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! — January 10, 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:**	34
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

- Future computational systems will require more computational resources to meet its requirements.

Chapter 2

Background

2.1 Energy consumption in Scientific Computing

2.1.1 Literature review

According to [3], the computing requirements for HPC have increased particularly in recent years. Projects of the magnitude and complexity of the Large Hadron Collider are overwhelming examples of that fact. To achieve results like the discovering of the Higgs boson and other significant scientific advances, it was necessary a to distribute the processing tasks across several partners and institutions through the WLCG. The equivalent capacity of such distributed systems was between 80,000 and 100,000 x86-64 cores in 2012. Further projects and discoveries will demand even more processing capacity from the WLCG. For example, as stated by [3], to upgrade the LHC detectors luminosity to its full power the datasets will increase sizes by 2-3 orders of magnitude and processing power will have to increase in proportion.

In [2], 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 [2], 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 [2].

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, [2] 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.

2.2 High Throughput Computing

2.2.1 Literature review

2.3 CERN and the LHC experiment

2.3.1 Literature review

2.4 Energy performance and measurement

2.4.1 Literature review

on importance of energy consumption for engineers and scientists
The study conducted by [7], 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* [1]). 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.

2.5 ARM architecture

2.5.1 Literature review

In [3]:

- 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.

2.6 Architecture scheduling based on dynamic energy pricing

In [6]:

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 [8]:

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 [8], 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 [8], 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 [8], 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 [4]:

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.

– [**Notes**]

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

- 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

- 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 [5])

2.6.1 Summary

There are two branches of researching on energy efficiency that are related with heterogeneous computing and dynamic energy pricing.
In the studies related with the spacial-time dynamic of energy pricing, the emphasis is given to the scheduling of jobs across data centers that are located in different places. The main idea is to exploit the fact that energy prices differ in data centers located in different places. There are concerns with fairness, server availability and queue delays. In addition, there are also several research studies related with migration of cloud computing jobs.
On the

Chapter 3

Tools and techniques for measuring energy efficiency of scientific software applications

Chapter 4

Experiments

4.1 Experiments methodology

The experiments were performed in different sets. Whereas the first two sets of experiments aim to provide a straightforward comparison between ARM and Intel technologies, the third set of experiments aims to study the influence of a NUMA environment in hight performance computing from an energy consumption perspective. In each set, we used different techniques and tools to perform the energy measurements. The techniques and tools used to perform the measurements are described and analyzed in depth in Section 3.

The first part of this chapter outlines the scope, methodology and measurement tools used for each set. The latest part shows the results of the experiments, which are analyzed in the next chapter.

Throughout the rest of the document, the different experiments will be termed as first (FSE), second (SSE) and third set of experiments (TSE).

4.1.1 Environment for Power and Performance measurements

outline CMSSW, architectures, and worloads

4.1.2 First Set of Experiments

Done at Aalto. Explain methodology and scope.

4.1.3 Second Set of Experiments

Done at CERN. Explain methodology and scope.

4.1.4 Third Set of Experiments: RAPL in NUMA environment

Done at CERN. Explain methodology and scope.

