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! — November 21, 2015 — DRAFT!

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

Aalto University
School of Science

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:**	64
Major:	Data Communication Software	**Code:**	T-110
Supervisors:	Professor Jukka K. Nurminen		
Advisor:	Zhonghong Ou (Post-Doc.)		
abstract			
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 chipset and the overall technology development 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 existent computing systems. This thesis focuses primarly on the later 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 take to improve the overall energy efficiency. Throughout this thesis, we will focus on measuring and improving the energy consumption in High Performance Computing (HTC) applied to Scientific Research.

The Large Handron 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 units that might be spread geographically.

The Large Hadron Collider (LHC) [ref] 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) [ref], which uses resources from 160 computer centers in 35 countries. Such computational resources have enabled the CMS [ref] and ATLAS [ref] experiments to discover the Higgs Boson [ref, ref], amongst other scientific achievements. The WLHC requires a massive amount of computational resources (250,000 x86 cores in 2012) and, proportionally, energy. In the future, with planned increases to the LHC luminosity [ref], 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 Problem Statement

A considerable amount of research has been done on leveraging Reduced Instruction Set Computing (RISC) architectures to minimize energy consumption on mobile and energy constrained devices. In the mobile and energy constrained devices case, energy consumption is a priority given the inherent reduced amount of energy available that affects the mobility of the devices.

The large quota 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 Troughtput 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 author 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 are a good match to HTC computing or not. There are open points regarding whether the performance constraints of RISC architectures and the high performance requirements of HTC workload are acceptable. In addition, it is still unclear if RISC architectures are more energy efficient under HTC workloads than the conventional Complex Instruction Set Computing (CISC) architectures.

Therefore, it is of our interest to study the potential impact of RISC architectures in the HTC and scientific computing industry. In our opinion, there are two major gaps that need to be fulfilled: Firstly, a comparisons between RISC and CISC architectures under authentic scientific workloads. Secondly, there is scarce solutions that leverage RISC architectures on HTC and scientific computing.

1.3 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 collider.

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 chipsets 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.

1.4 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 practives 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). The article was called *Techniques and tools for measuring energy efficiency of scientific software applications*

- 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 analysed the results (Chapter 5).

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

1.5 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 fouth 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, we present a scheduling algorithm that aims at lower the energy bill of HTC by using both ARM and Intel architectures, based on the results obtained in our experiments. Finally, we wrap up by outlining possible future work and presenting the conclusions of this thesis.

Chapter 2

Background

2.1 Energy consumption awareness in Scientific Computing

- current state of HPC systems and energy efficiency (p10)

Power consumption and energy efficiency have become a paramount research topic in computer science, for energy has become a major growth bottleneck in several systems. Moreover, the increasing concerns with energy consumption and its social, economical and environmental impact in our society has given a bigger dimension to the discussion. Given its importance, many research studies have looked into energy efficiency and power consumption. There is a vast panoply of studies with different approaches toward improving energy efficiency. For example, to use GPU for data processing [23], [25] or improve energy efficiency by using RISC architectures [27] [19], [17], . Several other directions have been taken toward energy efficient computing [28] [24], [32] and the number of related works keeps growing.

Scientific computing is often characterized by requiring enormous data storage capacity, high processing capabilities and complex configuration [34]. 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.

2.2 The European Organization for Nuclear Research and the LHC

The European Research for Nuclear Research (CERN) [3] is a particle physics research laboratory sited in the Franco-Swiss border where thousands of en-

gineers and physicists from about 21 state members conduct researches about the fundamental structures of the Universe. In order to perform experiments that support the researcherś studies, they have 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 in the CERNś datacenters for posterior reconstruction and processing. 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 resources are critical to manage efficiently.

According to [19], the computing requirements for HPC have increased particularly in recent years. Most notably, a project with the magnitude and complexity of the LHC is a sound example of it. To achieve results like the discovery of the Higgs boson [16] [21] 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 super-node. 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 WLCG [15] is a grid computing platform where more than 170 computing centers spread across 40 countries [15] collaborate to store and process the data coming from the LHC experiment. According to [15], the WLCG alone is responsible for the distribution, storage and processing of more than 30 Petabytes annually. According to [19], the equivalent capacity of WLCG in 2012 was between 80,000 and 100,000 x86-64 cores. In the future, other projects and researches will demand even more processing capacity from the WLCG. For example, as stated by [19], 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.

2.3 Tools and techniques for measuring energy consumption

2.3.1 Importance of measuring energy consumption

The study conducted by [28], shows that engineers have been considering energy consumption as an important factor when developing software. It consists on an empirical study that aims to understand the opinions and problems of software developers about energy efficiency. The data that sustain the conclusions are mined from a well-known technical forum (*StackOverflow* [13]). 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.

2.4 ARM architecture

2.4.1 ARM architecture in scientific computation

- more on (p7)

2.4.2 Literature review

In [19]:

- After 2015, processors have hit scaling limits. Two different paths started to be taken on the processor industry: development of multiprocessor architectures that allow to run parallel tasks and the time clock frequency - which have been increasing throughout the years - stabilized.

- Most High Physics Computing systems run in clusters of several cores. Additional cores are parallelized and can run at the same time, which allows the system to scale. However, also commodities such memory, I/O streams and energy scale proportionally in such architectures.

In [17], 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 HPC 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 an ARMv8 64-bits architecture.

In [17], they use 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. As the authors highlight, the firmware for managing processor ACPI power states was not yet available when the study was made. Thus, it is expected that the energy performance will improve once the firmware is available [17].

Under the circumstances of the experiment, the overall results show that APM X-Gene is 2.73 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 performances despite the absence of platform specific optimizations. Therefore, [17] concludes by stating that the APM X-Gene 1 Server-On-Chip ARMv8 64-bit solution is relevant and potentially interesting platform for heterogeneous high-density computing.

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, the more performant machines can be used without huge degradation of the overal price budget.

