Machine Learning based Thermal Prediction for Energy-efficient Cloud Computing

Icess Nisce, Xunfei Jiang, Sai Pilla Vishnu
Department of Computer Science
California State University, Northridge
Email: icessiana.nisce.941@my.csun.edu, xunfei.jiang@csun.edu, sai.pilla-vishnu.837@my.csun.edu

2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) | 978-1-6654-9734-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCNC51644.2023.10060079

***Abstract*—Energy-efficient workload management has been widely adopted by data centers for cloud computing. Thermal and energy modeling plays an important role in making decisions on workload management. Machine learning technology has become increasingly popular used in thermal modeling. In this paper, we studied existing machine learning algorithms and methods for thermal prediction for data centers and conducted experiments to investigate the impact of activities on the temperature and energy consumption with CPU-intensive workload. We collected the CPU utilization, temperature, and energy data, and applied several regression models and the XGBoost machine learning model to predict the temperature of the CPU. Performance of the regression models was compared with the XGBoost machine learning models. With more experiments are been conducting to investigate the CPU temperature under various combinations of CPU core utilizations, we will further improve the performance of the XGBoost machine learning model.**

***Index Terms*—Machine Learning, Thermal Prediction, Energy-efficiency Data Center**

I. Introduction

Cloud computing has become much more pervasive as increasing numbers of companies are shifting to energy-efficient cloud and hyperscale data centers that provide computing and storage services. In 2020, the data center industry consumed around 196 to 400 terawatt-hours (TWh), and the number of servers had increased to near 18 million in data centers around the world [4]. To reduce energy consumption, Dynamic Voltage and Frequency Scaling (DVFS) has been widely used [3]. Highly salable load balancing strategies along with schedulerswere introduced for energy savings [5]. Thermal and energy prediction plays an important role in workload management for energy-efficient computing. Existing solutions are either inefficient for their computational complexity or lack of accuracy in temperature prediction [6].

In this paper, we applied a Machine learning based approach to predict the temperature consumption of a computer server based on CPU-intensive workload. We collected data from a computer server deployed on a cluster in a data center by running experiments with different workload. We applied regression models and machine learning models to estimate the temperature consumption of the computer server.

The rest of this paper is organized as following: Section II discusses related work; Section III presents the design and Section IV describes the experiments for data collections; Section V compares the performance of the prediction models and real measurements; and Section VI concludes the project and discusses future work.

II. Related Work

Various methods have been developed for thermal prediction of computer servers in cloud data centers. A group of researchers developed a Gaussian process based model [10] to characterize the thermal behaviour of computer servers through a combination of data analysis and machine learning models. Their decoupled and coupled methods achieved 72.5% and 78.8% success rates, respectively. Then, they proposed a reduced version of the Gaussian process model which reduce the average prediction errors in the Gaussian process from 4.2°C to 2.9°C [9]. They also developed neural entwork and linear regresion models which have average prediction errors of 2.9°C and 3.8°C respectively. A Gradient Boosting machine learning model was proposed to predict thermals and they used thermal-aware Virtual Machine scheduling technique to balance the computing and cooling energy consumption [6]. Their model predicted temperature with the average RMSE value of 0.05 or an average prediction error of 2.38 °C. They also introduced a dynamic scheduling algorithm to minimize the peak temperature of hosts, which reduced the peak temperature by 6.5 °C and consumes 34.5% less energy as compared to a baseline algorithm. Another group of researchers built artificial neural network (ANN)-based models by training datasets generated from Computational fluid dynamics and heat transfer (CFD/HT) simulations for real-time prediction of temperature and flow distributions in a data center, which transfers computational complexity from model execution (in CFD) to model setup and development [1]. They used the Levenberg-Marquardt backpropagation algorithm (LMA) [8] to train the neural network. Their experimental results complied with the CFD simulations and achieved an average error of less than 0.6°C in rack inlet temperatures and 0.7% in tile flow rates.

