

Performance and Energy evaluation of Spark applications on low-power SoCs

Christoforos Kachris

Institute of Communication and Computer Systems (ICCS/NTUA)

Athens, Greece

Ioannis Stamelos, Dimitrios Soudris

National Technical University of Athens (NTUA)

Athens, Greece

Abstract—Microservers have recently gained attention as low-cost, low power, reduced footprint servers that are mainly based on energy efficient processors such as the ones used in embedded systems. Microservers based on low-power embedded processors are mainly targeting lightweight applications or parallel applications that benefit most from individual low-power servers with sufficient I/O between nodes rather than high performance processors. In this paper we evaluate the mapping of Apache Spark on low-power SoC-based processors. Apache Spark is one of the most widely framework in cloud computing for batch and streaming data analytics. We evaluate the energy efficiency of low-power SoCs that are used in embedded system for the execution of several Spark applications. The performance evaluation shows that low-power SoCs have the potential to offer up to 3x higher energy efficiency compared to high performance processors typically used in data centers.

I. INTRODUCTION

Emerging web applications like cloud computing and big data analytics have increased significantly the workload on the data centers during the last years. In 2015, the total network traffic of the data centers was around 4.7 Exabytes and it is estimated that by the end of 2018 it will cross the 8.5 Exabytes mark, following a cumulative annual-growth rate (CAGR) of 33% (Figure 1, [1]). In response to this scaling in network traffic, data-center operators have resorted to utilizing more powerful servers. Relying on Moore’s law for the extra edge, CPU technologies have scaled in recent years through packing an increasing number of transistors on chip, leading to higher performance. However, on-chip clock frequencies were unable to follow this upward trend due to strict power-budget constraints. Thus, a few years ago a paradigm shift to multicore processors was adopted as an alternative solution for overcoming the problem. With multicore processors we could increase server performance without increasing their clock frequency. Unfortunately, this solution was also found not to scale well in the longer term. The performance gains achieved by adding more cores inside a CPU come at the cost of various, rapidly scaling complexities: inter-core communication, memory coherency and, most importantly, power consumption [2].

In the early technology nodes, going from one node to the next allowed for a nearly doubling of the transistor frequency, and, by reducing the voltage, power density remained nearly constant. With the end of Dennard’s scaling, going from one node to the next still increases the density of transistors,

but their maximum frequency is roughly the same and the voltage does not decrease accordingly. As a result, the power density increases now with every new technology node. The biggest challenge therefore now consists of reducing power consumption and energy dissipation per mm^2 . The failure of Dennard’s scaling, to which the shift to multicore chips is partially a response, may soon limit multicore scaling just as single-core scaling has been curtailed. This issue has been identified in the literature as the *dark silicon* era in which some of the areas in the chip are kept powered down in order to comply with thermal constraints [3].

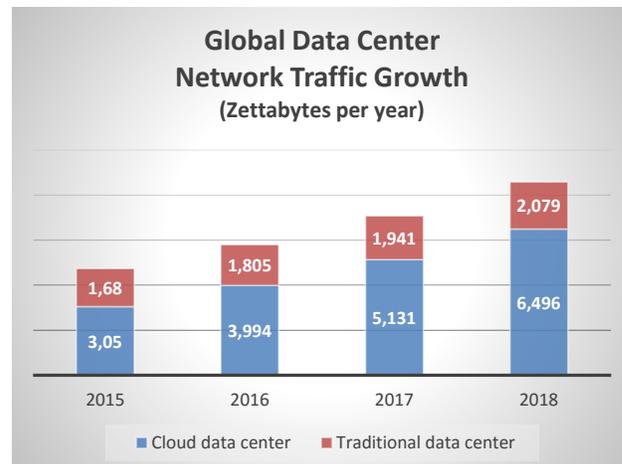


Fig. 1. Global Data Center Network Traffic Growth Projection, Source: Cisco Global Cloud Index 2016

One way to address this problem is through the utilization of microservers. Microservers have recently gained attention as low-cost, low power, reduced footprint servers that are mainly based on low-power energy-efficient SoC-based processors such as the ones used in embedded systems. Microservers are mainly targeting lightweight applications or parallel applications that benefit most from individual servers with sufficient I/O between nodes rather than high performance processors.

