

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 4, 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

<b>Author:</b>	Gonalo Marques Pestana		
<b>Title:</b>	Energy Efficiency in High Throughput Computing Tools, techniques and experiments		
<b>Date:</b>	1 December, 2014	<b>Pages:</b>	38
<b>Major:</b>	Data Communication Software	<b>Code:</b>	T-110
<b>Supervisors:</b>	Professor Jukka K. Nurminen		
<b>Advisor:</b>	Zhonghong Ou (Post-Doc.)		
abstract			
<b>Keywords:</b>	energy efficiency, scientific computing, ARM, Intel, RAPL, tools, techniques		
<b>Language:</b>	English		

# Acknowledgements

I wish to thank all students who use L<sup>A</sup>T<sub>E</sub>X for formatting their theses, because theses formatted with L<sup>A</sup>T<sub>E</sub>X are just so nice.

Thank you, and keep up the good work!

Espoo, 1 December, 2014

Gonalo 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 acronyms 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

<b>Abbreviations and Acronyms</b>	<b>4</b>
<b>1 Experiments</b>	<b>6</b>
1.1 Hardware . . . . .	7
1.1.1 ARM architecture . . . . .	7
1.1.1.1 Boston Viridis server . . . . .	7
1.1.1.2 ODROID-XU3 development board . . . . .	9
1.1.2 Intel x86 architecture . . . . .	10
1.2 Experiments setup . . . . .	12
1.2.1 First set of experiments . . . . .	12
1.2.2 Second set of experiments . . . . .	15
1.3 Summary . . . . .	18
<b>2 Results</b>	<b>19</b>
<b>3 Analysis</b>	<b>29</b>
3.1 First Set of Experiments . . . . .	29
3.2 Second Set of Experiments . . . . .	32
3.3 RAPL in a NUMA environment . . . . .	35
3.4 Conclusions . . . . .	35
<b>A First appendix</b>	<b>38</b>

# Chapter 1

## Experiments

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

The software used to run the computing tasks widely used in production and research at the CMS experiment. In order the results to be as realistic as possible, we used the CMSSW framework [ref] and ParFullCMS [ref] simulations, which are widely used in production at CERN.

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

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

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

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

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

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

## 1.1 Hardware

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

The ARM machines used were a single-board ARM processor developed by Odroid [11] and a server class ARM processor by Boston Viridis [12]. The Intel machines used were part of the microarchitectures family Sandy Bridges and Intel Bonnell. In the following sections, we will describe the hardware architecture, features of the hardware used to run the experiments and where the hardware is commonly used outside the scope of this study.

### 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 perform in a web server, cloud and data analytics environment with outstanding power performance [12].

The Boston Viridis server used in this study (which we will label as ARM\_viridis throughout the rest of the document) consists of a chassis with twelve racks, each with an energy card. Each energy card contains four nodes 1.1. A node is an ARM based CPU fabricated by Calxeda. The block diagram of a ARM\_viridis node is represented on 1.2 shows the architectures of the EnergyCore used. We can notice that the SoC has an energy management engine that will further on allow us to sample the energy consumed by the node.

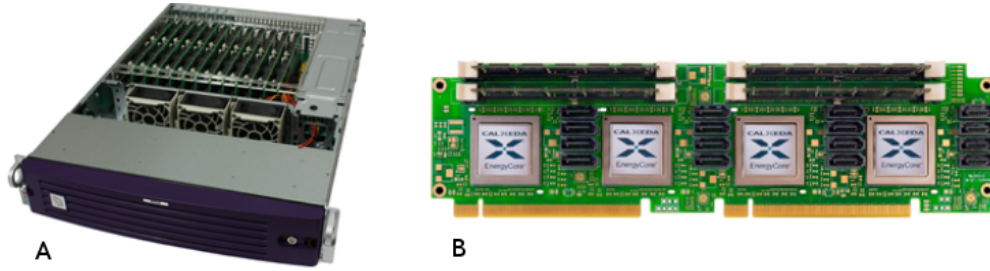


Figure 1.1: A. Viridis Server chassis with 12 energy card in it. B. Energy card with 4 nodes. Taken from [12]

Each ARM\_viridis node contains four ARM A9 Cortex core with a clock speed up to 1.4MHz. A memory controller and L2 cache is included on the chip. In addition, a couple of energy management blocks and IO controllers complete the Calxeda EnergyCore processor 1.2. These energy measurement blocks were used to perform part of the energy measurements with this setup.

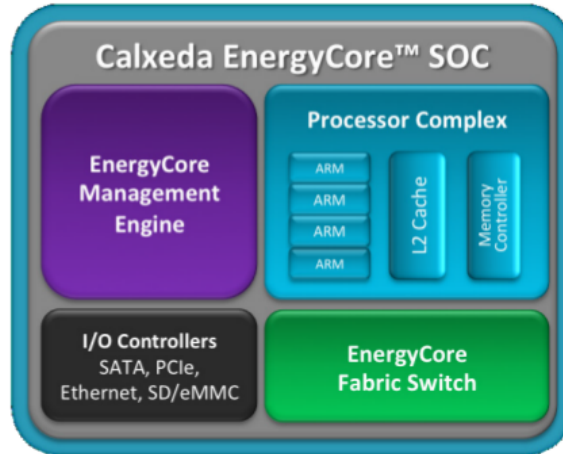


Figure 1.2: Block diagram of Calxeda EnergyCore. Taken from [12]

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].