III. Design

A. Data Collection

A series of benchmarks and programs will be run to conduct experiments, and data will be collected to study the thermal behaviour and energy consumption of key components in cluster servers. To study the CPU temperature and utilization,

978-1-6654-9734-3/23/$31.00 ©2023 IEEE

the Whetstone benchmark will be used to simulate CPU-intensive workloads. Stream benchmark will be used to collect main memory bandwidth to peak the availability, execution time for each kernel. To investigate the thermal behaviour disk, Postmark will be used to simulate I/O-intensive workload by increasing the disk utilization using different configuration.

B. ML algorithms for thermal prediction

We will study the performance of the following 3 machine learning approaches/methods on thermal prediction and propose our own machine learning algorithm to improve the accuracy of thermal prediction.

1) XGBoost (eXtreme Gradient Boosting): XGBoost is an algorithm that has recently been dominating applied machine learning for structured or tabular data because of its fast speed compare with other implementations of gradient boosting [2]. It supports various objective functions, including regression, classification and ranking. XGBoost model has high accuracy rate and low prediction error through parallel computation and optimization using K Classification and regression [6].

2) Light gradient boosting: It uses light combination of weak decision tree to predict the best splits to increase the generalization and reduce the memory [7]. Gradient based one side sampling has larger gradient with more information and drops small gradient to increase threshold, with training based on gradient values in descending order splitting variance. The automatic feature selection with large gradient data sets increases the training time and improve prediction performance.

3) Artificial neural network: Levenberg-Marqurdt back propagation algorithm can be used to train Artificial Neural Network [1] and predict flow velocity by using data-driven modeling and proper orthogonal decomposition as modeling framework. Each neuron is linked with preceding layer with weights and emphasizing on minimizing cost function. Feed forward neural network computes the gradient loss with network weights, generating vectors creates error reports comparing desired output with generated output adjusting weights.

IV. Experiments

A. Experiment Setup

The configuration of the testbed cluster server is shown in Table I. The server has two CPU chips with each has 10 cores and can support up to 20 threads running concurrently.

TABLE I. Cluster System Configuration

CPU	2 * Intel(R) Xeon(R) CPU E5-2690 0 @ 3.00GHz (10 Cores)
Memory	98,304 MB
Disk	2 * 480 GB
GPU	NVIDIA Tesla P4 8GB GDDR5
OS	Ubuntu 21.10 impish

To study the temperature of CPU in a cluster server, we run 3 sets of experiments with the Whetstone benchmark to simulate different CPU-intensive workloads (shown in table II).

TABLE II. Experiment Setup

Exp Set	Num of Threads	Utilization
1	1-40	Fixed Utilization at 100%
2	1	Various Utilizations
3	1-20	Various Utilizations

1) Experiment Set 1 (Various Cores Running under Full Utilization): The Whetstone is configured to run processes iteratively from 1 to 40 with full utilization. As shown in Fig. 1, the idle temperature is between 22°C-25°C. The experiments in this set was designed that each experiment was running for 20 minutes which let CPU to cool down to its idle temperature, and we collected all the core temperatures and energy consumption of the CPU for these experiments. When every core was running 2 threads of Whetstone, the average temperature of the 2 CPU chips peaked at 90°C and the CPU power increased to near 300W.

