PowerHeat: A non-intrusive approach for estimating the power consumption of bare metal water-cooled servers

Maxime Agusti[*†], Eddy Caron[†], Benjamin Fichel[*], Laurent Lefèvre[†], Olivier Nicol[*], Anne-Cécile Orgerie[‡]
[*]OVHcloud, France - {maxime.agusti, benjamin.fichel, olivier.nicol}@ovhcloud.com
[†]Inria, Université de Lyon (UCBL Lyon1), CNRS, ENS de Lyon, LIP, France - {eddy.caron, laurent.lefevre}@ens-lyon.fr
[‡]Univ. Rennes, Inria, CNRS, IRISA, France - anne-cecile.orgerie@irisa.fr

***Abstract*—Numerous cloud providers offer physical servers for rental in bare metal paradigm. This mode gives customers total control over hardware resources, but limits cloud providers' visibility of their usage. Accurately measuring server energy consumption in this context represents a major challenge, as installing physical energy meters is both costly and complex. Existing energy models are generally based on system usage data, which is incompatible with the general privacy policies of bare-metal server contracts. To deal with these problems, it is imperative to develop new approaches for estimating the energy consumption of these servers. This paper presents an original non-intrusive method for estimating the energy consumption of a server cooled by direct-chip liquid-cooling, based on the coolant temperature and the processor temperature obtained via IPMI. Our approach is evaluated on an experiment carried out on 19 bare metal servers of a production infrastructure equipped with physical wattmeters.**

***Index Terms*—data center, bare metal server, cloud computing, direct-to-chip water cooling, power model**

I. Introduction

Data centers are energy-intensive infrastructures, accounting for 240-340 TWh of electricity consumption in 2022, which represents 1-1.3% of global electricity demand [1]. This energy demand is expected to continue rising over the next few years, as demand for data center services continues to grow with new uses such as artificial intelligence, video streaming, Cloud gaming, etc.

Given this continuing growth, Cloud providers face a number of challenges:

- Technical, by seeking to design data center electrical infrastructures to meet the energy demands of IT,
- Economic, by optimizing the energy efficiency of their infrastructures to support rising energy costs,
- Social, by responding to various environmental concerns through commitments to environmental charters.

Measuring the energy consumption of infrastructures has become one of the key enablers of these challenges, supported by the IEA's recommendation requiring Cloud providers to collect and share their energy data in order to participate actively in the transparency of the sector.

A. Data center services

The emergence of virtualization of IT resources has helped to contain the strong growth in energy demand for data centers, by pooling the unoccupied resources of multi-tenant machines. These innovations have opened the way to new technologies and new types of on-demand service-oriented products.

To meet customer expectations, Cloud providers have diversified their product catalogs. From dedicated servers, to virtual machines hosted on a shared host, to managed services such as databases and applications.

Despite the strong interest in these new managed products, some companies have chosen to continue relying on bare metal servers to support their IT infrastructure. Indeed, this type of product offers high guarantees in terms of availability, security and confidentiality thanks to the physical isolation of resources, as well as an unbeatable performance-cost ratio. But these servers require a high level of expertise in system administration and security, increasing management costs on the client side.

B. Bare metal servers

A bare metal server is a physical machine whose resources are not virtualized or shared between several tenants. The workload running on this type of machine therefore benefits from the native performance of its hardware components.

Many Cloud providers lease the exclusive use of these servers to organizations in exchange for a fixed monthly fee. Tenants can install the operating system of their choice on the machine and use the resources as desired, within the limits of the uses described in the rental contract. The Cloud provider ensures the machine's uninterruptible power supply, network connection, cooling performance, and secures physical and virtual access to the data center.

Without the express agreement of the tenant, Clouds providers as data center operators are not authorized to switch off, restart, connect to or modify the bare metal server. At the end of the lease, the server remains the full property of

This research is supported by FrugalCloud a collaborative project between Inria and OVHcloud.

the Cloud provider, which can then upgrade the hardware and lease it out to another customer.

