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! — November 3, 2015 — DRAFT!

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

Aalto University
School of Science

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

ABSTRACT OF
MASTER'S THESIS

Author:	Gonçalo Marques Pestana		
Title:	Energy Efficiency in High Throughput Computing Tools, techniques and experiments		
Date:	1 December, 2014	**Pages:**	37
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

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. Ww 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).

1.1 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 [11] and a server class ARM processor by Boston Viridis [12]. 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.

1.1.1 ARM architecture

1.1.1.1 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 [12].

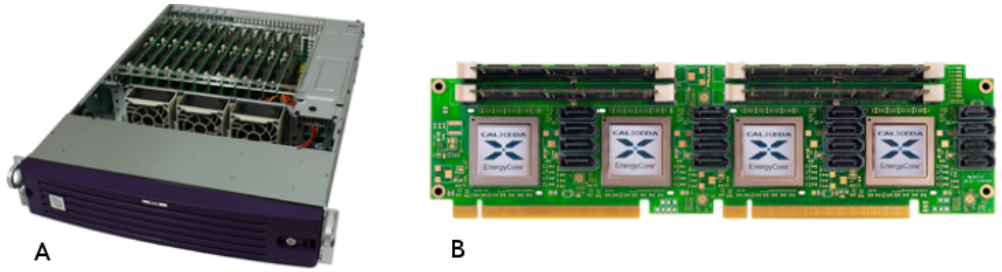

Figure 1.1: A. Viridis Server chassis with 12 energy card in it. B. Energy card with 4 nodes. Taken from [12]

The Boston Viridis server consists of a chassis with twelve racks, each for an energy. Each energy card contains four nodes 1.1. 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 1.2. These energy measurement blocks were used to perform part of the energy measurements with this setup.

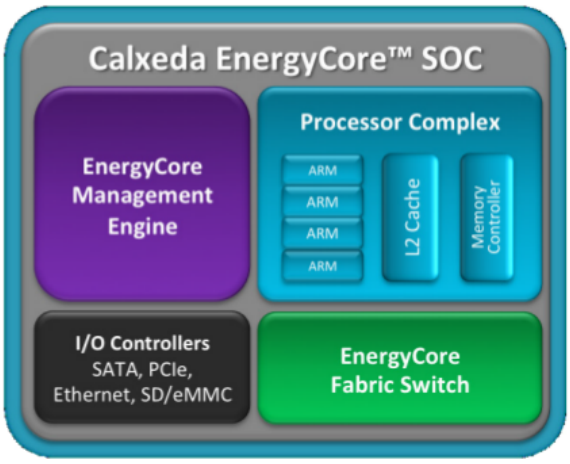

Figure 1.2: Block diagram of Calxeda EnergyCore. Taken from [12]

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].

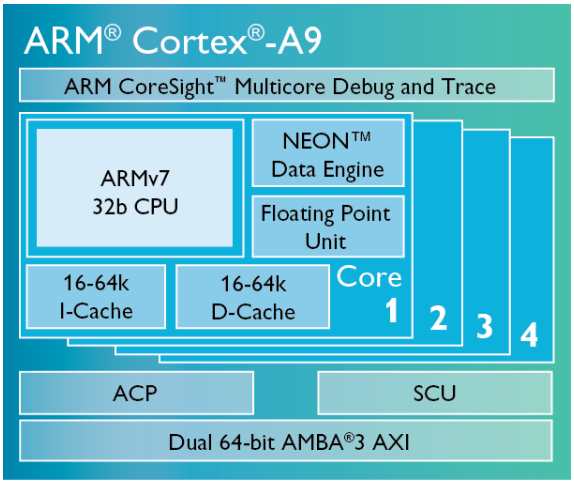

Figure 1.3: Block diagram of Cortex A9. Taken from [2]

1.1.1.2 ODROID-XU3 development board

The ODROID-XU3 [11] 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" [11]. At the time of these experiments, the ODROID-XU3 was mostly used for testing and platform development, rather than in production scenarios.

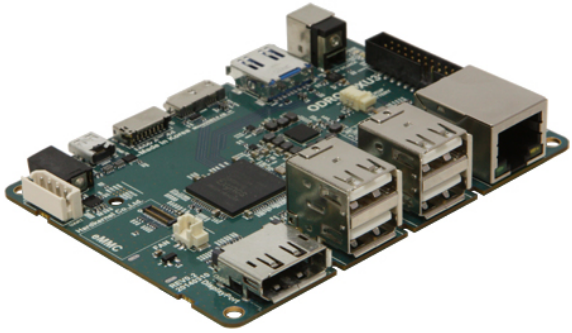

Figure 1.4: ODROID-XU3 development board. Taken from [11]

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 1.5.

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.

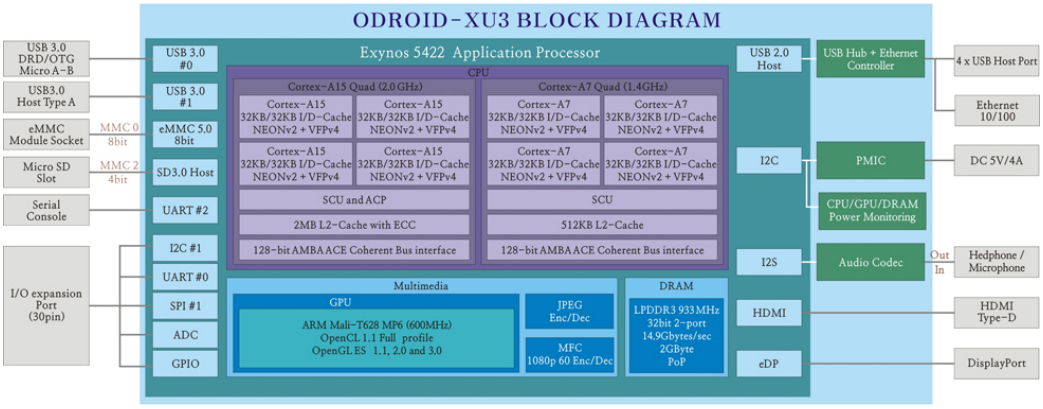

Figure 1.5: ODROID-XU3 block diagram. Taken from [11]

1.1.2 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.

