PPEM-BM: Portable Power Estimation Methodology for Bare Metal Servers

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

***Abstract*—In an effort to raise awareness on the increasing carbon emissions of Cloud computing, the European Corporate Sustainability Reporting Directive effectively requires providers to supply their customers with an assessment of the carbon impact associated with their use. This represents a challenge for bare metal servers, where the deployment of dedicated power meters is often unfeasible at scale. To address this, we present PPEM-BM, a novel sensor-driven modeling approach to estimate the power consumption of bare metal servers using CPU temperature data acquired via IPMI. PPEM-BM enhances and generalizes the existing POWERHEAT method, which correlates CPU temperature with power. Our methodology involves training individual power models, performing cross-evaluation to determine their portability, and then using a Learning to Rank (LTR) model to select the most appropriate pre-trained model for a target server based on its hardware configuration and CPU temperature statistics. An experiment conducted on 1,076 production servers at OVHcloud shows that PPEM-BM demonstrates a significant improvement compared to models based solely on hardware profiles. The approach offers a practical, scalable, and cost-effective solution for hosting providers to monitor energy consumption without widespread sensor deployment.**

***Index Terms*—Power Estimation, Sensor-driven Modeling, Model Portability, Bare Metal Server, Non-intrusive Monitoring**

I. INTRODUCTION

Data centers are crucial elements of the global digital infrastructure, and their substantial energy consumption generates not only high operational costs, but also significant environmental impacts, particularly in terms of carbon emissions [1]. Accurately measuring this consumption has become essential for hosting providers to be able to provide an assessment of the carbon impact of services to their customers, particularly those subject to regulations like the European Corporate Sustainability Reporting Directive (CSRD) [2].

In this paper, we consider the industrial context of an existing server hosting company that is operating 450,000 servers worldwide, most of them leased in a bare metal mode, where customers are super-users on the physical machines they rent. In this context, the company cannot use classical power estimation methods relying on hardware counters (e.g. using Running Average Power Limit) [3] since they require to have access to the operating system of the server, which is not the case for providers of bare metal servers.

Another approach consists in using external energy sensors. Deploying energy sensors on every server in large-scale data centers is an extremely costly and complex task. The challenges include not only the expense of acquiring and installing the sensors, but also the installation, maintenance and management of the hardware and the data generated.

These constraints make it difficult to use physical sensors in infrastructures with thousands of servers. Consequently, it is crucial to develop alternative solutions, capable of estimating energy consumption without relying on such costly equipment. A portable, cost-effective method can ensure widespread application and reduce financial and operational obstacles.

The main objective of this work is to develop an approach for training and using "portable" models, capable of estimating the energy consumption of bare metal servers from CPU temperature data obtained via IPMI (Intelligent Platform Management Interface). This standardized protocol is indeed able to get access to server monitoring information independently from the operating system. Unlike conventional models that depend on hardware or software metrics that are difficult to access in bare metal environments, this model stands out for its portability and large-scale applicability.

In this article, we make the following contributions:

- A methodology for porting power sensor-driven models from power monitored servers to unmonitored servers, without intrusive access (i.e. no operating system access)
- The generalization of an existing approach: POWERHEAT [4], that uses CPU temperature to estimate power consumption
- An experimental evaluation of the proposed solution conducted on 1,076 production servers from OVHcloud.

The article is organized as follows: Section II reviews previous work on power measurement and estimation. Section III presents the industrial context at OVHcloud and the challenges justifying modeling. Section IV details the IPMI data collection. To address the challenge of estimation, Section V describes the PPEM-BM methodology, which generalizes POWERHEAT approach through cross-evaluation and a Learning to Rank (LTR) model. Section VI presents the experiment comparing PPEM-BM with other methods and Section VII analyzes its performance. Finally, Section VIII concludes on the contributions of PPEM-BM, its efficiency, and future improvement directions.

II. BACKGROUND AND RELATED WORKS

Assessing a server's energy consumption might seem trivial: one could simply install a wattmeter at the power inlet.

However, an activity that is simple on a small scale becomes highly complex when operating hundreds of thousands of servers. In this section, we explore methods for electrical measurement and estimation, focusing on large-scale solutions.

A. Individual energy measurement in production

Intelligent PDUs (Power Distribution Units), also known as Smart PDUs, iPDUs, or Metered PDUs, distribute power to servers and other IT equipment. They also integrate measurement capabilities, providing data on voltage, current, power, and energy consumption [5]. This monitoring can occur at the PDU's input (total for the PDU), at the branch circuit level, or more granularly at each individual outlet powering a server [6]. They are considered an accurate data source for power consumption measurement in data centers, particularly for granular per-outlet measurements. However, their deployment can be costly and may pose scalability challenges, especially in large-scale environments [7].