4.2 Results

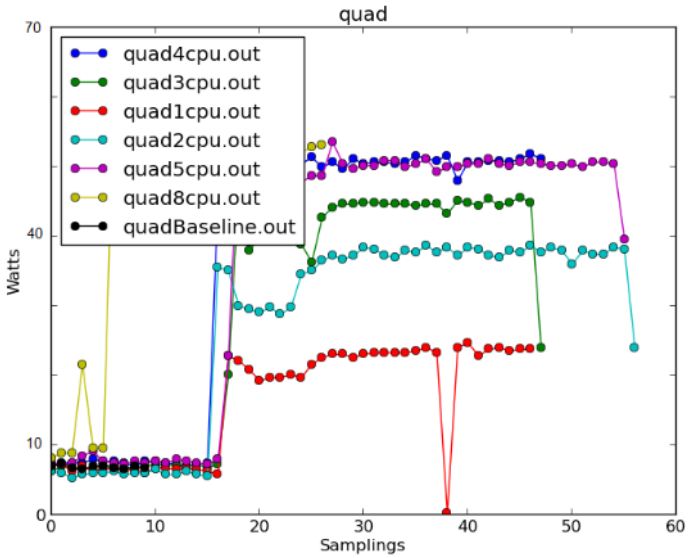

Figure 4.1: Full single threading CMS experiments on Intel Quad

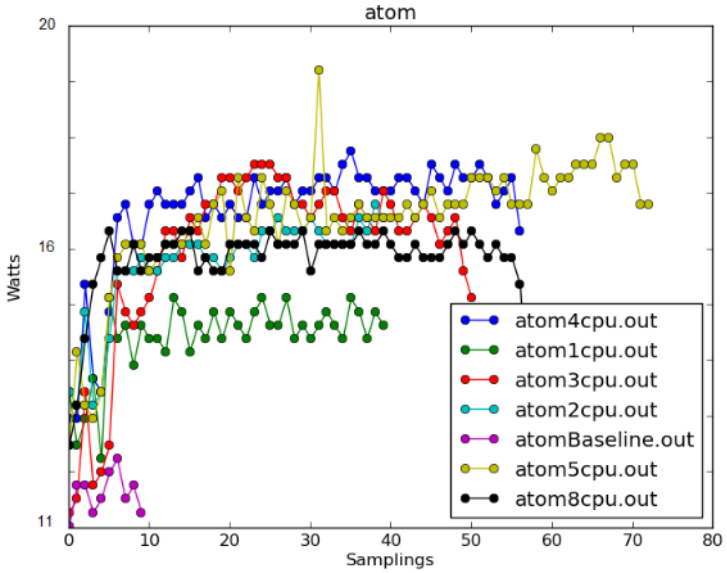

Figure 4.2: Full single threading CMS experiments on Intel Atom

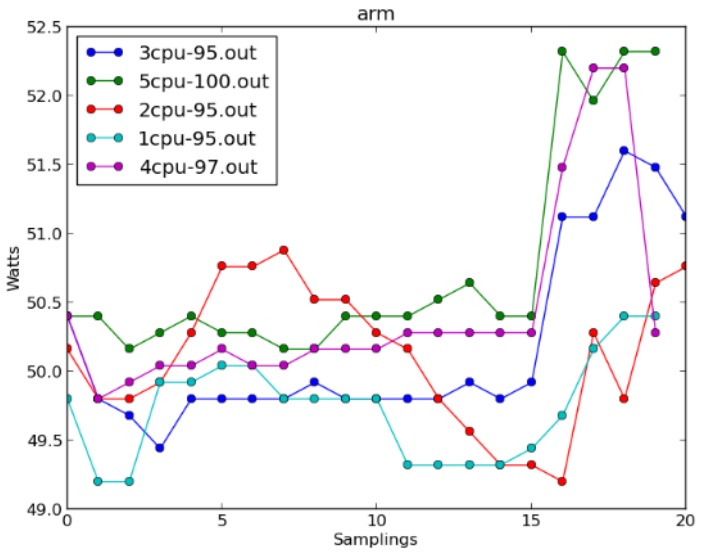

Figure 4.3: Full single threading CMS experiments on ARMv7 server

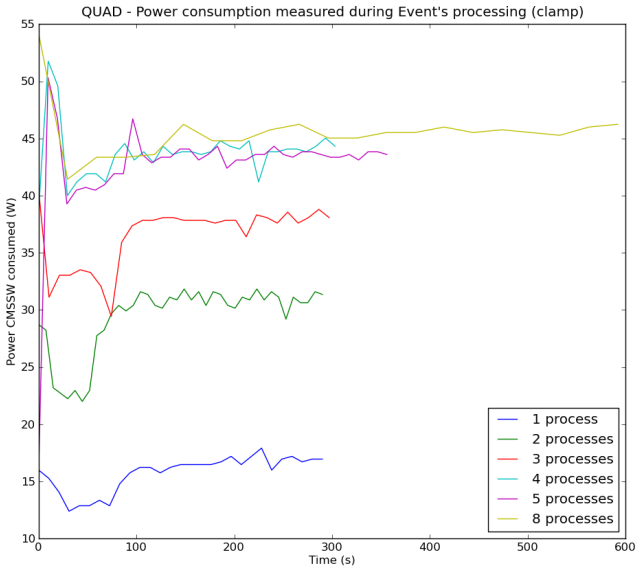

