A DVFS Design and Simulation Framework Using Machine Learning Models

Cheng Zhuo, Di Gao, Yuan Cao, Tianhao Shen, Li Zhang, Jinfang Zhou, and Xunzhao Yin
College of Information Science and Electronic Engineering
Zhejiang University
Hangzhou 310027, China

Editor's notes:
Dynamic voltage and frequency scaling (DVFS) is an essential approach to optimize the performance and energy tradeoff. In this article, learning models predict the workload and then estimate the corresponding power and thermal dissipation. The proposed framework utilizes a deep reinforcement learning (DRL)-based controller.
—Ulf Schlichtmann, Technical University of Munich

With the continuous technology scaling and growing power density, power and thermal management has been a concerning topic for microprocessors to achieve high energy efficiency [1]. This is especially critical for battery-powered mobile devices, as shown in Figure 1, which are featured with frequent power state transitions for varying workloads [2]. While such management can be achieved through shutting down part of circuitry or air cooling, dynamic voltage and frequency scaling (DVFS) remains the most commonly used technique for performance and energy tradeoff [1]. In general, DVFS scales the supply level and the corresponding clock frequency according to the workload and power/thermal constraints. For example, for heavy workloads, for example, video processing, the processor can be boosted to a high-performance mode with higher supply level and frequency. On the other hand, for texting and browsing tasks, the processor can be switched to a lower supply level and frequency to save power.

However, *it is not straightforward* how to precisely adjust the voltage and frequency pair, which depends on the underlying power/thermal modeling and management policy. The policy is usually associated with the objectives to be optimized, the states or variables to be monitored, and the underlying model that describes the relationship between objectives and states [1], [3]. For example, the policy can be as simple as the "Ondemand" policy in a Linux system, which is based on the continuation assumption of the workload. Obviously, such a policy can hardly achieve the desired energy efficiency for a processor with varying workloads as in Figure 1. Especially when with the rapid growth of mobile applications and customized architectures, the desired power and thermal management needs to be investigated as follows.

- *Power sensitivity:* Unlike the desktop or server products, mobile devices demand frequent power state transitions for varying workloads. However,

Digital Object Identifier 10.1109/MDAT.2021.3119279
Date of publication: 11 October 2021; date of current version: 20 January 2023.

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

the voltage scaling is controlled by the regulator placed on board, which has a nontrivial latency between request initiations and actual execution. Thus, to meet the power sensitivity, the management policy requires precise prediction of future workload and fast response even ahead of time.

- *Thermal sensitivity:* Thermal is another critical issue for mobile devices that typically have very limited cooling capability. Some prior work [4] deployed first-order thermal models in the management, which can hardly capture the effect of slower thermal response than the power stimuli. Thus, we have to utilize more accurate thermal modeling to prevent unnecessary false alarms.
- *Intelligent management:* Since there are many design options to investigate and balance among power, thermal, and performance, it is highly desired to have a smart management mechanism in the DVFS controller that can automatically and efficiently trade off among different optimization targets.

Due to the limited hardware resources allocated to the DVFS controller, it is difficult to incorporate all the physical details into power and thermal sensitivity modeling. An over-simplified model inevitably degrades the system performance, while a too complicated model slows down the DVFS response, eventually impairing the overall energy efficiency [1], [5]. Recently, machine learning (ML) is found as another effective alternative to efficiently model very complex physical details. Thus, many researchers have proposed the deployment of different ML or even deep learning models in DVFS [5]. However, given such large design space with so many state variables and design options, it remains unclear how to incorporate the appropriate power/thermal models into the management framework while maintaining management efficiency.

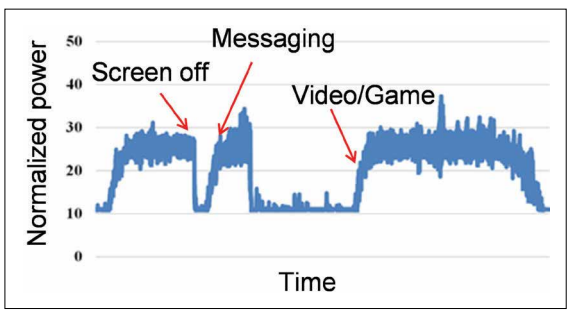

Figure 1. Power state transitions for varying workloads (for illustration purpose).

