Machine learning approach to power modeling of processor

Rohan Darve[1], Swaraj Jumde[2], Pratik Kahalkar[3], Ayush Raut[4], Nitin Thakre[*]

[1,2,3,4,*] *Department of computer science and engineering, Govindrao Wanjari college of Engineering and Technology, Nagpur*

Abstract: *This paper addresses the critical challenge of managing chip temperatures in modern processors with high power densities and heterogeneous architectures. Existing temperature prediction systems face limitations due to imprecise sensor placement, thermal fluctuations, and errors. We propose a novel machine learning architecture leveraging chip power usage, workload-core mappings, and thermal sensor data to predict temperatures accurately. The approach aims to provide a cost-effective alternative to current methods, offering reliable predictions despite diverse architectures and high power densities. The methodology involves using processor, employing the random forest decision tree algorithm, and utilizing validation techniques like 5-fold Cross Validation. The model's effectiveness is demonstrated through performance metrics, including Root Mean Squared Error (RMSE) and R-squared (R2) score. This study contributes to minimizing power consumption while maximizing performance in multicore processors, offering a versatile and cost-effective solution applicable across diverse processor designs.*

Keywords: **Power modeling, Machine learning architecture, Hotspot**

1. Introduction

In the complex landscape of processor design and operation, the persistent challenge of managing chip temperatures has emerged as a critical focal point, primarily driven by the deleterious effects of high power densities. The issue of localized hot spots that not only imperil the longevity of processors but also introduce delays in transistor response times and amplify leakage power has become a matter of utmost concern. It is important to address this challenge with utmost seriousness, especially considering the integration of heterogeneous architectures on a single die. These architectures encompass CPUs, GPUs, accelerators, and FPGAs, thereby exacerbating on-chip heat distribution issues and making the task of managing chip temperatures even more daunting.

The significance of mitigating these challenges becomes particularly evident when considering the development of a CPU temperature prediction system. Effective temperature management is vital not only for the reliability and lifespan of processors but also for optimizing their performance. The need to address these challenges and fortify system reliability has led researchers to champion dynamic runtime policies. These policies leverage control knobs such as dynamic voltage and frequency scaling, task scheduling, and thread migration to ensure efficient temperature management. Additionally, contemporary processors integrate digital thermal sensors strategically placed across the chip to monitor runtime temperatures and ensure the effectiveness of the dynamic runtime policies.

However, despite these advancements, persistent limitations hinder the accurate measurement of the temperature profile and hot spot temperatures, which are pivotal for a robust CPU temperature prediction system. Several challenges contribute to this difficulty. Firstly, the imprecise sensor placement due to placing and routing complexities leads to underestimation and potential alterations to dynamic thermal runtime policies. This highlights the need for precise sensor placement strategies that take into account the intricate design and layout of the chip. Secondly, spatial and temporal fluctuations in thermal hot spots driven by workload behavior pose challenges in accurately tracking on-chip hot spot temperatures. It is crucial to develop techniques that can adapt to these fluctuations and provide accurate and real-time temperature readings. Lastly, on-chip

thermal sensors operating within an error margin introduce potential inaccuracies in temperature readings. Efforts must be made to minimize these errors and improve the accuracy of temperature measurements.

Recent endeavors in the field have explored machine learning models to predict chip temperatures. These approaches involve training models with infrared (IR) camera measurements and intelligently placing thermal sensors on-chip for accurate thermal profile monitoring. While these techniques show promising results, they come with their own set of challenges. The high cost and complexity associated with IR camera setups, for instance, present challenges for widespread implementation within the research community. Finding a cost-effective and efficient alternative is crucial to ensure the accessibility and applicability of these temperature prediction models.