There are several studies exploring inter data center solutions to lower the electricity bill by leveraging the spacial-time dynamic 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 are 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 [31], [20], [30], [29], [26], amongst others.

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, one work that seems closer to our solution is [35]. In this study, the authors achieve better energy performance in a dynamic pricing environment with HPC systems by judiciously scheduling parallel jobs - which have different energy profiles - depending on the energy pricing of the moment. The main difference to out solution is that the performance of the machines are not taken into consideration when scheduling the jobs, but rather the job energy profiling.

Another research study that related to our solution is [33]. They came up with an optimal algorithm and two heuristic algorithms to schedule tasks to heterogeneous processors. In addition, they also take into consideration the memory allocation in heterogeneous memory in order to minimize energy consumption while meeting the assumed deadlines. Their work, though, seems to go further than our solution since it considers heterogeneous memory allocation as well. They consider is 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, but it seems that our approach is been used already.

A similar idea has been explored in [36]. 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 [31], 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 from the performance and energy efficiency. In addition, the data centers may be disposed in different geographical locations and, thus, have different energy tariffs. The authors of [31], 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.

In [31], the authors present an online 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 developed by Lyapunov ['Resource allocation and cross-layer control in wireless networks'] 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 tradeoff between energy pricing, fairness and queueing delay.

In [31], 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.

The [20] study aims to exploit 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 (as our study is aiming for)

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 tradeoff 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.

In [30], the researchers 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.

Again, the time constraints and delays considered in a inter data center study does not need to be considered in out work.

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 [20]. Again, it is part of the set of studies on inter datacenter and electrical costs optimizations that location and time based pricing allows.

The authors of [29] 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.

In [35], the authors 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 take an intra data center approach, since it considers a solution that can be put into practice at a data center level.

The authors claim that "A key challenge in HPC scheduling is that system utilization should not be impacted. HPC systems require a tremendous capital investment, hence taking full advantage of this expensive resources is of great importance to HPC 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 [35], 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.

The authors of [26] present a novel task scheduling algorithm for HPC systems which considers two main points: reducing the energy consuption of the overall system and minimize the schedule length. An HP system is defined by the authors as set of distributed computing machines with differrent configurations connected through a high speed link to compute paralell applications.

They assume a that all the information needed to schedule the taksk 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 online 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 redudantly, 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

HPC systems at a distributed level and do not focus, as our approach, on inter data center solutions.

In [33], the authors address the problem of an energy aware scheduling for heteregenous data allocation and task scheduling. The problem consists in finding the best taks scheduling in a heterogeneous system that meet the deadlines while minimizing the energy consumption.

The processors and memories come in different flavors nowadays in HPC systems, making complex the task of efficiently scheadule 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 heterogenous computing and, most notably for our research, [**?**].

They present an optimal algorithm and two heuristical 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.

The authors of [36] claim that, unfortunatelly, there are not many studies of processor scheduling algorithms that take into consideration both time and energy. In this study, they explore heuristical scheduling algorithms focused on high performance computing and green computing. They work on heuristical 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 whith 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 specificcally 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). The article was called *Techniques and tools for measuring energy efficiency of scientific software applications* [18] and can be found on Appendix A, at the end of this document.

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 sizeable fraction of the consumed power. They are also the place where the largest gains in terms of efficiency can be obtained through optimizations in the software [18].

Coarse grain or external measurements

As stated by [18], 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, plugin 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 datacenters or to provide a baseline for more detailed studies.

Fine grain or internal measurements

A alternative approach takes into account the internal structure of a computing element of an HTC system, as shown in figure 3.1. Nowadays, 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. An example of this chip monitors is the Texas Instruments TI INA231 [**?**] 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.e Compared to external methods, these on-board components provide high accuracy and reasonably high precision measurements (millisecond level).

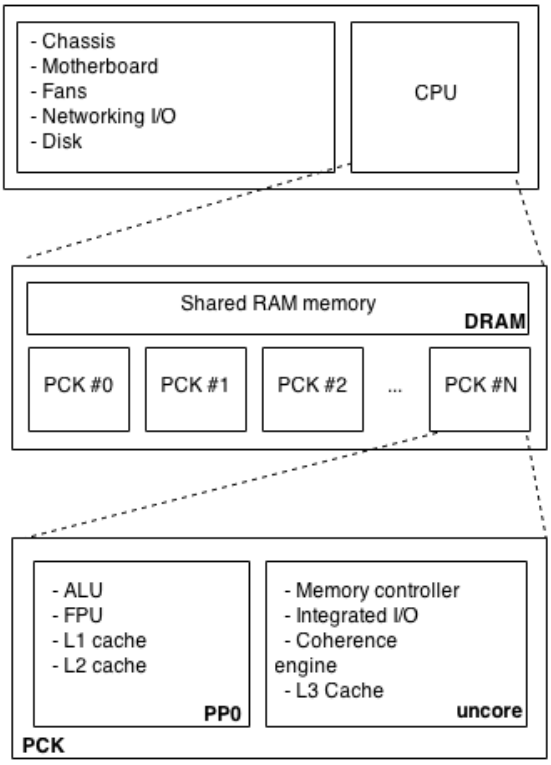

Figure 3.1: Components that contribute for power consumption in HPC. Taken from [18]

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 SoC 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 the CPU core(s), and **dram**, which accounts for the sum of energy consumed by memory in a given socket, therefore excluding the on-core caches [**?**]. As for the discrete components case, the timing resolution of measurements is in the millisecond range [**?**]. This is fine enough to allow exploiting such data to build an energy consumption sampling profiler for applications, similar to how performance sampling profilers work (see section **??**). 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 define the average power consumption limit of a domain in a defined time window and allows more accurate independent measurements of the non limited 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 [**?**]. 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 capanilities.

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.

Most of the techiques and tools outlined in this section were used in our experiments and are further discussed in the Experiments and Analysis chapters. In addition, the original paper which dwelves into the techniques and tools in more detail and was published at the ACAT'2014 conference proceding is attached as Appendix A.