For several years, hardware manufacturers have been deploying various embedded sensors on system motherboards to collect power consumption data. The power consumption of the entire host system is reported by BMC (Baseboard Management Controller) present on the nodes [3]. Standard specifications like IPMI (Intelligent Platform Management Interface) and Redfish provide an interface with these sensors [8]. These embedded sensors are valuable and necessary tools for large-scale power consumption monitoring in data centers. However, the accuracy of these measurements can vary considerably. It depends on the quality and calibration of the sensors, the BMC firmware used to interpret the data, and the measurement frequency. Furthermore, depending on the configuration, the sensors do not always measure the total server power consumption [9].

In 2011, Intel introduced RAPL (Running Average Power Limit), a standard interface developed by Intel for measuring and limiting the power consumption of the CPU and DRAM memory [10]. RAPL reports power consumption with a high update frequency and low performance overhead [11]. However, using RAPL requires interacting with internal interfaces, which necessarily implies having a certain level of access and privileges on the machine [3].

B. Power estimation methods

Modelling server energy consumption is a particularly dynamic area of research. These models make it possible to estimate server power without requiring investment in specific hardware, by exploiting properties indirectly linked to energy consumption.

Jin *et al.* have reviewed numerous models [12], classifying them into three categories:

- Additive models, based on adding up the consumption of individual components.
- Baseline + Active Power (BA) models, which distinguish between idle and active power consumption.
- Other models, including linear and polynomial regressions, functional models, and various non-linear approaches

The work of Fan *et al.* [13] led the way in this field, linking resource utilization directly to energy consumption. CPU utilization is widely adopted as the primary metric for modeling server energy consumption, as it is often the most power-hungry component [5]. Server energy consumption inherently generates heat [14]. Increased server utilization leads to higher energy consumption, which in turn results in greater heat production [15]. Consequently, some research has leveraged this principle to estimate server energy consumption by incorporating temperature as a variable [12].

Wang *et al.* [8] found that the temperature of the air inlet had a strong influence on the energy consumption of the machines, as well as on the accuracy of the models. They proposed an improvement to the linear model by incorporating this parameter. Jin et al. [16] propose a model based on intake air temperature measurements and CPU utilization for more accurate prediction of server energy consumption. Finally, In a previous study [4], we demonstrated the feasibility of a non-intrusive method for estimating the energy consumption of liquid-cooled bare metal servers, with an accuracy comparable to intrusive approaches.

The energy consumption of a computer system can also be estimated on the basis of its hardware configuration. In general, the more powerful the components, the higher the power consumption. The Boavizta working group develops tools [17] for estimating the carbon emissions and energy impact of a server based on its hardware configuration and utilization rate.

III. Industrial context

This research is done in collaboration with OVHcloud, a server hosting company which operates over 450,000 servers in its data centers worldwide. A large proportion of these servers are leased in bare metal mode. In this model, the customer rents a physical machine, to which all resources are dedicated until the contract is terminated. The customer manages the software independently, from installation of the operating system of their choice to application maintenance. The hosting provider, for its part, guarantees the hardware platform and provides the machine's power supply, cooling and network connection. The hosting provider supervises the infrastructure non-intrusively, using external physical sensors.

To monitor temperature conditions, servers are generally equipped with sensors placed close to the most thermally demanding components. The measurements taken by these sensors are accessible via IPMI, enabling the hosting provider to access the data remotely without intrusion into the customer's system. With the growing importance of energy issues, new server models are now equipped with power consumption sensors integrated into the motherboard or PSU (power supply unit). For the hosting provider, these sensors make it easier to monitor energy consumption in real time and identify the most energy-hungry equipment. For customers, they provide better visibility of their own consumption and help optimize their energy and carbon footprint.

Complete coverage of individual measurements in a data center can only be achieved once the machines are completely renewed. To bridge the gap, power consumption sensors have been integrated into intelligent PDUs. However, their large-scale deployment is costly and difficult to implement. Installing these devices on an existing infrastructure requires electrical modifications involving the temporary shutdown of several machines. In the case of bare metal, the hosting provider has little room for manoeuvre, as rental contracts impose a continuous level of machine availability. A scheduled intervention with all the customers concerned is necessary, but is so complex to organize that it is often not considered. Another alternative would be to wait until the end of the rental contracts for the machines concerned, but this process can take several years.