1.2 Experiments setup

1.2.1 First set of experiments

Hardware specifications

For the first set of experiments, we used three machines with different hardware setups. The three machines differ in architecture and general purpose. The ARM_virdis is a server rack with CPU consisting of ARMv7 processors produced by Boston Labs [ref]. We ran the same workloads in a x86 Intel Atom 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 refer to [6]

Intel_QUAD

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

OS: Scientific Linux release 6.5 (Carbon)

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

Memory (MemTotal): 7928892 kB (8GB)

For more detailed specs refer to [7]

ARM_Viridis

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 1.1: Summary of the 1-SE specifications

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 refer to [3]

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 physical 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 1.6). The clamp meter supports a maximum of 200A current, which was enough for our experiments. In addition, it presents an accuracy of +-2.5%. For more specifications about the clamp used, refer to [REF].

Figure 1.6: Mastech MS2102 clamp meter used to measure energy consumption. Taken from [] - cite Mastech website

1.2.2 Second set of experiments

Hardware specifications

For the second set of experiements, we again used three machines with different hardware setups. As in the 1SE, the three machines differ in architecture and general purpose. The ARM_virdis, which was used in the 1SE, was also used during the second set of experiments. In addition to ARM_viridis, we also resort to another machine powered by an ARM CPU. The ARM_odroid is a development board manufactured by HardKernel [ref] and it is intended to provide a cheap and easy way to develop hardware and software in a ARM architecture. To represent the Intel architecture we used Intel_xeon, a machine from the Intel Sandy Bridge family and powered by an Intel R5-2650 CPU. This machine was part of a server rack and it was intended for high performace scientific computation in a production scenario.

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

ARM_viridis

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

OS: Fedora release 18 (Spherical Cow)

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

Memory (MemTotal): 4137780 kB (4GB)

For more detailed specs refer to [3]

ARM_odroid

kernel & sys: Linux 3.10.24 LTS

OS: Ubuntu 14.04.3 LTS (Trusty Tahr)

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

Memory: 2GB

For more detailed specs refer to [10]

Intel_xeon

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

OS: Scientific Linux release 6.5 (Carbon)

CPU: 4x Intel^{TM} CPU E5-2650

Memory: 252GB

For more detailed specs refer to [8]

Software and workload

We used the CMSSW's mode ParCullCMS for generating the workload. The ParFullCMS mode is a multi-threaded Geant4 [13] benchmark. It uses a complex CMS geometry for the event simulation and has the advantage of being multithreaded in both Intel and ARM architectures. As in the first set of experiements, we measured the energy consumed by the machine under different physical core loads. The core-load levels used were 1/4, 1/2, 1 and 2 threads per number of physical cores.

Metrics

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

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

Tools for measuring energy consumption

For the 2SE, we performed both internal and external measurements in the Intel_xeon and ARM_odroid. On the ARM_viridis, we performed only internal measurements given the lack of a tool that would performe with the same degree of accuracy than the tools used for Intel_xeon and ARM_odroid. All the tools used to measure energy consumption were embeeded in the hardware setup of the machines.

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

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

For the ARM_viridis, we used the capabilities of the Intellegent Platform Management Interface (IPMI) [9] included in the server from origin. The IPMI is a chip that runs as a separate subsystem and is attached to the motherboard. The ARM_viridis implementation of IPMI provide several capabilities, namely interlal hardware energy monitoring. We leveraged the IPMI tools to perform internal energy consumption of the ARM cores during the experiments

Machine codename	Architecture	CPU	N° 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 1.2: Summary of the 2-SE specifications

1.3 Summary

In this chapter we outlined the setup of the experiments form both the hardware and software perspectives.

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

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 ARM_viridis and ARM_droid machines are much more recent than the compared Intel hardware. All the ARM machines used were still a technology that was yet to find production stability at the moment of the experiments. On the other hand, the Intel architecture used in this study was widely used in real HPC applications at the time of this study.

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

Chapter 2

Results

This chapter presents the results of the experiment descibed in the later chapter. The plots and figures in this chapter can be divided into 3 sections. The first section has the results of the first set of experiments. The second section has the results of the second set of experiments. The last section of figures shows results that related with the usage of CMSSW is a Non-Uniform Memory Access (NUMA) environment.

In the next chapter, we analyse the results based on the content of this chapter.