According to a new report by International Data Corp. (IDC), ARM-based server processors is expected to have a compound annual growth rate (CAGR) of 2.2% between 2015 and 2020 [4]. Based on this report, ARM processor vendors such as Applied Micro and Cavium have begun garnering notable design wins and partnerships from communications

service providers and systems vendors representing a wide spectrum of end customers and workloads. IDC predicts that the market for ARM and X86-based server processors will grow to reach \$15.3 billion in 2020. In 2016 IDC predicts that x86 and ARM server-class processor revenue will be about \$13.9 billion, up 1.3% from 2015.

In this paper we compare the energy efficiency of low-power 64-bit SoCs that are widely used in embedded applications with a high performance processor in terms of throughput, power and energy consumption for the the widely-used Spark framework.

The main contributions of this paper are the followings:

- An efficient mapping of Apache Spark on low-power embedded SoCs
- A thorough performance evaluation in terms of execution time, power and energy consumption
- A detailed analysis of the potential applications in Spark that could benefit most from low-power SoC-based processors.

II. RELATED WORK

In [5], a thorough study of power and performance measurements is performed for several BiG Data applications. In that study, an analysis is performed on system level and micro-architectural level, and a detailed characterization of several big data applications on big Xeon and little Atom-based server architecture. The characterization results across a wide range of real-world big data applications and various software stacks demonstrate how the choice of big vs little core-based server for energy-efficiency is significantly influenced by the size of data, performance constraints, and presence of accelerator. However, the paper is evaluating frameworks that are not based on Spark which is the most widely used in-memory data analytics framework. Furthermore, the paper compares the performance and the power consumption of Xeon and Atom processors that are both belong to the same x86 architecture.

In [6], they evaluate the performance and the energy efficiency of 64-bit eight-core ARM processors with the Intel Atom and Xeon processors. These platform are evaluated in terms of branch prediction performance, arithmetic performance, the memory system performance and the energy efficiency. For the performance evaluation they use several applications like memcached, FT and CG, but again in the applications it is not included the Spark programming framework.

In [7], [8], a performance evaluation study has been presented between high performance server cores (e.g. Intel Xeon processors) with low power general purpose cores (e.g. Intel Atom processors) specifically for web search applications. The comparison has shown that low power general purpose cores can achieve better energy efficiency in the domain of web search applications.

In [9], an analysis has been performed on the feasibility of building servers based on low power computers through a comparison of server applications running on x86 and ARM computer architectures. The comparison executed on web and

database servers includes power usage, CPU load, temperature, request latencies and the number of requests handled by each tested system. The use of ARM based systems has shown to be a good choice when power efficiency is needed without losing performance.

In [10], it is found that a cluster based on ARM processor had a power efficiency ranging from 1.3 to 6.2 times greater than that of Intel processor for the NAS benchmarks. In [11], they compare the performance and energy efficiency of high-performing processors (such as Intel Core i7 or E7) with energy efficient processors (such as Intel Atom, AMD Fusion or ARM Cortex A9). In this study it is also shown that for some HPC applications low-power processor offer better energy efficiency although the execution time is higher when compared to high performance processors.

In [12], low power processor and high performance processors are evaluated in three applications; Web server, in-memory database, and video transcoding. In the performance evaluation it is shown that the energy-efficiency ratio of the ARM cluster against the Intel workstation varies from 2.6-9.5 in in-memory database, to approximately 1.3 in Web server application, and 1.21 in video transcoding.

In [13], they investigate the performance and power measurements among various Hadoop configurations and system and architecture level parameters and how it affects the performance and the energy-efficiency across various Hadoop applications on Microservers.

