Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Gonçalo Marques Pestana

Energy Efficiency in High Throughput Computing

Tools, techniques and experiments

Master's Thesis
Espoo, 14 December, 2015

DRAFT! — May 28, 2016 — DRAFT!

Supervisors: Professor Jukka K. Nurminen
Advisor: Zhonghong Ou (Post-Doc.)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Gonçalo Marques Pestana

Title:
Energy Efficiency in High Throughput Computing Tools, techniques and experiments

Date:	14 December, 2015	**Pages:**	72
Major:	Data Communication Software	**Code:**	T-110
Supervisors:	Professor Jukka K. Nurminen		
Advisor:	Zhonghong Ou (Post-Doc.)		

The volume of data to process and store in high throughput computing (HTC) and scientific computing continues increasing many-fold every year. Consequently, the energy consumption of data centers and similar facilities is raising economical and environmental concerns. Thus, it is of paramount importance to improve energy efficiency in such environments. This thesis focuses on understanding how to improve energy efficiency in scientific computing and HTC. For this purpose we conducted research on tools and techniques to measure power consumption, as well as experiments to understand if low-energy processing architectures are suitable for HTC. To that purpose, we compared the energy efficiency of ARM and Intel architectures under authentic scientific workloads. Finally, we used the results to develop an algorithm that schedules tasks among ARM and Intel machines in a dynamic electricity pricing market, in order to optimally lower the overall electricity bill. Our contributions are three-fold: The results of the study indicate that ARM presents potential for being used in scientific and HTC from an energy efficiency perspective; We also outlined a set of tools and techniques to accurately measure energy consumption at the different levels of the computing systems; Finally, the developed scheduling algorithm showed potential savings in the electrical bill when applied to heterogeneous data centers working under a dynamic electricity pricing market.

Keywords: energy efficiency, scientific computing, ARM, Intel, RAPL, tools, techiques

Language: English

Acknowledgements

Acknowledgements!

Espoo, 14 December, 2015

Gonçalo Marques Pestana

Contents

Chapter 1

Introduction

1.1 Overview

Nowadays, computing technology is still developing according to Moore's Law. The number of transistors per chip-set is still increasing at a geometric rate. However, the energy consumption of the systems have begun to halt the usage of the technology at its full potential. Therefore, energy efficiency has became an important research topic in computer science, for energy has become a major growth bottleneck in computing systems. In addition, the increasing concerns with energy consumption and its social, economical and environmental impact in our society has given a bigger dimension to the discussion.

There are two major approaches to tackle the energy bottleneck in the current technology panorama. One, is to develop techniques and technologies to better harvest, transform and store energy to be used by computing systems. The second approach to the problem is to improve the energy efficiency of the existing computing systems. This thesis focuses primarily on the latter approach.

The concerns with energy consumption and its impact in the current applications affect several industries, from mobile device industries to big data centers. Given the several layers and complexity of the systems used nowadays, there are considerable number of directions that can be taken to improve the overall energy efficiency. Throughout this thesis, we will focus on measuring and improving the energy consumption in High Throughput Computing (HTC) applied to Scientific Research.

The Large Hadron Collider example

In some applications, a single computing unit does not have enough resources to accomplish the required tasks. A recurrent strategy is to distribute computational tasks across a set of computing centers that might be spread geographically.

The Large Hadron Collider (LHC) [18] at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland, is an example of a scientific project whose computing resource requirements are larger that those likely to be provided in a single computing unit. Thus, data processing and storage are distributed across the Worldwide LHC Computing Grid (WLCG) [24], which uses resources from 160 computer centers in 35 countries. Such computational resources have enabled the CMS [38] and ATLAS [59] experiments to discover the Higgs Boson by S. Chatrchyan, et al. [36] and Aad, Georges, et al. [26], amongst other scientific achievements. The WLHC requires a massive amount of computational resources (between 80.00 - 100.00 x86 cores in 2012) and, proportionally, energy. In the future, with planned increases to the LHC luminosity, the dataset size will increase by 2-3 orders of magnitude, posing even more challenges in terms of energy consumption.

The LHC is an example of a massive computational system that needs to improve its energy efficiency to reach its full potential in the present and future time. Throughout this thesis, we will focus primarily on the LHC case. When appropriate, we will use authentic data and current technology used by the CMS.

1.2 Reduced Instruction Set Computing and ARM

Reduced Instruction Set Computing (RISC) and Complex Instruction Set Computing (CISC) are computing architectures with different computing designs. RISC architectures are designed to operate smaller number of computer instructions than CISC. RISC aims at performing more instructions per second, resulting on more energy performance than CISC the architectures.

The most well known and widely used commercial implementation of CISC architecture are chipsets produced by Intel. On the other hand, ARM has been focusing on the development of RISC based architectures chipsets. Traditionally, ARM chipsets have been used in the energy constrained devices market.

Given the maturity and availability of ARM compared to other RISC computing chipsets, we used ARM chipsets to conduct our research.

1.3 Problem Statement

A considerable amount of research has been done on leveraging RISC architectures to minimize energy consumption on mobile and energy constrained devices. In the mobile and energy constrained devices case, it is important to build systems with low energy consumption due to constrains of available power.

The large share of ARM architectures in the mobile market supports the fact that RISC is a good fit for mobile and energy constrained devices.

Similarly to mobile devices, the High Throughput Computing (HTC) community major concerns with energy efficiency. However, studies focusing on the viability of RISC architectures in HTC as a way to minimize energy consumption are not abundant in the research technology. Furthermore, to the knowledge of the authors at the time of this research, there are no real world implementation of such technologies in HTC data centers.

It is still unclear whether RISC architectures - and notably ARM chipsets - are a good match to HTC computing. There are open points regarding whether the performance constraints of ARM architecture and the high performance requirements of HTC workload are acceptable. In addition, it is still unclear if ARM architecture is more energy efficient under HTC workloads than the conventional Intel CISC-based architectures.

Therefore, it is of our interest to study the potential impact of ARM architectures in the HTC and scientific computing industry. In our opinion, there are two major gaps that need to be filled: Firstly, a comparison between ARM and Intel architectures under authentic scientific workloads. Secondly, a solution that leverages RISC architectures under HTC and scientific computing.

1.4 Scope of the Thesis

The purpose of this thesis is to help filling the gaps above mentioned. Firstly, we aim at answering whether RISC architectures are a potential fit to HTC and scientific computing from a energy efficiency perspective. We focus mainly on comparing the ARM architecture and Intel, which is a widely used CISC architecture in HTC. During this research, we use authentic HTC workload and computing frameworks as the CMS experiment.

Secondly, we use the results of the comparison between ARM and Intel architectures to study how to lower down the electrical bill of HTC data centers running under a dynamic energy pricing ecosystem.

In order to accomplish the tasks in hand, we start by investigating the best and most accurate ways to measure power consumption and compare different architectures. After, we run several experiments in machines with different chip-sets using authentic workloads and software from the CMS collider. We compare the results and draw conclusions from them. Finally, based on our learnings, we frame a methodology for lowering the electrical bill of data centers running under a multi energy pricing policy, by leveraging the scheduling of machines with different efficiency profiles.

1.5 Contributions

Our main findings are:

- Accurate techniques and tools for measuring power consumption are important to develop more efficient systems. We researched and outlined best practices for different system levels in the Chapter 3. The results of this research were published in the conference proceedings of the 16th International workshop on Advanced Computing and Analysis Techniques in physics research (ACAT'2014) [30].

- The ARM architecture shows potential for an energy savings in HTC when compared to the x86 systems widely used nowadays. We performed experiments with real world workloads (Chapter 4) and analyzed the results (Chapter 5);

- Heterogeneous computing can be leveraged in HTC in markets where the price of the electricity is dynamic. We have shown how to achieve savings in such environment and developed a scheduling algorithm that accomplishes that (Chapter 6);

1.6 Structure of the Thesis

This thesis is structured as following. In the next chapter, we define the context and scope of the thesis by synthesizing relevant literature work. In the third chapter, we outline measurement tools and best techniques for measuring power consumption. The fourth chapter outlines the experiments methodology used during the different experiment sets. The fifth chapter outlines and analyzes the results based on the data obtained from the experiments. In the sixth chapter, based on the results of our experiments we present a scheduling algorithm that aims at lowering the energy bill of HTC by using both ARM and Intel architectures. Finally, conclude by outlining possible future work and presenting the conclusions of this thesis.

Chapter 2

Background

2.1 Energy consumption awareness in Scientific Computing

Power consumption and energy efficiency have become of paramount important topic in computer science, due to the major growth bottleneck that energy imposes to systems. This bottleneck has been hindering the further development and usage of technology such as mobile devices and low energy devices. In addition, high throughput and scientific computing have also been affected by energy consumption increase. In these fields, the large amount of data that has to be stored and processed have been increasing with the time.

Moreover, the increasing concerns with energy consumption and its social, economical and environmental impact in our society have given a bigger dimension to the discussion.

The importance given to energy consumption among the research community is visible in the amount of research done in this field.

Literature review

As stated by Chia-Lee Yang, et al. [61], scientific computing is often characterized by requiring enormous data storage capacity, high processing capabilities and complex configuration. High processing capabilities and massive data storage are requirements that potentially require massive amounts of energy. Thus, the scientific computing community is looking into energy efficiency with special interest.

Given its importance, many research studies have worked on technologies to increase energy efficiency and lower power consumption of computing systems. There is a vast panoply of studies with different approaches toward improving energy efficiency. For example, to use graphic processing units (GPU) for data processing as in Sangduk Kim, et al. [44] and Kai Ma, et al. [47]. The GPU is a specialized circuit developed to manipulate and process visuals and responsible for a fast display of complex animations based on mathematical renderings. According to Sangduk Kim, et al. [44], the benefits from using GPU to process data is due to its parallel architecture. The parallel architecture of GPU is well suited for intensive multimedia applications and it does the task in a low energy manner. Consequently, Sangduk Kim, et al. [44] concludes that the high-performance GPU achieves better results in terms of energy efficiency that any combination of CPU and GPU. They tested their results with a compute-intensive workloads. On the other hand, Kai Ma, et al. [47] state that with the correct scheduling between both GPU and CPU, the system can save up to 20% energy. Although Kai Ma, et al. [47] and Sangduk K. et al [44] reach contradictory conclusions, it shows that the research community has been actively seeing GPU as a potential processing unit to reduce energy efficiency.

Apart from using GPU for low energy data processing, there have been some attempts to leverage RISC architectures in scientific and high throughput environments. Some examples are Zhonghong Ou, et al. [49] and Abdurachmanov, et al. [31], [27]. In Zhonghong Ou, et al. [49], the authors show that ARM architectures present a potential energy saving alternative when compared to Intel x86 architectures. Although, in some use cases, the an ARM based data center can become more costly than the widespread solutions using Intel. Based on the study conducted by Abdurachmanov, et al. [49], the ARM clusters are financially viable when performing lightweight computational applications. On the other hand, when it comes to computation-intensive applications, the amount of ARM nodes needed to perform the tasks increases substantially. This makes the solution less interesting from a financial perspective. However, if one is only concerned with the overall energy efficiency of the cluster, ARM-based clusters show a better energy efficiency factor than Intel. Abdurachmanov, et al. [31], [27], present their initials results on using ARM architecture for computing intensive applications in a scientific environment. They report how a ARMv7 based development board performs under scientific computing workloads. Their conclusions show that ARM based architectures show great potential for use of the typical scientific workload with HTC.

Several other directions have been taken toward energy efficient computing, such as Gustavo Pinto, et al. [53], Zhenhua Liu, et al. [45], Sharifi, et al. [57] and the number of related works keeps growing. Gustavo Pinto, et al. [53] mined well-known on-line forums, such as StackOverflow [21], to understand what are the main concerns of developers regarding application level energy efficiency. An interesting conclusion then drawn from the study is that even though the questions asked are generally interesting and relevant from a scientific point of view and correctness, the answers are generally poorly addressed. The answers have been shown to be either vague or flawed. They also contributed with a summary of software energy consumption problems that developers and end users focus the most. Zhenhua Liu, et al. [45] explore the potential of geographically distributed in Internet-scale systems. Finally, Sharifi, et al. [57] compare how massive data centers compare to distributed data centers in terms of energy consumption. They conclude that distributed cloud platforms outperform the classic data center model. They also show how the MapReduce technique can be leveraged to save energy in a distributed data center model.

2.2 The European Organization for Nuclear Research and the LHC

The European Research for Nuclear Research (CERN) [6] is a particle physics research laboratory sited in the Franco-Swiss border. At CERN, thousands of engineers and physicists from about 21 state members conduct research about the fundamental structures of the Universe. In order to perform experiments that support scientific research, CERN has built several particle accelerators and detectors. The most outstanding collider is the Large Hadron Collider (LHC). The LHC consists of a 27-kilometer ring of superconducting magnets that boost the particles while traveling through it. The particles are accelerated in two beams, traveling inside the LHC in opposite directions. When the beams are traveling close to the speed of light, they are made to collide in the different colliders. After each collision, particles and subatomic particles are projected due to the collision, which is tracked and recorded by the colliders. The data acquired from the collisions is then filtered and the most interesting information is stored into the CERN data-centers for posterior analysis. The analysis consists in two stages: First, the data is reconstructed. This stage consists on re-arranging and decompressing the stored data. The second phase consists processing the raw data in order to be used by the researcher community.

Given the frequency of the collisions and the massive amount of data to store and process from each collision, the LHC is an example of a scientific computing endeavor which energy management is critical.

Literature review

According to Abdurachmanov, et al. [31], the computing requirements for HTC have increased particularly in recent years. Most notably, a project with the magnitude and complexity of the LHC is a sound example of it. In order to achieve results like the discovery of the Higgs boson by Georges Aad, et al. [26], [36] and other significant scientific advances, a massive amount of computational resources - and thus energy - was necessary. Given the enormous amount of resources needed for storing and processing the data, it was not viable to concentrate all the tasks in one single data center. Thus, the solution was to distribute the processing tasks across several partners and institutions through a distributed network of nodes, called the Worldwide LHC Computing Grid (WLCG) (the same approach taken by Sharifi, et al. [57]). The WLCG is a grid computing platform where more than 170 computing centers spread across 40 countries collaborate to store and process the data coming from the LHC experiment [24]. The WLCG alone is responsible for the distribution, storage and processing of more than 30 petabytes annually. According to Abdurachmanov, et al. [31], the equivalent capacity of WLCG in 2012 was between 80,000 and 100,000 x86-64 cores, which would be difficult to build and maintain in a single super data center.

In the future, other projects and researches will demand even more processing capacity from the WLCG. For example, as according to Abdurachmanov, et al. [31], in order to upgrade the luminosity of the LHC detectors to its full potential, the datasets will increase size by two to three orders of magnitude, with processing power increasing in proportion.

These outstanding numbers, in addition to the price of energy and the increasing concerns with green computing, highlight the importance of developing more efficient and methods and techniques high performance computing, both in the scientific computing in general and in the LHC computing grid in particular.

There are several studies concerning computational energy efficiency at CERN. Intel researchers [25] present methods to better use Intel hardware for energy efficient computing at CERN. The white paper also presents some future solutions based on the x86 technology developed by Intel that will eventually perform better from an energy point of view than the current solutions. Khan, et al. [43] describe how they developed a energy profiling module on top of IgProf to better understand how energy is consumed by the system. The authors focused primarily on running the energy profiling on IgProf with tasks used by the software of LHC. The starting point is marked by stating that high throughput computing in general - and scientific computing at CERN in particular - has to rely on tools that provide a deep understanding of how power is used in order to develop energy efficient solutions.