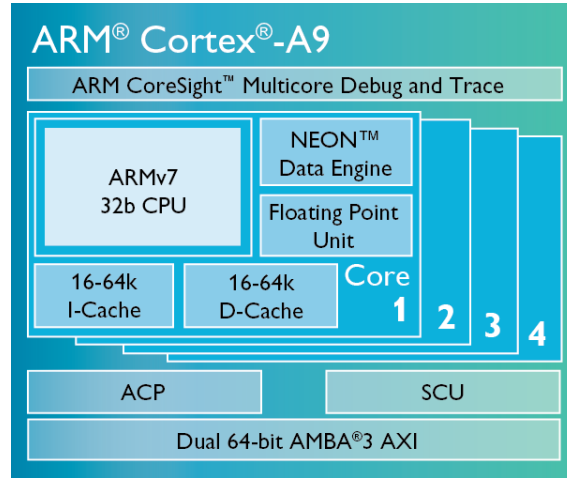


Figure 1.3: Block diagram of Cortex A9. Taken from [2]

#### 1.1.1.2 ODROID-XU3 development board

The ODROID-XU3 [11] is an open-source development board produced by Hardkernel. They claim that the ODROID-XU3 is a "new generation of computing device with more powerful, more energy efficient hardware and smaller form factor" [11]. At the time of these experiments, the ODROID-XU3 was mostly used for testing and platform development and it was not intended to run in production scenarios. Throughout this document, the ODROID-XU3 described in this section will be called ARM\_odroid.

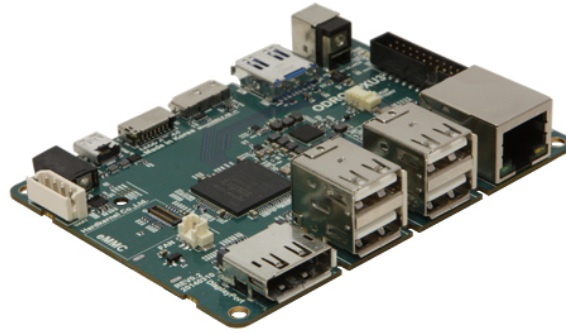


Figure 1.4: ODROID-XU3 development board. Taken from [11]

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 them 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 provides an API to read the energy consumed by the cores and DRAM at a sampling rate of microseconds. These readings can be easily triggered and read through software and consist of an accurate way to make fine-grained energy consumption measurements. We assumed that the measurements made by the TI INA231 can be compared to the RAPL technology by Intel.

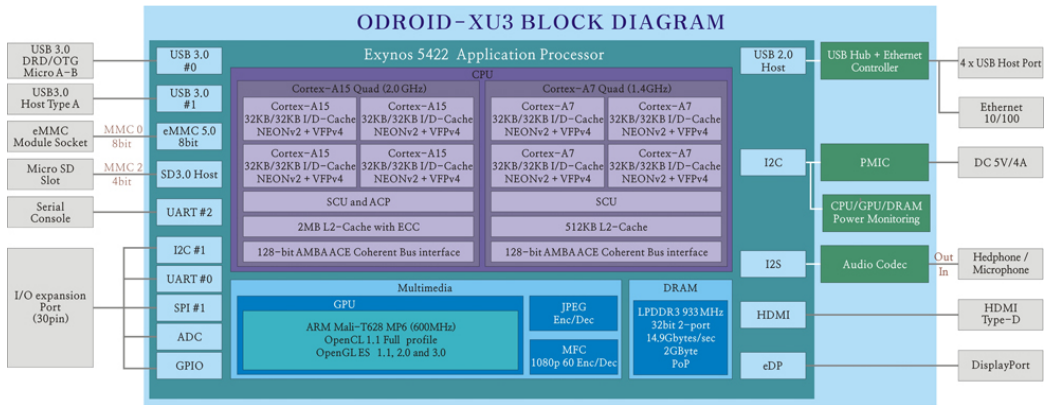


Figure 1.5: ODROID-XU3 block diagram. Taken from [11]

### 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 the final result are discussed in the Analysis section.

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

**have them been used in production**

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

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

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

#### Intel\_ATOM

**kernel & sys:** Linux cernvm 2.6.32431.5.1.el6.x86\_64

**OS:** Scientific Linux release 6.5 (Carbon)

**CPU:** 4x Intel<sup>TM</sup> Atom<sup>TM</sup> 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<sup>TM</sup> Core<sup>TM</sup>2 Quad CPU Q9400 2.66GHz

**Memory (MemTotal):** 7928892 kB (8GB)

For more detailed specs refer to [7]

#### ARM\_Viridis

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

**OS:** Fedora release 18 (Spherical Cow)

**CPU:** 4x Quad-Core ARM<sup>TM</sup> CortexA9<sup>TM</sup> processor 1.4GHz

**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 a Monte Carlo simulation of 8 TeV LHC Minimum bias event using Pynthia8 (generation step), followed by Simulation with Geant4 (simulation step). For more information about the CMSSW framework and its limitation on ARM, refer to Chapter X. At the time of the experiments, the CMSSW port for ARM had limitations on the multithreading support. We wanted to study the energy consumption of each hardware setup given different core load. Thus, we spinned up different processes instead of threads. The core-load levels used were 1/4, 1/2, 1 and 2 processes per number of physical cores.

### Metrics

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

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

### Tools for measuring energy consumption

For this set of experiments, we performed physical measurements using an external clamp meter. The clamp was a Mini AC/DC Clamp meter Mastech MS2102 AC/DC (see Figure 1.6). The clamp meter supports a maximum of 200A current, which was enough for our experiments. In addition, it presents an accuracy of  $\pm 2.5\%$ . 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

Machine codename	Architecture	CPU	N° active cores	RAM	Notes
<b>ARM_viridis</b>	Quad-Core ARM <sup>TM</sup> CortexA9 <sup>TM</sup>	ARMv7 32b (A7)	4	2 GB	Server class ARM processor with ipmitools
<b>Intel_ATOM</b>	Intel Bonnell <sup>TM</sup>	Atom D525	4	4GB	No internal measurement tool
<b>Intel_QUAD</b>	Intel Sandy Bridge <sup>TM</sup>	Quad CPU Q9400	4	8GB	No internal measurement tool

Table 1.1: Summary of the 1SE specifications

## 1.2.2 Second set of experiments

### Hardware specifications

For the second set of experiments, we again used three machines with different hardware setups. As in the 1SE, the three machines differ in architecture and general purpose. The ARM\_viridis, 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 performance scientific computation in a production scenario.