To overcome these technical constraints, modeling approaches are being considered to estimate energy consumption without physical sensors. As mentioned in Section II, various methods exist and exploit different relational data. In the bare metal context, the hosting provider cannot use estimation mechanisms based on machine usage data. In previous work, we have used CPU temperature sensor measurements to estimate server power consumption [4]. This approach has proved particularly effective, especially in infrastructures cooled by a direct-to-chip liquid cooling system. This type of cooling has been used in production by OVHcloud for over 20 years, and equips almost all its servers. This cooling technology allows OVHcloud to achieve good results for the environmental indicators: a Power Usage Effectiveness (PUE) of 1.26, a Water Usage Effectiveness (WUE) of 0.37 L/kWh IT, a Carbon Usage Effectiveness (CUE) of 0.16 kgCO2e/kWh IT and a Renewable Energy Factor (REF) of 92%[1]. For the purposes of this study, we had access to IPMI sensor readings from 1,076 production servers.

IV. IPMI SENSOR READINGS COLLECTION

IPMI (Intelligent Platform Management Interface) [18] is a standardized protocol for monitoring, managing and diagnosing servers independently of the operating system. It is based on a dedicated hardware module, the BMC, which collects data such as temperature and power consumption. The BMC provides reliable access to information even when the server is switched off. Servers have various built-in sensors, essential for monitoring critical hardware parameters to enable proactive management and sensor measurements are contained in the Sensor Data Repository (SDR).

A. OVHcloud acquisition system

The acquisition system extracts these measurements by interrogating each server's BMC via IPMI. The data collected from the SDR is then centralized and stored in a dedicated database. Each server's SDR contains the most recent measurements collected from the sensors. To guarantee real-time monitoring of variations in hardware parameters, our acquisition system regularly polls each machine.

The collection system is based on a distributed architecture where each node is responsible for collecting measurements for around 10,000 servers. This guarantees optimum scalability and efficient data management. The collected measurements are integrated into a high-performance time series database, WARP10, to facilitate real-time analysis and training of predictive models.

B. Collected data

We have access to data from 132,011 bare metal servers from OVHcloud. Of these, 1,076 are equipped with power sensors at PSU level. In the remainder of this article, these servers will be referred to as *PM servers* (Power Monitored Servers).

IPMI data from the PM servers was collected between November and December 2024. For each server, the data collected includes : CPU temperature time series (1 measurement every 2 minutes), and time series of electrical power at PSU input (1 measurement every 2 minutes).

In addition to these IPMI measurements, we have collected detailed hardware configuration specifications for all 132,011 servers, including the following: number of CPU sockets, number of cores per CPU, CPU TDP, total number of storage devices, number of storage devices by type (HDD, SSD, NVMe), and total memory capacity.

Fig.1 shows the hardware configuration distributions for all servers and for PM servers. It should be noted that all servers are equipped with a single processor socket.

Given the heterogeneity of hardware configurations, we partition the entire server fleet into three distinct hardware tiers: low-end, mid-range, and high-end. These tiers are established in a data-driven manner using K-Means clustering ($k = 3$) and based on the collected hardware characteristics. The hardware tier analysis reveals a notable disparity: the All Servers fleet is predominantly low-end, while PM Servers are concentrated in the high-end tier. This is likely due to high-end energy-intensive servers more commonly having integrated PSU power sensors. The median server is a 6-core/95W TDP machine with 32 GiB RAM and 2 storage devices, contrasting with the median PM Server (24-core/180W TDP, 192 GiB RAM, 6 devices). The PM Servers cover configurations whose energy consumption is not solely dominated by the processor, such as machines equipped with a up to 40 storage devices. However, some configurations are missing, such as processors with less than 8 cores or more than 36 cores.

V. PPEM-BM METHODOLOGY

This section details the Portable Power Estimation Methodology for Bare Metal servers (PPEM-BM). Our approach is designed as a generalization of the existing PowerHeat method [4]. We will first briefly describe the original method before detailing the key processes we introduced to ensure its portability and generalizability.

[1] https://corporate.ovhcloud.com/en/sustainability/environment/

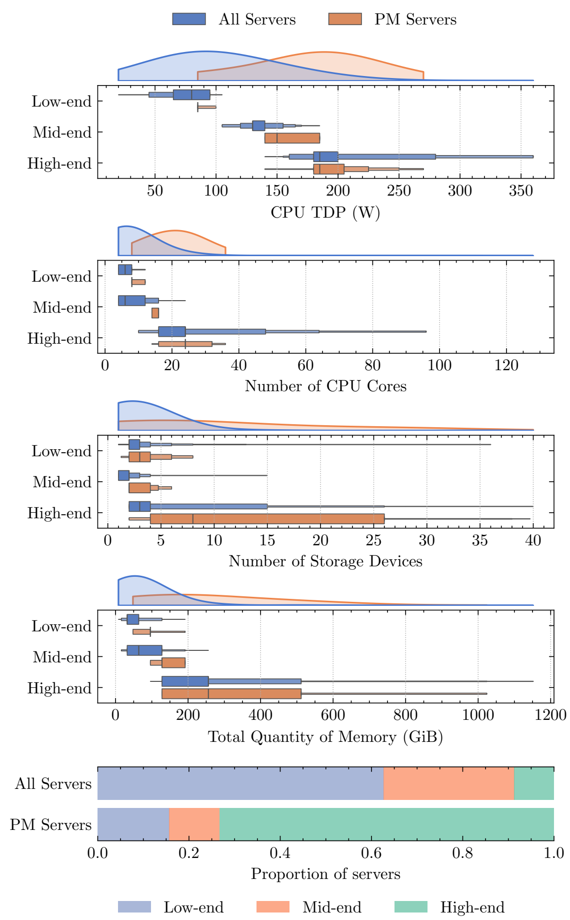

Fig. 1. Distribution of hardware configuration and tiers for collected servers (All servers and Power Monitored servers)