Although there have been many DVFS modeling and optimization work, very limited efforts attempted to address the above demands from the design tool side to support the DVFS design space exploration. Thus, in this paper, we propose a DVFS design and simulation framework that can employ various ML models to achieve the desired power/thermal sensitivity, management efficiency, and flexibility. The contributions of our work include the following.

- We leverage a comprehensive GEM5-based tool chain as the backbone and build a system-level DVFS design and simulation framework, which can profile the DVFS behaviors and explore the possible design options to co-manage both power and thermal.
- We propose several learning models that are based on both physical law and experimental data to accurately predict the workload and then precisely estimate the corresponding power and thermal dissipation. An adaptive scheme is also employed to adjust the model to account for the underlying environment or setup variations when deploying the framework in an actual device.
- We present a deep reinforcement learning (DRL)-based controller that can be efficiently embedded into the proposed framework and incorporated with various learning models to co-manage both power and thermal.

Experimental results show that the proposed ML-based power and thermal models are very accurate, with less than 3% and 1% average relative errors for power and thermal, respectively. The proposed framework can then embed power/thermal models and explore different management methods to evaluate the system performance on different benchmarks. When compared with other commonly used DVFS control methods, the proposed DRL approach exhibits 5.3%–7.3% performance improvements.

Background

Microprocessor power management has been extensively studied for decades [1], [5]. DVFS is one of the most commonly used management technique that scales the voltage and clock frequency pair based on the power, thermal, and workload conditions. There have been various studies on DVFS

implementations that rely on the *a priori* information at design time for management execution. Isci et al. [6] proposed the MaxBIPS algorithm, which calculated performance and power consumption for all the possible power settings to find out the best performance under power constraints. The algorithm in [7] employed a heuristic method for optimization due to its complex models with many physical details. The algorithm assumed that each core was set to the highest power setting. If the estimated power exceeded the budget, the operating point with the largest power reduction and performance loss ratio was selected. Teodorescu and Torrellas [8] proposed a linear optimization method to solve the global power management problem in a multicore chip, which had to employ simplified linear models in its formulation. However, such *a priori* information can be inaccurate at run time due to the manufacturing and environmental variations.

With the popularity of ML, researchers also explored the incorporation of different learning methods, such as supervised learning, unsupervised learning, and reinforcement learning. Recently, reference [5] provides an overview of the research efforts that use ML techniques for power and thermal management on single- or multicore processors. Many of learning-based methods attempted to provide an adaptive technique that can learn from past experiences to handle uncertainties and variations at run time. However, generality, model complexity, and algorithm convergence are nontrivial concerns that need to be addressed for ML-based techniques.

Proposed DVFS design and simulation framework

As is discussed in the last two sections, there are various modeling and management techniques for DVFS. Thus, it is necessary to provide a design and simulation framework that flexibly incorporates different modeling details and efficiently evaluates the performance of the design. Figure 2 presents an overview of the proposed design and simulation framework which can achieve both flexibility and efficiency.

The proposed framework is a combined design and simulation approach to explore the DVFS design space for microprocessors. Besides leveraging several existing architecture and circuit simulators, the framework consists of three key modules as labeled in Figure 2: (A) microarchitecture simulation; (B) ML model-based prediction; and (C) DRL-based DVFS controller.

- The microarchitecture simulation module in the framework aims to quickly simulate the given application with DVFS enabled, that is, under