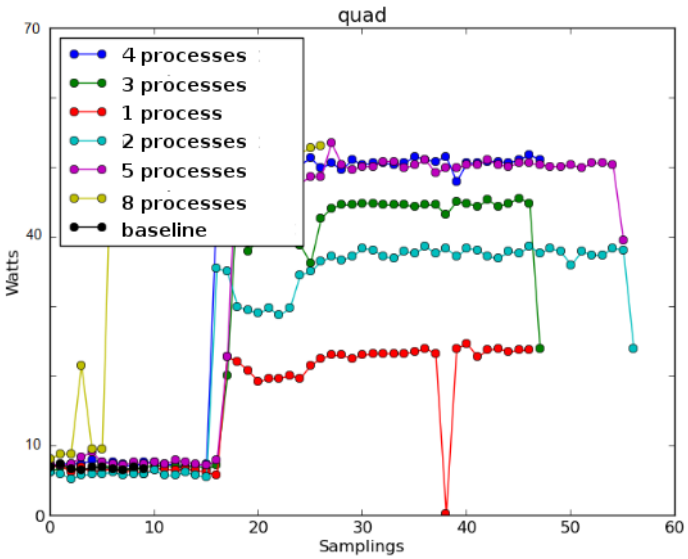

Figure 2.1: All stages of the CMSSW experiments on Intel_quad

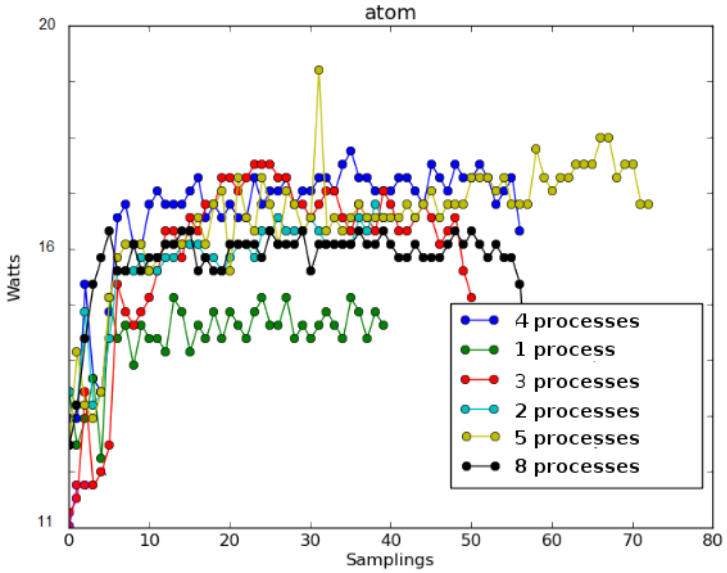

Figure 2.2: All stages of the CMSSW experiments on Intel_atom

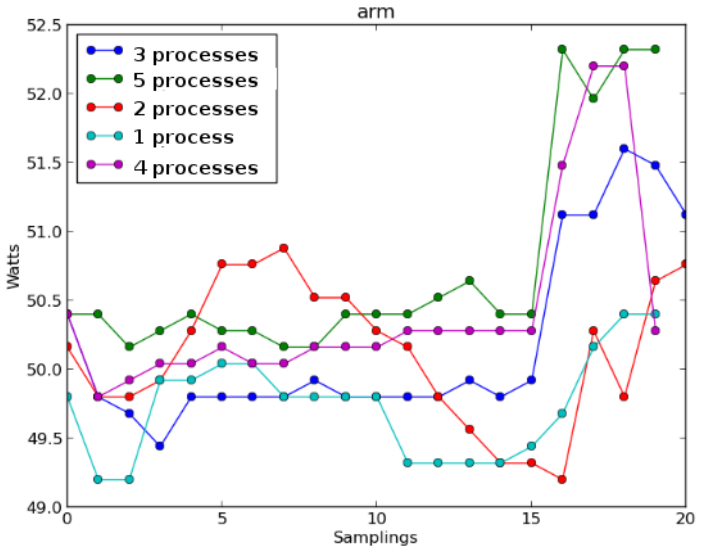

Figure 2.3: All stages of the CMSSW experiments on ARM_viridis

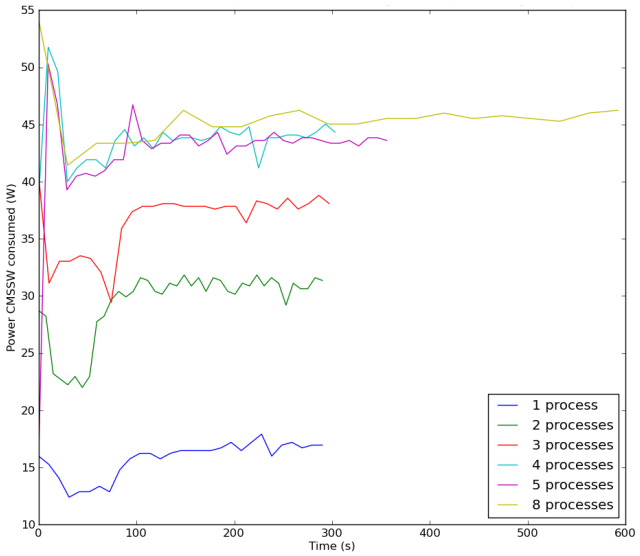

Figure 2.4: CMSSW experiments on Intel_quad - event processing stage

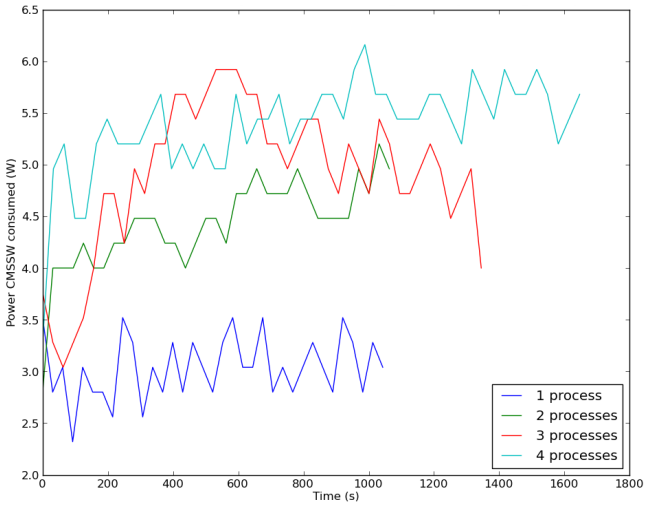

Figure 2.5: CMSSW experiments on Intel_atom - event processing stage

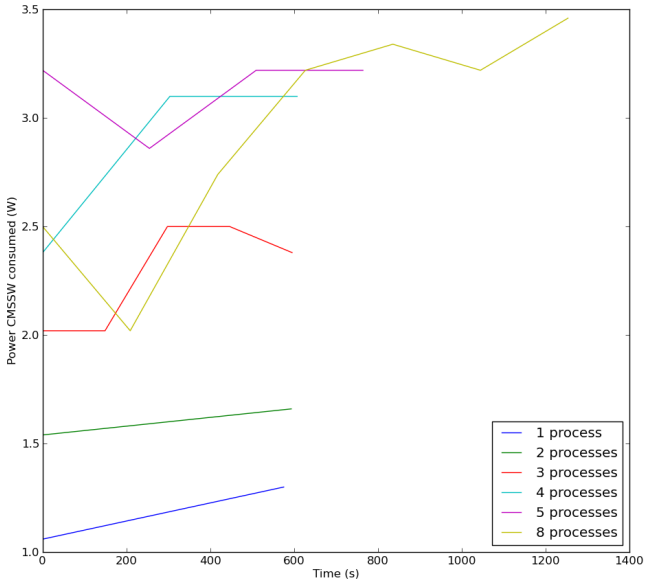