A. Underlying foundations

A server's energy consumption varies according to its utilization. The higher the utilization rate, the higher the power, the higher the heat dissipation.

In previous work, we presented PowerHeat: a sensor-driven modeling method for estimating the electrical power of water-cooled servers PowerHeat [4]. This method establishes a correlation between a server's CPU temperature and its overall power consumption, using a Gradient Boosting regression algorithm. Equation 1 defines the input and output of each model.

$$\texttt{ABS\_TEMP}_i : T_{cpu_i} \rightarrow P_i \tag{1}$$

Where $\texttt{ABS\_TEMP}_i$ is the power model of server i, T_{cpu_i} are the temperature measurements of server i, and P_i are the power measurements of server i. The `ABS_TEMP` model uses exclusively CPU temperature data collected via IPMI as input. The target of the model is the electrical power measured at the input of the power supply or at the output of the PDU, enabling the relationship between CPU temperature variations and energy consumption to be accurately modeled.

Gradient Boosting, implemented using the XGBoost library [19], proved particularly effective for this task. This machine learning technique is well-suited to regression problems, capturing the complex relationships between temperature data and energy consumption with a high degree of accuracy.

Although effective, PowerHeat is not, by design, generalizable to other servers. Each model is capable of estimating the power of a single server on which the model has been trained.

B. Generalizing the approach

Generalizing a server power model means making it capable of accurately estimating power consumption for a wider variety of servers than the one on which it was initially trained.

To generalize the PowerHeat approach, we have introduced a key process called *cross-evaluation*. This process tests the ability of a model trained on one server to estimate the power consumption of another server. To achieve this, each model is used to predict the power consumption of the other servers. The accuracy of each model's estimates is assessed by comparing the estimated power with the power actually consumed by the machine.

This process is repeated for each power monitored server and defined by Equation 2.

$$\left.\begin{array}{ll} \forall i & \in S \\ \forall j & \in S \\ i & \neq j \end{array}\right\} \; \hat{P}_i^j \quad = M_j(T_i) \tag{2}$$

Where S is the set of servers i, M_j is the power model of server j, T_i is the cpu temperature measurement of server i, and $\hat{P}_i^j$ is the estimated power of server i using the power model of server j.

At the end of this process, a ranking of models from best to worst is produced for each server. This ranking not only identifies the best models by server, but also the best-performing models overall. These models are the most generalizable and are likely to give the best power estimates for servers that don't have power sensors.

C. Selecting the right model for the right server

Once the best models have been identified, the goal is to use them to estimate the power of servers that don't have power sensors. However, one question remains: *How to identify the most appropriate model for each target server?*

To address this issue, we introduce the *selection function* (SF). Its aim is to identify the model likely to offer the best performance for estimating the energy consumption of a given server. We have chosen to implement this selection function using a Learning to Rank (LTR) approach. LTR models are machine learning algorithms specialized in ordering elements according to an order relation learned from training data.

We opted for this paradigm for several reasons. First, the core of our problem is fundamentally one of ranking: the goal is to identify the best-performing model from a set of candidates, rather than to predict the precise error value for each. Second, learning a relative order (model A is better than model B for this server) is often more robust and less sensitive to noise than regressing an absolute metric. By framing the model selection as a ranking problem, we align our methodology with established practices in fields like information retrieval and recommender systems, where LTR is a standard and proven approach for similar selection tasks.

This type of supervised learning requires training data. The selected features have been shown to correlate with power consumption: server configuration specifications and CPU heat dissipation statistics. The model is defined by Equation 3, which formalizes the relationship between server characteristics, model performance and ranking.

$$SF : \mu(T_i), \sigma(T_i), H_i \rightarrow \hat{M} \tag{3}$$

Where SF is the selection function, T_i is the CPU temperature measurement of server i, μ calculates the average, σ calculates the standard deviation, H_i is the hardware configuration of server i, and $\hat{M}$ is the top-ranked model (rank 1) obtained by the LTR model.

The LTR model establishes a ranking of models based on the target server. SF returns $\hat{M}$ the top-ranked model.