According to Abdurachmanov, et al. [29], the authors conducted a survey on the potential advantages of using x86-64 variants, ARMv7 32-bit, ARMv8 64-bit, Many-Core and GPU solutions in the LHC context. The main focus was to evaluate whether the solutions surveyed would present advantages from an energy efficiency perspective. According to Abdurachmanov, et al. [28], the authors conducted a research on how both power-aware software applications and scheduling algorithms could improve energy efficiency in both distributed and centralized data center models. They used the LHC as a powerful distributed processing data center to access the results and outlined a set of steps toward a distributed computational model using heterogeneous computing units. In Abdurachmanov, et al. [32], the authors conducted a research on the viability of ARM processor and the Intel Xeon Phi co-processor for scientific computing in the CERN context. The results show that single core processing performance is much lower for ARMv7 than Intel architectures. However, from an power consumption perspective, the performance per watt is better for ARMv7 architectures. This research shows the potential of ARMv7 to be used in scientific computing.

2.3 Tools and techniques for measuring energy consumption

Many research works have discussed the importance of the right tools and techniques for measuring the energy consumption of computing systems. Naehyuck Chang, et al. [35], Gustavo Pinto, et al. [53], Abdurachmanov, et al. [27], [31], [30], Hergenroder, A., et al. [41] and other studies mention, accurate techniques and tools for measuring power consumption are of paramount importance to understand the bottleneck components that need to be improved from a energy consumption point of views. Without accurate measurements and feedback from the computing systems, it becomes difficult to improve the energy performance in a larger scale. As stated by Gustavo Pinto, et al. [53], Naehyuck Chang, et al. [35] and Abdurachmanov, et al. [30], there are only few research studies on tools and techniques for energy measurement of computing systems at the different levels.

Literature review

The study conducted by Gustavo Pinto, et al. [53], show that engineers have been considering energy consumption as an important factor when developing software. The research consists of an empirical study that aims to understand the opinions and problems of software developers about energy efficiency. The data that sustains the conclusions are taken from a well-known technical forum, (*StackOverflow* [21]). Although the study is focused in an application-level energy efficiency, it shows that developers are aware of the importance of energy efficiency in computational systems. When trying to understand in depth what questions arise more frequently, it is shown that measurement techniques is amongst the most asked questions by developers. In addition, the study ascertains that the *"lack of tool support"* is an important handicap for the development of energy efficient software.

Naehyuck Chang et al. [35] recognize that the accurate understanding of how energy is measured in computing systems is the basis for any high level decision on how to improve the energy efficiency. In addition, the study states that conventional measurement techniques have serious limitations. Therefore, they introduce a new methodology to measure energy consumed by the system on the chip, which is capable of sampling energy consumption in real time. They used a ARMv7 processor to test their measurement technique, which is shown to be accurate and applicable for measuring energy consumed by computational systems.

Other research works such as Gorlatova, et al. [40], Jinhua Zhu, et al. [64] and Dayarathna, M, et al. [37] leverage mathematical models and algorithms to represent the energy consumption of complex computing systems and data centers. Even though we do not undertake this approach in our study, these studies highlight the necessity of proper understanding how the computing systems spend energy in order for researchers to be able improve energy efficiency. Jinhua Zhu, et al. [64] claim that energy models are pivotal in design and optimization of energy efficient data centers. They surveyed the state of the art models for prediction of energy efficiency in data centers. One interesting finding of this study is that many of the models existing to date take into consideration only the CPU related energy consumption, rather than the overall system power consumption of the data centers. The authors claim that this leads to less accurate energy consumption models of data centers.

2.4 RISC in scientific computing

One possible approach to increase the energy performance of HTC and scientific computing data centers is by reducing power consumption of the computing chipsets. RISC architectures have been developed and design to perform minimal number of types of instructions. This leads to outstanding performance with less power consumption when compared to CISC architectures. Thus, RISC architectures - most notably ARM chipsets - have been widely used to minimize energy consumption on mobile and energy constrained devices. An open question among the research community is whether RISC architectures are capable of performing well under the computing-intensive workload that HTC and scientific computing require. Accoriding to Abdurachmanov, et al. [3], ARM chipsets are an example of widely used RISC architectures.

According to Abdurachmanov, et al. [3] , the ARM architecture is a RISC architecture developed by the company ARM Holdings. As a RISC-based chip, the ARM processors require less transistors than the typical CISC solutions. According to Zhonghong Ou, et al. [50], this approach reduces the overall power consumption of the RISC architectures when compared to CISC architectures. For this reason, ARM units have been widely used in energy constrained devices such as embedded systems, laptops and smartphones. These applications are responsible for the rise of ARM processor in the market in the last decade. The 32 bit ARM architecture, mostly known for ARMv7, is one of the most popular architectures in the mobile devices. While ARMv7 has been taking over the low-energy processor market, the ARMv8 architecture [4] introduces the a RISC 64-bit with focus on power efficient devices. Given the fact that ARM architecture were developed from the ground up with energy efficiency as main priority, studies such as Abdurachmanov, et al. [27], [31] and this thesis have delved around the idea of using the ARM architectures to improve the energy efficiency of HTC data centers.

Literature review

Several studies have been researching on how can RISC architectures in general - and ARM in particular - be used to improve energy efficiency. Zhonghong Ou, et al.[50] compare ARM-processor based clusters with Intel clusters from a energy efficiency and cost efficiency perspective. They analyzed and compared ARM Cortex-A8 and Cortex-A9 with Intel Atom and Intel Sandy Bridge microprocessors and concluded that there is no fundamental difference in the design of CISC and RISC microprocessors that could make one more energy efficient than the other. There is, instead, different design considerations when Intel and ARM developed their processors and that is what makes the difference in terms of energy efficiency. Another study on the comparison between RISC and CISC architectures in terms of power consumption was conducted by Blem, E, et al. [33]. The study acknowledges that today, the once marginally used RISC architectures are blooming, due to their use in low energy devices such as mobile devices and embedded systems. Furthermore, they also state that the traditionally low power ARM architecture is entering in the high-performance server market. Zhonghong Ou, et al. [50], [33] conclude that the energy performance edge of RISC over CISC architectures exists because of design points, rather than intrinsic characteristics of either RISC or CISC. The authors conclude, thus, that the size of the instructions set in the CPU does not account for the energy performance of the cores. Indeed, the discussion among the research community about the differences between CISC and RISC architectures in performance has been happening for a while. Already in 2009, Ciji I. et al. [42] discuss about the community debate between CISC and RISC from a performance perspective.

Lozano, H, et. al. [46] acknowledge that to process HTC tasks in RISC architectures is a challenging task due to its design particularities that inhibits performance. The research conducted by Padoin, E.L., et al.[51] aims at studying several characteristics of ARM processors such as energy efficiency, execution time and power consumption. Their main goal was to verify whether clusters using RISC architectures - and thus with low power consumption - would be feasible. The experiments conducted during the study used an ARM development board, as in our study. Padoin, E.L., et al. [51] concluded that ARM clusters show potential as an alternative to the widely used Intel x86 architectures.

Even from a broader point of view, RISC architectures are been seen as an alternative for the widely used CISC architectures. Patterson, David A., et al. [52], the authors claim that the future generation of very large scale integration (VLSI) computers might be more effectively implemented as RISC than as a CISC architecture.

In Abdurachmanov, et al. [27], a server-purpose ARM machine is compared with the recent Intel architectures, such as the recent Intel Xeon Phi and a dominating Intel product intended for HTC workloads (Intel Xeon E5-2650). The workload for comparing the architectures was ParfullCMS. They based the results on performance (events per second) and scalability over power (watts). In addition to performance and energy consumption comparisons, the paper describes the porting endeavors of the CMSSW to ARMv8 64-bits architecture.

Abdurachmanov, et al. [27] used an APM X-Gene 1 running on a development board. It consists of a 8 physical core processor running at 2.4GHz with 16GB DDR3 memory. The authors highlight that the firmware for managing processor ACPI power states was not available when the study was made. It is expected that the energy performance will improve once the firmware will be available.

Under the circumstances of the experiment, the overall results show that APM X-Gene is 2.73 times slower than Intel Xeon Phi. From the energy consumption performance (events per second per watt), the Intel Xeon E-2650 is the most efficient, with APM X-Gene presenting similar performance despite the absence of platform specific optimizations. Therefore, they conclude that the APM X-Gene 1 Server-On-Chip ARMv8 64-bit solution is relevant and potentially interesting platform for heterogeneous high-density computing.

According to Abdurachmanov, et al. [31], processors have hit scaling limits after the year of 2015. There has been development of multiprocessor architectures that allow to run parallel tasks. However, also commodities such memory, I/O streams and energy need to scale proportionally in such architectures.

2.5 Workload scheduling based on dynamic energy pricing

The workload scheduling based on dynamic energy pricing can be a good approach when certain conditions are in place. Firstly, in situations when the energy price changes according to the overall power grid energy usage. Secondly, when there are machines with different energy efficiency and computing performance available. Given these conditions, the main concept is to schedule the workload to more energy efficient machines when the energy price is higher. On the other hand, when the energy price is lower, faster machines can be without major increase in electricity bill.

Literature review

There are several studies exploring inter data center solutions to lower the electricity bill by leveraging both location and price dynamics of energy pricing. The emphasis is given to job scheduling across data centers that are located in different places. The main idea is to exploit the fact that energy prices change based on location and time. The research community is mostly concerned with fairness, server availability, queue delays, bandwidth costs with job migration and quality of service. In addition there are several research studies related with migration of cloud computing jobs. Studies that in one way on another address this perspective are Ren, Shaolei, et al. [56], Buchbinder, Niv, et al. [34], Rao, Lei, et al. [55], Qureshi, Asfandyar, et al. [54], Mei, Jing, et al. [48].

Besides inter center solutions, the research community has been addressing the power consumption of the computing nodes specifically from a data center perspective. This perspective is closer to what we are trying to achieve with our solution. For example, Xu Yang, et al. [62] achieve better energy performance in a dynamic pricing environment with HTC systems by judiciously scheduling parallel jobs - with different energy profiles - depending on the energy pricing of the moment. The main difference to our solution is that the performance of the machines are not taken into consideration when scheduling the jobs, but rather the energy profile of each job.

Another related study was conducted by Yan Wang, et al. [60], in which the authors came up with an optimal algorithm and two heuristic algorithms to schedule tasks to heterogeneous processors. The study also takes into consideration the memory allocation in heterogeneous memory in order to minimize energy consumption while meeting the assumed deadlines. Their work also considers heterogeneous memory allocation as well. They consider a computing process executing several tasks in a parallel computing environment. The system consists in a variety of different computational node, each of one with a given number of processors. All computational nodes are connected by a high-speed network. Thus, all the processors can cooperate and realize complementary and parallel tasks. From the energy point of view, the processors of each computational node have an energy profile assigned and have a certain frequency, which will be taken into consideration when scheduling the task. The work dates from end of 2014, which indicates that this is a trendy and hot subject.

A similar idea has been explored by Guosun Zeng, et al. [63]. They present only heuristic algorithms to schedule tasks on heterogeneous computing systems, based on efficiency and energy consumption. They develop heuristic algorithms due to the fact that an optimal solution for the needed scheduling is NP-complete.

According to Ren, Shaolei, et al. [56], because of the magnitude of energy costs in data centers, it is important to lower the energy consumption in data centers. The servers are composed of heterogeneous machines. In addition, the data centers may be located in different geographical locations and, thus, have different energy tariffs. Ren, Shaolei, et al. [56], claim that the key idea to lower the energy bill in data centers is to have energy efficiency servers and schedule the jobs to where energy is more affordable at a given time. In the context of servers distributed over different geographical locations, it is also important to satisfy fairness and delay constraints. This scenario is less critical when the server is not distributed, as in our case.

Ren, Shaolei, et al. [56] present an on-line scheduler that distributes batch workloads across multiple data centers geographically distributed. The scheduler aims to minimize the energy consumption of the set of servers having into consideration fairness and delay requirements. The scheduler is inspired on the technique that optimized time-varying systems. The algorithm takes a queue of jobs schedule them to the different servers having in consideration the (1) server availability, (2) energy price and (3) job fairness distribution. Consequently, the algorithm is tuned to calculate the trade-off between energy pricing, fairness and queuing delay.

In Ren, Shaolei, et al. [56], the scheduler developed takes into consideration the server availability, energy costs, fairness and queuing delay to schedule random jobs arrivals. It opportunistically schedules jobs when (and to where) energy prices are low. Comparing to our study, though, we do not consider geographically distributed servers but rather, we have schedule the jobs based on the heterogeneous set of machines existing on the server.

Buchbinder, Niv, et al. [34] aim to leverage the temporal and geographical variation of electricity prices, in the context of data centers. They study algorithms to schedule (migrate) jobs in data center based on the energy cost and availability. When the servers are in different geographical location, costs with data migration have to be taken into consideration, namely bandwidth costs of moving the application state and data between data centers. The bandwidth costs increase proportional to the amount of data migrated between servers. Their study focuses on inter data center optimization, rather than intra data center optimization. The algorithm differs from others in 3 major differences: First, they consider migration of batches of jobs. Second, the algorithm has into consideration the future influence of the job scheduling, providing robustness against any future deviations of the energy price. Finally, they also take into consideration the bandwidth costs associated with job migration across data servers.The main point is to provide a good trade off between the energy pricing and the job migration, taking into consideration the bandwidth prices. Comparing to our study, we do not approach the problem from an inter data center perspective, but rather from an intra data center, by scheduling the jobs to machines depending on their energy performance and the actual energy prices. One interesting idea from this study that can be used, is the usage of an online algorithm that takes into consideration the expected prices and also the actual prices.

Rao, Lei, et al. [55] try to systematically study the problems of how minimize the electricity cost in data centers while guaranteeing minimal quality of service. To that end, they take into consideration the local and time diversity of electricity prices. The contributions are twofold: In one hand, they show that local and time dependent electricity pricing can be leveraged to minimize total energy price of clusters of data centers. On the other hand, they present a mixed-integer optimization formula with linear programming formulation to show that the energy pricing of clustered data centers can be improved under such conditions. To model the total of electricity costs, they assume that all the servers have a similar power profile - which means that all the servers, disregarding their locations, have the same workload. They calculate the power consumed by the server by multiplying the total of servers at a certain region by the total of workload they have. They obtain the most efficient solution, they approximate an optimization problem through a linear programming formulation and then, convert the linear programming formulation to a minimum cost flow problem.

This work dates from 2010 and don't take into consideration the bandwidth costs of migrating the batches between data centers. Even though that is not an issue in our study, this is taken into consideration in other works such as conducted by Buchbinder, Niv, et al. [34]. Again, it is part of the set of studies on inter data-center and electrical costs optimizations that location and time based pricing allows.

Qureshi, Asfandyar, et al. [54] show that existing systems may be able to save millions of dollars by judiciously schedule workload to servers taking into consideration the temporal and geographical variation of energy prices. The results are based in historical data collected on Akamai's CDN.

Xu Yang, et al. [62] leverage the fact that parallel jobs have distinct energy profiles. Taking it into consideration, they study the impact of scheduling jobs according to the energy prices at a given moment and the job's energy profiles. So, the study aims to reduce the electricity bill by scheduling and dispatching jobs according to their energy profile. Their solution has a negligible impact on the system's utilization and scheduling fairness.

Their basic idea is to schedule jobs with low energy profile during on-peak electricity time and, on the other hand, schedule jobs with high energy profile during the off-peak electricity time. In addition, the scheduling is done in such a way that it is guaranteed that there is no degradation of the overall system performance.

