Performance Evaluation of Embedded Processor in MapReduce Cloud Computing Applications

Christoforos Kachris[1,*], Georgios Sirakoulis[1], and Dimitrios Soudris[2]

[1] Department of Electrical and Computer Engineering,
Democritus University of Thrace, Xanthi, Greece
{ckachris,gsirak}@ee.duth.gr
[2] Department of Electrical and Computer Engineering,
National Technical University of Athens, Athens, Greece
dsoudris@microlab.ntua.gr

Abstract. Current data centers consume huge amount of power to face the increasing network traffic. Therefore energy efficient processors are required that can process the cloud applications efficiently without consuming excessive power. This paper presents a performance evaluation of the processors that are mainly used in high performance embedded systems in the domain of cloud computing. Several representative applications based on the widely used MapReduce framework are mapped in the embedded processor and are evaluated in terms of performance and energy efficiency. The results shows that high performance embedded processors can achieve up to 7.8x better energy efficiency than the current general purpose processors in typical MapReduce applications.

Keywords: cloud computing, green computing, embedded processors, mapreduce, data centers.

1 Introduction

Over the last few years, the exponential increase of the Internet traffic, mainly driven from emerging applications like streaming video, social networking and cloud computing has created the need for more powerful warehouse data centers. These data centers are based on thousands of high performance servers interconnected with high performance switches. Most of the applications that are hosted in the data center servers (e.g. cloud computing applications, search engines, etc.) are extremely data-intensive and require high interaction between the servers in the data center.

A main concern in the design and deployment of a data centers is the power consumption. Many data consume a tremendous amount of electricity; some consume the equivalent of nearly 180,000 homes [1]. Greenpeace's Make IT Green report [2],

[*] The research project is implemented within the framework of the Action «Supporting Postdoctoral Researchers» of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

M. Yousif, L. Schubert (Eds.): Cloudcomp 2012, LNICST 112, pp. 43–51, 2013.

estimates that the global demand for electricity from data centers was around 330bn kWh in 2007 (almost the same amount of electricity consumed by UK [3]). This demand in power consumption demand is projected to more than triple by 2020 (more than 1000bn kWh). According to some estimates [3],[4], the power consumption of the data centers in the US in 2006 was 1.5% of the total energy consumed at a cost of more than $4.5B.

	Derive electricity consumption	**Forecast electricity consumption**
	Billion kWh 2007	**Billion kWh 2020**
Data Centers	330	1012
Telecoms	293	951
Total Cloud	**623**	**1963**

Fig. 1. Power consumption of data centers, Source: Greenpeace, [2]

The power consumption inside the data center is distributed in the following way: the servers consume around 40% of the total IT power, storage up to 37% and the network devices consume around 23% of the total IT power [5]. And as the total power consumption of IT devices in the data centers continues to increase rapidly, so does the power consumption of the HVAC equipment (Heating-Ventilation and Air-Conditioning) to keep steady the temperature of the data center site. Therefore, the reduction in the power consumption of the network devices has a significant impact on the overall power consumption of the data center site. According to a study from Berk-Tek, saving 1W from the IT equipment results in cumulative saving of about 2.84W in total power consumption due to the reduced power consumption of the cooling systems [6]. Therefore, a reduction on the power consumption of the interconnection network will have a major impact on the overall power consumption of the data center.

The power consumption of the data centers has also a major impact on the environment. In 2007, data centers accounted for 14% of the total ICT greenhouse gases (GHG) emissions (ICT sector is responsible for 2% of global GHG emissions), and it is expected to grow up to 18% by 2020 [2]. The global data center footprint in greenhouse gases emissions was 116 Metric Tonne Carbon Dioxide (MtCO2e) in 2007 and this is expected to more than double by 2020 to 257 MtCO2e, making it the fastest-growing contributor to the ICT sector's carbon footprint.

Therefore, more energy efficient servers are required for the emerging cloud computing applications. 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). The comparison has shown that low power general purpose cores can achieve better energy efficiency in the domain of web search applications. One of the first companies that adopted the used of low power general purpose processors was SeaMicro [9]. SeaMicro introduced in 2011 a new version of servers that packed 768 Intel Atom cores into a 10U chassis. According to the company the Atom-based data center could achieve ¼ the power and 1/6 the space of the commodity volume servers. Another company, Calxeda Inc. has recently presented a server based on the ARM cores called Server-on-a-Chip (SoC).

