Study on data center server management system for using server power estimation technology

Katsumi Fujita
NTT Network Innovation Center
NTT Corporation
Tokyo, Japan
katsumi.fujita.ze@hco.ntt.co.jp

Eriko Iwasa
NTT Network Innovation Center
NTT Corporation
Tokyo, Japan
eriko.iwasa.xt@hco.ntt.co.jp

Masashi Kaneko
NTT Network Innovation Center
NTT Corporation
Tokyo, Japan
masashi.kaneko.dr@hco.ntt.co.jp

***Abstract*— The increasing power consumption in servers is a problem in data centers. One approach to address this is to optimize physical resource allocation for virtualized functions. For this approach, power consumption needs to be estimated, and many papers have proposed models to estimate server power consumption. However, applying such estimation techniques to data center servers presents an operational challenge. Specifically, it is difficult to reduce the cost of model building and data collection while maintaining estimation accuracy. To address this problem, we propose a management system with a modeling function and a data selection function. The modeling function builds accurate models in accordance with application requirements, and the data selection function selects data related to server power consumption and then instructs data collection to each processing server. Our main effort in this paper was to address the monitoring function issue to reduce server power consumption during data collection. We analyze the relationship between server performance data and server power consumption and reduce the type of data. Experimental results showed that our selected data are sufficient to construct a server power modeling. Furthermore, by limiting the amount of data collected, we were able to reduce server power consumption by about 3 W and monitoring overhead by about 30%.**

Keywords*— *server, power consumption, statistical analysis, machine learning

I. Introduction

With the spread of big data and cloud computing, the energy consumption of data centers has become a problem [1]. Some studies have proposed power-aware scheduling algorithms to minimize the power consumption for servers in a datacenter [2][3]. Their target environment is a service infrastructure that executes virtualized functions on servers. Their techniques reduce the number of servers used by consolidating virtual machines or containers. Thus, their power-aware scheduling algorithm requires power consumption information in two situations: control judgment by the power consumption estimation of the server, and the power consumption of each service component. Our target is the former situation. There are two ways to monitor server power consumption: a physical method using a wattmeter and an estimation method based on the power model. A physical method (like a wattmeter) incurs operating costs to put the instrument on servers, whereas the estimation cost is lower than the physical method cost because the power model is constructed using server performance data that is collected from a software monitoring tool. For the server power modeling, many researchers have proposed models by using machine learning [4].

Our motivation is to apply the server power consumption estimation technology to a data center and use an estimation value as a control parameter. The data center we envision runs multiple different applications. Each server processes a different load because virtualization technology enables different applications to be installed on the same server.

There are two possible directions for data center managers to use server power estimation techniques. (1) The first direction is to build individual models for all servers in the data center and estimate power, and (2) the second direction is to build one model for all servers in the data center and estimate power. However, they have some problems. (1) The first direction incurs costs for modeling because a model needs to be built for each server. In addition, the accuracy of the model drops when the combination of applications changes, so the model needs to be rebuilt. Additionally, centralized management is difficult. When an estimation model is constructed for each server, the input data (explanatory variables) required for each model may differ. In such cases, the settings need to be changed to select and collect data for each server. (2) The second direction may reduce the accuracy of the model. In our envisioned datacenter, it is more precise to build a power consumption model for each load feature of servers. For example, Weiwei et al. [5] evaluated the accuracy of server power estimation by distinguishing four types of load: CPU load, memory load, I/O load, and mixed load. If a single model is created for all servers in the data center, the estimation accuracy of servers with biased load characteristics (e.g., I/O intensive workload) is reduced. In addition, collecting various data to create a generic model for different load characteristics increases the power consumption of the server during collection.

From the perspective of management and operating costs, we consider that the second direction is better. However, there are two points that need further improvement. The first point is the estimation accuracy. Compared with building a model with a fixed combination pattern of servers and applications, the estimation accuracy of the model is reduced when there are many variations in the combination of applications running on the server and when the situation changes dynamically. Some users who use server power consumption as a metric for scaling control or resource utilization calculations require highly accurate estimates. The second point is the power consumption for data collection. Ziyu et al. [6] collected a total of 158 data. The reason they adopted 158 data for explanatory variables was to enhance the versatility. They explained that their method can predict the power of the server on other applications by changing the relevant training dataset. However, when many types of data are collected for various load characteristics, the power consumption for data collection increases.