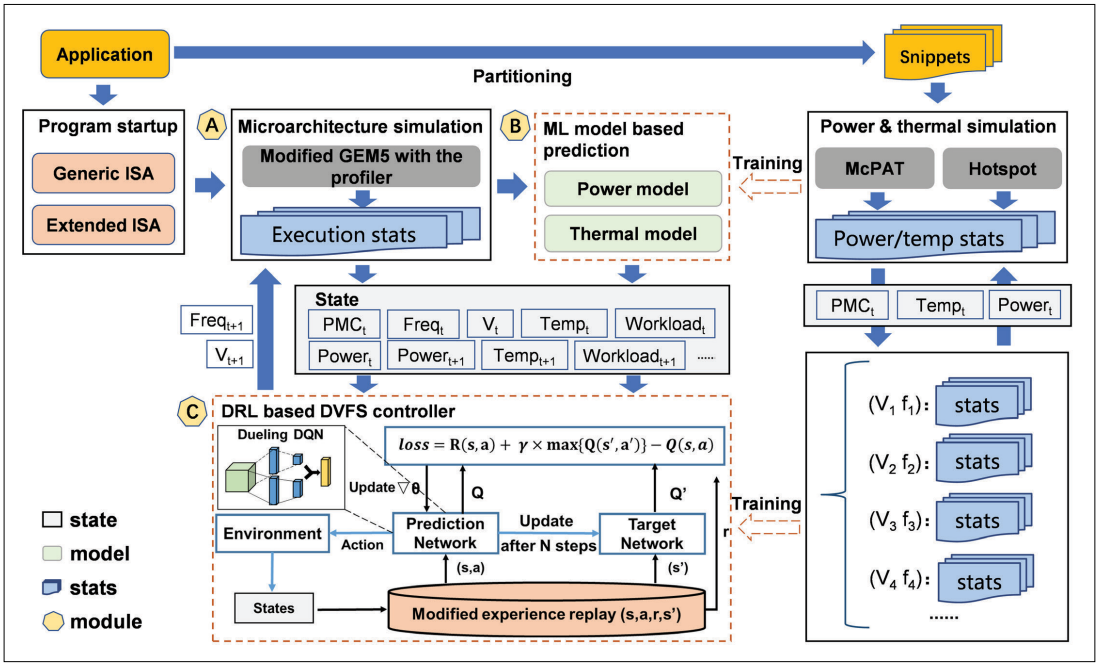

Figure 2. Overview of the proposed DVFS design and simulation framework.

different supply voltage and clock frequency settings (details in the "Microarchitecture simulation" section). As the framework supports breakpoint execution with an embedded profiler, we can evaluate the application binaries of interest and investigate the execution behavior of a particular set of models and controllers.

- The ML model-based prediction module utilizes ML models to estimate and predict the power and temperature change (details in the "ML model-based prediction" section). Instead of using tools like McPAT and Hotspot, the lightweight ML models can provide much faster estimation/prediction with very good accuracy to facilitate microarchitectural profiling and DVFS controlling.
- The DRL-based DVFS controller module takes the outputs of the last two modules to decide the appropriate supply voltage and clock frequency pair (details in the "DRL-based DVFS controller" section). A DRL-based controller is used in the framework as in Figure 2 but can be replaced by other management policy or controller for DVFS design space exploration.

The tool chain of the proposed DVFS design and simulation framework in Figure 2 leverages three existing tools. GEM5 is modified to profile the multiprocessors architecture and support the switching among different supply voltage and clock frequency pairs [9]. McPAT is enhanced to provide more efficient energy profiling and used as the training tool to extract a lightweight power model [10]. Similar to McPAT, a thermal simulator Hotspot is used for thermal modeling [11]. However, it is noted that the two tools are just used to provide the reference values for the models, while the framework is applicable to other power/thermal simulators. After all the models are ready, the proposed framework then provides designers with a unified interface to evaluate the pros and cons of different management policies under different scenarios. In the following, we will use an ARM v8-A Cortex-A72 architecture (with 48 kB L1 iCache, 32 kB L1 dCache, and 2 MB L2 cache) as the underlying system platform. It can work with four supply voltage and clock frequency pairs: 0.6 V–0.6 GHz, 0.8 V–0.8 GHz, 1.0 V–1 GHz, and 1.2 V–1.2 GHz. A Ubuntu operating system with a Linux kernel version 4.14 is boosted on GEM5, supporting the executions of different benchmarks.

Microarchitecture simulation

At the application level, our framework can take any binary compiled from GEM5-compatible general-purpose or specialized compilers to support generic and extended ISAs [9]. The benchmark binaries are fed to the modified GEM5 with specific probes to extract the desired statistics information. By setting up the full-system mode in GEM5, we further employ a run-time agent to monitor and adjust the supply voltage and clock frequency pair feedback from the DVFS controller module. Multiple enhancements are provided to GEM5 to enable its profiling and DVFS adjustion capabilities:

- GEM5 is augmented with a profiler collecting the necessary execution statistics from the performance monitor counters (PMCs) within a predefined period. Such information covers the pipeline execution for each instruction, triggered functional units, and memory access statistics.
- A workload (or CPU utilization) prediction model is embedded into GEM5. We here employ support vector regression (SVR) with Gaussian kernels for tradeoff between accuracy and complexity. While SVR workload model can achieve high accuracy for heavy workloads, it also averages out the random fluctuations, acting as a natural noise filter to ensure good stability.
- A supply voltage and clock frequency adjustion agent is embedded into the Linux system. It stalls the system every t_{dvfs} second and checks whether the DVFS controller module provides any feedback. Once the updated supply voltage and clock frequency pair is received, the agent updates the system settings in GEM5 and then revokes the system to run under the updated supply voltage and clock frequency.