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

#### ARM\_Viridis

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

**OS:** Fedora release 18 (Spherical Cow)

**CPU:** 4x Quad-Core ARM<sup>TM</sup> CortexA9<sup>TM</sup> processor 1.4GHz

**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) - A7 at 1.4GHz and A15 at 2GHz

**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<sup>TM</sup> CPU E5-2650 2GHz

**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 experiments, we measured the energy consumed by the machine under different physical core loads. The core-load levels used were 1/4, 1/2, 1 and 2 threads per number of physical cores.

### **Metrics**

As in the 1SE, the energy efficiency metric used in this study is the ratio of performance per power consumed (Watts). The hardware setups used in the 2SE differ in specs and features. Therefore, we used the this metric as a way to uniform the results.

### **Tools for measuring energy consumption**

For the 2SE, we performed both internal and external measurements in the Intel\_xeon and ARM\_odroid. On the ARM\_viridis, we performed only internal measurements given the lack of a tool that would perform with the same degree of accuracy than the tools used for Intel\_xeon and ARM\_odroid. All the tools used to measure energy consumption were embedded in the hardware setup of the machines.

For the ARM\_odroid, we used a Texas Instrument power monitor chip (TI INA231) for internal measurements. The TI INA231 allowed us to sample the energy consumed by the cores and DRAM at a frequency rate of



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

microseconds. For the external measurements on the ARM\_odroid, we used an external plug-in power monitor with a computer interface for sampling and storing the results.

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

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

### 1.3 Summary

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

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

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

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

## Chapter 2

# Results

This chapter presents the results of the experiment described in the later chapter. The plots and figures in this chapter can be divided into 3 sections. The first section outlines the results of the first set of experiments. The second section outlines the results of the second set of experiments. The last section outlines the results that aim at studying the CMSSW framework in a Non-Uniform Memory Access (NUMA) environment.

