Power Conversion Efficiency-Aware Mapping of Multithreaded Applications on Heterogeneous Architectures: A Comprehensive Parameter Tuning

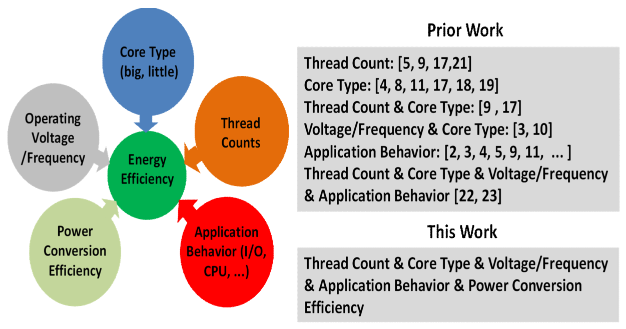
Hossein Sayadi*, Divya Pathak†, Ioannis Savidis†, Houman Homayoun*
*Department of Electrical and Computer Engineering, George Mason University, Fairfax, Virginia, USA
†Department of Electrical and Computer Engineering, Drexel University, Philadelphia, Pennsylvania, USA
e-mail: *{hsayadi, hhomayou}@gmu.edu †{divya.pathak, ioannis.savidis}@drexel.edu

Abstract - Heterogeneous Multicore Processors (HMPs) are comprised of multiple core types (small vs. big core architectures) with various performance and power characteristics which offer the flexibility to assign each thread to a core that provides the maximum energy-efficiency. Although this architecture provides more flexibility for the running application to determine the optimal run-time settings that maximize energy-efficiency, due to the interdependence of various tuning parameters such as the type of core, run-time voltage and frequency, and the number of threads, the scheduling becomes more challenging. More importantly, the impact of Power Conversion Efficiency (PCE) of the On-Chip Voltage Regulators (OCVRs) is another important parameter that makes it more challenging to schedule multithreaded applications on HMPs. In this paper, the importance of concurrent optimization and fine-tuning of the circuit and architectural parameters for energy-efficient scheduling on HMPs is addressed to harness the power of heterogeneity. In addition, the scheduling challenges for multithreaded applications are investigated for HMP architectures that account for the impact of power conversion efficiency. A highly accurate learning-based model is developed for energy-efficiency prediction to guide the scheduling decision. Using the predictive model, we further develop a PCE-aware scheduling scheme is developed for effective mapping of multithreaded applications onto an HMP. The results indicate that the proposed learning-based scheme outperforms the state of the art solution by 10% when there is no PCE gap between big and little cores. The energy-efficiency improves up to 60% when the PCE gap between big and little cores increases.

I. INTRODUCTION

Heterogeneous multicore processors offer significant advantages over homogeneous designs in terms of both performance and power by executing workloads on the most appropriate core type. By running multithreaded applications on a heterogeneous architecture, each thread is able to run on a core that matches required resources more closely than a one-size-fits-all solution [11]. Commercially available heterogeneous architectures include Intel Quick IA [6], ARM's big.LITTLE [8], and the Nvidia Tegra 3 [7] that integrates a high performance big core with a low power little core on a single chip. Although heterogeneous architectures take advantage of variation in the application characteristics at run-time to improve energy-efficiency, they create unique challenges in the effective mapping of threads to cores. The effectiveness of heterogeneous architectures significantly depends on the scheduling policy and how efficiently the application is assigned to the most appropriate processing core [1,3,4,9,11].

Previous studies have mainly examined the advantages of using single threaded applications in HMPs [1,3,4,10,11,13]. However, running multithreaded applications on HMPs and choosing the ideal processor architecture to optimize energy-efficiency is a more challenging problem, that must consider the possible number of cores and threads, type of core micro-architecture, and the potential to combine multiple core types [22,23]. In addition, prior work have ignored power conversion efficiency as a critical optimization parameter. In fact, unlike a homogeneous architecture, in an HMP, the maximum load on cores varies significantly depending on the core type. For instance, in an Exynos 5, the maximum power of big A15 is five times more than little A7 [12]. Therefore, there is a


Core Type (big, little)
Operating Voltage /Frequency
Thread Counts
Energy Efficiency
Power Conversion Efficiency
Application Behavior (I/O, CPU, ...)

Prior Work

Thread Count: [5, 9, 17,21]
Core Type: [4, 8, 11, 17, 18, 19]
Thread Count & Core Type: [9 , 17]
Voltage/Frequency & Core Type: [3, 10]
Application Behavior: [2, 3, 4, 5, 9, 11, ...]
Thread Count & Core Type & Voltage/Frequency & Application Behavior [22, 23]

This Work

Thread Count & Core Type & Voltage/Frequency & Application Behavior & Power Conversion Efficiency

Fig.1. Tuning parameters influencing energy-efficiency in heterogeneous architectures and prior work on scheduling using these tuning parameters.

difference in power conversion efficiency on big and little cores for the same application that is critical for scheduling. For instance, assume the same application executed on a big core and little core dissipates 1W and 0.9 W of power, respectively. Now consider a PCE of 90% and 70% for the big and little cores, respectively. The execution of the application now requires 1.1 W and 1.4W of power supplied to the big and little cores, respectively, which implies a change in the most efficient core type after accounting for PCE. Since power conversion efficiency is dependent on the load (core type), it is shown in this paper that it is critical to account for PCE when making scheduling decisions. The experimental results demonstrate that PCE directly affects the choice of the right core type (big vs. little) optimize energy-efficiency. In this work, an energy-efficient scheduling approach is proposed that accounts for the interplay between various application and micro-architectural tuning parameters with respect to the impact of on-chip power delivery on the energy-efficiency of the HMP executing multithreaded programs. To the best knowledge of the authors, there has been no prior effort to concurrently fine-tune the core type, operating voltage/frequency, and application thread counts that also considers the impact of the PCE of the voltage regulators on the optimization of the energy-efficiency in an HMP.

