

Energy Efficiency Platform Characterization for Heterogeneous Multicore Architectures

Hergys Rexha

Faculty of Science and Engineering
Åbo Akademi University
Turku, Finland
hrexha@abo.fi

Sébastien Lafond

Faculty of Science and Engineering
Åbo Akademi University
Turku, Finland
slafond@abo.fi

Abstract—Runtime estimation of power dissipation and performance is crucial in every computing platform. In mobile systems, a special focus is set on energy efficiency in order to achieve the longest possible battery life and at the same time adhering to performance requirements. Powered by heterogeneous SoC's, mobile systems are called to reach an energy efficient state of execution, with a runtime system or scheduler that requires knowledge on the current performance and power dissipation. Today, highly heterogeneous architectures provide many actuators to reach better efficiency, the effect of which is usually unknown at runtime. In this paper, we propose a fast approach to build an energy efficiency model based on hardware performance counters. Our approach obviates the need for power sensors present at the chip level and deals with high numbers of execution modes. In building the energy efficiency model we account for the change in temperature which, as we show, has an impact on the optimal energy efficiency choice. The proposed approach reduces significantly the time to characterize the energy efficiency of a Multiprocessor System-on-Chip (MPSoC) and includes the environment temperature as a variable in determining the energy efficiency.

Index Terms—MPSoC, energy efficiency models, platform configuration point, PMC, power models

I. INTRODUCTION

The past years have seen rapid development in the amount of data produced, processed and exchanged through computing systems, ranging from high-end server farms to simple household devices, and the trend of technology seems to fuel even more this direction. Based on electricity usage ascribed to Information and Communication Technology (ICT), it is predicted that by the end of 2030 this sector will use as much as 51% of global electricity production [5]. Following this scenario, by the year 2030, the only ICT industry will be responsible for up to 23% of the globally released greenhouse gas emissions [5]. A 2016 report [24] says that the US datacenters held 350 million terabytes of data in 2015, and by 2020 they will require 100TWh of electricity to operate. This is the equivalent of 7 nuclear power stations like Olkiluoto 3 in Finland. There is also an increase of datacenters capacity in Europe, with London, Frankfurt, Paris, and Amsterdam which grew their electricity consumption by 200MW in 2017. Countries like Ireland and Denmark in Europe are becoming a data base for the world's biggest tech companies and by the next 5 years promise to increase the power consumption by

1TW [12]. The emergence of the Internet of Things (IoT) with devices operating at the edge of the network, poses a new challenge to the Cloud to provide efficient service provisioning. IoT devices are low powered devices and their usage promises to decrease the overall power consumption by increasing energy efficiency, but their number could be overwhelming with the consequence of having a "rebound effect" [9]. Cisco predicts that by the year 2020 in the world will be 50 billion IoT devices, which is an order of magnitude bigger than the number of smartphones and tablets working today. So in this scenario, using the cloud services offered by large datacenters to receive the data generated by IoT devices will not be a sustainable solution in terms of cost, latency, and environmental impact [6]. Recently the idea of edge devices that provide the computation and storage closer to the source of data has been formulated under the term of Edge or Fog computing [25]. As an edge device example, we can mention smartphones, as intermediates between body sensors and the cloud services, gateways as intermediates for smart homes, or nano data centers that manage the caching or processing of video contents. By using these edge devices in the proximity of data sources, we could have as an end result in a reduction of energy consumption w.r.t. implementing the logic in the cloud, and at the same time keeping latency requirements of certain applications [17].

Therefore one key requirement of such computing systems is undoubtedly energy efficiency. Basically, this means that systems should minimize their energy consumption to complete the required task and achieve a satisfying energy proportionality [20]. One of the largest consumers of energy in computing environments is the CPU [8], which requires special attention especially in the multicore era. Today mobile devices are using the same CPU as traditional gateways or cloudlets in Edge Computing. The need to achieve energy efficiency in today's MPSoC is stringent, especially for mobile devices that operate on battery, and that is a clear scenario where the end user wants a better experience and longer battery life.

Workload variability makes the control of energy expenditure especially difficult in mobile CPUs. Mobile devices are not the only which require energy efficient solutions, but also cloud providers need to lower the energy cost of

computations and cooling [19]. Today large scale computing facilities are using energy as a resource to be scheduled and charge according to the energy consumption [14]. Heterogeneity shows a promise to increase the energy efficiency levels achieved in MPSoC, hence several paths have been followed by research and industry. For example, exploring heterogeneity inside the CPU chip by using multiple technologies with different power and performance characteristics or using cores that alternatively behave as out-of-order computing elements or as in-order cores [22]. Probably one of the most popular and researched types of heterogeneity is the one provided by different computing cores integrated into the same physical chip. This type of heterogeneity is the one where computing cores share the same Instruction Set Architecture (ISA) but have different microarchitectures. However, an intelligent use of these power and performance tradeoffs proves to be not a simple challenge [23]. Being able to predict the optimal choice between a number of hardware actuators such as the number of cores, type of core and operating performance point, or Dynamic Voltage and Frequency Scaling (DVFS), is a difficult task that must be handled well in order to achieve energy efficiency.

With asymmetric multiprocessing (AMP) architecture there is a better way to respond to the diversity of applications present in the mobile environment. We have compute-intensive applications which need to produce results in real time and must use fast cores in order to meet the deadlines. On the other side, background processes that may be memory bound require little computation and are more suitable to run on simple cores that achieve better levels of energy efficiency. Even within a single application, we have different “windows of activity” which may require varying levels of computing intensity, e.g. reading, scrolling, responding through different messages inside a social media application. Recently industry has moved towards increasing the level of heterogeneity found inside a single chip. From examples such as ARM big.LITTLE with two types of cores, to Mediatek tri-cluster MPSoC [16] which promise to increase performance and reduce power dissipation. DynamIQ from ARM [1] advances the concept of big.LITTLE by providing better flexibility in the cluster organization and frequency setting.

High levels of heterogeneity present in recently embedded architectures produce an increase in the design space exploration to find an efficient use of platform actuators. By increasing the number and type of cores and the number of voltages and frequency levels for each computing element, there is an increasing number of operating points on which the platform may perform. In this scenario making the right choice for execution could have a tremendous impact on energy efficiency. Temperature also has a major effect on the power dissipation of today’s systems [15], which makes it an important factor to account for in order to make the optimal energy efficient choice.