For each point, we propose the following solutions: For the first point, we propose a management system that basically uses a single model and builds individual server power models only for servers equipped with applications that require high estimation accuracy, thereby reducing the number of models to be built. For the second point, we identify the data commonly needed to estimate the power consumption of servers in the data center, and reduce the power consumption of data collection by reducing unnecessary data collection.

Our main contributions are as follows:

1. We propose a management system for using server power estimation technology in data center. This management system basically uses a single model and builds individual server power models only for servers equipped with applications that require high estimation accuracy. This management system also has a function for reducing server power consumption during data collection. For that, we proposed an approach that reduces the types of collecting data.
2. We conducted experiments and show that our approach reduced server power consumption by about 3W and monitoring overhead by about 30%.

The main focus of this paper is on efforts to reduce server power consumption during data collection and the experimental results. The rest of the paper is organized as follows. Chapter II describes the our proposed the management system with a new function and the approach that reduces types of collecting data. Chapter III describes experimental methods to verify the power-saving effects of the data reduction approach. Results and discussion are in Chapter IV. In Chapter V, the related works are discussed. Chapter VI presents the conclusion on this work.

II. Method

In this chapter, we first describe a management system with a new function for using power estimation technology in data centers (in Section A). Then we describe specific techniques for reducing power consumption during data collection in Sections B and C.

A. Management System

A data center management system generally collects data such as CPU utilization from servers that perform computational processing and stores the data in a data store. On the basis of the collected data, an estimation model is constructed, and the estimated results are used as a control decision index. Our proposed monitoring system (Fig. 1) adds the data selection function (Data Selector in Fig. 1) to the conventional system. The data selector selects data relevant to server power consumption. Then the data selector instructs data collection to each processing server. The detailed method is described in the next section.

The modeling function (Model Constructor in Fig. 1) builds one estimation model for all servers in the data center and builds individual estimation models for some servers deploying an application that requires high estimation accuracy. To build a unique model for an application that requires high estimation accuracy, the model constructor uses specific data that are only needed to build the model for that application. Thereby, the system can meet the needs of applications that require high estimation accuracy. Also, the system can reduce modeling costs compared with the method that builds individual models for all servers.

The following sections explain how the data selection function determines the data to be acquired. The data to be selected will be used as explanatory variables (input data) for model construction and will be referred to as data or explanatory variables in the following text.

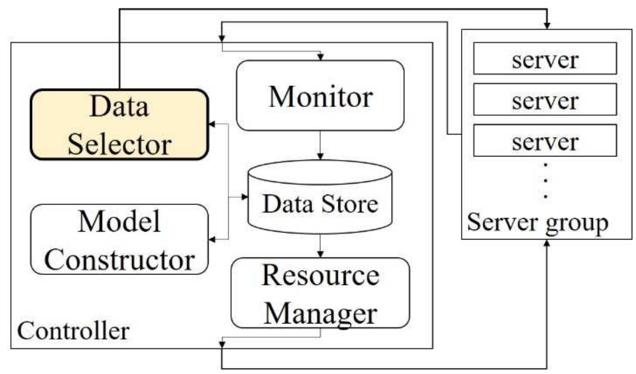

Fig. 1. Overview of the data center management system.

B. Explanatory variable selection

The criteria for choosing explanatory variables depend on previous studies. There are two approaches for selecting variables: qualitative and quantitative. The qualitative approach selects variables that are expected to be related to power consumption. This approach requires researchers to have expert knowledge of server subsystems. The quantitative approach, on the other hand, uses methods such as statistical analysis and machine learning to select explanatory variables for modeling.

Examples of the quantitative approaches are as follows. Yunfang et al. [7] calculated the Pearson correlation coefficient between all collected variables and the power consumption. They selected variables with the absolute value of the correlation coefficient greater than 0.5. Guang et al. [8] selected variables using the stepwise analysis. Bergamaschi and Rigo [9] used the correlation-based feature selection (CFS) algorithm proposed by Hall et al. [10]. Zhou et al. [11] used the principal component analysis (PCA) to reduce dimensionality.