The tuning parameters that influence scheduling decisions in an HMP are shown in Fig. 1. In addition, recent prior work as well as the contribution of this work is illustrated in Fig. 1. Previous studies on mapping applications to multicore architectures have focused primarily on 1) homogeneous architectures, and 2) configuring individual or a subgroup of tuning parameters at a time, such as application thread counts [5,9,17,21], voltage/frequency [3,10], core type [4,8,11,17,18,19] and have ignored the interplay among all parameters. In addition, the recent studies in [22, 23] attempt to examine the interplay among tuning parameters for an HMP but have ignored the impact of PCE of voltage regulators. This study indicates that tuning parameters individually, while important, do not produce an optimized configuration that achieves the best energy-efficiency on an HMP. The best configuration for a multithreaded application is effectively found, only when tuning parameters are jointly optimized. Exploring the impact of on-chip voltage regulator PCE on the energy-efficiency of an HMP running multithreaded applications is an additional main contribution of this work.

Notice: This paper was presented at ASP-DAC 2018 but not by one of the authors.

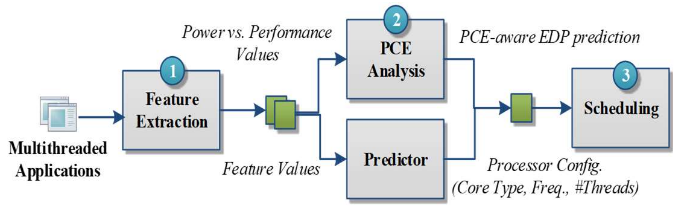

Power vs. Performance Values
PCE Analysis
PCE-aware EDP prediction
Feature Extraction
Scheduling
Multithreaded Applications
Feature Values
Predictor
Processor Config. (Core Type, Freq., #Threads)

Fig. 2. An overview of the PCE-aware learning-based approach.

The key contributions of this work are summarized as follows:

- The interplay of tuning parameters on performance, power, and energy-efficiency is evaluated for an HMP. The specific parameters at the micro-architecture, system and application levels that are critical to performance as well as power and energy-efficiency and are studied in this work are core type, voltage/frequency settings and the running thread counts.
- The impact of power conversion efficiency of on-chip voltage regulators on the selection of tuning parameters is investigated for maximizing the energy-efficiency. Specifically, four different settings for PCE of VRs are implemented to examine the effect of PCE on the energy-efficiency of multithreaded applications running on the HMP. The results indicate that the energy-efficiency of the multithreaded applications running on an HMP significantly depends on the power conversion efficiency of the on-chip voltage regulators.
- A system level optimization technique is developed that is aware of the PCE of the on-chip VRs. Based on conducted workload characterization and analysis of the PCE, a machine learning-based model is proposed for predicting the energy-efficiency of various configurations to guide scheduling of multithreaded applications.

II. Overview of PCE-Aware Scheduling

The primary objective of this paper is to analyze the interplay of various tuning parameters, (core type, voltage/frequency, number of threads), for running multithreaded applications on HMPs and to highlight the importance of accounting for the PCE of OCVRs for each core type to assist in scheduling decisions. An overview of the three-stage PCE-aware approach for predicting the right core type and application configuration is depicted in Fig. 2. The machine learning-based approach begins from extracting micro-architectural data (referred as feature extraction) and the power and performance (execution time) characteristics, to characterize the multithreaded workload, prepare the dataset for PCE analysis, and train the prediction model. The extracted features include the hardware performance counter data, which represent the application behavior at run-time.

Since PCE is dependent on voltage regulator design, the architecture of the big and little cores, and the maximum load gap between the two, in this work, no specific assumption is made regarding the PCE of the big and little cores. Instead, all possible scenarios representing various differences between the PCE of the big and little cores are explored. Next, a comprehensive PCE analysis is performed by implementing various PCE models for both big and little cores and evaluating the impact of power conversion efficiency on the energy-efficiency of multithreaded applications. Furthermore, a machine learning-based predictor (that is built off-line) accordingly takes in feature data as well as the PCE of the regulator and predicts the best system configuration for a given application. Finally, the processors are configured and the application is scheduled to run on the predicted configuration.

TABLE I. ARCHITECTURAL SPECIFICATION.