To manage efficiently the workload scenarios faced by mobile devices, edge devices in IoT, or nano data centers, there is a need to continuously monitor power data in order to

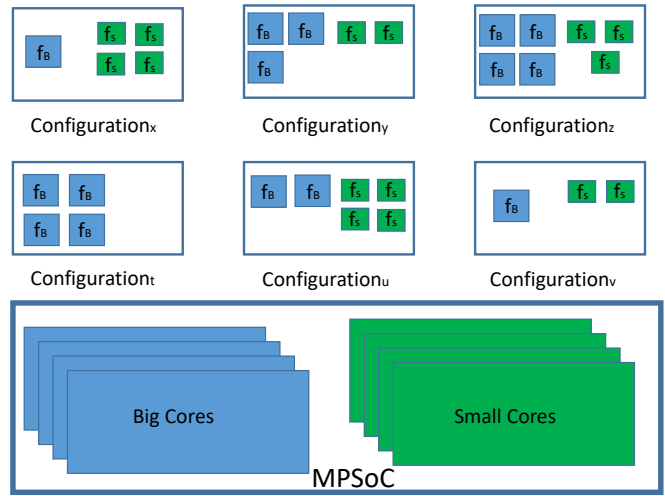


Fig. 1. Examples of possible platform configuration points in a multicore architecture

choose the optimal power and performance trade-off. Unfortunately, most of the hardware platforms today are not equipped with power sensors, which significantly complicates energy-efficient management of the system settings.

This paper follows our previous work which experimentally builds an energy efficiency model based on platform configuration points, for ARM big.LITTLE architecture [21]. As platform configuration point we denoted the set of platform actuators such as *number*, *type of core*, *core performance level or DVFS and core utilization level*. The model is derived by testing all the possible configuration points of the platform. Following the recent trend in platform complexity, this approach is difficult to apply in the case of the combinatorial explosion in the number of configuration points. The goal of this paper is to explore new approaches in providing knowledge of the platform energy efficiency to a runtime system based on the concept of platform configuration points. We redefine the set of parameters in the configuration point by removing *utilization level* from the aforementioned description. Meaning of the notion of platform configuration point is demonstrated with several examples (from x to v) in a multicore platform (Figure 1). In our energy efficiency model, we account for the environment temperature variable, which provides valuable information for the correct accounting of the CPU dissipated power. Knowing the large impact that static power has on the energy efficiency achieved in today’s CPUs the second purpose of this work is to build thermally aware energy efficiency models.

The contributions of this paper are the following:

- we propose an approach to characterize the energy efficiency of a hardware platform based on the notion of configuration points.
- we include environment temperature in the energy efficiency model and show the impact this variable has on the relative efficiency of the points from the model.

II. RELATED WORK

Exploring the usage of platform actuators for energy management was studied by different research works. The authors in [23], [10], and [18] all propose the creation of a runtime system which is able to manage the scheduling and mapping of threads dynamically with the objective of maximizing the energy efficiency of MPSoC. In [23] a load balancer schedules the workload in periodic time frames called *epochs*, wherein each, a set of actions are performed to set the threads in the appropriate core type. The platform considered is highly heterogeneous with 4 types of core and in each *epoch* the load balancer estimates the performance and power of every thread in each core type. This information is used by the internal algorithm to decide where to map the threads. Similarly, in [18] is proposed a runtime scheme which is used to schedule dynamically workloads in a MPSoC. The approach is based on the sense-decide-act policy and operates on an aggressive heterogeneous environment. It uses regression models for estimating performance and power of threads in different core type and also the contribution of a thread in a total load of a core. An evolutionary algorithm is used to decide in each term the scheduling of the threads. The authors in [10] propose a run-time task allocation approach called SPARTA which categorizes task in computing bound or memory bound and a heuristic that selects the configuration that achieves the requested throughput with the minimal power consumption. In these works is not considered the possibility of DVFS as a mechanism to reduce power consumption and also the hardware counters used for estimating performance are not easily found in real hardware platforms. Sensors for estimating the power consumption of different mapping decisions are not available in many of today's platforms. Finding the optimal configuration for executing workloads in a data-center in order to achieve better energy efficiency is the goal presented in [11]. Authors present a programming and execution platform called Empya that uses hardware and software techniques to determine the best trade-off between performance and energy consumption. The run-time system continuously monitors application performance and energy consumption through Running Average Power Limit (RAPL) registers. As actuators, the system operates on the number of threads to use and the power cap on the CPU. In contrast with this, our work focuses on heterogeneous platforms where for achieving energy efficiency we use actuators such as number, type of core and DVFS point. In [26] authors target again High-Performance Computing applications running on a single node with the goal of reducing the energy consumption by choosing the right configuration, which is composed of the number of cores and DVFS level. The work is based on the application-agnostic power model and the performance model of the application is obtained with a supervised learning method of regression. Frequency, number of cores and input size are used in the regression model. The methodology is clear and straightforward, but there is no mention of the performance requirement which is the value we trade off for

less energy consumption.

III. CMOS POWER DISSIPATION

CMOS technology has been mostly used in MPSoCs due to the fact that has quite good noise immunity and low heat production while the device is in operation mode. Power in these circuits can be divided into two categories: dynamic power and static power. Dynamic power is created by the circuit activity (transistor switching) and is dependent on the usage scenario, clock rates, and I/O activity. Switching power is dissipated during the transistor changing from 0 to 1 and vice versa, the dynamic power is defined as:

$$P_{\text{dynamic}} = \alpha * C * V_{DD}^2 * f_{clk} \quad (1)$$

where C is the load capacitance, V_{DD} is the source voltage, α is the activity factor and f is the operating frequency. Static power is dissipated due to the leakage currents on the transistors while they are in the "OFF" mode. There are several sources of the leakage current which are strongly influenced by the chip temperature. The dynamic part of the power dissipated from the chip is modeled by two terms in Equation 2, as a dynamic activity which relates to the active running workloads and the background activity that represents the system processes that run on the background. In Equation 3 the dynamic power is modeled by a single term due to the low power dissipated by background processes in the A7 cluster. Static power is modeled by the third term in Equation 2 and is dependent on temperature and the supply voltage. For the A7 cluster, there is no temperature sensor to monitor, hence the static part is modeled together with the dynamic power dissipation of background activity.