Through re-configuring the Linux kernel to support the aforementioned enhancements, the modified GEM5 is then able to profile the key statistics, predict the future workload, and adjust the supply voltage/clock frequency for the underlying system.

ML model-based prediction

The power/thermal estimation and prediction is indispensable for power management, as the on-die sensing and processing typically has limited spatial and temporal resolutions, and consumes nontrivial hardware resources. Thus, it is necessary to build

an accurate lightweight model that incorporates sufficient physical details for power and temperature prediction.

Power modeling

Without loss of generality, the power model can be written as

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{leakage}} = aCV^2 f + VI_{\text{leakage}} \quad (1)$$

where a is the activity factor, C is the switching capacitance, V is the supply voltage, f is the clock frequency, and I_{leakage} is the leakage. Instead of conducting gate-level analysis, we here use the microprocessor PMCs to estimate the processor activity and then predict the power consumption, such as the number of instructions executed, L1 cache access, memory hit and miss, and so on, within a period of time Δt. Thus, (1) can be further written to the following to estimate the power at the kth time stamp:

$$P_{\text{total}}^{k} = V^2 f \times \sum_i a_i \times \text{PMC}_i^k + VI_{\text{leakage}} + b \quad (2)$$

where a_i and b are the coefficients of the model, the superscript k denotes the kth time stamp, and the subscript i denotes the ith PMC.

However, for large processors such as ARM A72, it may still incur nontrivial computational overhead using the model in (2) by accounting for all the PMCs in power modeling, which may have over 100 counters. Thus, we need to reduce the dimension of the model by only using the parameters highly related to the processor activity and removing collinearity. We sort the criticality of each PMC according to its correlation with the dynamic power and then use variance inflation factor (VIF) as the measure to remove the redundant parameters with collinearity. Eventually we can achieve a compact model using only six PMCs, that is, the number of cycles, retired instruction, L1 cache access, L2 cache access, memory access, and mispredicted branch. The coefficients in (2) can be obtained through SVR or other ML models using the training data. Such a compact model not only works well for simulations on GEM5, but also has good performance when embedded and executed on a development board. We can employ a similar model as in (2) to predict the power at the next time stamp, that is, $k + 1$th time stamp

$$P_{\text{total}}^{k+1} = V^2 f \times \sum_i \hat{a}_i \times \text{PMC}_i^k + VI_{\text{leakage}} + \hat{b} \quad (3)$$

where $\hat{a}_i$ and $\hat{b}$ denote a different set of coefficients from a_i and b in (2).

Thermal model

While processor temperature is closely related to power, the thermal response is much slower with a time constant of millisecond to second. While we need a second-order model to capture such thermal effect, it is computationally impossible to embed a finite-difference method-based model for DVFS control. Thus, we here propose a data-driven second-order model to characterize the thermal profile. For a processor with n modules, we can always have the following thermal formulation:

$$\mathbf{C_T} \frac{d\mathbf{T_{1\times n}}(t)}{dt} = -\mathbf{G_T}\mathbf{T_{1\times n}}(t) + \mathbf{P_{1\times n}} \quad (4)$$

where $\mathbf{T} = [T_1, T_2, ..., T_n]$ is an $1 \times n$ vector for the temperatures of n modules, $\mathbf{P_{1\times n}}$ is the power consumption for the n modules, and $\mathbf{C_T}$ and $\mathbf{G_T}$ are the thermal capacitance and conductance matrices for the system, respectively. It is known to be difficult to directly measure thermal conductance and capacitance, which varies with floorplan, heat sink geometry, and material properties. Thus, here we directly measure the power and thermal at different time stamps through sensory or simulation and then formulate a time series system for thermal estimation and prediction. Through Back–Euler discretization, the system can be written as

