

Energy Efficiency of a Low Power Hardware Cluster for High Performance Computing

Michael Dominik Görtz,¹ Roland Kühn,² Oliver Zietek,³ Roman Bernhard,³ Michael Bulinski,³ Dennis Duman,³ Benedikt Freisen,³ Uwe Jentsch,³ Tobias Klöppner,⁴ Dragana Popovic³ and Lili Xu³

Abstract: High performance computing has become more and more limited by the hardware's energy consumption, rendering it increasingly difficult to build even faster compute clusters, while modern low power hardware is making great improvements, regarding its performance. In order to overcome the limited energy density, we propose to use low power *System on a Chip* (SoC) devices, instead of high performance CPUs, exploiting the energy efficiency of modern low power processors. We evaluated our suggestion by building a low power cluster based on 40 single board computers with ARM Cortex-A53 quad-core SoCs and measuring its performance, energy consumption and efficiency using synthetic and application benchmarks with different workload types. Our tests demonstrated that our cluster could perform the given benchmarks, using up to 70% less energy than an Intel-based reference server system, which lead to an increase in efficiency of up to 425%. Our evaluation showed that modern low power processors have become a good alternative for high performance computing, with large workloads profiting from massive parallelization and that we can expect further improvements in this field, regarding the hardware's performance and efficiency.

Keywords: energy efficiency; low power hardware; high performance computing; ARM processors

1 Introduction

Current data clusters for scientific and commercial applications are dominated by high-performance hardware which delivers large amounts of compute power at the cost of a very high power consumption. While data centers have improved their energy efficiency and while new data center concepts enable the creation of compute clusters with a very high energy density, the energy consumption of traditional server hardware has stalled at a high level. Since the hardware's compute power is still improving, its energy efficiency is getting better as well, but the pace of those improvements has slowed down dramatically over the last few years.

On the other hand, low power processors, which were originally intended for the use in mobile and ultra-low power devices, have made huge improvements regarding their compute

¹ TU Dortmund University, dominik.goertz@tu-dortmund.de

² TU Dortmund University, roland.kuehn@tu-dortmund.de

³ TU Dortmund University, firstname.lastname@tu-dortmund.de

⁴ TU Dortmund University, tobias.kloppner@tu-dortmund.de

power. Modern low power processors like the latest ARM designs are very energy efficient and provide a lot of performance in relation to their low power consumption. Today, most of our mobile devices are small super computers on their own, providing more than enough compute power with multiple processor cores and sophisticated graphics processors, while still maintaining their low power consumption that is needed for mobile applications.

During the course of our master's degree's project group „Green Cluster Computing“, we designed and built a low power cluster for scientific applications and evaluated its energy efficiency in comparison to conventional server hardware. We wanted to evaluate if low power processors could replace high-performance hardware in certain use cases or if it might already be efficient enough to provide a full replacement for hardware with high energy consumption. Our main goal was to find out if a slower low power cluster is able to be more energy efficient than high-performance hardware.

2 Background

The use of high performance hardware comes with a lot of advantages. The latest processor architectures by the major chip manufacturers like Intel and IBM are very robust, offer a lot of compute power per core and are scalable to large clusters. The development of these high performance processors however has become more and more limited by their energy consumption and thermal problems [Wa16].

During the planning of compute clusters, the designers usually try to cram as much compute power into the available space. The denser the systems become, the harder it becomes to provide the processors with enough power and cool them at the same time, since the heat cannot be transferred away from the compute cores fast enough.

Another aspect of the hardware's high power consumption is the cost of the needed energy. Not only the compute hardware has to be provided with power but the air conditioning systems, which are needed to cool the processors, need power as well. While more traditional data center concepts used more than a third of their energy consumption for cooling, modern and specialized data centers improved their energy efficiency dramatically and lowered the energy required for cooling to under 15% of the total power consumption [PL14].

Despite the data centers' great improvements, the limitations by the power density still exist and every additional Watt of power consumed leads to additional requirements for the cooling solution.

A possible solution for these limitations might be the use of low power hardware. While this type of processors has been on the market for many years, it has only become popular in the last few years with the digitization of our everyday life. Mobile devices have only small power capacities and therefore must be fitted with very energy efficient processors. The commercial success of „smart devices“ led to great improvements of these low power

processors, which are able to provide respectable amounts of compute power in addition to their low power consumption.