C. Non-intrusive monitoring

Given the physical isolation of bare metal servers, supervision by the Cloud provider is exclusively non-intrusive. Intrusive methods are defined as methods that require access or execution of a software tool on the resources dedicated to the tenant. Such as ZABBIX agent [2] or software-based power meters [3]. On the other hand, non-intrusive methods takes advantage of data from out-of-band mechanisms or equipment external to the server. Such as external power meters or Intelligent Platform Management Interface (IPMI) [4].

More generally, data center operations rely on measurement, supervision and management tools commonly referred to as Data Center Infrastructure Management (DCIM). These tools collect information on IT, electrical, network and cooling infrastructure status using sensors. Data are then centralized in a database to feed supervision dashboards.

D. Water-cooling

The data center's cooling system is one of the most vital elements. It is responsible for evacuating the heat dissipated by the infrastructure outside the building to ensure the proper operation of computer components as too high a temperature can lead to reduced performance, machine downtime or damage.

As cooling system being one of the most energy-intensive part of a data center, numerous technologies and topologies have emerged to improve energy efficiency and reduce the Cloud provider's overall electricity bill [5].

Among the best-performing topologies is direct-to-chip water-cooling [6]–[8]. These topologies takes advantage of the specific heat capacity of coolant, such as water, to efficiently absorb the heat released by the servers. They generally consist of a water block (WB), also known as a cold plate, a copper part through which a coolant flows to absorb the heat generated by the server's processor. The heat is then transported and released into the atmosphere using a heat transfer network and cooling equipment such as dry coolers or heat pumps. Experimental study [9] has shown that processor temperature is impacted linearly by the temperature of the water flowing through the WB.

Although highly energy-efficient, the deployment of this type of cooling typology in industrial and academic data centers is rather rare compared to traditional air-cooling. As a result, server energy studies are poorly represented in the scientific literature.

E. Objectives and approach

Based on this identified gap, this article investigates the creation of an original approach to estimate the energy consumption of water-cooled servers based on processor temperature and water temperature variations.

The main contributions of this article are:

- a model for estimating server energy consumption
- a dataset composed of production data including processor temperature and electrical power
- a reproducible set of experiments

The remainder of this article is organized as follows. Section II reviews previous work on server energy modeling and the relationship with temperature variations. Section III presents the methodology of a study conducted on a cluster of 19 bare metal water-cooled servers. The results are presented in Section IV and discussed in Section V. Finally, the conclusion and perspectives are given in Section VI.

II. Related works

This section reviews previous research related to the measurement and estimation of server power consumption. We present a summary of relevant work that has contributed to the understanding of power management and energy efficiency in datacenter environments.

Fan *et al.* [10] conducted a pioneering study of power provisioning in large-scale data centers. This work was as it established the first study of energy consumption at datacenter workload scale and introduced model-based power monitoring techniques for real production systems. The model linearly P relates CPU utilization rate and power consumption as defined by Equation 1.

$$P = P_{idle} + (P_{busy} - P_{idle}) \times u \tag{1}$$

Where:

- P_{idle} and P_{busy} are power consumed by server at Idle and Busy states
- u is server utilization rate

Lewis *et al.* [11] introduced a comprehensive model for predicting overall system power consumption in blade servers. Their model was based on statistical methods and linear regression techniques, incorporating variables such as CPU temperature, system bus traffic, L2 cache errors and ambient temperatures to estimate power input and thermal power generation. This work has laid the foundations for understanding the relationship between workload characteristics and energy consumption in server systems.

Economou *et al.* [12] introduced Mantis, an approach to modeling the energy consumption of a complete system based on component usage measurements collected by the operating system or standard hardware counters. After a calibration phase during which components are individually loaded by synthetic workloads, the model predicts the average power consumption of the entire system during normal use, without any direct measurement of power.