IV. PROPOSED APPROACH

Today embedded systems face a multitude of working scenarios that range from burst in high performance requests, to low power operation modes, going through the need to provide sustainable performance in thermally constrained situations. To do an efficient managing of such a number of use cases the runtime scheduling manager need to have refreshed information about the effect of changing different actuators on the running applications. Thus there is a need for an energy efficiency model which is based on the current runtime power data. The envisioned system diagram is shown in Figure 2, where our work in this paper is focused in providing the platform configuration points database for helping the scheduler decisions in reaching the optimal efficiency level of the running applications.

The work in this paper is based on power models for mobile CPUs based on hardware program counters (HPC). The methodology for building such models is adopted from [27], which presents a statistical method for identifying and using hardware counters. Their analyses propose the usage of counters which show a high correlation to power and have also the

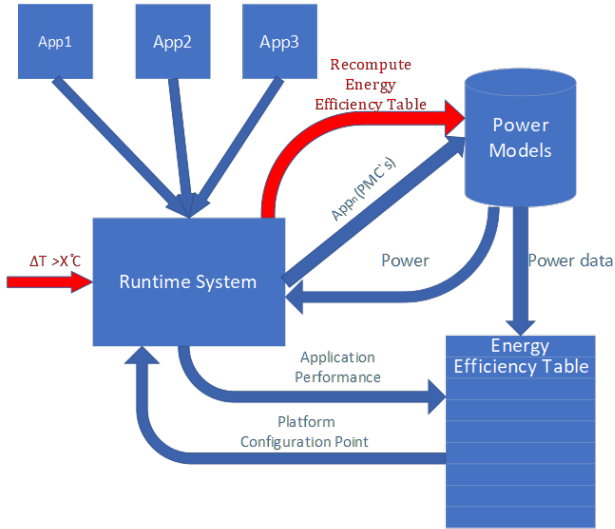


Fig. 2. Proposed Approach schematics.

TABLE I
HARDWARE EVENTS USED IN THE POWER MODELS

Event list		
Nr	ARM Cortex-A7	ARM Cortex-A15
1	L2D_CACHE_ACCESS:0x16	L2D_CACHE_LD:0x50
2	MEM_ACCESS:0x13	DP_SPEC:0x73
3	L1I_CACHE_ACCESS:0x14	L1I_CACHE_ACCESS:0x14
4	UNALIGNED_LDST:0x0F	UNALIGNED_LDST_SP:0x6A
5	CYCLE_COUNT:0x11	BUS_ACCESS:0x19
6		INST_SPEC:0x1B
7		CYCLE_COUNT:0x11

smallest multicollinearity. The authors in [27] show that this brings high model stability with an average error of 3,8%.

We start by building power models for two popular ARM v7a architecture CPU's, which are ARM Cortex-A7 and ARM Cortex-A15. The micro-architecture limits the number of events which can be sampled at once: 6 counters for A15 and 4 counters for A7 plus the cycle counter. The goal is to search for those events which have the highest correlation with power dissipation and at the same time show the smallest intercorrelation with each other. To have high model stability the predictors should be chosen to keep low levels of multicollinearity in multivariate models. First, is measured the correlation of all available events with the power, then the counters are divided into clusters which include events with high intercorrelation. Then, from each cluster is selected the event which has more impact on the power dissipation but keeping a low Variance Inflation Factor (VIF). The total amount of events for the A7 is 40 and for the A15 in 120, among these are selected 7 for the A15 and 5 for the A7. The events used in the models are general and can be found on most core types used in mobile systems. For each core type, the events are listed on Table I. The power for A15 and A7 is divided in dynamic and static, plus the background power which is related to the operating system activities.

The modelled formula for the power dissipation is showed in Equation 2 and 3,

$$P_{A15} = \underbrace{\left(\sum_{n=0}^{N-1} \beta_n E_n V_{DD}^2 f_{clk} \right)}_{\text{dynamic activity}} + \underbrace{\beta_b V_{DD}^2 f_{clk}}_{\text{BG dynamic}} + \underbrace{f(V_{DD}, T)}_{\text{static}} \quad (2)$$

$$P_{A7} = \underbrace{\left(\sum_{n=0}^{N-1} \beta_n E_n V_{DD}^2 f_{clk} \right)}_{\text{dynamic activity}} + \underbrace{f(V_{DD}, f_{clk})}_{\text{static and BG dynamic}} \quad (3)$$

where N is the number of events selected, β_n is the weight given to certain event, E_n is the number of events per second divided by the frequency (f_{clk}) in MHz, V_{DD} is the operating voltage and T is the temperature of the core.

The power model for the A15 has a thermal compensation term for calculating the static power and background dissipated power when the system is idling (Equation 2). In the power model for A7 the static and background power are included in the second term of Equation 3. This is related to the absence of a thermal monitoring sensor in the A7 cluster. We have calculated four sets of model coefficients for the parameters in each cluster, representing the power with a different number of cores for each CPU type. The model parameters for each core type are given in Tables II and III. In the tables, it is shown the event rate divided by the frequency in MHz, the weight given to each coefficient and the statistical significance. In some model terms, f and V are respectively the operating frequency and voltage of each cluster (Table IV). The event rates are divided by the operating frequency in order to avoid correlation with it in the first term of power equations. The power models need to be obtained only once by running on the target platform a set of embedded representative workloads which we call platform characterization set. After obtaining the power model we compute the energy efficiency table which provides a sort of database of all the possible platform configuration points and the resulting performance, power and energy efficiency values. By having this information the runtime system is able to make decisions about the mapping of a certain application with regard of the performance. If there is a change in the environment temperature above a certain threshold, then the power dissipation can be recomputed and the table is redefined for the new thermal level.

