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, 1 December, 2014

DRAFT! — October 21, 2015 — DRAFT!

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

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

ABSTRACT OF
MASTER'S THESIS

Author:	Gonçalo Marques Pestana		
Title:	Energy Efficiency in High Throughput Computing Tools, techniques and experiments		
Date:	1 December, 2014	**Pages:**	58
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

I wish to thank all students who use LaTeX for formatting their theses, because theses formatted with LaTeX are just so nice.

Thank you, and keep up the good work!

Espoo, 1 December, 2014

Gonçalo Marques Pestana

Abbreviations and Acronyms

2k/4k/8k mode	COFDM operation modes
3GPP	3rd Generation Partnership Project
ESP	Encapsulating Security Payload; An IPsec security protocol
FLUTE	The File Delivery over Unidirectional Transport protocol
e.g.	for example (do not list here this kind of common acronymbs or abbreviations, but only those that are essential for understanding the content of your thesis.
note	Note also, that this list is not compulsory, and should be omitted if you have only few abbreviations

Contents

Chapter 1

Introduction

1.1 Overview

Nowadays, Moore's Law continues to increase the number of transistors per chipset and the overall technology development at a geometric rate. However, the energy consumption of the systems have begun to halt the usage of the technology at its full potential. It is well known that energy efficiency is an important research topic in computer science, for energy has become a major growth bottleneck for the 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 the systems. This approach aims to provide the needed energy for technology to reach its full potential. The second path is to improve the energy efficiency of the systems. This Thesis focuses on a specific area of the later approach.

The concerns with energy consumption and its impact in the current applications affect industries ranging from mobile devices to big data centers. Given the several layers and complexity of the systems nowadays, there are considerable number of directions to improve the energy efficiency of the systems. Throughout this Thesis, we will focus on improving the energy consumption in High Performance Computing (HPC) applied to Scientific Research.

1.1.1 The LHC example

In some applications, a single computing unit does not have enough resources to accomplish its 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 use by the CMS to study and to draw conclusions with respect to energy efficiency.

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 such cases, energy consumption is a priority given the inherent reduced amount of energy available.

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 HPC community has being considering energy efficiency as a priority in the foreseeable future. However, studies focusing on viability of RISC architectures on HPC as a way to minimize energy consumption are not abundant in the research technology. Furthermore, to the knowledge of the author, there are no major implementations of such technologies being used in HPC systems nor in scientific computing.

It is still unclear whether RISC architectures are a good match to HPC computing or not. There are open points regarding whether the performance constraints of RISC architectures and the high performance requirements of HPC workload are acceptable. In addition, it is still unclear if RISC architec-

tures are more energy efficient under HPC 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 HPC and scientific computing industry. In our opinion, there are two major lacks that need to be fulfilled: Firstly, there are lack of comparisons between RISC and CISC architectures under authentic scientific workloads. Secondly, there are scarce proposal for solutions using RISC in the HPC and scientific computing.

1.3 Scope of the Thesis

The purpose of this Thesis is to answer whether RISC architectures are a potential fit to HPC and scientific computing from a energy efficiency perspective. We focus mainly on comparing RISC - most notably ARM chipsets - and widely used CISC architectures such as Intel processors. For the endeavor, we use authentic HPC workload from the CMS collider at CERN.

In order to accomplish the task, we start by investigating the best and most accurate ways to measure power consumption and compare different architectures. After, we run several experiments in different chipsets using authentic workloads from LHC and software used by the CMS team to process the data generated by the 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 ..

1.5 Structure of the Thesis

This thesis is structured as following. Firstly, we define the context and scope of the thesis by reviewing relevant and up to date research work. Secondly, we outline measurements tools and best techniques for energy measurement and performance in the scientific computing context. Thirdly, we outline the experiments methodology for comparing the performances of the architectures and respective results. In the Chapter 5, we analyze and draw conclusions

based on the results obtained in the experiments. In the Chapter 6, we present some thoughts on how to lower the energy bill by implementing our learnings thus far. Finally, we wrap up by outlining possible future work and presenting the conclusions of this thesis.

Chapter 2

Background

2.1 Energy efficiency 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 [14], [16] or improve energy efficiency by using RISC architectures [19] [10], [9], . Several other directions have been taken toward energy efficient computing [20] [15], [24] 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 [26]. 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.

2.1.1 CERN and the LHC experiment

The European Research for Nuclear Research (CERN) [3] is a particle physics research laboratory sited in the Franco-Swiss border where thousands of engineers 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 [10], 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 [8] [12] 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 [7] is a grid computing platform where more than 170 computing centers spread across 40 countries [7] collaborate to store and process the data coming from the LHC experiment. According to [7], the WLCG alone is responsible for the distribution, storage and processing of more than 30 Petabytes annually. According to [10], 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 [10], 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.

2.2 Energy performance and measurement

2.2.1 Importance of measuring energy consumption

The study conducted by [20], 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* [5]). 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.

2.3 ARM architecture

- more on (p7)

2.3.1 Literature review

In [10]:

- 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 [9], 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 [9], 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 [9].

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, [9] 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.

2.4 Scheduling based on dynamic energy pricing

2.4.1 Summary

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 [23], [11], [22], [21], [17], 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 [27]. 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 [25]. 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 [28]. 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.

Our solution takes a different perspective when compared with the inter data center solutions. Studies like [25] and [28] do not take the dynamics of electrical pricing into consideration. However, their algorithm is already quite complex and proved NP-complete, to the point they have to come up with heuristic algorithms to apply it in the real world.

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

However, there are some open points that we still have might want to consider. First, as [27] mentions, it is important to insure 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 [25], 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.

2.4.2 General notes

- Intra data center judicious job scheduling based on the heterogeneous architecture of the machines.

- Minimize the electricity costs in data centers by leveraging the dynamical electricity pricing models and heterogeneous computing.

- Online computation schedule the jobs. Jobs are in a queue in a serial fashion and are scheduled depending on the decision of the algorithm at a given time. On the other hand, there are the static scheduling algorithms. These algorithms know all the data they need beforehand and map the jobs to the machines taking that into consideration.

- There are several studies that aim to leverage the potential of geographical load balancing to provide significant cost savings (see [24, 28, 31, 32, 34, 39] in [15])

- Make a problem specification as in [21]

- In our solution, we could also coinsider different stages of functioning such as idling and turning of the machines, depending on the expected workload and the server configuration. The problem might be to understand if we have (or not) knwoledge of the server utilization in the future and its worklaods. Actually, this is an important factor to consider - wether we have or not idea of the future workload.

- It would be interesting to test and simulate our model and algorithm using, for example, workloads and electricity prices in a Google data center. As many studies do, show the potential of our approach by simulating scnarios based on real case data.

- As [27] briefly mentions, does Dynamic Voltage and Frequency Scaling (DVFS) have the same results than our heterogeneous approach in a homogeneous data center ? i.e. are the benefits of a more efficient processor such as ARM surpassed (or the same) as an INTEL working under a DVFS ? The principle seems the same: when energy consumption is smaller (in ARM or INTEL under low DVFS), the jobs take longer to accomplish. This said, is ARM more efficient than INTEL under low DVFS ?

- Should our work be an extension on [27] where, instead of scheduling the workload taking into consideration the job's energy profile, also consider the machine's energy performance ?