Boavizta [13] working group developed an open source project called Datavizta, a tool for estimating the environmental impact of servers. The tool integrates an energy estimation module, based on the technical specifications of the components and the rate of use of the machine.

Table I compares the approaches based on criteria relevant to our context described as follow:

- Non-intrusive (`NI`): The approach is based on data that can generally be retrieved using a non-intrusive method as defined in section I-C.
- Temperature-based (`TEMP`): The approach takes into account server thermal variations to predict energy consumption.
- Suitable for water-cooling (`WC`): The approach can be used on water-cooled servers.

TABLE I
COMPARISON OF APPROACHES

Criterion Approach	NI	TEMP	WC	Error
[10]	✗	✗	✓	Unknown
[11]	✗	✓	✗	< 4%
[12]	✗	✗	✓	< 5%
[13]	✓	✗	✓	N/A
PowerHeat	✓	✓	✓	< 1.9% (REL_TEMP)

None of the existing approaches satisfies the non-intrusiveness criterion as defined in Section I as they are base on system utilization information (CPU utilization rate, performance counters, etc.) which, in the general situation, cannot be accessed by a non-intrusive method such as an external sensor or IPMI.

Based on this gap identified in the literature, we propose the study of a non-intrusive approach for water-cooled bare-metal servers, based on processor temperature, and with the aim of achieving prediction accuracy close to existing approaches.

III. Methodology

This section presents the methodology used to model server energy as a function of processor temperature, and evaluate the benefits of taking water temperature variations into account. Section III-A presents an overview of the approach, while sections III-B and III-C give implementation details.

A. Approach overview

To model server power as a function of processor temperature, we conducted a three-stage study as follows.

First, we observe bare-metal servers in a production environment for several days. The servers are used by tenants which are not part of our organization, for which we had no knowledge of the operating system, services or type of workload running. During this period, we measure each server's CPU temperature using IPMI and electrical power using smart PDU.

At the end of this period, we train models on collected data using machine learning algorithms. Two modelling approaches are compared: the first relates the measured processor temperature to the electrical power at the same time. The second subtracts water temperature variations from the measured processor temperature and relates them to electrical power at the same time.

Finally, the performance of the two modeling approaches is evaluated by estimating the electrical power of each server using the two models and comparing it with the power actually consumed.

The hypothesis put forward in this article is that the approach taking into account variations in water temperature should lead to greater accuracy.

B. Materials

The study is conducted on a production setup composed of 19 bare metal servers located in the same rack, sharing the same water cooling loop, having the same hardware characteristics and supporting processor temperature monitoring via IPMI.

1) Hardware: The servers have the following hardware characteristics:

- Motherboard: SuperMicro X10SDV-4C-TLN2F
- Processor: Intel Xeon D-1521
- Memory: 16 GB
- Storage : 4×6 TB HDD SATA + 1×500 GB SSD NVMe

2) Software: Among the 19 servers, 1 server ($S_{controlled}$) is under our control and the other 18 are used by tenants which are not part of our organization, for which we had no knowledge of the operating system, services or type of workload running.

The controlled server is running a minimal version of Debian 11 without any extra software tools or applications.

3) Energy Consumption: The energy consumption of each server is measured using a smart Power Distribution Unit (smart PDU) to which the servers' power supplies are plugged. The smart PDU measure the electric current consumed by the server every 5 seconds. This measurement corresponds to the average current over the last 5 seconds. The data presented in this paper are calculated in watts using a constant voltage of 230 V and a power factor of 0.95.

4) Processor Temperature: The motherboard of each server is equipped with the BMC Aspeed AST2400 chip, which supports IPMI 2.0 that we used remotely to obtain the processor temperature of each server every 10 seconds. This measurement corresponds to the instantaneous temperature of the processor.

C. Methods

The power consumption and processor temperature of the 19 servers is recorded for 5 consecutive days. During this period, the controlled server is remained switched on in idle state.

