max} , as shown in Fig. 4.

The correlation between R_D and R_M shows similar behavior. Concerning the 1D jobs shown in Figs. 4(a) and 4(b), we can see that jobs with lower R_D tend to achieve higher ratios in R_M , and that such higher R_M leads to higher P_{node}^{max} . It is also noted that we could separate the 1D jobs into three groups about the P_{node}^{max} along R_M , as shown in

Fig. 4(b). The lower group up to about 30 W contained jobs ranging up to 0.2 in R_M , the next group from about 30 W to about 60 W contained jobs ranging from 0.2 to 0.5 in R_M , and the last group, which exceeded about 60 W contained jobs exceeding 0.5 in R_M . This grouping scheme in terms of P_{node}^{max} is discussed in greater detail in Section 4.1.

3D(malleable) jobs are localized in lower R_D as shown in Figs. 4(c) and (d), where lower NQ values were specified and most of the jobs achieved lower P_{node}^{max} . While the 3D(same) jobs in Figs. 4(e) and (f) show two groups; one ranging from zero to one in R_D with a lower ratio in R_M , and another that achieved a somewhat higher P_{node}^{max} with an increase in R_M at around zero in R_D .

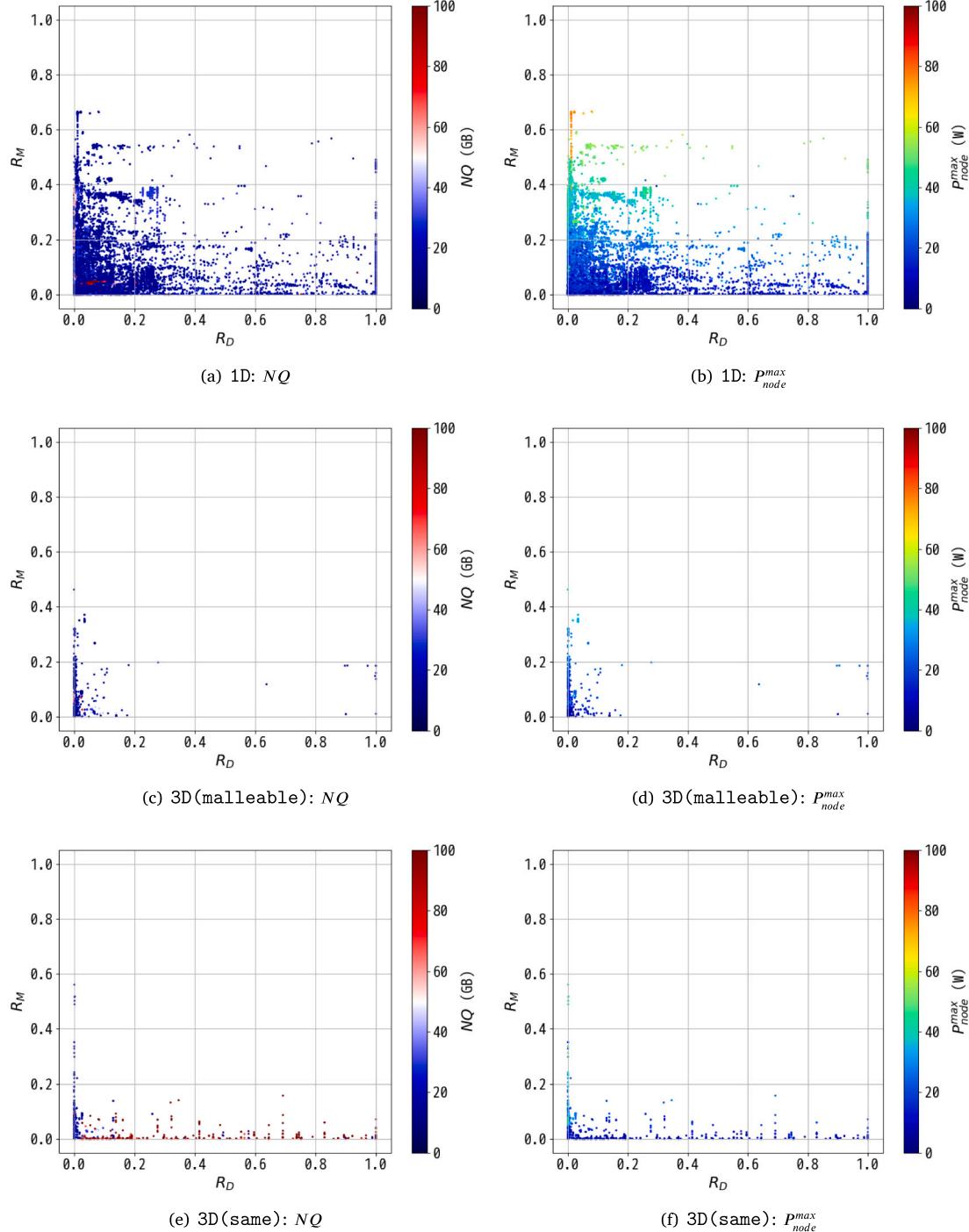


Fig. 4. Correlation between R_D and R_M with NQ and P_{node}^{max} in each color-bar providing indications about jobs in the three node-layout cases, 1D, 3D(malleable), and 3D(same). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3. Correlation between R_M and R_F

As described previously, the major sources of electric power consumption by compute nodes are CPU and memory devices, which means that compute- or memory-intensive HPC jobs tend to consume higher levels of electric power. We examined correlation between R_M and R_F with colored plots that indicate NQ and P_{node}^{max} .

Fig. 5 shows the correlations between R_M and R_F of each job in 1D, 3D(malleable), and 3D(same) job cases with the NQ and P_{node}^{max} shown in colored plots.

It is notable that linear dependencies were detected between R_M and R_F with several coefficient rates since most of the 1D jobs were compute- or memory-intensive. Fig. 5(a) shows some jobs with higher values more than 50 GB/node in NQ . In Fig. 5(b), jobs requiring high electric power levels with high utilization ratios in both R_M and R_F are shown in orange-colored plots. Here, it can be seen that the closer the linear coefficient rate approaches one, the higher P_{node}^{max} increases. We also noted that it is possible to separate the 1D jobs into the same three groups in terms of P_{node}^{max} , as was done in Section 3.2. This will be discussed further in Section 4.1.