These models are build by running the characterization workload set in each of the operating points of both CPUs. The set contains workloads that test different levels of the microarchitecture and memory subsystem. In part is composed of real applications from the embedded domain, and for the other part synthetic benchmarks designed to stress specific parts of the CPU. Having the power models and by measuring the performance in terms on instructions per second (IPS) we can obtain an energy efficiency model of the platform. The model is presented as a table that lists all the platform configuration points with the energy efficiency levels achieved in

TABLE II
MODEL PARAMETERS AND P-VALUES FOR THE A15

Nr	Coefficient	Weight	p-Value
1	Intercept	-5e-4	p<e-4
2	$EPH_0x11 * f * V^2$	7.9e-10	p<e-4
3	$(EPH_0x1b - EPH_0x73) * f * V^2$	e-10	p<e-4
4	$EPH_0x50 * f * V^2$	8.7e-9	p<e-4
5	$EPH_0x6a * f * V^2$	e-8	p<e-4
6	$EPH_0x73 * f * V^2$	2.6e-11	p<2e-3
7	$EPH_0x14 * f * V^2$	6.4e-11	p<e-3
8	$EPH_0x19 * f * V^2$	1.9e-9	p<e-4
9	V	0.17	p<e-4
10	$f * V^2$	1.6e-4	p<e-4
11	T	2.3e-2	p<e-3
12	T^2	2.9e-4	p<4e-3
13	$V * T^2$	-3.5e-5	p<e-3
14	$V * T$	1.1e-2	p<e-3

TABLE III
MODEL PARAMETERS AND P-VALUES FOR THE A7

Nr	Coefficient	Weight	p-Value
1	Intercept	-7.2e-4	p<0.003
2	$EPH_0x11 * f * V^2$	1.9e-10	p<e-4
3	$EPH_0x14 * f * V^2$	2.2e-10	p<e-4
4	$EPH_0x13 * f * V^2$	4.3e-10	p<e-4
5	$EPH_0x16 * f * V^2$	1.4e-9	p<e-4
6	$EPH_0x0f * f * V^2$	9.4e-11	p<0.0004

terms of instructions per Joule, performance point (instructions per second) and the power dissipation (W). The table is used to decide the optimal configuration point for an application that has defined performance requirements. Once an application is submitted into the system or is resumed by the scheduler, the runtime system can sample the hardware counters in a single frequency level and scans the table to find the optimal configuration point, to run the application, in terms of energy efficiency. In this work, we consider multi-threaded applications, which matches our methodology of achieving optimal levels of energy efficiency by using configuration points that possibly use several cores. In the case where the performance requirement of the application changes, the control logic of the runtime system can select another configuration point that provides the requested performance level and has a high level of energy efficiency. When the temperature of the environment changes above a certain threshold, the power model can be used to recompute the energy efficiency table in accordance with the new temperature conditions. A temperature increase in the outside environment produces an increased level of static power in the CPU, which affects the relative efficiencies of the configurations inside the energy efficiency table. The runtime system can continuously monitor the power usage of the running application in order to not exceed the Thermal Design Power (TDP) of the CPU. By sampling the performance counters of each running application the power model shows the power dissipation at runtime of the running applications, thus the runtime system can make a decision of reducing the power dissipation of certain applications by choosing another configuration point from the system.

The runtime system inputs temperature variations inside the model and can recompute the energy efficiency table by taking into account the new level of static power. The new table needs to be searched for configuration points that satisfy the performance request with the highest level of efficiency. A basic schematic of the proposed approach is given in Figure 2.

V. EXPERIMENTAL SETUP

To evaluate our approach we used an ODROID XU3 development board from HARDKERNEL. The application processor implements the ARM big.LITTLE architecture with two clusters composed of 4 cores each. The big cluster consists of a high-performance Cortex-A15 quad-core block, and a low power Cortex-A7 quad-core CPU. The board description is complete with a Mali-T628 GPU and 2GB LPDDR3 of memory. The board contains 4 current sensors that offer the possibility to measure power dissipation in four different domains: big cluster (A15), LITTLE cluster (A7), GPU and memory. Besides this, the board contains 4 temperature sensors for the cores in the big cluster and one temperature sensor for the GPU. The characteristics of the hardware can be found in Table IV.

TABLE IV
CHARACTERISTICS OF THE EXPERIMENTAL BOARD

Characteristic	ODROID Development Board
Model	XU3
SoC	Exynos 5422 Octa core
CPU's	Cortex-A15/A7
cores	4 + 4
Frequency A7 (MHz)	
min	200
max	1400
Frequency A15 (MHz)	
min	200
max	2000
Voltage A7 (V)	
min	0.9
max	1.24
Voltage A15 (V)	
min	0.9
max	1.36

To build the power model we used a set of benchmarks from different application domains. We call the training set as the platform characterization workloads. In the platform characterization set we include a sequence of 76 workloads which consists of a collection of synthetic and real world applications from Roy Longbottom [4], PARSEC [7], CoremarkPro [2], ParMiBench [13] and Multibench [3]. A full list of the used workloads is in Table V.

The choice of the workload set is based on the idea of all-inclusiveness of applications that characterize the embedded systems domain.

Experiments were conducted in different environments to account for the outside temperature change in the SoC power dissipation. The goal here is to evaluate the change in the energy efficiency table in accordance with temperature. For the first environment, the board fan is running with 100% speed with the system located in a highly refrigerated environment.