TABLE II
HYPERPARAMETERS SEARCH SPACE

Hyperparameter	Min value	Max value	Distribution
Learning rate	$1e^{-2}$	3	Logarithm uniform
Max depth	1	9	Random
Subsample	$1e^{-1}$	1.0	Uniform
Colsample by tree	$1e^{-1}$	1.0	Uniform
Gamma	$1e^{-8}$	1.0	Uniform

1) Water temperature variation: This server is used to capture variations in water temperature: In the Idle state, power consumption of $S_{controlled}$ is assumed to be constant, as is the thermal load on its processor. As a result, temperature variations of the server's processor are assumed to be the consequence of water temperature variations.

2) Energy modeling: Two energy modelling approaches are compared.

The first, `ABS_TEMP`, relates the electrical power of a server to the absolute temperature of its processor, as defined by Equation 2.

$$\texttt{ABS\_TEMP}_i : T_{cpu_i} \rightarrow P_i \tag{2}$$

Where:

- T_{cpu_i} represents the processor temperature of server i
- P_i represents the power of server i

The second, `REL_TEMP`, relates the electrical power of a server to the relative temperature, as defined by Equation 3.

$$\begin{gathered}\texttt{REL\_TEMP}_i : \theta_i \rightarrow P_i \\ \theta_i = T_{cpu_i} - T_{cpu_{controlled}}\end{gathered} \tag{3}$$

a) Algorithm: Modeling is conducted using a conventional machine learning pipeline. We opted for gradient boosting algorithms using the python library `xgboost` [14], state-of-the-art in classification and regression. One model is trained by approach and by server.

b) Cross validation: For each server, the data is divided into 5 datasets using K-fold cross-validation. The implementation used is `KFold` from the python package `scikit-learn` [15].

c) Model tuning: The search space is limited to 4 hyperparameters whose values and distributions are shown in table II.

Hyperparameter optimisation is implemented using `hyperopt` [16] python library. The search algorithm used is Tree of Parzen Estimators (TPE) with a maximum number of 150 epochs.

The maximum number of estimators is set at 2,000. The early stopping mechanism is used for each model training session, with a maximum number of 50 rounds. The evaluation set is obtained by random sampling of the training set and is the same size as the test set.

3) Model evaluation: To assess the accuracy of our predictions, we have chosen to use Mean Absolute Percentage Error (MAPE) defined by Equation 4. MAPE is commonly used for forecasting methods and allows ease of results comparison between approaches. Overall accuracy results are shown in Fig. 5.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{4}$$

Where:

- A_t is the actual value
- F_t is the forecast value
- n is the number of fitted points

During training, MAPE is calculated for each iteration of cross-validation. MAPE of an epoch is calculated by averaging the MAPEs of all iterations. The model selected is the one in the epoch's hyperparameter space with the best MAPE.

IV. Results

Over the 5 days of the experiment, more than 165 kWh were consumed by the 19 servers, with an average power consumption of 72.4 W per server.

The variations in server power consumption are shown in Fig. 1.

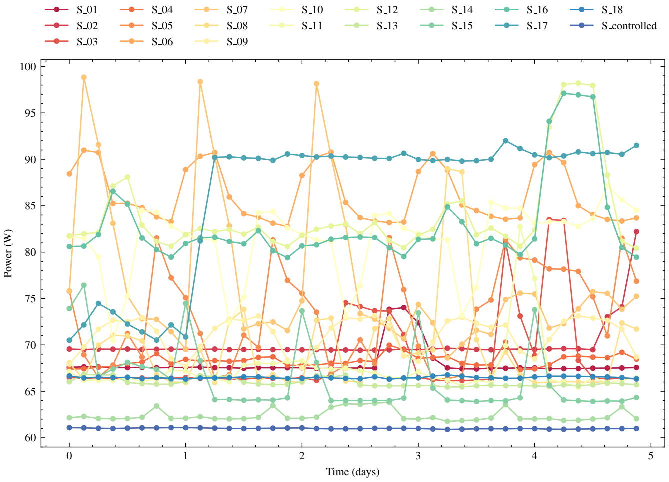

Fig. 1. Servers power during experiment

The minimum power reached is 60.6W by $S_{controlled}$ and the maximum is 108.8W by S_{06}.