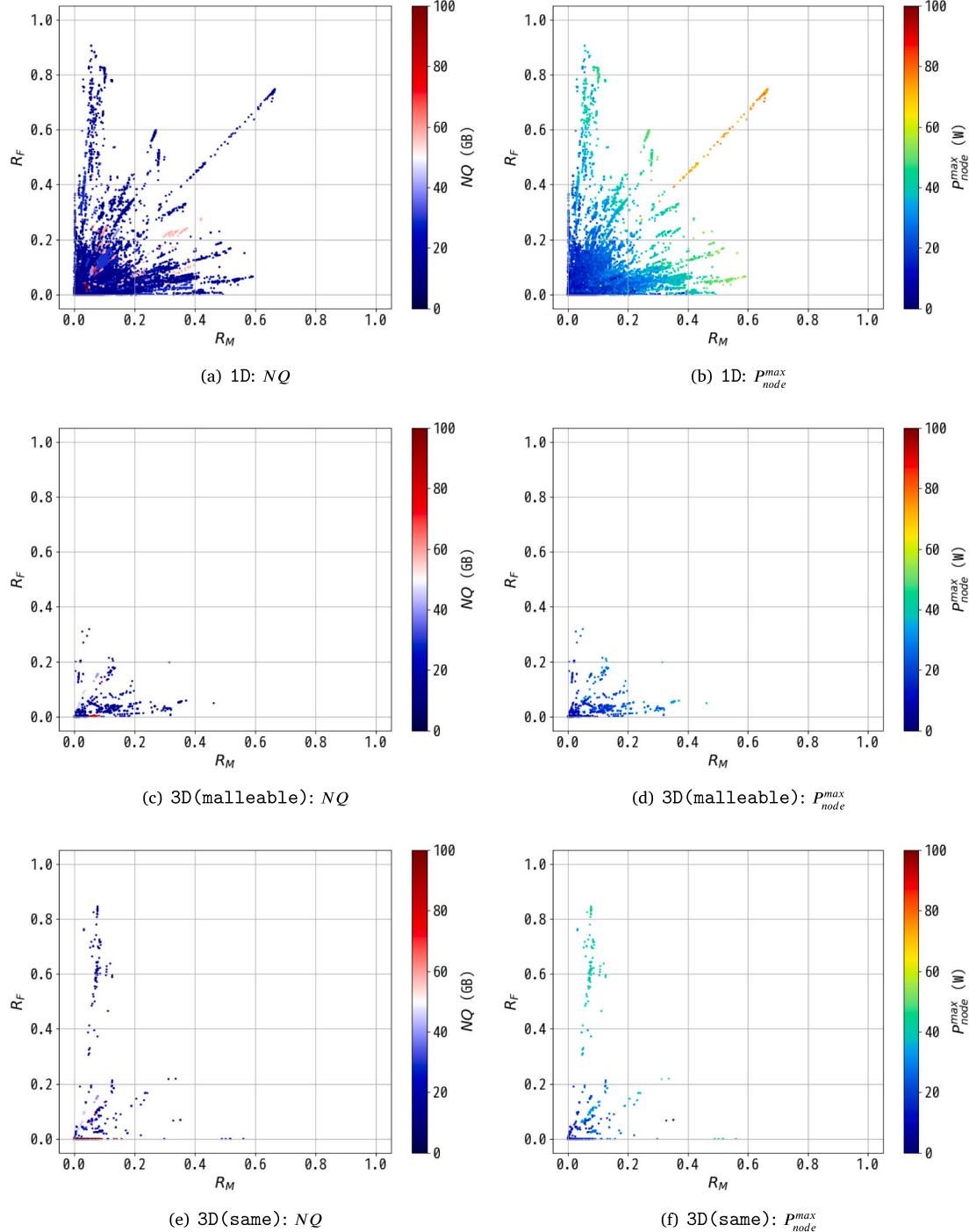


Fig. 5. Correlation between R_M and R_F with NQ and P_{node}^{\max} shown in each color-bar for jobs in the three node-layout cases, 1D, 3D(malleable), and 3D(same). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

When compared with the 1D jobs, the number of 3D jobs was smaller, and the 3D job situations were different from the 1D jobs. The figures in this group indicate lower utilization in both R_M and R_F . Jobs with NQ values higher than 50 GB/node were observed with lower values in the range from 0 to 0.2 in R_M and R_F , as shown in Fig. 5(c). Fig. 5(d) shows us that most of the jobs consumed relatively lower electric power and that jobs with relatively high electric power around 40 W had low NQ , as shown in Fig. 5(c).

On the other hand, the 3D(same) job situation showed differences from the 3D(malleable) job situation, and it was seen that there are jobs with higher R_F that have linear relationships with R_M . In Fig. 5(e),

it can be seen that there are jobs with high NQ at around 0 in R_F and in the range from 0 to 0.2 in R_M . Meanwhile jobs with higher P_{node}^{\max} around 40 W show higher values in R_F or R_M in Fig. 5(f). These jobs are somewhat more compute- or memory-intensive than I/O-intensive because of their very low NQ values.

3.4. Correlation coefficients among metrics

During the abovementioned correlation examinations, we also used the Pearson(Log), Spearman, and Kendall functions to evaluate the correlation coefficients of six metrics used with the P_{node}^{\max} in each

Table 2

Correlation coefficients (upper) and p-values (lower) between each metric used (NQ , ND_Z , R_D , $R_{I/O}$, R_F , and R_M) and P_{node}^{max} .

Node-layout (# Jobs)	Metric	Used correlation functions		
		Pearson(Log)	Spearman	Kendall
1D (127,439)	NQ	0.146 ($p < 0.001$)	0.203 ($p < 0.001$)	0.161 ($p < 0.001$)
	ND_Z	-0.0158 ($p < 0.001$)	0.0116 ($p < 0.001$)	0.00704 ($p < 0.001$)
	R_D	-0.187 ($p < 0.001$)	-0.123 ($p < 0.001$)	-0.0760 ($p < 0.001$)
	$R_{I/O}$	-0.181 ($p < 0.001$)	-0.123 ($p < 0.001$)	-0.0760 ($p < 0.001$)
	R_F	0.649 ($p < 0.001$)	0.693 ($p < 0.001$)	0.499 ($p < 0.001$)
	R_M	0.749 ($p < 0.001$)	0.756 ($p < 0.001$)	0.565 ($p < 0.001$)
3D(malleable) (1,512)	NQ	0.0266 ($p < 1$)	0.0901 ($p < 0.001$)	0.0722 ($p < 0.001$)
	ND_Z	0.0455 ($p < 0.1$)	0.239 ($p < 0.001$)	0.182 ($p < 0.001$)
	R_D	-0.0247 ($p < 1$)	-0.0784 ($p < 0.01$)	-0.0505 ($p < 0.01$)
	$R_{I/O}$	-0.0254 ($p < 1$)	-0.0785 ($p < 0.01$)	-0.0505 ($p < 0.01$)
	R_F	0.613 ($p < 0.001$)	0.662 ($p < 0.001$)	0.459 ($p < 0.001$)
	R_M	0.770 ($p < 0.001$)	0.773 ($p < 0.001$)	0.571 ($p < 0.001$)
3D(same) (985)	NQ	-0.460 ($p < 0.001$)	-0.408 ($p < 0.001$)	-0.320 ($p < 0.001$)
	ND_Z	-0.620 ($p < 0.001$)	-0.701 ($p < 0.001$)	-0.561 ($p < 0.001$)
	R_D	-0.500 ($p < 0.001$)	-0.433 ($p < 0.001$)	-0.282 ($p < 0.001$)
	$R_{I/O}$	-0.501 ($p < 0.001$)	-0.434 ($p < 0.001$)	-0.282 ($p < 0.001$)
	R_F	0.745 ($p < 0.001$)	0.693 ($p < 0.001$)	0.505 ($p < 0.001$)
	R_M	0.722 ($p < 0.001$)	0.736 ($p < 0.001$)	0.520 ($p < 0.001$)

node-layout case. In this examination, to ensure equal log data information in each metric combination, we excluded log data that were missing any of the metrics. The results obtained in the cases of 1D, 3D(malleable), and 3D(same) are summarized in Table 2.

In Table 2, strong or relatively strong positive correlation coefficients can be observed about R_F and R_M with P_{node}^{max} in every node-layout case because high utilization levels in memory devices or CPUs lead to high electric power levels. Among the three node-layout cases, 3D(malleable) shows very weak correlation coefficients for the metrics associated with file I/O (NQ , ND_Z , R_D , and $R_{I/O}$) with P_{node}^{max} compared with those in the other node-layout cases.

Another significant point is the relatively strong negative correlation coefficients observed in NQ , ND_Z , and R_D of the 3D(same) case. This is due to the I/O-intensive jobs observed at high frequency in this node-layout case. This kind of jobs explicitly specified the fixed compute node-layout due to its suitability for I/O operations. In such jobs, higher disk space was given in NQ , and a higher value in R_D was observed. Additionally, unlike the other node-layout cases, those jobs tended to use all the compute nodes in the Tofu Z-link.

4. Job classification using machine learning

Large-scale HPC systems such as Fugaku must accept inflexible restrictions in terms of electric power management in order to ensure

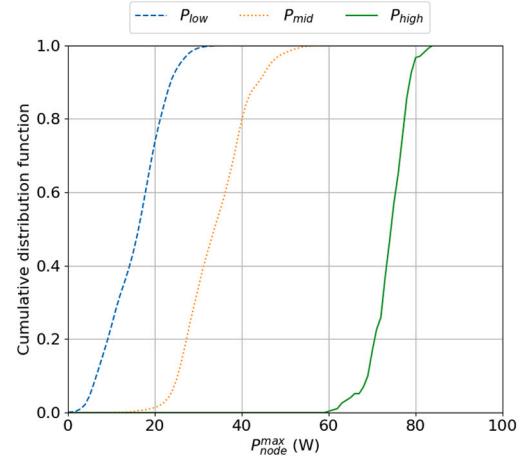


Fig. 6. Cumulative distribution function of the number of jobs in terms of the P_{node}^{max} for three categorized groups, P_{low} , P_{mid} , and P_{high} , by means of R_F and R_M .

stable system operations. Hence, prior to compute node allocation, HPC system operations require job predictions to determine whether or not each job will consume acceptable levels of electric power to prevent compute nodes from developing electric power consumption hot-spots [15–17]. Accordingly, we need the development of a prediction model based on job stats recorded in the past job executions that would identify, in advance, which jobs would consume high levels of electric power.

In the correlation examinations discussed in the previous section, we learned that there are relatively strong positive correlations between R_F and P_{node}^{max} and/or between R_M and P_{node}^{max} . However, we also found that there are relatively strong negative correlations between R_D and R_F in the 3D(same) job case. Based on those results, we applied our job stats log data to an examination of classification models using an ML approach.