Finally, in [14], it is presented a study of a Hadoop cluster for processing Big Data built atop 22 ARM boards. Experiments on three different hardware configurations were conducted to understand limitations and constraints of the cluster. From the experimental results, it was concluded that processing Big Data on an ARM cluster is highly feasible. The cluster could process a 34 GB Wikipedia article file in acceptable time, while generally consumed the power 0.061-0.322 kWh for all benchmarks. However, the paper does not present any comparison against high performance processor in terms of energy efficiency.

III. VERSATILE HETEROGENEOUS ENERGY-EFFICIENT DATA CENTERS: THE VINEYARD PROJECT

The study on energy-efficient systems as an alternative to high performance processors in data centers is part of the work in the VINEYARD project. The VINEYARD project, funded by EC-Horizon2020 program, has been formed in order to tackle the challenges on the high power consumption in the data centers. Future heterogeneous data centers consisting of different kinds of accelerators (FPGAs, GPUs, SoCs, etc.) will be able to provide higher performance under lower power consumption and higher energy-efficiency. However, to maintain in such heterogeneous systems the ease of programming of homogeneous ones, an integrated run-time scheduler and manager will be required to hide low-level details and relieve the user from the programming complexities involved (per different accelerator type). VINEYARD aims specifically at the automatic utilization of accelerators through developing

such an integrated framework that will control the hardware accelerators, while the user will still be allowed to use typical parallel programming frameworks.

VINEYARD aims to develop an energy-efficient integrated platform for data centers that will consist of energy-efficient servers based on customized hardware accelerators (i.e. FPGA-based and SoC-based servers) and a programming framework that will allow users to seamlessly utilize hardware accelerators in heterogeneous computing systems by using typical scalable cloud frameworks (i.e. Spark).

The VINEYARD project aims to develop novel energy-efficient servers based on programmable dataflow accelerators and integrated SoCs that can be customized based on the data-centers application requirements. These programmable dataflow accelerators will be used not only to increase the performance of servers but also to reduce the energy consumption in data centers. Furthermore, VINEYARD will develop a programming framework that will hide the complexity of programming heterogeneous systems while at the same time providing the optimized performance of customized and heterogeneous architectures. In this suite, the user works with familiar programming frameworks (i.e. Spark) while a run-time manager selects appropriate accelerators based on application requirements such as execution time and power consumption.

Finally, VINEYARD will provide the necessary middleware that binds together servers with accelerators. Along with this task, VINEYARD will consider both physical servers and virtual machines (VMs). The middleware also handles QoS concerns that arise with the shared use of the accelerators. In this way, the VINEYARD project will develop an integrated platform for heterogeneous accelerator-based servers. The VINEYARD platform will include both the hardware components (customized accelerators) and a software framework, which consists of two novel components: (i) a programming framework that integrates familiar programming models into heterogeneous systems, and (ii) a middleware layer that supports this heterogeneity in virtualized data centers.

Figure 2 depicts the high level diagram of the VINEYARD framework. Applications that are targeting heterogeneous data centers using traditional servers or micro-servers are programmed using traditional data center frameworks, such as Spark, and the widely used data management technologies, such as SQL (both OLTP and OLAP for operational databases and data warehouses), NoSQL (a key value data store), and Complex Event Processing (CEP). However, some of the tasks are common across several applications such as sorting of data, key/value processing, encryption, compression, pattern matching, etc. and are extremely computationally intensive. These tasks can be implemented in hardware as customized intellectual-property (IP) accelerators that can achieve much higher performance with lower power consumption. The implementation of the hardware accelerators can be achieved using traditional hardware description languages or other high-level (C, OpenCL) or domain-specific languages (i.e. OpenSPL). These hardware accelerators can be hosted in a