Figure 2.6: CMSSW experiments on ARM_viridis - event processing stage

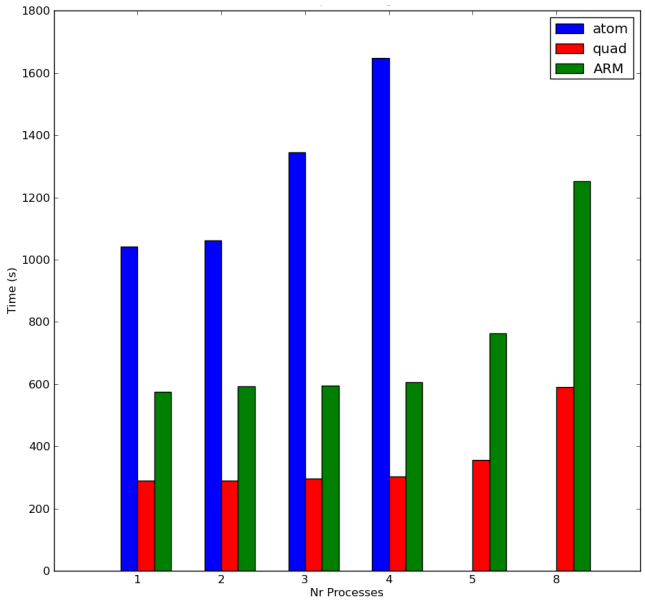

Figure 2.7: Processing time comparison

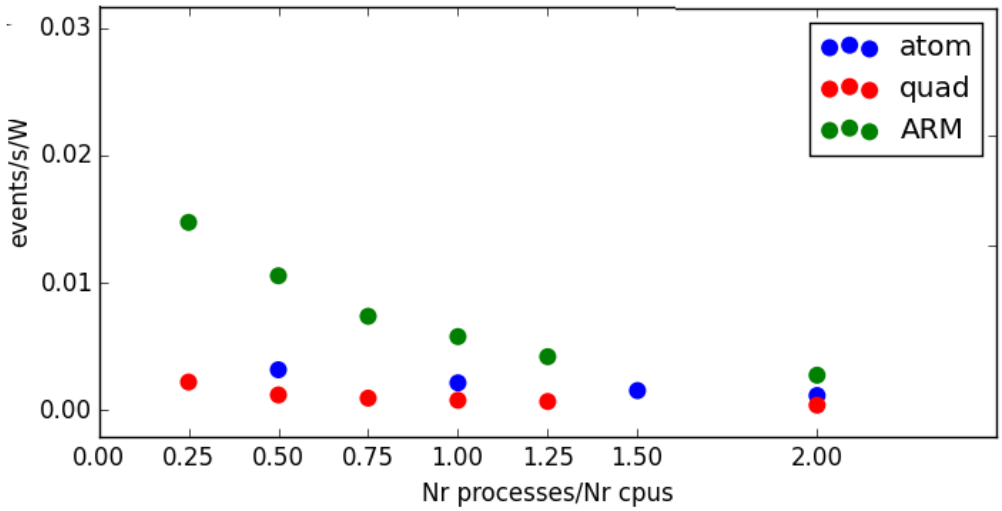

Figure 2.8: Energy efficiency comparison for the first set of experiments - External measurements

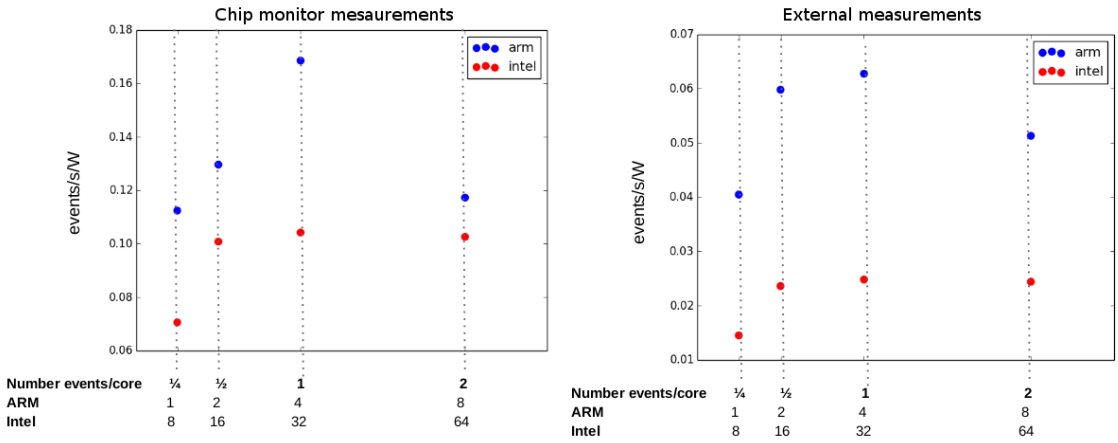

Figure 2.9: Multithreaded ParFullCMS comparison between Intel_xeon and ARM_odroid

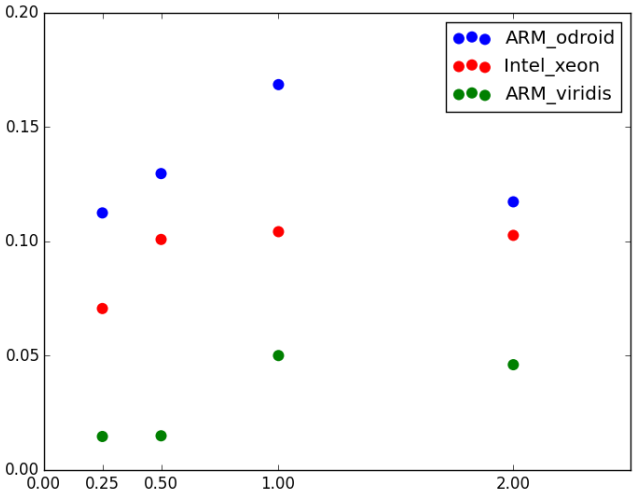

Figure 2.10: Multithreaded ParFullCMS comparison between Intel_xeon, ARM_viridis and ARM_odroid