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

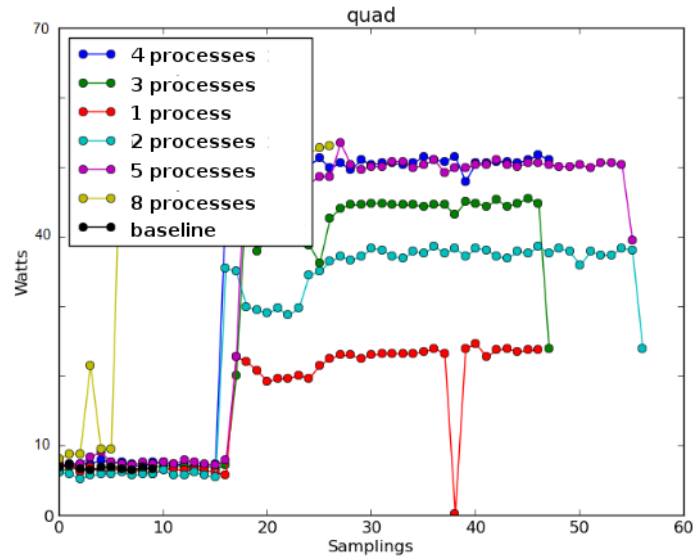


Figure 2.1: All stages of the CMSSW experiments on Intel\_quad

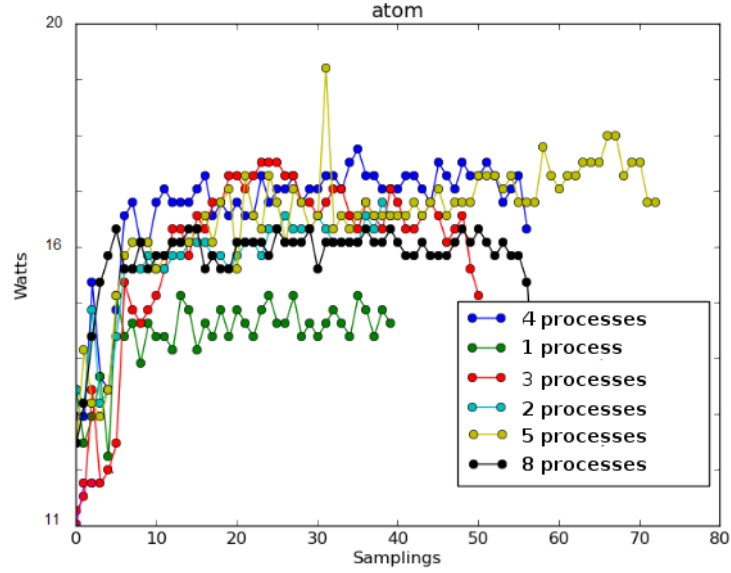


Figure 2.2: All stages of the CMSSW experiments on IntelAtom

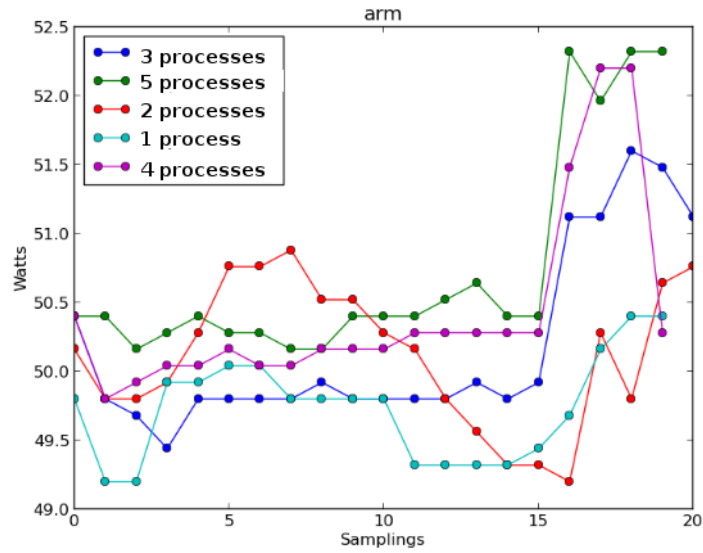


Figure 2.3: All stages of the CMSSW experiments on ARM\_viridis

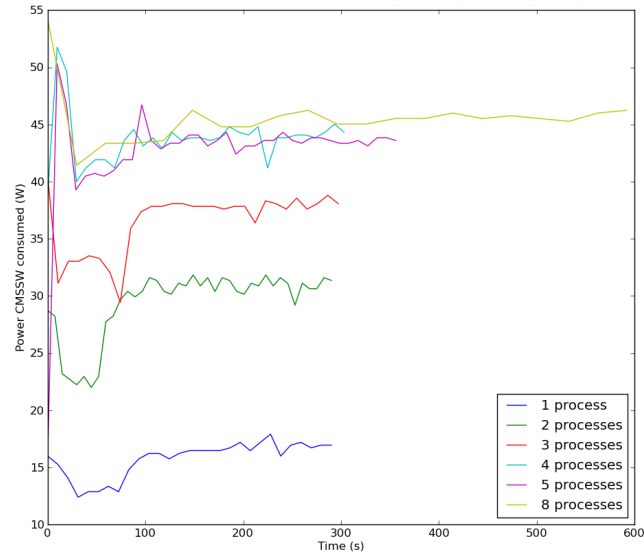


Figure 2.4: CMSSW experiments on Intel\_quad - event processing stage

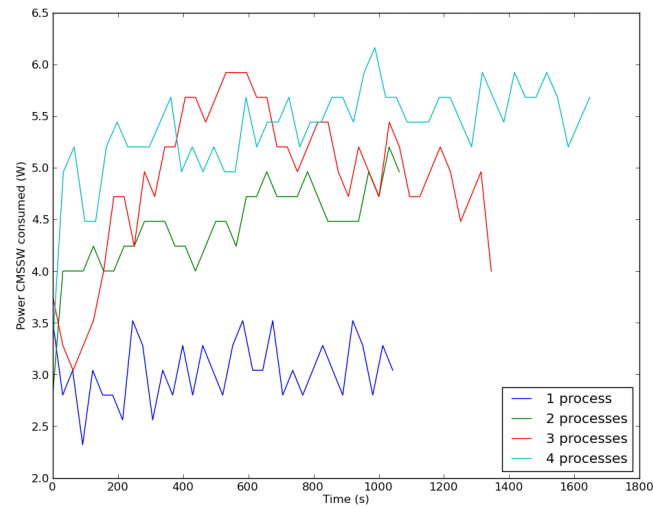


Figure 2.5: CMSSW experiments on Intel\_atom - event processing stage