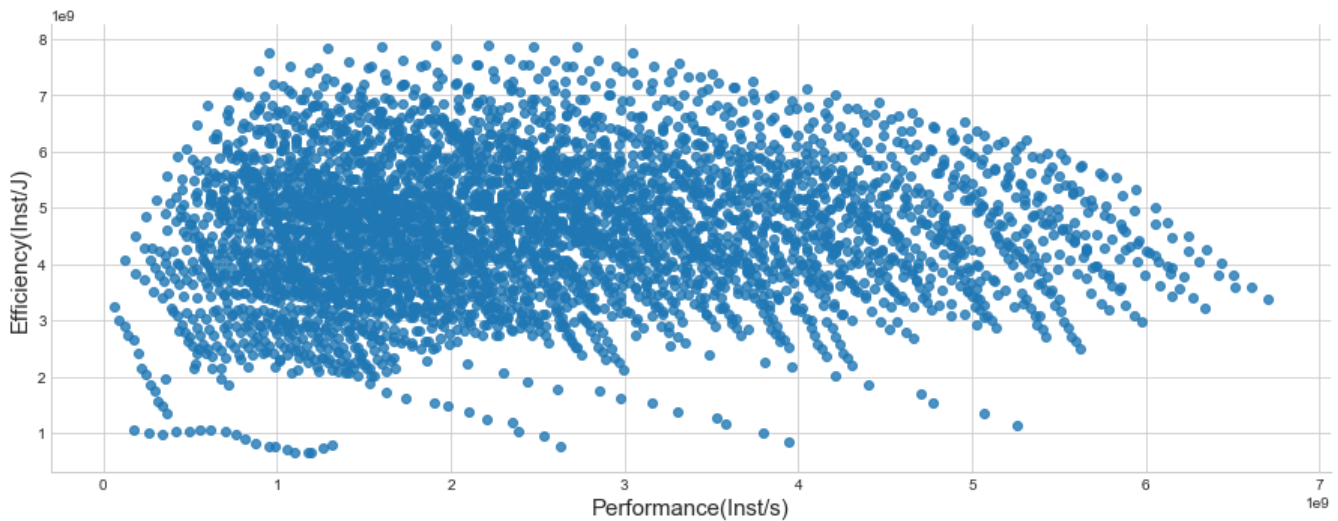


Fig. 3. Configuration points from the model

In the second case, the board is working with the fan disabled in a normal outside temperature to account for a high temperature outside the environment. In the third case, the board is working with the fan always on in a normal environment, to justify the middle case. In Table, VI on Section V we will show the result of the energy efficiency table computed in different environments.

VI. RESULTS

By using the power and performance models defined previously we are able to derive an energy efficiency model which is based on platform configuration points. In Figure 3 we show the efficiency of all configuration points from the model. Each point describes a single configuration that provides a certain level of performance in terms of instructions per second and energy efficiency. By going towards high levels of performance we notice a decrease in the density of the points. This means that fewer options for achieving good energy efficiency levels. The list of configurations is organized as an energy efficiency table that lists all possible configuration points with their efficiency and performance. An example of the table derived from the workloads in the training set of the power model is shown in Table VII. By searching inside the table we find several sets of configuration points that provide the same performance but with different energy efficiency levels, some of the sets are shown in Figure 4. First usage of the table would be the one for choosing the optimal configuration point based on a certain requirement for the performance level. As it is shown by Figure 4, it is possible to gain in terms of energy efficiency if we make the right choice for the configuration point. As a second objective of our work, we wanted to test the effect of temperature on the relative energy efficiency of configuration points in the model. For testing thermal effects on the efficiency model, we choose to run a testing application with the system located in different environments. We run Basicmath application from the ParMiBench suite [13]. In

environment 1, the system running in a highly refrigerated environment (we call it “cold” case). In Environment 2, the system is running without a fan with an outside temperature of 25°C (we call it “hot” case). Environment 3, consists of the system running on a 25°C outside temperature with the fan always on at 100% speed (we call it “middle” case). We noticed the relative order of configuration points changes between the environments and so does the energy efficiency levels achieved.

The top rows of the energy efficiency table for different temperature environments are shown in Table VI. Different temperature levels produce different order of configuration points and efficiency levels achieved. This shows that there is a need to change the platform configuration point when the temperature changes significantly, in order to keep the high levels of energy efficiency.

In Figure 5 we show a possible runtime scenario. We are running Basicmath test application with a required level of performance such as e.g. 1,61E+9 inst/s in a system with a temperature t_1 , according to the model the optimal configuration point for this performance level is composed by 2a7@400MHz + 4a15@500MHz. In the case, the temperature increases to t_2 , then the efficiency of that configuration point decreases and thus we need to reconfigure with the new table that shows that we should execute the application by using the following configuration 4a7@700MHz + 4a15@200MHz. Another example is shown with the performance requirement of 3,27E+9 inst/s, where again there is a need for reconfiguration in order to keep high levels of energy efficiency.

The change in the environment temperature of the system (from “cold” to “hot”) produces large differences in the energy efficiency levels that the model defines as an optimal configuration point for the required performance. By looking at the first 100 highly energy efficient configurations in the energy efficiency table, we find few test cases, whereby changing the configuration point when the system temperature changes the