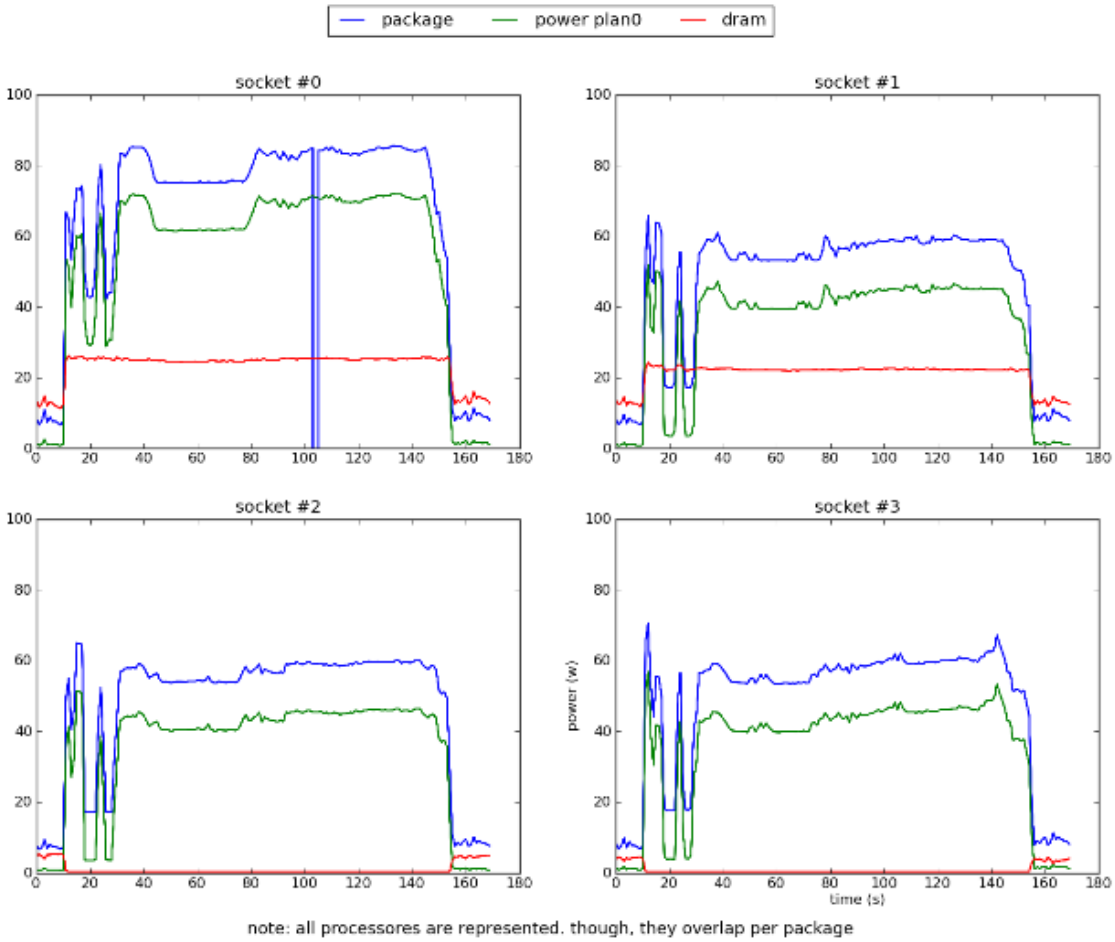

Figure 2.11: RAPL measurements of NUMA nodes - 16 processes with no explicit binding