Microarch. Parameter	Little Core	Big Core
Dispatch Width	2	4
Window Size	32	128
Levels of Cache	2	3
L1 I-Cache/Acc. Time	32KB, 8-way/4-cyc	32KB, 4-way/2-cyc
L1 D-Cache/Acc. Time	24KB, 6-way/4-cyc	32KB, 4-way/2cyc
L2-Cache/Acc. Time	1024KB/16-way/12- cyc	256KB/8-way/8cyc
L2-Shared Cores	2	1
L3 Cache	-	8MB/16-way

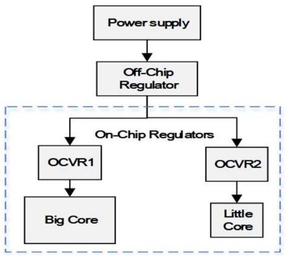


Fig. 3. The power-supply configurations for the experimental HMP.

III. Experimental Setup and Methodology

In this section details of the experimental setup are provided. Sniper [13] version 6.1, a parallel, high speed and cycle-accurate x86 simulator for multicore systems is used for simulation. McPAT is integrated with Sniper and is used to obtain power consumption of the cores. The SPLASH-2 [14] and PARSEC [2] multithreaded benchmark suites are examined through simulation. For architectural simulation, a big.LITTLE heterogeneous architecture is modeled. For the little core architecture, a core similar to the Atom Silvermont is modeled and the big core is configured with resources similar to the Xeon Gainestown. The Uncore event set of Silvermont and the Intelligent Performance Counter of Gainestown are used to collect data for characterization and drive the scheduling algorithm. The Energy Delay Product (EDP) is used to characterize energy-efficiency, that aims to balance performance and power consumption.

The micro-architectural configuration of the little and big core of the described HMP is listed in Table I. The examined HMP consists of 8 little and 4 big cores. It is important to note that for benchmark simulation, the binding (one-thread-per-core) model is applied with #threads = #cores to maximize the performance of multithreaded applications [4, 5].

IV. Power Conversion Efficiency Analysis

In this section, the motivation to include the voltage regulator efficiency as one of the tuning parameters influencing the energy-efficiency of the HMP is described. For analyzing the efficiency of on-chip voltage regulators, per-core voltage regulation is considered as shown in Fig. 3. In the model, each core has a dedicated OCVR, which is a flexible state of the art VR configuration that enables the system to set the voltage and frequency for each core individually to address core-to-core process variation [15, 16]. In addition, since the power management unit directly controls the OCVRs, turning them on or off, a power gating circuit is not needed. The impact of power conversion efficiency of the on-chip VRs on energy-efficiency of multithreaded applications is demonstrated by implementing the PCE scenarios listed in Table II.

The first case listed in Table I represents Full Efficient VRs. This is the ideal scenario in which the power conversion efficiency of the little and big cores is assumed to be 100%. The full efficiency case is used as a baseline for comparing the other PCE models and evaluating the impact of OCVR

TABLE II. FOUR PCE SCENARIOS FOR LITTLE AND BIG CORE VRS.

VRs PCE Models (Little Core vs. Big Core)	PCE_Little	PCE_Big
Full Efficiency	100%	100%
Low gap	60%	80%
Medium gap	40%	80%
Large gap	20%	80%

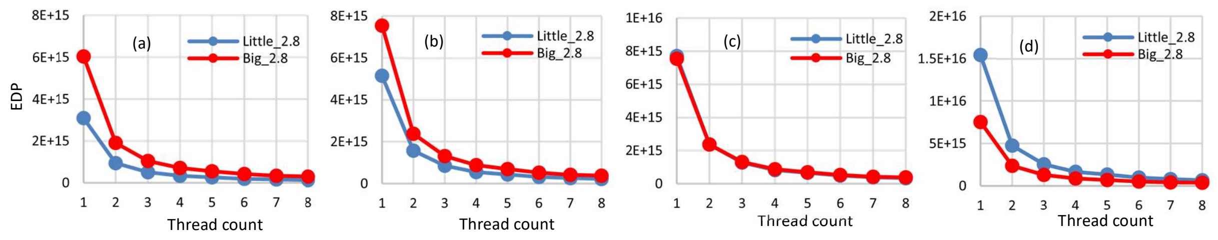


Fig. 4. Energy-efficiency (in terms of EDP) of *barnes* for four different PCE gaps between little and big cores: a) full efficiency, b) low, c) medium, d) large.

efficiency on the EDP. For this purpose, three different PCE sets are assigned to each OCVR with low, medium and large gap between the little and big core. The values are chosen more accurately than the baseline model represent the PCE of on-chip VRs and to effectively determine the impact of PCE variation on the energy-efficiency of HMP.

An example depicting the EDP of the *barnes* application while considering different on-chip VR models with varying PCE is shown in Fig. 4. The VR models are based on the 2-phase and 4-phase dc-dc buck converter models in [15]. The PCE of the VR models used for the big cores and little cores are listed in Table II. In order to effectively present the impact of voltage regulator PCE on the EDP in each case, in this section, the EDP results for one of the studied frequencies (2.8 GHz) is chosen to examine the gap between the energy-efficiency of the two cores for different per-core PCE values. Note that changing the number of threads interestingly affects the impact of the PCE on the energy-efficiency. As seen from Fig. 4-(a) and 4-(b), when the number of running threads is low (less than 3), the PCE significantly impacts the choice of selecting the more energy-efficient core as compared to the higher number of threads. The results shown in Fig 4-(a) clearly indicate there is a large gap between the energy-efficiency of little and big core when running the application with lower number of threads. However, that is not the case with higher thread counts. As the number of threads increases, the difference between the EDP of the little and big core reduces which makes the big core more competitive with the little core in terms of energy-efficiency.

