Copyright © 2018 American Scientific Publishers
All rights reserved
Printed in the United States of America

Journal of
Low Power Electronics
Vol. 14, 460–474, 2018

SmartDPM: Machine Learning-Based Dynamic Power Management for Multi-Core Microprocessors

P. D. Sai Manoj[1,*], Axel Jantsch[2], and Muhammad Shafique[3]

[1] *Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 22030, USA*
[2] *Institute for Computer Technology, TU Wien, Vienna, 1040, Austria*
[3] *Institute for Informatics, TU Wien, Vienna, 1040, Austria*

(Received: 3 September 2018; Accepted: 18 September 2018)

To address the power management challenge in multi-core microprocessors, we present a lightweight machine learning based dynamic power management (*SmartDPM*) scheme in which the voltage–frequency levels of the cores are dynamically adjusted along with online learning based workload prediction in an observer-controller loop. To enable scalability, our *SmartDPM* employs a per-application autonomous power management policy, in which online machine learning principles are employed for predicting the workload and capturing sporadic variations under the constraints of *accurate yet lightweight*. Further, applications are assigned appropriate voltage–frequency level towards an efficient power management. The learning helps in dynamically reducing prediction error. Compared to the non-DVFS implementation, *SmartDPM* achieves nearly 35% power saving and nearly 15% higher power savings on average compared to the existing machine learning based power management schemes for a microprocessor with up to 32-cores.

IP: 129.174.252.250 On: Tue, 20 Aug 2019 22:47:13
Copyright: American Scientific Publishers
Delivered by Ingenta

Keywords: Multi-Core Microprocessor, Power Management, Machine Learning, Dynamic Voltage Frequency Scaling, Prediction, Control Theory, Online Learning, Scalability, Energy Efficiency.

1. INTRODUCTION

Multi-core microprocessors evolved as a result of scaling down of transistor nodes and innovations in computer architecture designs. Increase in the number of cores in multi-core microprocessors made it feasible to enjoy the benefits of multi-threaded processing such as higher throughput, and many more performance improvements. However, this improved performance and benefits comes at the cost of increased power consumption. Nevertheless, the microprocessor comprises of multiple units, cores are one of the major power consuming units, encountering the difficulty to meet the power budget demands especially under unpredictable run-time scenarios of varying workloads for different concurrently executing applications.[1–14] This calls for an efficient and adaptive power management in multi-core microprocessors.

Dynamic voltage and frequency scaling (DVFS)[1,13,15–21] and dynamic power management (DPM)[22,23] have proven to be effective power management techniques for microprocessors. Dynamic power management refers to selective shutdown of system components; while DVFS refers to scaling down of voltage and frequency levels for under-utilized cores, thereby reducing the dynamic power consumption. DPM though efficient in terms of power savings, shutting down and turning-on a core causes additional energy and latency penalty.[3,24–27] However, this is worthwhile when the idle interval is longer than a certain threshold.[26,28–30] In this work, we consider using the DVFS methodology for power management in multi-core microprocessors, because of its lower overhead and wider applicability, especially for applications which are computationally expensive. Furthermore, our work is orthogonal and can be employed in systems that perform both DVFS and DPM, like current multi-core microprocessors. The characteristics of the workload running on core(s) have a direct impact on the power consumption of a microprocessor. As such, considering the facets of the workload to perform DVFS is effective.[13,16,30–35] Power management (DVFS) can be performed considering different parameters such as worst-case execution time of the task,[36] temperature,[37] workload behavior.[34,35] Predicting or knowing

*Author to whom correspondence should be addressed.
Email: saimanoj.p.2013@ieee.org

 1546-1998/2018/14/460/015 doi:10.1166/jolpe.2018.1576

the workload characteristics under unpredictable hardware factors (such as cache misses, on-chip network contention, and so on) aid in effective DVFS at run-time. The desired properties of the predictor for predicting the workload behavior and to perform DVFS could be outlined as follows: accurate yet lightweight in nature;[28] effectively capture sporadic variations; adapt to the variations; continuous self-optimization, as self-optimization is important to learn over time; and low complexity with reduced number of computations. The rationale to have a lightweight predictor is to avoid additional computational overheads caused and alleviate the processing delays in the prediction. Prediction with the aid of machine learning has been proven to be effective along with capturing the workload behaviors and underlying variations.[38,39] Along the same lines, we also adapt the machine learning to predict the workload(s) characteristics in this work. The considered workload characteristic for DVFS in this work is the power trace of the workload(s).

1.1. Motivational Case Study

The challenge of adapting to workload variations can be effectively addressed by learning workload behavior. As large number of relevant machine learning algorithms and heuristic approaches are available for prediction, it is critical to choose a suitable technique that facilitates minimizing the prediction errors by learning the underlying factors to predict the sporadic variations. The selection criteria for the predictor is to be fairly accurate, robust and lightweight in terms of computations and complexity. In order to select an appropriate learning technique, we perform a case study and evaluate the performance of different prediction algorithms. This case study is performed in Matlab. Workload power traces are obtained from Sniper-Sim,[40] with applications running on a single-core microprocessor of Nehalem architecture (In-depth details of experimental setup is presented in Section 6.1). Average RMSE for online learning based linear prediction, neural networks and support vector machine based prediction is indicated in Figure 1. The *Y*-axis represents the RMSE error and the executed benchmark is marked on *X*-axis of Figure 1.

Comparison of root-mean square error (RMSE) for SPLASH-2 benchmarks' workload (power trace) prediction with different prediction methods is illustrated in Figure 1. The employed prediction methods for comparison are: modified covariance based prediction (Mod. Cov.), linear prediction with online learning (Linear Pred.), traditional neural networks (Neural N/W) and support vector machine (SVM). For the linear predictor, an order of 4 is used. The order is choosen by experimenting the predictor with different orders and also the complexity involved. Similarly, for neural network based prediction, a single-hidden layer with 10 nodes is employed. Levenberg-Marquardt backpropagation based training is used for neural network.

IP: 129.174.252.250 On: Tue, 20 Aug 2019 22:47:18
Copyright: American Scientific Publishers
Delivered by Ingenta

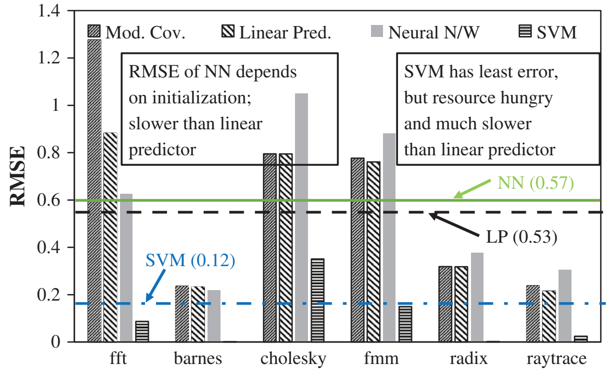


Fig. 1. RMSE of workload power trace prediction using covariance, linear prediction, neural network and support vector machine techniques (average RMSE is marked); *X*-axis represent benchmark.

From Figure 1, one can observe that the neural network has lower RMSE for some of the benchmarks and can be further reduced by modifying the neural network architecture. However, it is experienced during the simulations that the accuracy or RMSE of the neural network highly varies depending on the initial state i.e., initialization of the weights. Online learning based linear predictor outperforms modified covariance in terms of average RMSE. Support vector machine (SVM) based regression outperforms both neural network and online learning based linear predictor implementations and has a minimal average RMSE. However, the runtime for SVM based prediction is more than 10× higher than the neural networks and linear prediction. The complexity of a technique or a method can as well be observed experimentally from its runtime, as the involved computations determine the runtime of a technique. The runtime for Mod. Cov., Linear Pred., Neural N/W, and SVM is 0.026 s, 0.012 s, 0.021 s, and 0.93 s, respectively.

As the two main desired characteristics for the predictor in this work are: (a) ability to learn workloads, and capture their behavior—this can be observed from the RMSE; (b) lightweight (low complexity) and fast learning predictor—runtime is one of the factors that represents this. The linear predictor with online learning fairly performs well and has smaller average RMSE. Additionally, the linear predictor with online learning requires less computations and is faster compared to SVM and neural networks. Depending on this analysis and to satisfy the constraint of *accurate yet lightweight* prediction, we have chosen linear predictor with online learning for the purpose of predicting workloads in *SmartDPM*. It needs to be noted that the RMSE might be different if the parameters of the predictor are modified such as prediction order, number of layers in neural network. However, more number of layers in a neural network will increase the complexity, and cost for employing it for individual core will be significant.

1.2. Associated Research Challenges

Other than the basic challenges such as voltage–frequency level changing latency, and accuracy, some of the main challenges to be addressed for efficient power management using workload prediction are:

- *Learning and adapting to workload variations*: Workload(s) of an application often varies with time, underlying factors and computations can result in sporadic variations in workload. As such, it is of vital importance to capture these variations and learn the behavior of workload, adapt to it before performing the prediction of workloads.
- *Robustness of the prediction*: It is critical to perform prediction that is robust to the variations such as sporadic behaviors in the workload, and initial value in the prediction. Additionally, a small computation time is often desired.