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 the results to be as realistic as possible, we used the CMSSW framework [ref] and ParFullCMS [ref] simulations, 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 resonable 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.

Thoughout 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 en-

ergy measurement tools. During this chapter and throughout the rest of the thesis, we will describe each batch of experiments as first set of experiments (1SE) and second set of experiments (2SE).

The remainder of this chapter divided in two section. Firstly, we will outline the most relevant characteristics of the architectures used during the experiments. Secondly, we describe the setup of the experiments and methodology used to perform the experiments.

4.1 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 [12] and a server class ARM processor by Boston Viridis [14]. The Intel machines used were part of the microarchitectures family Sandy Bridges and Intel Bonnell. In the following sections, we will describe the hardware architecture, features of the hardware used to run the experiments and where the hardware is commonly used outside the scope of this study.

4.1.1 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 [14].

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 4.1. A node is an ARM based CPU fabricated by Calxeda. The block diagram of a ARM_viridis node is represented on 4.2shows the architectures of the EnergyCore used. We can notice that the SoC has an energy management engine that will further on allow us to sample the energy consumed by the node.

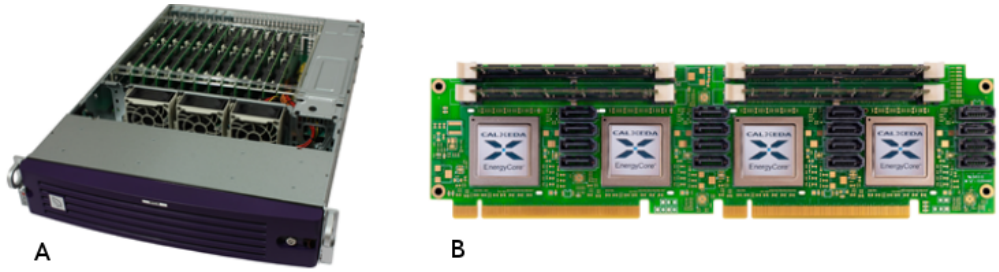

Figure 4.1: A. Viridis Server chassis with 12 energy card in it. B. Energy card with 4 nodes. Taken from [14]

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 includede 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.

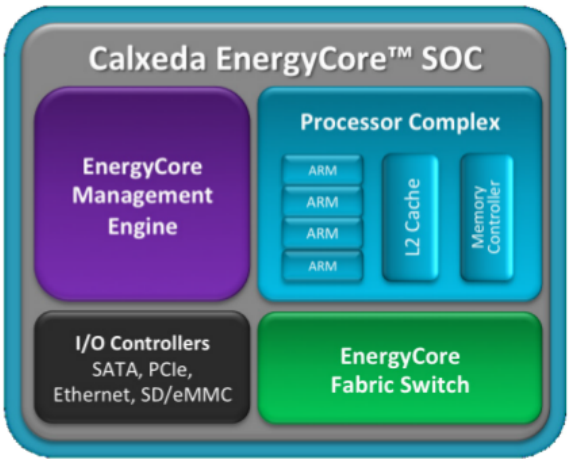

Figure 4.2: Block diagram of Calxeda EnergyCore. Taken from [14]

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

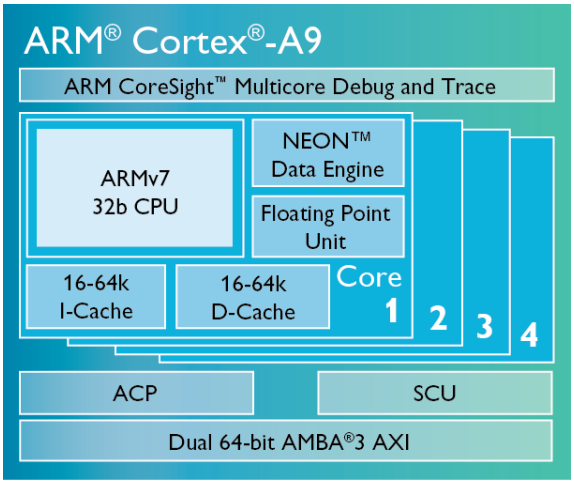

Figure 4.3: Block diagram of Cortex A9. Taken from [2]

ODROID-XU3 development board

The ODROID-XU3 [12] 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" [12]. 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.

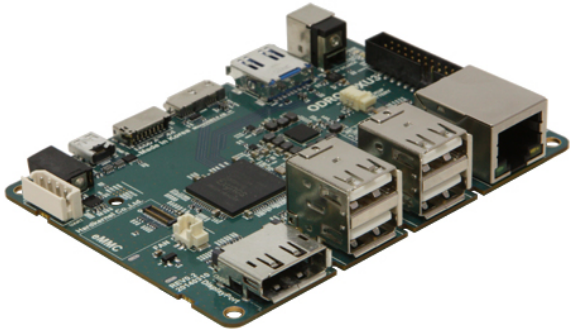

Figure 4.4: ODROID-XU3 development board. Taken from [12]

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 4.5.

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.

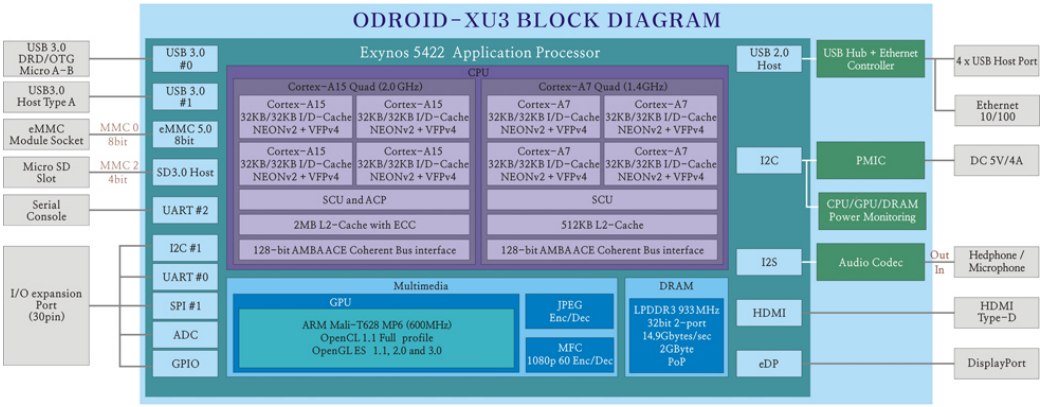

Figure 4.5: ODROID-XU3 block diagram. Taken from [12]

4.1.2 Intel x86 architecture

Across the different experiments, we have used three different machines running on top of x86 Intel instruction sets to compare with the ARM based machines. The Intel x86 machines are the most widely used solutions for server and workstation applications.