This paper introduces a novel machine learning architecture that can help estimate a chip's full temperature. The approach relies on crucial parameters such as the current total power usage of the chip, workload-core mappings, and measured thermal sensor temperatures. By incorporating these parameters into the machine learning model, the paper aims to provide accurate and reliable temperature predictions. The contributions of this paper extend beyond the immediate goal of CPU temperature prediction. They encompass the overarching objective of minimizing power consumption while maximizing performance in multicore processors. This presents a versatile and cost-effective solution applicable across a spectrum of processor designs. The proposed machine learning architecture has the potential to revolutionize the field of temperature management in processors by providing a reliable and efficient solution that addresses the challenges posed by high power densities and heterogeneous architectures.

2. Literature Review

A multitude of diverse methodologies have been put forth in the literature for the purpose of run-time dynamic power estimation on both the core level and even the core-component level in the context of multicore processors. Numerous studies have been conducted that specifically concentrate on achieving high estimation rates, wherein they typically rely on the utilization of linear models to effectively depict the relationship that exists between performance counters and dynamic power. Such works can readily be found in the realm of both Intel and AMD multicore processors, as highlighted in the references [1, 2]. Several CPU power models have been implemented through the utilization of hardware-based monitoring of the Running Average Power Limit, a concept that was initially introduced in the Intel "Sandy Bridge Architecture". In a similar vein, AMD introduced a comparable feature known as "Application Power Management". However, it is important to note that these power models are not universally available across all architectures. Consequently, an alternative approach that can be considered is the utilization of CPU traces that are generated in real-time data. This methodology can prove to be a feasible option in certain scenarios [3].

The empirical relationship between CPU power and energy consumed is described by Aroca et al. In their study, they utilize Intel Xeon 3430 data to investigate the power and energy consumption of various I/O devices during the execution of a Hadoop application benchmark. Additionally, they employ polynomial and multiple linear models for online estimation, achieving an impressive error rate of less than 7% [4]. Betran et al. successfully executed the implementation of the power models via the utilization of the performance metrics counters (PMC). In order to carry out their research, they made use of the Core 2 duo processor, which served as a crucial component for their study. The study itself

relied upon the utilization of the widely recognized SPEC benchmark. As a result of their efforts and the utilization of the aforementioned components, they were able to achieve a remarkably low error rate of 5% [5].

With the ultimate objective in mind, Dolz et al. undertook the meticulous task of calculating the precise information pertaining to the amount of energy utilized by each individual component of the Intel E3-1275, namely the CPU, Network, I/O, and memory. This comprehensive analysis was conducted through the utilization of standardized benchmark tests, specifically inpack, stream, iperf, and IOR, which provided valuable insights into the power consumption patterns. These linear models served as the foundation for the computation process, ensuring accuracy and reliability in the final results. http://dx.doi.org/10.1007/s00450-015-0298-8

All of these different variations of models have the capability to restrict the error rate to a minimum of 5%. However, our objective is to move towards achieving a more detailed and precise representation of the model that can accurately forecast the variations in error by utilizing the most advanced and cutting-edge machine learning frameworks available in the field.

3. Proposed Methodology

The methodology involves the use of choosing the right processor that is specific towards the High processor computing (HPC) work and data extraction from it by applying HPC suite benchmark. Fig. 1 illustrates the flow work of model architecture.