The purpose of reducing dimensionality and explanatory variables in previous works was to improve estimation accuracy. On the other hand, we first need to perform an analysis to understand the relationship between server power consumption and performance data and select performance data that provide good estimation accuracy. The methods and its problem are as follows:

- When examining the correlation of each variable in multivariate analysis, the results of the correlation coefficient may show a spurious correlation. It is more accurate to analyze using the partial correlation coefficient, which is a pure correlation coefficient with the interaction between variables removed.
- The stepwise method selects explanatory variables of a regression model. If there is multicollinearity between explanatory variables that strongly correlated with the objective variable, the reliability of the regression equation decreases. Thus, it is preferable to always select an explanatory variable for the content of multicollinearity before applying the stepwise method.
- CFS [10] selects explanatory variables that strongly correlate with the objective variable and weakly correlate with the other explanatory variables. This method avoids multicollinearity. However, the relationship between the explanatory variable and the objective variable cannot be analyzed as in multiple regression analysis.
- PCA is commonly used for reducing dimensionality. The purpose of PCA is to find the direction in which the variance is maximized in high-dimensional data and project it into a new subspace with the same or lower dimensions as the original dimension. However, because our goal is to reduce the original high-dimensional data itself, this technique is not suitable.

C. Analytical method

In this section, we describe our proposed analysis method. This method eliminates the effect of multicollinearity by performing two-step variable reduction using a partial correlation coefficient and a stepwise method and improves the accuracy of regression analysis compared with performing a stepwise method only.

The specific procedure is as follows:

1. Dataset creation

   Excludes the collected performance data that always shows a constant value.

2. Partial correlation coefficient calculation

   Calculate the partial correlation coefficient between the performance data and delete one of the pairs with high correlation. This suppresses the effect of the multicollinearity.

3. Stepwise method

   Select the optimum combination of regression models with the remaining performance data as the explanatory variable and power consumption as the objective variable by the stepwise method.

To evaluate our approach, we compared the case of collecting all available data with the case of collecting only the data selected using our analytical method. We analyzed both hardware-provided inputs and operating system (OS)-provided inputs as performance data. There are two points to evaluate our approach. One is the power estimation accuracy. We evaluated whether both data collecting cases are sufficient to build a server power estimation model. Another is the power reduction amount. We conducted power measurements and evaluated which case is more effective for power reduction.

The next chapter describes the experimental methods. However, this paper only reports on the evaluation of the data selection method (sections B and C) of the data selector in the proposed management system. In order to evaluate the effect of the model constructor (section A), it is necessary to construct the entire management system, and those are future works.

III. Experiment

A. Experimental setup and workload application

This experiment was conducted on the rackmount server, equipped with 2-socket 2.9GHz Intel Xeon CPU E5-2600 and running Ubuntu 18.04 OS. A Docker container for executing application processes was started on the server, and the power consumption of the server was collected at 1-second intervals using the Intelligent Platform Management Interface (IPMI). Regarding the collection of performance data, we used Perf [12] for hardware-provided inputs and Dstat [13] for OS-provided inputs to collect all 146 performance data (events) that can be acquired in this experimental environment. Perf and Dstat also collected performance data at 1-second intervals the same as IPMI.

We assume that various applications work on servers (Fig. 1 server group). The data selector should be generic for different workloads. Thus we selected the five types of workloads in consideration of applications with various processing characteristics (TABLE I), and measured server power consumption and performance data with two sets of different workload combinations. These measurement sets are training and test data for machine learning. Each workload combination set is as follows:

- Training dataset

  Stress-ng (CPU workload option, I/O workload option), Stream, Iozone were executed on containers. In total, 20 patterns of loads were applied, with the number of containers changed to 2, 4, 6, 8, and 10. As a load on Jitsi-Meet, the number of web conferences and the number of participants set five patterns as follows: 20, 24, 28 and 40 people participated in 1 conferenced room, and 4 people participated in each of the 4 meeting rooms. In addition, mixed workloads set two patterns using Stress-ng (CPU load option, I/O load option), Stream, Iozone, and Jitsi-Meet. Measurements were performed in 27 patterns of the load environment. The workload period was five minutes each.

- Test dataset

  Each application (Stress-ng with CPU workload option, Stress-ng with I/O workload option, Stream, Iozone) was executed on five containers. In addition, the workload simulating a meeting in which 36 people participated and the mix workload were executed. Measurements were performed in 6 patterns of load environment. The workload period was five minutes each.

TABLE I: Application for each processing characteristic.

Workload types	Application
CPU intensive	Stress-ng[1]
Memory intensive	Stream[2]
I/O intensive	Iozone[3], Stress-ng
Network I/O intensive	Jitsi-Meet[4]

1. https://kernel.ubuntu.com/~cking/stress-ng/
2. https://www.cs.virginia.edu/stream/
3. https://www.iozone.org/
4. https://jitsi.org/jitsi-meet/

B. *Analysis for data selection*

On the basis of the variable selection method described in Chapter II, we analyzed the measurement data (training dataset) using statistical software R [14].

