Big Data on Low Power Cores

Are Low Power Embedded Processors a good fit for the Big Data Workloads?

Maria Malik
Department of Electrical and Computer Engineering
George Mason University, Fairfax, VA, USA
mmalik9@gmu.edu

Houman Homayoun
Department of Electrical and Computer Engineering
George Mason University, Fairfax, VA, USA
hhomayou@gmu.edu

***Abstract*— The traditional low-power embedded processors such as ARM and Atom are entering the high-performance server market. At the same time, big data analytics are emerging and dramatically changing the landscape of data center workloads. Thus, the question of whether low-power embedded architectures are suited to process big data applications efficiently, is becoming important. In this work, through methodical investigation of power, performance measurements and comprehensive system level analysis, we demonstrate that low power embedded architectures can provide significant energy-efficiency for processing big data analytics applications.**

***Index Terms*—Performance, Power, Energy-efficiency, Big Data, Low-Power server**

I. Introduction

The volume of available data has exploded in recent years as a result of new social behaviors, societal transformations as well as advances in various branches of technology – data sensing, data communication, data computation, and data storage. The world of big data is changing constantly that creates challenges to process the applications using existing solutions. Big data applications require computing resources and storage subsystems that can scale to manage massive amounts of diverse data. Individuals, businesses, governments, and society as a whole are now having access to enormous collections of big data, are being forced to introduce more processing nodes to their computing infrastructure or replace their existing hardware with more powerful systems. This trend increases the infrastructure cost and power consumption. We believe this is the right time to identify an efficient computing platform for big data analytics that can provide balance between processing capacity and power efficiency.

Big data applications, in particular from the web service, share many inherent characteristics that are fundamentally different from traditional desktop, parallel, and scale-out applications [4, 6]. Big data analytics applications in these domains, heavily rely on big-data-specific deep machine learning and data mining algorithms, and are running complex database software stack with significant interaction with I/O and OS. In addition, unlike conventional CPU applications, big data applications combine high data rate requirement with high computational power requirement, in particular for real-time and near-time processing constraints.

This new set of characteristics is necessitating a change in the direction of server-class microarchitecture to improve their computational and memory efficiency. However, while demand for data center computational resources continues to grow as the size of data grows, the semiconductor industry has reached its physical scaling limits and is no longer able to reduce power consumption in the new chips. Physical design constraints, such as power and density, have therefore become the dominant limiting factor for scaling out data centers [3, 7]. In response to this challenge, low power embedded processors can be a promising solution to enhance energy-efficiency. Using little cores, such as Atom and ARM, is a new trajectory in server design that advocates the use of low-power cores to address the dark silicon challenge facing servers [14]. Due to the wide adoption of x86-based architectures in servers, we choose Atom to study in this paper, as it has a low power embedded micro-architecture with high performance x86 ISA.

In this paper, we present the power and performance analysis of the big data applications on Intel Atom platform. Additionally, we have performed the Energy-Delay Product (EDP) analysis to evaluate the trade-offs between power and performance on Little Atom and Big Xeon cores. The results demonstrate that big cores clearly offer better performance while little cores are significantly more energy efficient.

Contributions: This paper makes the following key contributions:

- *We analyze the measurements of performance and power of Big Data applications on two state-of-the-art server platforms, one with Intel™ Xeon; Big cores and the other with Intel™ Atom; Little cores. We compare the results with traditional CPU, parallel and scaleout applications.*
- *We analyze the EDP results to evaluate the trade-off between power and performance on little and big core.*
- *We evaluate the processing capacity and efficiency under different data sizes (per node) by using two metrics –Data Processed Per Second (DPS) and Data Processed Per Joule (DPJ)- to understand the impact of size of big data per node on power and performance efficiency.*

The rest of the paper is organized as follows. Section II describes big data workload, selected scale-out workloads and traditional CPU benchmark, which we have studied in this work. Section III presents the measurement and methodology details. Section IV presents the experimental results and provides the performance, power, energy-efficiency, and processing capacity analysis study of big data applications. Section V provides the related work. Lastly, we have presented the brief conclusion acquired from our analysis.

II. Dominant Big Data workloads

The studied big data workloads –microbenchmarks and application level benchmarks- in this paper are representative programs from 15 different domains such as graph and data mining, data analysis platform and pattern searching applications, which are frequently used in the real world. We provide these selected applications, along with their particular domain and datasets used to derive the studied applications in Table 1.

978-1-4673-7166-7/15/$31.00

III. MEASUREMENT TOOLS AND METHODOLOGY

We conduct our study on two state-of-the-art servers, Intel Xeon and Intel Atom. Intel Xeon E5 enclosed with two Intel E5-2420 processors that includes six aggressive processor cores per node with three-level of cache hierarchy. Intel Atom C2758 has 8 processor cores per node and a two-level cache hierarchy. Table 2 summarizes the key architectural parameters of these two servers. We analyze the architectural behavior using Intel VTune [2], a performance-profiling tool that provides an interface to the processor performance counters. We have used Watts up PRO power meter to measure the power consumption of the servers [15]. Wattsup power meter measures and records power consumption at one second granularity. The power reading is for the entire system, including core, cache, main memory, hard- disks and on-chip communications buses. We have collected the average power consumption of the studied applications and subtracted the system idle power to calculate the dynamic power dissipation of the entire system. The main analysis of this work includes the system-level analysis- performance, power, EDP, DPS, DPJ- for big data applications and Hadoop micro-benchmarks.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we discuss the system-level analysis of little core, when running traditional CPU benchmarks, scale-out, and big data applications. Due to space constraints, we are only reporting the average results for SPEC, PARSEC and Scale-Out applications. Moreover, we have conducted the data size sensitivity analysis of Hadoop micro-benchmarks with the dataset of 10MB, 100MB, 1GB and 10GB per node to understand the impact of size of data per processing node on system-level parameters.