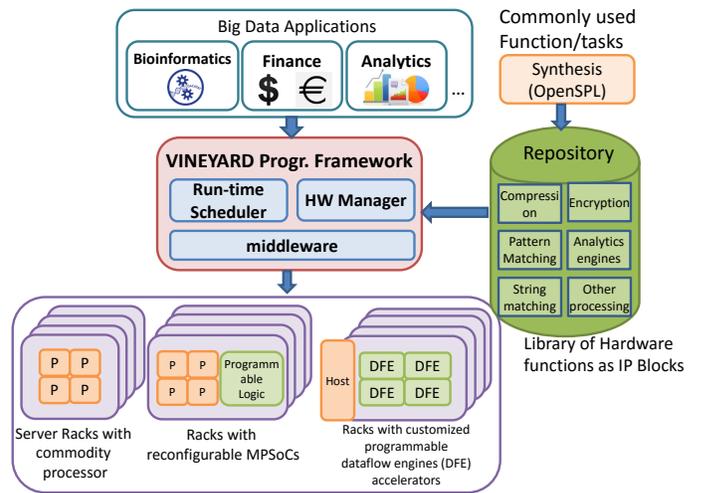


Fig. 2. High level block diagram of the VINEYARD integrated framework.

repository that can interface with the run-time scheduler.

The integrated data center that will be developed will be evaluated under three real-life workloads and industrial benchmarks for financial applications, data management, and bio-informatics. The first workload that will be evaluated will be in the domain of financial applications. For this reason the Greek Stock Exchange Market will be used as an end user demanding a) real-time analytics which are necessary for market surveillance and decision management and b) rapid computations for risk management, as an additional computation step within the trade process chain.

The second workload that will be evaluated will be in the domain of scientific computing, and more specifically in the domain of computational neuroscience which aims at better understanding the working of the human brain through use of mathematical models of biological neural networks. The particular application is a high-performance, high-accuracy simulation of the Olivocerebellar system of the brain, crucial to the understanding of cerebellar functionality. Better modeling and understanding of its function can lead to major breakthroughs in the treatment of cerebellum-related degenerative diseases (such as autism, fragile-X syndrome etc.).

The third workload will be based on data management based on TPC-C and TPC-H. TPC-C is representative of the transactional workloads run at operational databases of enterprises. It will be run on top of the LeanXcale OLTP database to represent the full stack of enterprise OLTP applications. TPC-H is representative of the analytical workloads run at data warehouses of enterprises. It will be run on top of the LeanXcale OLAP database to evaluate the efficiency improvements for analytical queries.

In this paper, we perform a first analysis of the potential energy saving that we could achieve in the data centers by deploying low-power embedded processors for the execution of the Spark framework.

TABLE I
MAIN FEATURES OF THE EVALUATED PROCESSORS

Features	Intel	Intel	Raspberry3	Snapdragon
Vendor	Intel	Intel	Broadcom	Qualcomm
Device	E5-2650	i5-430M	BCM2837	Snapdragon410
Cores (threads)	8(16)	2(4)	4	4
Processor	E5-2650	i5	A53	A53
Architecture	64-bit	64-bit	64-bit	64-bit
Architecture	CISC	CISC	RISC	RISC
Process	22nm	32nm	40nm	28nm
Clock Frequency	2.6 GHz	2.2 GHz	1.2 GHz	1.2 GHz
Level 1 cache	512 kB	128 kB	32 kB	32 kB
Level 2 cache	2048 kB	512 kB	512 kB	512 kB
TDP	95W	35W	4W	3.7W
Operating system	CentOS	Ubuntu	Debian	Debian