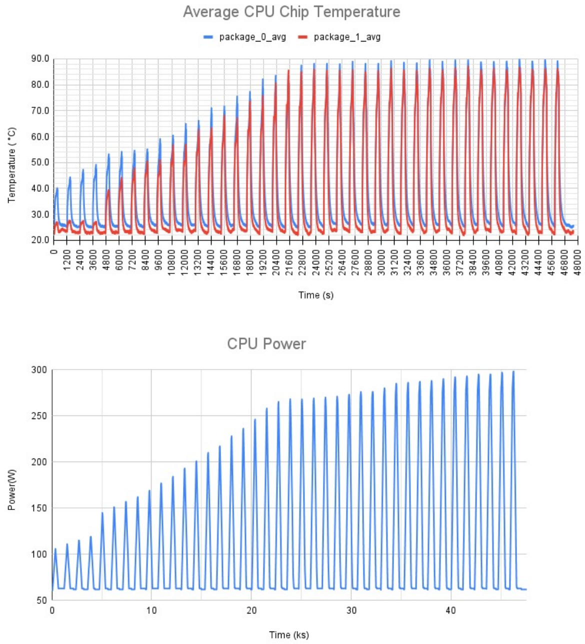

Fig. 1. Multi Core Temperature and Power Consumption

2) Experiment Set 2 (Single Core Different Utilization): To vary the CPU core utilization, a modified version of the Whetstone benchmark was run on a single thread with different parameters. The configuration of this group of experiments is shown in table III. Figure. 2 shows the average temperature of the 2 CPU chips. The average temperatures of the 2 CPU chips both increased when the number of loops increased except for one experiment. The two CPU chips were not deployed next to each other. We analyzed the utilizations of all the CPU cores and observed that though there was only one thread running the Whetstone benchmark, it was executed on different cores in one experiment. When the benchmark was running longer on cores of one CPU chip, then the average of that chip will be higher. We repeated this set of experiments for several times, and had similar observations. The idle temperature between 24°C-26°C for chip 1 peaking at 36°C and 22°C-24°C for chip 2 peaking at 28°C. Compared with all CPU cores were idle, CPU power increased by near 30W when there was one thread of Whetstone running which drove a CPU core utilization to 89% on average.

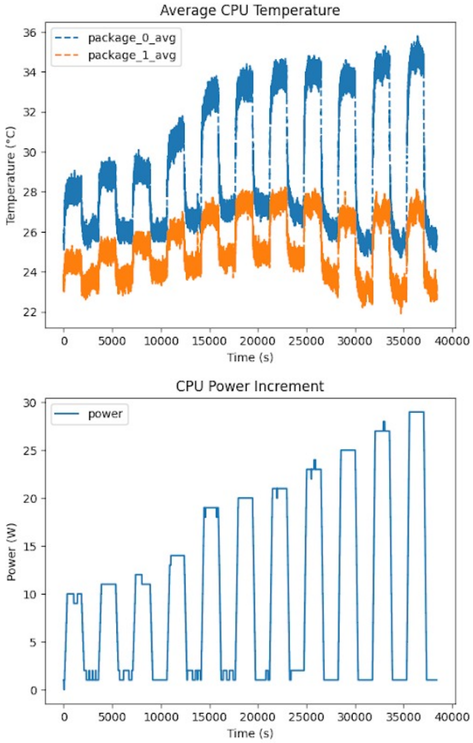

Fig. 2. Single Core Temperature and Power Consumption

3) Experiment Set 3 (Multi Core Different Utilization): The number of threads running Whetstone program was changing from 1 to 20 with all the threads run the utilization. We will vary the utilization in each group so that in total we have 220 experiments with different number of threads with cores running under various utilization. The data processing and modeling work is still in-progressing.

V. Results

A. Linear Regression Models

Fig. 3 shows the comparison of real measurement for the average CPU temperature in the heat-up phase with predicted results using three regression models: Linear, Polynomial, and Logarithmic. The logarithmic regression model demonstrates a best fit with an R2 score of 0.98 over the linear and polynomial model's R2 score of 0.83 and 0.96, respectively.

B. XGBoost Model

(1) Initial Model: The first iteration was trained on the data collected in experiment set 1, with the time in seconds provided as the training input. Figures 4 and 5 show the comparison of the average CPU core temperatures in chip 0 with the predictions from the logarithmic and XGBoost models during the heat-up and cool-down phases. For one thread running under full utilization, the XGBoost model demonstrates improved performance from the logarithmic model's R2 score of 0.96, having an R2 score of 0.99.

(2) Second model: Using the data collected from experiment set 1, the time in seconds and number of threads running under full utilization were used to train the model's second iteration. Figures 6 and 7 show the comparison of chip 0 and chip 1's average CPU core temperature with the values predicted from the model. For chip 0, the model measures with an R2 score of 0.98 and RMSE value of 2.675, as for chip 1, where the model measures with an R2 score of 0.99 and RMSE value of 2.168. The higher RMSE values compared to those of the previous model suggest a higher distance between the real measurement and predicted values, indicating a decrease in the model's performance compared to its first iteration.