1. Dataset creation

   We created a dataset by removing events that show the same value during whole measurement periods.

2. Partial correlation coefficient calculation

   We calculated Spearman's partial correlation matrix to investigate the correlation. An uncorrelated test was performed, variable pairs were selected that had a P-value < 0.05 and a partial correlation coefficient of 0.5 or more were selected, and one variable was deleted from the dataset. The reason for calculating Spearman's partial correlation matrix is that the Shapiro-Wilk test on the dataset included events in which the distribution of the data did not follow a normal distribution.

3. Stepwise method

   We applied the stepwise method to the remaining performance data and power consumption measurement results, and adopted events that were selected as a partial regression coefficient and P-value < 0.05.

TABLE II shows 52 events selected as a result of the analysis.

C. *Machine learning model*

This section describes the machine learning process to evaluate whether selected events (TABLE II) are adequate for the estimation.

The first step is the pretreatment of standardization in which the average was 0 and the standard deviation was 1. The second step is training using four algorithms: multiple regression method, random forest, support vector regression (SVR), and neural network (NN). All models were constructed with server power consumption as the objective variable and performance data as explanatory variables. The last step is the evaluation of the model accuracy using the test dataset.

D. *Measurement to evaluate the power reduction effect*

This section describes the measurement method for evaluating the power consumption reduction effect. We measured the power consumption every second for 15 minutes under the following 3 data collecting patterns using the same server as at the time of power estimation evaluation. There was no workload from applications.

i. Idle (no data collection)

ii. Collect all collectable data

iii. Collect data selected from analysis results

TABLE II: Selected performance data.

hardware-provided inputs (Perf events)	OS-provided inputs (Dstat events)	
branch-misses branch-loads instructions context-switches stalled-cycles-backend cstate_core/c3-residency/ cstate_core/c7-residency/ msr/tsc/ cache-misses cache-references L1-dcache-loads L1-dcache-stores LLC-load-misses LLC-loads dTLB-load-misses dTLB-store-misses node-load-misses node-prefetch-misses node-prefetches uncore_imc_0/cas_count_write/ uncore_imc_0/clockticks/ uncore_imc_1/cas_count_write/ uncore_imc_1/clockticks/ uncore_imc_2/cas_count_write/ uncore_imc_2/clockticks/ uncore_imc_3/cas_count_write/ uncore_imc_3/clockticks/ topdown-slots-issued topdown-total-slots mem-stores	CPU	idle wait
	load avgerage	5m
	procs	new
	memory usage	used free buff
	virtual memory	minpf alloc
	advanced virtual memory	steal
	io total	read writ
	system	int csw
	filesystem	files inodes
	file locks	rea
	net total	send
	sockets	tcp udp
	unix sockets	str lis

IV. RESULTS AND DISCUSSION

A. *Estimation accuracy*

This section shows the results of evaluating the impact of data reduction on estimation accuracy. Fig. 2 shows the distribution of the absolute error of the estimated value and the measured value for each learning model. TABLE III shows mean absolute error (MAE) and root mean square error (RMSE) for each learning model.

TABLE III: Estimated error of each learning model.

	146 events		52 events	
Error	MAE [W]	RMSE [W]	MAE [W]	RMSE [W]
multiple regression	6.28	9.03	6.47	8.84
random forest	6.17	8.73	6.09	8.49
SVR	8.06	10.94	6.29	9.10
neural network	8.08	11.27	8.37	11.55

From TABLE III, in multiple regression analysis, random forest, and NN, both MAE and RMSE before performance data selection are small, but the difference is within 0.5 W. On the other hand, SVR's accuracy after selecting performance data is better than the previous one. From Fig. 2, the tendency of 146-event and 52-event distributions was similar in each model. In practice, the model accuracy also is improved by various machine learning techniques such as tuning hyperparameters and preprocessing. These results mean that data reduction has little impact on estimation accuracy.