The authors claim that "A key challenge in HTC scheduling is that system utilization should not be impacted. HTC systems require a tremendous capital investment, hence taking full advantage of this expensive resources is of great importance to HTC centers.". This may make impractical and wreck our solution, because of the inevitability of turning off (or idle) great amounts of computing resources. Although, Internet data centers (cloud data centers) may be a good match to our solution: usually there are much less resources being used at a given time than in HTC computing [need confirmation, partially mentioned in this article]. The scheduling algorithm used places jobs in a time-window. The jobs are chosen to run based on job fairness, job energy profile and energy prices at a given time. A greedy algorithm and 0-1 Knapsack based policy are used to minimize the electrical costs. Their results show that gains in the order of 23% can be obtained without impact on the overall system.

According to the survey carried by Xu Yang, et al. [62], the dynamic energy pricing has been implemented in the biggest markets in Europe, North America, Oceania and China, while Japan was at the time starting to test it on its major cities. They develop two power aware job policies: 1) greedy approach, where jobs are allocated based on their energy profiles and 2) 0-1 Knapsack based policy, where both job profile and system utilization are taking into consideration.

TMei, Jing, et al. [48] present a novel task scheduling algorithm for HTC systems which considers two main points: reducing the energy consumption of the overall system and minimize the schedule length. An HP system is defined by the authors as set of distributed computing machines with different configurations connected through a high speed link to compute parallel applications.

They assume a that all the information needed to schedule the task is known beforehand. The scheduling algorithm assigns then the jobs to the different machines. Thus, the scheduling algorithm is said to be static, in opposition to, for example, the on-line algorithms. One of the particularities of the algorithm is to reduce the impact of duplication-based algorithms. The duplication-based algorithms schedule jobs across machines redundantly, in order to maximize performance by eliminating intercommunication between tasks. However, from the energy consumption point of view, it is not th ideal situation since more than one processor are performing the same job. Once again, this research work aims at improve the energy efficiency of HTC systems at a distributed level and do not focus, as our approach, on inter data center solutions.

Yan Wang, et al. [60], the authors address the problem of an energy aware scheduling for heterogeneous data allocation and task scheduling. The problem consists in finding the best tasks scheduling in a heterogeneous system that meet the deadlines while minimizing the energy consumption.

The processors and memories come in different flavors nowadays in HTC systems, making complex the task of efficiently schedule processor power and memory space in an energy efficient way. The problem of finding an optimal processor and data scheduling becomes critical when trying to minimize energy consumption and meet imposed deadlines. As the study shows, there are several research efforts tackling the task scheduling problems on heterogeneous computing and, most notably for our research. They present an optimal algorithm and two heuristic algorithms to solve the HDATS problem, since the optimal algorithm takes too long to solve problems until 100 nodes. The optimal solution has two phases: First is uses the DFG_Assign_CP algorithm to better map each task to node. Secondly, it choses the data assignment to whose total energy consumed is reduced and the deadlines met.

Guosun Zeng, et al. [63] claim that, unfortunately, there are not many studies of processor scheduling algorithms that take into consideration both time and energy. In this study, they explore heuristic scheduling algorithms focused on high performance computing and green computing. They work on heuristic algorithms and not in the optimal algorithm, because the optimal algorithm is proven to be NP-complete.

Chapter 3

Tools and techniques for energy measurement

The recent scientific applications have to process and store considerable volumes of data. It is expected that the volume of data will increase considerably in the future, as technology improvements and requirements increase. This fact increases considerably the costs with energy in a HTC system. Thus, energy consumption has become a major concern amongst the scientific community.

It is critical to understand how energy is used by the HTC systems and its components, in order to develop solutions for improving energy efficiency in HTC. The learnings, techniques and tools can be further implemented in non-HTC systems with the same results.

This chapter outlines and describes tools and techniques for measuring energy consumption in any machine. Our main goal is to use these tools and techniques to study energy efficiency of HTC systems, more specifically systems running on top of x86 and ARM architectures. The tools and techniques described in this chapter were used widely during the study of energy efficiency comparison described in the following chapters.

The results of this section were publish in the conference proceedings of the 16th International workshop on Advanced Computing and Analysis Techniques in physics research (ACAT'2014) [30].

3.1 The big picture

During this study, we will consider two different granularities at which is possible to measure the energy consumption of a computing system. The two granularities are coarse and fine granularity and they differ on the type of system components that are taken into consideration when measuring energy consumption.

The coarser granularity takes into account the behavior of the whole node. This is usually investigated when engineering and optimizing computing centers. Alternatively, a more detailed approach is to look into the components which make up the active parts of a node, in particular the CPU and its memory subsystem since these are responsible for a sizable fraction of the consumed power. According to Abdurachmanov, et al. [30], those are the components where the largest gains in terms of efficiency can be obtained through optimizations in the software.

Coarse grain or external measurements

As stated by Abdurachmanov, et al. [30], if one is simply interested in the coarse power consumption by node, external probing devices can be used: monitoring interfaces of the rack power distribution units, plug-in meters and non-invasive clamp meters (allowing measurement of the current pulled by the system by induction without making physical contact with it). They differ mostly in terms of flexibility. Their accuracy is typically a few percent for power, whereas their time resolution is in the order of seconds. This features are enough to optimize electrical layout of the data-centers or to provide a baseline for more detailed studies.

Fine grain or internal measurements

An alternative approach takes into account the internal structure of a computing element in an HTC system, as shown in figure 3.1. Almost all board manufacturer provides on-board chips which monitor energy consumption of different components of the system. These allow energy measurements of fine grained detail, as it is possible to individually monitor energy consumption of components such as the CPU, its memory subsystem, and others. Texas Instruments TI INA231 is an example of such chips[22]. The TI INA231 is a current-churn and power monitor which is found on the ARMv7 developer board which we used for our studies. It is quite common in the industry. Compared to external methods, these on-board components provide high accuracy and reasonably high precision measurements (millisecond level). Another example of a fine grained chip is the Running Average Power Limit (RAPL) chip included in the recent Intel boards.

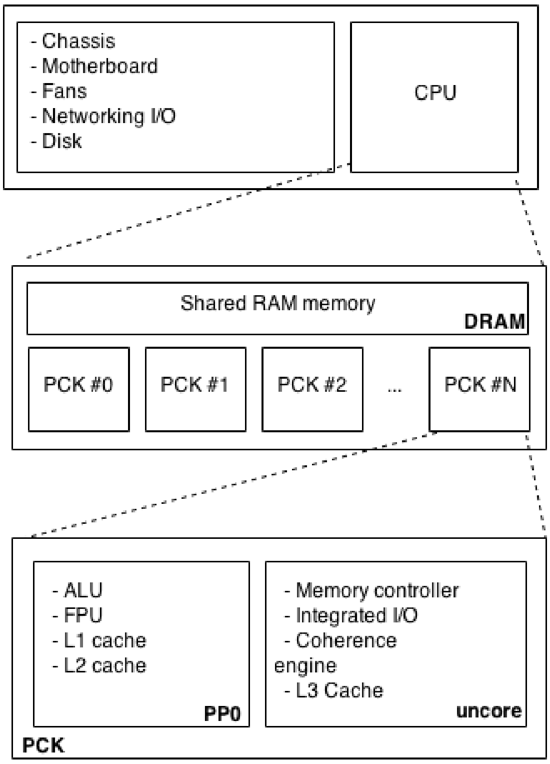


Figure 3.1: Components that contribute for power consumption in HPC. Taken from [30]

3.2 Tools for measuring energy consumption

Internal tools

Running Average Power Limit

Recent Intel boards provide a powerful way to measure fine grained energy consumption of their CPUs with the Running Average Power Limit (RAPL) technology. RAPL has been included on the board from the factory since the Sandy Bridge family of processors.

Contrary to other solutions (such as TI INA231, for example), which are implemented as discrete chips, RAPL is embedded as part of the CPU package itself and provides information on the CPUs own subsystems. In particular RAPL provides data for three different domains: **package** (pck), which measures energy consumed by the system's sockets, **power plane 0** (pp0), which measures energy consumed by each and every core, and **dram**, which accounts for the sum of energy consumed by memory in a given socket. [11]. As for the discrete components, the timing resolution of measurements is in the millisecond range, according to Balaji Subramaniam, et al. [58]. The range is fine grained enough to use the data to build an energy consumption sampling profiler for applications, similar to how performance sampling profilers work. Finally, in addition to power monitoring of the sockets, RAPL can limit the power consumed by the different domains. This feature, usually referred as power capping, allows the user to limit the average power consumption of a set of components.

TI INA 231

There are chips that provide similar power monitoring capabilities than RAPL, but for any architecture and technology. An example is the Texas Instrument (TI) INA231 [22]. The TI INA231 is a power monitor that reports current, voltage and power consumption of CPU, DRAM and cores of the SoC which is attached. Some vendors include the TI INA231 in their boards from origin, relying on an external technology to provide the same capabilities as RAPL monitoring.

Similarly to RAPL, the sampling resolution is high (at the sampling rate of microseconds) and the error is relatively low. On the other hand, the TI INA231 does not have capping capabilities.

External tools

There are different techniques and tools to measure the power consumed by the whole system 3.1. Usually when the system is part of a server rack, the rack itself offers an API to sample the power consumed by each of the systems and overall rack power consumed. For simpler systems, it is recommended to use either power sockets with power meters or an external power meter.

The resolution and errors of external tools vary according to their nature. The resolution can range from microseconds - when the sampling is done digitally - to seconds - when the sampling is done by a person. It is recommended to use API and digital based external tools in order to achieve the best resolution and error possible.

Conclusion

There are many different methodologies and tools to measure energy consumption of computing systems. The differences between the existent options range from granularity, accessibility, error and resolution and also from hardware or software approaches. The decision for which tools and techniques should be relied upon for the measurements is important when conducting research on energy efficiency. The decision is directly dependent on the goals of the experiment and the availability of the tools according to the technology to be used.

Chapter 4

Experiments

We performed experiments with different hardware setups. The experiments consisted on running simulations of HPC workload while measuring the energy consumed by CPU and by the whole machine. The main goal is to compare the energy efficiency of ARM and Intel architectures. To attain that goal, we compared the results of the experiments to evaluate the potential of ARM architectures to perform HPC tasks, in comparison to the Intel architectures.

The software used to run the computing tasks widely used in production and research at the CMS experiment. In order for the results to be as realistic as possible, we used data analysis software which are widely used in production at CERN.

We organized the experiments in 2 sets. The conditions under which the experiments were conducted were similar. Due to hardware and software limitations and availability, it was not possible to completely reproduce the experiment conditions across all the sets. However, we believe that the differences will affect the final results only to a reasonable degree, making it possible to scientifically compare the results. This and other considerations will be discussed further in the Analysis chapter.

The tools and techniques used to perform the energy consumption measurements were based on the study presented on the previous chapter. The setups of the experiments, methodology and tools used to perform the energy measurements during the experiments are explained and detailed in the following sections.

Throughout this chapter, we will label *set of experiments* as experiments conducted with the same hardware and software configuration. The degrees of freedom of each experiment are the number of events and number of threads processing the workload.

For each setup, we outline the hardware, software setups and the used energy measurement tools. During this chapter and throughout the rest of the thesis, we will describe each batch of experiments as first set of experiments and second set of experiments.

The remainder of this chapter divided in three sections. Firstly, we discuss about the software used to perform the experiments. Secondly, we outline the most relevant characteristics of the architectures used during the experiments. Lastly, we describe the setup of the experiments and methodology used to perform the experiments.

4.1 Software

In order to achieve result closest to reality, we ran our experiments in software which is widely used at CERN to analyze data from the CMS experiments. In order to analyze data from the CMS experiment, researchers use the CMS Software components (CMSSW) [7]. The CMSSW is a collection of software available as a framework. It contains the software components necessary for physicists to perform data analysis. The main components are event data model and services needed by the simulation, calibration and alignment and event reconstruction modules. The CMS data event model uses an *Event* as an abstraction. An *Event* is an object that contains all the raw and reconstructed data from a CMS collision. A collection on *Events* are processed and analyzed by researchers to understand how the particles in the collision behaved.

In some of the setups used during our experiments, the CMSSW could not be used due to lack of hardware support. In those cases we relied on a Geatnt4 benchmark [39] called ParFullCMS. ParFullCMS is a cross-platform test that simulates a subset of tasks carried by CMSSW. In addition, it is capable of running in multi-thread. ParFullCMS has been used in other research studies such as by Abdurachmanov, D. et all [27] to overcome similar the hardware support limitations as in our experiment setups.

4.2 Hardware

The focus of this work is to compare energy efficiency of ARM architectures and Intel based processors under similar workload. Our hardware choice was conditioned to the machine availability when the study was conducted. In addition, we also aimed at comparing similar conditions and workloads across all the set of experiments.

The ARM machines used were a single-board ARM processor developed by Odroid [20] and a server class ARM processor by Boston Viridis [23]. The Intel machines used were part of the micro-architectures family Sandy Bridge and Intel Bonnell. In the following sections, we will describe the hardware architecture, features of the hardware used to run the experiments.

4.2.1 ARM architecture

Boston Viridis server

The Boston Viridis server is one of the first ARM architecture based servers where the processors, IO and networking are fully integrated in one single chip. According to the vendor, the server is intended to perform in a web server, cloud and data analytics environment with outstanding power performance [23].

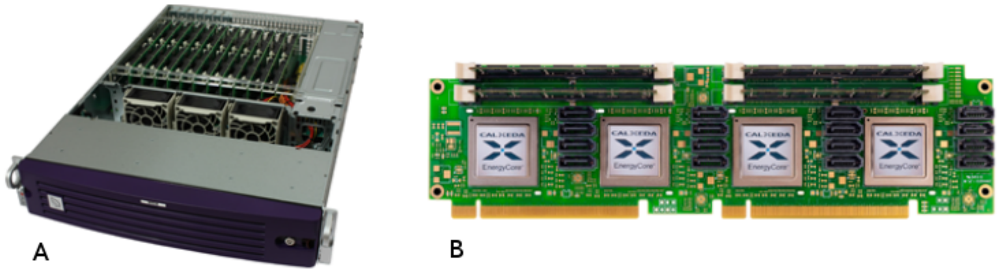


Figure 4.1: A. Viridis Server chassis with 12 energy card in it. B. Energy card with 4 nodes. Taken from [23]

The Boston Viridis server used in this study (which we will label as ARM_viridis throughout the rest of the document) consists of a chassis with twelve racks, each with an energy card. Each energy card contains four nodes, as represented in Fig. 4.1. A node is an ARM based CPU fabricated by Calxeda. The block diagram of a node is represented on Fig. 4.2.

Each ARM_viridis node contains four ARM A9 Cortex core with a clock speed up to 1.4MHz. A memory controller and L2 cache is included on the chip. In addition, a couple of energy management blocks and IO controllers complete the Calxeda EnergyCore processor 4.2. These energy measurement blocks were used to perform part of the energy measurements with this setup.