The Intel Xeon that we used had RAPL enable (refer to Chapter X), which allowed us to measure energy consumption accurately at a fine-grained level. Since the ATOM and 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.

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

describe more about Intel, its features (hyper threading, wich affects the results for example), its microarch families and where/how

have them been used in production

ex: "The Intel Atom is a brand name for a line of ultra-low-voltage CPUs by Intel. On the other hand, the x86 Intem Quad is brand name for a high performance family of Intel CPUs"

4.2 Experiments setup

4.2.1 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 general purpose. The ARM_virdis is a server rack with CPU consisting of ARMv7 processors produced by Boston Labs [ref]. We ran the same workloads in a x86 Intel Atom (Intel_atom) and Intel Quad (Intel_quad) for comparison. The Intel Atom is a brand name for a line of ultra-low-voltage CPUs by Intel. On the other hand, the x86 Intem Quad is brand name for a high performance family of Intel CPUs.

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™ Atom™ CPU D525 1.8GHz

Memory (MemTotal): 3925084 kB (4GB)

For more detailed specs refer to [7]

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™ Core™2 Quad CPU Q9400 2.66GHz

Memory (MemTotal): 7928892 kB (8GB)

For more detailed specs refer to [8]

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 [4]

Software and workload

We used the CMSSW framework in the generation-simulation mode (GEN-SIM). The workflow performs a Monte Carlo simulation of 8 TeV LHC Minimum bias event using Pynthia8 (generation step), followed by Simulation with Geant4 (simulation step). For more information about the CMSSW framework and its limitation on ARM, refer to Chapter X. At the time of the experiments, the CMSSW port for ARM had limitations on the multithreading support. We wanted to study the energy consumption of each hardware setup given different core load. Thus, we spinned 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 fd 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 4.6). 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].

Figure 4.6: 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™ CortexA9™	ARMv7 32b (A7)	4	2 GB	Server class ARM processor with ipmitools
Intel_ATOM	Intel Bonnell™	Atom D525	4	4GB	No internal measurement tool
Intel_QUAD	Intel Sandy Bridge™	Quad CPU Q9400	4	8GB	No internal measurement tool

Table 4.1: Summary of the 1SE specifications

4.2.2 Second set of experiments

Hardware specifications

For the second set of experiements, we again used three machines with different hardware setups. As in the 1SE, the three machines differ in architecture and general purpose. The ARM_virdis, which was used in the 1SE, was also used during the second set of experiments. In addition to ARM_viridis, we also resort to another machine powered by an ARM CPU. The ARM_odroid is a development board manufactured by HardKernel [ref] and it is intended to provide a cheap and easy way to develop hardware and software in a ARM architecture. To represent the Intel architecture 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 performace 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 [4]

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 [11]

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^{TM} CPU E5-2650 2GHz

Memory: 252GB

For more detailed specs refer to [9]

Software and workload

We used the CMSSW's mode ParCullCMS for generating the workload. The ParFullCMS mode is a multi-threaded Geant4 [22] benchmark. It uses a complex CMS geometry for the event simulation and has the advantage of being multithreaded in both Intel and ARM architectures. As in the first set of experiements, 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 1SE, the energy efficiency metric used in this study is the ratio of performance per power consumed (Watts). The hardware setups used in the 2SE 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 2SE, 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 performe 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 embeeded 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

Machine codename	Architecture	CPU	Nº active cores	RAM	Notes
ARM_odroid	Quad-Core ARMv7™	A15 and or A7 cores(big.LITTLE technology)	4	2 GB	Development board with TI INA231 chip
ARM_viridis	Quad-Core ARM™ CortexA9™	ARMv7 32b (A7)	4	2 GB	Server class ARM processor with ipmitools
Intel_xeon	Intel Sandy Bridge™	CPU E5-2650	32	252 GB	System on a rack with RAPL

Table 4.2: Summary of the 2-SE specifications

microseconds. For the extrenal 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) technolgy 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.

For the ARM_viridis, we used the capabilities of the Intellegent Platform Management Interface (IPMI) [10] 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 interlal hardware energy monitoring. We leveraged the IPMI tools to perform internal energy consumption of the ARM cores during the experiments

4.3 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 results shown 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 analyse the different tools and techniques used to perform the experiments, as well as the results itself. The first section analyses the 1SE, whereas the second section is dedicated to analyse the 2SE. As stated before, the difference between 1SE and 2SE 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.

Finally, the figures 5.8 and **??** compare the time spent by each of the

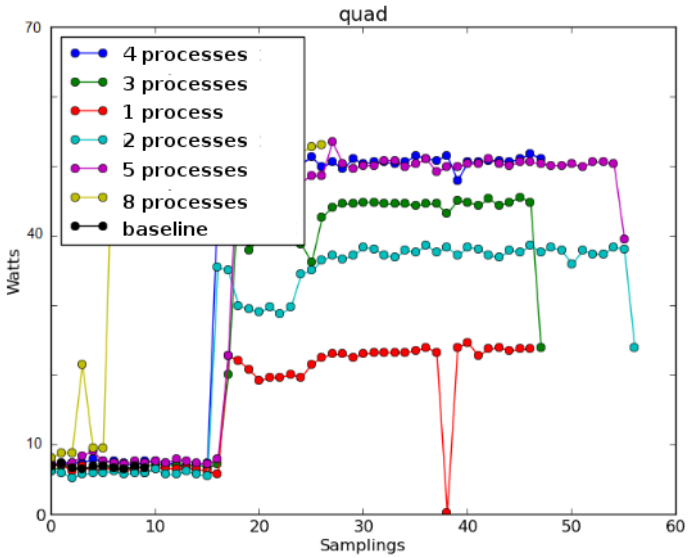

Figure 5.1: All stages of the CMSSW experiments on Intel_quad

hardware setups to process the workflow and the power consumption efficiency of the different setups, respectively.