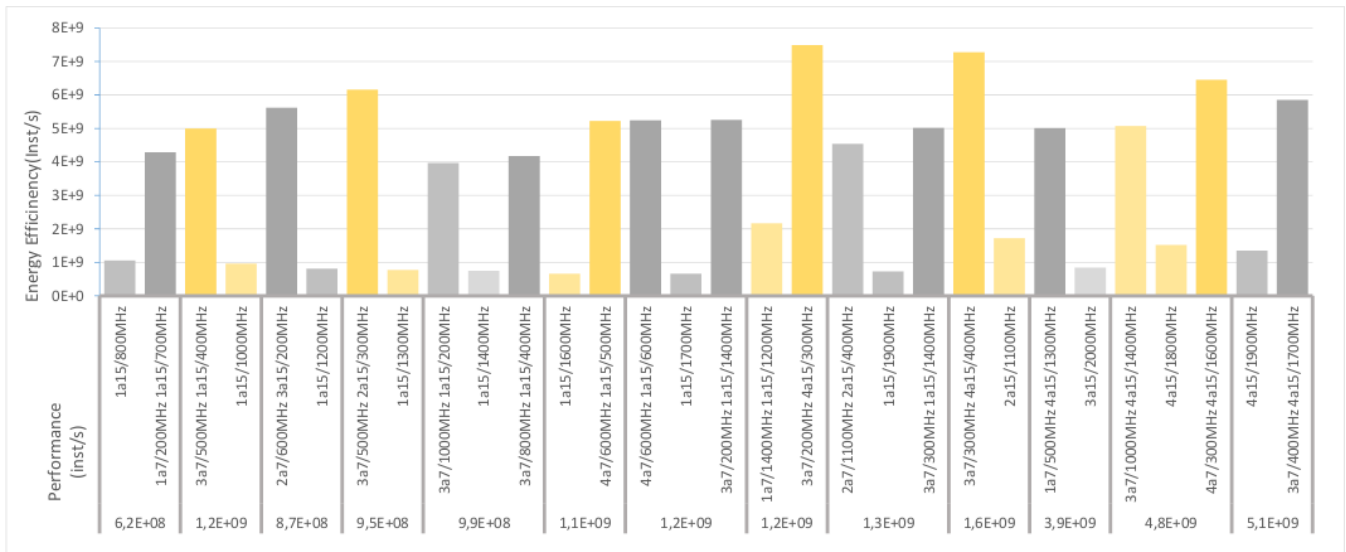


Fig. 4. List of configuration points grouped in different performance classes

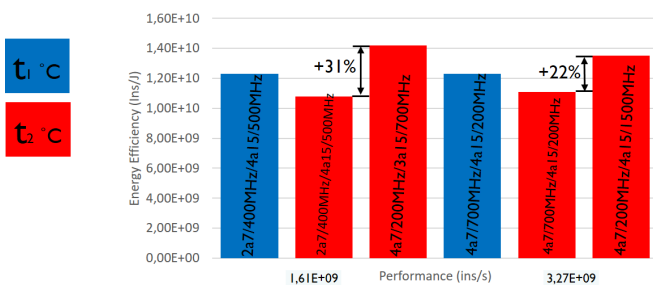


Fig. 5. Reconfigure examples in two temperature environments

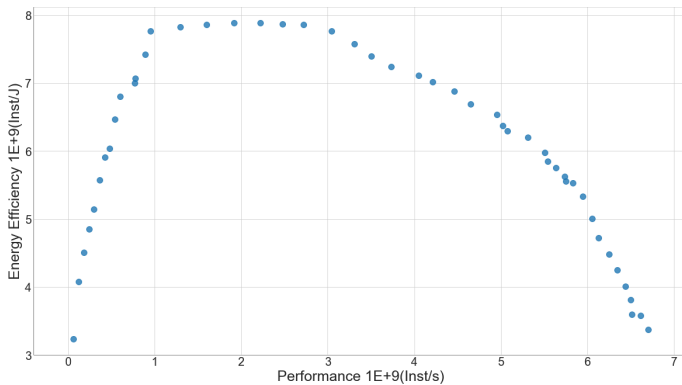


Fig. 6. Configuration points with high energy efficiency levels

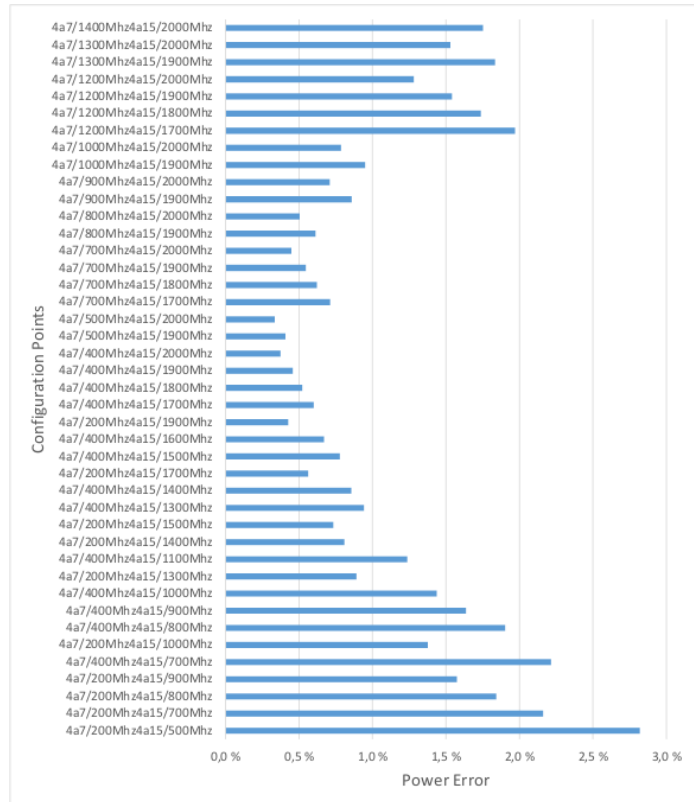


Fig. 7. Power errors for configuration points with high level of energy efficiency

gain in terms of energy efficiency is up to 33%. By searching for new target reconfiguration points we account for the same performance or 5% bigger. An interesting observation can be noticed in Figure 3 where all points are plotted in the energy efficiency and performance graph. If we take the points from

the upper outer layer of the scatter plot we have a situation like in Figure 6. Those points show the configurations with the optimal energy efficiency for a certain level of performance at a defined temperature. Or otherwise, we can think of the graph as the result of scanning the model from the lowest

TABLE V
PLATFORM CHARACTERIZATION SET

List of benchmarks		
Suite	Workload	
CoremarkPro	core	
	linear_alg-mid-100x100-sp	
	loops-all-mid-10k-sp	
	nnet_test	
	parser-125k	
	radix2-big-64k	
MultiBench	sha-test	
	zip-test	
	4M-check	
	4M-check-reassembly	
	4M-check-reassembly-tcp	
	4M-check-reassembly-tcp-cmykw2-rotatew2	
	4M-check-reassembly-tcp-x264w2	
	4M-cmykw2	
	4M-cmykw2-rotatew2	
	4M-reassembly	
	4M-rotatew2	
	4M-tcp-mixed	
	4M-x264w2	
	empty-wld	
	iDCT-4M	
	iDCT-4Mw1	
	ippkcheck-4M	
	ippkcheck-4Mw1	
	ipres-4M	
	ipres-4Mw1	
	md5-4M	
	md5-4Mw1	
	rgbcmk-4M	
	rgbcmk-4Mw1	
	rotate-4Ms1	
	rotate-4Ms1w1	
	rotate-4Ms64	
	rotate-4Ms64w1	
	x264-4Mq	
	x264-4Mqw1	
	MiBench	automotive/qsrt
		network/dijkstra
		consumer/typeset
telecomm/adpcm		
Parsec-3.0	blackscholes	
	bodytrack	
	cannal	
	dedup	
	ferret	
	fluidanimate	
	freqmine	
streamcluster		
swaptions		
ParmiBench	Office/stringsearch	
	Network/Patricia/Parallel	
	Automotive/Susan/Parallel	
	Automotive/Bitcount/Parallel	
	Network/Dijkstra/Parallel	
Office/stringsearch/Parallel		
Roy-Longbottom	rl-linpack-neon	
	rl-linpack-FSSP	
	rl-whetstone	
	rl-busspeed	
	rl-dhystone	
Lmbench	lat_ctx	
	lat_fs	
	lat_ops	
	lat_proc	
	lat_fifo	
	lat_http	
	lat_pagefault	
	lat_select	
	lat_sem	
	lat_unix_connect	
	lat_mem_rd	
	bw_mem	
	tlb_lmb3-tlb	
	line	
Whetstone	whetstone	
Drystone	dhystone	