2.4.3 Paper's notes

In [18]:

Demand side management are programs implemented by the utility companies to control and influence the user-side behavior. For example, electrical companies often fluctuate the energy price depending on the user's demand.

There is need to encourage household owners to *shift* high demand energy consumptions outside the peak hours, in order to reduce the peak-to-average (PAR) in load demand. [- we aim towards the shifting of schedule different machines depending on the PAR]

Direct load control (DLC) gives the utility companies the possibility to remotely control the household's applications (dim or turn of lights, turn of thermal equipment, amongst others). Though, this model arises some problems related with household's privacy [- see more 'A direct load control model for virtual power plant management' - what if DLC would be implemented for servers and in a heterogeneous scheduling scenario? Would it bing any advantage or liability ?]

An alternative to DLC is smart pricing, where users are encouraged to voluntarily and individually shift their loads out of the peak-hours be increasing the energy prices when the load is big.

One problem with this approach is synchronization: when a large number of users shift their peak at the same time for a low-peak time, the PAR may not be reduced due to the amount of users churning energy at low-peak time. [- this might happen as well with our scheduling strategy. If the amount of users running our scheduling system at the same time is the same, it does not help to reduce the PAR and the prices might get worst]

The paper suggests that households should synchronize their energy usage and schedule their energy applications not only according to the price of the energy at a given time, but also taking into consideration what others are consuming as well. Thus, by acting in synchronization, the group of users can optimize the energy the overall energy consumption and its pricing.

They propose an incentive-based energy consumption pricing model for the smart grid, where the energy source is shared by several users. The meters communicate between each other in a distributed network to find the optimal energy consumption for each user.

Based on game theory, it is shown that through an incentive-based pricing scheme, an optimal scheduling - where users consume less energy and pay

less money - can be achieved.

In [23]:

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 [23], 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 [23], 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.

- The model:

The data center model takes into consideration the possibility of the energy prices to vary over time. The state of the data center can be represented at a given time by a tuple of (i) server availability and (2) energy price.

The job model is characterized by a tuple of (1) service demand - job length - and (2) the set of data centers the job can be scheduled.

The scheduler can turn on/off a server when needed. The scheduling is done based on the server availability and job queue and thus, what matters is the energy consumed by the server when it is 'idle' or 'busy'.

The scheduler also considers he model fairness (which is not important to our study, since we focus in a non-distributed server) and queuing delay. Queuing delay defines the time a job will take to start to be processed, according to relation of the number of jobs scheduled and machine availability.

In [23], 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.

In [11]:

This 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 [22]: In [22], they 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 [11]. Again, it is part of the set of studies on inter datacenter and electrical costs optimizations that location and time based pricing allows.

In [21]: The authors of [21] 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 [27]: In [27], 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 [27], 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.

In [17]:

The authors of [17] 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 [25]:

In [25], 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.
They consider:

- Heteregenous processors
- Heteregenous memories
- Precedence constrained inputs
- Input/output of each task
- Processor execution times
- Data access times
- Time constraints
- and Energy consumption

When solving the data allocation and task scheduling problem, which is an approach much more solid and complete than ours.

In [28]

The authors of [28] 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.

2.4.3.1 New articles

In [29], the authors designed an online algorithm for dynamic pricing of VM resources across data centers in different geographical locations. The authors claim novelty by considering efficient strategies for joint dynamic pricing, job scheduling and resource provisioning.

The authors of [30], [13] also leverage the dynamic electricity pricing models for data centers.

Chapter 3

Energy measurement tools and techniques

The most recent scientific applications have to process and store a considerable volume of data. It is foreseeable that the volume of data will increase considerably in the future, as technology and requirements enhance. In addition to the physical limitations in terms of power density, this phenomenon also increase considerably the costs with energy in a HTC system. Thus, energy consumption has become a major concern amongst the scientific community.

re-write and re-organize everything from now on

In order to find and develop better solutions for improving energy efficiency in High Energy Physics (HEP) computing, it is important to understand how energy is used by the HEP systems themselves. We describe several tools and techniques that facilitate researchers to reach that goal.

As energy efficiency becomes a concern, new solutions have been considered to develop energy efficient systems. One potential solution is to replace the traditional Intel x86 architectures by low power architectures such as ARM. A comparison of the energy efficiency between ARMv7 and x86 Intel architecture is conducted in this article. The experiments use CMS workloads and rely on the techniques and tools described earlier to perform the measurements.

3.1 Tools and techniques for energy measurement

When optimizing power usage, there are two granularities at which one can look at a computing system. The coarser granularity takes into account

the behavior of the whole node (or some of its passive parts, e.g. the transformer) as part of a rack in a datacenter. 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.

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 is more than enough to optimize electrical layout of the datacenters or to provide a baseline for more detailed studies.

A alternative approach takes into account the internal structure of a computing element of an HTC system, as shown in figure 3.1. Nowadays, every 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. Compared to external methods, these on-board components provide high accuracy and reasonably high precision measurements (millisecond level).

A special and slightly different case of these on-board monitors is a new technology called Running Average Power Limit (RAPL), provided by Intel beginning from the Sandy Bridge family of processors.

Contrary to other solutions, 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 permit exploiting such data to build an energy consumption sampling profiler for applications, similar to how performance sampling profilers work (see sec-

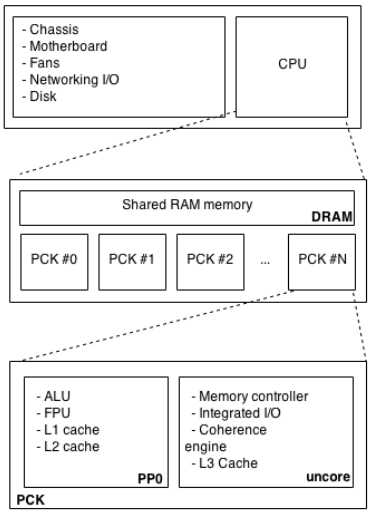

Figure 3.1: Components that contribute for power consumption in HPC

tion **??**). 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.

3.2 Power efficiency measurements with x86-64 and ARMv7

In this section, we demonstrate the potential of some of the tools we previously described. To that end, we perform several measurements of workloads from CERN, running on different architectures. The workloads used in the experiment run on top of Intel x86-64 architecture, traditionally used in HTC and data centers and 32 bit ARMv7 architectures (for similar studies for 64bit ARMv8 and Xeon Phi, please refer to [**?**]). The ARM architecture, initially developed for mobile devices, has been considered [10**?**] as a potential alternative to Intel in HTC, given its energy efficient computing. We also present a brief comparison between ARM and Intel architectures from the energy consumption perspective, based on the results obtained.

3.2.1 Tools and techniques

For the Intel architecture, we used the RAPL technology to perform measurements of the energy consumed by the package, DRAM and cores (figure 3.1). The external measurements for the baseline were performed using a rack PDU, which provides an online API to gather the energy consumed by the system on the rack at a sampling rate of 1 second. For the ARM board, we used the Texas Instrument power monitor chip TI INA231 which allows reading of the energy consumed by the cores and dram at a sampling rate of microseconds. The chip was embedded in the board from the vendor. For the external measurements, we used an external plug-in power monitor with a computer interface for gathering and storing the results. In both cases we read the data as it was exposed to the system via the sysfs / devfs knobs.