1.3. Contributions

We propose an online learning based adaptive dynamic power management (*SmartDPM*) technique for multi-core microprocessors. *SmartDPM* performs the prediction of workload in an observer-controller loop manner during run-time with online learning. The utilized observer-controller loop predictor continuously keeps the weights updated based on the prediction errors, thus ensuring both accuracy and robustness to the errors. Once the prediction of the workload is carried out, corresponding voltage and frequency levels are assigned towards effective power management. The assignment is based on the utilized power model. *The main contributions of this work can be outlined as follows*:

- A run-time adaptive dynamic power management for multi-core microprocessor with online learning based prediction.
- A run-time workload prediction with an accurate yet lightweight online learning observer-controller loop.
- Based on the learned workload characteristics and the predicted workload characteristics, voltage–frequency pairs are assigned to the application.

The proposed *SmartDPM* achieves a power saving of nearly 35%, 15%, and 14% compared to non-DVFS, approaches similar to space-time multiplexing,[8] and SVM[38] on average for a microprocessor with up to 32-cores running SPLASH-2 and Parsec benchmarks.

1.4. Paper Organization

The rest of this paper is organized as follows. A review of relevant existing works on power management is presented in Section 2. The system model and the application model are presented in Section 3. Section 4 presents the system architecture for *SmartDPM* with the problem formulation. The prediction technique and clustering used for *SmartDPM* are described in Section 5. Simulation results and comparisons are presented in Section 6 with conclusions drawn in Section 7.

2. LITERATURE REVIEW

In the past few years, some works started exploring learning based techniques for prediction under a *centralized predictor*. Regression[11, 13, 41, 42] is one of the most popular methods to predict the workloads, which fits a polynomial to the measured/observed values and extrapolate future values. Regression of different metrics such as LLC miss or hit rates, power trace of the workloads, memory accesses are utilized to perform DVFS. A comprehensive survey on machine learning based power management is presented in Ref. [1]. We review some of the relevant works below.

In Ref. [34], the workload for the next frame is estimated using linear-in-parameters model, in which the output for the next frame is estimated as a linear combination of previous workloads (frames) and a number of system parameter values. Similarly, a linear regression is adopted in Ref. [43] to estimate the workloads and perform DVFS accordingly. A gradient descent method based updating frequency by learning the workload using linear regression considering the observations from performance counters, power sensors and measured latencies is proposed in Ref. [44]. A space-time multiplexing (STM) based power management with auto-regressive moving average (ARIMA) for predicting the workloads and a singular value decomposition for clustering and voltage–frequency level assignment is proposed in Refs. [8, 13]. Different techniques to reliably predict workloads such as Last-value predictor, history table, and ARIMA are proposed in Ref. [38] along with an qualitative analysis. Some methods also make use of other workload characteristics such as deadlines, memory accesses and so on to perform DVFS.

In Ref. [36], a linear regression is used to predict memory accesses per cycle and CPU cycles per instruction (CPI). Based on the ratio of predicted CPI with on-chip access and overall CPI, frequency scaling is performed for power management. Similarly, in Ref. [45], a deadline for each task or application is considered to perform power management. Based on the slack time generated in the current time slot and the worst case execution of the task, the operating frequency for the application or task is scaled. In addition to workload timing behavior, memory access patterns are as well utilized to perform power management. A voltage scaling approach based on the obtained dynamic events such as cache hit or miss rate and memory-access counts at runtime from performance modeling unit (PMU) are used to determine the optimal frequency for performance constraint is proposed in Ref. [46]. Updating frequency levels at variable intervals for power management is proposed in Ref. [19]. This method makes use of idle time between two consecutive cycles and increases or decreases frequency levels based on that. Two registers and counters are employed to keep track of idle times and store the values for comparing the intervals. Apart from use

Copyright American Scientific Publishers
Delivered by Ingenta

of regression techniques using linear regression, wavelet-based power management scheme making use of wavelet transform to predict the signal and further based on the predicted values, a threshold dynamic power management is carried out.[47] The characteristics of workload used for power management greatly varies. However, it is clearly evident that regression in one form or other are considered to estimate the workloads to perform power management. It needs to be noted that the complexity and achieved accuracy varies with the applied regression technique and with application.

On the other hand, advancements in machine learning enabled adapting machine learning techniques for prediction purpose.[48] A Bayesian workload predictor with classification and policy generation is proposed in Ref. [49]. This uses iterative loops with dynamic programming with cost function objective. A model-free reinforcement learning for dynamic power management with Bayesian predictor for workload estimation is proposed in Ref. [6]. Based on the predicted workload and using reinforcement learning, power management is carried out. In a similar manner, a Q-learning based approach considering the clock frequencies, lowest frequency to meet deadline, CPU utilization rate for the current task as states with tuning as frequency and voltage as the action is proposed in Ref. [50]. The Q-learning power management is also presented in Refs. [16, 31, 51–53]. A two level hierarchical power management scheme is proposed in Ref. [54]. This method is performed at two levels: component-level local power manager and system-level global power manager. The component-level power management follows a pre-specified power management policy and is fixed; whereas the system-level power manager employs temporal difference learning on semi-Markov decision process as the model-free reinforcement learning technique, and it is specifically optimized for a heterogeneous application pool. In addition to traditional learning approaches, deep learning is also employed for power management in multi-core and many-core microprocessors. A deep learning approach for workload prediction and adaptive power scaling is proposed in Ref. [11]. Here, statistical relationships are exploited on the data obtained from hardware counters to predict the periods of low instruction throughput and the frequency, voltage are decreased to save power. A hierarchical sparse coding was adopted to capture complicated signature patterns over time, which has shown prediction performance improvement compared to regression and heuristic approaches. This sparse coding is succeeded by support vector machine (SVM) regression. Based on the true positives and accuracy, DVFS is performed. A support vector regression based regression followed by classification is used for DVFS in Ref. [38]. A comparison of neural networks and regression methods is presented,[39] which concludes that the neural networks outperform regression based prediction when the data used for training is smaller, whereas both of them perform similarly when fair amount of training data is available. As such, it can be observed that power management (DVFS) is performed with the aid of various machine learning techniques ranging from semi-supervised learning (reinforcement learning) to deep learning and supervised learning techniques. Learning based approaches are more robust and efficient, however, the resources and computation time it requires is high. Furthermore, most of the existing machine learning based works are supervised with offline learning, whereas the heuristics based predictors based works lack accuracy and self-learning.

2.1. Novelty Over State-of-the-Art Techniques

In addition to the trade-off analysis presented in Section 1.1, we present the similarities and deviations from the existing power management techniques in the literature. Similar to most of the existing techniques, our proposed power management technique *SmartDPM* also follows two-step procedure: prediction and voltage–frequency (VF) assignment. The main differences could be outlined as follows: unlike most of the power management schemes which utilizes offline learning, *SmartDPM* performs predication with an observer-controller loop utilizing online learning to capture underlying variations in the workload(s). The utilized observer-controller loop prediction technique is accurate and lightweight, faster compared to other techniques that makes power management feasible during runtime with online learning. This is evidently clarified with the aid of RMSE metric and runtime in Sections 1.1 and 6.3.4.

Copyright: American Scientific Publishers
Delivered by Ingenta

3. SYSTEM MODEL

3.1. Hardware Architecture Model

We consider a homogeneous multi-core processor comprising of N cores, $C = \{C_1, C_2, \ldots, C_N\}$. Due to varying workloads, different cores execute at different frequencies in order to ensure proper execution. There exists a maximum operating frequency level $f_{\max}$ for every possible operating voltage V. The frequencies of a core can be varied between $f_{\min}$ to $f_{\max}$, and the corresponding voltages between $v_{\min}$ and $v_{\max}$. The cores operating at higher VF levels consume more power when executing the application. Furthermore, similar to Ref. [55], we assume that performance of the processor core is higher when running at a higher VF level.

3.2. Application Model

We consider a mixture of single-threaded and multi-threaded applications in this work, and each core executes one thread. In Figure 2, different shades on cores represent different applications running on them. The distribution of applications are not uniform i.e., different applications can run on different number of cores, depending on the number

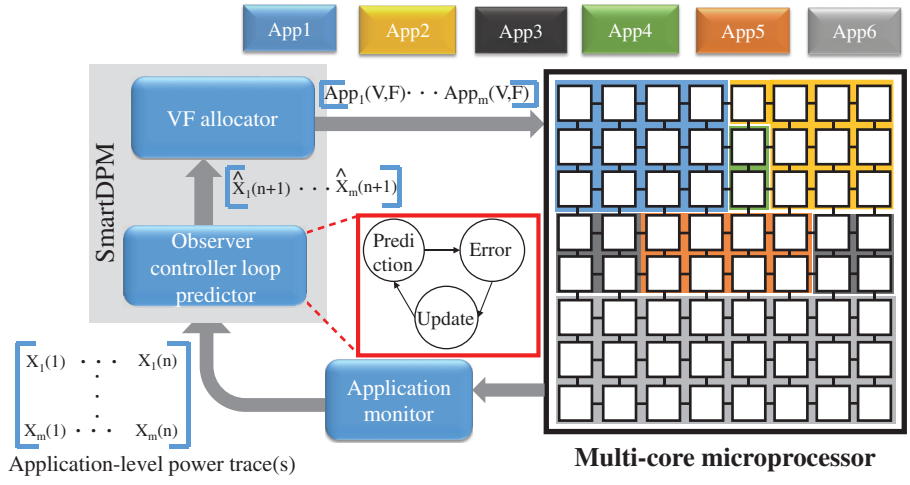


Fig. 2. Overview of proposed *SmartDPM* in a multi-core microprocessor.

of threads. Each of the application comprises of multiple tasks, as such a task τ requires w clock-cycles for execution. From the application perspective, we assume that the applications are multi-threaded and can be executed in parallel, similar to Ref. [3]. Each core can execute one thread. We assume that the total number of threads executing are smaller or equivalent to number of cores. To avoid the associated overheads, especially for multi-threaded applications, we employ per-application power manager rather than system-level or per-core power manager. To obtain per-application power or energy trace, we sum the power traces of the cores on which the application is currently executing.