Figure 4.4: Full single threading CMS experiments on Intel Quad - event processing only

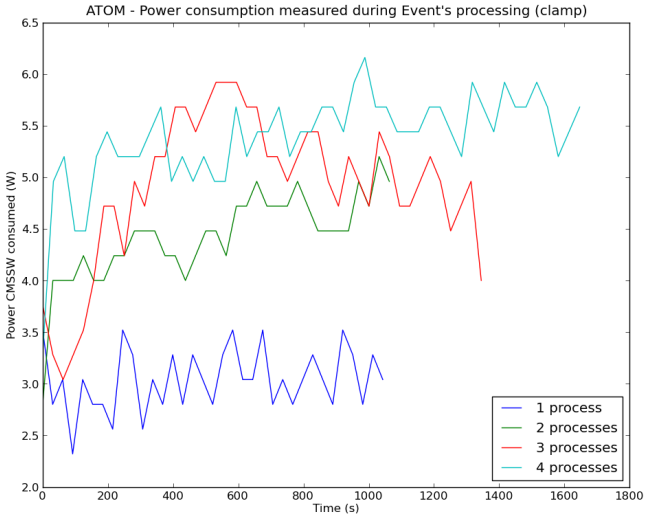

Figure 4.5: Full single threading CMS experiments on Intel Atom - event processing only

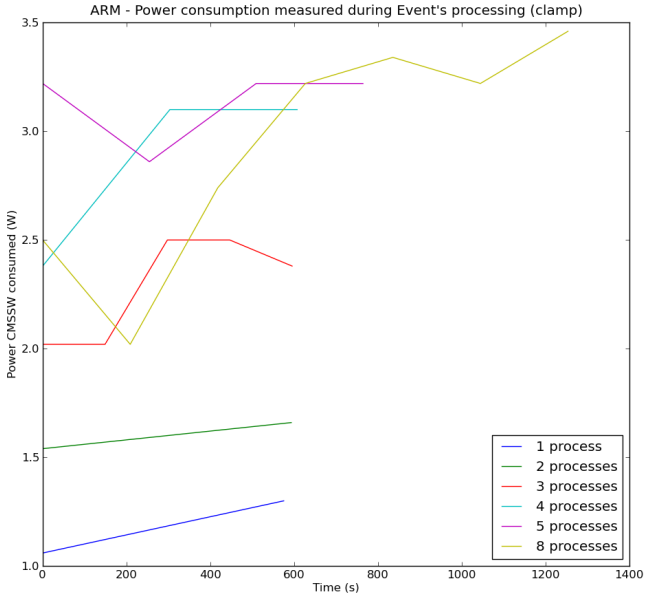

Figure 4.6: Full single threading CMS experiments on ARMv7 server - event processing only