Chapter 4

Experiments

We have performed several experiments under different hardware setups. The main goal was to understand how the ARM and Intel architectures perform under similar workloads from an energy consumption standpoint. Wwe compared the results obtained 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 is widely used in production and research at the CMS experiment. We have used the CMSSW framework [ref] (see section Y below) and ParFullCMS [ref] (see section X below). In addition, the workload used for our experiments is a close simulation of real scientific workloads processed in the CMS experiment.

We organized the experiments in 3 sets. The conditions under which the experiments were conducted are greatly similar. Due to hardware and software limitations, 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 small degree. 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 and tools used to perform the energy measurements during the experiments are explained and detailed in the following sections.

The remainder of this chapter in two main section. Firstly, we will describe the architectures of the hardware used during the experiments. Secondly, we describe the setup of the experiments. The former section is organized by what we will call set of experiments (SE). Set of experiments are experiments conducted with the same hardware and software configuration. The degrees of freedom of each experiment are number of events and number of threads processing the workload.

For each setup, we outline the hardware, software setups and the used

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

4.1 Hardware

The focus of this work is to compare energy efficiency of ARM architectures and x86 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 SE.

The ARM machines used were single-board ARM processors developed by Odroid [4] and a server class ARM processor by Boston Viridis [6]. The x86 machines used were manufactured by Intel with the microarchitectures Sandy Bridges and Intel Bonnell. In the following sections, we will describe the hardware architecture and features of the hardware used to run the experiments.

4.1.1 ARM architecture

4.1.1.1 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 work in web servers, cloud and data analytics environment with outstanding power performance [6].

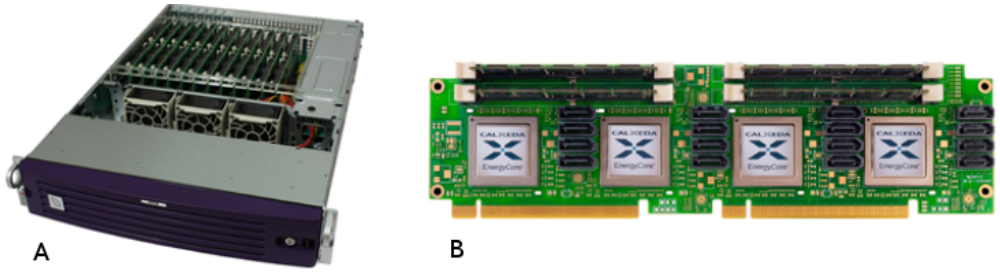

Figure 4.1: A. Viridis Server chassis with 12 energy card in it. B. Energy card with 4 nodes. Taken from [6]

The Boston Viridis server consists of a chassis with twelve racks, each for an energy. Each energy card contains four nodes 4.1. The nodes are ARM based cores fabricated by Calxeda. The block diagram of a Viridis node shows the architectures of the ARM system-on-a-chip. In Viridis server we used in our experiments, each node contains four ARM A9 Cortex core with a clock spead up to 1.4Hz. A memory controller and L2 cache can also be found 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.

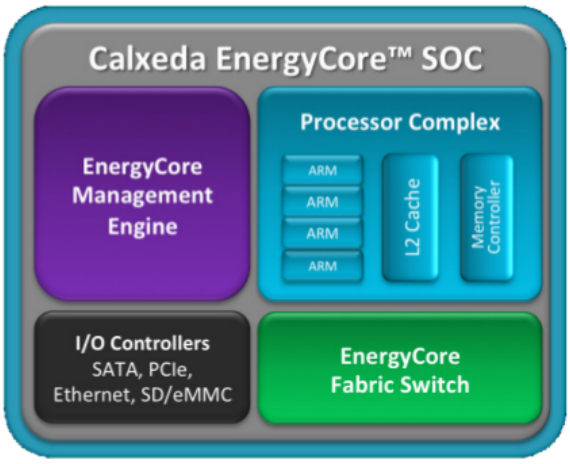

Figure 4.2: Block diagram of Calxeda EnergyCore. Taken from [6]

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].

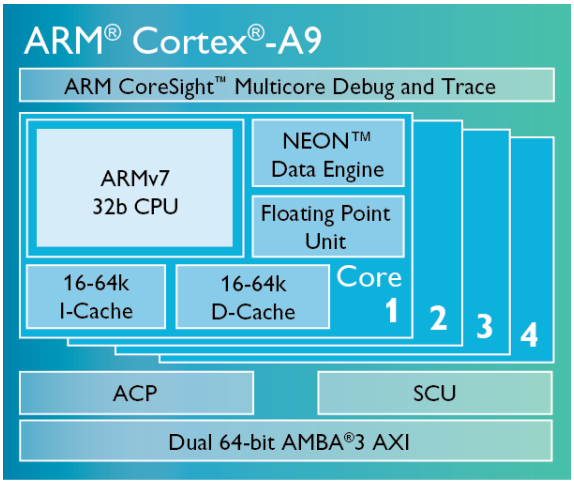

Figure 4.3: Block diagram of Cortex A9. Taken from [2]

4.1.1.2 ODROID-XU3 development board

The ODROID-XU3 [4] is an open-source development board produced by the company ODROID. They claim that the ODROID-XU3 is a "new generation of computing device with more powerful, more energy efficient hardware and smaller form factor" [4]. At the time of these experiments, the ODROID-XU3 was mostly used for testing and platform development, rather than in production scenarios.

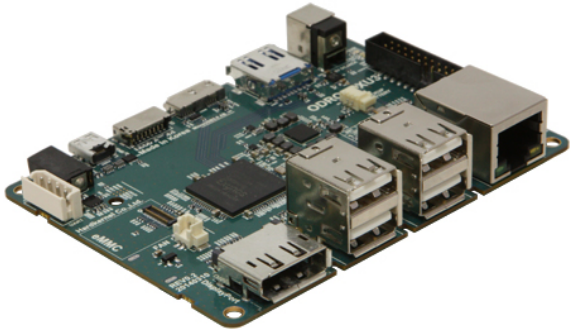

Figure 4.4: ODROID-XU3 development board. Taken from [4]

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 allows the users to read the energy consumed by the cores and DRMA at a sampling rate of microseconds. These readings can be easily triggered and read through software and are a handy and accurate way to make fine-grained energy consumption measurements. We assumed that the measurements made by the TI INA231can be compared to the x86 RAPL technology.

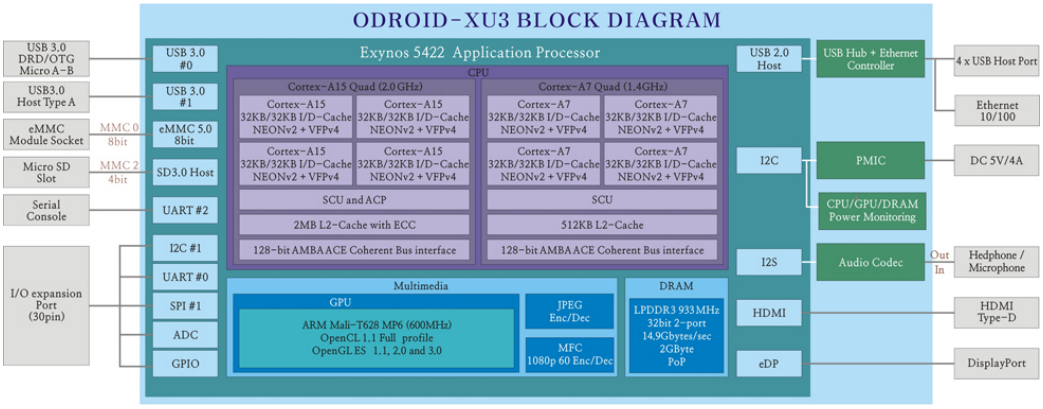

Figure 4.5: ODROID-XU3 block diagram. Taken from [4]

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.