As shown in Fig. 4-(b), it is assumed that there is a low gap between the PCE of the little and big cores. The assumed Little_PCE is 60% and the Big_PCE is 80%. As can be seen, even though the gap between the EDP of the little and big cores is relatively smaller than the baseline case, where the PCE for both core types is 100% (Fig. 4-a), as the number of threads is changed accordingly, the little core still outperforms the big core in terms of EDP delivering better energy-efficiency. Therefore, when the gap between the PCE of the little and big cores is relatively small, similar to the case when the PCE for both core types is 100%, it is more energy-efficient to migrate from the big core to the little core and run the application to achieve a lower EDP.

The EDP results for the scenario in which the PCE gap of the little and big core is increased to 40% (*medium* PCE gap) is depicted in Fig. 4-c. As is seen, the EDP for the little core increases and overlaps with the EDP of the big core, indicating that the two cores are almost as energy-efficient for PCE gap of 40%. In the last scenario, the PCE gap between the little and big core VRs is further increased. As shown, for large PCE gap, the big core outperforms the little core in terms of EDP. Therefore, as the PCE gap increases, the selection of the little core is no longer optimal in terms of EDP for running multithreaded applications.

The results of PCE analysis indicates that the most energy-efficient choice varies depending on the PCE gap between the big and little cores and the best choice changes compared to the case when the PCE is ignored. Also, as shown in Fig. 4, the number of threads along with the PCE gap determines the most efficient core. It is, therefore, important to explore the impact of the power conversion efficiency of the on-chip voltage regulators in an HMP to determine the optimal core configuration that achieves the optimized EDP.

V. Energy-Efficient Scheduling Framework

A. Joint analysis of (Core Type, Freqeuncy, Thread Counts) with respect to various PCE models

In section, to understand the interplay among all tuning parameters and determine the optimum configuration for maximizing the energy-efficiency, the interplay among the tuning parameters were comprehensively investigated with respect to each selected voltage regulator PCE model. Due to space limitations, the optimal configurations that yield the optimal EDP for two corner cases are reported which are the full efficiency and large PCE gap models listed in Table III. The relative EDP variation is also calculated for each benchmark, which indicates the relative difference between energy-efficiency for the best configuration of parameters in the little and big cores. The variation parameter (Var) is described as follows:

$$Var = \left(\frac{Little_{best\_EDP} - Big_{best\_EDP}}{Little_{best\_EDP}}\right) \times 100 \qquad (1)$$

The variation parameter indicates whether to run the application on the little core or big core. For this purpose, a variation threshold is defined that decides what type of core architecture is best suited for executing the corresponding multithreaded application more energy-efficiently. The user-defined threshold is adjusted based on the architecture and available resources as well as the cost of migration. Note that migrating applications from the little core to the big core or, vice versa, comes with power as well as delay overhead. In this

TABLE III. Optimal configuration with optimization target of EDP for full efficiency PCE and large gap PCE model.

Benchmark	Full Efficiency					Large PCE Gap				
	Best-Little		Best-Big		Var. (%)	Best-Little		Best-Big		Var. (%)
	Freq. (GHz)	*#Thread*	*Freq. (GHz)*	*#Thread*		*Freq. (GHz)*	*#Thread*	*Freq. (GHz)*	*#Thread*	
barnes	**2.4**	**8**	2.8	4	-444.8	**2.4**	**8**	2.8	4	-36.2
fmm	**2.4**	**8**	2.4	4	2.2	2.4	8	**2.4**	**4**	75.5
cholesky	2.4	8	**2**	**4**	28	2.4	8	**2**	**4**	77
radix	**2.8**	**8**	2.8	4	-138.7	2.8	8	**2.8**	**4**	40.3
radiosity	**2.4**	**8**	1.6	4	-102.8	2.4	8	**1.6**	**4**	49.3
raytrace	**2.4**	**5**	2	4	-28.9	2.4	5	**2**	**4**	67.7
fft	2	4	**2**	**2**	36.3	2	4	**2**	**2**	79.1
lu.cont	2.4	8	**2.8**	**4**	27.2	2.4	8	**2.8**	**4**	98.7
blackscholes	2.8	4	**2.4**	**4**	83.85	2.8	4	**2.4**	**4**	195.3
bodytrack	2	3	**2**	**3**	41.23	2	3	**2**	**3**	112.6
ferret	**2**	**6**	2	6	2.4	2	6	**2**	**6**	132.6

work, a conservative implementation with a delay overhead of 10K cycles is assumed, which is much longer than the overhead to flush the pipeline and copy the content of private cache [3,4,18]. Moreover, a 20% variation threshold is assumed to select the more energy-efficient core to run the multithreaded application. As a result, if the percentage of variation between the best-little and best-big architectures is found to be less than 20%, we use the little core for scheduling instead of the big core to avoid migration overhead.