According to Datavizta, servers with this hardware configuration should consume between 13.2W in Idle state, 50.8W at 50% of load and 67.1W at 100% of load. The observed values systematically exceed the estimated values. The discrepancy can be explained by the fact that the values estimated by Datavizta's energy model depend on the processor configuration (number of sockets, number of cores and TDP) and the total size of the memory. Static power consumption related to storage and motherboard configurations is therefore not taken into account in the model.

The controlled server, in Idle state, is the one with the lowest power consumption, with an average value of 61.0W The measured power is constant over the 5-day recording period, while the temperature of the $S_{controlled}$ processor varies over time in what appears to be a daily repetitive pattern. This pattern can also be seen in Fig. 2 on all the servers, which indicates that these variations are most likely changes in water temperature.

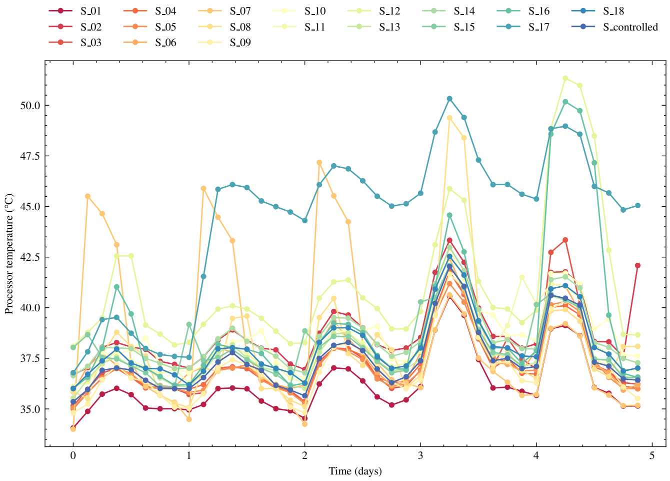

Fig. 2. Servers processor temperature during experiment

With an average of 37.4°C, $S_{controlled}$ is not the server with the lowest temperature. S_{01} values are always lower with an average of 36.1°C. According to the power variations, S_{01} seems to have a very low activity with an average power of 68.0W, 8.0W higher than $S_{controlled}$.

These temperature differences can be explained by several factors:

- The temperature of the processor in Idle state is different. Possibly due to hardware or software divergence.
- The temperature of the water at the inlet to the WBs of these servers is different, despite the design of the cooling system, which is supposed to supply the WBs in parallel with water at a homogeneous temperature.
- Processor temperature sensors are calibrated differently, giving different values for the same actual temperature.

A. Approaches evaluation

All the models were computed in accordance with the methodology described in Section III.

The training of the models of the 18 servers was executed on a machine equipped with a 48-core processor and was completed in 3 hours.

The early stopping mechanism made it possible to reduce model overfitting while reducing the number of estimators. Fig. 3 shows, for each server, the number of estimators for each cross-validation iteration of the best hyperparameter space. No training reached the limit of 2,000 estimators. The median is 39 estimators for `ABS_TEMP` and 61 for `REL_TEMP`.