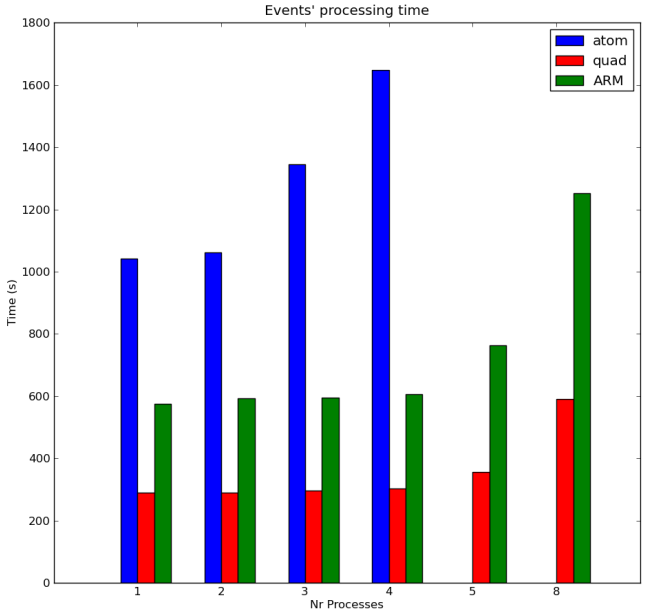

Figure 4.7: Processing time comparison

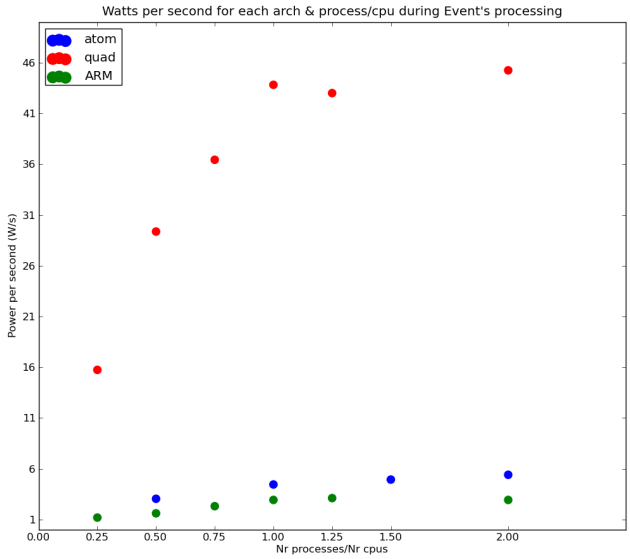

Figure 4.8: Energy efficiency comparison between architectures

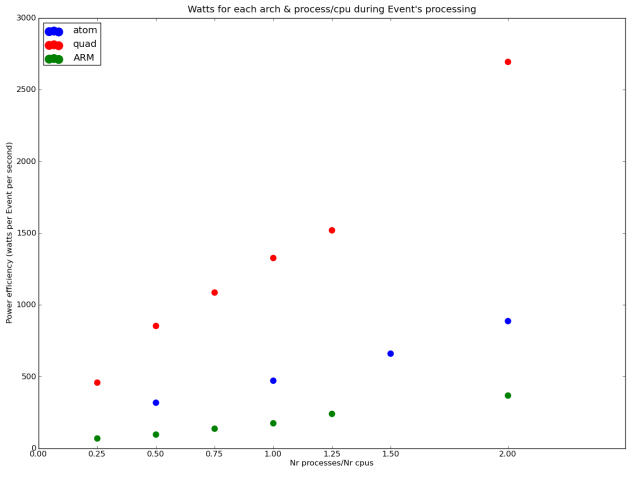

Figure 4.9: Processing stage comparison between architecturesi - 2

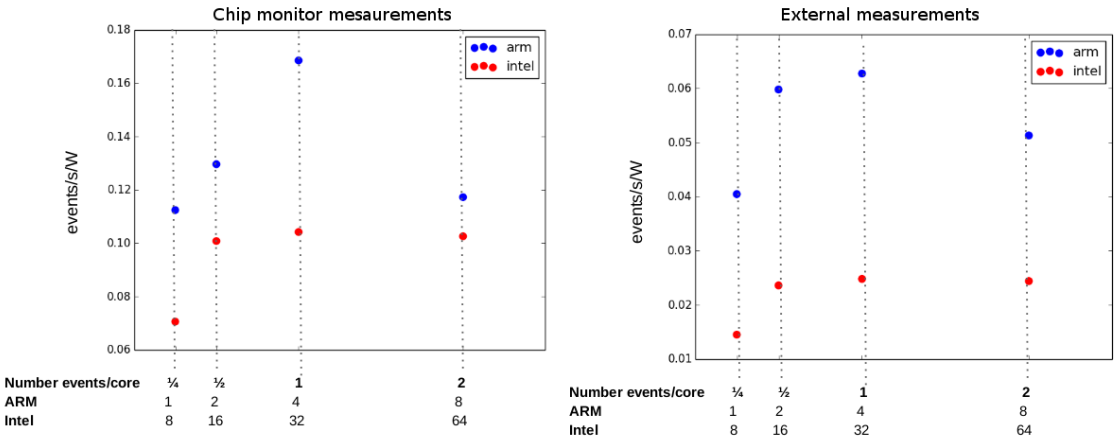