4.1. Potential of classification learned from correlation examination

Based on the correlation examination of HPC jobs as shown in Figs. 4(b) and 5(b), we examined the P_{node}^{max} to determine whether or not there was room to classify the jobs into three groups. Since R_F and R_M are tightly related to P_{node}^{max} , we examined the following classification for P_{node}^{max} in terms of R_F and R_M :

- P_{high} : $R_F \geq 0.4 \wedge R_M \geq 0.4$
- P_{mid} : $(R_F \geq 0.2 \wedge R_M < 0.4) \vee (R_F < 0.4 \wedge R_M \geq 0.2)$
- P_{low} : $R_F < 0.2 \wedge R_M < 0.2$

P_{high} represents HPC jobs that consumed high levels of electric power due to high R_F and R_M values, while P_{low} is the group consuming low levels of electric power, in which I/O-intensive jobs were frequently observed. P_{mid} is an intermediate group between those two groups. In our analysis, the numbers of large jobs categorized in P_{low} , P_{mid} , and P_{high} were 111,721, 18,700, and 270, respectively. Most of HPC jobs were in P_{low} , while very small number of jobs were categorized in P_{high} . However, impact of such high electric power jobs was very big in the K computer operations to keep stable electric power supplies.

Fig. 6 shows the cumulative distribution function of the number of large jobs in the three groups in terms of P_{node}^{max} based on the above classifications.

Since, as can be seen in this figure, we can easily separate jobs in the P_{high} group from the P_{mid} and the P_{low} groups at 60 W, and there is sufficient room to predict high power-consuming HPC jobs with R_F , R_M , and other metrics.

4.2. Job classification using imbalanced datasets

Based on the cumulative distributed function shown in Fig. 6, we further examined electric power classifications for improved compute node assignments in terms of electric power using an ML approach, where the following four classification models provided in *scikit-learn*[18] Python library were examined:

- LogisticRegression (hereinafter, LOR)
- DecisionTreeClassifier (hereinafter, DTC)
- RandomForestClassifier (hereinafter, RFC)
- KNeighborsClassifier (hereinafter, KNC)

We used job stats log data recorded in the same period as those used in the correlation coefficient examinations. More specifically, we examined the four prediction models for the P_{node}^{max} using several metrics combinations such as R_M , R_F , and R_D with the help of the correlation coefficient analyses reported in Section 3. Training and testing datasets were prepared using `train_test_split()` from major group data consisting of job information with lower electric power levels (X) and minor group data consisting of job information with higher electric power levels (y), which were obtained by splitting the log data information at the given electric power threshold. The dataset preparation was performed by

```
X_train, X_test, y_train, y_test
= train_test_split(X, y, random_state=42,
                   test_size=0.25, stratify=y)
```

`X_train` and `y_train` are training datasets for jobs with lower and higher electric power levels, respectively, while `X_test` and `y_test` are testing datasets for jobs with lower and higher electric power levels, respectively. Note that since the `stratify` specification implies the same y distribution characteristics for both training and testing, we used the same datasets every time we performed examinations with the same distribution characteristics. It should be also noted that 75% of the data were assigned to the training dataset, and the remainder were assigned to the testing dataset.

The large jobs can be divided into two groups easily, one in which the P_{node}^{max} is greater or equal to 60 W and another in which the P_{node}^{max} is less than 60 W, as can be seen in Fig. 6. Note that only the 1D job case includes jobs that exceed 60 W in the P_{node}^{max} . We consider 30 W as an another threshold candidate in the P_{node}^{max} because P_{low} jobs are below or equal to 30 W in Fig. 6. It is noted that setting 30 W for the threshold in P_{node}^{max} leads to a mixed situation because each job case consists of jobs that exceed 30 W in the P_{node}^{max} . We analyzed the classification models in terms of P_{node}^{max} by setting 60 W and 30 W in the classification threshold. In both threshold cases, we compared the precision of the models in the three job cases (1D, 3D(malleable), and 3D(same)) and the job case consisting of the three job cases (hereinafter, All) to examine electric-power-aware compute node allocation in the supercomputer Fugaku which is the successor to the K computer. Table 3 shows the number of jobs assigned for the testing dataset.

Since the two job cases, “3D(malleable)” and “3D(same)”, did not have jobs above 50 W, we excluded those node layout cases from the examination when classifying jobs at the threshold levels of 60 W and 50 W, even though both node layout cases had jobs below the threshold levels. The dataset that was split at 40 W did not have jobs to be analyzed at the “3D(malleable)” layout due to the same reason.

In the four classification models, we utilized `GridSearchCV()` [19] to find the optimal parameter-set for each model, and then used that set for classification. The argument parameters are summarized in Table 4.

To improve analysis performance, the number of threads available at a PC server was set for the `num_thread` to improve analysis performance. The `GridSearchCV()` provides the best parameters from the set of parameter candidates given in the `param_grid`.

Table 3

The number of jobs assigned to the testing dataset.

Range in the P_{node}^{max} (W)	Node-layout			
	1D	3D(malleable)	3D(same)	All
<60	31,789	–	–	32,413
≥60	71	–	–	71
<50	31,695	–	–	32,319
≥50	165	–	–	165
<40	30,857	–	231	31,465
≥40	514	–	16	1,019
<30	28,553	364	213	29,129
≥30	3,307	14	34	3,355

Table 4

`GridSearchCV()` parameters in each classification model, where `num_thread` was given at the execution startup based on the available CPU resources.

Used model	Argument	Specified parameter
LOR, DTC,	scoring	<code>scoring='roc_auc'</code>
RFC, KNC	n_jobs	<code>n_jobs=num_thread</code>
LOR	estimator	<code>LogisticRegression()</code>
	param_grid	<code>[{'C': [0.001, 0.01, 0.1, 1, 10], 'random_state': [42], 'solver': ['lbfgs'], 'max_iter': [10000]}]</code>
DTC	estimator	<code>DecisionTreeClassifier()</code>
	param_grid	<code>[{'max_depth': [i for i in range(1, 10, 2)]}]</code>
RFC	estimator	<code>RandomForestClassifier()</code>
	param_grid	<code>[{'n_estimators': [i for i in range(100, 1001, 100)], 'max_depth': [i for i in range(15, 21, 1)], 'min_samples_split': [i for i in range(15, 21, 1)], 'criterion': ['gini', 'entropy'], 'random_state': [42]}]</code>
KNC	estimator	<code>KNeighborsClassifier()</code>
	param_grid	<code>[{'n_estimators': [i for i in range(1, 501, 10)], 'weights': ['uniform', 'distance'], 'metric': ['euclidean', 'manhattan']}]</code>

Due to the existence of imbalanced datasets in which the number of negatives (such as jobs above 60 W in P_{node}^{max}) outweighs the number of positives (such as jobs less than or equal to 60 W in P_{node}^{max}), we evaluated the precision-recall area under the curve (PR-AUC) using the methods described in previous studies on binary classification evaluations of imbalanced datasets [20–22]. Additionally, the precision, recall, and F1-score of the target negatives were used to examine the precision of the evaluated classification models. The MCC values of the used classification models were obtained with the training dataset because that value has been shown to be accurate even in an imbalanced dataset [23]. In this paper, the overall score was defined in the range from 0 to 1 when evaluating the predicted classification models using the three tolerant measures, F1-score, PR-AUC, and MCC, as follows:

$$\text{Overall score} = \sqrt[3]{\text{F1-score} \times \text{PR-AUC} \times (\text{MCC} + 1)/2}$$

Higher overall scores indicate higher levels of precision for the predicted classification models. In this evaluation, we used the overall score to measure the classification effectiveness and the essential metrics used in the classification.

Fig. 7 shows the evaluated values of the three classification models in the P_{node}^{max} at 60 W in each node-layout case.

Note that only the 1D case has jobs exceeding 60 W, while DTC, RFC, and KNC achieve near 1.0 in each measure using only R_F and