While this energy efficient hardware is fitted to almost every device of our everyday life, it did not make its way into the field of high performance computing yet. We intend to close this gap by finding and evaluating possible use cases for low power hardware in a compute cluster.

3 Related Work

Even though the energy efficiency of low power hardware in high performance computing is mostly uncharted, there have been attempts to build clusters with this type of hardware in order to evaluate the capabilities of the new competitor. Over the last five years, multiple research teams built systems with different approaches and different use-cases.

In 2012 Ou et al. built a cluster out of PandaBoards, which were equipped with ARM Cortex-A9 MPCore dual-core processors [Ou12]. To evaluate performance and energy efficiency of the cluster they used web server, in-memory database and video transcoding applications and found out that the ARM processors were very efficient in less computation intensive applications and managed to perform their tasks up to 9.5 times more efficient than an Intel x86 platform. Computationally demanding workloads, however, were not the best application for the low power processors as their advantage shrunk to an efficiency ratio of 1.21 compared to the Intel system when transcoding videos.

In the same year, a European research team around Dominik G ddecke with members of the TU Dortmund Institute of Applied Mathematics built a cluster based on ARM Cortex-A9 processors as well [G 13]. They used 96 of these processors and concluded that the ARM processors can be more efficient than a traditional x86-based compute cluster, but it was limited by the hardware's 100MBit network interface. Additionally, heavily compute bound applications were proven to be a weakness of the low power hardware once again.

A year later, members of the same team took the concept of a HPC cluster based on ARM Cortex-A9 processors a step further and built a rack containing 128 processors which were integrated into NVIDIA Tegra SoCs [Ra14]. They benchmarked it using common benchmarks for raw computational power like micro benchmarks for double-precision floating points computation, Drystone and the SPEC CPU2006 benchmark. Their conclusion was that the cluster was 5 to 18 percent more efficient regarding its *energy to solution*, than the reference system, that was based on Intel Core i7 640M processors. They even tried to simulate the use of Cortex-A15 processors, which contain 16 cores each and estimated that these would increase their cluster's energy efficiency by 8.7x.

Picking up on the idea of using low power processors with larger amounts of cores, Michael Johan Kruger built a mini-cluster that consisted of four Paralela boards, which had 18 cores each [Kr15]. They were equipped with a dual-core ARM A9 CPU and a 16-core Epiphany

co-processor. He managed to achieve a performance similar to that of an Intel i5 3570, with only a third of the Intel processor’s power consumption. However, the Epiphany chip limited the systems performance, as it is not capable of complex arithmetic computations.

In general, the research in the field of high performance computing on low power hardware has shown that the used ARM processors were very energy efficient, but they lacked the computational power to achieve a performance, similar to that of high performance hardware. Our goal is to evaluate whether this lack of computational power has been compensated and whether it has increased the hardware’s energy efficiency even further.

4 System

4.1 Hardware Choice

The first objective was to choose an energy efficient single board computer that can provide a great amount of compute power. We considered both ARM- and x86-based platforms and ordered various single board computers to determine their energy consumption and computational power. MIPS-architecture-based processors were ignored due to the lack of suitable and available boards. We used the *Odroid Smart Power* by Hardkernel [Har17] to measure the energy consumption during idle and heavy load. The benchmarks for the load tests were taken from the Phoronix test suite [LT17].

Since the desired UDOO board was not available at that time, we ordered an ITX board with a similar SoC instead. Additionally, an Energenie EGM-PWM-LAN was used for the energy measurement of this board.

The examined boards and the corresponding power ranges are listed in table 1.

Board	RAM	CPU	Ethernet	Power
Odroid-C2	2GB DDR3	4 x 1.5 GHz ARM Cortex-A53	1GBit	2.0 - 4.2 W
Odroid-XU4	2GB LPDDR3	4 x 2 GHz ARM Cortex-A15 4 x 1.4 GHz ARM Cortex-A7	1GBit	3.8 - 10.5 W
Raspberry Pi 3	1GB LPDDR2	4 x 1.2 GHz ARM Cortex-A53	100MBit	1.3 - 4.2 W
ASRock J3160TM-ITX	4GB DDR3L	4 x 2.24 GHz Intel Celeron J3160	1GBit	10.6 - 16.0 W

Tab. 1: Examined boards

The benchmark results showed that the Celeron-based board and the Odroid-XU4 performed very well, but the energy consumption was significantly worse compared to the other boards. Considering the benchmark results and the peak power, the Odroid-C2 seemed to be the best suitable solution for our cluster [RB16]. Compared with the Odroid-XU4, it features the more modern ARMv8-architecture, which includes e.g. fully IEEE754-compatible double precision float SIMD-operations [St16] and in contrast to the Raspberry Pi, the Odroid-C2 has 2 GB of RAM and a much better network interface, which is a non-negligible aspect in a cluster.