Figure 4.10: Multithreaded ParFullCMS comparison Intel Xeon vs ODROID ARMv7

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 4.11: RAPL measurements with different load combinations

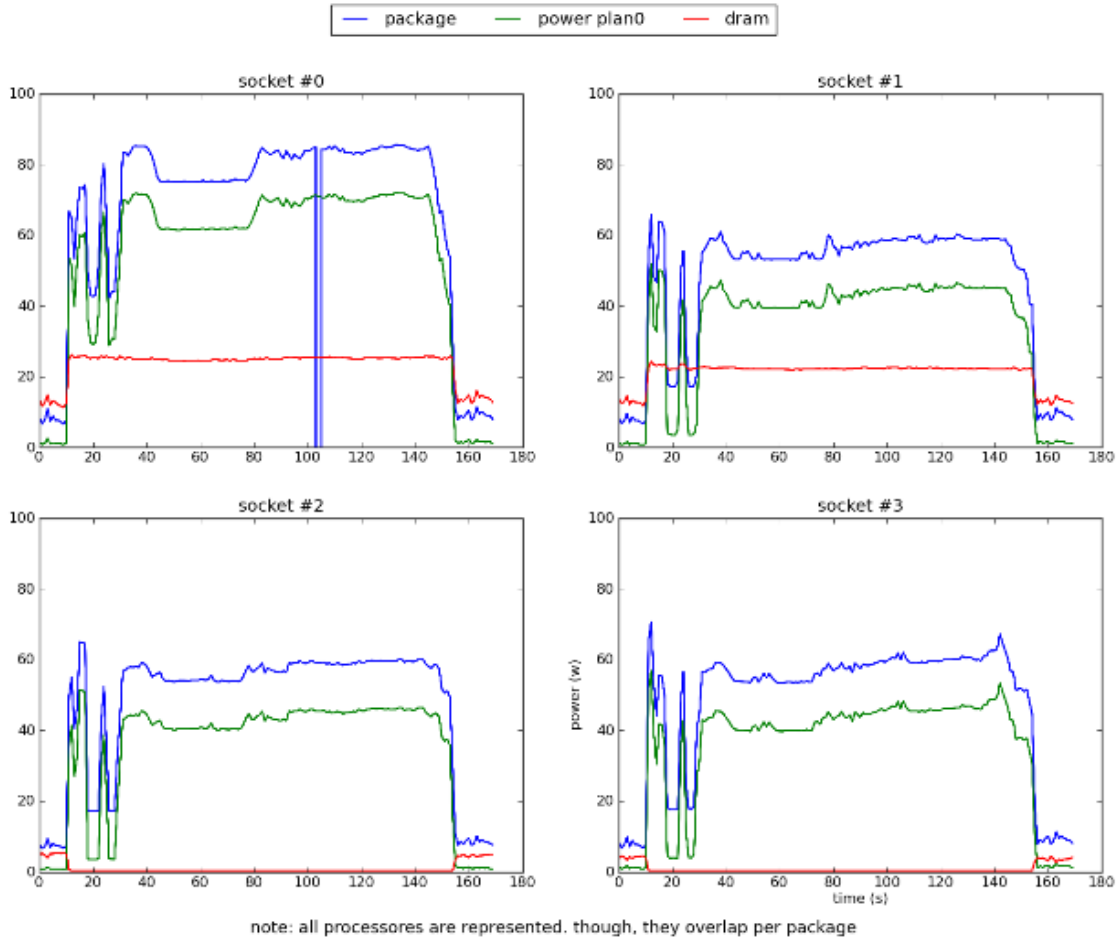

Figure 4.12: RAPL measurements of NUMA nodes - 16 processes with no explicit binding