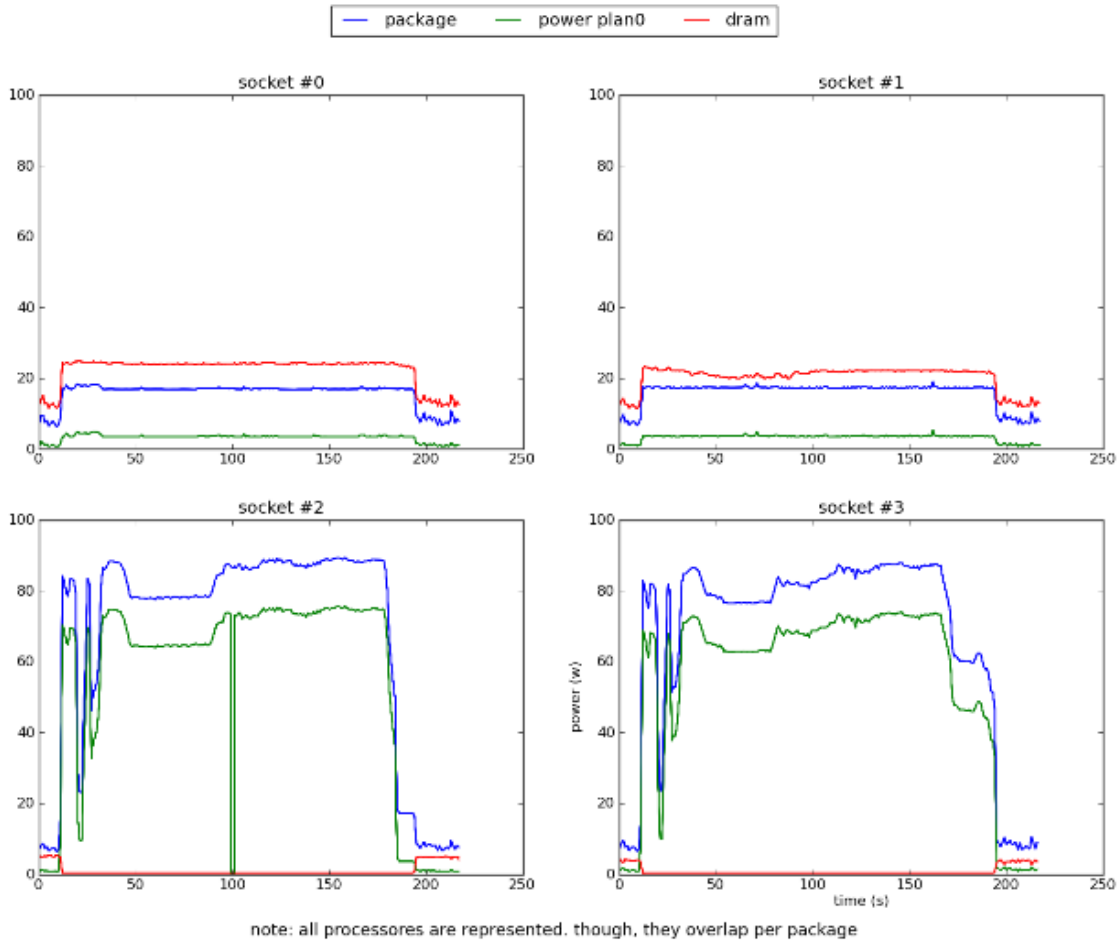

Figure 2.12: RAPL measurements of NUMA nodes - 16 processes. Explicit binding on node #2 and node #3 binding

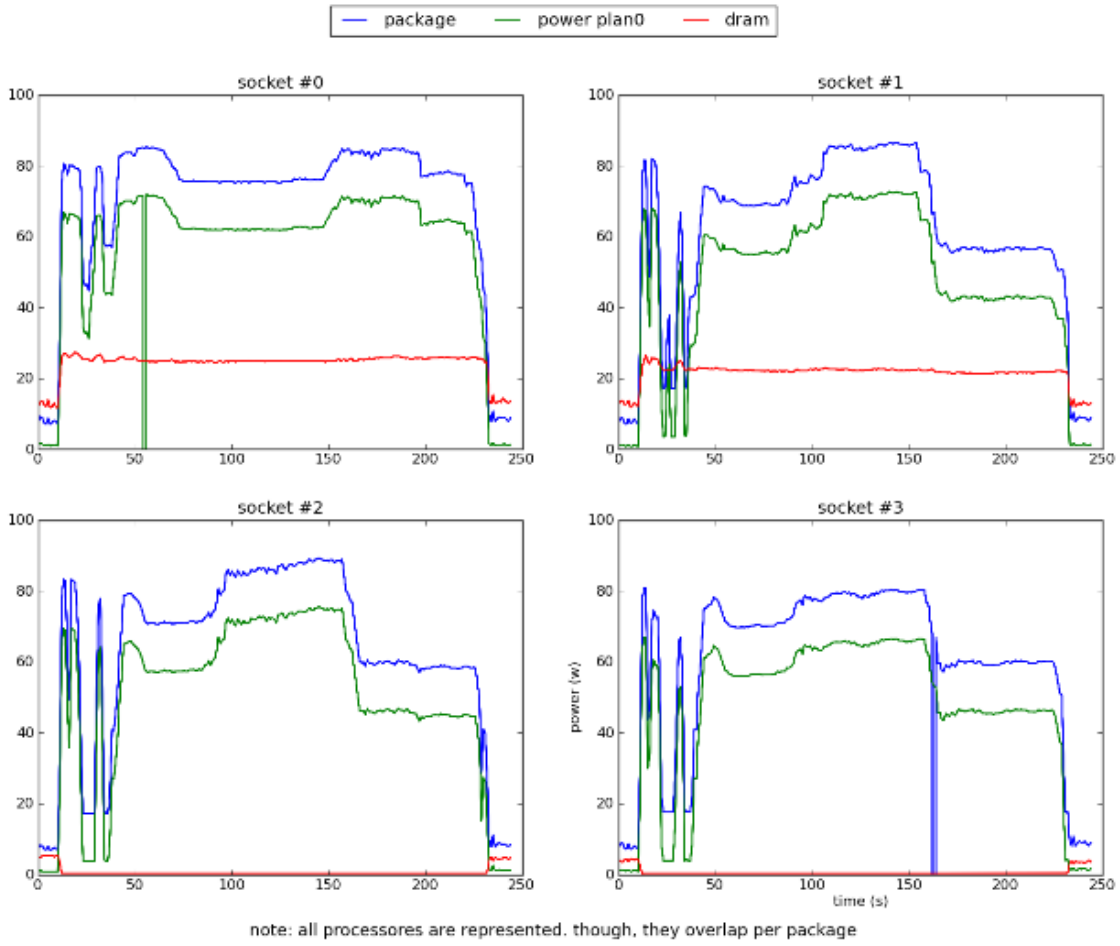

Figure 2.13: RAPL measurements of NUMA nodes - 32 processes with no explicit binding