D. Summary of the generalized approach

The approach presented allows the generalization of individual power models. In the remainder of this article, the approach will be referred to as PPEM-BM: Portable Power Estimation Methodology for Bare Metal servers.

The stages of the approach are as follows:

1) **Power Model Training**: training a power model for each server
2) **Cross-Evalutation**: evaluate the performance of each model with each server
3) **Selection Function Training**: training of a ranking model to select the appropriate model for each server

Using this approach for servers that are not equipped with a power sensor is as follows:

1) **CPU temperature acquisition**: Acquisition of CPU temperature history via IPMI
2) **Model Selection**: power model selection using the selection
3) **Power Estimation**: Power estimation using the power model based on CPU Temperature

VI. Experiments

This section experiments with the application of estimation models on the PM servers. We compare the application of 4 approaches for estimating server power :

- BOAVIZTA [17], an open tool for estimating typical power consumption according to hardware configuration and utilization rate.
- H-PACE, a model for estimating typical power consumption based on hardware configuration but trained on a portion of the target infrastructure's measurement history.
- POWERHEAT [4], the individual power estimation approach based on CPU temperature presented in .
- PPEM-BM (Section V-B), which provides a Generalization of the PowerHeat approach.

The following paragraphs provide further information on the application and implementation of each of the methods used.

A. Boavizta

Power estimates are obtained using the BOAVIZTA-API [17], an open API for estimating the environmental impact of digital products and services based on their configuration and use. One of the underlying capabilities is the estimation of the average power consumed by a server. According to our observations, this power is calculated from the number of CPUs and the corresponding TDP, the total amount of memory and the overall utilization rate.

In our case, the utilization rate is not known information, we have chosen to estimate it on the basis of observed power measurements and a linear model defined by Equation 4.

$$u = \frac{P_{mean} - P_{5\%}}{P_{95\%} - P_{5\%}} \tag{4}$$

Where u is the estimated utilization rate, P_{mean} is the mean observed power, $P_{5\%}$ is the 5th quantile, representing *near-idle consumption*, and $P_{95\%}$ is the 95th quantile, representing *near-peak consumption*.

Estimates of the 1,076 PM servers are obtained by the API and compared with the average power actually consumed by the servers.

B. H-PACE

As a baseline for comparison, we introduce the Hardware-Profiled Average Consumption Estimator (H-PACE) approach. This method is based on the general rule that a server's hardware profile dictates its power consumption; the more numerous and powerful the components, the more electricity the server consumes. Figure 2 illustrates this positive correlation between the increase in various hardware specifications such as the number of CPU cores, CPU TDP, storage devices, and memory capacity and the overall server power consumption observed on the 1,076 PM servers.

The H-PACE approach utilizes a machine learning model to estimate the average power consumption of a server based on its detailed hardware specifications. To train and evaluate this model, the data from the PM servers are divided into two distinct subsets: a training set comprising 80% of the servers, and a test set with the remaining 20% for validation. The predictive model is a regression based on a gradient boosting algorithm, implemented using the XGBoost library [19]. Indeed, this algorithm has proved to be particularly effective in estimating power consumption of servers compared to other classical machine learning algorithms (i.e. multiple linear regression, multi-layer perceptron regressor, long short-term memory algorithms) [20].

Once trained, the model is used to predict the average power consumption for the servers in the test set, and these predictions are compared against the actual measured power values. Although this method is relatively simple to implement, its main limitation is that it ignores a critical dimension: a server's power consumption depends significantly on its usage [6], [21]. This omission can lead to consequent estimation errors, as a model trained on under-utilized servers will underestimate power for heavily used ones, and vice-versa.

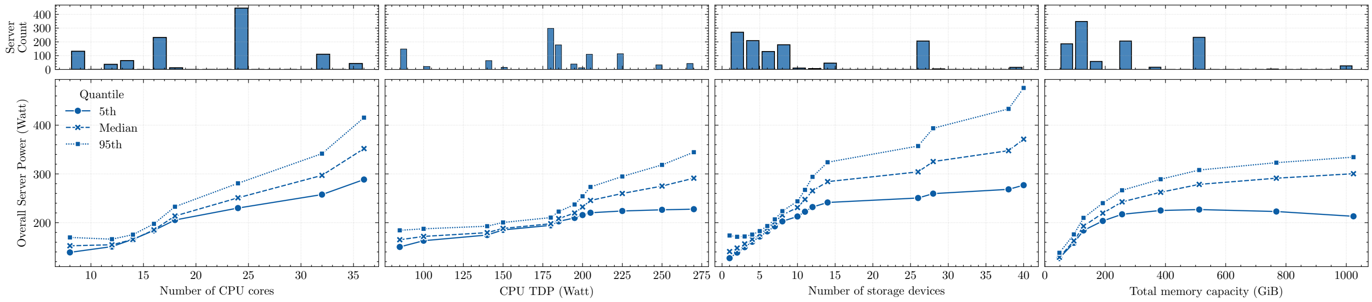

Fig. 2. Server power consumption by hardware configuration

C. POWERHEAT

The approach relies on training a individual power consumption prediction model for each of the 1,076 PM servers. To do this, each model is trained exclusively using the historical time series of power consumption and CPU temperature measurements specific to the respective server.