3.3. Power Model

The total power consumption of a core comprises of static and dynamic power. The static power is dominantly due to the leakage currents and varies exponentially with threshold voltage. The dynamic power consumption is due to the switching activities inside the core. As such, the total power consumption[56,57] when operating at voltage V and frequency f is modeled as

$$P(V,f) = P_{\text{static}} + P_{\text{Dynamic}} = I_0 e^{-V_{\text{th}}/(\eta V_T)} V + \alpha C V^2 f \quad (1)$$

Here I_0 and η are technology parameters; V_T is the thermal voltage; V_{th} is the threshold voltage; α represents the switching activity factors and C is the average capacitance.

4. SYSTEM ARCHITECTURE

4.1. System Overview

We present an overview of the system along with different components and their corresponding models that are utilized in this work. The main components of the *SmartDPM* system are: cores, voltage–frequency regulators and connectivity between cores.

An Intel Xeon microarchitecture[58] core model is chosen to build the system. QuickPath interface is used for point-to-point connection for cores. External components such as DRAM are connected with the help of buses. The system is supported with DRAM memory controller(s) for memory access. In the recent years, there has been a surge of interest to build on-chip integrated switching voltage–frequency regulators.[59–61] To enable the support for power management with *SmartDPM*, standard on-chip voltage–frequency regulators that are used in industrial processor like Xeon X5550 are embedded into the system. The latency involved in changing voltage and frequency levels is less than 2 μs.[58,59] It needs to be noted that the proposed *SmartDPM* is not bound to any specific core architecture, i.e., the proposed work can also be deployed in microprocessors with other (micro-)architectures. The system architecture is described below.

IP: 129.174.25.250 On: Tue, 23 Aug 2016 22:47:34
Copyright: American Scientific Publishers
Delivered by Ingenta

4.2. System Architecture

The system architecture to perform *SmartDPM* based power management is presented in Figure 2. The components used in the system inherits the model description in Section 4.1. The microprocessor comprises of multiple cores with private L1, L2 instruction and data caches and shared L3 cache(s). The microprocessor is equipped with on-chip DRAM controller for DRAM access. In addition to the traditional core and caches, system is equipped with an *Application monitor* unit that monitors the workload statistics of each application executing on the microprocessor core(s). The workload statistics refer to the power traces of the workload when executed on the core(s), which are then provided to the *SmartDPM* unit. Each *SmartDPM* unit performs prediction of the workload in two stages: modeling and optimization in an observer-controller loop fashion. Further, based on the predicted workload a suitable voltage–frequency (VF)

pair is decided and assigned to the core(s) running the corresponding application towards an efficient power management. These voltage–frequency values are set and provided to cores with the aid of on-chip power converters. As continuous prediction adds computational overheads, the prediction is performed at regular intervals of time.

A multi-core microprocessor running different applications, equipped with *SmartDPM* power management is depicted in Figure 2. Shading over cores in Figure 2 represent different applications. An application refers to a program that runs on the core(s) of the microprocessor. For instance, multi-threaded applications such as Parsec 'blackscholes' requires multiple cores to run. As such, to avoid the associated overheads, especially for multi-threaded applications, we employ per-application power manager that autonomously learns and updates the prediction model depending on the variations in the workload; and further assign the voltage and frequency to the core(s) for an effective power management.

4.3. Associated Research Questions

Other than the basic challenges such as voltage–frequency level changing latency, some of the main challenges to be addressed for an efficient power management using workload prediction are:

- *How to learn and adapt to workload variations*? Workload(s) of an application(s) often varies with time. Underlying factors and computations can result in sporadic variations in workload behavior. As such, it is of vital importance to effectively learn the behavior of workload by capturing the underlying variations, adapt the model in case of deviations.
- *How to learn fast*? In addition to learning and adapting to workload variations, the amount of time it needs to learn and adapt also impacts the performance.
- *Robustness of the prediction*? It is critical to perform prediction that is robust to the variations such as sporadic behaviors in the workload, and initial value.

IP: 129.174.252.250 On: Tue, 29 Jan 2019 22:47:43
Copyright: American Scientific Publishers
Delivered by Ingenta

4.4. Problem Statement

The power consumption of a core depends on the workload that it is running. For instance, a core running a computationally intensive application consumes more power compared to the core which executes computationally less intensive and lightweight workload. Based on the posed research questions and the presented system architecture, we define the problem as follows.

In a multi-core microprocessor, the workload(s) needs to be learned along with its underlying variations and further predict the behavior so that the core(s) can be provided with appropriate voltage and frequency levels towards an effective power management under the constraints of achieving a targeted performance. This could be mathematically defined as:

$$\begin{aligned} &\min(P_{\text{Total}}) \\ &\text{S.T.} \quad \text{(i)} \ ||x_i(n) - \widehat{x_i(n)}||_1 < \varepsilon, \quad \varepsilon \to 0 \\ &\qquad \text{(ii)} \ v_j(n), f_j(n) \leftarrow \widehat{x_i(n)} \\ &\qquad \text{(iii)} \ v_L \leq v_{\text{opt}} \leq v_H \\ &\qquad \text{(iv)} \ Thr \geq Thr_{\text{min}} \end{aligned} \tag{2}$$

The main objective of this work is to minimize the overall power consumption P_{Total} without violating the performance requirements. Here, the performance is measured as throughput. The throughput setting is provided externally to the system in this work. This objective has to be fulfilled subject to an accurate prediction of workload(s), as given in (i) of (2); it is desired to have a small prediction error ε between the predicted workload $\widehat{x_i(n)}$ and the actual workload $x_i(n)$ for application i for time instant n; Secondly, based on the predicted workload $\widehat{x_i(n)}$ at a phase n, the an optimal voltage $v_j(n)$ and frequency $f_j(n)$ levels have to be assigned to the core(s) running workload i without overpowering the core(s), as given in (ii) of (2); and the voltage level assigned to the application needs to be sufficient i.e., the core(s) must be supplied with a minimal voltage to keep the application running, as given in (iii) of (2) along with achieving a performance (throughput Thr) better than the lower bound (Thr_{min}). In addition to the above constraints, as mentioned in the previous sections, the power manager should not incur large overheads.

To predict and learn the workload behavior, the prediction needs to be based on the previous inputs (observer). To capture and adapt to the sporadic variations in the workload behavior, a feedback has to be provided to the predictor to update the model. As such, the problem of prediction can be mathematically stated as

$$\widehat{x(n)} = z(x(n-1), x(n-2), \ldots, x(n-p)) \tag{3}$$

where $x(n-p)$ is the workload at $(n-p)$-th time instant, and $z(\ldots)$ denotes the observer-controller loop based prediction function.

5. SMARTDPM: ONLINE LEARNING-BASED DYNAMIC POWER MANAGEMENT

Our proposed multi-core microprocessor power management scheme *SmartDPM* efficiently manages the system power under the constraints of lightweight yet low error in the workload prediction (with the aid of observer-controller loop) and a minimized overhead. This is performed in two steps. In the first step, an observer-controller loop based prediction is performed to learn and predict the workload(s) at an application-level granularity, under the constraints of low prediction error and lightweight. Further, depending on the application and the predicted workload characteristics, the voltage–frequency (VF) pairs are

assigned to the corresponding core(s) that run the workload towards minimizing the overall power consumption under the constraints of performance. Predicting at regular time-intervals is adopted in order to reduce the computational overheads. A detailed description of the whole process is presented below.

5.1. Learning and Prediction of Workload

At a more higher abstract level, observer-controller loop based predictor as a part of *SmartDPM* is shown in Figure 2, enclosed in a zoomed-out rectangle. A more detailed architecture of the observer-controller loop based predictor employed in *SmartDPM* is depicted in Figure 3. As continuous prediction of workload characteristics might cause additional overheads, we predict the workload characteristic, i.e., power trace at regular time intervals. Please note that the workload characteristic here refers to the power trace when the application is run on the core(s). The workload prediction in *SmartDPM* uses a observer-controller loop prediction equipped with an online learning to learn and predict the workload effectively by capturing the underlying variations in real-time. Additionally, the weights of the predictor are updated continuously depending on the prediction error. The prediction mechanism minimizes the prediction errors in a least-square sense. The online learner continuously learns the workloads running on the core(s) and adapts the weights in order to capture the variations, if the prediction error increases. This could be explained in two phases: *modeling phase* (observer) and *optimization phase* (controller to minimize the prediction error).