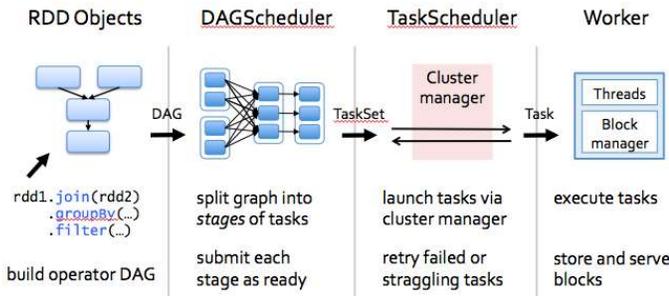


Fig. 3. The Spark framework

IV. SPARK FRAMEWORK

One of the typical applications that are hosted in cloud computing is data analytics. Apache Spark [15] is one of the most widely used frameworks for data analytics. Spark has been adopted widely in recent years for big data analysis by providing a fault-tolerant, scalable and easy to use in-memory abstraction. Specifically, Spark provides programmers with an application programming interface centered on a data structure called the resilient distributed dataset (RDD). RDD is a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way [16]. It was developed in response to limitations in the MapReduce cluster computing framework, which forces a particular linear dataflow structure on distributed programs. MapReduce programs read input data from disk, map a function across the data, reduce the results of the map, and store reduction results on disk. Spark’s RDDs function as a working set for distributed programs that offers restricted form of distributed shared memory. Therefore, the latency of such applications, compared to Apache Hadoop, may be reduced by several orders of magnitude.

When the user runs an *action* (like collect), a *Graph* is created and submitted to a DAG Scheduler. The DAG scheduler divides operator graph into (map and reduce) stages. A stage is comprised of tasks based on partitions of the input data. The DAG scheduler pipelines operators together to optimize the graph. The final result of a DAG scheduler is a set of stages. The stages are passed on to the Task Scheduler. The task scheduler launches tasks via cluster manager. The *Worker* then executes the tasks for the task processing as is depicted in Figure 3.

Spark libraries covers 4 main categories of applications: machine learning, graph computation, SQL query and streaming applications.

- Spark MLlib is Spark’s scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering,

collaborative filtering, dimensionality reduction, as well as underlying optimization primitives.

- GraphX is a Spark API (Application Programming Interface) for graphs and graph-parallel computation. At a high level, GraphX extends the Spark RDD by introducing the Resilient Distributed Property Graph: a directed multi-graph with properties attached to each vertex and edge. GraphX includes a growing collection of graph algorithms and builders to simplify graph analytics tasks.
- Spark SQL provides the capability to expose the Spark datasets over JDBC API and allow running the SQL like queries on Spark data using traditional business intelligence (BI) and visualization tools.
- Spark Streaming can be used for processing the real-time streaming data. This is based on micro batch style of computing and processing. It uses the DStream which is basically a series of RDDs, to process the real-time data.

V. LOW-POWER SOC

In this paper we compare the energy efficiency of a typical high-performance processor with two low-power embedded processors based on the ARM cores. Table I shows the main features of each platform. Both of the ARM-based processors host 4 64-bits ARM cores (A53). The Raspberry board is based on the Broadcom SoC (BCM2835) fabricated using 40nm process technology and clocked at 1.2GHz. The second board is from Snapdragon and is based on the 410 SoC from Qualcomm. This processor also host 4 64-bits ARM cores and is also clocked at 1.2GHz. However, this SoC is fabricated at 28nm process technology. As a reference platform, the Xeon E5-2650 processor has been selected with 8 cores clocked at 2.6GHz. Also in this paper, a commodity processor for laptops has been selected based on the Intel i5 processors to show how this platform is compared against the high-performance Xeon processors and the low-power SoCs.

VI. PERFORMANCE EVALUATION

To evaluate the performance and the energy-efficiency of the low-power SoCs, we have used a set of Spark applications

TABLE II
SUITE OF APPLICATIONS FOR THE PERFORMANCE EVALUATION

Application	Type	Description
Linear Regression	ML	Linear regression is used for modeling the relationship between a scalar dependent variable and one or more explanatory variables.
Logistic Regression	ML	Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function.
KMeans	ML	A method of vector quantization that is used for cluster analysis in data mining.
CC	GraphX	Compute the connected components of vertices
PageRank	GraphX	Used of page ranking in web search engines
Triangles	GraphX	Count the number of triangles

as benchmark. Spark provides a wide set of reference applications that can be used as micro-benchmarks. In the current evaluation we have evaluated three representative applications from the domain on machine learning and three representative applications from the domain of graph computation. The application and the description of each application is shown in Table II.