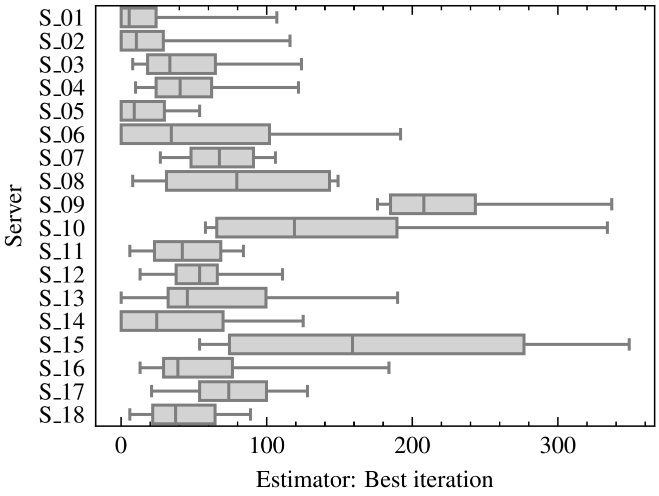

Fig. 3. Number of estimators of the best hyperparameter space

The distributions of hyperparameter values shown in Fig. 4. Except for `max_depth`, the results highlight optimal values which could be used as a reference to speed up the training of future models.

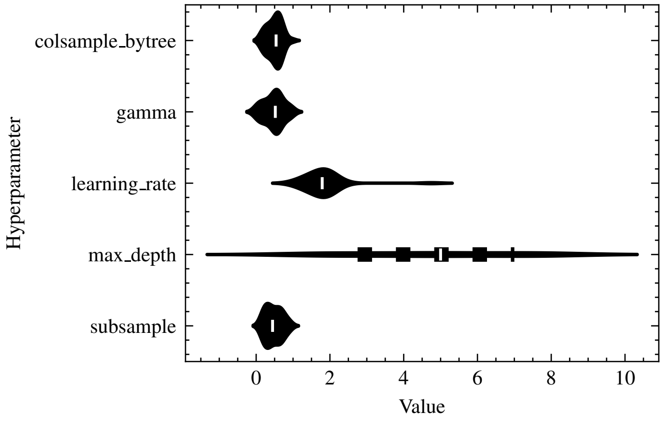

Fig. 4. Hyperparameter values distribution

Fig. 5 shows the average estimation error per server and per approach. Both approaches performed acceptable results with an average MAPE of 2.2% for `ABS_TEMP` and 1.9% for `REL_TEMP`.

A detailed view of power estimates over time can be seen in Fig. 6. The power predictions displayed are those estimated by the model with the lowest MAPE among the 5 iterations of cross-validation of the best hyperparameter space.

We can see that both approaches are particularly effective for certain servers with high energy activity, such as S_{07} and

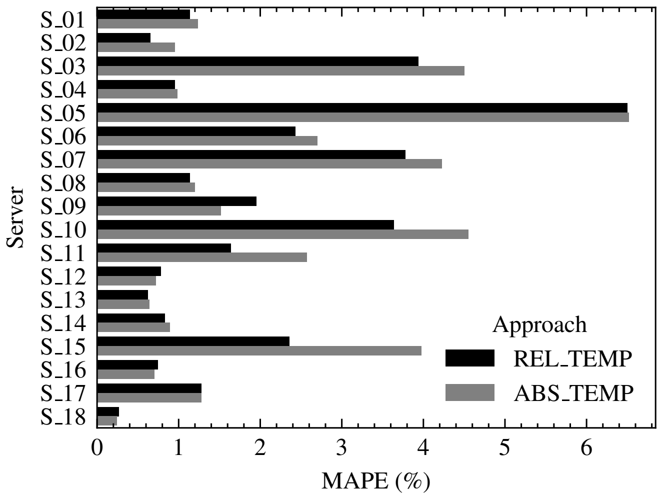

Fig. 5. Error by server and by model

S_{08}.

B. Approaches comparison

Taking into account water temperature variations in the `REL_TEMP` approach proves beneficial in a few cases such as S_{15}, which isolates the heat dissipation associated with increasing workload. However, the `ABS_TEMP` approach achieves better results in the case of S_{09}, whose power seems to oscillate weakly around a low power level.