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 od the scope of this work to analyse this stage in depth.

The second stage is the connection phase. The goal of this stage is to fetch the metadata from the CERN servers that allow the event generation-simulation. The metadata 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.

1. Add memory plots? – easier to identify the 3 stages but out of scope

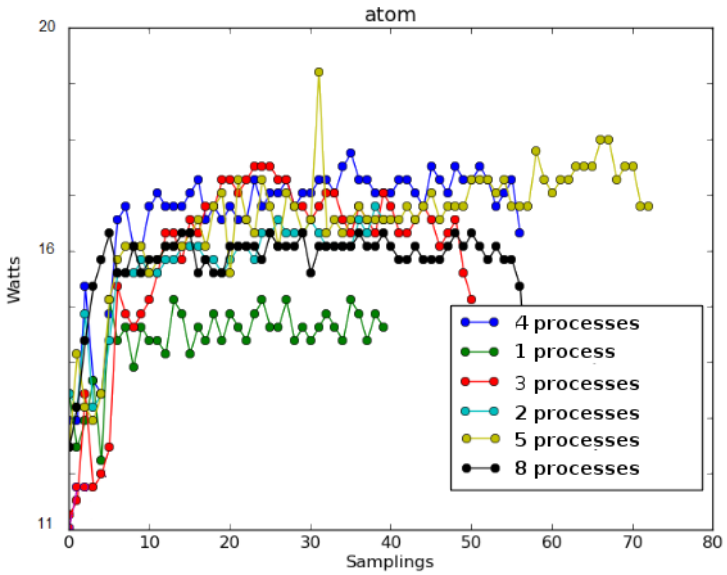

Figure 5.2: All stages of the CMSSW experiments on Intel_atom

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 [refs], 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 directly porpotional o 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

Overcommiting CPU and energy efficiency

We consider a CPU to be overcommited 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 overcommitted. When the number of processes exceed the number of available cores, the time to process the events increases since there are no available

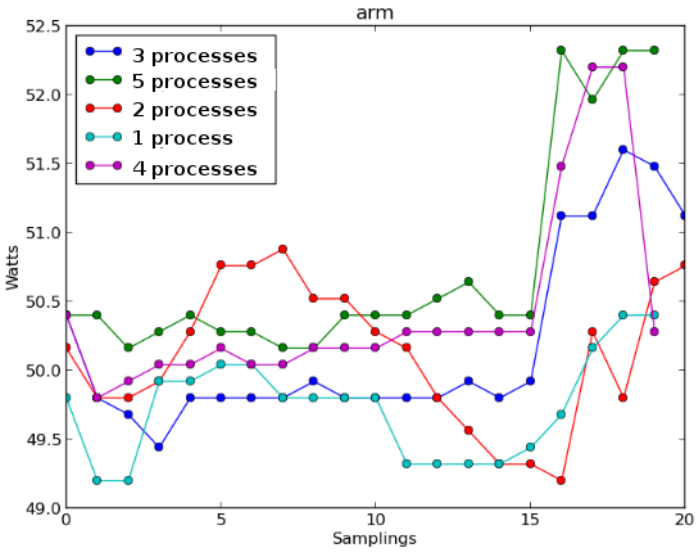

Figure 5.3: All stages of the CMSSW experiments on ARM_viridis

cores to process the events concurrently. In the overcommitted 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 overcommitted.

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 overcommited vs non overcommited, 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 and ARM_viridis and ARM_viridis is faster than Intel_atom. This fact is due to the architectures characteristics and its specifications, most notably the CPU clock speed.

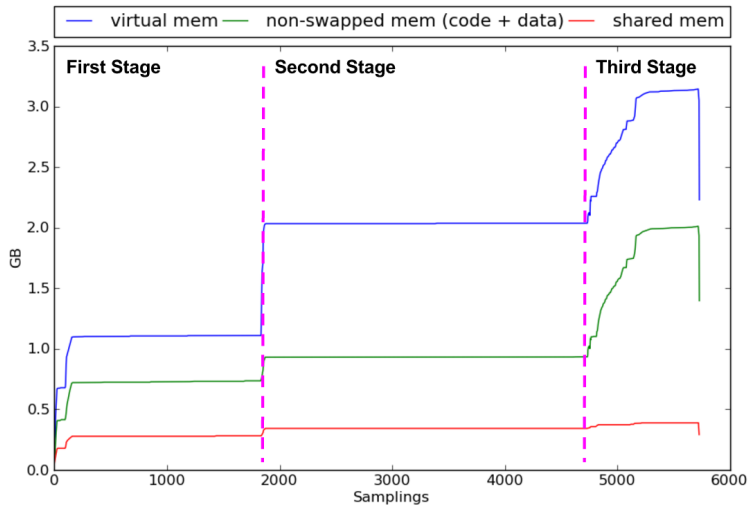

Figure 5.4: Memory usage during the 3 different stages

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. Clearly, the all the plot 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 mesaurement 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

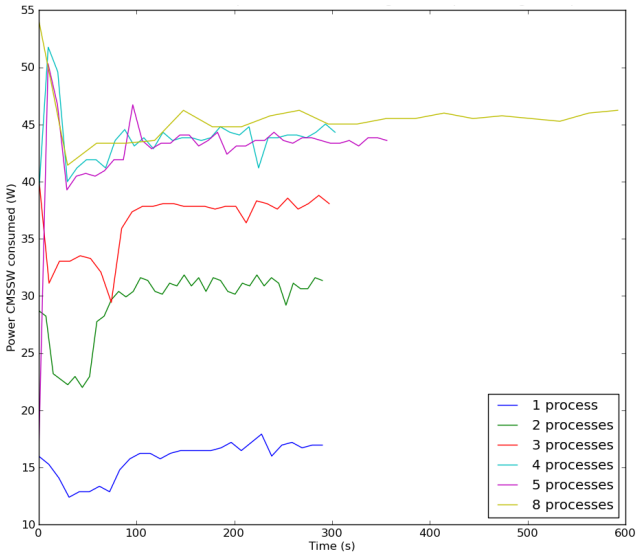

Figure 5.5: CMSSW experiments on Intel_quad - event processing stage

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 behaviours. 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.