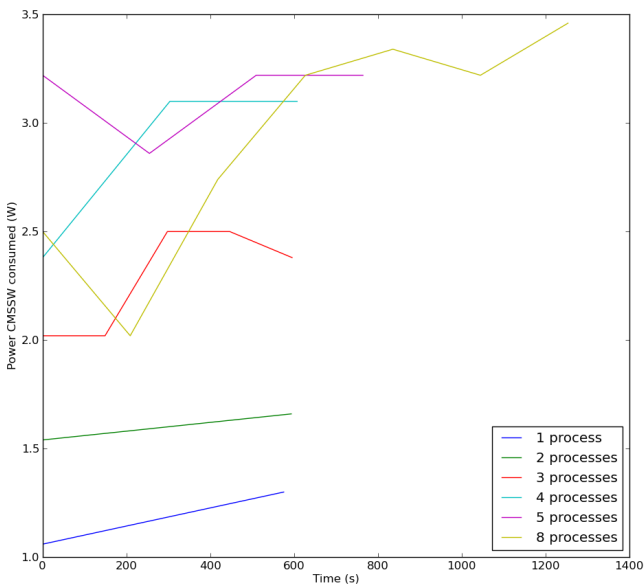


Figure 2.6: CMSSW experiments on ARM\_viridis - event processing stage

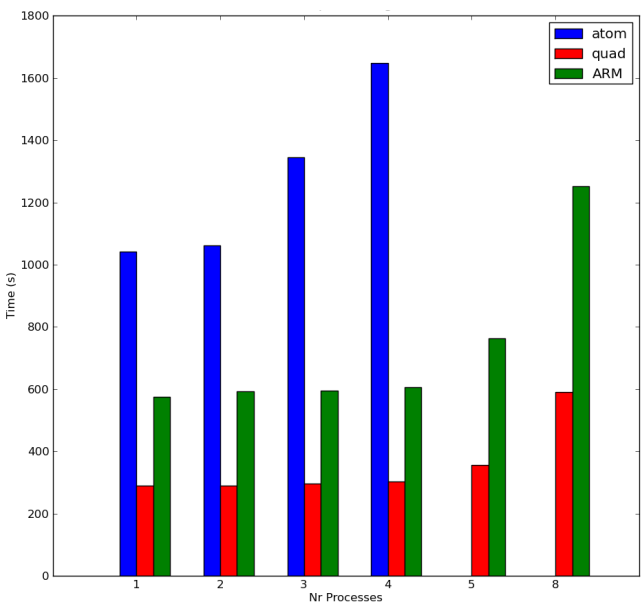


Figure 2.7: Processing time comparison

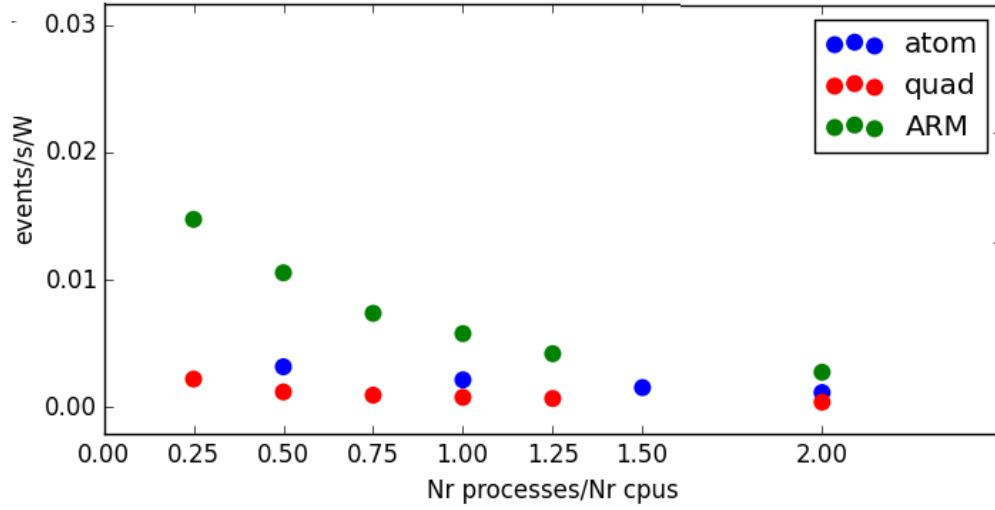


Figure 2.8: Energy efficiency comparison for the first set of experiments - External measurements

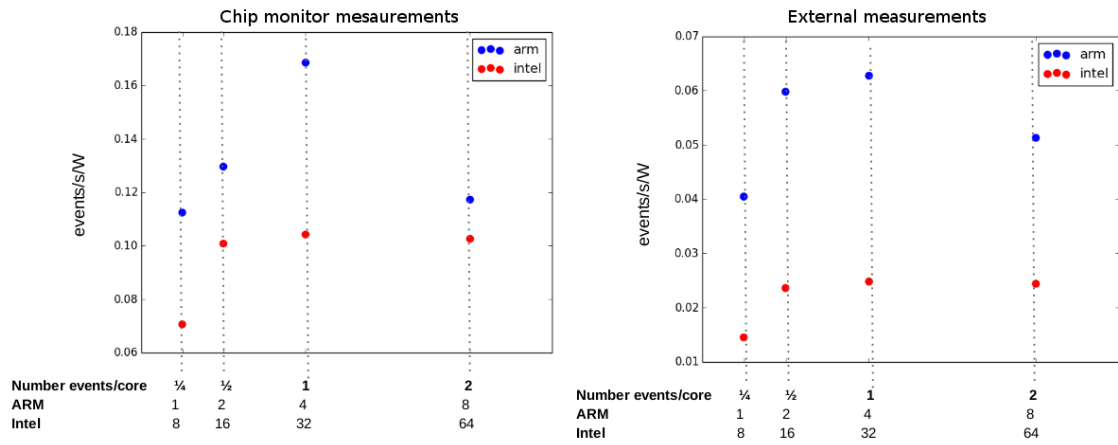


Figure 2.9: Multithreaded ParFullCMS comparison between Intel\_xeon and ARM\_odroid

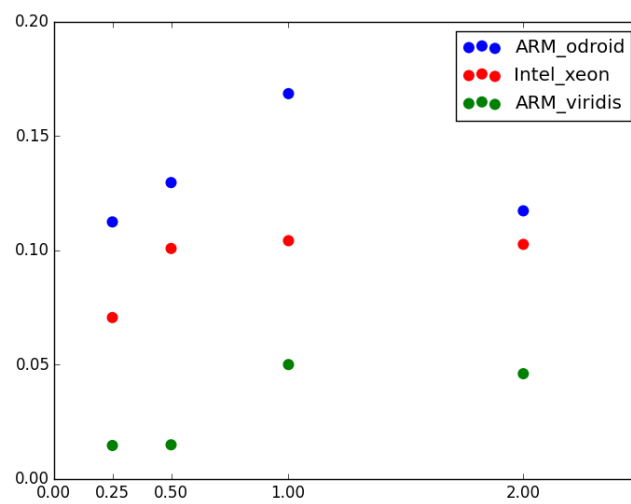


Figure 2.10: Multithreaded ParFullCMS comparison between Intel\_xeon, ARM\_viridis and ARM\_odroid