An important observation from the optimal configurations highlighted in Table III is that as the gap in PCE increases, the optimal configurations corresponding to the best EDP show to occur more on the big core. The percentage of applications executed on each of the two core types for the four PCE scenarios is shown in Fig. 5. As shown, for the full efficiency model, the little core has a higher probability of being the optimal configuration. However, as the PCE gap gradually increases, the energy-efficient core is shifted from the little to the big core. As a result, for the large PCE gap model, as compared to the full efficient scheme, the possibility of the big core being the more energy-efficient than the little core increases by 45%. The reason is due to the significant increase in the EDP of the little core as compared to the big core when the PCE gap between the two cores increases. As shown in Fig. 5, close to 55% of the optimal configurations indicate that the little core is more energy-efficient. The number of optimal configurations reduces to less than 10% as the PCE gap increases to 60%. Therefore, considering the PCE of the OCVRs in scheduling decisions of multithreaded applications on HMPs is critical. In order to perform a comprehensive EDP characterization of the studied architectures, all possible configurations (core types and number of threads) are categorized into four classes. The first two are Fully-Little and Fully-Big configurations that refer to cases in which the lowest EDP is achieved with full utilization of either the little or big core, respectively. In other words, the optimum number of threads is equal to the maximum number of existing little/big cores. On the other hand, Partially-Little and Partially-Big configurations are utilized when the best number of threads is lower than the maximum available cores.

The diversity of optimum configurations across various applications and on-chip voltage regulator PCE scenarios demonstrates that when running a given multithreaded workload on an HMP, depending on the application and the PCE of the OCVRs, different core configuration parameters (core type, voltage/frequency, number of threads) lead to the best energy-efficiency. The simulation results indicate that the optimal configuration varies across the applications. The dispersed pattern of optimum results implies that there is a necessity of developing a prediction method to guide scheduling decisions of unknown multithreaded applications in order to improve the EDP of an HMP with respect to a given PCE of the OCVRs.

B. Prediction Model for Energy-efficiency

1) Model selection: Recent studies have proposed ordinary least squares regression (OLSR) modeling to estimate the power [10] and performance [3, 11, 19] of a processor at run-time. The results of this work indicate that OLSR is not the best suited algorithm for performance and power estimation as outliers, particularly for heterogeneous architectures, mislead the model. In fact, various applications experience different phases with different behavior. In addition, superscalar processors are complex, which makes it difficult to develop a general model for estimation of power/performance. OLSR models are highly sensitive to the outliers and potentially produce misleading results as even a single point of data substantially impacts the regression efficiency. Thus, in this paper, based on a comprehensive characterization of various applications, a more robust regression algorithm is evaluated in addition to OLSR, referred as the Quantile Linear Regression (QLR) model [20], to predict the energy-efficiency for various configurations of the studied HMP. The primary advantage of QLR as compared to OLSR is the robustness against outliers. For the QLR model, a specific quantile of data is set instead of the mean value. The quantile is set to 0.1, which results in minimizing the median of the error values.

Although the use of non-linear regression or neural network models potentially provides a more accurate estimation of the energy-efficiency of an application, the complexity of the design is increased, with a corresponding increase in hardware complexity. The overhead in area, power and performance of implementing a linear regression model in hardware is minimal and shown to be easily integrated into a core [11]. The QLR model achieves higher accuracy as compared to OLSR. A comparison between the derived coefficients of the two different predictors using ordinary linear regression and quantile linear regression is shown in Fig. 6. In the figure, the black dotted line is the slope coefficient for the QLR and the red lines are the least squares estimate for OLSR and the corresponding confidence interval. The lower and upper quantiles are well beyond the least squares estimate. The effects of L2 cache access and branch miss prediction vary over quantiles, and the magnitude of the effects at various quantiles differs considerably from the OLSR coefficient, even in terms of the confidence intervals around each coefficient (58% for the L2-access and 30% for the branch miss predictor). Therefore, an ordinary least squares regression is not an optimal solution to capture the actual behavior of applications when predicting the energy-efficiency.

2) QLR derivation and training: Training process involves finding the best processor and application configuration and extracting feature values for each training workload, reducing the extracted features to the most vital performance counters, and developing a learning model from the training data. Note that the input variables in the developed classifiers are extracted performance counter information from different training applications as well as the PCE value for each on-chip voltage regulator, while the output variable is the EDP for a given set of tuning parameters. Therefore, a subset (less than two third) of applications from the SPLASH2 and PARSEC multithreaded benchmark suites is considered. The studied

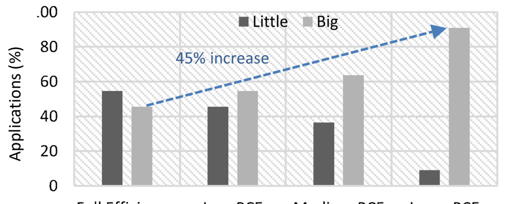


Fig. 5. Optimal core type selection for different power conversion efficiencies.

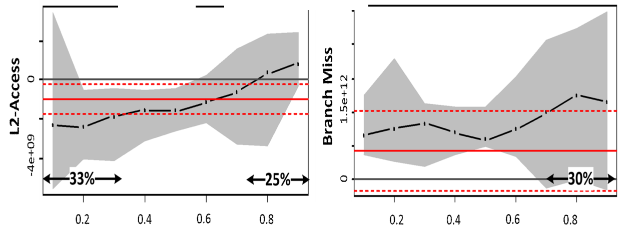


Fig. 6. Quantile graphs for predictors: a) L2-Access, b) Branch miss prediction.

TABLE IV. HARDWARE PERFORMANCE DATA USED FOR THE REGRESSION MODEL.