$$\mathbf{M_T}\mathbf{T_{1\times n}}^{k+1} = \mathbf{C_T}/\Delta t_T \mathbf{T_{1\times n}}^{k} + \mathbf{P_{1\times n}}^{k} \quad (5)$$

where the superscript k denotes the kth time stamp, $\mathbf{M_T} = \mathbf{G_T} + \mathbf{C_T}/\Delta t_T$ is an $n \times n$ matrix, and Δt_T is the time interval to monitor the temperature. By moving $\mathbf{M_T}$ to the right hand, we can simplify the formulation to the following time series:

$$\mathbf{T_{1\times n}}^{k+1} = \mathbf{A_T}\mathbf{T_{1\times n}}^{k} + \mathbf{B_T}\mathbf{P_{1\times n}}^{k} \quad (6)$$

where $\mathbf{B_T} = \mathbf{M_T}^{-1}$ and $\mathbf{A_T} = \mathbf{M_T}-1\mathbf{CT}/\Delta t_T$. Since the temperature is slowly varying both temporally and spatially, it is unnecessary to maintain a high-density monitor array or high sampling rate. Then, similar to power estimation and prediction, we can use the

training data to learn the coefficients in (6) for thermal estimation and prediction.

Adaptive scheme for model adjustion

Modern processors often come with a few (but limited) thermal, power, and noise sensors. While DVFS management cannot fully depend on those sensors due to the overhead and latency to take measurements, we actually can use the sensors to calibrate and improve the model robustness. To account for the underlying environment and setup variations when deploying the models in a processor, we here propose a scheme to adaptively adjust the model parameters when subject to environmental or setup changes. Kolmogorov–Smirnov test (K–S test) [12] is employed to justify the similarity between the proposed model and actual measured/sensory data. If the K–S test fails, we will update the underlying models by incorporating the latest measurement/sensed data. For example, at a particular time stamp, the K–S test is employed to test the fitness of predicted thermal series and the measured thermal data. The K–S test uses the maximum difference between the two cumulative distribution curves as the test statistics

$$D = \max_{l=1}^{N} \left| F_0\left(x_l\right) - F_{\text{data}}\left(x_l\right) \right| \tag{7}$$

where N is the number of data points, F_0 and F_{data} are the theoretical and actual cumulative distribution of the thermal series, respectively, and x_l is the lth point in the series. These statistics are used to describe the difference between the two sets of data. If D is greater than a predefined threshold, it indicates that the trained thermal model deviates from the actual measurements and hence needs update.

DRL-based DVFS controller

As shown in Figure 2, we employ a DRL-based controller to manage the system. It is noted that the proposed DVFS design and simulation framework is applicable to other controlling mechanisms. To achieve the desired efficiency and flexibility, we cannot simply employ the widely used Q-learning algorithm for DRL [5] [also known as deep Q-learning (DQN)], as the Q-learning algorithm demands discrete states for lookup table and incurs a large table for DVFS. The poor searching performance in such a large Q-table inevitably degrades the benefits brought by DRL.

Thus, in our framework, we modified the prior DQN algorithms to address the concerns of deploying DRL for power and thermal management [5]. We first converted the underlying Q-table to a function and then combined it with the prioritized experience replay to solve the inherent overestimation and computational overhead in Q-learning. As shown in the module *C* of Figure 2, during training, the algorithm reads the states of environment and takes actions to maximize the cumulative reward. During the execution, a lookup table is employed to replace the typical forward propagation for speed up and power saving. In the following, we briefly elaborate the algorithm, state, action, and reward of the proposed DRL framework:

- *Algorithm*: As shown in Figure 2, the proposed DRL training employs a priority experience replay to avoid data unbalance and speed up training, that is, adding priority weights to the data to determine the sampling probability. A priority weight function is

$$w_i = g_i \times s_i / s_{\text{total}} \times r_i \tag{8}$$

  where s_{total} is the total number of configurations, s_i is the number of occurrences for the ith configuration, g_i is the gain parameter that changes the shape of the distribution, and r_i is the reward for configuring i. This priority weight can increase the priority for the data that appear less frequently, thereby speeding up training.
- *State*: The states in Figure 2 are critical to the management strategy. Here, we select ten terms as the input states, including the six PMCs listed in the "ML model-based prediction" section voltage, frequency, temperature, and power consumption.
- *Action*: To manage both power and thermal, we need to select different supply voltage and clock frequency pairs as actions according to the states of the system.
- *Reward*: Reward directly affects the quality of the learned strategy, which is built upon the performance, power, thermal, and even the penalty of DVFS switching [5]. To distinctively differentiate the awards for different program snippets, we can define an adaptive parameter a_r as below that

punishes the deviation from the desired execution time

$$\alpha_r = \alpha_r + \text{IPC} \times \frac{t - t_0}{t_0} \qquad (9)$$

where IPC is the instructions per cycle when the program snippet is executed, t is the execution time, and t_0 is the desired time for program execution.