According to the company the EnergyCore ECX-1000 is the most energy-efficient processor for data centers [10]. However, until now there is not any qualitative comparison between the embedded processors and the general purpose processors in the domain of cloud computing applications.

In this paper we present a performance evaluation between the general purpose processors and the embedded processors (that are inherently designed for low power applications) for cloud computing applications that are based on the MapReduce framework. Section 2 presents the most common cloud computing applications under the MapReduce framework. Section 3 presents the architectural details of the processors and shows the comparison of these processors in terms of performance and energy consumption. Finally, Section 4 presents the conclusions of this paper.

2 Cloud Computing Applications

One of the most widely used frameworks that are hosted in the data centers is the MapReduce framework. **MapReduce** is a programming model and an associated implementation for processing and generating large data sets [11]. Users specify a map function that processes a *key/value* pair to generate a set of intermediate *key/value* pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

The **Map function** takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key *I* and passes them to the Reduce function.

The **Reduce function** accepts an intermediate key *I* and a set of values for that key. It merges together these values to form a possibly smaller set of values. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.

2.1 The Phoenix MapReduce Framework

Phoenix is a programming API and runtime system based on Google's MapReduce model developed by Stanford University [12],[13]. The main implementation of the Phoenix framework is based on the same notions of the MapReduce framework as it is depicted in Figure 2. The **Map** function processes the input data and generates a set of intermediate key/value pairs. The **Reduce** function properly merges the intermediate pairs which have the same key. Given such a functional specification, the MapReduce runtime automatically parallelizes the computation by running multiple **map** and/or **reduce** tasks in parallel over disjoined portions of the input or intermediate data. Google's MapReduce implementation facilitates processing of terabytes on clusters with thousands of nodes. The Phoenix implementation is based on the same principles but targets shared-memory systems such as multi-core chips and symmetric multiprocessors.

Phoenix uses threads to spawn parallel Map or Reduce tasks. It also uses **shared-memory** buffers to facilitate communication without excessive data copying. The runtime schedules tasks dynamically across the available processors in order to achieve load balance and maximize task throughput. Locality is managed by adjusting the granularity and assignment of parallel tasks.

In this paper we evaluate 5 applications (4 of them commonly used in cloud application and on general application) that have been implemented using the Phoenix MapReduce framework [12]:

- **Word Count:** This application is commonly used in search engines for the indexing of the web pages based on the words. It counts the frequency of occurrence for each word in a set of files. The Map tasks process different sections of the input files and return intermediate data that consist of a word (key) and a value of 1 to indicate that the word was found. The Reduce tasks add up the values for each word (key).
- **String Match:** It processes two files: the "encrypt" file contains a set of encrypted words and a "keys" file contains a list of non-encrypted words. The goal is to encrypt the words in the "keys" file to determine which words were originally encrypted to generate the "encrypt file".
- **Histogram:** It analyzes a given bitmap image to compute the frequency of occurrence of a value in the 0-255 range for the RGB components of the pixels. It can be used in image indexing and image search engines.
- **Linear Regression:** It computes the line that best fits a given set of coordinates in an input file. The algorithm assigns different portions of the file to different map tasks, which compute certain summary statistics like the sum of squares.
- **Matrix Multiply:** Each Map task computes the results for a set of rows of the output matrix and returns the (x,y) location of each element as the key and the result of the computation as the value. This application is a mainly computational intensive application and has been added to show the differences between typical mathematic benchmarks with the applications that are used in cloud computing applications.

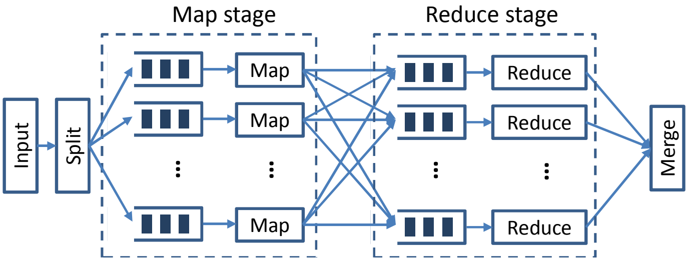

Fig. 2. The Phoenix MapReduce framework

3 Performance Evaluation

In this section we evaluate the Phoenix MapReduce framework in terms of performance and energy efficiency. The Phoenix MapReduce framework has been mapped to three different processors. The first processor is based on a high performance general purpose processor (HP-GPP: Intel i7-2600). This processor has 4 cores and the clock speed is 3.4GHz. The second processor is based on a low power general purpose processor (LP-GPP: Intel U5400 processor) with 2 cores and maximum clock frequency 1.2GHz. The third processor is an embedded system processor that is based on the OMAP4430 SoC with 2 ARM Cortex A9 cores [14]. The detailed characteristics of the processors are shown in the following table.