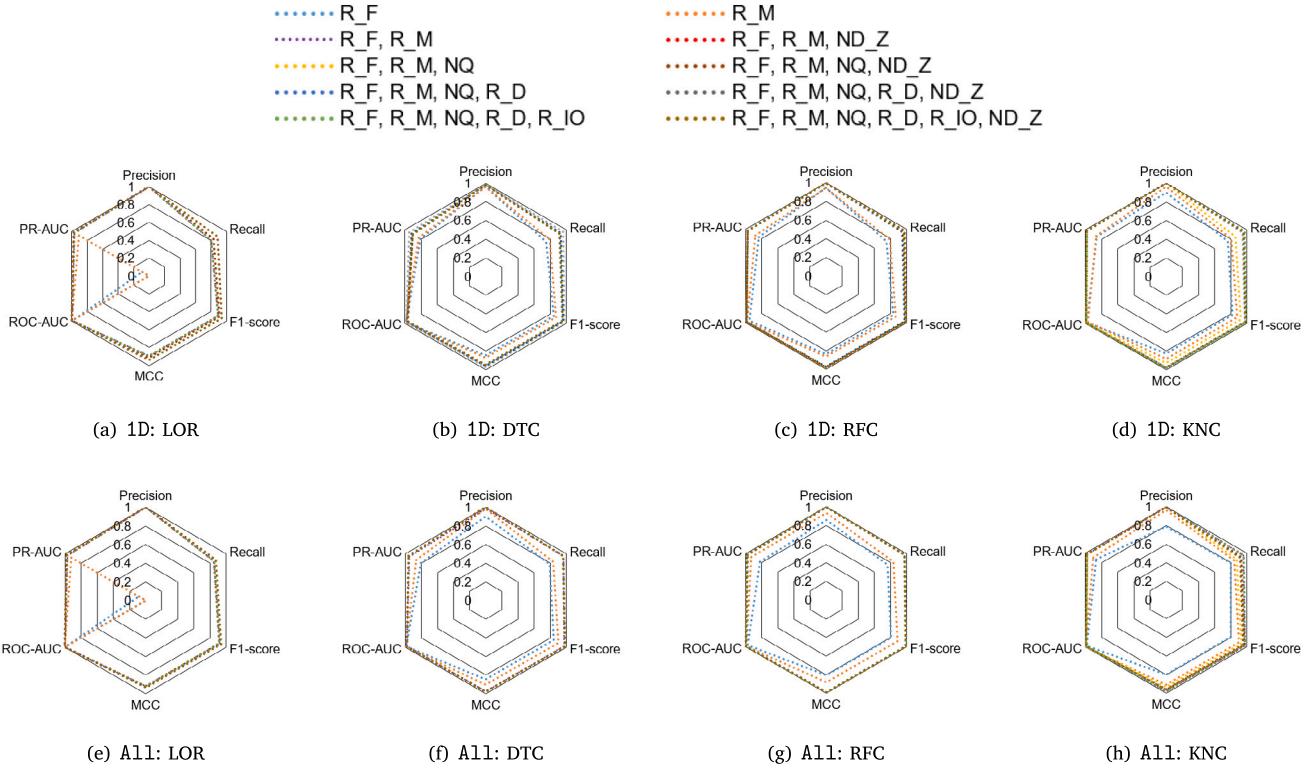


Fig. 7. Scores of predicted classification models with the threshold set at 60W in the P_{node}^{max} .

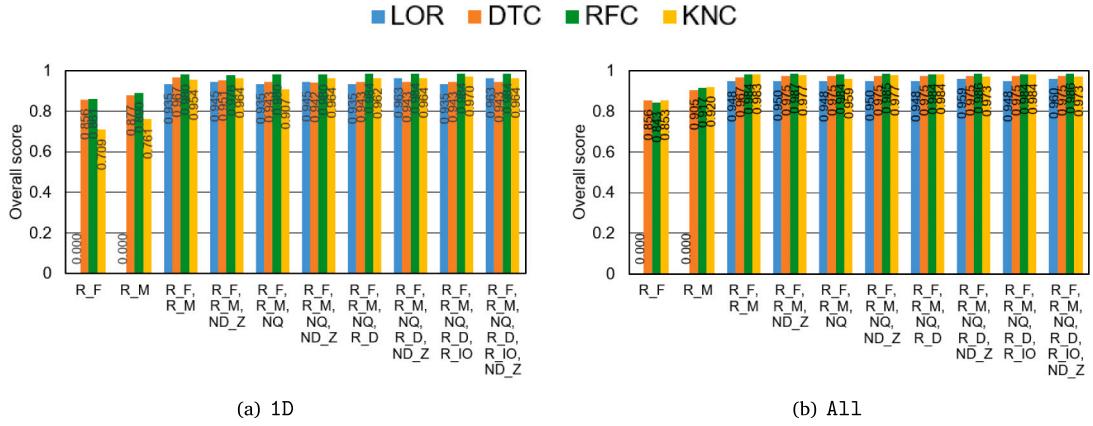


Fig. 8. Overall scores of the classification models with the threshold set at 60W in the P_{node}^{max} , where numbers in the graph represent each overall score.

R_M as shown in Figs. 7(b), 7(c), and 7(d), respectively. The MCC value of the LOR model is around 0.9 with R_F and R_M . However, since this model cannot achieve a score that is comparable to other models, as shown in Fig. 7(a), it cannot be used in the LOR model with fewer metrics from the evaluation.

Although the All case consists of the 3D(malleable) and 3D(same) jobs below 60W, almost the same situation is observed in the evaluated models, as shown in Figs. 7(e), 7(f), 7(g), and 7(h). This is because of the smaller number of jobs ($32,413 - 31,789 = 624$ from Table 3) in the testing datasets for both 3D(malleable) and 3D(same).

The overall scores of the classification models in terms of the metrics used are summarized in Fig. 8.

In the overall 1D scores shown in Fig. 8(a), the RFC model achieved the highest score (0.98) for the evaluated models with only R_F and R_M . The RFC model also slightly improved scores up to 0.984 by incorporating other metrics such as R_D or $R_{I/O}$ and showed similar

scores in the All case, as shown in Fig. 8(b). Furthermore, this model achieved the highest score (0.987) with R_F , R_M , and ND_Z , which indicates that it is the best choice in the classification at 60W in the P_{node}^{max} .

On the other hand, as we mentioned above, every job case has jobs exceeding 30W in the P_{node}^{max} . Fig. 9 shows the evaluated values of each classification model when the jobs are split at the 30W in the P_{node}^{max} .

Overall, the RFC model performed well compared with other models in relation to the two primary metrics related to electric power used in model predictions, R_F and R_M . However, the precision of the predicted models in the node-layouts of 3D(malleable) and 3D(same) became worse, especially in the 3D(malleable) case, when compared with the cases of 1D or All. Furthermore, due to the very weak correlation coefficients of metrics associated with file-I/O in the 3D(malleable) case reported in Table 2, it was difficult to improve the precision of the prediction models in comparison with the other node-layout cases, even though the RFC and the KNC did

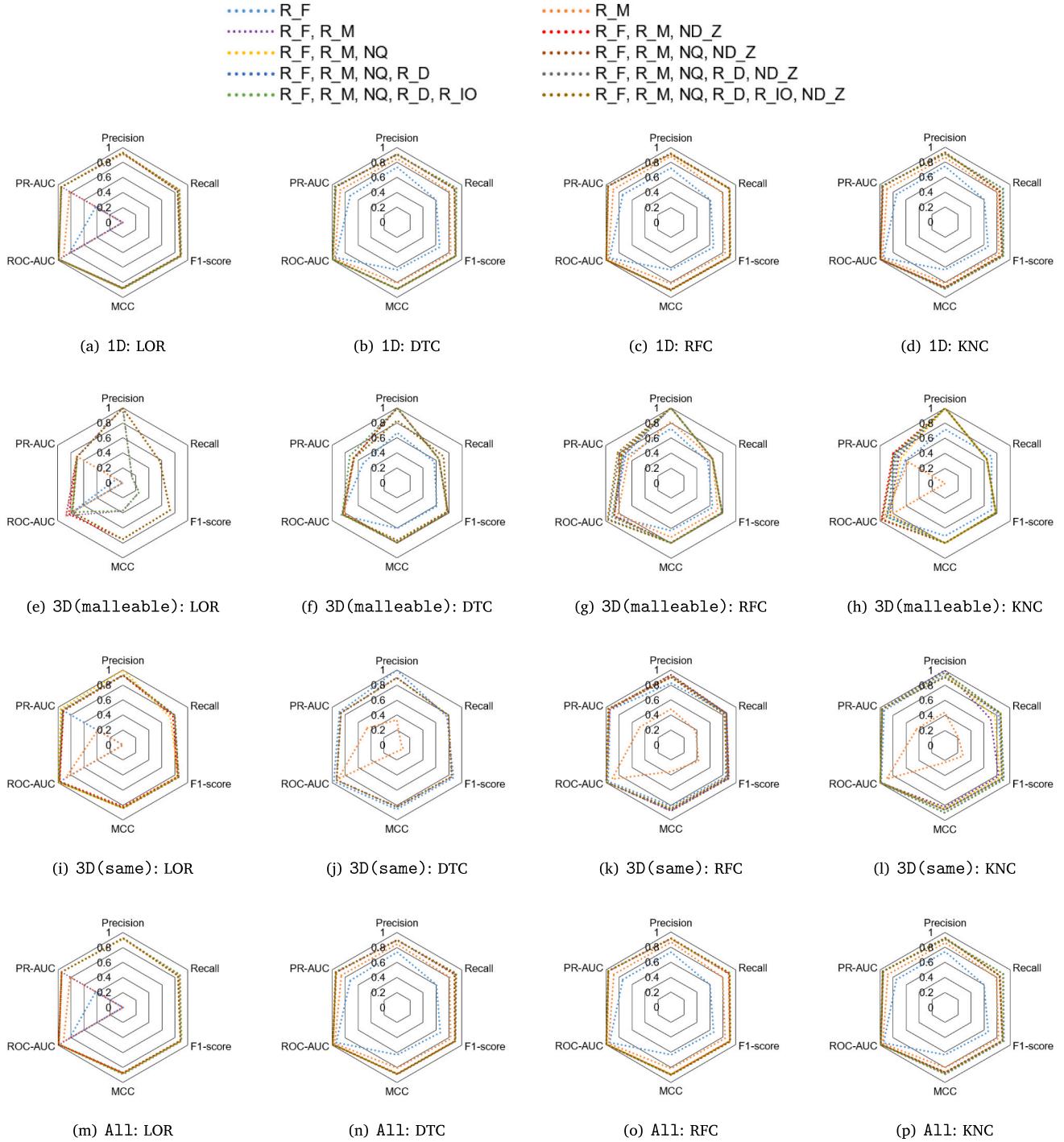


Fig. 9. Scores of predicted classification models with the threshold set at 30 W in the P_{node}^{max} .

improve scores to levels equal to or exceeding 0.8 in F1-score, MCC, and PR-AUC.