Category	Hardware performance counter
Memory subsystem	L1 D-cache access, L1 D-cache miss, L1 I-cache access, L1 I-cache miss, L2 cache access, L2 cache miss, I-TLB miss, D-TLB miss
Instructions	Integer instruction issue, Integer floating point issue
Branch	Branch instruction, Branch misprediction

applications represent diverse compute, memory and I/O intensity behavior. For each benchmark, twelve pieces of hardware performance counter data are collected on all possible configurations of core types, voltage/frequency operating points, and thread counts. The micro-architectural parameters are listed in Table IV.

Given the twelve hardware performance counters, Principle Component Analysis (PCA) and correlation analysis are used on the training set to monitor the critical micro-architecture parameters and capture application characteristics. By applying the attribute reduction method, the four most related performance counters are determined which include the L1 D-cache access, L2 cache-access, L2 cache-miss and branch miss prediction. Since the primary purpose of the prediction model is to predict energy-efficiency across various application, system and micro-architectural parameters, the primary tuning parameters in must be considered in the model as well. Therefore, along with the identified key performance counter parameters, three tuning parameters (core type, frequency, #threads) are included as input variables to the model to enable predicting the EDP for each configuration that results when changing the core type, operating frequency, and/or thread counts.

After identifying the four key hardware performance parameters and considering the tuning parameters, the proposed PCE-aware energy-efficiency prediction model is formulated using quantile linear regression as follows:

$$QLRM_{PCE} = \left(\beta 0 + \sum_{i=1}^{4} \beta i \times Pi\right) + \beta 5 \times CT + \beta 6 \times f \qquad (2)$$

where $\beta 0$ is the intercept, βi denotes the corresponding coefficients of the regression model, Pi are extracted hardware performance counters, and $\mathrm{QLRM}_{\mathrm{PCE}}$ is the estimated energy-efficiency (in terms of EDP) given the PCE of the on-chip voltage regulators. In addition, the core/thread configurations are given by CT, and f represents the frequency on the corresponding core architecture. The βi coefficients can be interpreted as the expected change in EDP per unit change in L1 D-cache access, L2 cache-access, L2 cache-miss, branch misprediction, core/thread and frequency setting. The model predicts continuous values representing the energy delay product as a function of performance counter inputs and tuning parameters, which are then used to make the scheduling decisions at run-time. During run-time, given an unknown application, the $\mathrm{QLRM}_{\mathrm{PCE}}$ predict the EDP of all possible configurations based on a single set of executing data. The configuration corresponding to the lowest estimated EDP is then selected for the run.

C. *Energy-Efficient PCE-aware Scheduling Algorithm*

An overview of the PCE-aware scheduling scheme using the regression-based prediction model is provided in Fig. 7. As illustrated, the scheduling algorithm is split between an offline step and an online step. In offline analysis, the prediction model is trained using quantile linear regression, as described in section V-B. For the online tuning step, a multithreaded application is run with the most aggressive configuration settings, where all tuning parameters are set to maximum (maximum number of threads, highest frequency, and the big core). Next, date from the hardware performance counters is extracted by profiling the multithreaded application, as it is running with the maximum configuration settings. The profiling stage is used for run-time characterization and resource utilization of the applications. The regression classifier takes the key performance counter parameters and configuration settings as inputs, and outputs the system energy-efficiency for the given configuration and given PCE. Note that the linear weights are estimated using the training data set. Given the input configuration parameters during run-time, the $\mathrm{QLRM}_{\mathrm{PCE}}$ predicts the optimal energy-efficiency. The output resulting in the optimal energy-efficiency and the corresponding new configuration is then chosen as the current operating point at run-time. The predictive model, by observing the run-time behavior of a multithreaded application running with a specific configuration, predicts the right configuration of parameters that includes the number of threads, operating voltage and frequency, and core type (big or little) to achieve the maximum energy-efficiency for a given PCE. It is important to note that the $\mathrm{QLRM}_{\mathrm{PCE}}$ can be trained for other objectives such as $\mathrm{ED^2P}$ optimization.

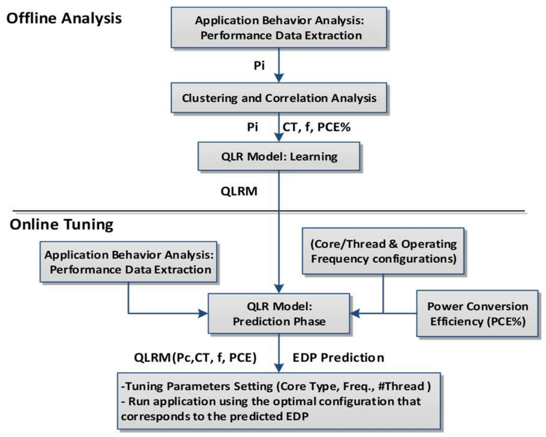

Offline Analysis
Application Behavior Analysis: Performance Data Extraction
Pi
Clustering and Correlation Analysis
Pi
CT, f, PCE%
QLR Model: Learning
QLRM
Online Tuning
Application Behavior Analysis: Performance Data Extraction
(Core/Thread & Operating Frequency configurations)
QLR Model: Prediction Phase
Power Conversion Efficiency (PCE%)
QLRM(Pc,CT, f, PCE)
EDP Prediction
-Tuning Parameters Setting (Core Type, Freq., #Thread)
- Run application using the optimal configuration that corresponds to the predicted EDP

Fig. 7. Proposed PCE-aware scheduling scheme with energy-efficiency prediction.