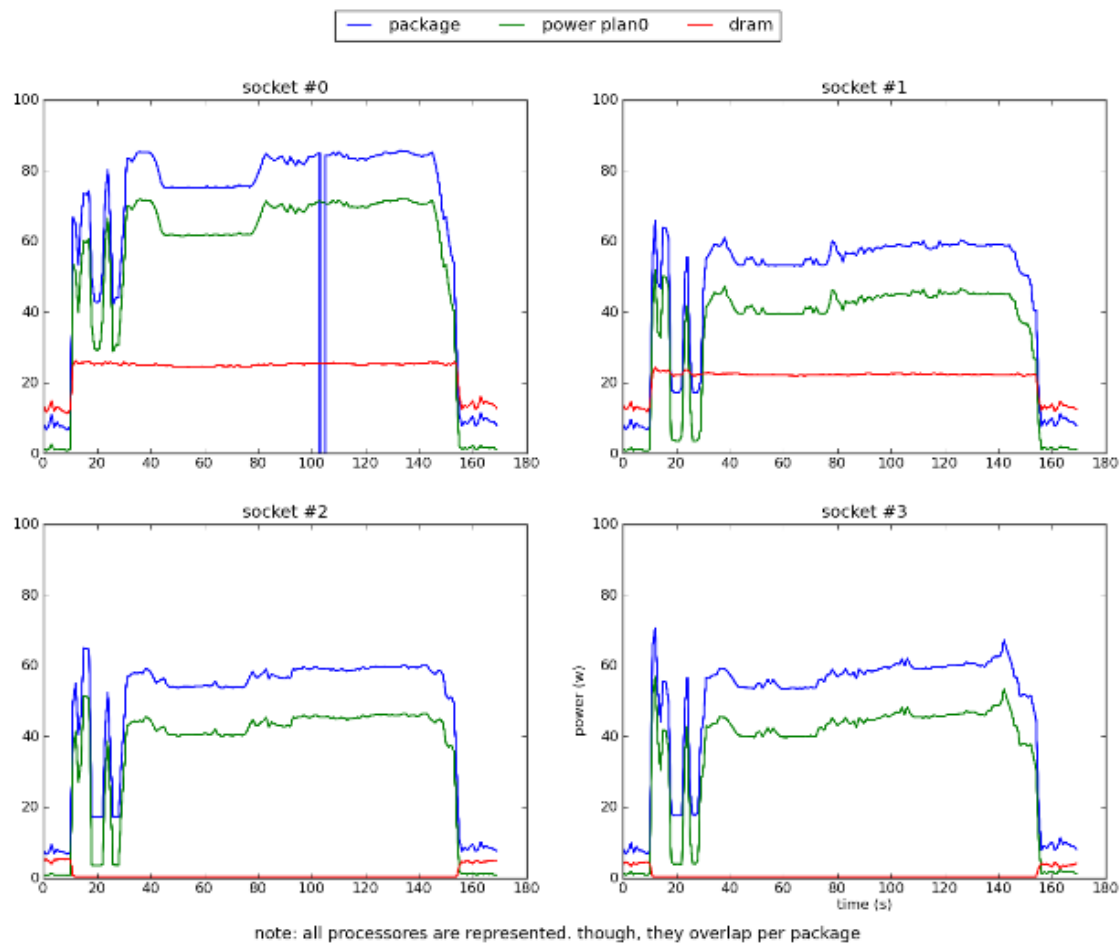


Figure 2.11: RAPL measurements of NUMA nodes - 16 processes with no explicit binding

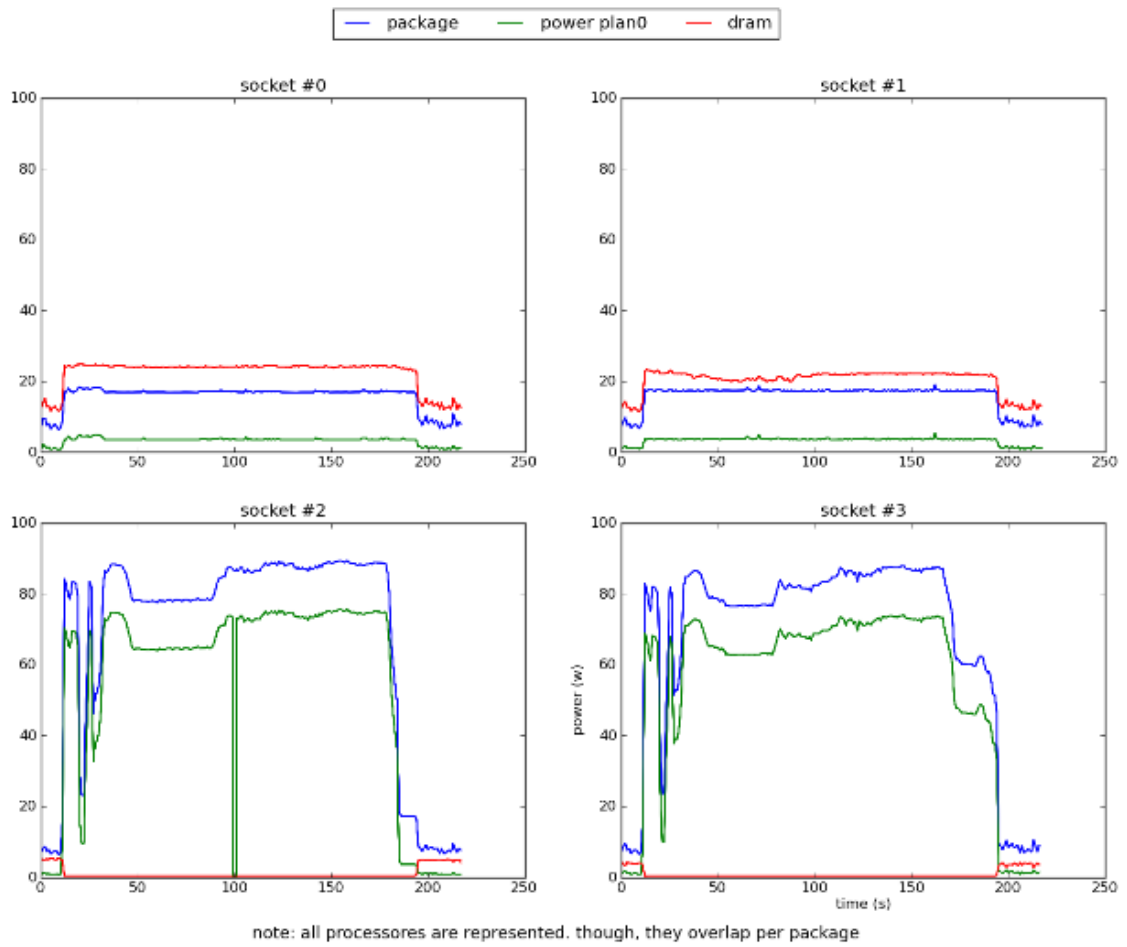


Figure 2.12: RAPL measurements of NUMA nodes - 16 processes. Explicit binding on node #2 and node #3 binding