4.2 Hardware Layout

We packaged the 40 boards of our cluster into a 2 RU 19" case, which we designed specifically for this purpose. It contains two MeanWell RSP-200-5 power supplies, which provide enough power to run at an utilization of under 50%, providing redundancy at the same time as a good efficiency level. Furthermore, the case is fitted with four distribution boards to provide power to the 40 SoCs and an additional Odroid C2, which is used for energy measurements.

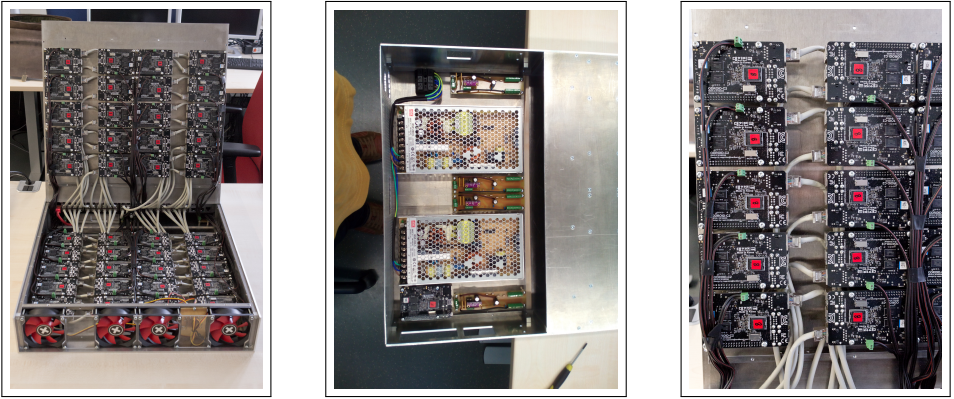


Fig. 1: Hardware Layout of the Low Power Cluster *Eriador*

For the network backbone the system is equipped with an HP 1920 48 Port Gigabit Switch, that is integrated into the case's structure.

5 Evaluation

5.1 Test Hardware

To evaluate the performance and energy efficiency of our cluster *Eriador*, we ran it against various synthetic and application benchmarks with different workloads. In our cluster, we used all 40 Odroid C2 boards, connected via the aforementioned HP 1920 48 Port Gigabit Switch. Every board is equipped with an ARM Cortex-A53 quad-core processor at a frequency of 1.5 GHz and 2 GB of RAM.

As a reference system, we used a high performance server system provided by the DBIS group of our compute science department. It is equipped with two modern Intel Xeon E5-2695 v2 processors from the Ivy Bridge EP family, at a frequency of 2.4 GHz and 256 GB of RAM, which makes it a good representative of the high-performance hardware that is deployed in HPC data centers right now.

5.2 Energy Measurements

For the energy measurements inside of *Eriador*, we used 4 measurement and distribution boards that we designed ourselves. These are equipped with two ACS758 Hall sensors [AM17] and ADS1115 I²C-AD-converters [TI17] and provide us with accurate measurements of the power consumption, within a measuring error of less than 10%. The power measurements of the 8 sensors are then accumulated to a total power consumption.

In order to measure the reference system's power consumption, we used a network-based EnerGenie EGM-PWM-LAN Energy Meter, that we monitored with a network script.

We used the Energy-Delay-Product (ED-Product), measured in Js, as a metric for energy efficiency of the system, as well as the performance per Watt, measured in FLOPS/W.

5.3 Heat Management

Even though each Odroid C2 board draws only a small amount of power, the total power consumption of the 40 boards leads to significant heat that has to be dissipated away from the processors. For cooling we integrated four 80mm fans with a speed of 1500rpm to push enough air through the system's chassis and keep the Odroid boards in an acceptable temperature range. We conducted measurements during stress tests with the whole system running at maximum load. The core temperatures of the boards do not exceed 76°C which is within the specifications of the Odroid C2 [RB16]. During normal workloads and benchmarking the core temperature stays within the recommended operating temperature of 70°C.