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 a ARMv7 processor produced by Boston Labs [ref]. We ran the same workloads in a x86 Intel Atom and 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 aspects 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 at http://ark.intel.com/products/49490/Intel-Atom-processor-D525-1M-Cache-1_80-GHz (transform in reference)

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 at http://ark.intel.com/products/35365/Intel-Core2-Quad-Processor-Q9400-6M-Cache-2_66-GHz-1333-MHz-FSB (transform in reference)

ARM_Viridis

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

OS: Fedora release 18 (Spherical Cow)

CPU: 4x Quad-Core ARM™ CortexA9™ processor

Memory (MemTotal): 4137780 kB (4GB)

For more detailed specs at http://www.arm.com/products/processors/cortex-a/cortex-a9.php (transform in reference)

Software and workload

We used the CMSSW framework in the generation-simulation mode (GEN-SIM). The workflow performs 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 cores.

Tools for measuring energy consumption

For this set of experiments, we performed physical measurements using the a 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].

4.2.2 Second set of experiments

Hardware specifications

Software and workload

Tools for measuring energy consumption

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 @ 1.8GHz	4	4GB	No internal measurement tool
Intel_QUAD	Intel Sandy Bridge™	Quad CPU Q9400 @ 2.66GHz	4	8GB	No internal measurement tool

Table 4.1: Summary of the 1-SE specifications

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

4.3 Summary

- the hardware setups were chosen given their similarity and possibility of a reliable comparison and hardware availability. it is important to note as well that both Viridis and Odroid machines are much more recent than the x86 hardware compared. It is still a technology that is yet to find production stability.

- We assume that the RAPL and the internal TI INA231 chip for internal energy consumption measurement are similarly accurate and would produce the same results if interchanged.

- the ARM is low energy hardware and was designed and developed with low energy consumption as one of its main goals.

- the x86 intel architecture used in this study is widely used in real world HPC applications. It has a bigger throughput (bits processed by second) per per bit when compared to the ARM architecture.

Chapter 5

Results

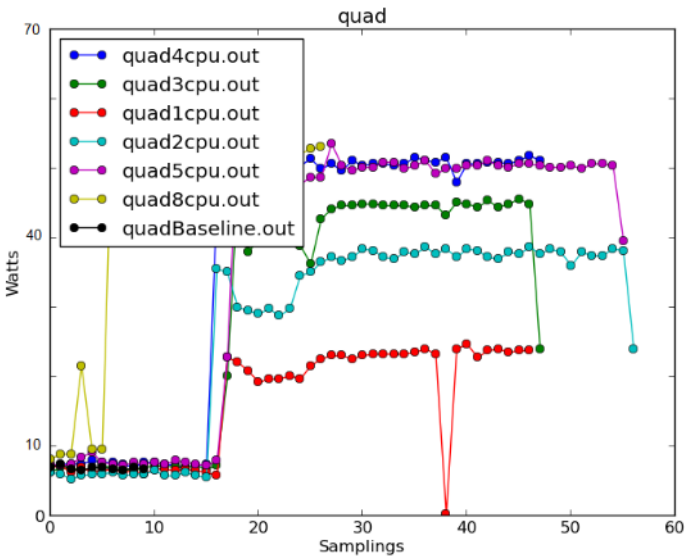

Figure 5.1: Full single threading CMS experiments on Intel Quad

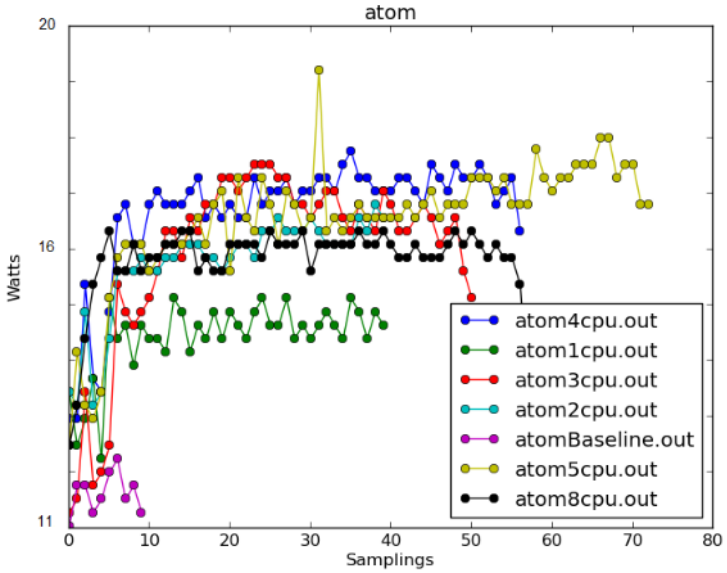

Figure 5.2: Full single threading CMS experiments on Intel Atom

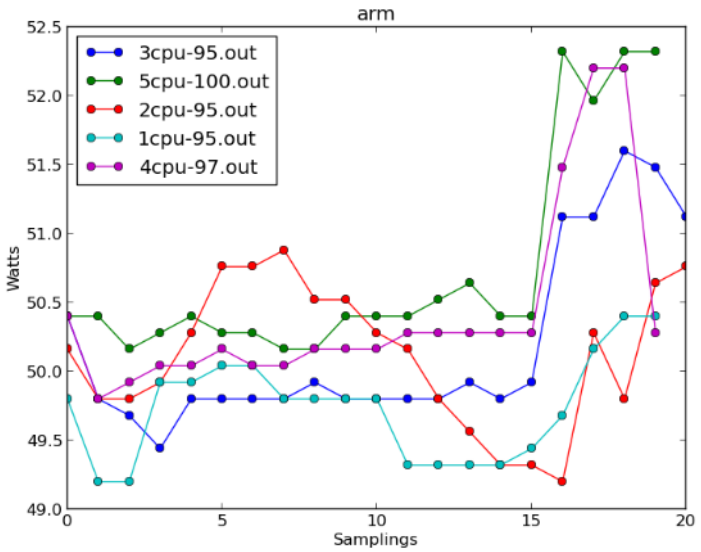

Figure 5.3: Full single threading CMS experiments on ARMv7 server

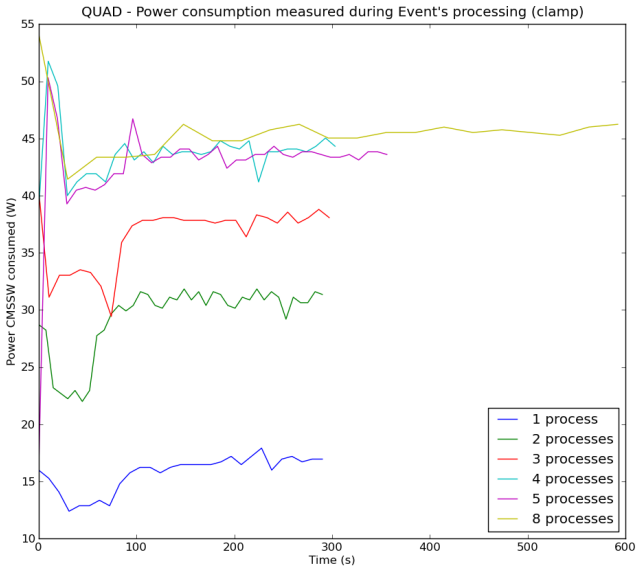

Figure 5.4: Full single threading CMS experiments on Intel Quad - event processing only