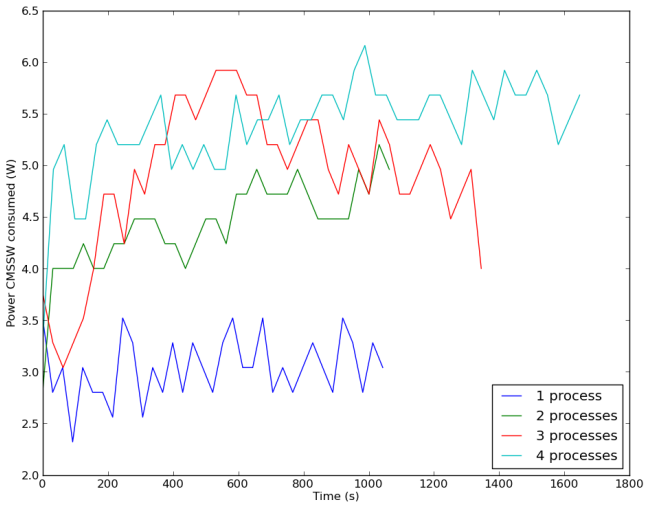

Figure 5.6: CMSSW experiments on Intel_atom - event processing stage

Energy performance and overcommited CPUs

It is noticeable that ARM_odroid has a significant energy performance decline when its cores are overcommited. 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 phenomenona is the large increase of time taken to process the events when the cores are overcommited. Thus, even if the cores are consuming the same Watts per second 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 overcommited. This phenomenon is explain 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 [5]. 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 overcommit Intel_xeon with more than 4 threads per core, the time to process the workload would increase, which

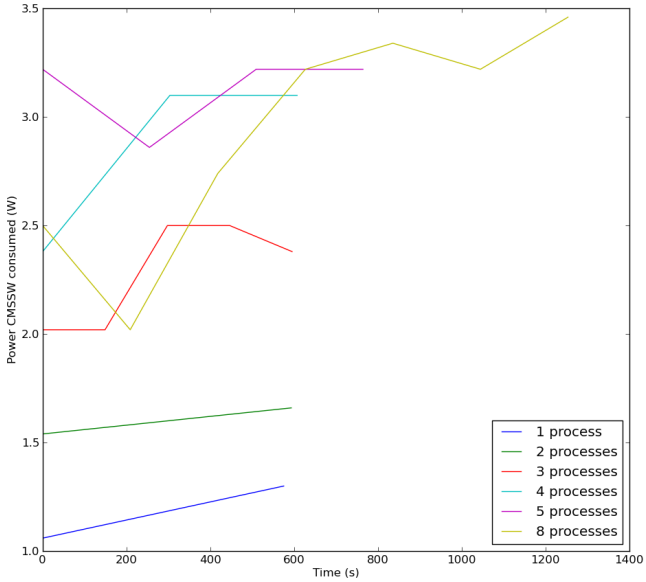

Figure 5.7: CMSSW experiments on ARM_viridis - event processing stage

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 [6], 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. The components that are measured by the IPMI tools at each energy sample are shown in the Figure **??**.

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,

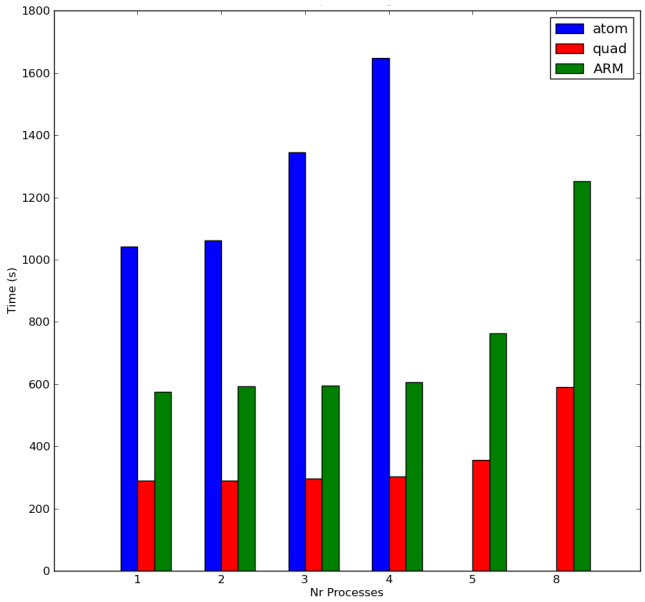

Figure 5.8: Processing time comparison

at least, perform better that Intel_xeon from an energy consumption 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 1SE (Figure 5.11) and 2SE (Figure 5.10) we may be inclined to conclude that the setups in the 2SE presented an overall more efficiency than the setups the 1SE. Again, the used measurement tools play an important role and should not be disregarded when analysing the results. In the 1SE, we only performed external measurements. Thus, we discard the possibility to compare the 1SE results with the results of the internal measurements of the 2SE. As for the external measurements performed in both set of experiments, the tools for measuring the energy consumption of both experiments have disctinct resolution and grain. In the 1SE, we used the clamp meter for measurements in all setups. As for the 2SE, we used embeeded and computer-assisted tools to perform the external measurements. This discrepancy of tools, its resolutions and errors, make it