5.4 Performance Tests with Himeno

First, we conducted experiments with the Himeno Benchmark, in order to gain some knowledge about the raw compute power of *Eriador* and a rough understanding about its energy efficiency in relation to its performance. The benchmark was developed by Dr. Ryutaro Himeno at the RIKEN Advanced Center for Computing and Communication and is built on a Poisson Equation Solver that uses the Jacobi iteration method [Adv17]. It measures a systems floating point compute power in FLOPS. There are different data set sizes available consisting of different sized data matrices. Table 2 contains the four data set sizes with their data matrix dimensions.

S:	128 x 64 x 64	L:	512 x 256 x 256
M:	256 x 128 x 128	XL:	1024 x 512 x 512

Tab. 2: Data Matrix Dimensions for the Himeno Benchmark Poisson Equation Solver

We ran the benchmark with the smallest data set size and multiple partition configurations and discovered, that the energy consumption had deviations of up to 40% between measurements.

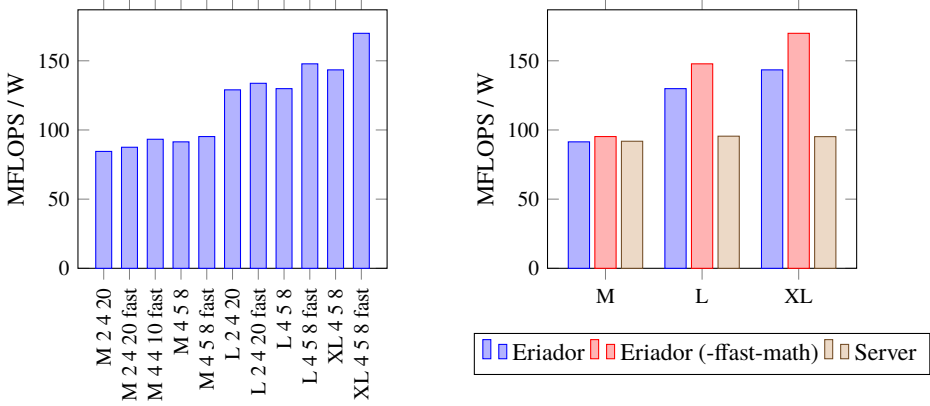


Fig. 2: Energy efficiency of *Eriador* with different Himeno benchmark configurations (left) and in comparison with the reference system (right)

These deviations occurred for our cluster, as well as for the reference system, since the data management and job distribution overhead is bigger, than the actual computation load. With a data set size of M, L and XL the deviations between measurements decreased to less than 2% and therefore the results are usable.

However, even with a sufficiently big data set, the results are still dependent on the used benchmark configuration. The way the benchmark's data matrix is partitioned, influences the performance and energy consumption and therefore the energy efficiency as well. To evaluate this dependency, we ran both *Eriador* and the reference system against different configurations and calculated their energy efficiency. As displayed in Fig. 2, our cluster reaches its highest efficiency with the biggest data set. While GCC's code generator for Intel processors is already very mature and provides very good performance on the -O3 optimization level, it does not profit from the -ffast-math option significantly. In contrast, the back-end for the ARM architecture is still in rapid development and in order to evaluate future optimization capabilities, we activated the GCC compiler option -ffast-math and added the options -march=armv8-a and -mtune=cortex-a53, to improve the compiler's code optimization results.

For the comparison between the Intel system and our ARM cluster, we used the best configurations in each data set size per system and additionally included the result for our cluster, with activated -ffast-math option, to show its potential with activated auto vectorization. The results for the best configurations with a XL sized data set are displayed in Tab. 3. While *Eriador* only has about half the MFLOPS the Intel server can provide, it uses just a little over a third of the energy during a benchmark run and therefore has a significantly higher MFLOPS/W ratio.

Considering all three data set sizes, the energy efficiency advantage of the ARM cluster

System	Threads	MFLOPS	Energy [KJ]	Time [s]	MFLOPS/W
Eriador	160	19,343.57	8.52	63.19	143.45
Eriador with -ffast-math	160	23,051.04	8.57	63.16	169.87
Server	48	41,990.85	22.32	50.6	95.18

Tab. 3: Performance, Execution Times and Energy Consumption for Himeno with XL data set

grows with a bigger data set size. Because the cluster is still bound by the data management overhead, both systems are more or less on par at the M data set size, while it can deliver a far better performance with bigger data sets. This leads to an efficiency advantage of 50% compared with the Intel reference system. Furthermore, the `-ffast-math` compiler option is able to provide an additional 18% efficiency increase.