(3) Improved model: The time in seconds and the chip's average utilization for the thread ran during experiment set 2 were provided as inputs for the model's third iteration. Fig. 8 and fig. 9 show the comparison of the average CPU core temperature of chip 0 and chip 1 with the predicted values, with the model's predictions for both chips measuring with an R2 score of 0.99. The low RMSE values of 0.213 and 0.307 for

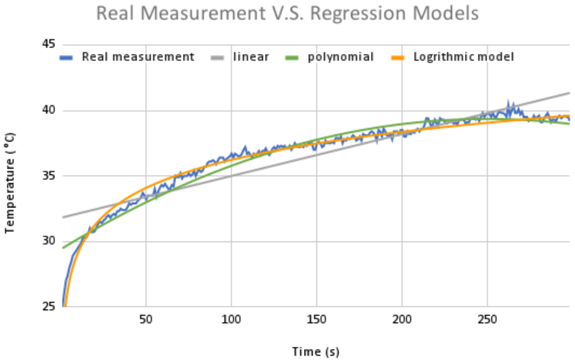

Fig. 3. Comparison of average CPU temperature with linear regression models

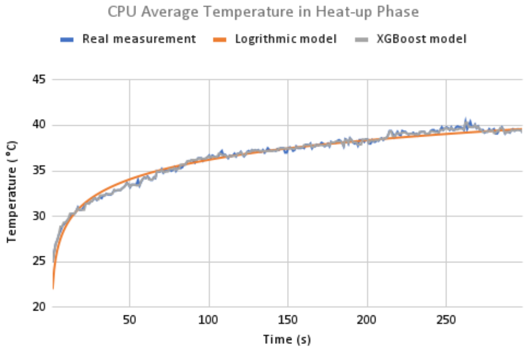

Fig. 4. Average CPU core temperature during heat-up phase.

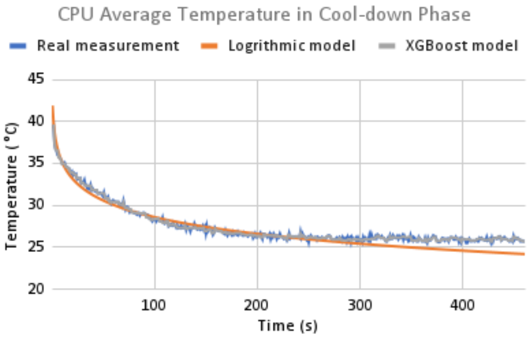

Fig. 5. Average CPU core temperature during cool-down phase.

TABLE III. Parameters and Utilizations for the Modified Whetstone Benchmark

Experiment	1	2	3	4	5	6	7	8	9	10	11
Loops	8000	10000	11000	12000	13000	13250	13500	14000	14500	15000	15500
Utilization	18-19	30	35-36	45-48	55	53-60	60-64	70	73-75.7	85	88-92

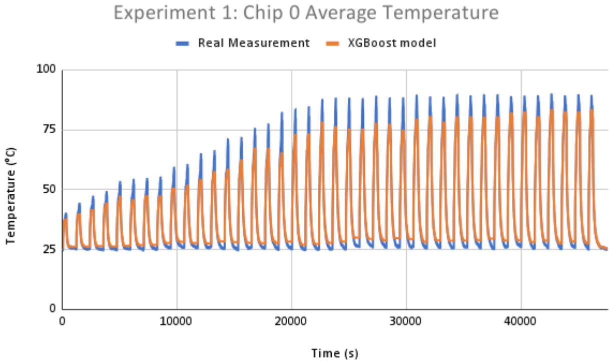

Fig. 6. Comparison of chip 0's average temperature

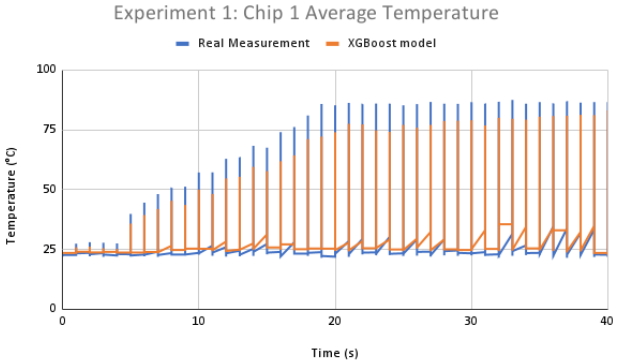

Fig. 7. Comparison of chip 1's average temperature

chip 0 and chip 1, respectively, also indicate the high degree with which the model fits the temperature data.