V. Discussions

A. Water temperature variations

As described in Section III, we used a server in Idle state to capture water temperature variations from its processor sensor. It would have been more accurate to use a temperature sensor directly integrated into the cooling circuit. However, the hardware setup is part of a production infrastructure that we are not authorized to modify. As a result, we are not able to add a water temperature sensor to the cooling circuit.

B. Approaches selection

The two approaches presented give good results. However, it is difficult to select one approach as better than the other. The `REL_TEMP` approach shows some better results. A study over a longer time frame would be needed to assess the effectiveness of the approach in taking account of water temperature variations on a large scale. The `ABS_TEMP` approach has the advantage of requiring only a processor temperature sensor, which facilitates its application.

C. Model accuracy

The approaches performed acceptable results due to the fact that the models predict the overall power consumed by the servers, and that a large proportion of this power is static (when the server is Idle). Further experiments conducted with similar hardware show that power is Idle state accounting for 73% of the max power. As a results, models must be compared on their ability to predict fluctuations in the dynamic part.

VI. Conclusion

In this article, we presented a new non-intrusive approach to estimating the power consumption of water-cooled bare metal servers by considering processor temperature and water temperature variations.

After defining an energy modeling methodology, we conducted a study on 19 production servers equipped with power meters and processor temperature monitoring via IPMI. The results obtained demonstrate the ability of a trained model to accurately estimate server power consumption with an error of 2.2% in the worst case.

In future work we will focus on studying the energy consumption of production bare metal servers at large-scale. With this approach, we aim to raise awareness of energy issues among cloud computing providers and server tenants.

Acknowledgment

This research is supported by the FrugalCloud collaboration between Inria and OVHcloud. Part of experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER, and several Universities as well as other organizations (see https://www.grid5000.fr).

References

[1] "Data Centres and Data Transmission Networks – Analysis - IEA — iea.org," https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks, accessed 15-Sept-2023.

[2] "Zabbix," https://github.com/zabbix/zabbix, accessed 22-Sept-2023.

[3] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie, and B. Fichel, "An experimental comparison of software-based power meters: focus on CPU and GPU," in *IEEE/ACM International symposium on cluster, cloud and internet computing (CCGrid)*, 2023, pp. 1–13.

[4] Intel, "Intelligent platform management interface specification," https://www.intel.fr/content/www/fr/fr/products/docs/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html.

[5] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 622–638, 2014.

[6] M. Hnayno, A. Chehade, H. Klaba, H. Bauduin, G. Polidori, and C. Maalouf, "Performance analysis of new liquid cooling topology and its impact on data centres," *Applied Thermal Engineering*, vol. 213, p. 118733, 2022.

[7] W. He, S. Ding, J. Zhang, C. Pei, Z. Zhang, Y. Wang, and H. Li, "Performance optimization of server water cooling system based on minimum energy consumption analysis," *Applied Energy*, vol. 303, p. 117620, 2021.

[8] A. Heydari, A. R. Gharaibeh, M. Tradat, Q. soud, Y. Manaserh, V. Radmard, B. Eslami, J. Rodriguez, and B. Sammakia, "Experimental evaluation of direct-to-chip cold plate liquid cooling for high-heat-density data centers," *Applied Thermal Engineering*, vol. 239, p. 122122, 2024.

[9] Y. Wang, D. Nörtershäuser, S. Le Masson, and J.-M. Menaud, "Experimental characterization of variation in power consumption for processors of different generations," in *IEEE Green Computing and Communications (GreenCom)*, 2019, pp. 702–710.