5.5 NASA NAS Parallel Benchmarks

In order to test how much *Eriador* is able to profit from its massive parallelization capabilities we chose the well-respected NASA NAS Parallel Benchmark Suite[NAS17]. It is derived from computational fluid dynamics applications and has multiple options for small and large test problems. While the suite provides a large variety of different benchmarks we chose two multi-zone benchmarks for our testing. The BT-MZ and SP-MZ benchmark both make use of the different levels of parallelization, that our hardware provides. It can parallelize over all boards, as well as between the cores of each processor.

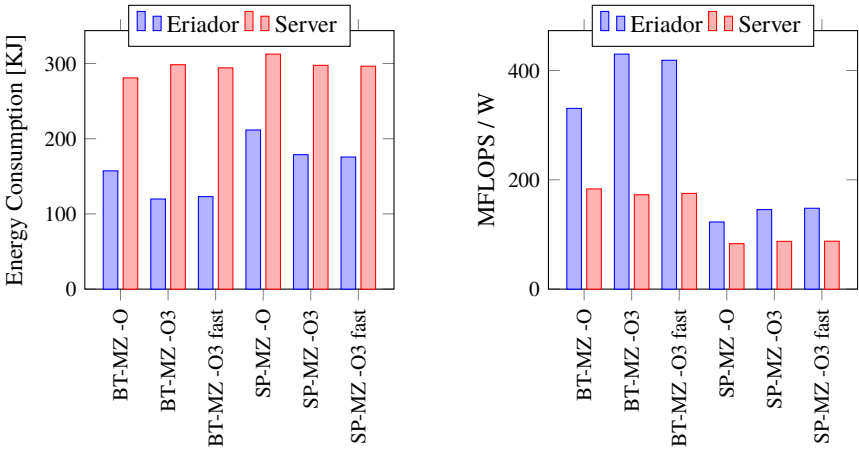


Fig. 3: Energy consumption (left) and efficiency comparison (right) with different compiler settings

The optimization settings used for compiling the benchmark can have a significant impact on the energy efficiency. While the reference system does not achieve a significant gain in performance and even loses performance with the `-O3` flag and `-ffast-math` option, going along with an increased energy consumption, *Eriador* is able to exploit the better code

optimization, which leads to a 30% gain in efficiency for the BT-MZ Benchmark and 18% more MFLOPS per Watt with the -O3 option. However, the -ffast-math option does not bring an additional increase in performance or efficiency for the cluster.

Benchm.	System	Optimization	MFLOPS	Energy [KJ]	Time [s]	MFLOPS/W
BT-MZ	Eriador	-O3	64,751.66	119.78	795.3	429.94
	Server	-O	76,084.89	280.9	676.83	183.33
SP-MZ	Eriador	-O3	23,391.96	178.72	1,111.11	145.43
	Server	-O3	39,481.15	297.53	658.31	87.35

Tab. 4: Performance, Execution Times and Energy Consumption for the NASA NAS Benchmark Suite

All in all, our cluster is able to achieve a very good efficiency with this benchmark suite, providing 2.3 times more MFLOPS per Watt in the BT-MZ benchmark, than the reference system. Even for the SP-MZ Benchmark, *Eriador* has a 66% better efficiency, compared to the server system, but execution times of the cluster are significantly longer, taking up to 70% more time to compute.

5.6 Classification with an MPI-based k-means Algorithm

In order to measure the efficiency in a more realistic setting compared to the aforementioned synthetic benchmarks, we used an MPI-based k-means implementation and compared the results with those from the reference server.

The k-means algorithm is used for data clustering tasks in the fields of signal processing and data mining [Na17]. The problem is to partition n objects into k data cluster while minimizing the deviation within each cluster. The idea behind the used k-means implementation is to define k cluster centroids and try to associate each object to one centroid, if no centroid fits the object the k centroids are moved. This application can be parallelized very well but depends on a central node, which fetches the results of all compute nodes after each iteration and computes new centroids for the next iteration.

For our measurements, we used a dataset with 1.59M samples each consisting of 20 features, with 5 features being redundant. The data set, with a size of roughly 780MB, was loaded off the same network storage by the cluster, as well as the reference server system.