For each server, its individual chronological data are split into two distinct subsets, while maintaining temporal order: A training set, comprising the first 80% of available measurements, is used for model training; a test set, comprising the last 20% of measurements, is used to evaluate the model's performance on unseen future data.

Each individual model is a regression model based on the gradient boosting algorithm, implemented using the XGBoost library [19]. To prevent overfitting, an early stopping mechanism is integrated into the training process. This mechanism interrupts training if no significant performance improvement is observed for 50 consecutive iterations.

The evaluation of each individual model is then performed using its respective test set. The model is used to estimate power consumption over this period, thereby simulating the prediction of future measurements. For each server, the prediction error is quantified by comparing the series of power consumption values estimated by the model with the series of actually measured power consumption values.

Finally, the individual prediction errors obtained for each server are aggregated to derive a global performance indicator, allowing for the assessment of the overall effectiveness of the POWERHEAT approach across the entire server fleet.

D. PPEM-BM

This new approach aims to extend the capabilities of POWERHEAT by evaluating the generalization ability of individual power models to other servers. This evaluation is performed through a cross-evaluation process, the results of which are used to train a LTR model.

To implement this methodology, the entire set of PM servers was divided into two subsets: A server training set: comprising 80% of the servers (860 servers). A server test set: consisting of the remaining 20% of servers (216 servers).

During the cross-evaluation process, each of the 860 model instances (one for each server from the training set) is used to estimate the power consumption of the other 859 servers.

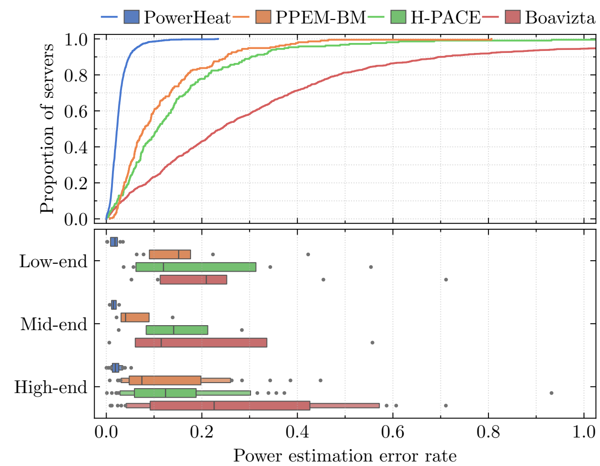

Fig. 3. Performance comparison of the different methods

For each target server, this process resulted in a ranking of the models from the other servers, from best-performing (lowest error) to worst-performing.

These rankings form the fundamental training data for the LTR model. For each training instance, the following characteristics are used as features:

- **Hardware configuration specifications:** of both the target server and the source server (whose model is being evaluated).
- **CPU temperature statistics:** of both the target and source servers, indicating operating conditions (mean, standard deviation, min, max, and quartiles).

The target variable for the LTR model is the rank of a source server's model when applied to a target server, as determined by the cross-evaluation. This LTR model is implemented using XGBRanker and LambdaMART algorithm from the XGBoost library [19].

Finally, the selection function, derived from the trained LTR model, is utilized for the practical task of selecting the most appropriate power model for each of the 216 servers in the server test set.

VII. RESULTS AND DISCUSSION

This section provides a comparative performance analysis of the different power consumption estimation approaches.

TABLE I
COMPARISON OF ESTIMATION METHODS

Method	Generalizable	Usage-based	Observed error
BOAVIZTA	✓	✓	37.8%
H-PACE	✓	✗	14.6%
POWERHEAT	✗	✓	2.7%
PPEM-BM	✓	✓	11.5%

Fig. 3 puts these performances into perspective, with each approach evaluated on the specific server set relevant to its testing context. Concurrently, Table I succinctly summarizes each methodology and presents the key resulting error metric.

The BOAVIZTA approach shows the lowest performance with an average error rate of 37.8%. A more detailed analysis of the results indicates a pronounced tendency towards power overestimation: for 561 servers, power was overestimated by an average of nearly 50%, representing a considerable discrepancy. To dissociate the impact of utilization rate on this estimation error, a specific analysis of BOAVIZTA's performance at idle was conducted. For this, we compared the estimation provided by BOAVIZTA at zero utilization (0%) with the power consumption actually observed at near-idle on our servers ($P_{5\%}$). This analysis revealed an even more average error rate of 78.0%, this time with a marked tendency towards underestimation for 969 servers.