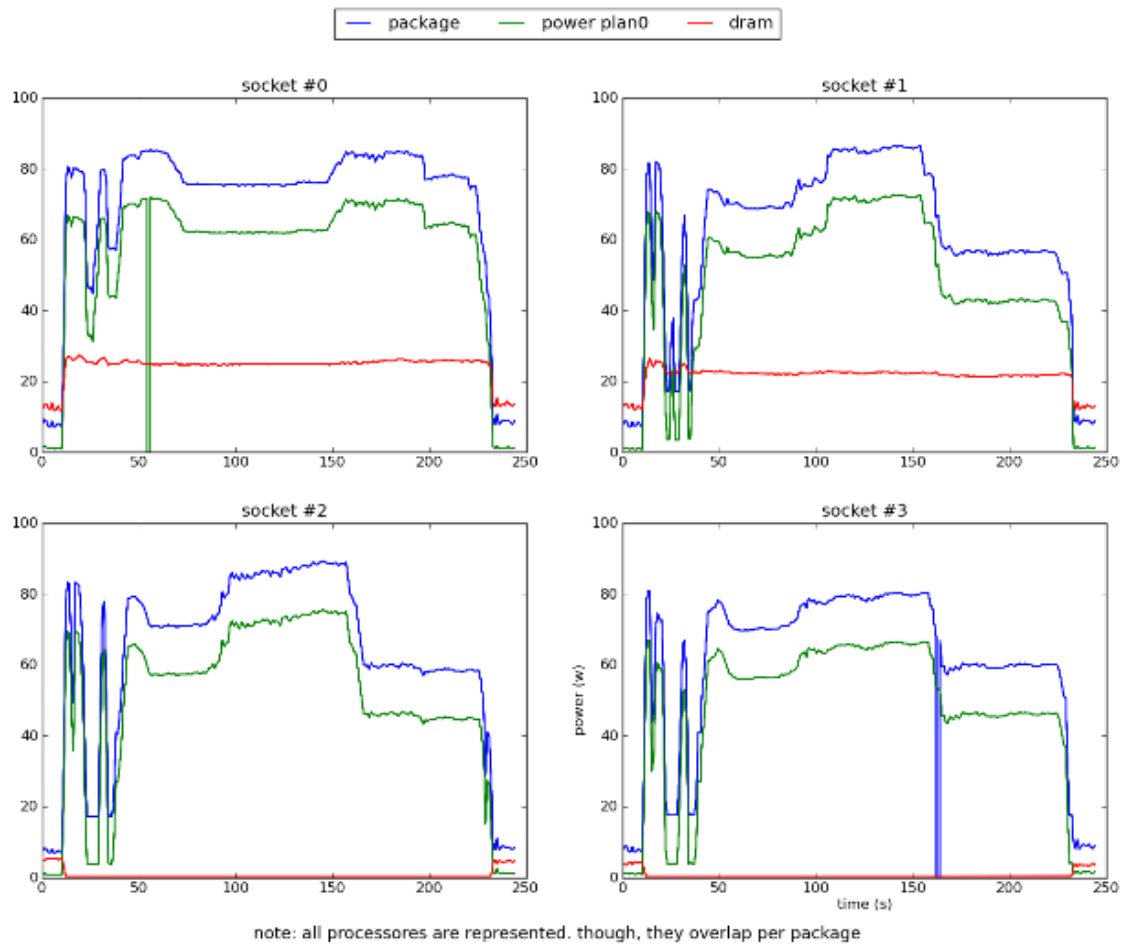


Figure 2.13: RAPL measurements of NUMA nodes - 32 processes with no explicit binding

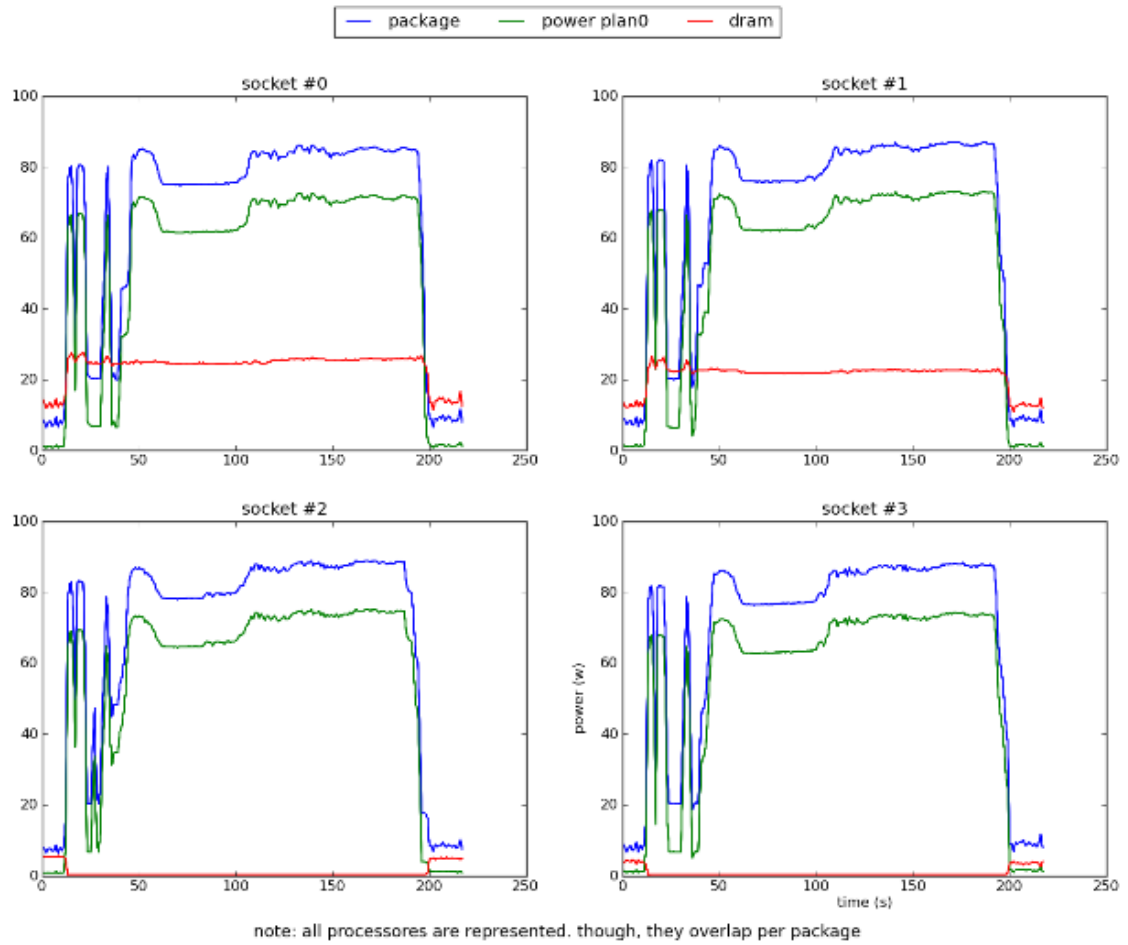


Figure 2.14: RAPL measurements of NUMA nodes - 32 processes. Processes distributed evenly explicitly - 8 processes per node.