Table 1. Processor architecture characteristics

	HP-GPP	**LP-GPP**	**EP**
Processor	i7-2600	U5400	OMAP4430
# of Cores	4	2	2
Cores	Intel i7	Intel Pentium	ARM Cortex A9
Process	32nm	32nm	45nm
Frequency	3.4GHz	1.2GHz	1GHz
ISA	CISC	CISC	RISC
L1 Cache	64KB (I),64KB (D)	64KB (I),64KB (D)	32KB (I),32KB (D)
L2 Cache	256KB per core	256KB per core	1MB (shared)
L3 Cache	8MB	3MB	-
Instruction Set	64-bits	64-bits	32-bits
Integrated Graphics	YES	YES	YES

As it is shown in the table the high performance processor has larger cache size and higher clock frequency while the other two processors that are optimized for low power consumption have much smaller caches and lower clock frequencies. The main difference is that the first two processors are based on the Pentium x86 CISC instruction sets while the OMAP4430 processor is based on the ARMv7 RISC instruction set and is optimized for embedded systems applications. Furthermore, the Intel processors are 64-bits wide while the ARM cores are based on 32-bits. In both cases the Phoenix MapReduce framework was hosted on the same operating systems (Ubuntu Linux 11.10).

3.1 Performance Evaluation

Figure 3 depicts the performance evaluation of 4 different common cloud tasks using the Phoenix MapReduce framework in terms of execution time. Besides the 4 common cloud tasks it shows also the execution time of a typical benchmark (matrix multiplication) in order to show the differences between the cloud tasks and other common tasks used in benchmarking. The figure shows the normalized execution time compared to the HP-GPP (Intel i7-2600). As it is shown in this figure the execution time of the embedded processor is 3x to 8.4x higher than the execution time of the HP-GPP while it is 3x to 5x higher than the low power GPP. The lower execution time of the GPPs can be justified by the higher clock frequency and the more advanced instruction set (e.g. deeper pipeline scheme, larger L3 cache and more advanced branch prediction schemes). The highest difference in the execution time is however noticed on the *matrix multiplication* which is not used in the cloud computing applications. In the commonly used cloud computing tasks such as the *word count*, *histogram, linear regression* and *string match* the execution time of the embedded processor range from 3x to 5x higher than the HP-GPP.

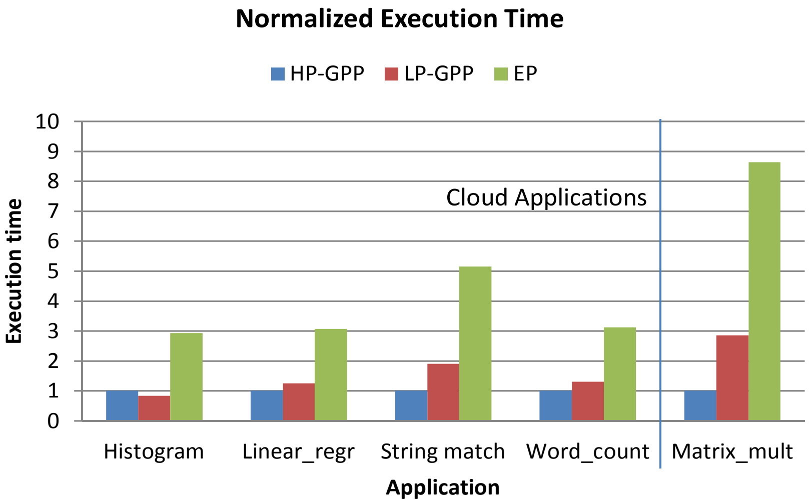

Fig. 3. Normalized Execution time for difference applications

The higher speedup of the *matrix multiplication* can be also justified by Figure 4. This figure shows the average miss rate of the branch predictions for the Intel low power processor. In this figure it is clear that the typical benchmark applications such as the matrix multiplication are much more predictable due to the control structure and therefore the branch miss rate is much smaller than the common cloud application tasks.