System	Threads	Time [s]	Energy [KJ]	ED-Product [MJ]
Eriador	160	2,194.46	314.36	689.86
System	48	1,138.56	462.35	526.42

Tab. 5: Performance, Energy Consumption and Efficiency for k-means

Eriador struggles with the workload provided by the k-Means algorithm. During each iteration, the algorithm calculates new centroids, which is done on a single node and leads to an uneven hardware utilization. Fig. 4 contains a 30 second interval, taken from the power consumption measurements and shows the phases with more and less heavy compute load

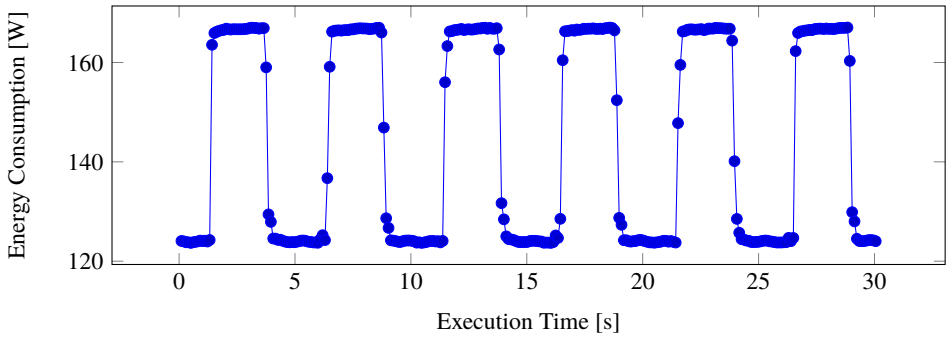


Fig. 4: Uneven Hardware Utilization caused by the centroid calculation phase of the k-Means-Algorithm

on the cluster nodes. The grade of partitioning needed to exploit all nodes of our cluster leads to an additional computation overhead of roughly 10% of the total computation time.

As displayed in Tab. 5, while the cluster’s overall energy consumption is more than 45% lower, than the energy consumption of the reference system, it can’t hide the fact, that the execution takes nearly twice as long. With an energy delay product of 690 MJs, *Eriador* is about 25% less efficient than the reference server, which takes 526 MJs to complete the task. Therefore, our cluster is far more efficient regarding the pure energy consumption, but lacks the power to be time and energy efficient at the same time.

5.7 Distributed Video Encoding

Another typical workload for compute clusters are video encoding tasks. By partitioning the input data and encoding it separately for each partition, the workload can be parallelized very well, which leads to a good scaling behavior with high node counts. For our tests, we used the *Distributed Video Encoder* (DVE) by Tessa Nordgren [No17] and encoded a video from the 31c3 conference. The 63min long and 742MB sized video file [Kr14] was encoded with a fixed video bitrate of 1000 kbit/s, in order to produce comparable and reproducible results.

System	Threads	Chunks	Time [s]	Energy [KJ]	ED-Product [MJs]
Eriador	160	40	341.32	42.25	14.42
		445	320.18	40.54	12.98
Server	48	1	491.7	142.32	69.98
Server with ffmpeg	48	1	416.67	133.01	55.42

Tab. 6: Performance, Energy Consumption and Efficiency for DVE

In addition to a partitioning with one chunk per Odroid board, we also chose the highest partitioning level available, which resulted in 445 chunks. This leads to a small increase in

performance, as well as efficiency for the cluster. The reference system, on the other hand, suffers from a higher chunk count and has its best results with only one chunk, effectively encoding the whole video at once, with all 48 processor threads. Encoding the video directly with `ffmpeg` did bring small improvements in efficiency and performance for the reference server. However, the *Eriador* does perform as well as expected. The nature of this highly parallelized workload suits the low power cluster very well and results in only a third of the reference systems energy consumption and a 4 times better ED-product. While *Eriador* is behind in performance for the other benchmarks, it even delivers a better performance than the reference system resulting in 30% faster execution times.

6 Conclusion and Outlook

With various synthetic and application benchmarks, we were able to prove, that low power hardware, represented by our ARM Cortex-A53-based compute cluster, has become a viable option for high performance computing. It can provide decent compute power and uses significantly less energy. The reduced power draw leads to great cost reductions for energy, as well as cooling and due to their low heat loss, the processors can be packaged very densely.