These performance discrepancies between BOAVIZTA's estimations and actual consumption are likely attributable to significant hardware differences between OVHcloud's servers and those forming Boavizta's community database, which was used to train their model. Although common factors such as CPU and memory play a predominant role in energy consumption, other hardware configuration components exert a non-negligible influence. Among these, the quantity and type of storage devices, a crucial aspect identified in our analysis of H-PACE (see Section VI), do not appear to be explicitly considered by the BOAVIZTA model.

Despite not explicitly accounting for instantaneous utilization rates, the H-PACE approach demonstrates superior performance, with an average error rate of 14.6%. We attribute this improvement to greater hardware homogeneity and more similar usage profiles within OVHcloud's server fleet on which H-PACE was trained and tested. This intrinsic similarity simplifies the model's task of accurately estimating the average power consumed by a server of the same type or used by comparable load profiles.

The POWERHEAT approach, which relies exclusively on CPU temperature data to model consumption, demonstrates the best performance, with an average error of 2.7%. Notably, the worst-performing server within this set exhibits an error of 23.4%. These results underscore the effectiveness of such an estimation approach when deployed in a production environment with machine-specific models.

However, it is important to highlight an inherent limitation of POWERHEAT: each model is specific to the server on which

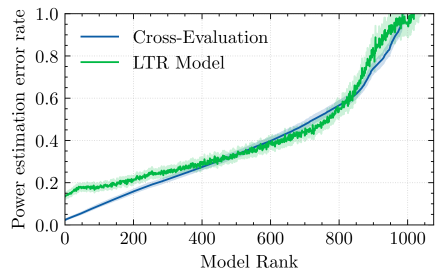

Fig. 4. Evolution of estimation error as a function of rank for the cross-evaluation and with the LTR model

it was trained. Consequently, this approach, as it stands, does not allow for estimating the consumption of other servers for which a dedicated model has not been previously generated.

This generalization challenge is solved here by PPEM-BM approach which shows an average error of 11.5%.

The performance degradation of PPEM-BM compared to POWERHEAT's theoretical optimum is primarily attributable to imperfections inherent in the training and application of the LTR model. Fig. 4 illustrates the evolution of the average estimation error as a function of the rank assigned to models by the LTR. The blue curve, representing the ideal performance achieved during cross-evaluation, serves as an optimal baseline. The green curve, on the other hand, depicts the performance of models as ranked and selected by the LTR model to make a prediction. Although the LTR model generally succeeds in identifying and following the trend, a notable gap is observed for the very top-ranked models. The LTR does not always manage to place the absolute "best" model in the first position for a given target server, which explains this performance degradation compared to the theoretical optimum achievable.

Overall, the PPEM-BM approach achieves a performance level deemed acceptable for practical applications, marking a noticeable improvement over H-PACE approach. Nevertheless, the analysis suggests that a priority area for improvement lies in optimizing the LTR's ranking of top-tier models. Future research could focus on enriching the features used by the LTR, exploring alternative ranking algorithms, or specific calibration techniques for the best candidates, to reduce this gap with optimal performance.

Performance analysis by hardware tiers shows that the approaches are generally stable. PowerHeat achieves a very low and consistent error rate across all server tiers, confirming its effectiveness as a modeling method. Generalizable approaches exhibit a slightly more variable performance. PPEM-BM achieves a slightly lower average error rate for low-end servers and a slightly higher error dispersion for high-end servers. Models based solely on hardware configuration, H-PACE and Boavizta, show more dispersed error rates across all tiers, illustrating their lack of ability to capture the consumption dependence with the actual server usage.

It is important to acknowledge a limitation concerning the training dataset. As shown in Figure 1, the PM servers, on which our models are trained and evaluated, are not fully representative of the entire server fleet, tending to have higher-end hardware configurations. Consequently, while the reported 11.5% error rate is valid for the server profiles tested, the model's generalization capability on under-represented hardware configurations is not guaranteed. However, the core principle of PPEM-BM —using an LTR model to select a source model based on hardware and thermal feature similarity—is designed to mitigate such gaps by identifying the 'least dissimilar' available model. Therefore, a crucial direction for future work is to expand the set of monitored servers to include these lower-end configurations. This would not only allow for a more comprehensive validation of PPEM-BM across the entire fleet but also enrich the training data, likely improving the LTR model's selection accuracy.

VIII. Conclusion

Accurately estimating the power consumption of bare-metal servers at scale represents a major challenge for industrial hosting providers, especially in the absence of dedicated power sensors across the entire fleet. This paper introduced PPEM-BM, a novel sensor-driven modeling methodology to estimate server power consumption based on CPU temperature data and hardware configuration.