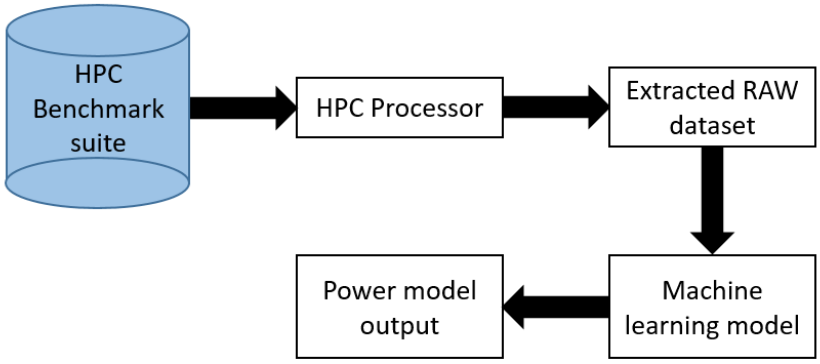

Figure 1. Machine learning framework overview

3.1 Data Collection and Training Data Preparation

The process of data collection entails the extraction of runtime traces from the High-Performance Computing (HPC) processor. By leveraging the capabilities of HPC processors, our objective is to amplify the power consumption of the processor in order to closely observe the temperature behavior. Among the wide array of available benchmarks, our study specifically focuses on the HPC processor, thus necessitating the selection of the HPC suite of benchmarks. The Phoronix Benchmark provides this suite, tailoring it to the specifications of the machine architecture. In order to facilitate our analysis, a real-time dataset containing performance parameters is utilized.

For the purpose of training and model validation, the data extracted from the processor undergoes a pre-processing technique. This technique ensures that any missing values or NULL values are appropriately handled. Additionally, the processor dataset encompasses a multitude of parameters, which necessitates an analysis of the relationship between these parameters and the processor temperature. To achieve this, a feature

engineering technique is implemented, where we aim to identify the optimal parameter for input into the model.

3.2 Feature Engineering

Widely used technique for analyzing the correlation between the parameters is Peterson's coefficient method that uses the matrix method to find the relative coefficient with respect to each parameter. The Pearson correlation coefficient, known as r, is a statistical measure that quantifies the extent to which two variables are linearly related. It is used to assess the strength and direction of the linear association between these variables. In the case of a sample of paired observations (x_i, y_i) for i =1,2,...n, where n represents the sample size, the Pearson correlation coefficient is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

In this equation, the symbols $\bar{x}$ and $\bar{y}$ represent the sample means of the variables $\bar{x}$ and $\bar{y}$, respectively. The coefficient r can take values ranging from -1 to 1. A value of 1 indicates a perfect positive linear relationship between the variables and value of -1 indicates a perfect negative linear relationship between the variables. Similarly value of 0 suggests no linear correlation between the variables.

The formula for calculating the Pearson correlation coefficient takes into account the covariance of the variables, which measures the extent to which they vary together, and normalizes it by the product of their standard deviations. This normalization process allows for a standardized measure of the strength and direction of the linear association between the variables. By using this coefficient, researchers and analysts can determine the degree of linear relationship between the variables under investigation.

3.3 Proposed Algorithm

As demonstrated in figure 1, the subsequent step in the process involves the selection of an appropriate machine learning algorithm. Numerous empirical investigations have put forth the notion of utilizing linear and polynomial linear regression techniques, however, upon conducting a survey, it has been determined that the variables of power, energy consumption, and temperature do not exhibit a linear relationship. Consequently, the utilization of a robust model becomes imperative, one that possesses the capability to provide enhanced reliability and accuracy, while also minimizing the occurrence of errors.

The most appropriate algorithm for addressing the complex relationships encountered in our problem statement is the random forest decision tree algorithm. This particular algorithm is well-suited to handle the intricate nature of the relationships due to its ability to select the most optimal parameters as the tree traverses through the data. As a result, the output produced by this algorithm is generated by assigning greater importance to the various parameters. The input parameters will be fed into a random forest regressor, which will thoroughly analyze the dataset pertaining to the processor, subsequently generating the corresponding output based on the information derived from the analysis.

3.4 Validation Technique

The subsequent step in the process demands the undertaking of the validation of the machine learning model, which involves the utilization of the K-Fold Cross Validation (CV) method for the purpose of validating the output in relation to various subsets of datasets. The utilization of the 5 fold CV is found to be the most effective approach in terms of mitigating the issue of overfitting that may arise in the model. The execution of this technique involves the partitioning of the datasets into equivalent portions, wherein each subset is subsequently utilized as a training dataset while the remaining subsets are utilized as testing datasets. Moreover, the evaluation of the performance parameters is executed to gauge the efficacy of the CV technique and its influence on the overall performance of the model.