## 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 and how the CMSSW framework performs in such conditions.

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 event generation-simulation by the CMSSW. The energy measurements were done using a meter clamp. The energy measured is represented in the Y-axis and the X-axis represents the time of the experiment in samplings. For each experiment, a sample corresponds to the same time.

In the figures 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-axis 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 to process the workflow and the power consumption efficiency of the different setups, respectively.

### CMSSW stages

Based on the figures 2.1, 2.2 and 2.3, we can distinguish three different patterns of energy consumption during the experiment. 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 of memory usage in GDrive - Add them or our of scope??).

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

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

The third stage corresponds to the event processing. This last stage is CPU intensive and it has the most relevant data to 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 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 proportional to 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

### Overcommitting CPU and energy efficiency

We consider a CPU to be overcommitted 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 overcommitted vs non overcommitted, as we can see 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 study (see Metrics section in the Experiments chapter), it is clear that systems are proportionally energy efficient with its ratio performance per watts. Therefore, by analyzing the Figure 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, Intel\_atom is more energy efficient than the Intel\_quad.

### Measuring tools: external monitoring

For this set of experiments, the external samples were acquired and recorded manually. This factor had a visible impact on the resolution of the measurements. Clearly, the all the plot show spikes and rough transitions between samples. Moreover, the error tends to increase proportional to the human

interaction with the experiment. Therefore, we conclude that it is more effective to use digital and automated ways to sample and log the data acquired during the measurements.

### Measuring tools: software-based monitoring

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

### 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 rightmost 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 overcommitted CPUs

It is noticeable that ARM\_odroid has a significant energy performance decline when its cores are overcommitted. It is also interesting to see that the energy performance decline in the ARM\_odroid is relatively larger on the internal energy measurements. One of the reasons we found in our raw results to explain this phenomenon is the large increase of time taken to process the events when the cores are overcommitted. 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 overcommitted. 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.

We believe that if we would overcommit Intel\_xeon with more than 4 threads per core, the time to process the workload would increase, which would be followed by a degradation of energy performance.

### Measurement tools and techniques

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

On the other hand, as stated in [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 performs worse than any other machine. We believe that this result can be misleading, due to the fact that the tools used to measure the energy consumed by each of the setups measure different components in the CPU. We believe that if components measured in the ARM\_viridis would be same as the components measured on the ARM\_odroid and Intel\_xeon measurements, we would obtain a different result. Namely that ARM\_viridis would, at least, perform better than Intel\_xeon from an energy consumption perspective. Given the actual setup, we can not scientifically directly compare the results.

### Comparison between First set of experiments and Second set of experiments

When we compare the main results of the 1SE (Figure 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

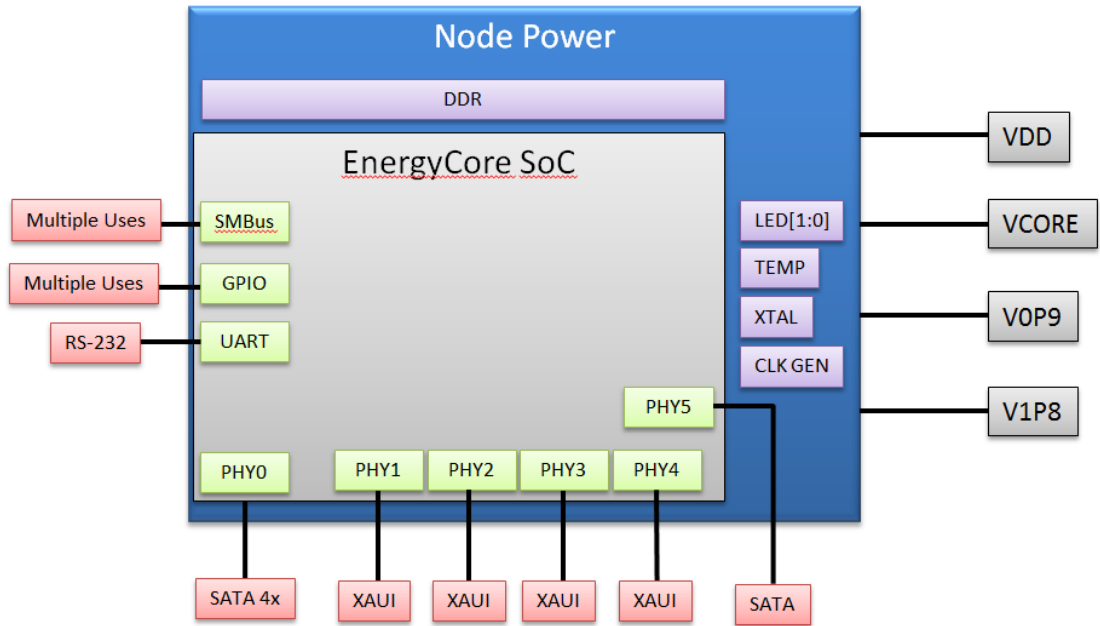


Figure 3.1: Representation of the Power Node measured by IPMI tools [make one myselfd and change!]

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 distinct resolution and grain. In the 1SE, we used the clamp meter for measurements in all setups. As for the 2SE, we used embedded 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 and every 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 architecture used. Also, it goes a bit out of scope of the thesis. Maybe I should drop this part?

## 3.4 Conclusions

The main conclusion of our experiments is that given the setups used in our study, ARM architecture outperforms Intel in terms of energy efficiency.

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

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

# 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](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](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](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](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.