D. *Evaluation Results*

In order to evaluate the accuracy of the prediction model, the value of the relative mean absolute error (RMAE) is calculated which is defined as $\frac{|estimated\ value - actual\ value|}{(actual\ value)} \times 100\%$. The RMAE metric indicates the relative difference between the predicted and observed maximum energy-efficiency. To validate the $\mathrm{QLRM}_{\mathrm{PCE}}$ model, we applied percentage split method to divide the dataset into two sets, using 60% (known applications) of the data to train the model and 40% (unknown applications) to simulate and evaluate.

The average relative errors of the $\mathrm{QLRM}_{\mathrm{PCE}}$ are listed in Table V. As shown, all possible configurations of the 16 operating points consisting of various frequencies and core/thread configurations are characterized. As shown, the proposed prediction classifier is most accurate in estimating the energy-efficiency of the Fully-Big and Fully-Little architectures, both operating at 2 GHz. In addition, the developed learning model achieves an average error of 6.85% across all training data samples and possible configurations. The proposed classifier assists the scheduling decisions of multithreaded applications on an HMP that include choosing the core type, setting the operating voltage and frequency, and adapting the number of running threads. The performance overhead of implementing the $\mathrm{QLRM}_{\mathrm{PCE}}$ in hardware and calculating values at each interval is negligible. The power

TABLE V. AVERAGE RELATIVE ERROR OF $\mathrm{QLRM}_{\mathrm{PCE}}$

	Core/Thread Configurations			
Freq.	**Full-Little**	**Partial-Little**	**Full-Big**	**Partial-Big**
2.8 GHz	10.5%	10.74%	11.69%	2.03%
2.4 GHz	22.49%	21.4%	4.67%	4.87%
2.0 GHz	1.9%	3.9%	1.74%	3.1%
1.6 GHz	3.35%	2.2%	3.6%	2.61%

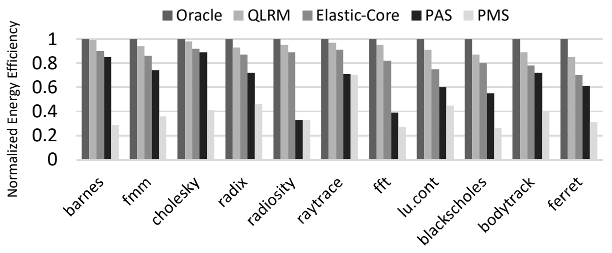


Fig. 8. Normalized energy-efficiency of applications on various scheduling schemes relative to Oracle scheduling.

overhead of implementing the $QLRM_{PCE}$ is 5uW, which is further reduced by gating idle units during each interval [11]. In order to evaluate the efficiency of the prediction model, the following scheduling schemes are studied for comparison:

- *Oracle*: This model is based on the heterogeneous architecture with an ideal energy-efficiency predictor, where all future behavior of the application as well as the power and performance for various configurations are known in advance. Since the Oracle scheme provides the upper bound for energy-efficiency, it is used to normalize and compare the other schemes.
- *$QLRM_{PCE}$*: This scheme is based on the proposed PCE-aware quantile linear regression model to estimate the EDP for various core sizes, frequency/voltage points, and number of threads for a given voltage regulator PCE.
- *Elastic-Core* [3]: This dynamic scheme proposed recently uses a linear regression model to predict the power and performance of single-threaded applications as a function of core type and frequency settings. However, unlike the model in this paper, the impact of PCE is not accounted for. In addition, although Elastic-Core does not account for the number of threads, the thread counts in this paper are set to maximum values to better evaluate the model by fairly comparing it against a recent dynamic scheduling solution.
- *Performance Aggressive Scheduling (PAS)*: In this scheme, all tuning parameters are set to maximum values to achieve the maximum performance. Therefore, each application is executed on big cores operating at 2.8 GHz and with 4 threads.
- *Power Minimized Scheduling (PMS)*: This scheduling scheme attempts to minimize power consumption. The application is running on a little core and the frequency and number of threads are set to minimum values.

The energy-efficiency of the studied applications normalized to the Oracle model with a fully efficient VR is shown in Fig. 8. The $QLRM_{PCE}$ on average achieves close to 95% efficiency as compared to the Oracle model. The $QLRM_{PCE}$ has improved energy-efficiency as compared to the Elastic-Core and PAS schemes by an average of respectively, 10% and 30% across all benchmarks, respectively. The average energy-efficiency of different scheduling schemes across various studied PCE gap models is shown in Fig. 9. By increasing the PCE gap, the energy-efficiency of the Elastic-Core, PAS, and PMS scheduling schemes diminishes. For the largest PCE gap, the proposed $QLRM_{PCE}$ outperforms a state of the art solution, the Elastic-Core, in energy-efficiency by 60%. The results verify the efficacy of the proposed prediction model and the effectiveness of the proposed scheduling scheme to harness the power an HMP for enhanced energy-efficiency.