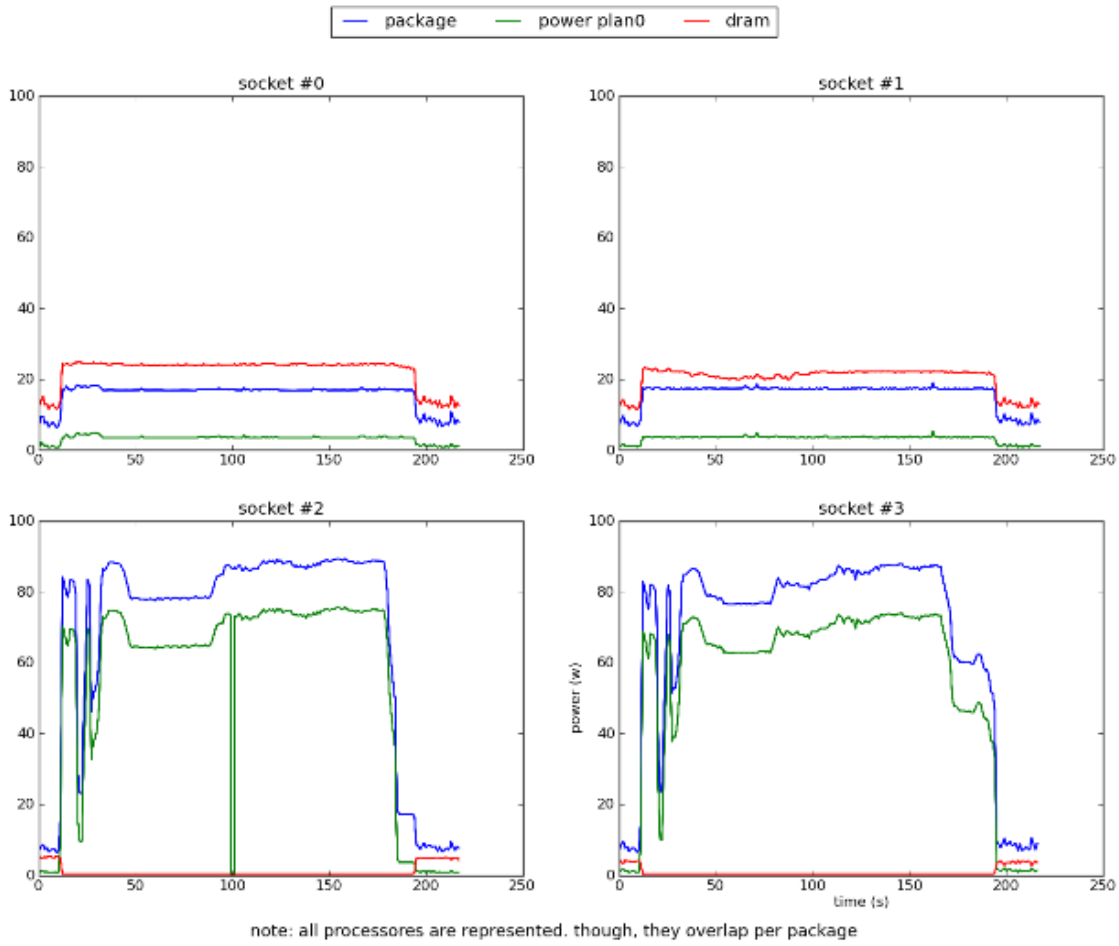

Figure 4.13: RAPL measurements of NUMA nodes - 16 processes. Explicit binding on node #2 and node #3 binding