Phase 1: *Modeling (Observer).* Towards predicting the workloads (at phase-level), the first step is to build a model that effectively represents the workload. A linear model is initially built to estimate the workload at the phase-level and perform prediction. The model is built considering the workload values at previous phases under the constraint of minimizing the prediction errors. The prediction errors aid in adapting the model by learning the variations more effectively and capture the variations effectively. This modeling phase can be seen as learning phase, where the model is built so as to estimate the workload effectively. As shown in Figure 3, the observer monitors the workload

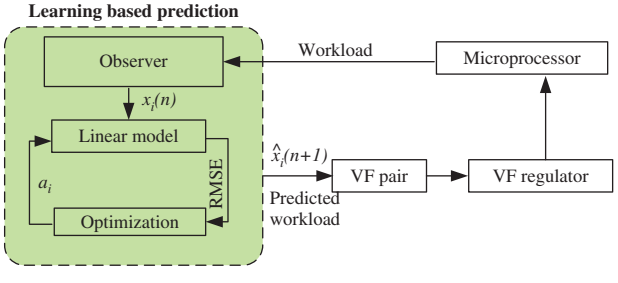


Fig. 3. An observer-controller loop based workload prediction used for *SmartDPM*.

of the applications and provides the observed values to initially build a model. The built linear model for prediction can be given as

$$\widehat{x_i(n)} = a_1 x_i(n-1) + a_2 x_i(n-2) + \cdots + a_p x_i(n-p) + \gamma = \sum_{j=1}^{p} a_j x_i(n-j) + \gamma \quad (4)$$

here a_j, $j = 1, 2, \ldots, p$ are the coefficients, p representing the order of prediction i.e., number of previous samples considered for the prediction; $x_i(n)$ represents the workload at phase n for application i; γ represents the prediction error; and the predicted value is indicated as $\widehat{x_i(n)}$. The utilized modeling and learning phase is linear in nature, and hence faster.

As the workloads may deviate from the built model under sporadic events, it is necessary to update the built linear model with time without overhead. To capture such sporadic variations, the model building phase is succeeded by an optimization phase with an objective to minimize the prediction errors along with updating the built model with time.

Phase 2: *Optimization (Controller).* The model built in the modeling phase is efficient in predicting the workloads based on the trained data. However, workloads could experience sporadic variations during the runtime due to underlying factors compared to the data learned during the modeling phase. In order to capture those variations in the workload and predict more precisely using (4), the model needs to be updated if the workloads change i.e., the coefficients (a_i) need to be tuned if the prediction is not close to the actual values. This is performed by updating the weights in order to minimize the prediction error i.e., to satisfy auto-correlation criterion. As such, the choice of coefficients is made by

$$\sum_{i}^{p} a_i R(j-i) = -R(j) \quad (5)$$

for $1 \leq j \leq p$, and $R(j)$ is the autocorrelation for the signal, given by

$$R(j) = E[x_i(n) x_i(n-j)] \quad (6)$$

where $E[x_i(n)]$ denotes the expected value of workload $x_i(n)$.

This optimization phase can be seen as updating of the prediction model based on the observed prediction errors and deviation of the predicted and actual values. This optimization phase helps to capture underlying factors that contribute to sporadic variations.

5.2. Voltage–Frequency Assignment

Once the workload is efficiently predicted for an application, the corresponding core(s) must be provided with the voltage–frequency (VF) levels that meet the workload requirement. This ensures that the core(s) running

the workload is neither overpowered or underpowered to run the application workload. The VF assignment to the application based on the predicted workload (power) is presented in this section. We chose four VF levels in this work, as the chosen levels support the considered Intel Nehalem microarchitecture.[58] However, the same technique could be extended to more levels, depending on the requirements.

SmartDPM has a set of pre-defined voltage–frequency pairs (VF levels) for which the corresponding providable power is calculated based on model in (1), and based on the predicted workload value the corresponding application is assigned with one of the suitable VF levels. A *Mahalanobis distance* metric based clustering is employed to determine the cluster (pre-defined VF level based on the power model) to which the application will be assigned. The clustering is done by considering the suitability of the predicted power trace for an application and the power that can be supplied when a VF level is assigned. The Mahalanobis distance between two vectors can be given as

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})S^{-1}(\vec{x} - \vec{y})} \tag{7}$$

where $\vec{x}$ and $\vec{y}$ are the vectors between which the Mahalanobis distance is calculated and S is the covariance matrix. In our case, the vectors are the predicted workloads and the formed centroid for four voltage–frequency levels i.e., the power that can be provided when run at particular VF level. The closer the Mahalanobis distance similar the vectors. The application (workload) will be assigned with the VF level to which it has smaller Mahalanobis distance. The same clustering technique could be utilized when considering additional multiple metrics. It needs to be noted that when the performance is not met with the assigned VF settings, the VF of the next higher level are assigned to the application.

The multi-core power management using our proposed *SmartDPM* is presented in Algorithm 1. Initially, the workloads are monitored by the observer for few phases instances. Then, a linear model is built based on the previous p values, as given in Line 1 of Algorithm 1. Further, an online learning is employed to update the coefficients and minimize the prediction in case of variations, as in Line 4 of Algorithm 1. Further, based on the predicted workload values, the Mahalanobis distance metric is used to map the application (workload) to one of the four VF levels for which the providable powers are defined based on the model, represented by c_k. In Line 5, q represents the number of VF levels. The VF level with smaller distance is assigned to the predicted workload of the application, as described in Line 5–11 of Algorithm 1. For the brevity, the performance requirement is not presented in the Algorithm, however, when the performance constraint is not met, the VF settings of the next higher-level is set until the performance is met.

ALGORITHM 1 (PROPOSED *SmartDPM* BASED POWER MANAGEMENT).

Input: Workloads ($W = \{W_1, W_2, \ldots\}$), supported voltage–Frequency (VF) pairs ($C = \{c_1, c_2, \ldots, c_q\}$)

Output: Minimized power consumption with VF Scaling

1: Workload $W_i = \{x_i(1), x_i(2), \ldots, x_i(n)\}$;
2: **for** $j = p$ to n **do**
3: $\quad \widehat{x_i(j)} = \sum_{k=1}^{p} a_k * x_i(j-k)$;
4: $\quad \sum_i^p a_i R(j-i) = -R(j)$; (Find a_i)
5: $\quad$ **for** $k = 1$ to q **do**
6: $\quad\quad d(\widehat{x_i(j)}, c_k) = \sqrt{(\widehat{x_i(j)} - c_k)S^{-1}(\widehat{x_i(j)} - c_k)}$;
7: $\quad\quad$ **if** $d(\widehat{x_i(j)}, c_k) = \min$ **then**
8: $\quad\quad\quad W_i(j) \leftarrow c_k$;
9: $\quad\quad$ **else**
10: $\quad\quad\quad k = k + 1$;
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: **end for**

6. RESULTS AND DISCUSSION

First, we present the simulation settings and our tool flow for evaluating the efficacy of our *SmartDPM* technique, followed by achieved power savings.

6.1. System Settings

The proposed *SmartDPM* power management scheme is implemented in the SniperSim multi-core simulator,[40] which is a parallel, interval-accurate, and high-speed ×86 simulator. The maximum voltage and frequency levels are 1.2 V and 2.66 GHz, respectively. In simulations, we use four voltage–frequency levels for power management, that are supported by standard Nehalem microachitecture based cores: (1.2 V, 2.66 GHz), (1.1 V, 1.8 GHz), (1.0 V, 1.5 GHz) and (0.9 V, 1.0 GHz). However, this could be modified depending on the simulation environment and the utilized cores. A standard switching time for the voltage–frequency regulator (less than 2 μs) is considered in the simulations, as reported in Ref. [58]. Additional details on the configuration of microprocessor core and other components are presented in Table I. In order to validate the performance of *SmartDPM*, simulations are run with Parsec[62] and SPLASH-2[63] benchmarks. The number of cores are varied from 1 to 32. A machine learning predictor with observer controller loop based predictor of order 4, as

Table I. Overview of core configuration.

Item	Description	Value
Microprocessor core	Frequency (Max)	2.66 GHz
	Voltage (Max.)	1.2 V
	Technology node	22 nm
	L1-I cache	32 KB
	L1-D cache	32 KB
	L2 cache	256 KB
L3-cache	8 MB	

Copyright American Scientific Publishers
Delivered by Ingenta to: Tue, 20 Aug 2019 22:47:13

described in Section 5.1 is employed to learn and predict workloads.