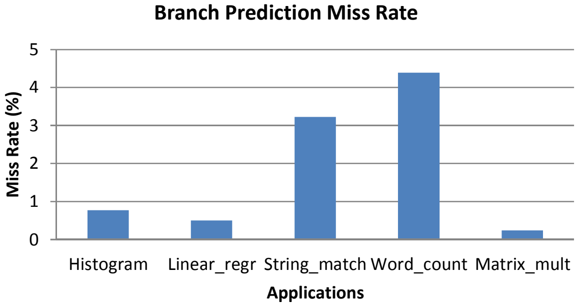

Fig. 4. Branch prediction miss rate

3.2 Energy Efficiency

In this section we evaluate the energy efficiency of the embedded processors compared to the general purpose processors. The energy efficiency is measured by the product of the power consumption by the total execution time of a specific task [15].

$$Energy = ExecTime \cdot Power$$

For the ARM processor we measure the power consumption of the processors using the *Pandaboard* [16] which integrates the OMAP4430 chip with the ARM processors and the DRAM memory. The power measurements are based on the current that is drawn by the ARM processors [17]. On the Intel processors we measure the power consumption using the *powerstat* application [18]. In all cases the CPU utilization is above 80% for the cloud computing applications which means that all processors consume almost the maximum power consumption (Figure 5). Again this figure shows the difference between the matrix-multiplication applications with the typical cloud computing applications. In the case of *matrix multiplication* the processor is fast enough to perform the tasks and the lower utilization is due to the system calls.

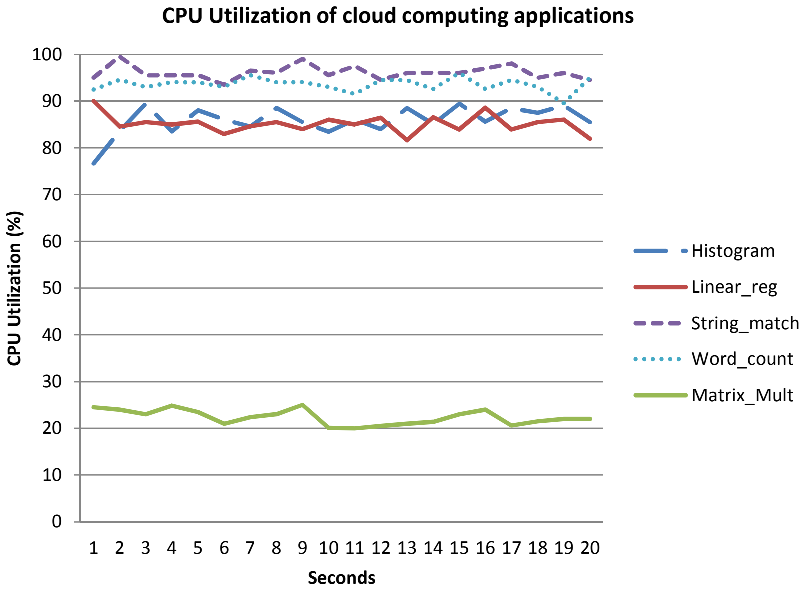

Fig. 5. CPU Utilization of the applications in time

Figure 6 depicts the normalized energy consumption of the HP-GPP, the LP-GPP and the embedded processor for different applications of the Phoenix MapReduce framework. The figure shows the normalized energy based on the energy consumption of the embedded processor. As is it shown in this figure the embedded processors can achieve up to 7.8x lower **energy** consumption compared with the HP-GPP. This is due to the fact that the **power** consumption of the embedded processor is much lower than the power of the GPP. The high power consumption of the GPP is due to the complex instruction set, the advanced branch prediction schemes and the larger caches of the processors. Therefore, even in the case that the embedded processor has longer execution time than the GPPs the overall energy that it consumes is much lower than the GPP. Therefore, in data centers which require energy efficient servers such as the microservers [9], embedded systems could be utilized efficiently reducing the overall power consumption. Furthermore, as the most cloud applications than are based on MapReduce framework are designed to run in parallel systems, the servers could even achieve the same performance in terms of throughput by replicating more embedded system cores but consuming much lower energy.