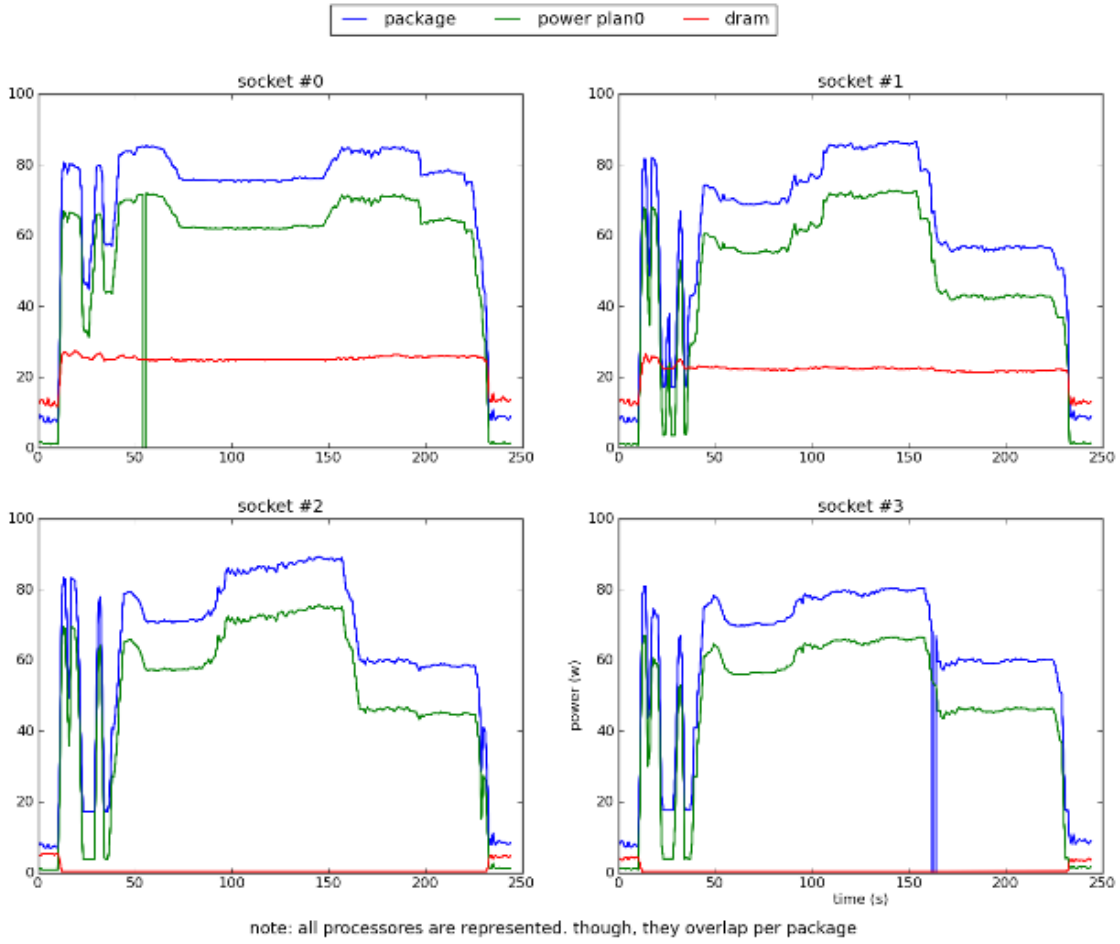

Figure 4.14: RAPL measurements of NUMA nodes - 32 processes with no explicit binding