VI. Conclusion

Emerging heterogeneous multicore architectures are complex processors with various tuning optimization knobs for improving performance and energy-efficiency. Scheduling multithreaded applications in these architectures is a challenging problem given the various optimization parameters at the application (number of running threads),

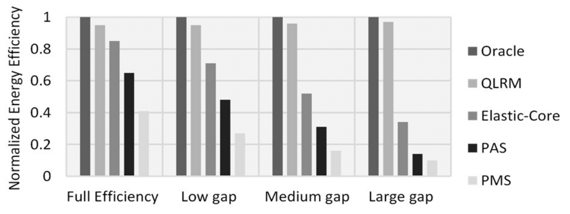


Fig. 9. Average energy-efficiency results of different scheduling schemes with respect to various PCE models.

system (operating voltage and frequency), and architecture (core type- big vs. little) levels. In addition, unlike homogeneous architectures, the efficiency of on-chip voltage regulators and the power conversion efficiency gap between the big and little cores in these architectures are critical parameters that must be accounted for. The interplay among the tuning parameters and the influence each has on the energy-efficiency, make the scheduling and tuning of the application even more challenging. In this paper, a PCE-aware scheduling and tuning solution is developed that highlights the importance of accounting for the PCE of big and little core to find the appropriate core type that optimizes the EDP. A predictive model is developed for estimating the energy-efficiency of multithreaded applications. Based on the predictive model, a scheduling scheme is developed for effective mapping of multithreaded applications to an HMP by setting the tuning parameters to maximize the energy-efficiency. The results indicate that the proposed scheduling scheme achieves on average close to 95% efficiency as compared to the Oracle scheduler.

References

[1] H. Homayoun et al., "Dynamically heterogeneous cores through 3D resource pooling," *HPCA-12*, pp. 1-12, Feb. 2012.
[2] C. Bienia et al., "Parsec 2.0: A new benchmark suite for chip multiprocessors" *MoBS-09*, June 2009.
[3] M. K. Tavana et al., "ElasticCore: enabling dynamic heterogeneity with joint core and voltage/frequency scaling," *DAC-15*, pp. 1-6, June 2015.
[4] K. V. Craeynest et al., "Scheduling heterogeneous multi-cores through performance impact estimation (PIE)," *ISCA-12*, pp. 213-224, 2012.
[5] K. Pusukuri et al.,"Thread reinforcer: dynamically determining number of threads via OS-level monitoring," *IISWC-11*, pp. 116-125, 2011.
[6] N. Chitlur et al., "QuickIA: Exploring heterogeneous architectures on real prototypes", *HPCA-12*, pp. 1-8, Feb. 2012.
[7] NVidia. The benefits of multiple CPU cores in mobile devices. http://www.nvidia.com/content/PDF/tegra_white_papers/Benefits-of-Multi-core-CPUs-in-Mobile-Devices_Ver1.2.pdf, 2010.
[8] P. Greenhalgh. Big.LITTLE processing with ARM Cortex_A15 & Cortex_A7:http://www.arm.com/files/downloads/big_LITTLE_Final_Final.pdf, Sept. 2011.
[9] G. Liu et al., "Dynamic thread mapping for high performance, power-efficient heterogeneous many-core systems," *ICCD-13*, pp. 54-61, 2013.
[10] J. Cong et al., "Energy-efficient scheduling on heterogeneous multi-core architectures," *ISLPED-12*, pp. 345-350, 2012.
[11] A. Lukefahr et al., "Composite cores: Pushing heterogeneity into a core," *MICRO-45*, pp. 317-328, 2012.
[12] Nikov, Krastin et al.. "Evaluation of hybrid run-time power models for the ARM big. LITTLE architecture," *EUC-15*, 2015.
[13] T. E. Carlson et al., "An evaluation of high-level mechanistic core models," *TACO-14*, vol. 11, Oct. 2014.
[14] S. C. Woo et al., "The SPLASH-2 programs: characterization and methodological considerations," *ISCA-95*, pp. 24-36, 1995.
[15] D.Pathak et al., "Energy Efficient On-Chip Power Delivery with Run-Time Voltage Regulator Clustering," *ISCAS-16*, pp. 1210-1213, May 2016.
[16] H. Asghari-Moghadam et al., "VR-Scale: Runtime dynamic phase Scaling of processor voltage regulators for improving power effciency," *DAC-16*, pp. 151-156, June 2016.
[17] M. Becchi et al., "Dynamic thread assignment on heterogeneous multiprocessor architectures," *ACM CF*-06, pp. 29-40, May 2006.
[18] C. Kim et al., "Composable lightweight processors," *MICRO*, pp. 381-394, Dec. 2007.
[19] B.C. Lee et al., "Accurate and efficient regression modeling for microarchitectural performance and power prediction," *SIGPLAN'06*, pp. 185-1984, Oct. 2006.
[20] R. Koenker, "Quantile regression," No. 38, Cambridge university press.
[21] R. Kumar et al., "SingleISA Heterogeneous Multi-core Architectures for Multithreaded Workload Performance," *ISCA-04*, pp. 64, June 2004.
[22] H. Sayadi et al, "Scheduling multithreaded applications onto heterogeneous composite cores architecture," *IGSC-17*, October 2017.
[23] H. Sayadi et al, "Machine learning-based approaches for energy-efficiency prediction and scheduling in composite cores architectures," *ICCD-17*, Nov. 2017.