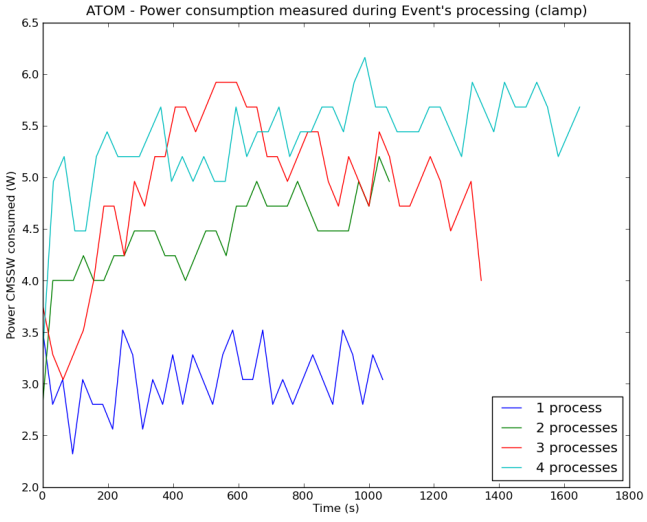

Figure 5.5: Full single threading CMS experiments on Intel Atom - event processing only

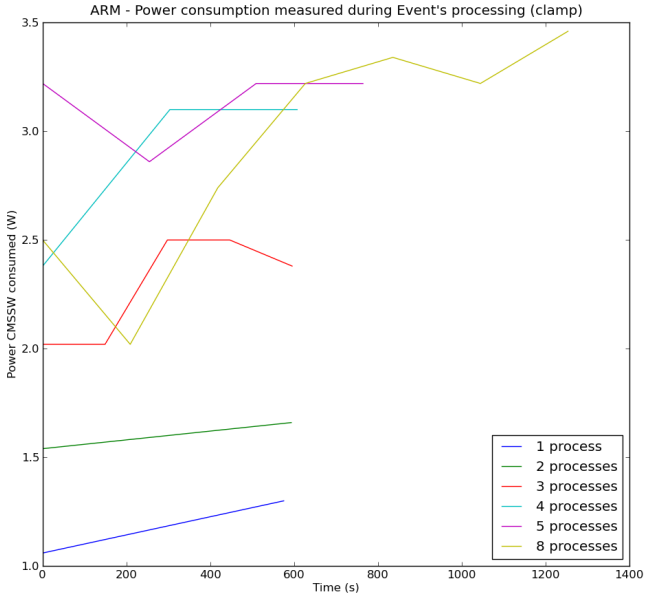

Figure 5.6: Full single threading CMS experiments on ARMv7 server - event processing only