According to [2], the ARM A9 Cortex is a popular and mature general purpose core for low-power devices. It was introduced in 2008 and it remains a popular choice in smart-phones and applications enabling the Internet of Things (IoT) [2]. The ARM A9 Cortex supports the ARMv7A instruction set architecture. A detailed study of the ARMv7A internals is out of scope of

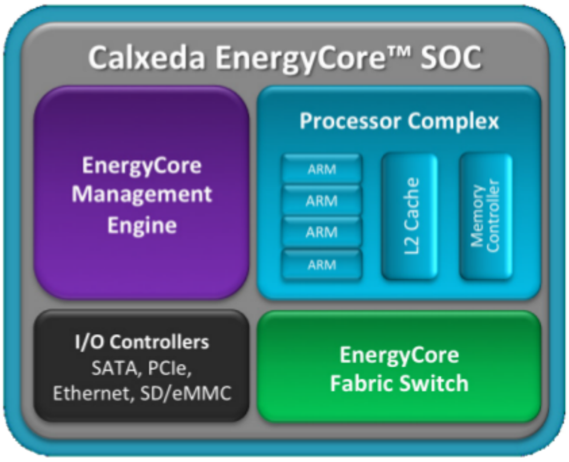


Figure 4.2: Block diagram of Calxeda EnergyCore. Taken from [23]

this work. More detailed specifications about the internals of the ARMv7A instructions set can be found in [2].

ODROID-XU3 development board

The ODROID-XU3 [20] is an open-source development board produced by Hardkernel. They claim that the ODROID-XU3 is a ”new generation of computing device with more powerful, more energy efficient hardware and smaller form factor” [20]. At the time of these experiments, the ODROID-XU3 was mostly used for testing and platform development and it was not intended to run in production scenarios. Throughout this document, the ODROID-XU3 described in this section will be called ARM_odroid.

The ODROID-XU3 processor has four Samsung Exynos-5422 Cortex A15 and four Cortex A7 cores, with 2GB of LPDDR2 RAM. Only four cores are working at the same time and they are scheduled based on the big.LITTLE technology. The big.LITTLE technology [1] automatically schedules workloads across cores based on performance and energy needs. The vendor claims that the big.LITTLE technology can achieve energy savings from 40% to 75%, depending on the performance scenario [1]. It is important to note that, even though the CPU contains eight cores, only four of then are working at a given moment. The block diagram of the ODROID-XU3 can be seen in 4.3.

The ODROID-XU3 has a Texas Instrument power monitor chip (TI INA231) embedded from origin. The TI INA231 provides an API to read the energy consumed by the cores and DRAM at a sampling rate of microseconds. These readings can be easily triggered and read through software and consist of an accurate way to make fine-grained energy consumption measurements. We assumed that the measurements made by the TI INA231 can be compared to the RAPL technology by Intel.

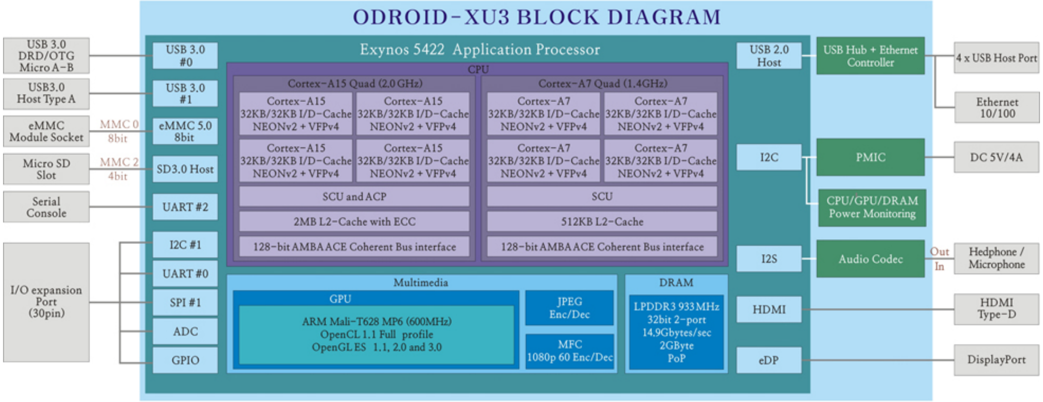

ODROID-XU3 BLOCK DIAGRAM
Exynos 5422 Application Processor
USB 3.0 DRD/OTG Micro A-B
USB3.0 Host Type A
eMMC Module Socket
MMC 0 8bit
Micro SD Slot
MMC 2 4bit
Serial Console
I/O expansion Port (30pin)
USB 3.0 #0
USB 3.0 #1
eMMC 5.0 8bit
SD3.0 Host
UART #2
I2C #1
UART #0
SPI #1
ADC
GPIO
CPU
Cortex-A15 Quad (2.0 GHz)
Cortex-A15 32KB/32KB I/D-Cache NEONv2 + VFPv4
Cortex-A15 32KB/32KB I/D-Cache NEONv2 + VFPv4
Cortex-A15 32KB/32KB I/D-Cache NEONv2 + VFPv4
Cortex-A15 32KB/32KB I/D-Cache NEONv2 + VFPv4
SCU and ACP
2MB L2-Cache with ECC
128-bit AMBA ACE Coherent Bus interface
Cortex-A7 Quad (1.4GHz)
Cortex-A7 32KB/32KB I/D-Cache NEONv2 + VFPv4
Cortex-A7 32KB/32KB I/D-Cache NEONv2 + VFPv4
Cortex-A7 32KB/32KB I/D-Cache NEONv2 + VFPv4
Cortex-A7 32KB/32KB I/D-Cache NEONv2 + VFPv4
SCU
512KB L2-Cache
128-bit AMBA ACE Coherent Bus interface
Multimedia
GPU
ARM Mali-T628 MP6 (600MHz) OpenCL 1.1 Full profile OpenGL ES 1.1, 2.0 and 3.0
JPEG Enc/Dec
MFC 1080p 60 Enc/Dec
DRAM
LPDDR3 933MHz 32bit 2-port 14.9Gbytes/sec 2GByte PoP
USB 2.0 Host
I2C
I2S
HDMI
eDP
USB Hub + Ethernet Controller
PMIC
CPU/GPU/DRAM Power Monitoring
Audio Codec
Out
In
4 x USB Host Port
Ethernet 10/100
DC 5V/4A
Hedphone / Microphone
HDMI Type-D
DisplayPort

Figure 4.3: ODROID-XU3 block diagram. Taken from [20]

4.2.2 Intel x86 architecture

We used three different machines running CPUs from the x86 Intel instruction family. The Intel x86 machines are the most widely used solutions for server and workstation applications nowadays. According to Gartne [5], Intel market share accounts for 92% of worldwide sold server processors in 2013. Following the trend, the majority of computing nodes at CERN Intel x86 CPUs.

The Intel Xeon used in the experiments had RAPL enabled (refer to Chapter 3), which allowed us to measure energy consumption accurately at a fine-grained level. Since the Intel Atom and Intel Quad machines did not have RAPL technology enabled, we used a clamp power meter to measure the energy consumed by the CPU at a given time during our experiments.

The different types of measurements within the same architecture and its possible affect on he final result are discussed in the Analysis section.

Intel Xeon

The Intel Xeon family [15], is an Intel x86 microprocessor release that is targeted to non-consumer workstation. The main target of this family of processors are data centers, servers and embedded devices. Based on Intel's official web page [15], Xeon family of processors offers a range of solutions for real time analytic services. Among those capabilities, Intel Xeon features an implementation of RAPL, which is a mechanism that provides feedback on the performance impact of power limiting.

Intel Quad

The Intel Quad Core 2 family [14] is a consumer targeted 64-bit x86 four core microprocessors CPU. The first Intel Quad core 2 qas released in 2006 and according to official sources this family of processors is in end of life status and seen as legacy technology. This family of CPUs has thermal monitoring techniques but does not include the RAPL component as the more recent Intel products.

Intel Atom

Intel Atom [13] is the family of Intel CPUs targeted at low power consuming devices, such as laptops, smart phones and embedded devices. Its ultra-low-voltage 32 bits architecture was design for energy efficiency and low power consumption. As in Intel Xeon, RAPL was not included in this family of CPUs.

The chips that are part of the Intel Atom family are the ones closer to the ARM chipset in terms of design, purpose and power consumption.

4.3 Experiments setup

4.3.1 First set of experiments

Hardware specifications

For the first set of experiments, we used three machines with different hardware setups. The three machines differ in architecture and purpose. We ran the same workloads in a x86 Intel Atom and Intel Quad for comparison.

Below, we outline the most important specifications of the hardware setups we used for the experiments.

Intel Atom

kernel & sys: Linux cernvm 2.6.32431.5.1.el6.x86_64

OS: Scientific Linux release 6.5 (Carbon)

CPU: 4x Intel^{TM} Atom^{TM} CPU D525 1.8GHz

Memory (MemTotal): 3925084 kB (4GB)

For more detailed specs refer to [12]

Intel Quad

kernel & sys: Linux cern-vm 2.6.32-431.5.1.el6.x86_64

OS: Scientific Linux release 6.5 (Carbon)

CPU: 4x Intel^{TM} Core^{TM}2 Quad CPU Q9400 2.66GHz

Memory (MemTotal): 7928892 kB (8GB)

For more detailed specs refer to [14]

ARM Viridis

kernel & sys: Linux 3.6.10-8.fc18.armv7hl.highbank

OS: Fedora release 18 (Spherical Cow)

CPU: 4x Quad-Core ARM^{TM} CortexA9^{TM} processor 1.4GHz

Memory (MemTotal): 4137780 kB (4GB)

For more detailed specs refer to [8]

Software and workload

We used the CMSSW framework in the generation-simulation mode (GEN-SIM). The work-flow performs a Monte Carlo simulation of 8 TeV LHC Minimum bias event using Pynthia8 (generation step), followed by Simulation with Geant4 (simulation step). At the time of the experiments, the CMSSW port for ARM had limitations on the multi-threading support. We wanted to study the energy consumption of each hardware setup given different core load. Thus, we spin up different processes instead of threads. The core-load levels used were 1/4, 1/2, 1 and 2 processes per number of physical cores.

Metrics

The energy efficiency metric used in this study is the ratio of performance per power consumed (in Watts). Performance consist on the average of events computed per second. Considering this metrics for comparing energy consumption, we consider a system to be as energy efficient as higher the ratio *nr_of_events/s/W* is.

Given the hardware disparities of the setups we had in place to run our experiments, we used the performance (average events computed per second) as a way to uniform the results.

Tools for measuring energy consumption

For this set of experiments, we performed physical measurements using an external clamp meter. The clamp was a Mini AC/DC Clamp meter Mastech MS2102 AC/DC (see Figure 4.4). The clamp meter supports a maximum of 200A current, which was enough for our experiments. In addition, it presents an accuracy of +-2.5%. For more specifications about the clamp used, refer to [REF].

4.3.2 Second set of experiments

Hardware specifications

For the second set of experiments, we again used three machines with different hardware setups. As in the first set of experiments, the three machines differ in architecture and purpose. The ARM Virdis - which was used in the first

Figure 4.4: Mastech MS2102 clamp meter used to measure energy consumption. Taken from [] - cite Mastech website

Machine codename	**Architecture**	**CPU**	**Nº active cores**	**RAM**	**Notes**
ARM Viridis	Quad-Core ARM$^{\text{TM}}$ CortexA9$^{\text{TM}}$	ARMv7 32b (A7)	4	2 GB	Server class ARM processor with ipmitools
Intel Atom	Intel Bonnell$^{\text{TM}}$	Atom D525	4	4GB	No internal measurement tool
Intel Quad	Intel Sandy Bridge$^{\text{TM}}$	Quad CPU Q9400	4	8GB	No internal measurement tool

Table 4.1: Summary of the 1SE specifications

set of experiments - was also used during the second set of experiments. In addition, we also used another machine powered by an ARM CPU. The ARM Odroid is a development board manufactured by HardKernel [20]. ARM Odroid provides a cheap and easy way to develop hardware and software on top of an ARM architecture. As for the Intel architectures, we used Intel Xeon, a machine from the Intel Sandy Bridge family and powered by an Intel R5-2650 CPU. This machine was part of a server rack and it was intended for high performance scientific computation in a production scenario.

Below, we outline the most important aspects of the hardware setups we used for the experiments.

ARM Viridis

kernel & sys: Linux 3.6.10-8.fc18.armv7hl.highbank

OS: Fedora release 18 (Spherical Cow)

CPU: 4x Quad-Core ARM^{TM} CortexA9^{TM} processor 1.4GHz

Memory (MemTotal): 4137780 kB (4GB)

For more detailed specs refer to [8]

ARM Odroid

kernel & sys: Linux 3.10.24 LTS

OS: Ubuntu 14.04.3 LTS (Trusty Tahr)

CPU: 2x A15 and/or A7 cores(big.LITTLE technology) - A7 at 1.4GHz and A15 at 2GHz

Memory: 2GB

For more detailed specs refer to [19]

Intel Xeon

kernel & sys: Linux cern-vm 2.6.32-431.5.1.el6.x86_64

OS: Scientific Linux release 6.5 (Carbon)

CPU: 4x Intel$^{\text{TM}}$ CPU E5-2650 2GHz

Memory: 252GB

For more detailed specs refer to [16]

Software and workload

We used the ParCullCMS mode of CMSSW for generating the workload. The ParFullCMS mode is a multi-threaded Geant4 [39] benchmark. It uses a complex CMS geometry for the event simulation and has the advantage of being multi-threaded in both Intel and ARM architectures. As in the first set of experiments, we measured the energy consumed by the machine under different physical core loads. The core-load levels used were 1/4, 1/2, 1 and 2 threads per number of physical cores.

Metrics

As in the first set of experiments, the energy efficiency metric used in this study is the ratio of performance per power consumed *nr_of_events/s/W*. The hardware setups used in the second set of experiments differ in specs and features. Therefore, we used the this metric as a way to uniform the results.

Tools for measuring energy consumption

For the second set of experiments, we performed both internal and external measurements in the Intel Xeon and ARM Odroid. On the ARM Viridis, we performed only internal measurements given the lack of a tool that would perform with the same degree of accuracy than the tools used for Intel Xeon and ARM Odroid. All the tools used to measure energy consumption were embedded in the hardware setup of the machines.

For the ARM Odroid, we used a Texas Instrument power monitor chip (TI INA231) for internal measurements. The TI INA231 allowed us to sample the energy consumed by the cores and DRAM at a frequency rate of microseconds. For the external measurements on the ARM Odroid, we used an external plug-in power monitor with a computer interface for sampling and storing the results.

For the Intel Xeon machine, we used the Running Average Power Unit (RAPL) technology to perform internal measurements. The RAPL allowed us to sample the energy consumed by the CPU's package, DRAM and cores. For the external measurements, we used an API provided by the server rack's PDU. This API provides a measure sampling rate of around 1 second.

Machine codename	**Architecture**	**CPU**	**N° active cores**	**RAM**	**Notes**
ARM Odroid	Quad-Core ARMv7^{TM}	A15 and or A7 cores(big.LITTLE technology)	4	2 GB	Development board with TI INA231 chip
ARM Viridis	Quad-Core ARM^{TM} CortexA9^{TM}	ARMv7 32b (A7)	4	2 GB	Server class ARM processor with ipmitools
Intel Xeon	Intel Sandy Bridge^{TM}	CPU E5-2650	32	252 GB	System on a rack with RAPL

Table 4.2: Summary of the 2-SE specifications

For the ARM Viridis, we used the capabilities of the Intelligent Platform Management Interface (IPMI) [17] included in the server from origin. The IPMI is a chip that runs as a separate subsystem and is attached to the motherboard. The ARM Viridis implementation of IPMI provide several capabilities, namely internal hardware energy monitoring. We leveraged the IPMI tools to perform internal energy consumption of the ARM cores during the experiments

4.4 Summary

We have performed several experiments under different hardware setups. Our main goal was to understand how the ARM and Intel architectures perform under similar workloads from an energy consumption standpoint.