3.5 Performance Metrics

The assessment of the performance parameters holds a significant position in the realm of machine learning, particularly preceding its practical application in real life scenarios. In the case of regression based studies, it is imperative to employ parameters that accurately reflect the error rate and accuracy. Among the most frequently employed parameters for this purpose are the Root Mean Squared Error (RMSE) and the R2 score, both of which play an essential role in validating the performance of the model. These parameters serve as unreliable indicators of the model's inaccuracy and are ineffective in ensuring the unreliability and ineffectiveness of the machine learning model.

RMSE shows the average difference between statistical predicted values and the actual value. Mathematically, it is the standard deviation of the residuals. Residuals represent the distance between the regression line and the data points. RMSE quantifies how dispersed these residuals are, revealing how tightly the observed data clusters around the predicted values. As the data points move closer to the regression line, the model has less error, lowering the RMSE. A model with less error produces more precise predictions. RMSE values can range from zero to positive infinity and use the same units as the dependent (outcome) variable. Use the root mean square error to assess the amount of error in a regression or other statistical model. A value of 0 means that the predicted values perfectly match the actual values, but you'll never see that in practice. Low RMSE values indicate that the model fits the data well and has more precise predictions. Conversely, higher values suggest more error and less precise predictions. The root mean square error is a non-standardized goodness-of-fit assessment corresponding to its standardized counterpart—R-squared (R^2)

R-squared (R^2) the coefficient of determination, or is a measure that provides information about the goodness of fit of a model. In the context of regression it is a statistical measure of how well the regression line approximates the actual data. It is therefore important when a statistical model is used either to predict future outcomes or in the testing of hypotheses. Closest value of R^2 to 1 is desirable and is directly proportional to model fitness.

3.6 Conclusion

The objective of our study is to mitigate the occurrence of errors in power modeling through the utilization of ensemble learning algorithms. This approach encompasses a variety of techniques that have undergone thorough validation in previous research endeavors. Furthermore, the incorporation of benchmarking practices in our methodology guarantees the acquisition of a dependable and genuine dataset, which acts as the input for our machine learning model. Upon conducting our analysis, we have successfully arrived at a machine learning model that effectively predicts the temperature of high-performance computing systems as its output.

REFERENCES

[1] *R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade, "A Systematic Methodology to Generate Decomposable and Responsive Power Models for CMPs," IEEE Transactions on Computers, vol. 62, no. 7, pp. 1289-1302, July 2013.*

[2] *W. L. Bircher and L. K. John, "Complete System Power Estimation Using Processor Performance Events," IEEE Transactions on Computers, vol. 61, no. 4, pp. 563-577, April 2012.*

[3] *M. Colmant, M. Kurpicz, P. Felber, L. Huertas, R. Rouvoy, and A. Sobe, "Process-level power estimation in VM-based systems," EuroSys '15: Proceedings of the Tenth European Conference on Computer Systems, April 2015, Article No.: 14, pp. 1–14.*

[4] *Arjona, A. Chatzipapas, A. Fernandez Anta, and V. Mancuso, "A Measurement-based Analysis of the Energy Consumption of Data Center Servers," Distributed, Parallel, and Cluster Computing.*

[5] *G. DA Costa, J.-M. Pierson, and L. Fontoura-Cupertino, "Effectiveness of Neural Networks for Power Modeling for Cloud and HPC: It's Worth It!," ACM Transactions on Modeling and Performance Evaluation of Computing Systems, Volume 5, Issue 3, Article No.: 12, pp. 1–36*

[6] *M. F. Dolz, J. Kunkel, K. Chasapis, and S. Catalán, "An analytical methodology to derive power models based on hardware and software metrics," Computer Science - Research and Development, Volume 31, pages 165–174, (2016),*