6.2. Tool Flow for SmartDPM

The tool flow to perform *SmartDPM* in Sniper multi-core simulator is presented in Figure 4. The Sniper multi-core simulator is initialized with the configurations such as number of cores, VF levels, applications to be run and so on. For an unbiased and better evaluation of *SmartDPM*, applications are assigned to the core(s) in a random manner. Next, the workload statistics at application level granularity are monitored from the simulator and are provided to the *SmartDPM* power manager. The predictor builds the model during the training phase based on the training data and to minimize the root mean square error (RMSE) of the predicted values. During the test phase or post-training phase, the model will be updated with the aid of online learning in an observer-controller loop manner so as to adapt to the workload variations, as described in Section 5.1. Based on the predicted workload(s), the corresponding voltage and frequency levels are assigned to the core(s) that are running the corresponding applications. Lastly, McPAT[64] is used to obtain the power consumption statistics of the microprocessor. To perform Non-DVFS, the workloads can be run on similar settings without invoking the DVFS.

6.3. SmartDPM Performance

In this section we present the power savings in a multi-core system with *SmartDPM* and compare the achieved power savings with non-DVFS technique and other similar proposed approaches such as use of traditional linear regression,[43,44] SVM regression based power management[38] and space-time multiplexing (STM) based power management.[8] For a fair comparison, these techniques are re-implemented as described in Refs. [8, 38, 43, 44] and adapted for application based power management. Before presenting the performance of *SmartDPM*, we illustrate the performance of *SmartDPM* when performed at different granularities and the associated overheads.

6.3.1. Power Management at Different Granularities

The proposed *SmartDPM* focuses on power management at application level. However, it is also possible to employ *SmartDPM* at lower granularity (core-level) and higher granularity (system-level or per-chip level). As a case study, we present the impact of power management at different granularity levels for a 4-core microprocessor. For analysis, multi-threaded applications are chosen based on the manner the workloads are distributed among cores. The two workloads categories chosen are: (a) tightly coupled workload; (b) loosely coupled workload. Here, tightly coupled workload indicates that the workloads of an application are evenly distributed among multiple cores, and loosely coupled workload indicates that the workload of an application is unevenly distributed among multiple cores.

The power consumption at three different granularity levels for a microprocessor running multi-threaded application(s) is shown in Figure 5. Following are the observations:

- For loosely coupled multi-threaded applications, application level power management has better power savings compared to system level, if the applications are uncorrelated i.e., applications are dissimilar.
- If the workloads are loosely coupled and correlated i.e., similar workloads, system-level and application-level power management achieve similar power savings.
- In case of single multi-threaded application distributed among all the cores, irrespective of granularity, the power management achieves similar performance, if the application is tightly coupled.
- For a loosely coupled application, system-level and application level power management has similar performance.

IP: 129.174.252.250 On: Tue, 20 Aug 2019 22:17:13
Copyright: American Scientific Publishers
Delivered by Ingenta

Per-core power management has better power savings, however this adds additional overheads such as monitoring, power regulators for each of the core. System level power management has smaller overhead, and reduced power saving compared to per-core power management. Per-application level power management has performance in-between per-core and system-level power management. As running multiple applications is much realistic on multi-core microprocessor, per-application based power

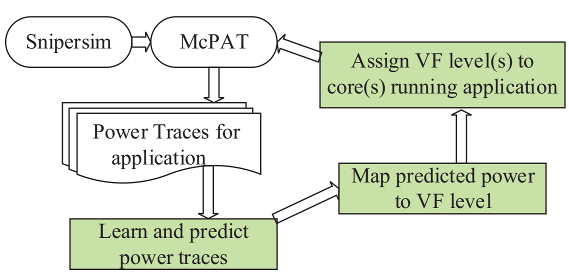


Fig. 4. Experimental tool flow to perform *SmartDPM* in SniperSim.

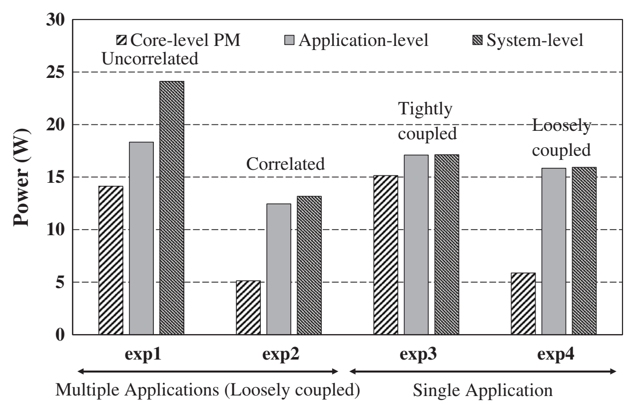


Fig. 5. Power savings with *SmartDPM* at different granularity levels.

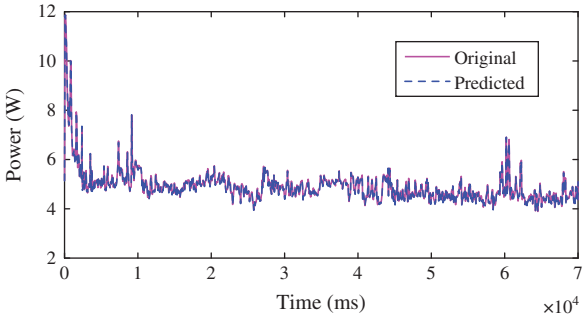


Fig. 6. An observer-controller loop based workload prediction for *SPLASH-2 Raytrace* benchmark.

management is a better choice for power management to have power savings with smaller overheads.

6.3.2. Workload Prediction

Workload prediction at phase level is one of the key steps in the proposed *SmartDPM*. Prediction of the SPLASH-2 Raytrace benchmark's power trace with time is depicted in Figure 6. As it can be clearly observed that the predicted power trace follows the original trace, even under some of the sporadic variations, such as at 1.1 ms. The RMSE is 0.21. The number of previous inputs considered for prediction in observer-controller loop is set to 4. It can be observed that the sudden variations such as spikes and peaks are well captured with the use of observer-controller loop based prediction. On an average, the RMSE for all the benchmarks is 0.49. The RMSE varies with benchmark. For the employed benchmarks, a max RMSE error of 0.88 is observed, which is smaller compared to maximum RMSE error obtained with other prediction techniques such as covariance, and neural networks.

6.3.3. Power Saving with SmartDPM

As each of the applications has different power consumption, we present the normalized power saving with *SmartDPM*. Figure 7(a) shows the normalized power consumption of different SPLASH-2 benchmarks that run on a single-core microprocessor, whose configuration is presented in Table I. In Figures 7(a), 8(a) and 9(a), *X*-axis represents the application that is run on microprocessor and *Y*-axis represent the normalized power consumption.

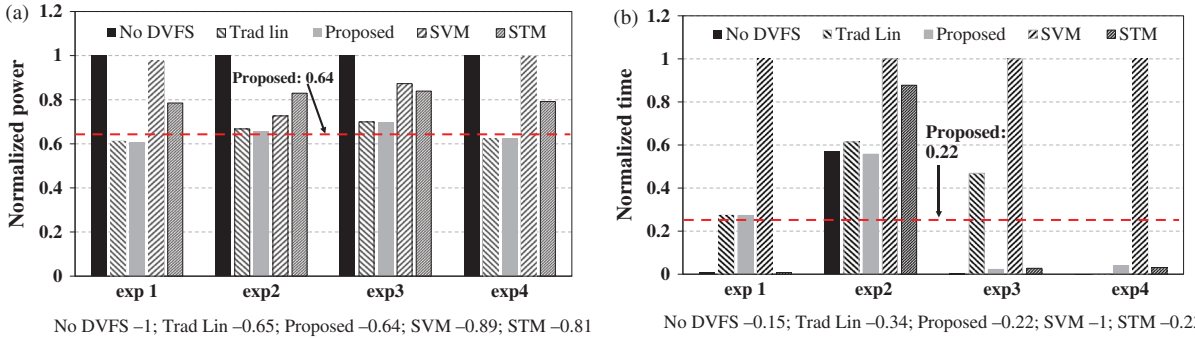
For a single-core microprocessor, an average power saving of 36% is observed compared to non-DVFS implementation as shown in Figure 7(a). Similarly, the power consumption by employing *SmartDPM* on a dual-core microprocessor compared to non-DVFS technique is depicted in Figure 8(a). In case of dual-core microprocessors, where SPLASH-2 and Parsec benchmarks are randomly executed in different experiments, nearly 36% of power saving on average compared to non-DVFS implementation is observed.

On an average, 34% power saving is observed for microprocessors with up to 32-cores, running SPLASH-2 and Parsec benchmarks on it, as illustrated in Figure 9(a).