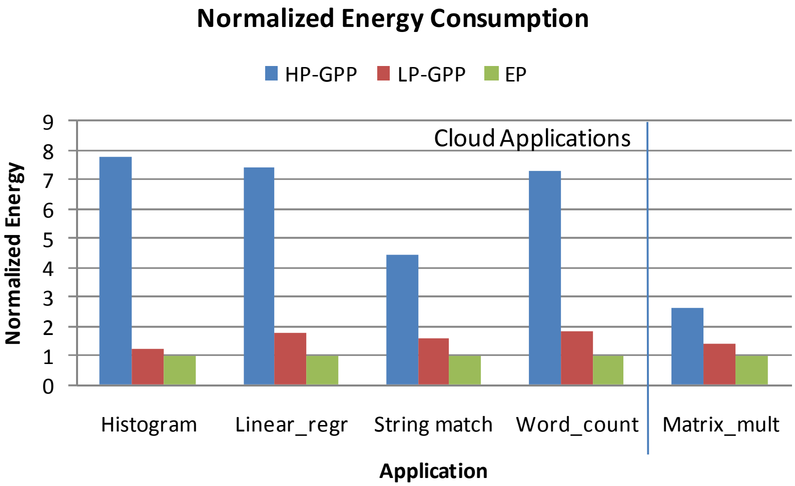

Fig. 6. Normalized Energy Consumption for different applications

4 Conclusions

In this paper we evaluate high performance embedded processors in the domain of cloud computing. We map typical cloud computing application in the ARM Cortex A9-MPCore cores and we compare it with high performance and low power general purpose processors. The performance evaluation shows that the execution time of the embedded processors is up 5x higher than the general purpose processors in tasks common in the cloud applications (word count, string match, etc.). However, the power consumption of the embedded processors is significantly lower the general purpose processors. Therefore high performance embedded processors can achieve up to 7.8x better energy efficiency in cloud computing applications, compared with the general purpose processors and they could be a viable alternative in data centers with lower energy consumption requirements such as microservers. These embedded processors could also be a promising alternative to any other cloud computing applications that can tolerate a small increase in the overall execution time but consuming much lower energy and thus reducing the operating cost of these data centers.

References

1. How Clean is Your Cloud. Greenpeace Technical Report (2012)
2. Make IT Green: Cloud Computing and its Contribution to Climate Change. Greenpeace International Technical Report (2010)
3. Report to Congress on Server and Data Center Energy Efficiency, U.S. Environmental Protection Agency, ENERGY STAR Program (2007)
4. SMART 2020: Enabling the low carbon economy in the information age, A report by The Climate Group on behalf of the Global eSustainability Initiative (GeSI) (2008)
5. Where does power go? GreenDataProject (2008), `http://www.greendataproject.org`
6. Huff, L.: Berk-Tek: The Choise for Data Center Cabling (2008)
7. Reddi, V.J., Lee, B.C., Chilimbi, T., Vaid, K.: Mobile processors for energy-efficient web search. ACM Trans. Comput. Syst. 29(3) (2011)
8. Reddi, V.J., Lee, B.C., Chilimbi, T., Vaid, K.: Web search using mobile cores: quantifying and mitigating the price of efficiency. In: Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA 2010) (2010)
9. The SeaMicro SM10000 Family System Overview, Datasheet, SeaMicro Inc. (2011)
10. Calxeda EnergyCore: ECX-1000 Series, Datasheet, Calxeda, Inc. (2012)
11. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: 6th Symposium on Operating Systems Design and Implementation (OSDI 2004) (2004)
12. Ranger, C., Raghuraman, R., Penmetsa, A., Bradski, G., Kozyrakis, C.: Evaluating MapReduce for Multi-core and Multiprocessor Systems. In: Proceedings of the 13th Intl. Symposium on High-Performance Computer Architecture (HPCA), Phoenix, AZ (February 2008)
13. Yoo, R.M., Romano, A., Kozyrakis, C.: Phoenix Rebirth: Scalable MapReduce on a Large-Scale Shared-Memory System. In: Proceedings of the 2009 IEEE International Symposium on Workload Characterization (IISWC), Austin, TX, pp. 198–207 (October 2009)
14. OMAP4430 Device Silicon Revision, Technical Reference Manual, Texas Instrument
15. Kaxiras, S., Martonosi, M.: Computer Architecture Techniques for Power-Efficiency. Morgan & Claypool Publishers (2008) 1598292080
16. Pandaboard, `http://www.pandaboard.org`
17. van Eijndhoven, J.: Measuring Power Consumption of the OMAP4430 using the PandaBoard. Vectofabrics Inc. (November 2011)
18. Powerstat: Power Consumption Calculator for Ubuntu Linux
19. Bohra, N., Eylon, E.: Micro Servers:An Emerging Category For Data Centers. Server Design Summit (November 2011)