In this chapter we outlined the setup of the machines used during the experiments.

The hardware setups were chosen given their similarity and possibility of a reliable comparison and hardware availability. It is important to note that both ARM Viridis and ARM Odroid machines are much more recent than the compared Intel hardware. All the ARM machines used were still a technology that was yet to find production stability at the moment of the experiments. On the other hand, the Intel architecture used in this study was widely used in real HPC applications at the time of this study.

For this study, we assume that the RAPL, the internal TI INA231 chip and the IPMI tools for internal energy consumption measurement are similarly accurate and would produce the same results if interchanged.

Chapter 5

Analysis

In this chapter, we present our analysis based on the experiments outlined in the last chapter. The scope of the analysis presented in this section is twofold: to compare the platforms from an energy efficiency perspective and analyze the tools and techniques used on the different experiment sets.

The first section and second section of this chapter analyze the different tools and techniques used to perform the experiments, as well as the results. The first section analyses the first set of experiments, whereas the second section is dedicated to analyses the second set of experiments. As stated before, the difference between first set of experiments and second set of experiments has to do with the machine setups and the tools used to measure the energy consumed. The data and results of the experiments are shown along the analysis.

In the final of this section, we outline the highlights of the analysis for each set of experiments.

5.1 First Set of Experiments

In figures 5.1, 5.2 and 5.3 it is plotted the energy measurements from the beginning until the end of the event generation-simulation by the CMSSW. The energy measurements were done using a meter clamp. The energy measured is represented in the Y-axis and the X-axis represents the time of the experiment in samplings. For each experiment, a sample corresponds to the same time.

In the figures 5.5, 5.6 and 5.7, we trimmed out the initialization stage and connection stage of the workload and only show the event processing stage. Whereas the Y-axi represents the energy measured in Watts, the X-axis represents the time spent until the correspondent energy sampling.

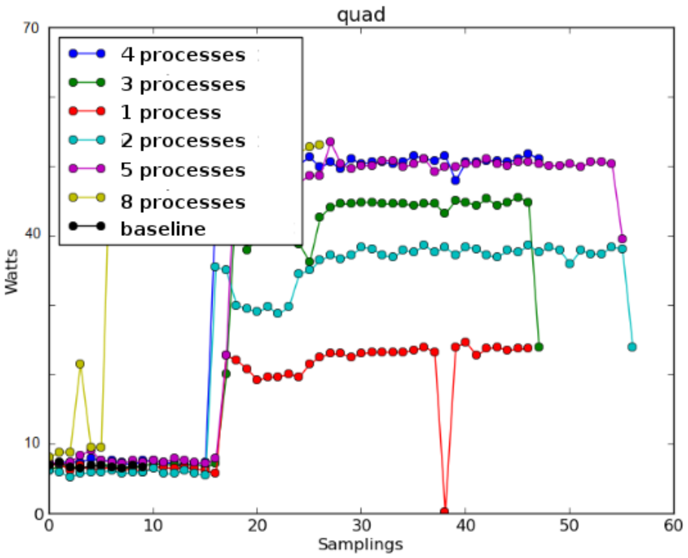


Figure 5.1: All stages of the CMSSW experiments on Intel Quad

Finally, the figures 5.8 compare the time spent by each of the hardware setups to process the tasks.

CMSSW stages

Based on the figures 5.1, 5.2 and 5.3, we can distinguish three different patterns of energy consumption during the expriement. We refer to each pattern as being part of a different CMSSW stage. The stages can be better identified when plotting the memory workload and the CPU usage (see 5.4).

The first stage consists is the initialization process. During this stage, the memory is the main module being used and thus, it is out of the scope of this work to analyze this stage in depth.

The second stage is the connection phase. The goal of this stage is to fetch the meta-data from the CERN servers that allow the event generation-simulation. The meta-data is needed to perform the reconstruction of the events. Once again, during this stage the CPU load is low when compared to the memory workload.

The third stage corresponds to the event processing. This last stage is CPU intensive and it has the most relevant data to to our study, since we our goal is to compare the energy efficiency of the different CPUs. The event processing stage alone is represented by the figures 5.5, 5.6 and 5.7.

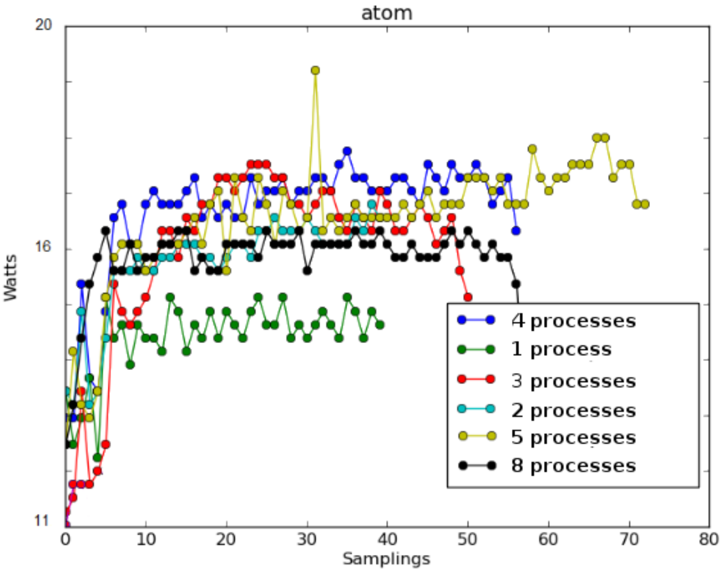


Figure 5.2: All stages of the CMSSW experiments on Intel Atom

Relative importance of the stages

The most important stage when studying the energy efficiency of workload with the CMSSW is the last stage. There are three main reasons for that: Firstly, the CMSSW configuration at CERN has caches that speed up considerably the second stage, thus reducing the energy consumed in the connection stage. Secondly, the first and second stages are not CPU intensive. Lastly, the processing stage is the only one that the energy consumption is dependent on the amount of events. Therefore, given any large amount of data to be processed, the last stage will consume so much more energy than the former stages that the first two stages will become irrelevant in terms of overall energy consumption. Therefore, we focus our energy consumption analysis on the event processing stage only. The event processing stage alone is represented by the figures 5.5, 5.6 and 5.7

Over-committing CPU and energy efficiency

We consider a CPU to be over-committed when it has to process more threads or processes than the physical cores available.

If we consider each hardware setup individually, the time needed for running the three stages of the experiment is roughly the same, if the CPU is not over-committed. When the number of processes exceed the number of available cores, the time to process the events increases since there are no available

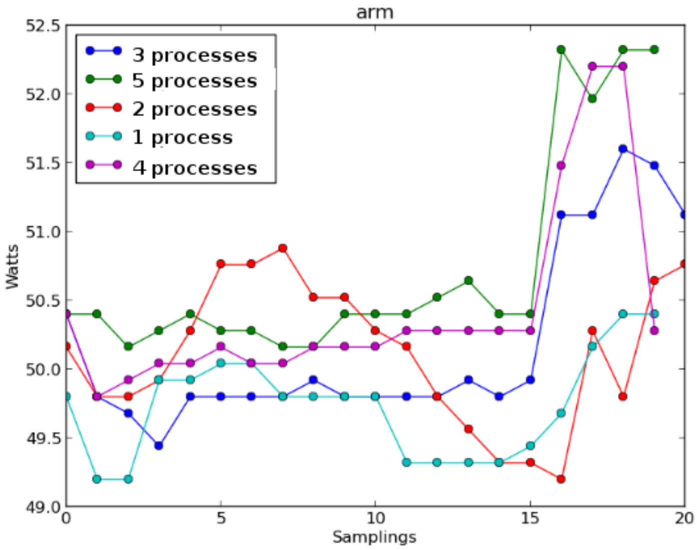


Figure 5.3: All stages of the CMSSW experiments on ARM Viridis

cores to process the events concurrently. In the over-committed situation, the time increase follows the ratio *nr_of_processes/nr_of_cores_available*. For example, if the number of processes running is 6 and the number of cores available is 4, the time needed to process the events increases roughly 2/3 compared to when the CPU is not over-committed.

In terms of energy consumed by the CPU, we do not find any outstanding difference in terms of overall energy efficiency by comparing CPUs that are over-committed vs non over-committed, as we can seen in the Figure 5.9. However, we expect that if the ratio *nr_of_processes/nr_of_cores_available* is large enough, it can affect negatively the energy performance given the energy overhead spent when the jobs are being swapped.

Time comparison

When comparing the time taken by the different architectures to process the same task (Figure 5.8), the pattern is evident. Regardless the number of processes, the Intel Quad architecture is faster than Intel Atom. In addition, ARM Viridis is faster than Intel Atom. This fact is due to the architectures characteristics and its specifications, most notably the CPU clock speed.

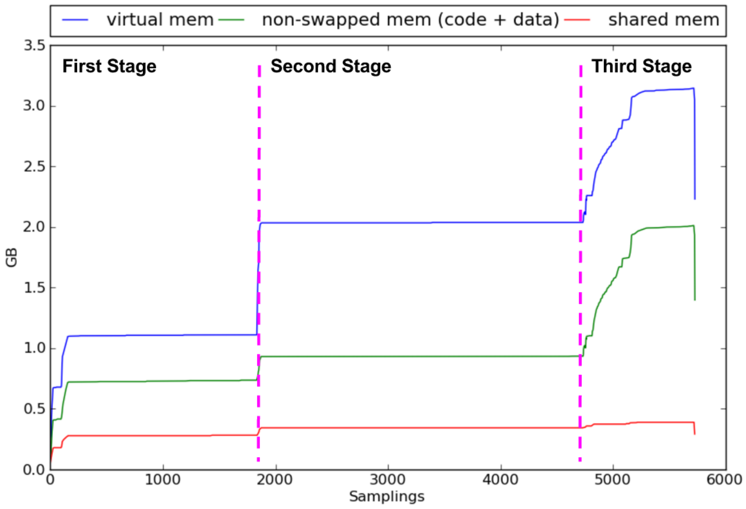


Figure 5.4: Memory usage during the 3 different stages

Energy efficiency comparison

Given the metrics used in this study (see Metrics section in the Experiments chapter), it is clear that systems are proportionally energy efficient with its ratio performance per watts. Therefore, by analyzing the Figure 5.9, it is evident that given the architectures and its configurations, ARM architecture outperforms in terms of energy efficiency its concurrence in all considered scenarios. In addition, we conclude that between Intel architectures, Intel Atom is more energy efficient than the Intel Quad.

Measuring tools: external monitoring

For this set of experiments, the external samples were acquired and recorded manually. This factor had a visible impact on the resolution of the measurements. It is clear that the plots show spikes and rough transitions between samples. Moreover, the error tends to increase proportional to the human interaction with the experiment. Therefore, we conclude that it is more effective to use digital and automated ways to sample and log the data acquired during the measurements.

Measuring tools: software-based monitoring

We used software measurement tools to get an estimated energy consumption by the memory and other system components. In this particular set of experiments, the memory energy measurements done with software were

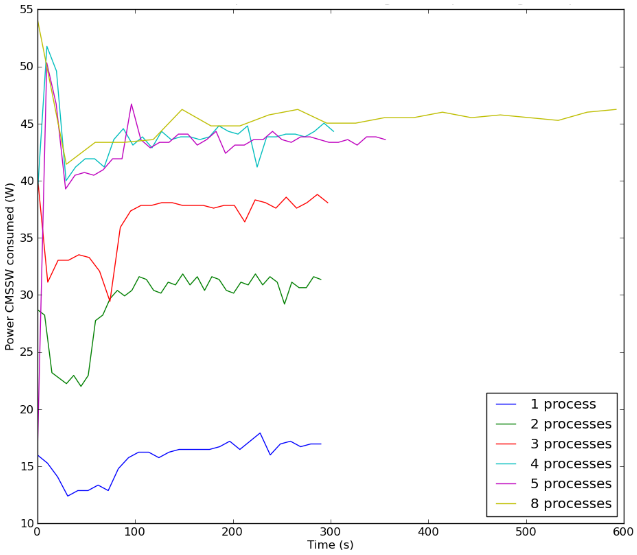


Figure 5.5: CMSSW experiments on Intel Quad - event processing stage

of particular help to distinguish the different stages, which existence was unknown before the experiment. The software-based tools can be used as a decision support and for learning about unknown and unexpected system behaviors. Thus, even if the output does not directly show information about energy consumption of the system, it can be important to support and explain expected - and unexpected - behaviors.

5.2 Second Set of Experiments

Energy efficiency comparison between Intel Xeon and ARM Odroid

In the Figure 5.10, we can see the energy efficiency comparison of Intel Xeon and ARM Odroid. The righmost plot represents the internal energy measurements, whereas the leftmost plot represents the external energy measurements. As in other energy efficiency comparisons in this study, we used the metrics *nr_of_events/s/W* to represent the energy performance of the measured systems.

The main conclusion from 5.10 is that ARM Odroid outperforms Intel Xeon in both internal energy efficiency and external energy efficiency.

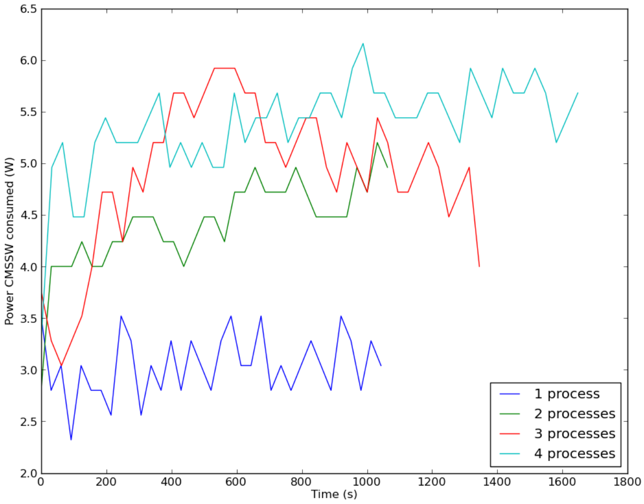


Figure 5.6: CMSSW experiments on Intel Atom - event processing stage

Energy performance and over-committed CPUs

It is noticeable that ARM Odroid has a significant energy performance decline when its cores are over-committed. It is also interesting to see that the energy performance decline in the ARM Odroid is relatively larger on the internal energy measurements. One of the reasons we found in our raw results to explain this phenomenon is the large increase of time taken to process the events when the cores are over-committed. Thus, even if the cores are consuming the same power during the event processing stage, the energy efficiency will decrease with the time taken to process the events.

On the other hand, the energy performance of Intel Xeon does not seem to be significantly affected when over-committed. This can be explained by the fact that Intel Xeon took roughly the same time to process the events when using one core per event and half a core per event.

We believe that the different results between ARM Odroid and Intel Xeon discussed below are due to the fact that ARM Odroid is a development board and it does not implement sophisticated techniques such as Hyper Threading Technology (HTT) by Intel [9]. According to Intel, HTT delivers two processing threads per physical core, which allows highly threaded applications to be processed faster. It is expected that if the ratio of *nr_of_threads/core* would be larger than 2, energy efficiency of Intel Xeon would start to decline.