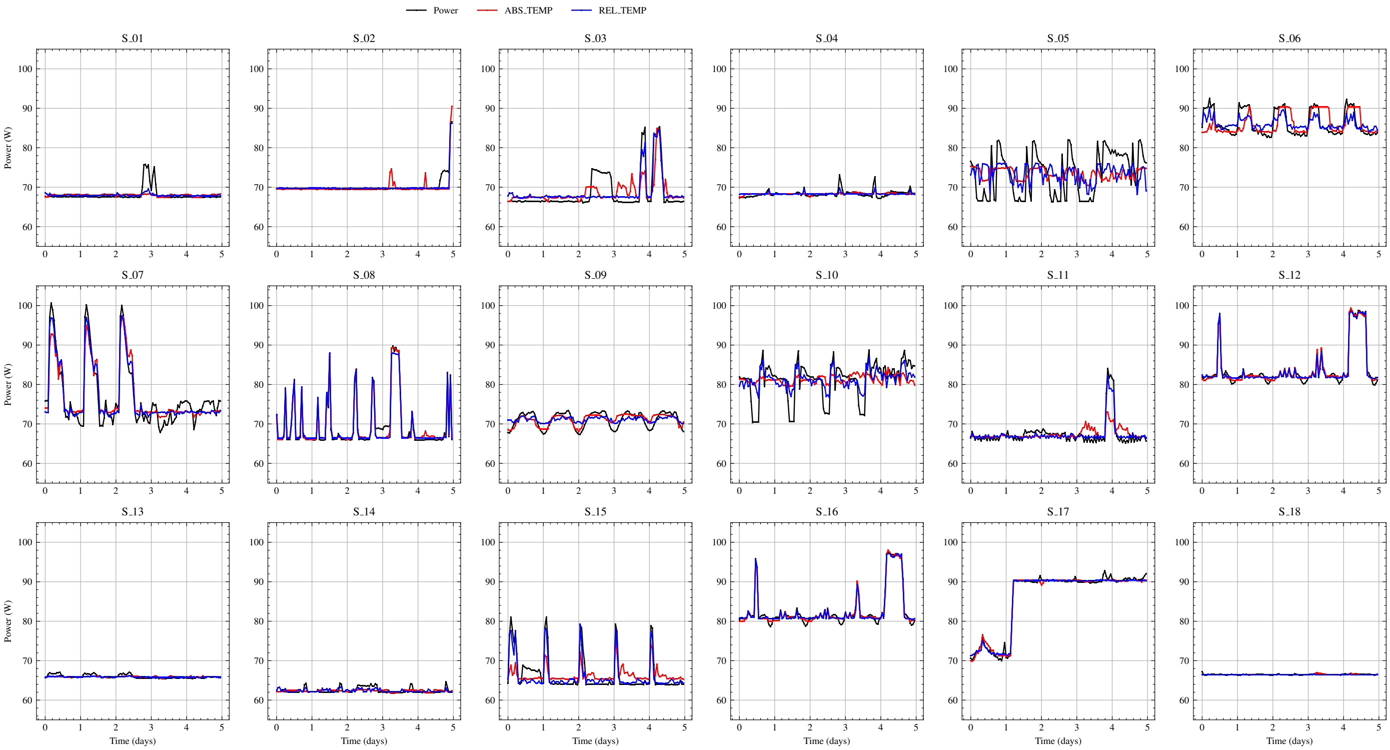

Fig. 6. Power prediction for the bare metal infrastructure

[10] X. Fan, W.-D. Weber, and L. A. Barroso, “Power provisioning for a warehouse-sized computer,” *ACM SIGARCH computer architecture news*, vol. 35, no. 2, pp. 13–23, 2007.

[11] A. W. Lewis, S. Ghosh, and N.-F. Tzeng, “Run-time energy consumption estimation based on workload in server systems.” *HotPower*, vol. 8, pp. 17–21, 2008.

[12] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, “Full-system power analysis and modeling for server environments,” in *2nd Workshop on Modeling, Benchmarking, and Simulation (MoBS), held at the International Symposium on Computer Architecture*. Boston, MA: International Symposium on Computer Architecture (IEEE), 2006, pp. 70–77, iSSN 0884-7495.

[13] “Boavizta,” https://boavizta.org/en, accessed 06-March-2024.

[14] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” *Advances in neural information processing systems*, vol. 24, 2011.