In the current evaluation we have measured the execution time and the energy efficiency of these application on 4 platforms; one Intel Xeon processor, One Intel i5 processor and two 64-bit ARM-based SoCs. We note that for the performance evaluation we have not used a cluster of nodes but just a simple node in which both the Worker and the Master node are hosted. Therefore, the current performance evaluation does not measure the performance of the processors on the Worker nodes but evaluates the performance of the the whole Spark framework. The characteristics of the processors and the operating system for each platform are presented in Table I. In all cases, 4 cores are used for the performance evaluation in order to make a fair comparison between the processors.

A. Execution time

The execution time refers to the total execution time of the evaluated application including the processes for the Master node, the Worker nodes and the resource management. The total execution time was measured based on the inherent timers of the application. As is shown in Figure 4, the total execution time of the applications running on the 64-bit SoCs are 6.1x to 13x higher than the one in the Xeon processor. It is interesting to note that although both the Qualcomm and the Broadcom SoC are using the same 64-bit A53 processors and these processors are clocked at the same frequency, there is overall higher execution time in the Snapdragon platform. Specifically, the average execution time of the Raspberry platform based on the Broadcom processor is 8x longer than

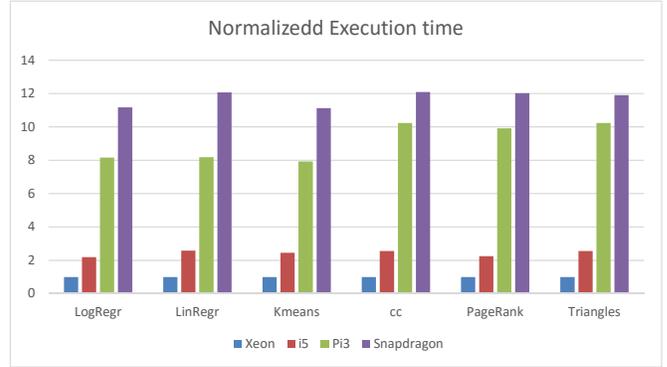


Fig. 4. Comparison of the execution time for the Spark applications. The execution times are normalized to the Intel Xeon processor

the Xeon processor and the execution time of the Snapdragon platform is 11x longer than the Xeon processor.

B. Power and energy consumption

The main advantage of the SoCs is that they are optimized for low power consumption. In this section we evaluate the energy efficiency ($power \times execution_time$) of the SoC for Apache Spark. In the current evaluation the energy consumption is based on TDP (Thermal Density Power) features of the processors. TDP refers to the maximum dissipated power of the processors. Therefore, the comparison on the energy efficiency between the platforms are just indicative and are used as a first approximation for the potential energy savings based on the SoCs. Figure 5 depicts the normalized energy efficiency of the Spark in the four platforms. As it is shown in this figure, the energy consumption of the low-power SoC is 2-3.5x better than the energy consumption based on the Intel Xeon processor.

VII. ACKNOWLEDGMENTS

This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 687628 - VINEYARD: Versatile Integrated Accelerator-based Heterogeneous Data Centers www.vineyard-h2020.eu

VIII. CONCLUSIONS

In this paper we evaluate the energy efficiency of low-power SoC platforms for the Apache Spark framework used in data analytics. The performance evaluation shows that the SoC-based processors are 8x to 11x worst in terms of total execution time. However, due to the lower power consumption they have the potential to offer much better energy efficiency. For machine learning application based on Spark they can provide up to 3x better energy efficiency while for graph computations can provide up to 3.5x better energy efficiency. Therefore, in cases where we care more about the energy

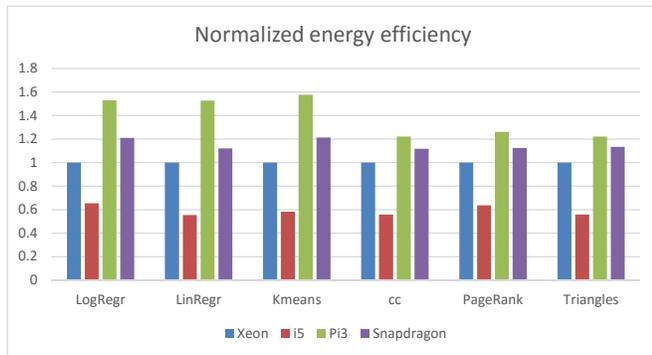


Fig. 5. Comparison of the energy efficiency. The energy efficiency is normalized to the Intel Xeon processor

efficiency and not mainly on the execution time, SoC-based servers could provide a promising alternative in order to reduce the power and the TCO of data centers.