Considering the rapid development in the ARM processor community, we can expect even better results for low power systems in the future. But not only the hardware side is improving, the software is getting better as well. The higher compiler optimization levels and especially the experimental optimization options show, that there is a lot more potential in the hardware, than currently meets the eye. In the future we are going to upgrade *Eriador*'s cooling system with better fans and automatic temperature monitoring. Furthermore we are going to evaluate the cluster with a standalone version of the Hybrid Seeding Algorithm of the Large Hadron Collider beauty experiment [Qu17].

7 Acknowledgement

This work was supported by the DBIS Group of the Department of Computer Science at TU Dortmund University. The authors would like to thank the DBIS Group, especially Thomas Lindemann and Jens Teubner for their support and supervision.

References

- [Adv17] Advanced Center of Computing and Communication. Himeno benchmark, accessed May 12, 2017. <http://accr.riken.jp/en/supercom/himenobmt/>.
- [AM17] Allegro MicroSystems, LLC: . ACS758xCB: Datasheet, 2016 (accessed April 25, 2017). <http://www.allegromicro.com/~media/Files/Datasheets/ACS758-Datasheet.ashx>.

- [Gö13] Göddeke, Dominik; Komatitsch, Dimitri; Geveler, Markus; Ribbrock, Dirk; Rajovic, Nikola; Puzovic, Nikola; Ramirez, Alex: Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster. *Journal of Computational Physics*, 237:132–150, 2013.
- [Har17] Hardkernel co. Ltd. ODROID Smart Power, 2014 (accessed May 13, 2017). http://www.hardkernel.com/main/products/prdt_info.php?g_code=G137361754360.
- [Kr14] Kriesel, David: , Traue keinem Scan, den du nicht selbst gefälscht hast. MP4-Video, 2014. http://cdn.media.ccc.de/congress/2014/h264-hd/31c3-6558-de-en-Traue_keinem_Scan_den_du_nicht_selbst_gefaelscht_hast_hd.mp4.
- [Kr15] Kruger, Michael Johan: Building a Parallella board cluster. Bachelor of science honours thesis, Rhodes University, Grahamstown, South Africa, 2015.
- [LT17] Larabel, Michael; Tippet, M: . Phoronix test suite, 2011 (accessed May 13, 2017). <https://www.phoronix-test-suite.com>.
- [Na17] Naik, Azad: . k-means clustering algorithm, accessed May 12, 2017. <https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm>.
- [NAS17] NASA Advanced Supercomputing Division. NAS Parallel Benchmarks, accessed May 12, 2017. <https://www.nas.nasa.gov/publications/npb.html>.
- [No17] Nordgren, Tessa: . dve - the distributed video encoder, 2016 (accessed May 13, 2017). <https://github.com/nergdron/dve>.
- [Ou12] Ou, Zhonghong; Pang, Bo; Deng, Yang; Nurminen, Jukka K; Ylä-Jääski, Antti; Hui, Pan: Energy-and cost-efficiency analysis of arm-based clusters. In: *Cluster, Cloud and Grid Computing (CCGrid)*, 2012 12th IEEE/ACM International Symposium on. IEEE, pp. 115–123, 2012.
- [PL14] Phan, Long; Lin, Cheng-Xian: A multi-zone building energy simulation of a data center model with hot and cold aisles. *Energy and Buildings*, 77:364–376, 2014.
- [Qu17] Quagliani, Renato; Billoir, Pierre; Polci, Francesco; Amhis, Yasmine Sara: The Hybrid Seeding algorithm for a scintillating fibre tracker at LHCb upgrade: description and performance. Technical report, 2017.
- [Ra14] Rajovic, Nikola; Rico, Alejandro; Puzovic, Nikola; Adeniyi-Jones, Chris; Ramirez, Alex: Tibidabo: Making the case for an ARM-based HPC system. *Future Generation Computer Systems*, 36:322–334, 2014.
- [RB16] Roy, Rob; Bommakanti, Venkat: . ODROID-C2 Beginner’s Guide, 2016. <https://magazine.odroid.com/wp-content/uploads/odroid-c2-user-manual.pdf>.
- [St16] Stephens, N.: ARMv8-A Next-Generation Vector Architecture for HPC. *Hot Chips 28*, Cupertino, August 2016.
- [TI17] Texas Instruments, Incorporated: . ADS111x Ultra-Small, Low-Power, I 2 C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator, 2016 (accessed April 25, 2017). <http://www.ti.com/lit/ds/symlink/ads1113.pdf>.
- [Wa16] Waldrop, M Mitchell: The chips are down for Moore’s law. *Nature*, 530(7589):144–147, 2016.