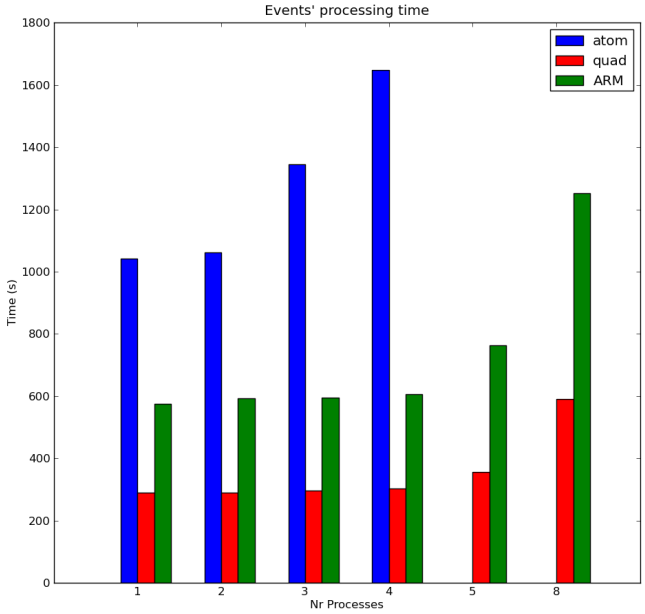

Figure 5.7: Processing time comparison

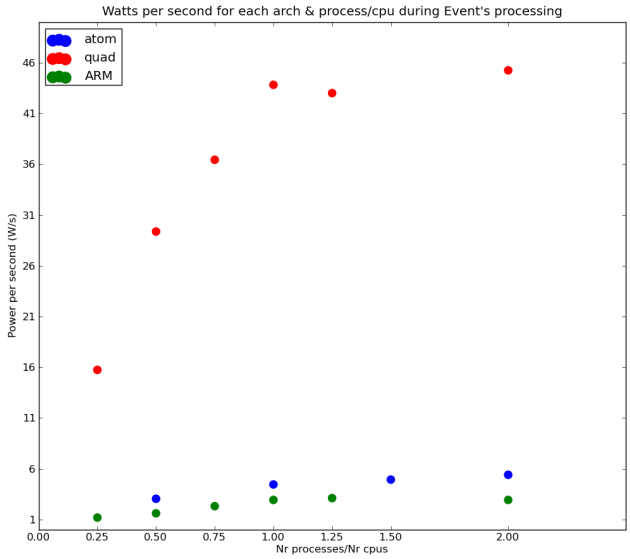

Figure 5.8: Energy efficiency comparison between architectures

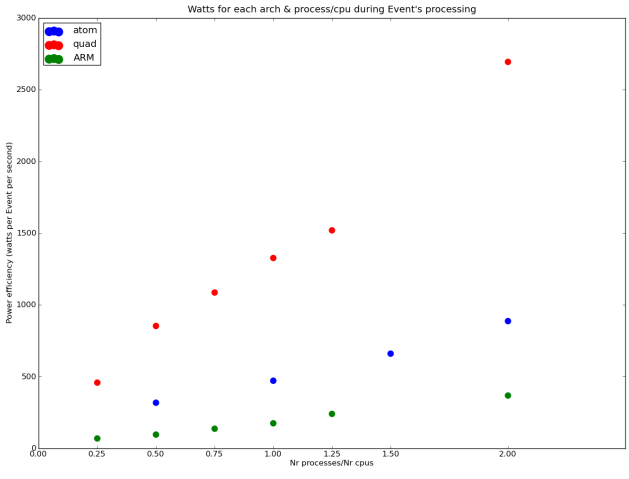

Figure 5.9: Processing stage comparison between architecturesi - 2

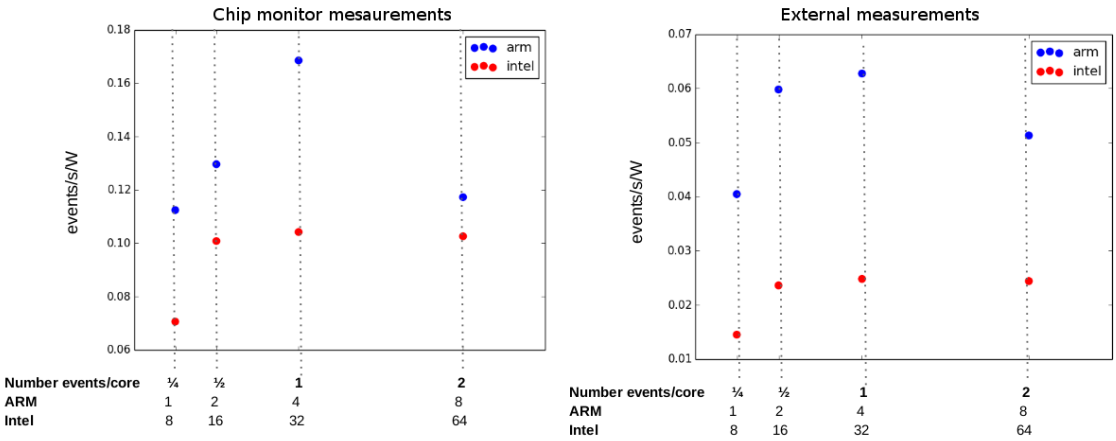

Figure 5.10: Multithreaded ParFullCMS comparison Intel Xeon vs ODROID ARMv7

1. RAPL measurements

	avg pck [W]	**avg pp0** [W]	**avg dram** [W]	**power eff.** [ev/s/W]
A. 32 threads	24.64	11.28	11.61	0.029023
B. 4 processes x 8 threads	39.65	26.20	12.01	0.068764
C. 8 processes x 4 threads	40.41	26.95	12.02	**0.070478**
D. 2 processes x 32 threads	30.95	17.55	11.85	0.045032

Figure 5.11: RAPL measurements with different load combinations

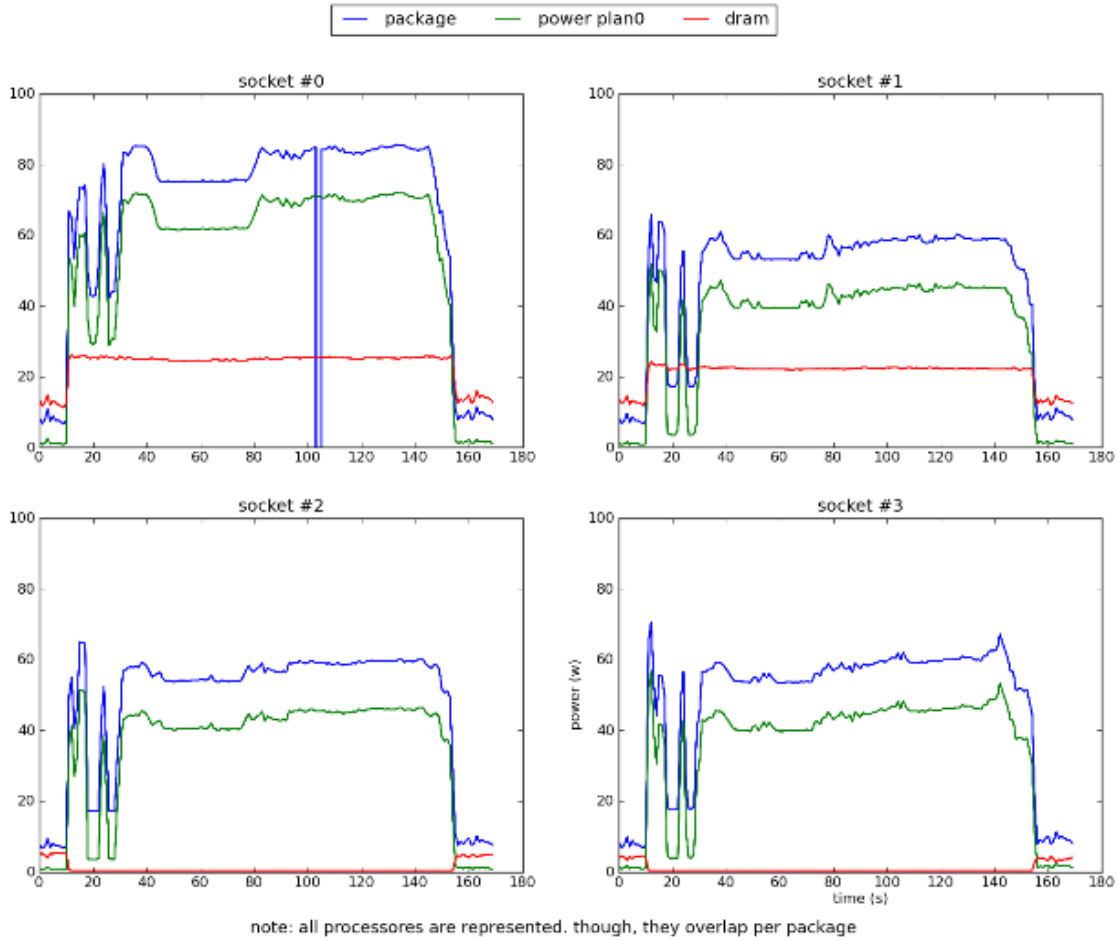

Figure 5.12: RAPL measurements of NUMA nodes - 16 processes with no explicit binding