Based on the evaluated scores in F1-score, MCC, and PR-AUC, we summarized the overall scores of the predicted classification models in Fig. 10.

The RFC achieved the highest values of about 0.945, 0.832, and 0.944 in the cases of 1D, 3D(malleable), and All, respectively. In the 3D(same) case, the KNC achieved the highest score of 0.942 but the RFC achieved a comparable score of 0.929. In [24], the authors reported that the RFC model performed well when used to classify a large imbalanced dataset and we achieved a similar result with our imbalanced dataset.

Since most jobs are 1D, we can see almost the same scores in All. However, some degradation in the scores occurred if the number of jobs in 3D(malleable) and 3D(same) increased when considering the scores in the two cases mentioned above. This indicates that it might be better to perform job classification separately in each node-layout.

We also made classification at 40 W and 50 W, and the overall scores of them are shown in Figs. 11 and 12, respectively.

Compared with the results in Figs. 10 and 8, we have found degradation in the overall score except the 3D(same) jobs in the case of 40 W. As we have noticed about Figs. 3(e) and 3(f) in Section 3.1, the

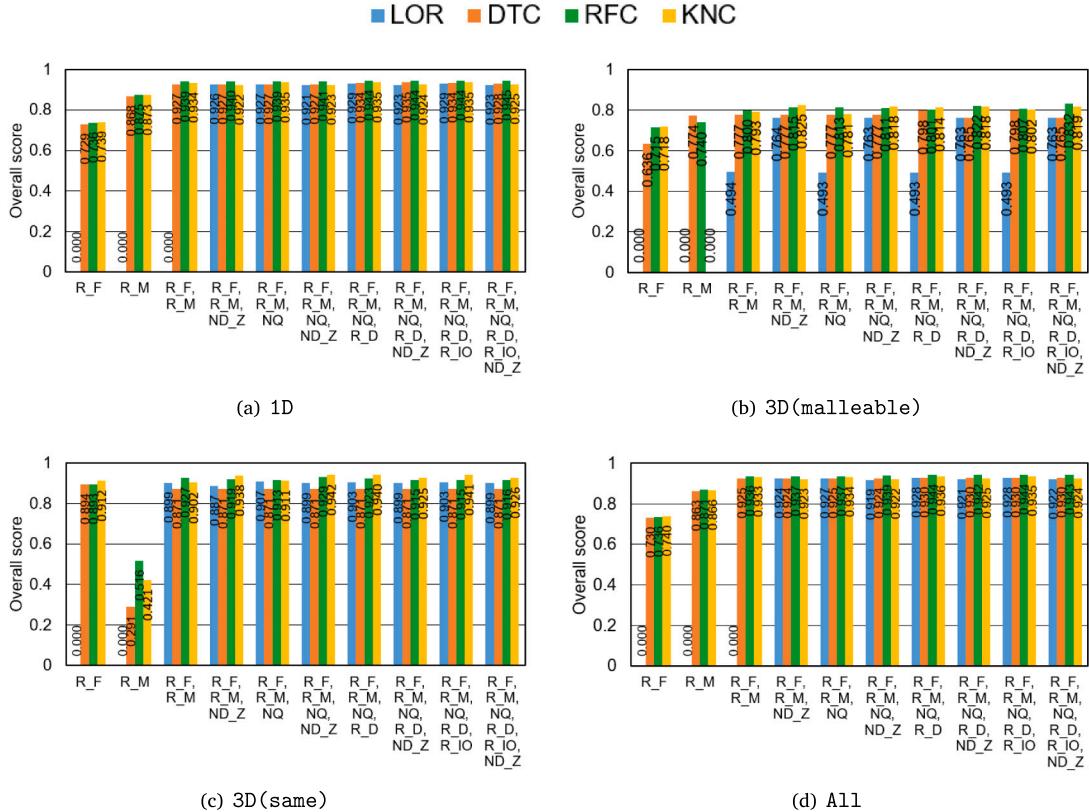


Fig. 10. Overall scores of the four classification models with the threshold set at 30W in the P_{node}^{max} , where the numbers in the graph represent each overall score.

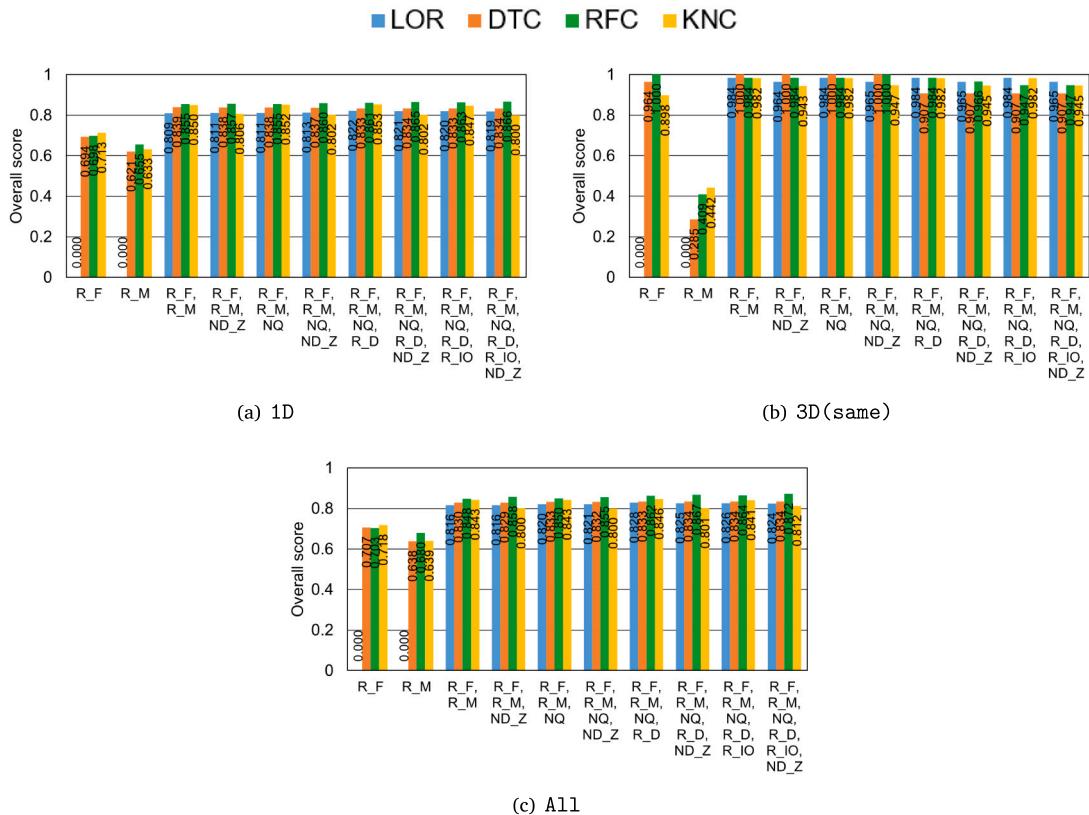


Fig. 11. Overall scores of the four classification models with the threshold set at 40W in the P_{node}^{max} , where the numbers in the graph represent each overall score.

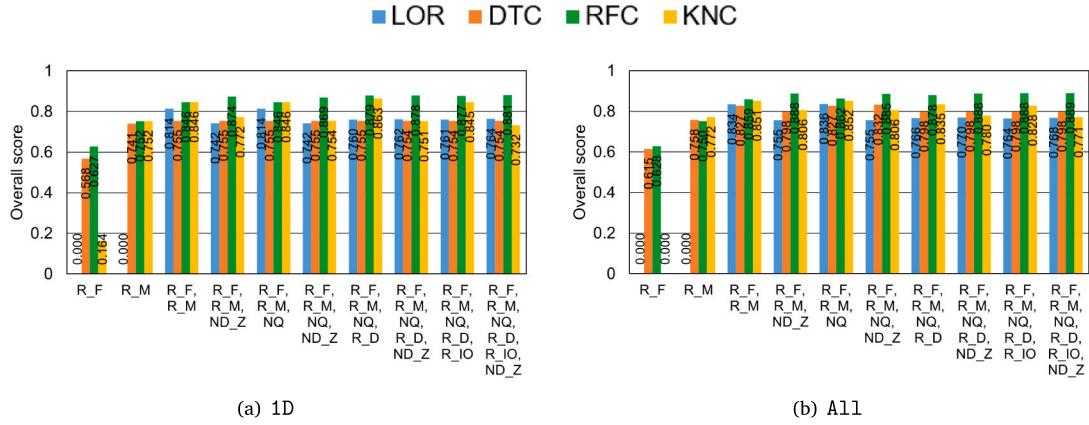


Fig. 12. Overall scores of the four classification models with the threshold set at 50W in the P_{node}^{max} , where the numbers in the graph represent each overall score.