6.3.4. Comparison

To evaluate the effectiveness of *SmartDPM*, we compare the achieved power savings of *SmartDPM* with other proposed works. Despite the fact that there exist many works on power management, we implemented few works such as Refs. [8, 38, 43, 44] (with minor adaptations such as power management at application level) for a fair comparison. The rationale for choosing those works are as follows: In Ref. [8], prediction of workload using auto-regressive moving average (ARIMA) and a singular value decomposition (SVD) based voltage–frequency level assignment is carried out, which is close to our work. Machine learning equipped power management is proposed in Ref. [38], where support vector machine (SVM) based regression for predicting workloads and SVM classifier based voltage–frequency level assignment is employed. The sparse encoding is not implemented, as the data is not as large as that in the original work. A linear regression with offline learning or modeling based workload prediction and VF level assignment is utilized in Refs. [43, 44]. Similar resemblances can be observed from other existing works as well. Our proposed methodology as well follows a two step process for power management: prediction

IP: 128.17.255.250 On: Tue, 21 Aug 2018 22:47:13
Copyright: American Scientific Publishers
Delivered by Ingenta



No DVFS –1; Trad Lin –0.65; Proposed –0.64; SVM –0.89; STM –0.81

No DVFS –0.15; Trad Lin –0.34; Proposed –0.22; SVM –1; STM –0.23

Fig. 7. Performance comparison of non-DVFS, *SmartDPM*, neural network and SVM based power management in a single-core microprocessor: (a) Power savings; and (b) runtime.

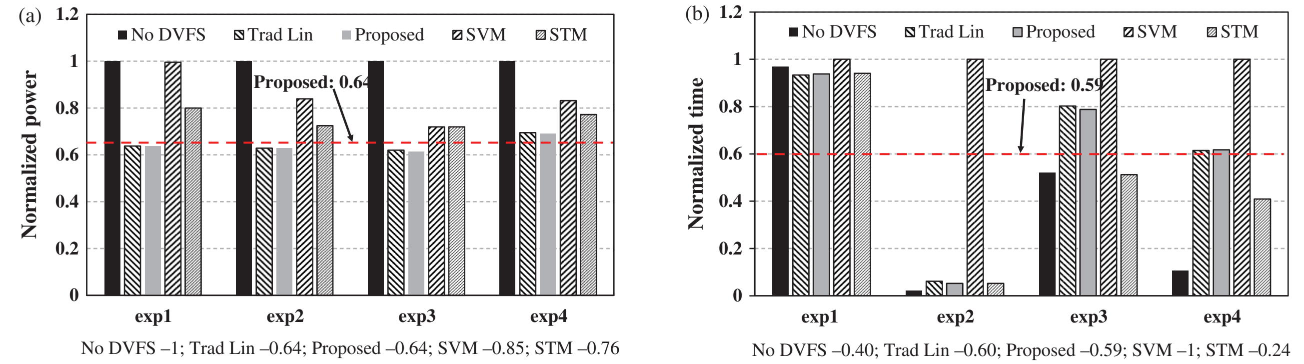


Fig. 8. Performance comparison of non-DVFS, *SmartDPM*, neural network and SVM based power management in a dual-core microprocessor: (a) Power savings; and (b) runtime.

of workload in an online learning observer-controller loop manner and voltage–frequency level assignment by clustering of data. The updating of predictor model based on the observation and the voltage–frequency assignment methodology with Mahalanobis distance primarily differentiates our work from other works. The comparison is provided in Figures 7–9. In the legend, 'Proposed,' 'No DVFS,' 'Trad Lin,' 'SVM,' and 'STM' represents the achieved performance (power or runtime) with *SmartDPM*, non-DVFS, traditional linear regression based power management, SVM and space-time multiplexing based power management techniques, respectively.

For a single-core microprocessor, *SmartDPM* consumes nearly 36% less than that of non-DVFS implementation, whereas linear regression based power management consumes 35% less than non-DVFS implementation; similarly, space-time multiplexing and SVM based power management techniques consume 11% and 19% less power compared to the non-DVFS technique. As such, it can be derived that linear regression and proposed *SmartDPM* based power management performs nearly similar in case of single-core microprocessors for the experimented benchmarks. Nearly 24% and 16% additional power saving is achieved with *SmartDPM* compared to SVM and STM based power management techniques, respectively, in a single-core microprocessor.

For multi-core microprocessor simulations, benchmarks (mixture of Parsec and SPLASH-2) are assigned to cores with *SmartDPM* deployed for power management. A comparison of power consumption in a 2-core microprocessor with randomly assigned benchmarks is presented in Figure 8. For a dual-core microprocessor, proposed *SmartDPM* has nearly 20% and 12% more power saving compared to SVM and STM based approaches, respectively. However, the linear regression performs similar to that of *SmartDPM*.

Similarly, we compare the power savings of *SmartDPM* for more number of cores. Compared to similar implementation as space-time multiplexing (STM) in Ref. [8], proposed method achieves nearly 14% higher power saving on average; compared to proposed method, support vector machine based implementation similar to Ref. [38], has nearly 14% lower power savings on average for 32 cores; and compared to linear regression based approach in Ref. [43], a power saving of nearly 2% is obtained. However, it can be observed from Figure 9(a), that the linear regression and proposed method performs similarly for smaller number of cores. However, for microprocessors with large number of cores such as 16 cores or 32 cores, *SmartDPM* outperforms linear regression in terms of power saving, showing that proposed technique

IP: 129.174.252.250 On: Tue, 28 Aug 2018 22:11:13
Copyright: American Scientific Publishers
Delivered by Ingenta

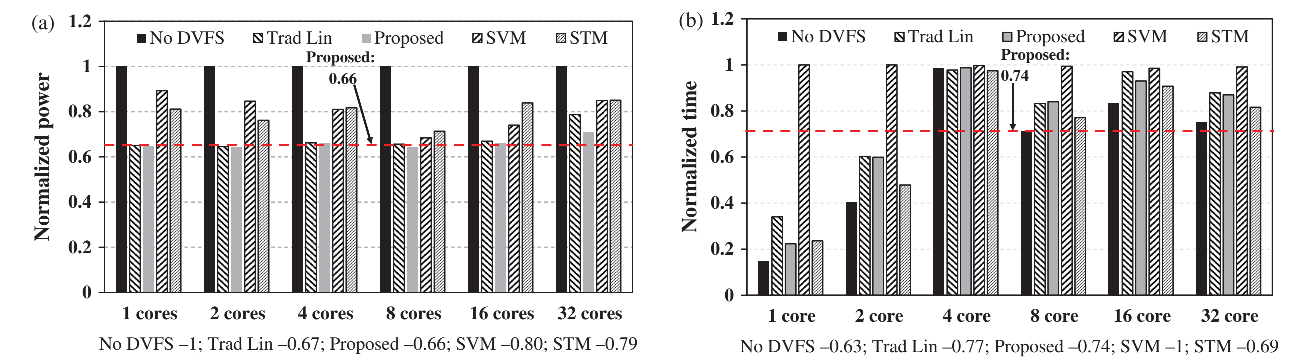


Fig. 9. Performance comparison of non-DVFS, *SmartDPM*, neural network and SVM based power management in a multi-core microprocessor: (a) Power savings; and (b) runtime.

is scalable for many-core systems with large number of cores.

The achieved power savings with *SmartDPM* depends highly on the distribution of application(s) in microprocessor. A tightly coupled core-application distribution achieves higher power savings similar to per-core DVFS i.e., when an multi-threaded application is partitioned and evenly distributed among cores, the power saving is higher than unbalanced distribution. As one more instance, in an 8-core microprocessor, we have observed a saving of nearly 30% compared to non-DVFS when the cores run similar workloads, whereas to analyze the impact of distributing application in an unbalanced manner is carried out, a power saving of only 8% is observed. In the demonstrated results, especially in multi-core simulations, applications are distributed among cores randomly to analyze the power savings with *SmartDPM* under no controlled application distribution among cores.

6.3.5. Overhead Analysis

The proposed *SmartDPM* power management scheme is effective in minimizing the overall system power. However, the power management with *SmartDPM* involves some overhead, that are: (a) additional computations needed for predicting the workload characteristics; (b) Updating the prediction model; (c) voltage frequency assignment based on the predicted workload characteristics. These overheads can be justified as follows: Prediction of workload helps to assign the VF level to the core(s) running the application, thereby avoiding overpowering of cores. Updating the model leads to learning the workload effectively and predict the workloads accurately during sporadic variations as well. As the predictions are performed at phase-level granularity, the computations required for prediction and assigning the VF levels are smaller compared to continuous prediction and changing the assigned VF levels.

We provide an analysis of the proposed method in terms of its impact on runtime. Similar to power savings, we compare the overhead as well with other works. As each application has different runtime, we present the runtime for different benchmarks on single-core, dual-core and multi-core in a normalized manner. The runtime for single-core microprocessor, dual-core microprocessor and multi-core microprocessor running SPLASH-2 and Parsec benchmarks are depicted in Figures 7(b), 8(b) and 9(b) respectively.

On average the proposed method, *SmartDPM*, requires nearly 10% more time than non-DVFS technique; 3% lower time than linear regression based power management; 23% lower runtime compared to SVM based power management; and 5% higher runtime than STM based implementation for microprocessors up to 32 cores, which can be seen in Figure 9(b). As the non-DVFS methodology involves no additional computations, it is obvious that the non-DVFS is faster than *SmartDPM*, but at the cost of additional power. As seen in motivational case study that SVM requires more time for prediction, the same is reflected in the runtime measurements. As traditional linear regression and proposed *SmartDPM* uses linear models, they have similar runtime, however due to better convergence, and adaptations, *SmartDPM* is faster and has better power savings even for large number of cores. In case of STM, the clustering is performed relatively simpler compared to the proposed work, though the prediction takes similar time. As such the *SmartDPM* has 5% more runtime on average than STM based power management technique.