We believe that if we would over-commit Intel Xeon with more than 4 threads per core, the time to process the workload would increase, which

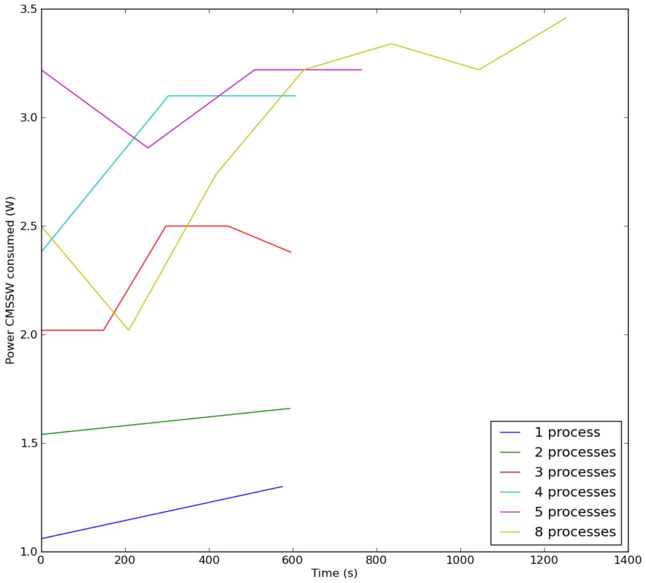


Figure 5.7: CMSSW experiments on ARM Viridis - event processing stage

would be followed by a degradation of energy performance.

Measurement tools and techniques

The internal measurement tools used in ARM Odroid and Intel Xeon provide a fine grained resolution to the core level. The TI INA231 and RAPL chips can isolate the pp0, which consists of ALU, FPU, L1 cache and L2 cache when performing energy measurements.

On the other hand, as stated in [10], the lower resolution that IPMI tools offers for internal measurements include energy consumed by the 0P9V, 1P8V, VDD and Vcore rails, which includes the system on the chip, DRAM, Temperature Sensors, and ComboPHY Clock.

As a result of this measurement discrepancy, 5.11 shows that ARM Viridis performs worse than any other machine. We believe that this result can be misleading, due to the fact that the tools used to measure the energy consumed by each of the setups measure different components in the CPU. We believe that if components measured in the ARM Viridis would be same as the components measured on the ARM Odroid and Intel Xeon measurements, we would obtain a different result. Namely that ARM Viridis would, at least, perform better that Intel Xeon from an energy consump-

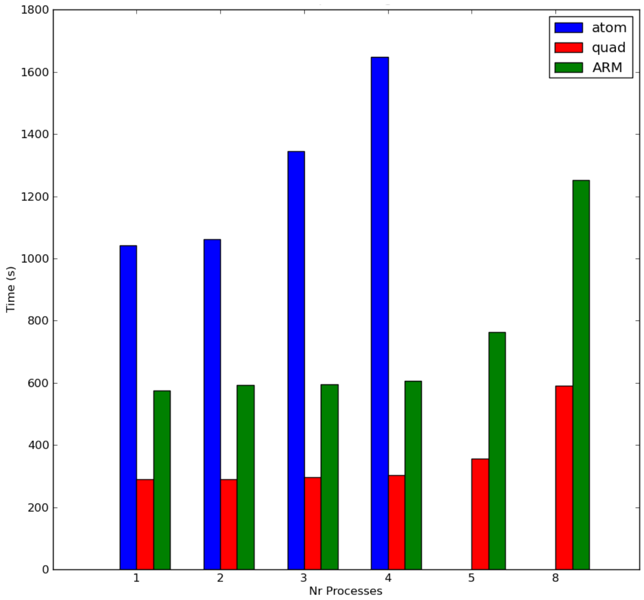


Figure 5.8: Processing time comparison

tion perspective. Given the actual setup, we can not scientifically directly compare the results.

Comparison between First set of experiments and Second set of experiments

When we compare the main results of the first set of experiments (Figure 5.11) and second set of experiments (Figure 5.10) we may be inclined to conclude that the setups in the second set of experiments presented an overall more efficiency than the setups the first set of experiments. Again, the used measurement tools play an important role and should not be disregarded when analyzing the results. In the first set of experiments, we only performed external measurements. Thus, we discard the possibility to compare the first set of experiments results with the results of the internal measurements of the second set of experiments. As for the external measurements performed in both set of experiments, the tools for measuring the energy consumption of both experiments have distinct resolution and grain. In the first set of experiments, we used the clamp meter for measurements in all setups. As for the second set of experiments, we used embedded and computer-assisted

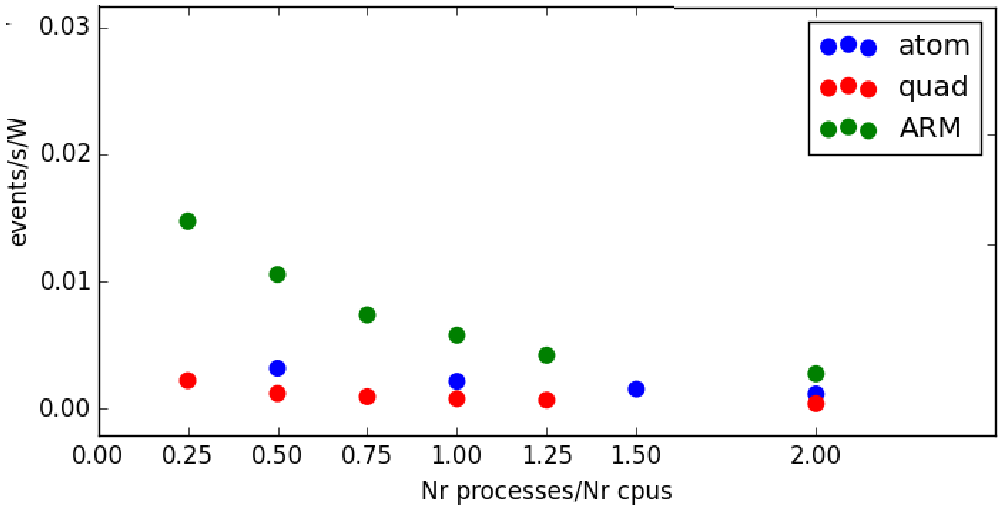


Figure 5.9: Energy efficiency comparison for the first set of experiments - External measurements

tools to perform the external measurements. This discrepancy of tools, its resolutions and errors, make it difficult to compare the results of the first set of experiments and second set of experiments.

However, we can conclude that ARM architecture outperforms the Intel architectures in each and every experiment, regardless the measurement tools and methodologies used.

5.3 Conclusions

The main conclusion of our experiments is that given the setups used in our study, the ARM chip-sets outperform Intel in terms of power efficiency in every experiment.

We learned that the gen-sim mode of CMSSW has different stages and the most relevant from an energy consumption point of view is the latest one, when the events are processed.

In addition, we learned that the tools used to make the experiments play a crucial role in the whole experiment. It is important to assure that the measurement tools and methodologies in use are compatible and suitable to produce results that can be compared. This aspect can be hindered based on the availability of hardware and measurement tools. cite

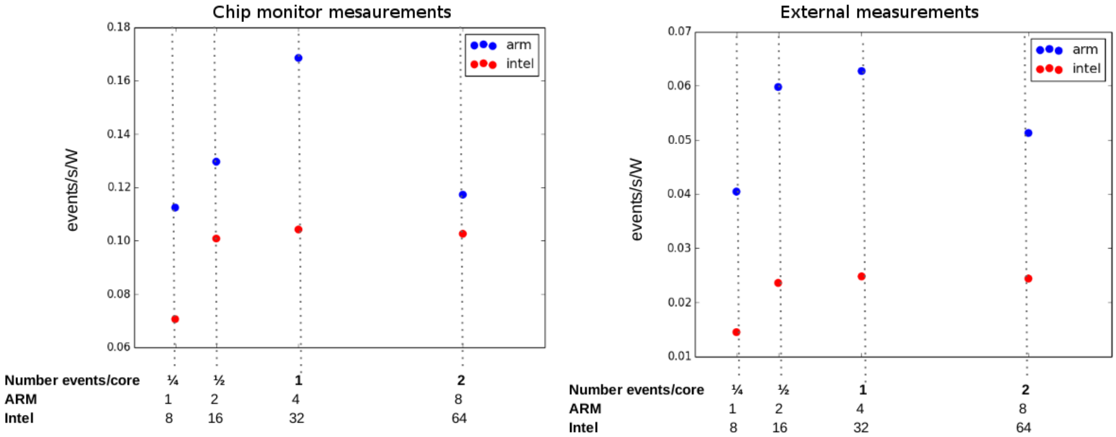


Figure 5.10: Multithreaded ParFullCMS comparison between Intel Xeon and ARM Odroid

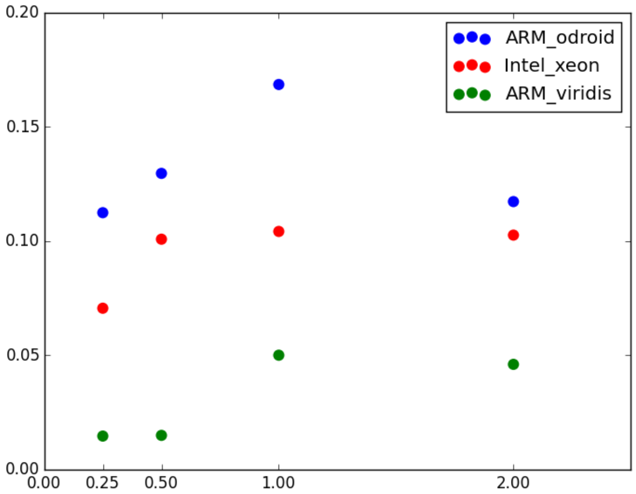


Figure 5.11: Multithreaded ParFullCMS comparison between Intel Xeon, ARM Viridis and ARM Odroid

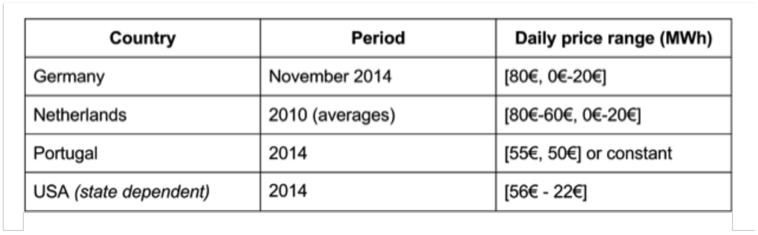
Chapter 6

HTC in a dynamic energy pricing market

One of the ways to utilize the conclusions from the experiments conducted in this study is to actively lower the energy bill in HTC. One approach could be to schedule jobs between more energy efficient but slower ARM architectures and the less energy efficient but faster Intel machines. This approach makes sense in a multi energy price ecosystem. A multi energy price ecosystem is an energy market where the prices float according to the overall power grid usage.

In this chapter, we present a study about the potential of an algorithm that schedules workload to machines with different energy and computation performance, based on the the daily dynamics of energy price. The main goal is to leverage computing heterogeneity to achieve the optimal ratio between work produced and price paid.

6.1 Dynamic electricity pricing model



Country	Period	Daily price range (MWh)
Germany	November 2014	[80€, 0€-20€]
Netherlands	2010 (averages)	[80€-60€, 0€-20€]
Portugal	2014	[55€, 50€] or constant
USA *(state dependent)*	2014	[56€ - 22€]

Figure 6.1: Examples of dynamic power energy pricing in different markets

We will use a simplified dynamic pricing model based on the empirical research of such models in real power grids. Based on some examples obtained, we can conclude that not all the countries adopt a dynamic electricity pricing model 6.1. For the sake of this study, we will consider a hypothetical case where the dynamic pricing model works as in 6.2. The red line in 6.2 represents the electricity price along the day. For the sake of simplicity, during this study we define that the price of electricity can is 60 euros/Mwh during 12 hours per day and 20 euros/Mwh during the remaining 12 hours of the day.

As we can see based on the 6.2 and 6.1, our simplified energy model presents a similar pattern to the energy prices of some countries.

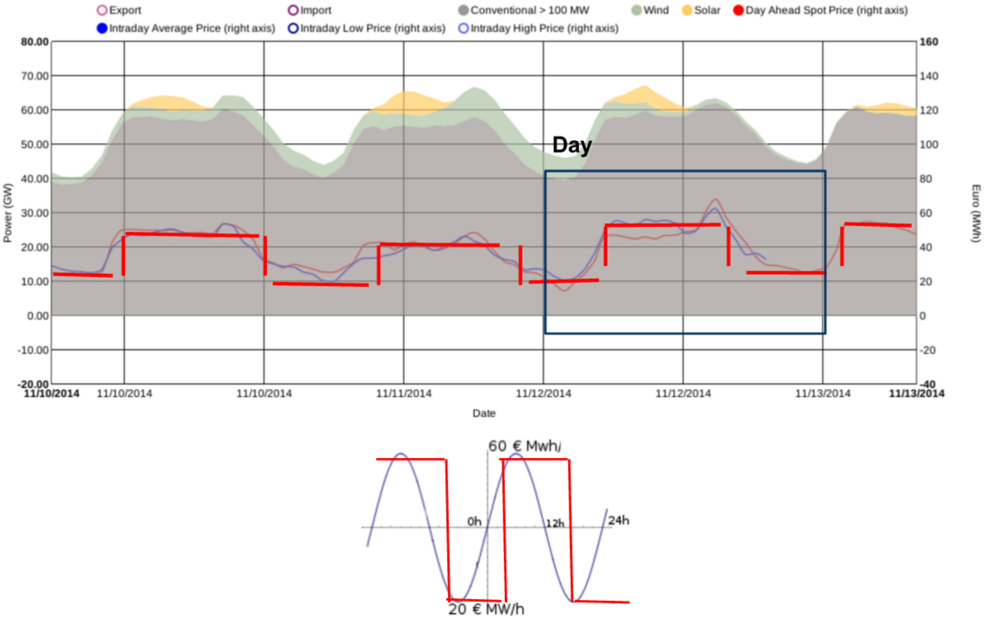


Figure 6.2: Simplified pricing model based on the Germany energy market

6.2 Scheduling algorithm

Problem formulation

The main goal of the algorithm is to lower the electric bill in dynamic electricity markets. The algorithm schedules HPC workload among nodes with different energy profiles, depending the energy price at the time. In addition, the algorithm should ensure that a certain minimum amount of workload is processed. Thus, the algorithm input can be defined as (see also 6.3):

Energy profile of the machines:

Energy efficiency of the machines (ev/s/W);

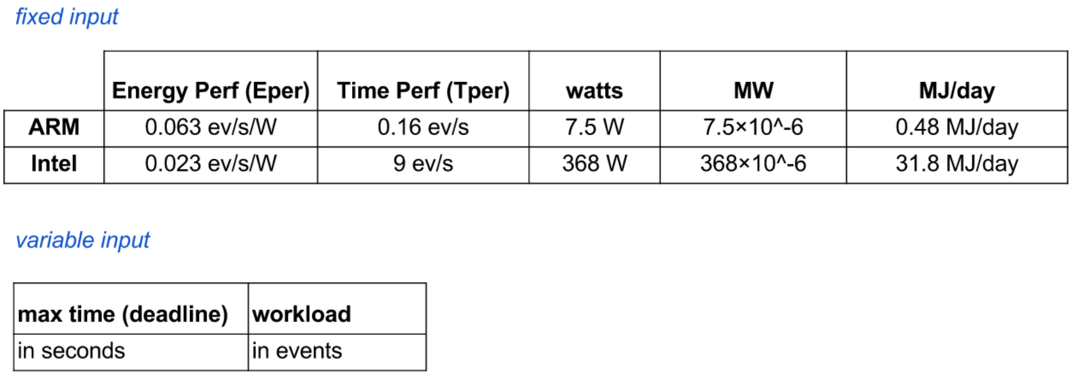
Time performance (ev/s);

Computing requirements:

Time deadline (s);

Nr. events to be processed (ev);