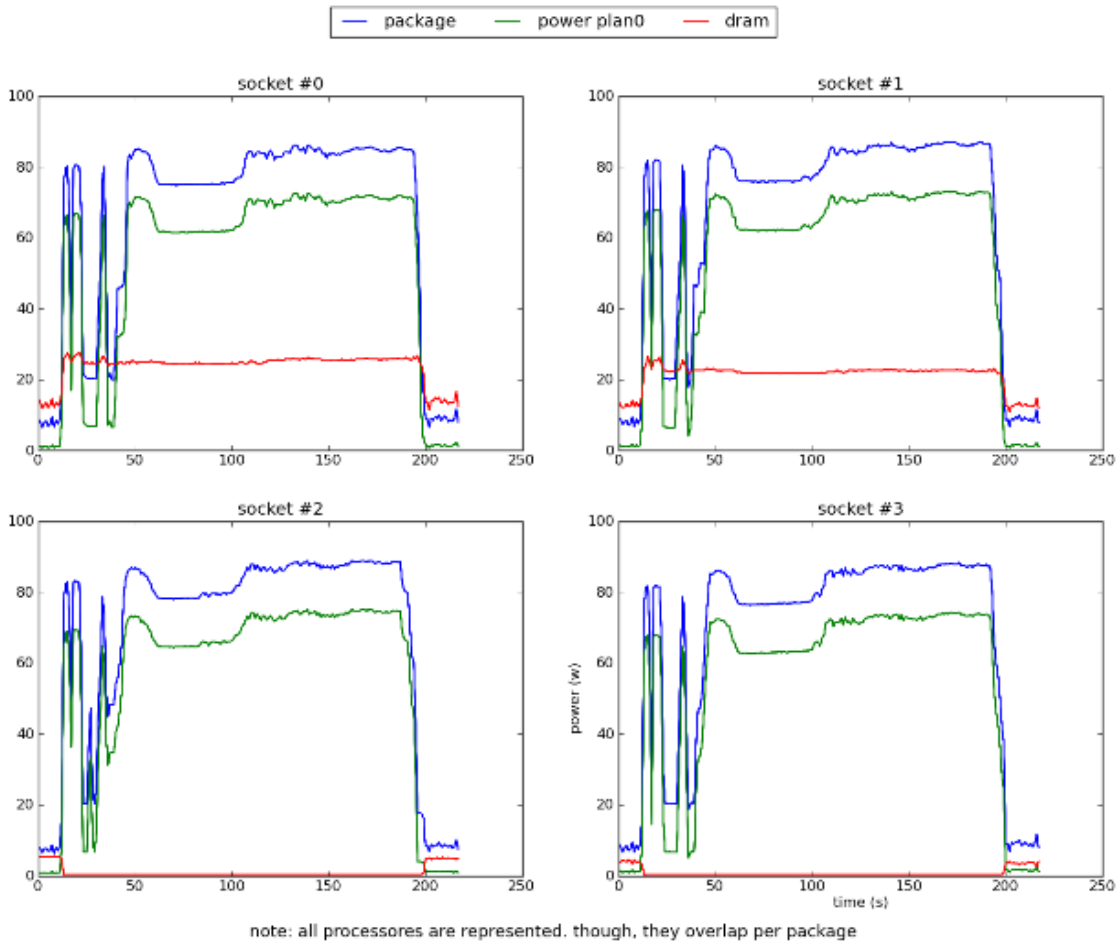

Figure 2.14: RAPL measurements of NUMA nodes - 32 processes. Processes distributed evenly explicitly - 8 processes per node.

Chapter 3

Analysis

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

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.

3.1 First Set of Experiments

In figures 2.1, 2.2 and 2.3 it is plotted the energy measurements from the beginning until the end of the generation-simulation CMSSW workflow. The energy measurements were done using a meter clamp. The energy measured is represented in the Y-axis and the X-axis represents the time of the experiment in samplings. For each experiment, a sample corresponds to the same time.

The figures 2.4, 2.5 and 2.6 we trimmed out the initialization stage and connection stage of the workload and only show the event processing stage. Whereas the Y-axi represents the energy measured in Watts, the X-axis represents the time spent until the correspondent energy sampling.

Finally, the figures 2.7 and **??** compare the time spent by each of the hardware setups and the power consumption efficiency of the different setups.

CMSSW stages

Based on the figures 2.1, 2.2 and 2.3, we can distinguish three different patterns of energy consumption during the expriement. We refer to each pattern as being part of a different CMSSW stage. The stages can be better identified when plotting the memory workload and the CPU usage(see Figures in GDrive-Add?).

The first stage consists is the initialization process. During this stage, the memory is the main module being used and thus, it is not an interesting stage to be analysed.

The second stage is the connection phase. The goal of this stage is to fetch the metadata from the CERN servers. The metadata is needed to perform the reconstruction of the events. Once again, during this stage the CPU load is low when compared to the memory workload.

The third stage corresponds to the event processing. This last stage is CPU intensive and it has the most relevant data to to our study, since we our goal is to compare the energy efficiency of the different CPUs. The event processing stage alone is represented by the figures 2.4, 2.5 and 2.6.

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

Relative importance of the stages

The most important stage when studying the energy efficiency of workload with the CMSSW is the last stage. There are three main reasons for that: Firstly, the CMSSW configuration at either CERN has caches that speed up considerably the second stage [refs], thus reducing the energy consumed in the connection stage. Secondly, the first and second stages are not CPU intensive. Lastly, the processing stage is the only one that the energy consumption is directly porpotional o the amount of events. Therefore, given any large amount of data to be processed, the last stage will consume so much more energy than the former stages that the first two stages will become irrelevant in terms of overall energy consumption. Therefore, we focus our energy consumption analysis on the event processing stage only. The event processing stage alone is represented by the figures 2.4, 2.5 and 2.6

Overcommiting CPU and energy efficiency

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

If we consider each hardware setup individually, the time needed for running the three stages of the experiment is roughly the same, if the CPU is not

overcommitted. When the number of processes exceed the number of available cores, the time to process the events increases since there are no available cores to process the events concurrently. In the overcommitted situation, the time increase follows the ratio *nr_of_processes/nr_of_cores_available*. For example, if the number of processes running is 6 and the number of cores available is 4, the time needed to process the events increases roughly 2/3 compared to when the CPU is not overcommitted.

In terms of energy consumed by the CPU, we do not find any outstanding difference in terms of overall energy efficiency by comparing CPUs that are overcommited vs non overcommited, as we can seen in the Figure 2.8. However, we expect that if the ratio *nr_of_processes/nr_of_cores_available* is large enough, it can affect negatively the energy performance given the energy overhead spent when the jobs are being swapped.

Time comparison

When comparing the time taken by the different architectures to process the same task (Figure 2.7), the pattern is evident. Regardless the number of processes, the Intel_quad architecture is faster than Intel_atom and ARM_viridis and ARM_viridis is faster than Intel_atom. This fact is due to the architectures characteristics and its specifications, most notably the CPU clock speed.