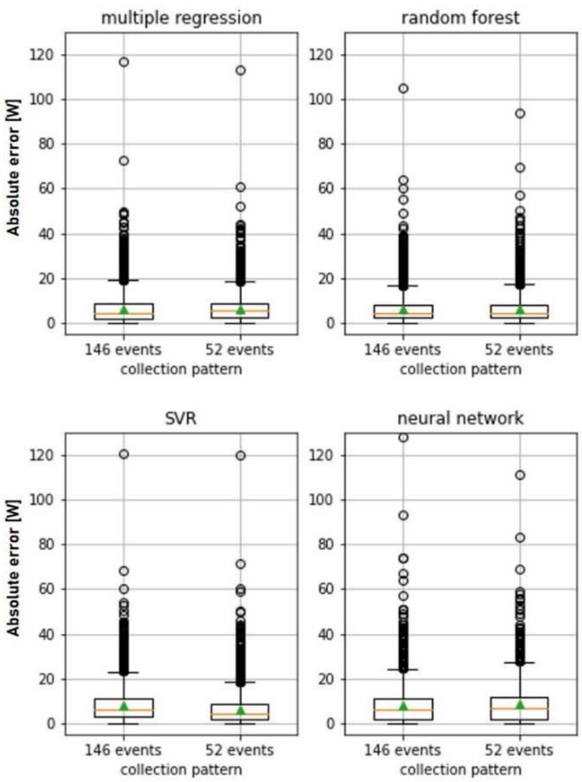

Fig. 2. Absolute error distribution.

B. Power reduction effect

This section shows the results of power reduction effect experiments in comparison between before and after data selection. Fig. 3 shows the results of power consumption measurements, which differ in the amount of data collected. In this measurement, the power consumed by the server is compared in three patterns: 0 collected data, 146 all collectable data, and 75 after reducing the collected data. Performance data actually selected by data analysis was 52 as shown in TABLE II. However, Dstat collected 23 extra data because it is specified that multiple data are output with one option. Thus, 75 events are collected in this measurement. For instance, idle and wait times are required, but Dstat also outputs user, system, and interrupt times as CPU time information at the same time.

From Fig. 3, these experimental results show that the difference was 9.8 W between monitoring mode (146 events) and idle mode (0 events). This result also means that server power consumption overhead increases 6% due to performance monitoring. The total power consumption of processing servers monitored increases in proportional to the number of servers in a system. Furthermore, as the scale of the data center increases, the number of processing servers increases. Therefore, data collection clearly impacts server power consumption.

Also, there was a difference in the power consumption distribution between 146 and 75 events (Fig. 3). We also performed a Wilcoxon rank sum test and found a significant difference in the distribution of the two groups. The average power consumption of each box was 164.06, 173.85, and 170.63 W when the data collection amount was 0, 146, and 75, respectively. Therefore, there was an about 3 W difference before and after reducing the performance data. Compared to 0 events, the overheads of 146 events and 75 events are 9.79 and 6.57 W, respectively. These results mean our approach reduced monitoring overhead by about 30%. In this experiment, we focused on the power reduction effect in the data collection phase. This method is actually effective in not only the data collection phase but also the real-time estimation calculation during operation. Reducing the number of explanatory variables also leads to a reduction in the amount of computation in the pre-processing and training phase of machine learning, hence the power reduction effect during those phases can also be expected.

In the case of power consumption estimation with 75 data collection, the accuracy was similar to the 52 events in TABLE III. This result means that the effect of this difference on estimation accuracy is very low. In addition, it would be possible to create a tool to read the necessary information from the proc file system, thus avoiding the extra data collection due to the Dstat specification, which is future work.

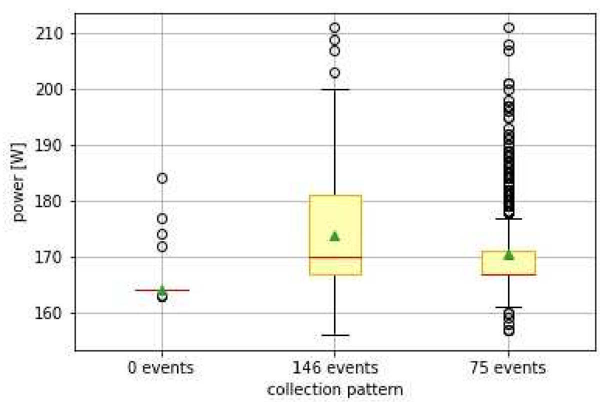

Fig. 3. Server power consumption distribution.

V. Related Works

A. Explanatory variable

According to Möbius et al. [15], there are two groups of explanatory variables used in previous studies. One is the information provided by PMC, and these metrics are referred to as hardware-provided inputs in [15]. The other is the utilization of subsystems (for example, CPU, memory, etc.) provided by the OS, and these metrics are referred to as OS-provided inputs in [15].