REFERENCES

- [1] In *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 20142019 White Paper*.
- [2] Hadi Esmailzadeh, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Power challenges may end the multicore era. *Commun. ACM*, 56(2):93–102, February 2013.
- [3] Hadi Esmailzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, pages 365–376, New York, NY, USA, 2011. ACM.
- [4] Gradual change in server microprocessor market; idc expects competition and evolving workloads to change supply ecosystem in 2017. In *IDC Report*, 2016.
- [5] M. Malik, S. Rafatirah, A. Sasan, and H. Homayoun. System and architecture level characterization of big data applications on big and little core server architectures. In *Big Data (Big Data)*, 2015 *IEEE International Conference on*, pages 85–94, Oct 2015.
- [6] R. Azimi, X. Zhan, and S. Reda. How good are low-power 64-bit socs for server-class workloads? In *Workload Characterization (IISWC)*, 2015 *IEEE International Symposium on*, pages 116–117, Oct 2015.
- [7] Vijay Janapa Reddi, Benjamin C. Lee, Trishul Chilimbi, and Kushagra Vaid. Mobile processors for energy-efficient web search. *ACM Trans. Comput. Syst.*, 29(3):9:1–9:39, August 2011.
- [8] Vijay Janapa Reddi, Benjamin C. Lee, Trishul Chilimbi, and Kushagra Vaid. Web search using mobile cores: Quantifying and mitigating the price of efficiency. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ISCA '10, pages 314–325, New York, NY, USA, 2010. ACM.
- [9] Rafael Vidal Aroca and Luiz Marcos Garcia GonçAlves. Towards green data centers: A comparison of x86 and arm architectures power efficiency. *J. Parallel Distrib. Comput.*, 72(12):1770–1780, December 2012.
- [10] K. L. Keville, R. Garg, D. J. Yates, K. Arya, and G. Cooperman. Towards fault-tolerant energy-efficient high performance computing in the cloud. In *2012 IEEE International Conference on Cluster Computing*, pages 622–626, Sept 2012.
- [11] Mateusz Jarus, Sébastien Varrette, Ariel Oleksiak, and Pascal Bouvry. *Energy Efficiency in Large Scale Distributed Systems: COST IC0804 European Conference, EE-LSDS 2013, Vienna, Austria, April 22-24, 2013, Revised Selected Papers*, chapter Performance Evaluation and Energy Efficiency of High-Density HPC Platforms Based on Intel, AMD and ARM Processors, pages 182–200. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [12] Z. Ou, B. Pang, Y. Deng, J. K. Nurminen, A. Yi-Jski, and P. Hui. Energy- and cost-efficiency analysis of arm-based clusters. In *Cluster, Cloud and Grid Computing (CCGrid)*, 2012 *12th IEEE/ACM International Symposium on*, pages 115–123, May 2012.
- [13] Maria Malik, Avesta Sasan, Rajiv Joshi, Setareh Rafatirah, and Houman Homayoun. Characterizing hadoop applications on microservers for performance and energy efficiency optimizations. In *2016 IEEE International Symposium on Performance Analysis of Systems and Software*, pages 153–154, 2016.
- [14] C. Kaewkasi and W. Srisuruk. A study of big data processing constraints on a low-power hadoop cluster. In *Computer Science and Engineering Conference (ICSEC)*, 2014 *International*, pages 267–272, July 2014.
- [15] Apache, spark.
- [16] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.