Energy efficiency comparison

Given the metrics used in this stufy (see Metrics section in the Experiments chapter), it is clear that systems are proportionally energy efficient with its ratio performance per watts. Therefore, by analyzing the Figure 2.8, 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 all the plot show spikes and rough transitions between samples. Moreover, the error tends to increase proportional to the human interaction with the experiment. Therefore, we conclude that it is more effec-

tive to use digital and automated ways to sample and log the data acquired during the measurements.

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

3.2 Second Set of Experiments

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

Energy efficiency comparison between Intel_xeon and ARM_odroid

In the Figure 2.9, we can see the energy efficiency comparison of Intel_xeon and ARM_odroid. The righmost plot represents the internal energy measurements, whereas the leftmost plot represents the external energy measurements. As in other energy efficiency comparisons in this study, we used the metrics *nr_of_events/s/W* to represent the energy performance of the measured systems.

The main conclusion from 2.9 is that ARM_odroid outperforms Intel_xeon in both internal energy efficiency and external energy efficiency.

Energy performance and overcommited CPUs

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

On the other hand, the energy performance of Intel_xeon does not seem to be significantly affected when overcommited. This phenomenon is explain

by the fact that Intel_xeon took roughly the same time to process the events when using one core per event and half a core per event.

We believe that the different results between ARM_odroid and Intel_xeon discussed below are due to the fact that ARM_odroid is a development board and it does not implement sophisticated techniques such as Hyper Threading Technology (HTT) by Intel [4]. According to Intel, HTT delivers two processing threads per physical core, which allows highly threaded applications to be processed faster. It is expected that if the ratio of *nr_of_threads/core* would be larger than 2, energy efficiency of Intel_xeon would start to decline.

Measurement tools and techniques

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

On the other hand, as stated in [5], the lower resolution that IPMI tools offers for internal measurements include energy consumed by the 0P9V, 1P8V, VDD and Vcore rails, which includes the system on the chip, DRAM, Temperature Sensors, and ComboPHY Clock. The components that are measured by the IPMI tools at each energy sample are shown in the Figure **??**.

As a result of this measurement discrepancy, 2.10 shows that ARM_viridis has worse energy efficiency than any of the other machines. We believe that this result can be misleading, due to the fact that the tools used to measure the energy consumed by each of the setups measure different components in the CPU. We believe that if the energy measured in the ARM_viridis is correspondent to many more components compared to the ARM_odroid and Intel_xeon measurements, we can not scientifically compare the results.

Comparison between First set of experiments and Second set of experiments

When we compare the main results of the 1SE (Figure 2.10) and 2SE (Figure 2.9) we may be inclined to conclude that the setups in the 2SE presented an overall more efficiency than the setups the 1SE. Again, the used measurement tools play an important role and should not be disregarded when analysing the results. In the 1SE, we only performed external measurements. Thus, we discard the possibility to compare the 1SE results with the results of the internal measurements of the 2SE. As for the external measurements performed in both set of experiments, the tools for measuring the energy consumption of both experiments have disctinct resolution and grain. In the

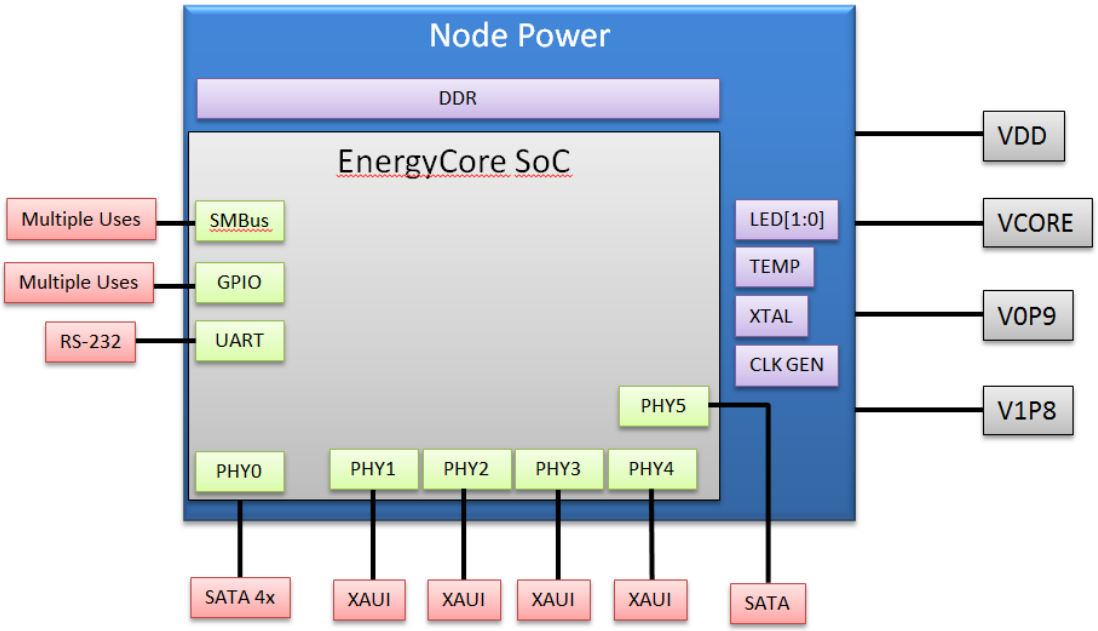

Figure 3.1: Representation of the Power Node measured by IPMI tools [make one myselfd and change!]

1SE, we used the clamp meter for measurements in all setups. As for the 2SE, we used embeeded and computer-assisted tools to perform the external measurements. This discrepancy of tools, its resolutions and errors, make it difficult to compare the results of the 1SE and 2SE.

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

3.3 RAPL in a NUMA environment

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

Should I include this in the thesis? I worked briefly on this at CERN and couldnt conclude anything because RAPL in the Intel_xeon was not 100 per cent compatible with the NUMA architeture used. Also, it goes a bit out of scope of the thesis. Maybe I should drop this part?

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

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

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

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

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

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

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

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

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

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

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

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.