As a summary, the proposed *SmartDPM* technique performs efficient per-application power management utilizing *accurate yet lightweight online learning observer-controller loop predictor* with less overhead. It outperforms some of the existing power management techniques.

7. CONCLUSION

SmartDPM with online learning observer-controller loop based workload prediction at phase-level to perform power management in a multi-core microprocessor is proposed in this paper. *SmartDPM* performs power management at an application level and utilizes lightweight linear predictor with continuous feedback of prediction error to learn the underlying patterns in the workload behavior. Based on the predicted workload characteristics, voltage and frequency pairs are assigned to the applications. A power saving of nearly 34% on average is achieved with *SmartDPM* compared to non-DVFS technique and nearly 15% more power saving on average compared to other power management techniques for a microprocessor with up to 32-cores.

Copyright American Scientific Publishers
Delivered by Ingenta to: Guest User Tue, 20 Nov 2018 12:34:45

References

1. S. Pagani, P. D. S. Manoj, A. Jantsch, and J. Henkel, Machine learning for power, energy, and thermal management on multi-core processors: A survey. *IEEE Transactions on Computer Aided Systems of Integrated Circuits and Systems* **(2018)**, pp. 1–17.
2. H. Khdr, S. Pagani, Éricles Sousa, V. Lari, A. Pathania, F. Hannig, M. Shafique, J. Teich, and J. Henkel, Power density-aware resource management for heterogeneous tiled multicores. *IEEE Transactions on Computers* 66, 488 **(2017)**.
3. S. Pagani, A. Pathania, M. Shafique, J. Chen, and J. Henkel, Energy efficiency for clustered heterogeneous multicores. *IEEE Trans. on Parallel and Distributed Systems* 28, 1315 **(2017)**.
4. P. D. S. Manoj, J. Lin, S. Zhu, Y. Yin, X. Liu, X. Huang, C. Song, W. Zhang, M. Yan, Z. Yu, and H. Yu, A scalable network-on-chip microprocessor with 2.5D integrated memory and accelerator. *IEEE Transactions on Circuits and Systems I*: *Regular Papers* 64, 1432 **(2017)**.
5. M. U. K. Khan, M. Shafique, A. Gupta, T. Schumann, and J. Henkel, Power-efficient load-balancing on heterogeneous computing platforms, *Design, Automation Test in Europe Conference Exhibition (DATE)* **(2016)**.
6. Y. Wang and M. Pedram, Model-free reinforcement learning and bayesian classification in system-level power management. *IEEE Trans. on Computers* 65, 3713 **(2016)**.

7. M. Shafique, D. Gnad, S. Garg, and J. Henkel, Variability-aware dark silicon management in on-chip many-core systems, *Design, Automation Test in Europe Conference Exhibition (DATE)* **(2015)**.

8. P. D. S. Manoj, H. Yu, and K. Wang, 3D many-core microprocessor power management by space-time multiplexing based demand-supply matching. *IEEE Trans. on Computers* 64, 3022 **(2015)**.

9. J. Lin, S. Zhu, Z. Yu, D. Xu, P. D. S. Manoj, and H. Yu, A scalable and reconfigurable 2.5D integrated multicore processor on silicon interposer, *IEEE Custom Integrated Circuits Conf.* **(2015)**.

10. S. Pagani, H. Khdr, W. Munawar, J. Chen, M. Shafique, M. Li, and J. Henkel, TSP: Thermal safe power—Efficient power budgeting for many-core systems in dark silicon, *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)* **(2014)**.

11. S. J. Tarsa, A. P. Kumar, and H. T. Kung, Workload prediction for adaptive power scaling using deep learning, *IEEE Int. Conf. on IC Design Technology* **(2014)**.

12. P. D. S. Manoj, H. Yu, Y. Shang, C. S. Tan, and S. K. Lim, Reliable 3-D clock-tree synthesis considering nonlinear capacitive TSV model with electrical-thermal-mechanical coupling. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 32, 1734 **(2013)**.

13. P. D. S. Manoj, K. Wang, and H. Yu, Peak power reduction and workload balancing by space-time multiplexing based demand-supply matching for 3D thousand-core microprocessor, *ACM/EDAC/IEEE Design Automation Conf.* **(2013)**.

14. H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, Dark silicon and the end of multicore scaling, *Int. Symposium on Computer Architecture* **(2011)**.

15. M. Salehi, M. K. Tavana, S. Rehman, F. Kriebel, M. Shafique, A. Ejlali, and J. Henkel, DRVS: Power-efficient reliability management through dynamic redundancy and voltage scaling under variations, *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* **(2015)**.

16. H. Hantao, P. D. S. Manoj, D. Xu, H. Yu, and Z. Hao, Reinforcement learning based self-adaptive voltage-swing adjustment of 2.5D I/Os for many-core microprocessor and memory communication, *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)* **(2014)**.

17. D.-C. Juan, S. Garg, J. Park, and D. Marculescu, Learning-based power/performance optimization for many-core systems with extended-range voltage/frequency scaling. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 35, 1318 **(2016)**.

18. J. Howard, S. Dighe, S. R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. K. De, and R. Van Der Wijngaart, A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. *IEEE J. of Solid-State Circuits* 46, 173 **(2011)**.

19. M. E. Salehi, M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, and S. M. Fakhraie, Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs. *IEEE Trans. on Very Large Scale Integration Systems* 19, 1931 **(2011)**.

20. P. Choudhary and D. Marculescu, Power management of voltage/frequency island-based systems using hardware based methods. *IEEE Trans. on Very Large Scale Integration Systems* 17, 427 **(2009)**.

21. A. Miyoshi, C. Lefurgy, E. Van Hensbergen, R. Rajamony, and R. Rajkumar, Critical power slope: Understanding the runtime effects of frequency scaling, *Int. Conf. on Supercomputing* **(2002)**.

22. R. David, P. Bogdan, and R. Marculescu, Dynamic power management for multicores: Case study using the intel SCC, *IEEE/IFIP Int. Conf. on VLSI and System-on-Chip* **(2012)**.

23. M. Ghasemazar and M. Pedram, Variation aware dynamic power management for chip multiprocessor architectures. *Design, Automation Test in Europe* **(2011)**.

24. M. Shafique, A. Ivanov, B. Vogel, and J. Henkel, Scalable power management for on-chip systems with malleable applications. *IEEE Transactions on Computers* 65, 3398 **(2016)**.

25. H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, Malleable NoC: Dark silicon inspired adaptable network-on-chip, *Design, Automation Test in Europe Conference Exhibition (DATE)* **(2015)**.

26. G. Chen, K. Huang, and A. Knoll, Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination. *ACM Trans. Embed. Comput. Syst.* 13, 111:1 **(2014)**.

27. M. Shafique, L. Bauer, and J. Henkel, Adaptive energy management for dynamically reconfigurable processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33, 50 **(2014)**.

28. H. Lee, M. Shafique, and M. A. Al Faruque, Low-overhead aging-aware resource management on embedded GPUs, *Design Automation Conference* **(2017)**.

29. A. Pathania, H. Khdr, M. Shafique, T. Mitra, and J. Henkel, Scalable probabilistic power budgeting for many-cores, *Design, Automation Test in Europe Conference Exhibition (DATE)* **(2017)**.

30. M. Shafique, M. U. K. Khan, O. Tüfek, and J. Henkel, EnAAM: Energy-efficient anti-aging for on-chip video memories, *ACM/EDAC/IEEE Design Automation Conference (DAC)* **(2015)**.

31. D. Xu, P. D. S. Manoj, H. Huang, N. Yu, and H. Yu, An energy-efficient 2.5D through-silicon interposer i/o with self-adaptive adjustment of output-voltage swing, *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* **(2014)**.

32. M. Shafique, M. U. K. Khan, and J. Henkel, Power efficient and workload balanced tiling for parallelized high efficiency video coding, *IEEE International Conference on Image Processing (ICIP)* **(2014)**.

33. S. S. Wu, K. Wang, P. D. S. Manoj, T. Y. Ho, M. Yu, and H. Yu, A thermal resilient integration of many-core microprocessors and main memory by 2.5D TSI I/Os, *Design, Automation Test in Europe Conference Exhibition (DATE)* **(2014)**.

34. B. Dietrich, S. Nunna, D. Goswami, S. Chakraborty, and M. Gries, LMS-based low-complexity game workload prediction for DVFS, *IEEE Int. Conf. on Computer Design* **(2010)**.

35. K. Choi, R. Soma, and M. Pedram, Dynamic voltage and frequency scaling based on workload decomposition, *Int. Symp. on Low Power Electronics and Design* **(2004)**.

36. K. Choi, R. Soma, and M. Pedram, Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 24, 18 **(2005)**.

37. J. S. Lee, K. Skadron, and S. W. Chung, Predictive temperature-aware DVFS. *IEEE Trans. on Computers* 59, 127 **(2010)**.