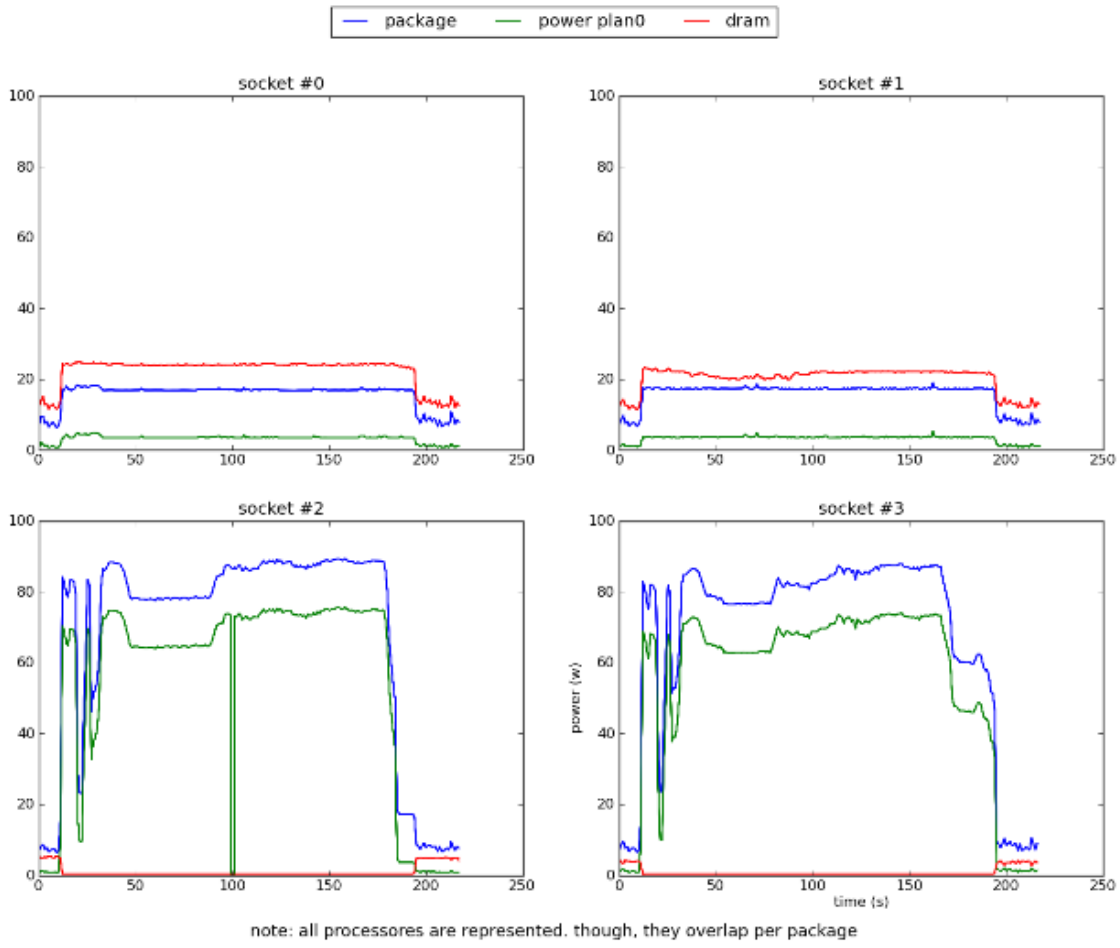

Figure 5.13: RAPL measurements of NUMA nodes - 16 processes. Explicit binding on node #2 and node #3 binding

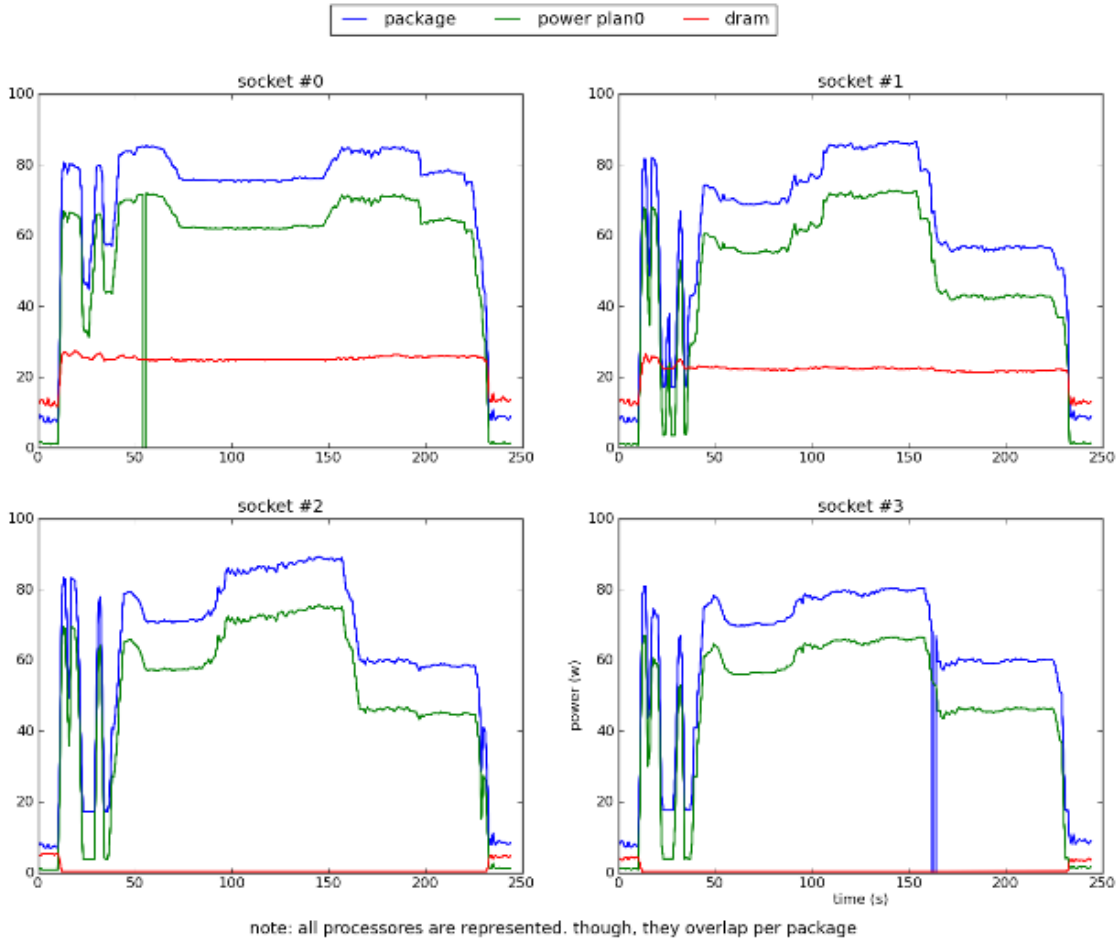

Figure 5.14: RAPL measurements of NUMA nodes - 32 processes with no explicit binding