Once invoked, the proposed DVFS control module runs periodically to decide whether to adjust the supply voltage and clock frequency pair. In DRL, the forward propagation consumes significant resources (both computation and energy) for calculation. Thus, we discretize the mapping function of forward propagation and then replace it with a lookup table for speed and power improvements.

Experimental results

Training

We employ multiple learning models in the framework. The model quality may be impacted by the sufficiency of data. As shown in Figure 2, we partition the benchmarks to program snippets to improve the data sufficiency and coverage. The snippets are randomly selected from both SPEC CPU2006 and PARSEC3.0 benchmarks by running part of the benchmarks, each consuming several hundreds to thousands of cycles under different supply level and

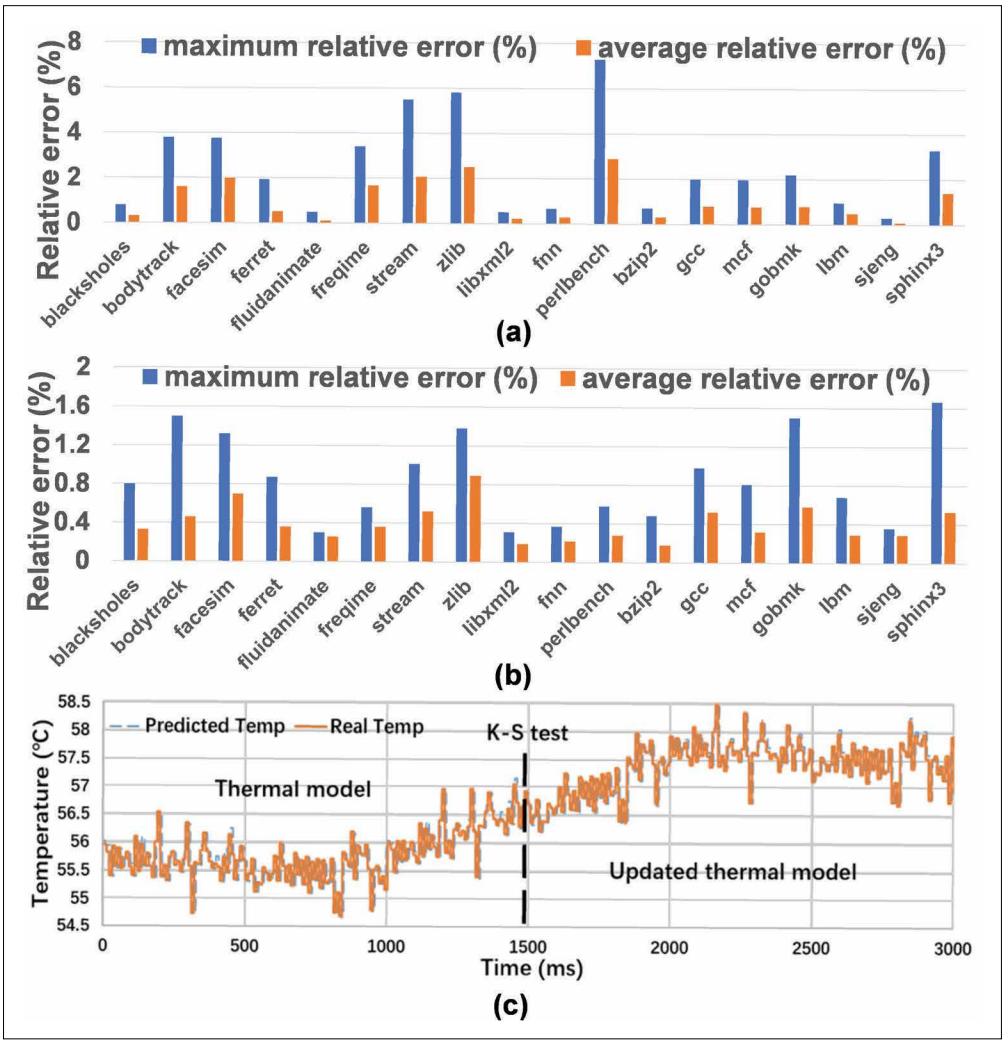

Figure 3. (a) Maximum and average relative error of the power model. (b) Maximum and average relative error of the thermal model. (c) Proposed adaptive adjustion using the K–S test.