Therefore, given a set of machines with different energy profiles; the computing requirements (how many events must be processed in how much time); and the energy pricing dynamics during 24h, *what is the optimal machine scheduling that ensure the computing requirements and achieve the lowest price budget at the end of 24 hours* ?



fixed input

	Energy Perf (Eper)	Time Perf (Tper)	watts	MW	MJ/day
ARM	0.063 ev/s/W	0.16 ev/s	7.5 W	7.5×10^-6	0.48 MJ/day
Intel	0.023 ev/s/W	9 ev/s	368 W	368×10^-6	31.8 MJ/day

variable input

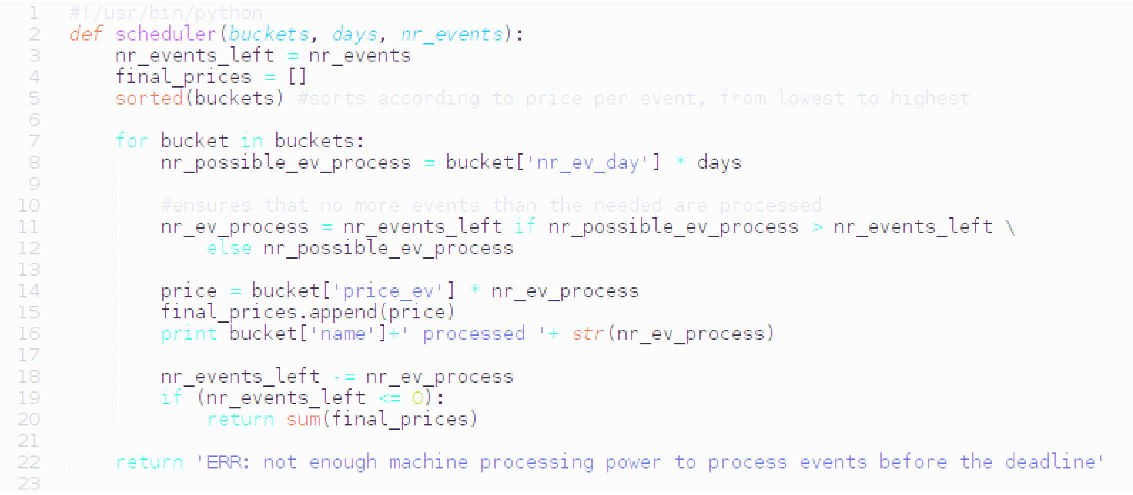
max time (deadline)	**workload**
in seconds	in events

Figure 6.3: Example input for the scheduling algorithm

The scheduling algorithm

The scheduling algorithm is presented in 6.4. For simplicity sake, the algorithm is written in Python.

The main idea behind the construction of the algorithm is that it should assign the maximum number of events to be processed to the lowest priced configuration possible, given the existent constrains (deadline). For example, if it is possible to process all the data only using the ARM architecture in the



```
1   #!/usr/bin/python
2   def scheduler(buckets, days, nr_events):
3       nr_events_left = nr_events
4       final_prices = []
5       sorted(buckets) #sorts according to price per event, from lowest to highest
6
7       for bucket in buckets:
8           nr_possible_ev_process = bucket['nr_ev_day'] * days
9
10          #ensures that no more events than the needed are processed
11          nr_ev_process = nr_events_left if nr_possible_ev_process > nr_events_left \
12              else nr_possible_ev_process
13
14          price = bucket['price_ev'] * nr_ev_process
15          final_prices.append(price)
16          print bucket['name']+' processed '+ str(nr_ev_process)
17
18          nr_events_left -= nr_ev_process
19          if (nr_events_left <= 0):
20              return sum(final_prices)
21
22      return 'ERR: not enough machine processing power to process events before the deadline'
23
```

Figure 6.4: Proposed scheduler algorithm written in Python

lowest and highest energy pricing times, then there is no need to use the Intel machines to process the data. This way it is possible to reach the optimal scheduling of processing power given a deadline and several machines with different configurations.

Algorithm walk-through

The input of the algorithm are the number of days that we have to process the data (deadline); the number of events to process; and a set of buckets. We define a bucket as a tuple of *(machine, energy_price_level)* . The table 6.5 shows an example of what buckets can be. The expected output os the final price of the scheduled processing. If the processing power of the data center is not enough to complete the task before the set deadline, the output will be an error.

Based on the code in 6.4:

Line 5 Sorts the buckets from lowest to highest price per event. The goal is to use the as much processing power of the cheaper buckets as possible.

Line 7 The algorithm goes through all the available buckets. It starts by considering the cheapest options and only uses the more expensive if needed.

Line 11 Ensures that the money spent by the last bucket will only take into consideration the events left.

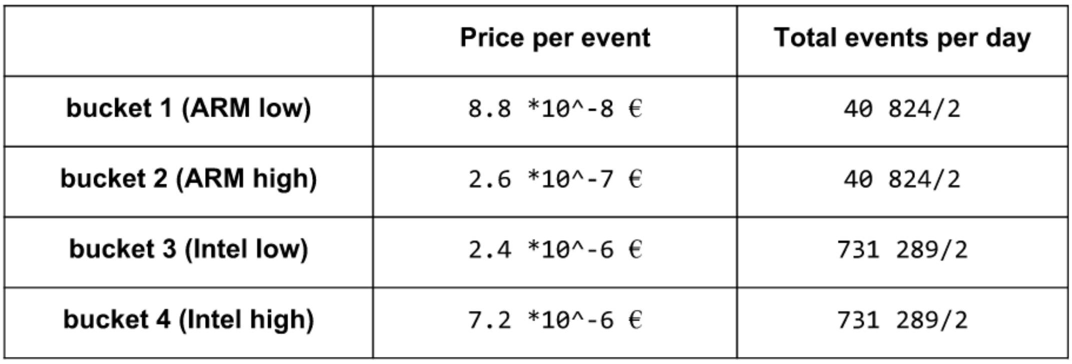
Lines 14-15 The money spent by a given bucket is the number of events processed times the price per event.

Lines 18-20 When there are no events left to process, finish the algorithm and calculates the final price (the sum of the money spent by all the buckets to compute the tasks)

Lines 22 After the buckets had all the events processed given the deadline, if there is still events left the algorithm throws an errors. In this case, more buckets have to be added of the deadline should be increased.

Use case

Let us consider the values presented in 6.5 as a case scenario to use the scheduling algorithm. The values presented show the energy profile of a hypothetical mini data center set up. The data center has only two machines: a ARMv7 and a Intel x86 server. The energy profile (number of events processed by day and energy consumed) is based on the values from the experiments analyzed in the previous chapters. The prices are based on the simplified dynamic energy pricing model of 60 euros/Mwh in the high pricing hours (comprised of 12h/day) and 20 euros/Mhw in the low pricing hours (comprised on 12h/day)

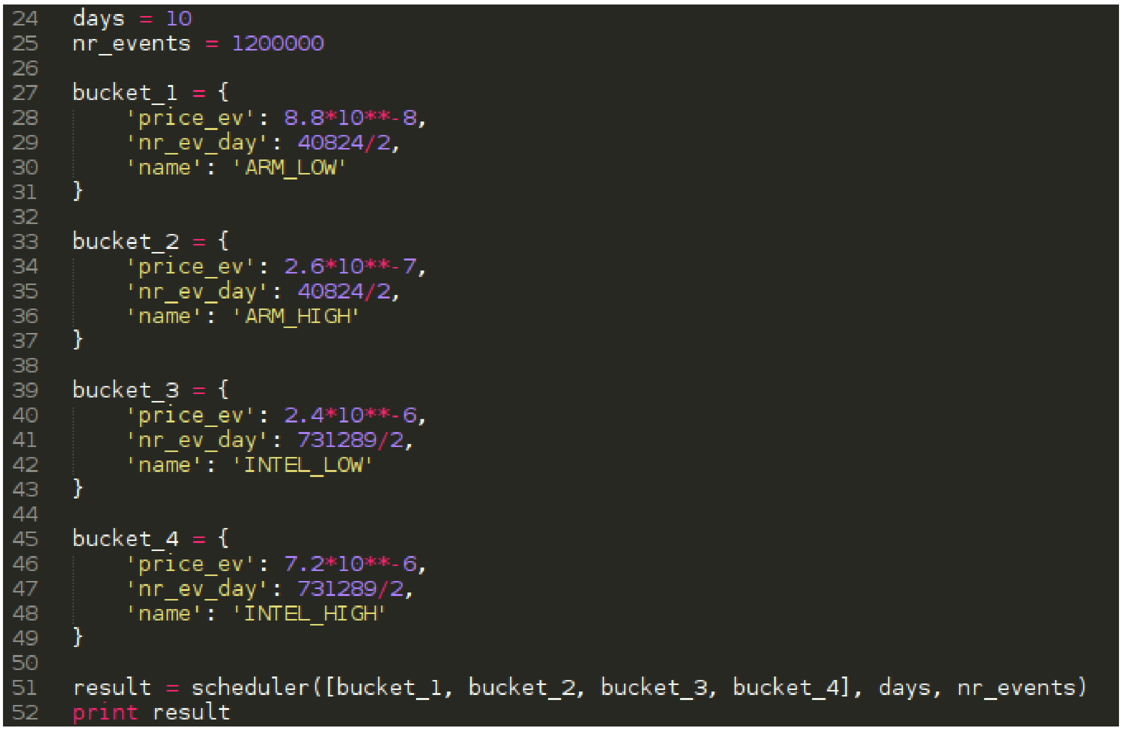


	Price per event	Total events per day
bucket 1 (ARM low)	8.8 *10^-8 €	40 824/2
bucket 2 (ARM high)	2.6 *10^-7 €	40 824/2
bucket 3 (Intel low)	2.4 *10^-6 €	731 289/2
bucket 4 (Intel high)	7.2 *10^-6 €	731 289/2

Figure 6.5: Example input for the scheduling algorithm

Given the data in 6.5, the algorithm should be started as in 6.6.

Given the input 6.6 and the algorithm in 6.4, the result is that the optimal final price for computing 1200000 events in 2 days is 1.97125776 euros, where the ARM machine processed 204120 events in the low pricing



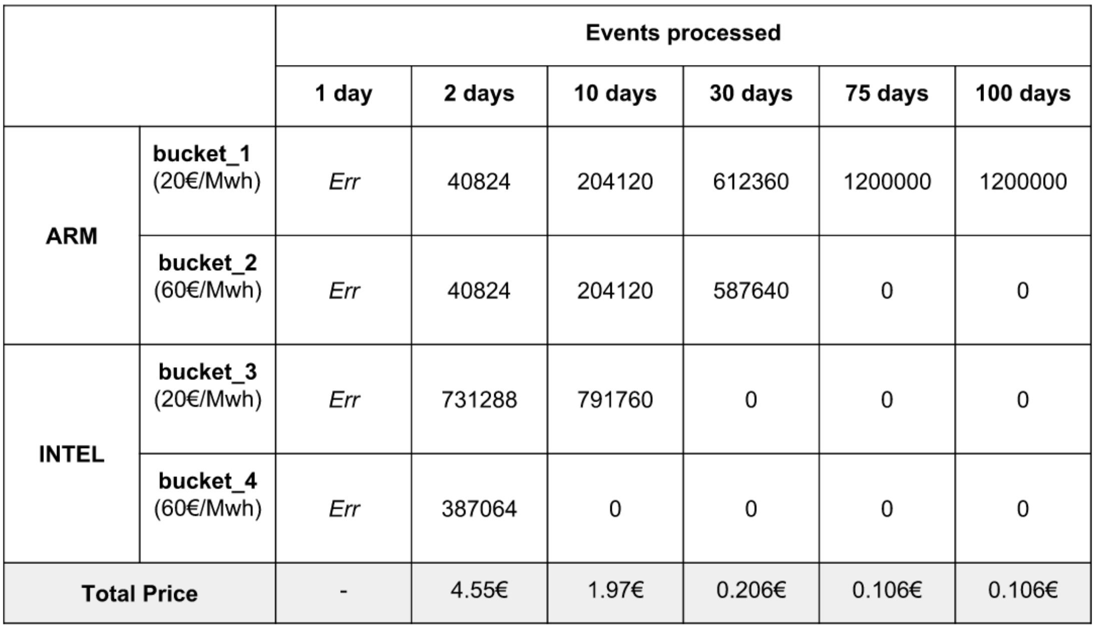
```
days = 10
nr_events = 1200000

bucket_1 = {
    'price_ev': 8.8*10**-8,
    'nr_ev_day': 40824/2,
    'name': 'ARM_LOW'
}

bucket_2 = {
    'price_ev': 2.6*10**-7,
    'nr_ev_day': 40824/2,
    'name': 'ARM_HIGH'
}

bucket_3 = {
    'price_ev': 2.4*10**-6,
    'nr_ev_day': 731289/2,
    'name': 'INTEL_LOW'
}

bucket_4 = {
    'price_ev': 7.2*10**-6,
    'nr_ev_day': 731289/2,
    'name': 'INTEL_HIGH'
}

result = scheduler([bucket_1, bucket_2, bucket_3, bucket_4], days, nr_events)
print result
```

Figure 6.6: Start the algorithm programatically

window and 204120 in the high pricing window during the 2 days. As for the Intel machine, it had to process only 791760 events during the low pricing window. The Intel machine does not need to be working during the high pricing windows in order for the data center to meet the deadline.

We ran the scheduler algorithm with different deadlines to compare the final prices and how much work do the different machines have to perform in the different energy pricing window. The results are condensed in the 6.7.

Based on the information of the table 6.7, if we the deadline is stricter, the price to pay will be bigger as expected. We can also conclude that the machines we have access to are not able to process the 1200000 events in 1 day only. As expected, the more time there is to process the events, the less the algorithm uses the most expensive buckets to process the data. When the deadline is 75 and 100 days, all the events are processed by the ARM machine during the time when the energy price is lower. Thus, this is the cheapest possible case given the data center setups and energy price model used in this use case.



		Events processed					
		1 day	2 days	10 days	30 days	75 days	100 days
ARM	**bucket_1** (20€/Mwh)	*Err*	40824	204120	612360	1200000	1200000
	bucket_2 (60€/Mwh)	*Err*	40824	204120	587640	0	0
INTEL	**bucket_3** (20€/Mwh)	*Err*	731288	791760	0	0	0
	bucket_4 (60€/Mwh)	*Err*	387064	0	0	0	0
Total Price		-	4.55€	1.97€	0.206€	0.106€	0.106€

Figure 6.7: Algorithm results for different deadlines

6.3 Further developments

Well known algorithms such as job shop scheduling algorithm and others can be applied using the same rational. We expect that different algorithms present different results and that the nearly-optimal scheduling algorithm can be achieved with one well studied existing algorithm. We believe that to research heterogeneous HTC in a dynamic energy pricing market further on may present potential to unlock savings in energy budget for data centers. It would be interesting to apply several other scheduling algorithms to this scenario and compare the obtained results.

6.4 Conclusions

Our solution takes a different perspective when compared with related research. Studies conducted by Yan Wang, et al. [60] and Guosun Zeng, et al. [63] do not take the dynamics of electrical pricing into consideration. However, their algorithm is already quite complex and proved NP-complete, to the point they have to come up with heuristic algorithms to apply it in the real world.

Therefore, our approach may have some novelty in a really narrow and still unexplored idea: to develop a scheduling algorithm for heterogeneous HPC that takes into consideration the nodes' energy profile, the dynamic electricity price and also, eventually, the tasks' energy profiling. The algorithm would schedule the jobs in order to minimize the energy consumption and energy bill (note: energy consumption and energy bill are not the same thing), while the deadline is met.