38. M. Zaman, A. Ahmadi, and Y. Makris, Workload characterization and prediction: A pathway to reliable multi-core systems, *IEEE Int. On-Line Testing Symp.* **(2015)**.

39. D. B. L. Bong, J. Y. B. Tan, and K. C. Lai, Application of multilayer perceptron with backpropagation algorithm and regression analysis for long-term forecast of electricity demand: A comparison, *Int. Conf. on Electronic Design* **(2008)**.

40. T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, An evaluation of high-level mechanistic core models. *ACM Trans. on Architecture and Code Optimization* 11, 28:1 **(2014)**.

41. P. D. S. Manoj, K. Wang, H. Huang, and H. Yu, Smart I/Os: A data-pattern aware 2.5D interconnect with space-time multiplexing, *ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)* **(2015)**.

42. R. Zamani and A. Afsahi, Adaptive estimation and prediction of power and performance in high performance computing. *Computer Science—Research and Development* 25, 177 **(2010)**.

IP: 129.174.252.230 On: Tue, 23 Aug 2016 21:08:21
Copyright: American Scientific Publishers
Delivered by Ingenta

43. B. Rountree, D. K. Lowenthal, M. Schulz, and B. R. de Supinski, Practical performance prediction under dynamic voltage frequency scaling, *Int. Green Computing Conference and Workshops* (**2011**).

44. S. Yang, R. A. Shafik, G. V. Merrett, E. Stott, J. M. Levine, J. Davis, and B. M. Al-Hashimi, Adaptive energy minimization of embedded heterogeneous systems using regression-based learning, *Int. W. on Power and Timing Modeling, Optimization and Simulation* (**2015**).

45. S. Lee and T. Sakurai, Run-time power control scheme using software feedback loop for low-power real-time applications, *Asia-Pacific Design Automation Conf.* (**2000**).

46. A. Weissel and F. Bellosa, Process cruise control-event-driven clock scaling for dynamic power management, *Int. Conf. for Compilers, Architectures, and Synthesis for Embedded Systems (CASES)* (**2002**).

47. A. Abbasian, S. Hatami, A. Afzali-Kusha, and M. Pedram, Wavelet-based dynamic power management for nonstationary service requests. *ACM Trans. Des. Autom. Electron. Syst.* 13, 13:1 (**2008**).

48. G. Tesauro, R. Das, H. Chan, J. O. Kephart, C. Lefurgy, D. W. Levine, and F. Rawson, Managing power consumption and performance of computing systems using reinforcement learning, *Int. Conf. on Neural Information Processing Systems* (**2007**).

49. H. Jung and M. Pedram, Supervised learning based power management for multicore processors. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 29, 1395 (**2010**).

50. F. Islam and M. Lin, A framework for learning based DVFS technique selection and frequency scaling for multi-core real-time systems, *IEEE Int. Conf. on Embedded Software and Systems* (**2015**).

51. D. Xu, N. Yu, H. Huang, P. D. S. Manoj, and H. Yu, Q-learning based voltage-swing tuning and compensation for 2.5D memory-logic integration. *IEEE Design and Test* 35, 91 (**2018**).

52. D. Xu, N. Yu, P. D. S. Manoj, K. Wang, H. Yu, and M. Yu, A 2.5-D memory-logic integration with data-pattern-aware memory controller. *IEEE Design Test* 32, 1 (**2015**).

53. P. D. S. Manoj, H. Yu, H. Huang, and D. Xu, A Q-learning based self-adaptive I/O communication for 2.5D integrated many-core microprocessor and memory. *IEEE Trans. on Computers* 65, 1185 (**2016**).

54. Y. Wang, M. Triki, X. Lin, A. C. Ammari, and M. Pedram, Hierarchical dynamic power management using model-free reinforcement learning, *Int. Symp. on Quality Electronic Design (ISQED)* (**2013**).

55. M. Salehi, M. Shafique, F. Kriebel, S. Rehman, M. K. Tavana, A. Ejlali, and J. Henkel, dsReliM: Power-constrained reliability management in dark-silicon many-core chips under process variations, *Int. Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS)* (**2015**).

56. D. Brooks, R. P. Dick, R. Joseph, and L. Shang, Power, thermal, and reliability modeling in nanometer-scale microprocessors. *IEEE Micro* 27, 49 (**2007**).

57. A. Ejlali, B. M. Al-Hashimi, and P. Eles, Low-energy standby-sparing for hard real-time systems. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 31, 329 (**2012**).

58. R. Singhal, Inside intel® core microarchitecture (nehalem), *IEEE Hot Chips Symp.* (**2008**).

59. W. Kim, M. S. Gupta, G. Wei, and D. Brooks, System level analysis of fast, per-core DVFS using on-chip switching regulators, *Int. Symp. on High Performance Computer Architecture* (**2008**).

60. S. Abedinpour, B. Bakkaloglu, and S. Kiaei, A multi-stage interleaved synchronous buck converter with integrated output filter in a 0.18/spl mu/SiGe process, in *IEEE Int. Solid State Circuits Conf.* (**2006**).

61. P. Hazucha, G. Schrom, J. Hahn, B. A. Bloechel, P. Hack, G. E. Dermer, S. Narendra, D. Gardner, T. Karnik, V. De, and S. Borkar, A 233-MHz 80%–87% efficient four-phase DC–DC converter utilizing air-core inductors on package. *IEEE J. of Solid-State Circuits* 40, 838 (**2005**).

62. C. Bienia, S. Kumar, J. P. Singh, and K. Li, The PARSEC benchmark suite: Characterization and architectural implications, *Int. Conf. on Parallel Architectures and Compilation Techniques* (**2008**).

63. S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, The SPLASH-2 programs: Characterization and methodological considerations. *SIGARCH Comput. Archit. News* 23, 24 (**1995**).

64. S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures, *IEEE/ACM Int. Symp. on Microarchitecture* (**2009**).

IP: 130.174.252.250 On: Tue, 20 Aug 2019 22:47:13
Copyright: American Scientific Publishers
Delivered by Ingenta

P. D. Sai Manoj

P. D. Sai Manoj *is a research assistant professor at George Mason University (GMU), Fairfax, VA, United States. Prior to joining GMU, Dr. Manoj worked as a post-doctoral research fellow at TU Wien, Austria. He received his Ph.D. in Electrical and Electronic Engineering from Nanyang Technological University, Singapore, in 2015. Dr. Manoj has organized multiple special sessions at premier venues such as DAC, ICCAD, ESWEEK and served on TPC of various conferences. His research interests are in adversarial learning, neuromorphic computing, hardware accelerators, cyber-security for embedded processors, security in Internet of Things networks, machine learning for on-chip data processing, and self-aware SoC design. He is a recipient of A. Richard Newton Young Research Fellow Award' in DAC 2013. He is a Member of the IEEE.*

Axel Jantsch

Axel Jantsch *is professor of Systems on Chip at TU Wien, Vienna, Austria. He received his Ph.D. degree from TU Wien, Austria, in 1992. Between 2002 and 2014 he was professor in Electronic Systems design at KTH, Stockholm, Sweden. He has published over 300 papers in international conferences and journals, and 6 books as author or co-editor, and he has served on numerous program and organization committees. He has organized many special sessions, tutorials and workshops at international conferences. His current research interests include Dependability, robustness, and self-awareness in embedded systems. He is a Member of the IEEE.*

Muhammad Shafique

Muhammad Shafique *is a full professor of Computer Architecture and Robust Energy-Efficient Technologies (CARE-Tech.) at the Embedded Computing Systems Group, Institute of Computer Engineering, Faculty of Informatics, Vienna University of Technology (TU Wien) since November 2016. He received his Ph.D. in Computer Science from Karlsruhe Institute of Technology (KIT), Germany in January 2011. Dr. Shafique has given several Invited Talks, Tutorials, and Keynotes. He has also organized many special sessions at premier venues (like DAC, ICCAD, DATE, and ESWeek) and served as the Guest Editor for IEEE Design and Test Magazine (D&T) and IEEE Transactions on Sustainable Computing (T-SUSC). He has served as the TPC co-Chair of ESTIMedia and LPDC, General Chair of ESTIMedia, and Track Chair at DATE and FDL. He has served on the program committees of numerous prestigious IEEE/ACM conferences including ICCAD, ISCA, DATE, CASES, ASPDAC, and FPL. His research interests are in computer architecture, power- and energy-efficient systems, robust computing, dependability and fault-tolerance, hardware security, emerging computing trends like Neuromorphic and Approximate Computing, neurosciences, emerging technologies and nanosystems, self-learning and intelligent systems, FPGAs, MPSoCs, and embedded systems. Dr. Shafique received the 2015 ACM/SIGDA Outstanding New Faculty Award as well as several best paper awards and nomination in premier conferences. He is a senior member of IEEE and a member of ACM, SIGDA, SIGARCH, SIGBED, and HiPEAC.*

IP: 129.174.252.250 On: Tue, 20 Aug 2019 22:47:13
Copyright: American Scientific Publishers
Delivered by Ingenta