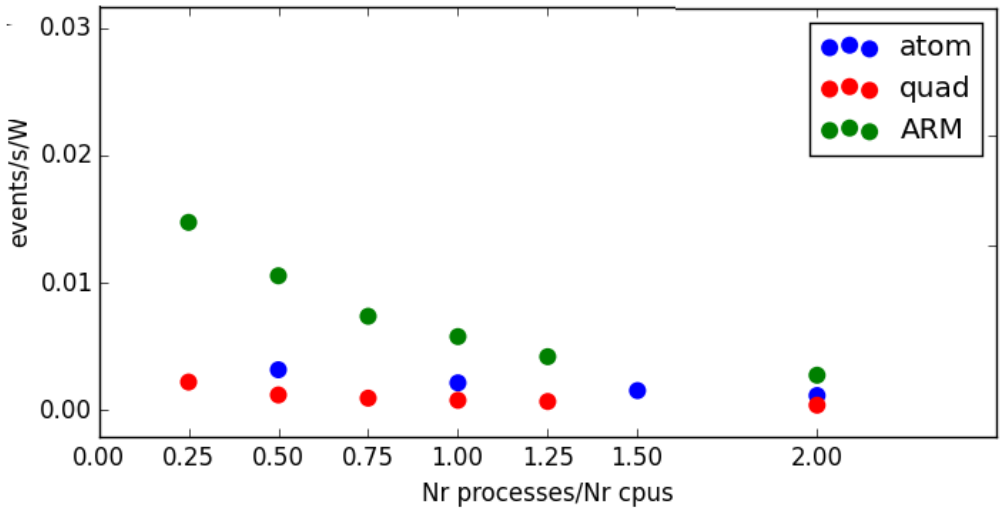

Figure 5.9: Energy efficiency comparison for the first set of experiments - External measurements

difficult to compare the results of the 1SE and 2SE.

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 chipset Intel in terms of energy 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 leanerd 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.

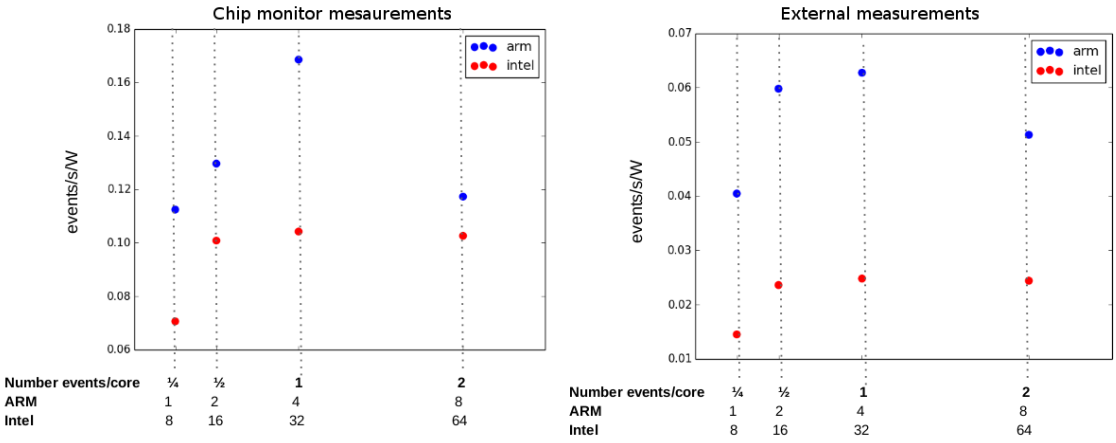

Figure 5.10: Multithreaded ParFullCMS comparison between Intel_xeon and ARM_odroid

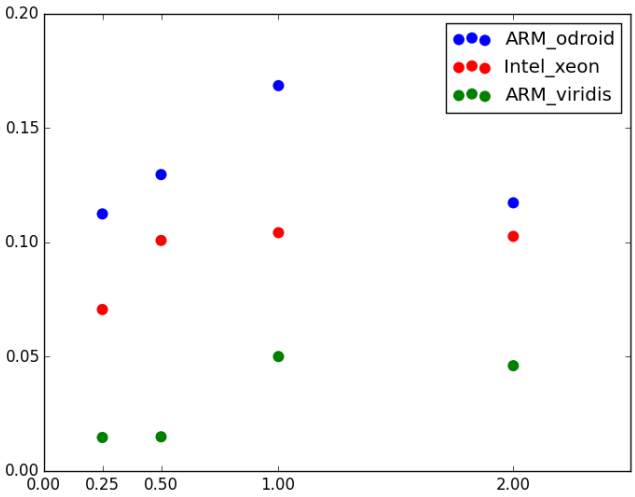

Figure 5.11: Multithreaded ParFullCMS comparison between Intel_xeon, ARM_viridis and ARM_odroid