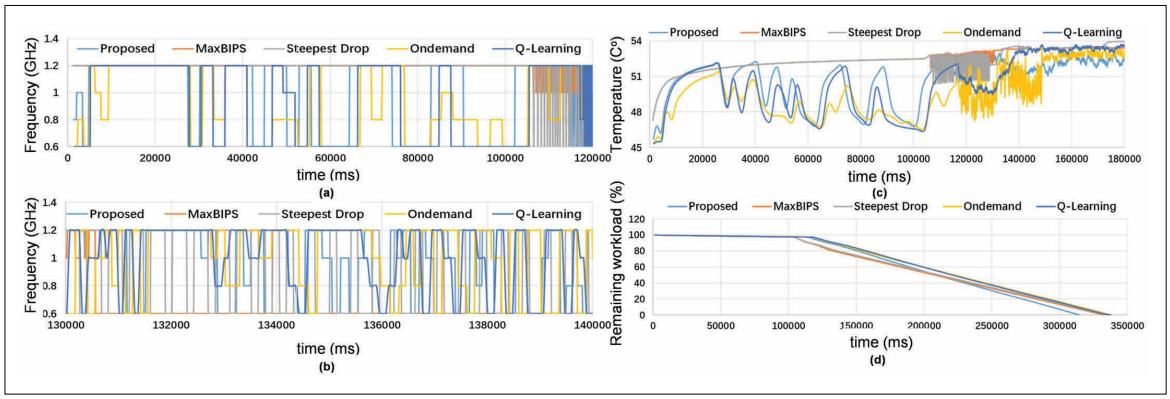

Figure 4. Frequency comparison among the proposed framework and the other methods: (a) before and (b) after reaching the thermal limit; (c) performance and (d) temperature comparison among the proposed framework and the other methods.

clock frequency pairs. The unselected benchmarks are used for evaluation and design space exploration. With GEM5 extracting the PMC and workload statistics, McPAT and Hotspot are then used to obtain the power and thermal information for the snippets. The power and thermal estimation/prediction models can then be trained with such statistics and power/thermal references. The DRL model is a five-layer neural network, which uses the same benchmarks for training as the power/thermal models.

Power and thermal modeling

We first studied the accuracy of the proposed power and thermal modeling in the framework. The ML-based power and thermal models were obtained using (2) and (6). As shown in Figure 3a, for all the 18 benchmarks, the power model can achieve consistently good accuracy, with a maximum relative error 0.5%–7.3% and an average relative error of only 0.1%–2.9%. Similarly, as shown in Figure 3b, the thermal model can achieve even higher accuracy, with a maximum relative error of 0.3%–1.7% and an average error of 0.2%–0.9%. Figure 3c shows the impact of the proposed adaptive adjustion scheme. The pretrained thermal model starts with 55° centigrade, which shows an average relative error below ~1%. However, as the temperature increases, the prediction accuracy drops with a maximum relative error of ~1.5%. The K–S test is carried out at 1,500 ms and finds out that there is a deviation between the model and actual measurement. Thus, by incorporating the measured data to improve the model, the maximum relative error is reduced back to ~1.2%.

DVFS control

We can use the proposed framework to explore different DVFS management methods with bzip2 benchmark used for illustration purpose, which is a data compression program in SPEC CPU2006. In addition to the proposed DRL approach, we also implement other four commonly used methods for evaluation: 1) *MaxBIPS* is a greedy algorithm sweeping all the voltage–frequency pair to select the pair that maximizes the performance [6]; 2) *Steepest Drop* uses a heuristic optimization to select the voltage–frequency pair that maximizes the performance under the power budget [7]; 3) *Ondemand* is the default management method in Linux that selects the voltage–frequency pair according to the processor utilization; and 4) *Q-Learning* selects a voltage–frequency pair from a precomputed Q-table [5]. Figure 4 summarizes the comparison results among the five methods. Figure 4a and b compares the clock frequency selection among the five methods before and after reaching the thermal limit. After the processor reaches the thermal limit, the five methods respond with different strategies, as shown in Figure 4b. MaxBIPS and Steepest Drop continue selecting the largest voltage–frequency pair, causing repeated switching between the highest and lowest voltage–frequency pairs. Ondemand, Q-Learning, and the proposed DRL approach select lower frequencies based on the processor states to prevent thermal violation. As the DVFS switching cost is accounted for in our DRL approach, the approach has fewer frequency switching than the other two methods, which eventually helps the overall performance. As shown in