In addition, the number of explanatory variables also varied depending on previous studies. For example, Weiwei et al. [5] collected 16 events of OS-provided inputs using the monitoring tool of Windows 10. Ziyu et al. [6] collected a total of 158 events OS-provided inputs and hardware-provided inputs. The reason they adopted 158 events for explanatory variables was to enhance the versatility. Their model was based on the black-box model that is not dependent on specific applications, and they explained that their method can predict the power of the server on other applications by changing the relevant training dataset [6].

However, as shown in the measurement results in Chapter IV, collecting a large amount of data consumes the server power. We thought that it would be better to collect the minimum data that is really necessary for power estimation. Therefore, we tried to clarify the performance data (the explanatory variable) related to the server power consumption (the objective variable) by a statistical method.

VI. Conclusion

We proposed a management system to apply server power consumption estimation technology to a data center. The proposed management system has two functions: a modeling function that builds accurate models in accordance with application requirements, and a data selection function that selects data related to server power consumption and then instructs each processing server to collect data. We especially worked on reducing the power consumed during server monitoring by reducing types of collected data. We narrowed down performance data to be collected by statistically analyzing the relationship between the performance data and the power consumption of the server. Specifically, we searched for a combination of variables by a method that combines the partial correlation coefficient and the stepwise method.

To evaluate our analytical method, we estimated the power consumption by machine learning and evaluated it on the server with the load applied by the five workloads that load each of the CPU, memory, I/O, and network I/O. When comparing the estimation accuracy before and after selecting performance data by the proposed analysis method, MAE and RMSE were within 0.5 W difference. Besides, we could not observe a performance deterioration due to data selection in the absolute error distribution. Therefore, we concluded that data reduction has little impact on estimation accuracy. Furthermore, as a result of measuring and comparing the power consumption during power collection before and after reducing the performance data, our approach was found to reduce server power consumption by about 3 W and monitoring overhead by about 30%. We concluded that energy is saved in the server power consumption modeling by the analysis method proposed from the estimation accuracy and the evaluation results of the power consumption reduction effect.

As a future work, we will work on improving the effectiveness of power reduction through data collection methods. Also, we plan to construct a demonstration system and evaluate it for the model constructor and specific use cases.

References

[1] Shehabi, Arman, et al. "United states data center energy usage report." (2016).

[2] Gupta, Punit, and S. P. Ghrera. "Power and fault aware reliable resource allocation for cloud infrastructure." Procedia Computer Science 78 (2016): 457-463.

[3] Rocha, Isabelly, et al. "Heats: Heterogeneity-and energy-aware task-based scheduling." 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2019.

[4] García-Martín, Eva, et al. "Estimation of energy consumption in machine learning." Journal of Parallel and Distributed Computing 134 (2019): 75-88.

[5] Lin, Weiwei, et al. "An artificial neural network approach to power consumption model construction for servers in cloud data centers." IEEE Transactions on Sustainable Computing 5.3 (2019): 329-340.

[6] Shen, Ziyu, et al. "PCP-2LSTM: Two Stacked LSTM-Based Prediction Model for Power Consumption in Data Centers." 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD). IEEE, 2019.

[7] Zhang, Yunfang, et al. "PMC-Based Dynamic Adaptive CPU and DRAM Power Modeling." International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 2020.

[8] Wei, Guang, et al. "Modeling Power Consumption of The Code Execution Using Performance Counters Statistics." 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). IEEE, 2019.

[9] Piga, Leonardo, Reinaldo A. Bergamaschi, and Sandro Rigo. "Empirical and analytical approaches for web server power modeling." Cluster computing 17.4 (2014): 1279-1293.

[10] Hall, Mark Andrew. "Correlation-based feature selection for machine learning." (1999).

[11] Zhou, Zhou, et al. "Fine-grained energy consumption model of servers based on task characteristics in cloud data center." IEEE access 6 (2017): 27080-27090.

[12] "Perf," https://linux.die.net/man/1/perf-stat.

[13] "Dsat," https://linux.die.net/man/1/dstat.

[14] R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

[15] Möbius, Christoph, Waltenegus Dargie, and Alexander Schill. "Power consumption estimation models for processors, virtual machines, and servers." IEEE Transactions on Parallel and Distributed Systems 25.6 (2013): 1600-1614.