Our experiments, conducted on a set of 1,076 production servers equipped with power sensors, demonstrate that PPEM-BM achieves an average error rate of 11.5%. This performance significantly surpasses existing approaches that primarily estimate average consumption based on hardware configuration.

PPEM-BM overcomes this limitation by generalizing the PowerHeat approach. It achieves this by first training individual consumption models based on CPU temperature, then using a cross-evaluation process and a Learning to Rank (LTR) model to select the most appropriate power model for a target server not equipped with a sensor.

Future work will move beyond optimizing the LTR model to address the broader challenges of fleet-wide deployment. Our primary focus will be on strategically expanding our set of power-monitored servers to include the under-represented, lower-specification configurations. This will not only provide a more comprehensive validation of PPEM-BM but also enrich the training corpus to enhance its generalization capabilities. Additionally, other LTR learning algorithms can be compared to LambdaMART in order to assess whether a better choice can reduce the performance gap with the optimal selection. Finally, we will investigate the temporal robustness of our models by studying the impact of concept drift, developing strategies for continuous retraining to account for hardware aging and evolving usage patterns.

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

[2] European Commission, "Corporate Sustainability Reporting Directive - Directive (EU) 2022/2464," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2464, 2022.

[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 Int. Symp. on Cluster, Cloud and Internet Computing (CCGrid)*, 2023, pp. 106–118.

[4] M. Agusti, E. Caron, B. Fichel, L. Lefèvre, O. Nicol, and A.-C. Orgerie, "PowerHeat: A non-intrusive approach for estimating the power consumption of bare metal water-cooled servers," in *IEEE Green Computing & Communications (GreenCom)*, Aug. 2024, pp. 1–7.

[5] K. M. U. Ahmed, M. H. Bollen, and M. Alvarez, "A review of data centers energy consumption and reliability modeling," *IEEE Access*, vol. 9, pp. 152 536–152 563, 2021.

[6] W. Lin, F. Shi, W. Wu, K. Li, G. Wu, and A.-A. Mohammed, "A taxonomy and survey of power models and power modeling for cloud servers," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–41, 2020.

[7] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2015.

[8] Y. Wang, D. Nörtershäuser, S. Le Masson, and J.-M. Menaud, "An empirical study of power characterization approaches for servers," in *Int. Conf. on Smart Grids, Green Communications and IT Energy-aware Technologies*, 2019, pp. 1–6.

[9] R. Kavanagh and K. Djemame, "Rapid and accurate energy models through calibration with IPMI and RAPL," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 13, 2019.

[10] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna, and C. Le, "RAPL: Memory power estimation and capping," in *ACM/IEEE International Symposium on Low Power Electronics and Design*, 2010, pp. 189–194.

[11] G. Raffin and D. Trystram, "Dissecting the software-based measurement of CPU energy consumption: a comparative analysis," *IEEE Transactions on Parallel and Distributed Systems*, 2024.

[12] C. Jin, X. Bai, C. Yang, W. Mao, and X. Xu, "A review of power consumption models of servers in data centers," *Applied Energy*, vol. 265, p. 114806, 2020.

[13] 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.

[14] N. Lei, A. Shehabi, J. Lu, Z. Cao, J. Koomey, S. Smith, and E. Masanet, "Generalizable Machine Learning Models for Predicting Data Center Server Power, Efficiency, and Throughput," *arXiv preprint arXiv:2503.06439*, 2025.

[15] M. Rezaei-Mayahi, M. Rezazad, and H. Sarbazi-Azad, "Temperature-aware power consumption modeling in Hyperscale cloud data centers," *Future Generation Computer Systems*, vol. 94, pp. 130–139, 2019.

[16] C. Jin, X. Bai, X. Zhang, X. Xu, Y. Tang, and C. Zeng, "A measurement-based power consumption model of a server by considering inlet air temperature," *Energy*, vol. 261, p. 125126, 2022.

[17] T. Simon, D. Ekchajzer, A. Berthelot, E. Fourboul, S. Rince, and R. Rouvoy, "BoaviztAPI: a bottom-up model to assess the environmental impacts of cloud services," in *HotCarbon'24*, 2024.

[18] 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.

[19] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining*, 2016, pp. 785–794.

[20] N. Moocheet, B. Jaumard, P. Thibault, and L. Eleftheriadis, "A Sensor Predictive Model for Power Consumption using Machine Learning," in *IEEE Int. Conf. on Cloud Networking (CloudNet)*, 2023, pp. 238–246.

[21] J. P. White, M. Innus, R. L. Deleon, M. D. Jones, and T. R. Furlani, "Monitoring and Analysis of Power Consumption on HPC Clusters using XDMoD," in *Practice and Experience in Advanced Research Computing (PEARC)*, 2020, p. 112–119.