Figure 4c, the proposed DRL approach can achieve performance improvement of 6.8%, 7.3%, 5.3%, and 6.2%, when compared to Q-Learning, Ondemand, MaxBIPS, and Steepest Drop, respectively. Finally, Figure 4d demonstrates the temperature dynamics using the five methods. It is noted that the proposed DRL oscillates less frequently than the other methods, which reduces performance degradation due to the switching. Without the proposed framework, it is difficult to find out such dynamics details among different methods.

We proposed a DVFS design and simulation framework using ML models for DVFS design space exploration. The framework leverages the existing tool chains and consists of three key modules, microarchitectural simulation, ML model-based prediction, and DRL-based DVFS control. The proposed framework can evaluate different benchmarks and different management methods to profile the underlying dynamics of power, thermal, and performance. ■

Acknowledgments

This work was supported in part by the National Key Research and Development (R&D) Program of China under Grant 2018YFE0126300, in part by the NSFC under Grant 61974133 and Grant 62034007, and in part by Zhejiang Provincial Key R&D Program under Grant 2020C01052.

References

[1] S. Hajiamini et al., "A dynamic programming framework for DVFS-based energy-efficiency in multicore systems," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 1, pp. 1–12, Jan. 2020.

[2] C. Zhuo et al., "Noise-aware DVFS for efficient transitions on battery-powered IoT devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 7, pp. 1498–1510, Jul. 2020.

[3] J. Charles et al., "Evaluation of the Intel Core i7 turbo boost feature," in *Proc. IISWC*, Oct. 2009, pp. 188–197.

[4] G. Bhat et al., "Algorithmic optimization of thermal and power management for heterogeneous mobile platforms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 3, pp. 544–557, Mar. 2018.

[5] S. Pagani et al., "Machine learning for power, energy, and thermal management on multicore processors: A survey," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 1, pp. 101–116, Jan. 2020.

[6] C. Isci et al., "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *Proc. MICRO*, Dec. 2006, pp. 347–358.

[7] A. Madureira et al., "Negotiation mechanism for self-organized scheduling system with collective intelligence," *Neurocomputing*, vol. 132, pp. 97–110, May 2014.

[8] R. Teodorescu and J. Torrellas, "Variation-aware application scheduling and power management for chip multiprocessors," *ACM SIGARCH Comput. Archit. News*, vol. 36, no. 3, pp. 363–374, Jun. 2008.

[9] N. Binkert et al., "The Gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, May 2011.

[10] S. Li et al., "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. MICRO*, 2009, pp. 469–480.

[11] W. Huang et al., "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[12] J. Millard and L. Kurz, "The Kolmogorov–Smirnov tests in signal detection (corresp.)," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 341–342, Apr. 1967.

Cheng Zhuo is a professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. His research interests include 3-D integration, hardware acceleration, and power and signal integrity. He is a Senior Member of IEEE and a Fellow of IET.

Di Gao is pursuing a PhD with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. Her current research interests include power management and low-power design.

Yuan Cao is pursuing a master's with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. His research interest is power management.

Tianhao Shen is pursuing a master's degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. His research interest is power integrity.

Li Zhang is a postdoctoral researcher at Zhejiang University, Hangzhou 310027, China. His research interest is low-power design. Zhang has a PhD from Huazhong University of Science and Technology, Wuhan, China.

Jinfang Zhou is an associate professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. Her research interests include power and signal integrity.

Xunzhao Yin is an assistant professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. His research interests include low-power circuit/architecture designs with both CMOS and emerging technologies. He is a Member of IEEE.

■ Direct questions and comments about this article to Cheng Zhuo, College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China; e-mail: czhuo@zju.edu.cn.