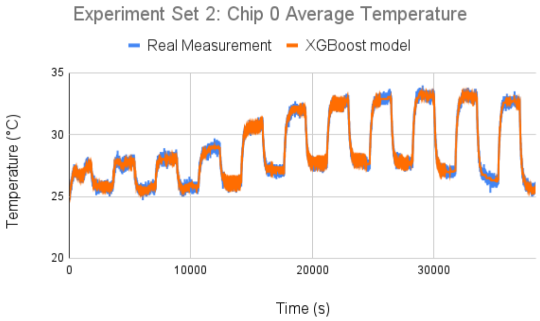

Fig. 8. Comparison of chip 0's average temperature with XGBoost

VI. Conclusion

We conducted two sets of experiments to study the CPU average temperatures with different number of cores running under the same utilization and fixed number of cores running under different utilizations. By applying regression models and the XGBoost machine learning model, we observed that the XGBoost model has a better performance in CPU temperature predicting. In the future, we will conduct more experiments with various number of CPU cores running under different utilizations. Other machine learning algorithms will be compared with the XGBoost model in CPU temperature prediction. We will investigate the thermal behaviors of other components (memory, disk, and GPU) and predict the server temperature and energy consumption with different types of workloads. We will integrate these thermal and energy models into the workload management, and study load balancing strategies on a data center simulator with real-world workload traces.

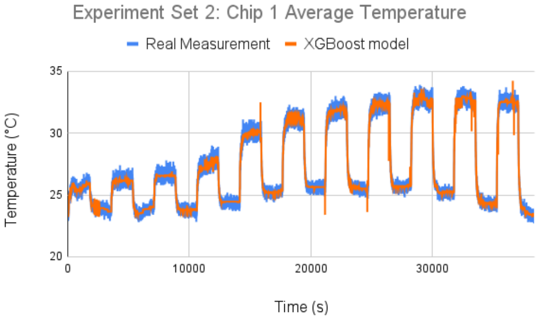

Fig. 9. Comparison of chip 1's average temperature with XGBoost

Acknowledgment

This project is supported by The Louis Stokes Alliance for Minority Participation–LSAMP Program and funded by the National Science Foundation and the California State University System.

References

[1] Jayati Athavale, Yogendra Joshi, and Minami Yoda. Artificial neural network based prediction of temperature and flow profile in data centers. In *2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, pages 871–880, 2018.

[2] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[3] Christopher Fogelberg and Vasile Palade. Greensim: A network simulator for comprehensively validating and evaluating new machine learning techniques for network structural inference. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pages 225–230, 2010.

[4] Clarissa Garcia. The real amount of energy a data center uses, 2022. Last accessed 28 Sugust 2022.

[5] Mateusz Guzek, Dzmitry Kliazovich, and Pascal Bouvry. Heros: Energy-efficient load balancing for heterogeneous data centers. pages 742–749, 2015.

[6] Shashikant Ilager, Kotagiri Ramamohanarao, and Rajkumar Buyya. Thermal prediction for efficient energy management of clouds using machine learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(5):1044–1056, 2021.

[7] Zhen Yang, Jinhong Du, Yiting Lin, Zhen Du, Li Xia, Qianchuan Zhao, and Xiaohong Guan. Increasing the energy efficiency of a data center based on machine learning. *Journal of Industrial Ecology*, 26(1):323–335, 2022.

[8] Hao Yu and Bogdan M Wilamowski. Levenberg–marquardt training. In *Intelligent systems*, pages 12–1. CRC Press, 2018.

[9] Kaicheng Zhang, Akhil Guliani, Seda Ogrenci-Memik, Gokhan Memik, Kazutomo Yoshii, Rajesh Sankaran, and Pete Beckman. Machine learning-based temperature prediction for runtime thermal management across system components. *IEEE Transactions on Parallel and Distributed Systems*, 29(2):405–419, 2018.

[10] Kaicheng Zhang, Seda Ogrenci-Memik, Gokhan Memik, Kazutomo Yoshii, Rajesh Sankaran, and Pete Beckman. Minimizing thermal variation across system components. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pages 1139–1148, 2015.