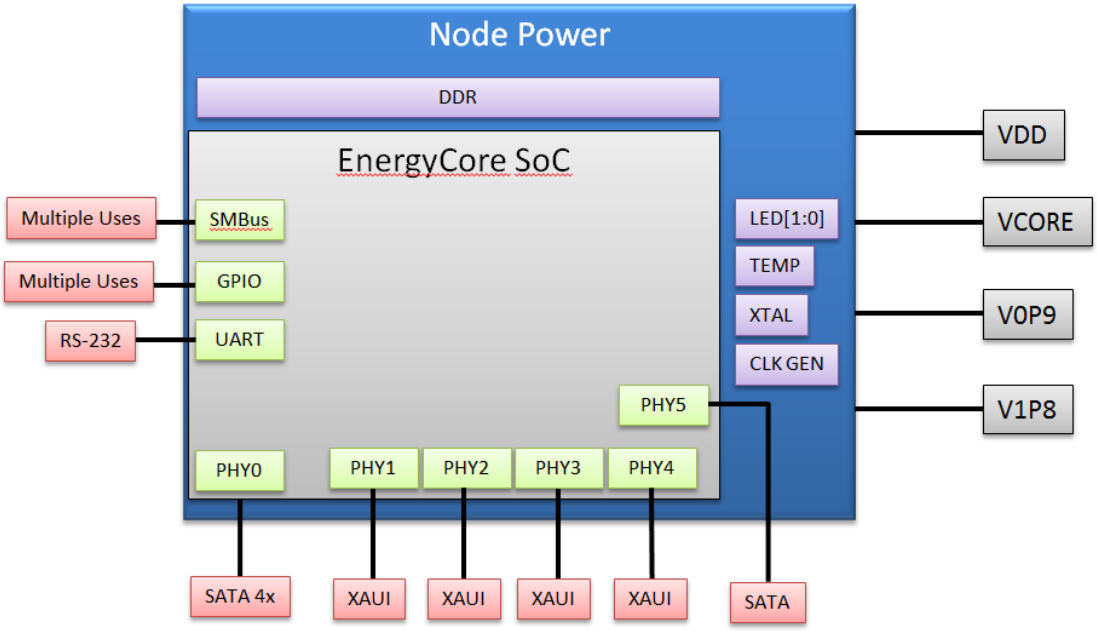

Figure 5.12: Representation of the Power Node measured by IPMI tools [make one myselfd and change!]

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 accorading 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 eletricity pricing model

Country	Period	Daily price range (MWh)
Germany	November 2014	[80€, 0€-20€]
Finland	-	-
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 eletricity 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 eletricity price along the day. For the sake of simplicity, during this study we define that the price of eletricity 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.

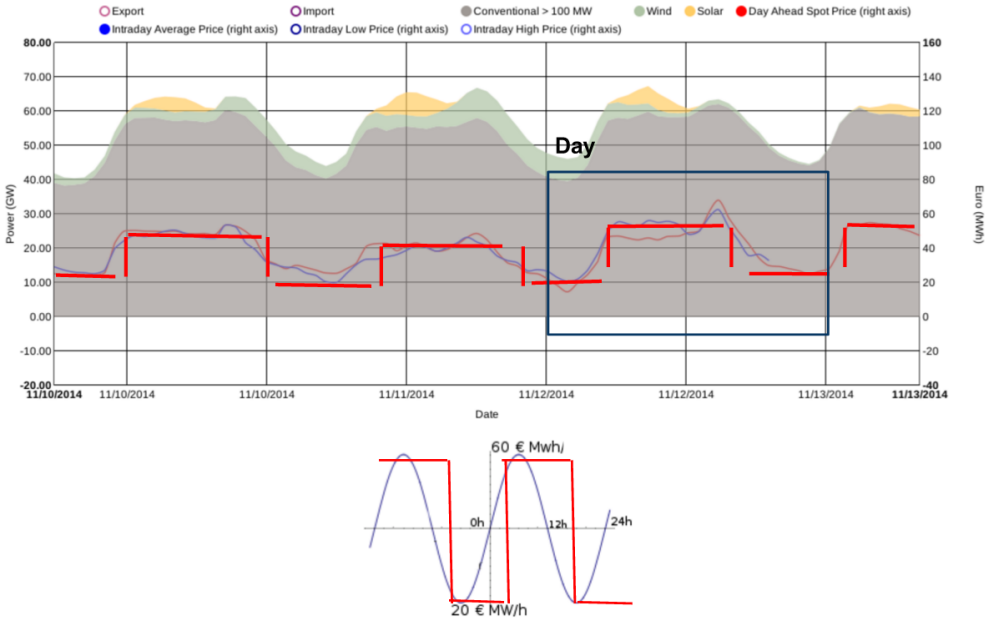

Figure 6.2: Simplified pricing model based on the Germany energy market

6.2 Scheduling algorithm

Problem formulation

The main goal of the algorithm is to lower the electric bill in dynamic eletricity 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.

```
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

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 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 diferent configurations.

Algorithm walkthrough

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 mroe 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 enregy consumed) is based on the values from the experiments analysed 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.

```
24  days = 10
25  nr_events = 1200000
26
27  bucket_1 = {
28      'price_ev': 8.8*10**-8,
29      'nr_ev_day': 40824/2,
30      'name': 'ARM_LOW'
31  }
32
33  bucket_2 = {
34      'price_ev': 2.6*10**-7,
35      'nr_ev_day': 40824/2,
36      'name': 'ARM_HIGH'
37  }
38
39  bucket_3 = {
40      'price_ev': 2.4*10**-6,
41      'nr_ev_day': 731289/2,
42      'name': 'INTEL_LOW'
43  }
44
45  bucket_4 = {
46      'price_ev': 7.2*10**-6,
47      'nr_ev_day': 731289/2,
48      'name': 'INTEL_HIGH'
49  }
50
51  result = scheduler([bucket_1, bucket_2, bucket_3, bucket_4], days, nr_events)
52  print result
```

Figure 6.6: Start the algorithm programatically

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 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

		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

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.

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 like [33] and [36] do not take the dynamics of electrical pric-

ing 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 [35] mentions, 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 [33], 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.

It would be interesting to develop further the scheaduling algorithm for dynamic eletricity 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 analyse has been increasing manyfold 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 wheter 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 reasearch 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 scheaduling algorithm that optimizes the eletricity bill of heterogeneous data centers working in a dynamic eletricity pricing markets.

Our main contribuitions are:

- We researched and outlined best practives 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 eletricity is dynamic. We shown how to achieve savings in such environment and developed a scheaduling 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] Cern website. http://home.web.cern.ch/about. Accessed: 2015-2-27.

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

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

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

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

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

[9] 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.

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

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

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

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

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

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

[16] 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.

[17] 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).

[18] 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).

[19] Abdurachmanov, D., Elmer, P., Eulisse, G., and Muzaffar, S. Initial explorations of arm processors for scientific computing. *Journal of Physics: Conference Series 523*, 1 (2014), 012009.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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