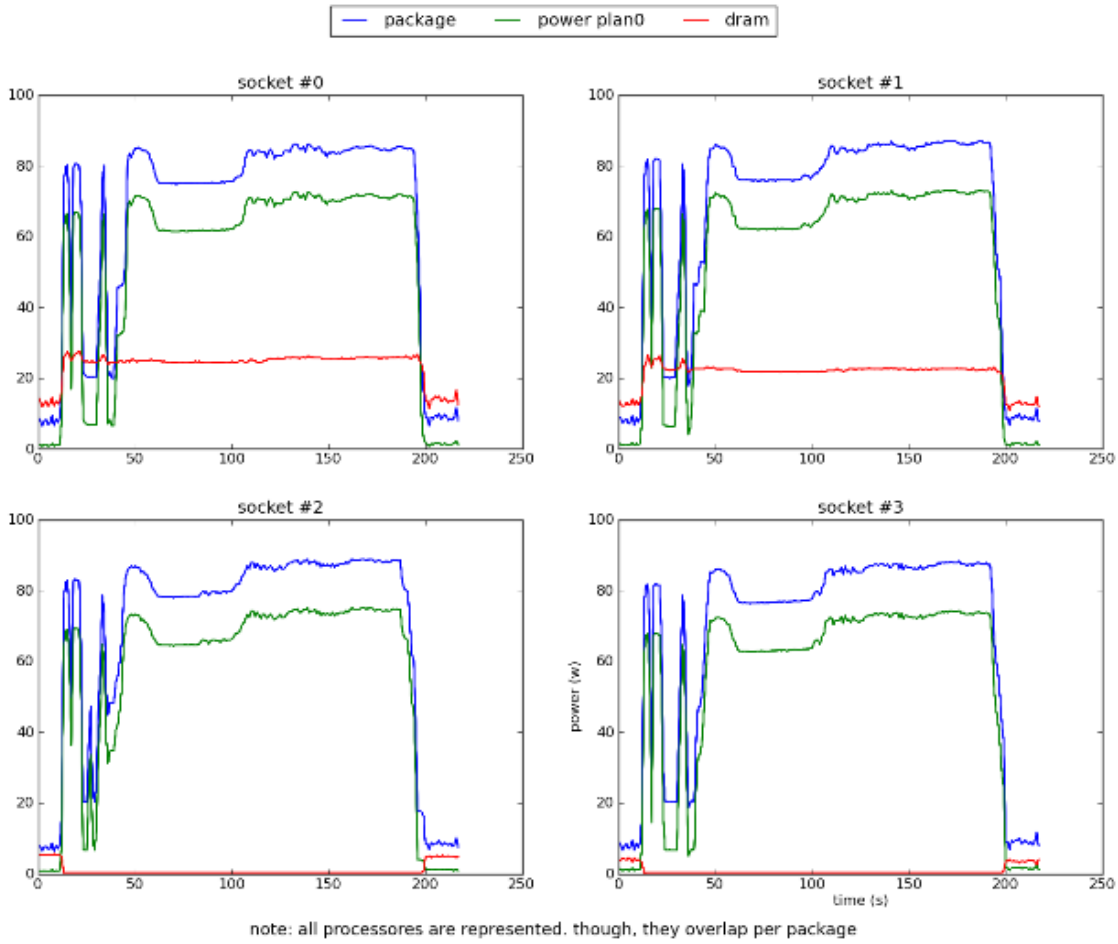

Figure 4.15: RAPL measurements of NUMA nodes - 32 processes. Processes distributed evenly explicitly - 8 processes per node.

Chapter 5

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.

5.1 First Set of Experiments

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

In figures 4.1, 4.2 and 4.3, 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 4.4, 4.5 and 4.6.

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 4.4, 4.5 and 4.6. As expected, when the CPU is overcommitted the task takes more time than otherwise. For the QUAD 4.4 and ARM 4.6 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 4.5, 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 4.7, 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 4.8, 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.

5.1.1 Comparison ARM and Intel architecures

5.1.2 Tools and techniques

5.2 Second Set of Experiments

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

5.2.1 Comparison ARM and Intel architecures

5.2.2 Tools and techniques

5.3 Third Set of Experiments

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

Chapter 6

Future Work

Chapter 7

Conclusions

Bibliography

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

[2] 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).

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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.