TABLE VI
TOP ENERGY EFFICIENCY CONFIGURATIONS FOR THREE ENVIRONMENTS

Temperature Environment 1			
Configuration	Energy Efficiency (Inst/J)	Power(W)	Performance(Inst/s)
4a7/200MHz4a15/500MHz	1,517e+10	0,465	1,61e+09
4a7/200MHz4a15/700MHz	1,515e+10	0,599	2e+09
4a7/200MHz4a15/400MHz	1,512e+10	0,382	1,39e+09
4a7/200MHz4a15/300MHz	1,511e+10	0,305	1,17e+09
4a7/200MHz4a15/200MHz	1,50e+10	0,219	9,37e+08
....
Temperature Environment 2			
4a7/200MHz3a15/300MHz	1,424e+10	0,333	9,92e+08
4a7/200MHz3a15/500MHz	1,421e+10	0,518	1,32e+09
4a7/200MHz3a15/400MHz	1,420e+10	0,428	1,15e+09
4a7/200MHz3a15/600MHz	1,420e+10	0,608	1,48e+09
4a7/200MHz3a15/700MHz	1,416e+10	0,697	1,61e+09
....
Temperature Environment 3			
4a7/200MHz4a15/600MHz	1,49e+10	0,586	1,82e+09
4a7/200MHz4a15/400MHz	1,49e+10	0,415	1,39e+09
4a7/200MHz3a15/700MHz	1,49e+10	0,668	2e+09
4a7/200MHz3a15/500MHz	1,480e+10	0,511	1,61e+09
4a7/200MHz3a15/300MHz	1,486e+10	0,337	1,17e+09
....

TABLE VII
ORDERED ENERGY EFFICIENCY TABLE .

C	$C(N_i / F_i / N_b / F_b)$	Perf.(inst/s)	P_{avg} (W)	Efficiency(inst/J)
1	4a7/200MHz/4a15/600MHz	2.219115e+09	0.699744	7.889801e+09
2	4a7/200MHz/4a15/500MHz	1.916094e+09	0.600826	7.885497e+09
3	4a7/200MHz/4a15/700MHz	2.475814e+09	0.788427	7.872383e+09
4	4a7/200MHz/4a15/800MHz	2.723064e+09	0.873142	7.861730e+09
5	4a7/200MHz/4a15/400MHz	1.601398e+09	0.501352	7.857119e+09
6	4a7/200MHz/4a15/300MHz	1.294310e+09	0.402370	7.830159e+09
7	4a7/200MHz/4a15/900MHz	3.042998e+09	1.010040	7.765476e+09
8	4a7/200MHz/4a15/200MHz	9.541939e+08	0.293673	7.763320e+09
9	4a7/300MHz 4a15/600MHz	2.338974e+09	0.728441	7.647120e+09
10	4a7/300MHz 4a15/500MHz	2.035953e+09	0.629523	7.642816e+09
11	4a7/300MHz 4a15/700MHz	2.595672e+09	0.817124	7.629703e+09
12	4a7/300MHz 4a15/800MHz	2.842923e+09	0.901839	7.619049e+09
13	4a7/300MHz 4a15/400MHz	1.721256e+09	0.530049	7.614439e+09
14	4a7/300MHz 4a15/300MHz	1.414169e+09	0.431067	7.587478e+09
15	4a7/200MHz 4a15/1000MHz	3.310742e+09	1.173238	7.580142e+09
-	-	-	-	-
4078	1a15/1800MHz	1.193975e+09	1.795146	6.651129e+08
4079	1a15/1700MHz	1.176482e+09	1.776230	6.623477e+08
4080	1a15/1600MHz	1.101565e+09	1.670471	6.594337e+08

performance point and keeping only those points which have higher performance and the highest possible level of energy efficiency. As a further validation of our approach, we measure in percentage the difference between the predicted power dissipation and the measured power in configuration points with high levels of energy efficiency. The results are shown in Figure 7, where we notice the highest error is 2,82%. We measure the model errors in configurations that provide the highest levels of energy efficiency for different performance levels. These are more intriguing configuration points, which give the best of the platform’s energy efficiency. Knowing that most of the time these points will be used as configuration options, having a low error rate from the model is very useful.

VII. CONCLUSION

In this work, we present an approach for building an energy efficiency model which is based on platform configuration points. The target of the approach are heterogeneous platforms which are continuously increasing the depth of heterogeneity. The model is based on hardware performance counters which are widely available in today’s CPU architectures. The set of workloads for building the model is representative of the

embedded domain which has shown to be more critical to the energy efficient application execution. But also, the training set, in inclusive of the IoT world. The novelty of this approach compared to previous works is that it doesn't necessarily need power sensors for measuring the power dissipation in each configuration point, but by sampling the counters on one configuration point we can characterize the efficiency of other configuration points. From all the points in the model, we show that less than 1% of them (see points in Figure 7) represent the highest levels of energy efficiency possible, in all the performance spectrum offered by the platform. Also, we include the environment temperature as a variable for defining the need for application reconfiguration. As we show by the tests if the temperature changes, by reconfiguring the application execution we can gain up to 33% in terms of energy efficiency.