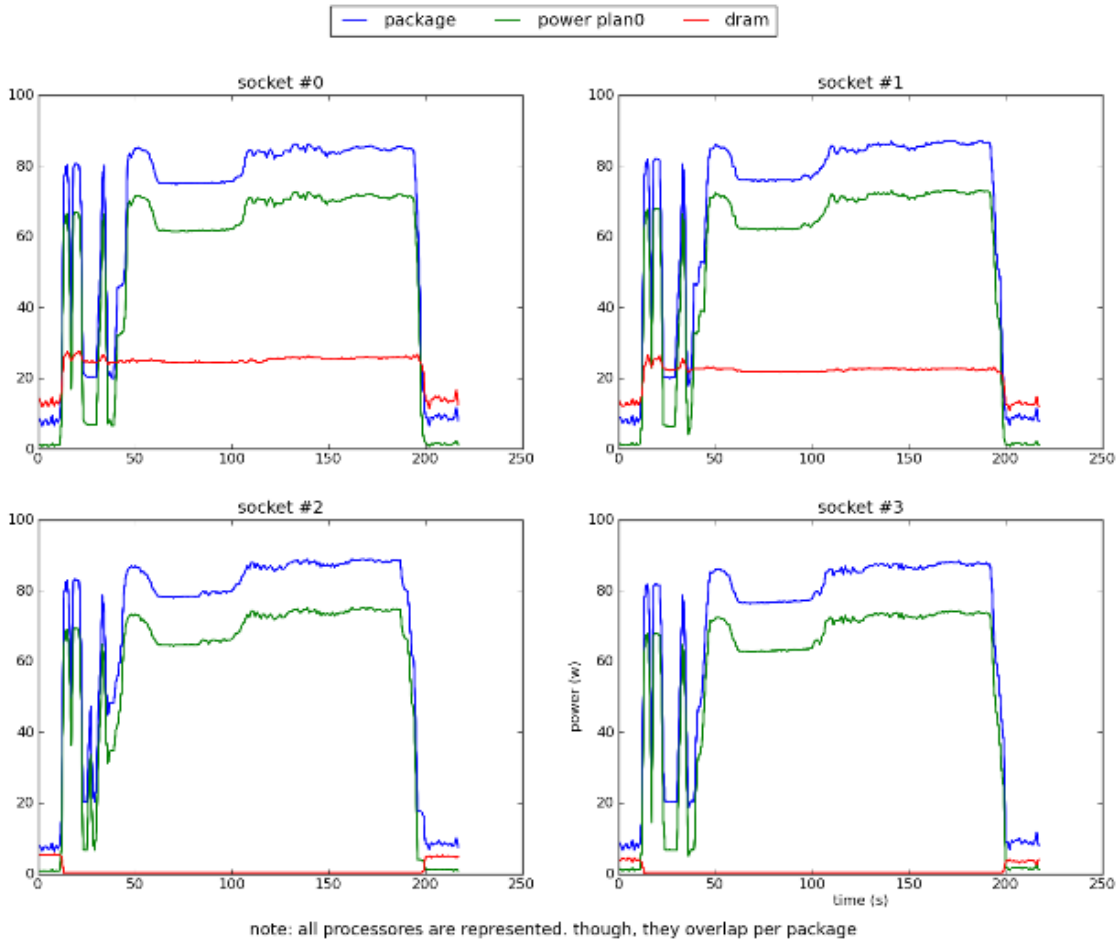

Figure 5.15: RAPL measurements of NUMA nodes - 32 processes. Processes distributed evenly explicitly - 8 processes per node.

Chapter 6

Analysis

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.

Whereas the first two sections analyze the energy efficiency of the platforms studied and the particularities of the tools and techniques used, the last section covers the results and issues which arose when using RAPL to measure the energy consumption in a NUMA environment.

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

Note: do not forget to write about how the different ways of measuring the energy consumption - both in terms across as within architectures (RAPL was not enable in all the x86 measured, for example) - might affect the final results

Note: do not forget to write about how the different set ups might change the final results

6.1 First Set of Experiments

ARM server vs ATOM and QUAD, using clap and software-based experiments

In figures 5.1, 5.2 and 5.3, it is plotted the physical measurements from the beginning of the workload until the end.

Stages

All the experiment sets show 3 stages. The stages can be better identified when plotting the memory workload against cpu usage, rather than the energy consumption measurements (see Figures in GDrive-Add?). The three stages consist in different phase of the experiment. The first

stage consists on the initialization process. During this stage mostly memory is being used, rather than cpu workload. The second stage is the connection phase. It has the goal of fetching the meta data fetching from the CERN servers needed to perform the reconstruction of the events. Anew, during this stage, the cpu load is low when compared to the memory workload. Lastly, the third stage corresponds to the event processing phase. Therefore, the last stage is cpu intensive and the one that is performing the useful computation for the reconstruction of events.

Stages comparison

Regardless the number of processes running, the time for the three stages is constant in all the experiment sets, 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 cores to process the events concurrently. In the overcommitted situation, the time increase follows a 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.

Importance of the stages

Unarguably, the most important stage when studying the energy efficiency of workload in CERN is the third stage. There are two main reasons for that: first, the CMSSW configuration at either CERN, 2nd and 3rd tiers has proxies and caches that speedup the second stage [refs]. Lastly, given the amount of data to be processed in the last phase and thus the energy consumed by the event processing stage, the energy consumed by the former stages becomes irrelevant. Therefore, in the remainder of the chapter we focus our analysis on the event processing stage only. The energy measurements of only the third stage are shown in the figures 5.4, 5.5 and 5.6.

Relation number processes/number cores

The relation between the number of processes and number of cores and the influence of its ratio is clear in the figures 5.4, 5.5 and 5.6. As expected, when the CPU is overcommitted the task takes more time than otherwise. For the QUAD 5.4 and ARM 5.6 architectures, it is clear that when the number of processes is bigger than 4, the task takes more time to be processes. In the ATOM architecture 5.5, the same happens when the number of processes exceed 2. More detailed information about this behavior can be drawn by analyzing the data

acquired by the software-based tools during the experiments [**include ps, powertop, ect.. plots ?**]

Time comparison

When comparing the time taken by the different architectures to process the same task 5.7, the pattern is evident. Regardless the number of processes launched, the QUAD architecture is faster than ATOM and ARM, whereas ATOM is faster than ARM. This fact is due to the architectures characteristics and its specifications, most notably the CPU clock speed.

Energy efficiency comparison

The energy efficiency metric used in this study is the ratio of performance per power consumed. Performance consist on the average of events computed per second for each architecture. More details about the reasons why Events were considered the main data unit for CERN workloads are explained in the Methodology Section. Given the above mentioned metrics, it is clear that systems are proportionally energy efficient with its ratio performance per watts. Therefore, by analyzing the Figure 5.8, 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, ATOM is more energy efficient than QUAD architecture.

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 plot shows spikes and rough transitions between samples. Moreover, the error tends to increase proportional to the human interaction with the experiment. Therefore, it is more effective to use digital and automated ways to sample and log the data acquired during the measurements. The advantages of using digital and automated ways to sample and log data can be seen further on in the SSE.

Measuring tools: software-based monitoring

In this particular set of experiments, the software monitoring tools used were 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 system behavior learning.

Thus, even if the output is does not directly show information about energy consumption of the system, it can be important to support and explain expected - and unexpected - behaviors.

6.1.1 Comparison ARM and Intel architecures

6.1.2 Tools and techniques

6.2 Second Set of Experiments

ARM board and Intel Xeon, using on chip and external measurements

6.2.1 Comparison ARM and Intel architecures

6.2.2 Tools and techniques

6.3 Third Set of Experiments

Intel Xeon, using RAPL to measure energy consumed by the different nodes, with different types of binding

Chapter 7

Lowering the energy bill in a multi energy price environment

- Use the greedy and jobshop models to schedule works across different energy profile machines.
    - Use experiments done to characterize the machines
    - Code scheduling algorithm

Chapter 8

Future Work

Chapter 9

Conclusions

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

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

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

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

[8] 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.

[9] 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).

[10] 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.

[11] 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.

[12] 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.

[13] Chen, H., Hankendi, C., Caramanis, M. C., and Coskun, A. K. Dynamic server power capping for enabling data center participation in power markets. In *Proceedings of the International Conference on Computer-Aided Design* (2013), IEEE Press, pp. 122–129.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Mohsenian-Rad, A.-H., Wong, V., Jatskevich, J., Schober, R., and Leon-Garcia, A. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *Smart Grid, IEEE Transactions on 1*, 3 (Dec 2010), 320–331.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Zhao, J., Li, H., Wu, C., Li, Z., Zhang, Z., and Lau, F. Dynamic pricing and profit maximization for clouds with geo-distributed datacenters.

[30] Zheng, X., and Cai, Y. Reducing electricity and network cost for online service providers in geographically located internet data centers. In *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on* (2011), IEEE, pp. 166–169.

Appendix A

First appendix

This is the first appendix. You could put some test images or verbose data in an appendix, if there is too much data to fit in the actual text nicely.