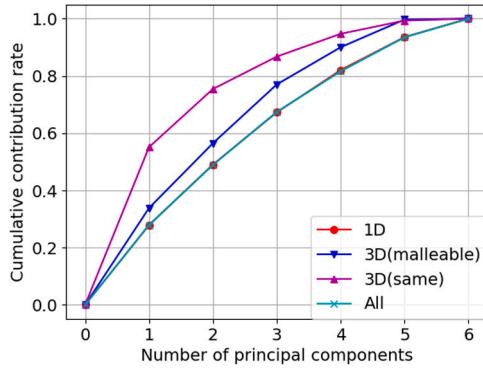


Fig. 13. Cumulative contribution rate of principal components in each node-layout case.

3D(same) jobs had two distinct groups: one group consisted of I/O-intensive jobs, while other had compute-intensive jobs. As a result, we have achieved higher scores only in the 3D(same) jobs.

4.3. Factor loading of metrics in the job classification

In order to examine the contribution rate of the principal components and the factor loading of the metrics used in the principal components, we performed principal component analysis using `sklearn.decomposition.PCA()` in *scikit-learn* after standardizing each metric. Fig. 13 shows cumulative contribution rate relative to the number of principal components ranging from the most influential principal component to the least influential one. This figure shows that we can achieve higher HPC job classification levels of precision in the cases of 3D(malleable) and 3D(same) than can be achieved for the All case, which means that when performing job classification, it is better to split the HPC jobs in each node-layout.

Fig. 14 indicates the factor loading of each metric relative to the two major principal components, PC1 and PC2, in each node-layout.

In Figs. 14(a), 14(b), and 14(d), it can be seen that two file I/O-related metrics, R_D and $R_{I/O}$, are related to the first principal component, PC1, at around 0.75 relative to the second principal component, PC2, while Fig. 14(c) shows that the two metrics are related to both of PC1 and PC2 around 0.5.

The computing related metrics, R_F and R_M , showed similar behavior expressing a higher contribution of R_F to PC2, as shown in Figs. 14(a) and 14(d), while Figs. 14(b) and 14(c) show different behavior in each case. When compared with R_F , it can be seen that R_M has a smaller contribution to both PC1 and PC2 in the cases of 1D and All, as shown in Figs. 14(a) and 14(d). Since most HPC jobs were

executed in the 1D case, similar behaviors for all six metrics used were observed in the All case.

The behavior of two file I/O-related metrics, NQ and ND_Z , are noteworthy because they showed different properties in each node-layout. In the cases of 1D, 3D(malleable), All shown in Figs. 14(a), 14(b), and 14(d), the contribution rates of NQ and ND_Z to PC2 were larger than those to PC1, while their contribution rates to PC1 increased in the case of 3D(same), as shown in Fig. 14(c). This is because of the I/O-intensive jobs that specified higher values in NQ and which, to the maximum extent possible, had compute nodes in the same Tofu Z-link of the 3D(same) node-layout.

From the above analysis in addition to correlation coefficients reported in Table 2, we can state the following:

- Computing-related metrics, R_F and R_M , are major classification metrics.
- File I/O-related metrics, R_D and $R_{I/O}$, are counterparts to the computing-related metrics.
- Other file I/O-related metrics, NQ and ND_Z , show distinct behaviors in the 3D(same) node-layout case due to the frequency of I/O-intensive jobs in that case.

Taken together, these results indicate that it is preferable to classify HPC jobs in each compute node-layout rather than to perform classification without separating the jobs into each node-layout.

5. Related work

To keep the HPC system operations within a limited power budget, power management is currently one of the most important issues in HPC system operations. Because of this, power capping has been proposed as a way to control electric power consumption in order to facilitate stable HPC system operations in numerous studies [15,16,25]. In the K computer operation, we also worked diligently to manage system operations within our power budget. Although we did not have electric power measurement devices, full-scale power predictions were performed for jobs using our prediction models [4] in a few days of special full-scale job execution period held once per month according to the recorded metrics with their smaller jobs. In Fugaku, which is the successor to the K computer, we have implemented power control and power measurement functions for its CPUs, memory devices, and compute node peripheral equipment. Along with the adopted enhancements, we have also implemented power measurement and control features in the operation software [2]. The metrics obtained the measurement functions are also provided for users in a summary data of the executed job to tune their applications, for instance.

Another aspect of controlling electric power consumption is I/O operation management. Due to the negative correlation between I/O

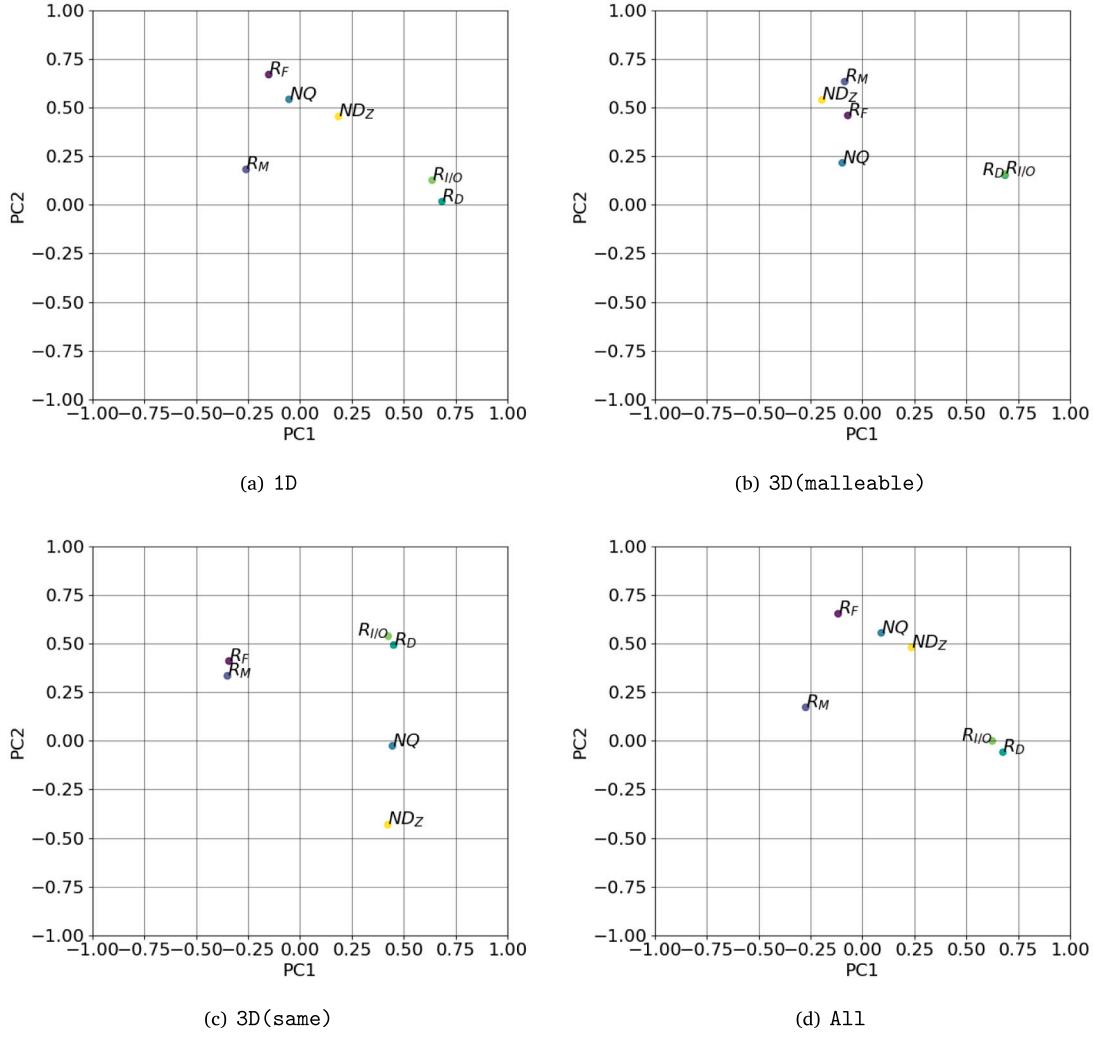


Fig. 14. Factor loading of the metrics used relative to PC1 and PC2.