However, there are some open points that we still have might want to consider. First, as Xu Yang, et al. [62] state, it is important to ensure that the hardware existent in the data center is used at its full potential, in order to not waste the investment made when it was purchased. Our solution, though, does not insure that since the idea is to power down/idle machines that are less power efficient in high-peak times. Secondly, from a practical perspective, if we consider only the scheduling between ARM and Intel architectures, it seems not likely that the data center will haver the same software running over both architectures at the same time, give the expertise and investment needed to have the application stack running properly in both architectures (as we witness with CERN's efforts). If we decide to abstract from that point and see the machine's architectures as a black box, then that's not a problem. Thirdly, comparing with other recent research works such as the one conducted by Yan Wang, et al. [60], our algorithm model seems to be over simplifying the problem to an extent that might hinder our purposes of creating a practical and energy efficient scheduling algorithm for heterogeneous HPC under dynamic electrical pricing.

Our solution is a starting point for a deeper study of heterogeneous computing applied to HTC in a dynamic energy pricing market. We used the data and experiments obtained in the previous chapters of this study to test our algorithm in a scenario as closest to the reality as possible. Based on the budget savings shown in the results, our approach shows potential to be used in a production scenario.

Chapter 7

Future Work

From the experiments perspective, we believe that it would be valuable to run more experiments in environments closer to production than development boards. The output of such experiments would present valuable complementary data to what we have got. Another interesting research work would be to understand if the drawbacks of the approach make it inviable in a real production scenario.

In addition, to run more experiments using a methodology where the tools and techniques are as accurate as possible (based on the learning of Chapter 3) and where the results can be compared across all the experiments. This might be difficult to achieve given the different tools available to measure in the different platforms. Another interesting research subject would be to develop a cross platform and accurate way to perform high resolution power consumption measurements. In addition, it would be interesting to develop accurate mathematical models of energy consumption by data centers consisting of heterogeneous (RISC and CISC) processing nodes.

It would be interesting to develop further the scheduling algorithm for dynamic electricity markets and HTC presented in the Chapter 6. In addition, to apply the same idea to different well known scheduling algorithms. Also, to improve the energy model used and increase the complexity of the energy profiles in order to the results to be as close to the reality as possible. Another interesting research work would be to understand if the drawbacks of the approach taken make it inviable to use in a real production scenario.

Chapter 8

Conclusions

Energy consumption has has become a major bottleneck in HTC and scientific computing, where the amount of data to analyze has been increasing many-fold every year. Besides the economical issues of the increasing energy consumption, social and environmental concerns should be also considered. Therefore, to research how to build and develop energy efficient HTC systems has become of paramount importance among the research community.

The goal of this research was to understand whether RISC architectures are capable of improving the energy efficiency of scientific computing without performance degradation, when compared with the current x86 Intel architectures.

In order to achieve that goal, we conducted searcher on tools and techniques for measuring energy consumption at different system levels. In addition, we conducted experiments that aimed at comparing the energy performance of ARM and Intel architectures, working under real world workloads and frameworks. Finally, we used the results of the experiments to develop a scheduling algorithm that optimizes the electricity bill of heterogeneous data centers working in a dynamic electricity pricing markets.

Our main contributions are:

- We researched and outlined best practices for different system levels. The results of this work were published in the conference proceedings of the 16th International workshop on Advanced Computing and Analysis Techniques in physics research (ACAT'2014). The article was called *Techniques and tools for measuring energy efficiency of scientific software applications*

- Our experiments and research shown that ARM architecture shows potential for an energy savings in HTC when compared to the x86 systems widely used nowadays.

- Our research shown that heterogeneous computing can be leveraged in HTC in markets where the price of the electricity is dynamic. We shown how to achieve savings in such environment and developed a scheduling algorithm that accomplishes that.

Bibliography

[1] Arm big.little technology. http://www.arm.com/products/processors/technologies/biglittleprocessing.php. Accessed: 2015-2-27.

[2] Arm cortex a9 website. http://www.arm.com/cortex-a9.php. Accessed: 2015-2-27.

[3] Arm holding website. http://www.arm.com/. Accessed: 2015-2-27.

[4] Armv8 official website. http://www.arm.com/products/processors/armv8-architecture.php. Accessed: 2015-2-27.

[5] Can amd, arm, or ibm dent intel's server dominance? http://uk.pcmag.com/opinion/32218/can-amd-arm-or-ibm-dent-intels-server-dominance. Accessed: 2015-2-27.

[6] Cern website. http://home.web.cern.ch/about. Accessed: 2015-2-27.

[7] *CMSSW Framework Workbook*. https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBoo

[8] Cortexa9 specs. http://www.arm.com/products/processors/cortex-a/cortex-a9.php. Accessed: 2015-10-27.

[9] Hyper threading technology. http://www.intel.com/content/www/us/en/architecture-and-technology/hyper-threading/hyper-threading-technology.html. Accessed: 2015-2-27.

[10] Hyper threading technology. http://wiki.bostonlabs.co.uk/w/index.php?title=Calxeda:Get_node_power_using_ipmitool. Accessed: 2015-2-27.

[11] *Intel 64 and IA-32 Software Developer Manuals - Volume 3*. www.intel.com/content/www/us/en/processors/ architectures-software-developer-manuals.html.

[12] Intel atom processor d525 specs. http://ark.intel.com/products/49490/Intel-Atom-processor-D525-1M-Cache-1. Accessed: 2015-10-27.

[13] Intel atom processor presentation page. http://www.intel.com/content/www/us/en/processors/atom/atom-processor.html. Accessed: 2015-10-27.

[14] Intel core2 quad processor q9400 specs. http://ark.intel.com/products/35365/Intel-Core2-Quad-Processor-Q9400-6M-Cache-2. Accessed: 2015-10-27.

[15] Intel xeon processor. http://www.intel.com/content/www/us/en/processors/xeon/xeon-processor-e7-family.html. Accessed: 2015-10-27.

[16] Intel xeon processor e5-2650. http://ark.intel.com/products/64590/Intel-Xeon-Processor-E5-2650-20M-Cache-2_00-GHz-8_00-GTs-Intel-QPI. Accessed: 2015-10-27.

[17] Ipmi tools. http://www.boston.co.uk/technical/2012/03/supermicro-ipmi-what-can-it-do-for-you.aspx. Accessed: 2015-10-27.

[18] Lhc website. http://home.cern/topics/large-hadron-collider. Accessed: 2015-2-27.

[19] Odroid-xu3 specs. http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127&tab_idx=1. Accessed: 2015-10-27.

[20] Odroid xu3 website. http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127. Accessed: 2015-2-27.

[21] Stack overflow. http://stackoverflow.com/. Accessed: 2014-10-27.

[22] Texas Instruments (TI) INA 231. http://www.ti.com/product/ina231. Last time accessed: 17-09-2014.

[23] Viridis boston website. http://www.boston.co.uk/solutions/viridis/introducing-the-viridis-2.aspx. Accessed: 2015-2-27.

[24] Wlcg website. http://wlcg.web.cern.ch/. Accessed: 2015-2-27.

[25] Reducing data center energy consumption.

[26] Aad, G., et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys.Lett. B716* (2012), 1–29.

[27] Abdurachmanov, D., Bockelman, B., Elmer, P., Eulisse, G., Knight, R., and Muzaffar, S. Heterogeneous high throughput scientific computing with APM x-gene and intel xeon phi. *CoRR abs/1410.3441* (2014).

[28] Abdurachmanov, D., Elmer, P., Eulisse, G., Grosso, P., Hillegas, C., Holzman, B., Klous, S., Knight, R., and Muzaffar, S. Power-aware Applications for Scientific Cluster and Distributed. *PoS ISGC2014* (2014), 018.

[29] Abdurachmanov, D., Elmer, P., Eulisse, G., and Knight, R. Future computing platforms for science in a power constrained era. *CoRR abs/1510.03676* (2015).

[30] Abdurachmanov, D., Elmer, P., Eulisse, G., Knight, R., Niemi, T., Nurminen, J. K., Nyback, F., Pestana, G., Ou, Z., and Khan, K. N. Techniques and tools for measuring energy efficiency of scientific software applications. *CoRR abs/1410.3440* (2014).

[31] Abdurachmanov, D., Elmer, P., Eulisse, G., and Muzaffar, S. Initial explorations of ARM processors for scientific computing. *J. Phys. Conf. Ser. 523* (2014), 012009.

[32] Abdurachmanov, D., et al. Explorations of the viability of ARM and Xeon Phi for physics processing. *J. Phys. Conf. Ser. 513* (2014), 052008.

[33] Blem, E., Menon, J., and Sankaralingam, K. Power struggles: Revisiting the risc vs. cisc debate on contemporary arm and x86 architectures. In *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on* (Feb 2013), pp. 1–12.

[34] Buchbinder, N., Jain, N., and Menache, I. Online job-migration for reducing the electricity bill in the cloud. In *NETWORKING 2011*. Springer Berlin Heidelberg, 2011, pp. 172–185.

[35] Chang, N., Kim, K., and Lee, H. G. Cycle-accurate energy measurement and characterization with a case study of the arm7tdmi [microprocessors]. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 10*, 2 (April 2002), 146–154.

[36] Chatrchyan, S., et al. Observation of a new boson at a mass of 125 gev with the {CMS} experiment at the {LHC}. *Physics Letters B 716*, 1 (2012), 30 – 61.

[37] Dayarathna, M., Wen, Y., and Fan, R. Data center energy consumption modeling : A survey. *Communications Surveys Tutorials, IEEE PP*, 99 (2015), 1–1.

[38] Drouhin, F., Gross, L., Vintache, D., Marchioro, A., Paillard, C., Ljuslin, C., Mirabito, L., and Verdini, P. The cern cms tracker control system. In *Nuclear Science Symposium Conference Record, 2004 IEEE* (Oct 2004), vol. 2, pp. 1196–1200 Vol. 2.

[39] et al, S. A. Geant4 a simulation toolkit. In *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, pp. 250–303.

[40] Gorlatova, M., Wallwater, A., and Zussman, G. Networking low-power energy harvesting devices: Measurements and algorithms. *Mobile Computing, IEEE Transactions on 12*, 9 (Sept 2013), 1853–1865.

[41] Hergenroder, A., and Furthmuller, J. On energy measurement methods in wireless networks. In *Communications (ICC), 2012 IEEE International Conference on* (June 2012), pp. 6268–6272.

[42] Isen, C., John, L., and John, E. A tale of two processors: Revisiting the risc-cisc debate. In *In 2009 SPEC Benchmark Workshop.*

[43] Khan, K., Nyback, F., Ou, Z., Nurminen, J., Niemi, T., Eulisse, G., Elmer, P., and Abdurachmanov, D. Energy profiling using igprof. In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on* (May 2015), pp. 1115–1118.

[44] Kim, S., Kim, H.-E., Kim, H., and Lee, J.-A. Computing energy-efficiency in the mobile gpu. In *SoC Design Conference (ISOCC), 2013 International* (Nov 2013), pp. 219–221.

[45] Liu, Z., Lin, M., Wierman, A., Low, S. H., and Andrew, L. L. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2011), SIGMETRICS '11, ACM, pp. 233–244.

[46] Lozano, H., and Ito, M. A reduced complexity instruction set architecture for low cost embedded processors. In *High Performance Computing Simulation (HPCS), 2015 International Conference on* (July 2015), pp. 400–407.

[47] Ma, K., Li, X., Chen, W., Zhang, C., and Wang, X. Greengpu: A holistic approach to energy efficiency in gpu-cpu heterogeneous architectures. In *Parallel Processing (ICPP), 2012 41st International Conference on* (Sept 2012), pp. 48–57.

[48] Mei, J., and Li, K. Energy-aware scheduling algorithm with duplication on heterogeneous computing systems. In *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing* (Washington, DC, USA, 2012), GRID '12, IEEE Computer Society, pp. 122–129.

[49] Ou, Z., Pang, B., Deng, Y., Nurminen, J., Yla-Jaaski, A., and Hui, P. Energy- and cost-efficiency analysis of arm-based clusters. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on* (May 2012), pp. 115–123.

[50] Ou, Z., Pang, B., Deng, Y., Nurminen, J., Ylaa-Jaaski, A., and Hui, P. Energy- and cost-efficiency analysis of arm-based clusters. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on* (May 2012), pp. 115–123.

[51] Padoin, E., de Oliveira, D., Velho, P., and Navaux, P. Evaluating performance and energy on arm-based clusters for high performance computing. In *Parallel Processing Workshops (ICPPW), 2012 41st International Conference on* (Sept 2012), pp. 165–172.

[52] Patterson, D. A., and Ditzel, D. R. The case for the reduced instruction set computer. *SIGARCH Comput. Archit. News 8*, 6 (Oct. 1980), 25–33.

[53] Pinto, G., Castor, F., and Liu, Y. D. Mining questions about software energy consumption. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (New York, NY, USA, 2014), MSR 2014, ACM, pp. 22–31.

[54] Qureshi, A., Weber, R., Balakrishnan, H., Guttag, J., and Maggs, B. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review 39*, 4 (2009), 123–134.

[55] Rao, L., Liu, X., Xie, L., and Liu, W. Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment. In *INFOCOM, 2010 Proceedings IEEE* (2010), IEEE, pp. 1–9.

[56] Ren, S., He, Y., and Xu, F. Provably-efficient job scheduling for energy and fairness in geographically distributed data centers. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on* (2012), IEEE, pp. 22–31.

[57] Sharifi, L., Rameshan, N., Freitag, F., and Veiga, L. Energy efficiency dilemma: P2p-cloud vs. datacenter. In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on* (Dec 2014), pp. 611–619.

[58] Subramaniam, B., and chun Feng, W. Towards energy-proportional computing for enterprise-class server workloads. *CPE '13 Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering* (2012).

[59] ten Kate, H. Superconducting magnet system for the atlas detector at cern. *Applied Superconductivity, IEEE Transactions on 9*, 2 (June 1999), 841–846.

[60] Wang, Y., Li, K., Chen, H., He, L., and Li, K. Energy-aware data allocation and task scheduling on heterogeneous multiprocessor systems with time constraints. *Emerging Topics in Computing, IEEE Transactions on 2*, 2 (June 2014), 134–148.

[61] Yang, C.-L., Hwang, B.-N., and Yuan, B. Key consideration factors of adopting cloud computing for science. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on* (Dec 2012), pp. 597–600.

[62] Yang, X., Zhou, Z., Wallace, S., Lan, Z., Tang, W., Coghlan, S., and Papka, M. Integrating dynamic pricing of electricity into energy aware scheduling for hpc systems. In *High Performance Computing, Networking, Storage and Analysis (SC), 2013 International Conference for* (Nov 2013), pp. 1–11.

[63] Zeng, G., Yu, L., and Ding, C. An executing method for time and energy optimization in heterogeneous computing. In *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on* (Aug 2011), pp. 69–74.

[64] Zhu, J., Qiao, C., and Wang, X. On accurate energy consumption models for wireless ad hoc networks. *Wireless Communications, IEEE Transactions on 5*, 11 (November 2006), 3077–3086.

Appendix A

Techniques and tools for measuring energy efficiency of scientific software applications

The article in appendix is called *Techniques and tools for measuring energy efficiency of scientific software applications* and it is the result of the research about tools and techniques for energy consumption done during this study. The article was published in the conference proceedings of the 16th International workshop on Advanced Computing and Analysis Techniques in physics research (ACAT'2014).

PDF append article