REFERENCES

- [1] Arm DynamIQ Technology for the next era of compute - Processors blog - Processors - Arm Community. <https://community.arm.com/processors/b/blog/posts/arm-dynamiq-technology-for-the-next-era-of-compute>.
- [2] CPU Benchmark - CoreMark-PRO - EEMBC Embedded Microprocessor Benchmark Consortium. <https://www.eembc.org/coremark-pro/index.php>.
- [3] EEMBC - MultiBench - Multicore Benchmark. <https://www.eembc.org/multibench/>.
- [4] Roy Longbottom's PC Benchmark Collection - Free PC Benchmarks. <http://www.roylongbottom.org.uk/>.
- [5] Anders S. G. Andrae and Tomas Edler. On global electricity usage of communication technology: Trends to 2030. *Challenges*, 6(1):117–157, 2015.
- [6] M. Ashouri, P. Davidsson, and R. Spalazzese. Cloud, edge, or both? towards decision support for designing iot applications. In *2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pages 155–162, Oct 2018.
- [7] Christian Bienia. *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, January 2011.
- [8] W. L. Bircher and L. K. John. Complete System Power Estimation Using Processor Performance Events. *IEEE Transactions on Computers*, 61(4):563–577, April 2012.
- [9] Peter M. Corcoran. Third time is the charm - why the world just might be ready for the internet of things this time around. *CoRR*, abs/1704.00384, 2017.
- [10] Bryan Donyanavard, Tiago Mück, Santanu Sarma, and Nikil Dutt. SPARTA: Runtime Task Allocation for Energy Efficient Heterogeneous Many-cores. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, CODES '16, pages 27:1–27:10, New York, NY, USA, 2016. ACM.
- [11] C. Eibel, T. Do, R. Meissner, and T. Distler. Empya: Saving Energy in the Face of Varying Workloads. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pages 134–140, April 2018.
- [12] EIRGRID. All-island generation capacity statement 2017-2026, 2017.
- [13] S. M. Z. Iqbal, Y. Liang, and H. Grahn. ParMiBench - An Open-Source Benchmark for Embedded Multiprocessor Systems. *IEEE Computer Architecture Letters*, 9(2):45–48, February 2010.
- [14] V. Jimenez, F. Cazorla, R. Gioiosa, E. Kursun, C. Isci, A. Buyuktosunoglu, P. Bose, and M. Valero. Energy-Aware Accounting and Billing in Large-Scale Computing Facilities. *IEEE Micro*, 31(3):60–71, May 2011.
- [15] J. S. Lee, K. Skadron, and S. W. Chung. Predictive Temperature-Aware DVFS. *IEEE Transactions on Computers*, 59(1):127–133, January 2010.
- [16] H. Mair, E. Wang, A. Wang, P. Kao, Y. Tsai, S. Gururajao, R. Lagerquist, J. Son, G. Gammie, G. Lin, A. Thippana, K. Li, M. Rahman, W. Kuo, D. Yen, Y. Zhuang, U. Fu, H. Wang, M. Peng, C. Wu, T. Dosluoglu, A. Gelman, D. Dia, G. Gurumurthy, T. Hsieh, W. Lin, R. Tzeng, J. Wu, C. Wang, and U. Ko. 3.4 A 10nm FinFET 2.8GHz tri-gear deca-core CPU complex with optimized power-delivery network for mobile SoC performance. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 56–57, February 2017.
- [17] Jozef Mocnej, Martin Miškuf, Peter Papcun, and Iveta Zolotová. Impact of Edge Computing Paradigm on Energy Consumption in IoT. *IFAC-PapersOnLine*, 51(6):162–167, 2018. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018 Citation Key: MOCNEJ2018162.
- [18] Tiago Mück, Santanu Sarma, and Nikil Dutt. Run-DMC: Runtime Dynamic Heterogeneous Multicore Performance and Power Estimation for Energy Efficiency. In *Proceedings of the 10th International Conference on Hardware/Software Codesign and System Synthesis*, CODES '15, pages 173–182, Piscataway, NJ, USA, 2015. IEEE Press.
- [19] V. Petrucci, M. A. Laurenzano, J. Doherty, Y. Zhang, D. Mossé, J. Mars, and L. Tang. Octopus-Man: QoS-driven task management for heterogeneous multicores in warehouse-scale computers. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 246–258, February 2015.
- [20] L. Ramapantulu, D. Loghin, and Y. M. Teo. On Energy Proportionality and Time-Energy Performance of Heterogeneous Clusters. In *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 221–230, September 2016.
- [21] Hergys Rexha, Simon Holmbacka, and Sebastien Lafond. Core Level Utilization for Achieving Energy Efficiency in Heterogeneous Systems. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 401–407, St. Petersburg, Russia, 2017. IEEE.
- [22] V. Saripalli, G. Sun, A. Mishra, Y. Xie, S. Datta, and V. Narayanan. Exploiting Heterogeneity for Energy Efficiency in Chip Multiprocessors. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 1(2):109–119, June 2011.
- [23] Santanu Sarma, T. Muck, Luis A. D. Bathen, N. Dutt, and A. Nicolau. SmartBalance: A Sensing-driven Linux Load Balancer for Energy Efficiency of Heterogeneous MPSoCs. In *Proceedings of the 52Nd Annual Design Automation Conference, DAC '15*, pages 109:1–109:6, New York, NY, USA, 2015. ACM.
- [24] Arman Shehabi, Sarah Josephine Smith, Dale A. Sartor, Richard E. Brown, Magnus Herrlin, Jonathan G. Koomey, Eric R. Masanet, Nathaniel Horner, Inês Lima Azevedo, and William Lintner. United states data center energy usage report. Technical report, 06/2016 2016.
- [25] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016.
- [26] Vitor R. G. Silva, Alex Furtunato, Kyriakos Georgiou, Kerstin Eder, and Samuel Xavier-de-Souza. Energy-Optimal Configurations for Single-Node HPC Applications. *arXiv:1805.00998 [cs]*, May 2018.
- [27] M. J. Walker, S. Diestelhorst, A. Hansson, A. K. Das, S. Yang, B. M. Al-Hashimi, and G. V. Merrett. Accurate and Stable Run-Time Power Modeling for Mobile and Embedded CPUs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(1):106–119, January 2017.