operations and electric power, another solution for power management is monitoring I/O activities in HPC jobs and characterizing those jobs based on their I/O intensity. Numerous studies have focused on log analysis as a way to reveal the I/O activities of HPC jobs [26–28]. For example, Liu et al. [26] developed an application characterization analysis framework that included I/O activities and successfully incorporated consideration for I/O activities in job scheduling. Separately, Wang et al. [27] proposed a multilateral framework to investigate I/O activities and the root causes of I/O problems, from a comprehensive analysis of large-scale I/O log data, in which Darshan [29,30] was introduced to collect I/O performance metrics. Lockwood et al. [28] proposed a holistic analysis framework for use in investigating whole HPC systems using numerous analysis and monitoring tools such as Darshan. However, the abovementioned studies did not discuss electric power issues, even though there have been studies that discussed power management from the viewpoint of I/O activities [31,32]. For example, Manousakis et al. [31] proposed a framework that can be used to minimize electric power in I/O-intensive jobs, and Lee et al. [32] addressed providing a framework that can be used to supply more electric power to compute-intensive jobs using I/O aware power shifting algorithms. Similar to these research efforts, we focused on using several I/O activity metrics to classify HPC jobs in terms of electric power and have observed some specific features of I/O-intensive HPC jobs, such as node-layout specifications or required disk space allocations, through correlation studies of those metrics. Through the Fugaku operation, we have faced I/O contentions among jobs. Fugaku has different system

configuration where a subset of compute node also acts as I/O node. Although I/O node tasks are assigned for extra CPU cores on such node, there are risks that we may have a sort of interference among jobs sharing the same I/O node, especially when such I/O node fail to operate due to heavy I/O operations. Since our proposed ML classification also shows inverse proportional among I/O activities and electric power, such methodology can be one of the solutions to prevent such problems in compute node allocation.

Here, it should be noted that ML has been actively and widely introduced for huge scale log analysis in HPC systems [33–35]. For example, He et al. [33] studied failure prediction by parsing logs, followed by performing ML through system log analysis. To accomplish this, they evaluated three supervised and two unsupervised methods in their ML analysis. Separately, Das et al. [34] proposed a framework for predicting failure nodes in HPC systems using their time-based phrase (TBP) prediction scheme. The TBP framework was built with an ML technique called Latent Dirichlet Allocation (LDA) [36] with a Topics over Time (TOT) [37] enhancement to take the time correlation aspect into consideration. The effectiveness of that framework was demonstrated in their log analysis of a Cray HPC system. Kim et al. [35] also studied I/O performance prediction using an ML approach through log analysis study in which their framework analyzed preferable metrics for ML through correlation analyses conducted before the ML phase. Their regression-based ML approach achieved higher prediction accuracy in I/O-intensive applications from a number of different system logs and the automatic selection of the best regression algorithm in each

prediction. However, in our work, which is focused on electric power consumption using ML classification models with metrics recorded in job stats log data, we found that correlation studies can produce relatively strong positive associations between the utilization ratios of CPUs and memory devices and electric power levels per node. Additionally, correlation studies between disk utilization ratios and electric power levels per node showed strong negative associations in a fixed 3D node-layout case because I/O-intensive jobs were frequently executed in that node-layout. Based on these results, we applied an ML-based approach to classifying jobs in terms of electric power levels per node. The major factors used to predict electric power levels are the utilization ratios of CPUs and memory devices, but additional metrics such as the disk utilization ratio have also improved the precision of the predicted classification models. Finding preferable metrics through correlation analysis and automatic selection of prediction models applied in [35] is a process worth being introduced into our ML scheme.

One of the remaining that will be discussed in this paper is the method used to evaluate predicted classification models for imbalanced datasets in ML studies. In this context, He and Garcia reported improved precision-recall curve efficiency, even in imbalanced datasets in their paper [38] in which they clearly discussed the problems that occurred in those datasets along with some state-of-the-art solutions for those problems. Separately, Jauk et al. produced a survey paper in which they discussed state-of-the-art studies about failure prediction in many leading HPC systems [39]. Although they discussed effective evaluations among different classification models in reference to imbalanced datasets, they only touched briefly on studies that paid close attention to the problem, despite the large number of research papers surveyed. Additionally, they limited their remarks to stating that no numerical or statistical analysis studies have yet been conducted to address this problem. Contrastingly, in our research, we proposed a combined evaluation score by introducing F1-score, PR-AUC, and MCC for use in comparisons among different classification models using imbalanced datasets. Although our proposal works acceptably in our ML study using K computer log data, it will be necessary to propose an improved or optimum method of conducting ML-based evaluations of imbalanced datasets in the future.

6. Conclusions

Herein we reported the analysis of two years of job stats log data collected from the K computer to classify HPC jobs in terms of electric power levels per compute node. This study is part of efforts to facilitate future advanced power-aware compute node allocation in accordance with effective job scheduling in Fugaku, which is the successor of the K computer. Our results showed that we need the ability to perform more suitable compute node allocations in order to manage power consumption within controllable conditions and to ensure stable operations in such situations.

Based on the above, we studied classification methodology in terms of electric power consumed by compute nodes through correlation coefficient analysis and ML-based classification using associated metrics in job stat log data and determined that there were relatively strong correlations between electric power and CPU and memory bandwidth utilization ratios. Additionally, there were relatively strong negative correlations between metrics related to file I/O and electric power in the node-layout in which I/O-intensive jobs were most frequently executed. Based on this correlation study, we examined four classification models provided from *scikit-learn* and discovered some specific results related to allocated compute node-layouts. For example, we established the overall score ranging from 0 to 1 to evaluated predicted classification models using F1-score, PR-AUC, and MCC, which are tolerant measures even in imbalanced datasets. Through that examination, we found that the RFC model achieved the highest overall score (0.98) using only CPU and memory bandwidth utilization ratios when classifying at 60W in the maximum electric power per node.

Furthermore, in the classification at 30W, that model achieved high scores of up to 0.945, 0.832, and 0.944 among the evaluated models in the three node-layout cases of 1D, 3D with layout changes, and all node-layout cases, respectively. Although the KNC model achieved the highest score (0.944) in the 3D node-layout without layout changes, the RFC model achieved a comparable score. Classifying the jobs at 50W or 40W did not show good scores except the compute node layout of 3D with layout changes because of difficulty in classifying jobs at those threshold levels among mixed jobs about electric power levels per compute node. Overall, we determined that the RFC was the best model for use in our HPC job classifications in terms of electric power levels per compute node if we adopted appropriate threshold level such as 60W or 30W.

Through an analysis of the contribution rate of principal components and the factor loading of the metrics used in the principal components, we observed that the computing-related metrics, such as utilization ratios in CPU FLOPS and memory devices, are counterparts of the two file I/O-related metrics, which are the ratio of disk utilization relative to allocated disk space and the ratio of used disk space relative to achievable maximum I/O size amount. The other file I/O-related metrics, the size of allocated disk space per node and the number of nodes in the Tofu Z-link direction, showed features that were similar to the other file I/O-related metrics in the 3D node-layout without layout changes. This is due to the fact that I/O-intensive jobs specified higher values in the assigned disk space per node and in the occupancy in compute nodes on the same Tofu Z-link. Since every metric showed different properties in each node-layout case, not only from this analysis but also from our correlation study, we concluded that when it comes to facilitating power-aware HPC system operations, performing job classification in each node-layout case is preferable to building a job classification model without any attention to node-layouts.

Our future work will focus on automatic selection among evaluated classification models based on target metrics like those explored in [35] with periodical updates issued as we introduce our approach in Fugaku. Since Fugaku supports an electric power measurement function inside its CPU nodes, there is expected to be additional room for improving the precision of the classification models used. Since we need to supply stable electric power in Fugaku operation, optimization in compute node allocation is one of the important aspects in future advanced operation. We expect that classification of HPC jobs in terms of electric power at the given threshold level using collected log data prevents hot-spots in electric power in system racks containing a large number of compute nodes if such classification is introduced in compute node allocation during job scheduling with the job script classification reported in [11]. Within this context, another future work will be proposing tolerant scoring in comparison among various classification models using imbalanced datasets. However, to accomplish this, it will be necessary to study other researches in leading HPC systems in order to determine optimal scoring methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research used system logs and job stats data collected from the K computer operated by the RIKEN Center for Computational Science.

References

- [1] The supercomputer Fugaku [cited 2022-04-26]. [link]. URL <https://www.r-ccs.riken.jp/en/fugaku/>.
- [2] Akimoto H, Okamoto T, Kagami T, Seki K, Sakai K, Imade H, Shinohara M, Sumimoto S. File system and power management enhanced for supercomputer Fugaku. *Fujitsu Tech Rev* 2020;(3). [cited 2022-04-26]. URL <https://www.fujitsu.com/global/about/resources/publications/technicalreview/2020-03/article05.html>.
- [3] Uno A, Sueyasu F, Sekizawa R. Operations management software of supercomputer Fugaku. *Fujitsu Tech Rev* 2020;(3). [cited 2022-04-26]. URL <https://www.fujitsu.com/global/about/resources/publications/technicalreview/2020-03/article10.html>.
- [4] Uno A, Hida H, Inoue F, Ikeda N, Tsukamoto T, Sueyasu F, Matsushita S, Shoji F. Operation of the k computer focusing on system power consumption. *IPSJ Trans Adv Comput Syst* 2015;8(4). 13–25. (in Japanese).
- [5] Tsujita Y, Uno A, Sekizawa R, Yamamoto K, Sueyasu F. Job classification through long-term log analysis towards power-aware HPC system operation. In: 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE; 2021, p. 26–34. <http://dx.doi.org/10.1109/PDP52278.2021.00014>.
- [6] Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta (BBA) - Protein Struct* 1975;405(2). 442–451. [http://dx.doi.org/10.1016/0005-2795\(75\)90109-9](http://dx.doi.org/10.1016/0005-2795(75)90109-9).
- [7] Ajima Y, Inoue T, Hiramoto S, Takagi Y, Shimizu T. The Tofu interconnect. *IEEE Micro* 2012;32(1). 21–31. <http://dx.doi.org/10.1109/MM.2011.98>.
- [8] Sakai K, Sumimoto S, Kurokawa M. High-performance and highly reliable file system for the K computer. *Fujitsu Sci Tech J* 2012;48(3). 302–309.
- [9] Hirai K, Iguchi Y, Uno A, Kurokawa M. Operations management software for the K computer. *Fujitsu Sci Tech J* 2012;48(3). 310–316.
- [10] Sumimoto S. An overview of Fujitsu's Lustre based file system. In: Lustre User Group 2011. 2011.
- [11] Yamamoto K, Tsujita Y, Uno A. Classifying jobs and predicting applications in HPC systems. In: High Performance Computing - 33rd International Conference, ISC High Performance 2018, Frankfurt, Germany, June 24–28, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10876, Springer; 2018, p. 81–99.
- [12] Redash [cited 2022-04-26]. [link]. URL <https://redash.io/>.
- [13] SciPy [cited 2022-04-26]. [link]. URL <https://scipy.org/>.
- [14] You H, Zhang H. Comprehensive workload analysis and modeling of a petascale supercomputer. In: Job Scheduling Strategies for Parallel Processing, 16th International Workshop, JSSPP 2012, Shanghai, China, May 25, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7698, Springer; 2012, p. 253–271. http://dx.doi.org/10.1007/978-3-642-35867-8_14.
- [15] Dutot P, Georgiou Y, Glessner D, Lefevre L, Poquet M, Rais I. Towards energy budget control in HPC. In: 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE Press; 2017, p. 381–390. <http://dx.doi.org/10.1109/CCGRID.2017.16>.
- [16] Rajagopal D, Tafani D, Georgiou Y, Glessner D, Ott M. A novel approach for job scheduling optimizations under power cap for ARM and Intel HPC systems. In: 24th IEEE International Conference on High Performance Computing (HiPC). IEEE Computer Society; 2017, p. 142–151. <http://dx.doi.org/10.1109/HiPC.2017.00025>.
- [17] Saillant T, Weill J-C, Mougeot M. Predicting job power consumption based on RJMS submission data in HPC systems. In: High Performance Computing - 35th International Conference, ISC High Performance 2020, Frankfurt/Main, Germany, June 22–25, 2020, proceedings. Lecture Notes in Computer Science, vol. 12151, Springer; 2020, p. 63–82. http://dx.doi.org/10.1007/978-3-030-50743-5_4.
- [18] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É. Scikit-learn: Machine learning in Python. *J Mach Learn Res* 2011;12(85). 2825–2830. [cited 2022-04-26]. URL <https://jmlr.org/papers/v12/pedregosa11a.html>.
- [19] Scikit-learn [cited 2022-04-26]. [link]. URL https://scikit-learn.org/stable/modules/grid_search.html.
- [20] Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* 2015;10(3). 1–21. <http://dx.doi.org/10.1371/journal.pone.0118432>.
- [21] Salfner F, Lenk M, Malek M. A survey of online failure prediction methods. *ACM Comput Surv* 2010;42(3). 10:1–10:42. <http://dx.doi.org/10.1145/1670679.1670680>.
- [22] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning. 2006, p. 233–240. <http://dx.doi.org/10.1145/1143844.1143874>.
- [23] Boughorbel S, Jarray F, El-Anbari M. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLoS ONE* 2017;12(6). 1–17. <http://dx.doi.org/10.1371/journal.pone.0177678>.
- [24] Sırbu A, Babaoglu O. Towards operator-less data centers through data-driven, predictive, proactive autonomies. *Cluster Comput* 2016;19. 865–878. <http://dx.doi.org/10.1007/s10586-016-0564-y>.
- [25] Ellsworth D, Patki T, Schulz M, Rountree B, Malony A. A unified platform for exploring power management strategies. In: Proceedings of the 4th International Workshop on Energy Efficient Supercomputing. E2SC '16, IEEE Press; 2016, p. 24–30. <http://dx.doi.org/10.5555/3018076.3018080>.
- [26] Liu Y, Gunasekaran R, Ma X, Vazhkudai SS. Server-side log data analytics for I/O workload characterization and coordination on large shared storage systems. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '16, IEEE; 2016, p. 819–829. <http://dx.doi.org/10.1109/SC.2016.69>.
- [27] Wang T, Snyderly S, Lockwood GK, Carns P, Wright NJ, Byna S. IOMiner: Large-scale analytics framework for gaining knowledge from I/O logs. In: 2018 IEEE International Conference on Cluster Computing (CLUSTER). IEEE; 2018, p. 466–476. <http://dx.doi.org/10.1109/CLUSTER.2018.00062>.
- [28] Lockwood GK, Wright NJ, Snyder S, Carns P, Brown G, Harms K. TOKIO on ClusterStor: Connecting standard tools to enable holistic I/O performance analysis. In: 2018 Cray User Group Meeting (CUG). 2018.
- [29] DARSHAN [cited 2022-04-26]. [link]. URL <https://www.mcs.anl.gov/research/projects/darshan/>.
- [30] Carns P, Harms K, Allcock W, Bacon C, Lang S, Latham R, Ross R. Understanding and improving computational science storage access through continuous characterization. *ACM Trans Storage* 2011;7(3). <http://dx.doi.org/10.1145/2027066.2027068>.
- [31] Manousakis I, Marazakis M, Bilas A. FDIO: A feedback driven controller for minimizing energy in I/O-intensive applications. In: Proceedings of the 5th USENIX Workshop on Hot Topics in Storage and File Systems. HotStorage'13, USENIX Association; 2013, [cited 2022-04-26]. URL <https://www.usenix.org/conference/hotstorage13/workshop-program/presentation/manousakis>.
- [32] Lee S, Lowenthal DK, De Supinski BR, Islam T, Mohror K, Rountree B, Schulz M. I/O aware power shifting. In: 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS). 2016, p. 740–749. <http://dx.doi.org/10.1109/IPDPS.2016.15>.
- [33] He S, Zhu J, He P, Lyu MR. Experience report: System log analysis for anomaly detection. In: 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). 2016, p. 207–218. <http://dx.doi.org/10.1109/ISSRE.2016.21>.
- [34] Das A, Mueller F, Hargrove P, Roman E, Baden S. Doomsday: Predicting which node will fail when on supercomputers. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. SC '18, IEEE Press; 2018, p. 9:1–9:14. <http://dx.doi.org/10.1109/SC.2018.00012>.
- [35] Kim S, Sim A, Wu K, Byna S, Son Y, Eom H. Towards HPC I/O performance prediction through large-scale log analysis. In: Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing. HPDC '20, ACM; 2020, p. 77–88. <http://dx.doi.org/10.1145/3369583.3392678>.
- [36] Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. *J Mach Learn Res* 2003;3. 993–1022.
- [37] Wang X, McCallum A. Topics over time: A non-Markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '06, ACM; 2006, p. 424–433. <http://dx.doi.org/10.1145/1150402.1150450>.
- [38] He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 2009;21(9). 1263–1284. <http://dx.doi.org/10.1109/TKDE.2008.239>.
- [39] Jauk D, Yang D, Schulz M. Predicting faults in high performance computing systems: An in-depth survey of the state-of-the-practice. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '19, ACM; 2019, <http://dx.doi.org/10.1145/3295500.3356185>.