A. Performance Analysis

In this section, we analyze the performance measurements of big data applications and compare it with the traditional benchmarks. Figure 1.1 presents the processor's performance in terms of IPC. The average IPC of big data is 1.65 times lower than the traditional benchmarks on big core and 1.21 times lower than the traditional benchmarks on little core. In general, we observe lower IPC in big data applications compared with the traditional benchmarks. Furthermore, little core is experiencing 1.43 times lower IPC in comparison to big core as Xeon can process up to 4 instructions simultaneously while Atom is limited to 2 instructions per cycle. Figure 1.2 shows the IPC of Hadoop micro-benchmarks for different data sizes. The results are consistent with the results in Figure 1.1 showing lower IPC on little core compared to big core across all data sizes. We also observe that on the little core, increasing the data size reduces the IPC since the cache misses increases. However, on big core while for most cases, increase in data size per node reduces the IPC, there are few exceptions where increase the data from 100MB to 1000MB per node increases the IPC. This is mainly due to higher cache locality as a result of large and more complex cache subsystem in big core.

B. Power Consumption and Energy-Efficiency Analysis

In this section, we report the power consumption of big data applications and discuss the energy-efficiency analysis to evaluate the trade-off between power and performance in embedded Atom core compare to high performance Xeon core.

1) Power Characterization

Figure 2.1 shows the average power consumption of studied applications on big core; Intel Xeon and little core; Intel Atom. Big core consumes on average 35 Watts of power with the peak of 44 Watts in cluster application. Little core consumes much lower power per core as expected, ranging from 0.9 to 6.5 Watts with an average of 4.8. Figure 2.2 shows that the power consumption increases as the size of data per node increases in most cases across both big and little architectures. This is more noticeable in little core. While increasing the data size in little core reduces the IPC and therefore core power, it increases cache and off-chip traffic in DRAM and bus subsystem. Therefore for embedded little core where cache, DRAM and off-chip components are dominant power consumer (unlike high performance Xeon core), a clear rise in power consumption is observed as the size of data increases.

TABLE 2: Architectural Parameters

Processor	Intel Atom C2758	Intel Xeon E5-2420
Operating Frequency	2.40GHz	1.9GHz
Micro-architecture	Silvermont	Sandy Bridge
L1i Cache	32 KB	32 KB
L1d Cache	24 KB	32 KB
L2 Cache	4*1024 KB	256KB
L3 Cache	-	15MB
PageTable	16972 KB	4260 KB
System Memory	8GB	32GB
TDP	20W	95W

TABLE 1: Studied Big Data Applications

Workloads	Applications		Data Semantics	Software Stacks
Big Data	Hadoop Micro-benchmarks	WordCount	Generated	Hadoop 1.2.1
		Sort		
		Grep		
		TeraSort		
		TestDFSIO-Write		
		TestDFSIO-Read		
	Graphbuilder [9]		Wikipedia	GraphBuilder 0.0.1, Hadoop 1.2.1
	Collaborative Filtering [11]		MovieLens	Hadoop 1.2.1, Mahout 0.6
	Clustering [11]		UCI Machine learning Repository	
	FP-Growth [11]		Frequent Itemset Mining Dataset Repository [10]	
	SPMF [8]			SPMF 0.96
Scale-out	Classification [3]		Wikipedia	Hadoop 1.2.1, Mahout 0.6, Memcached server/client 1.4.15, CloudSuite 2.0
	Graph-Anlysis [3]		Twitter data set	
	Data-caching [3]		Twitter data set	
Traditional	Spec2006		Reference input	Spec 2006
	Parsec		Native input	Parsec 2.1

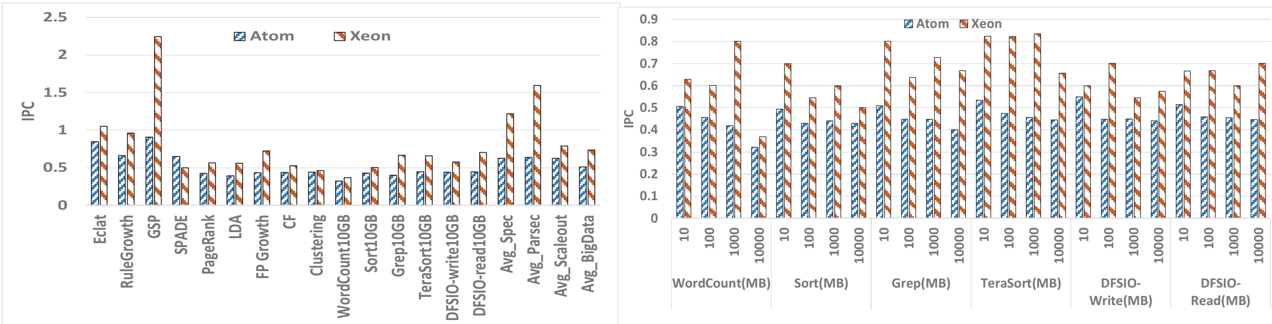

Figure 1: IPC (1.1) Big Data workloads (1.2) Different configurations of Hadoop micro-benchmarks

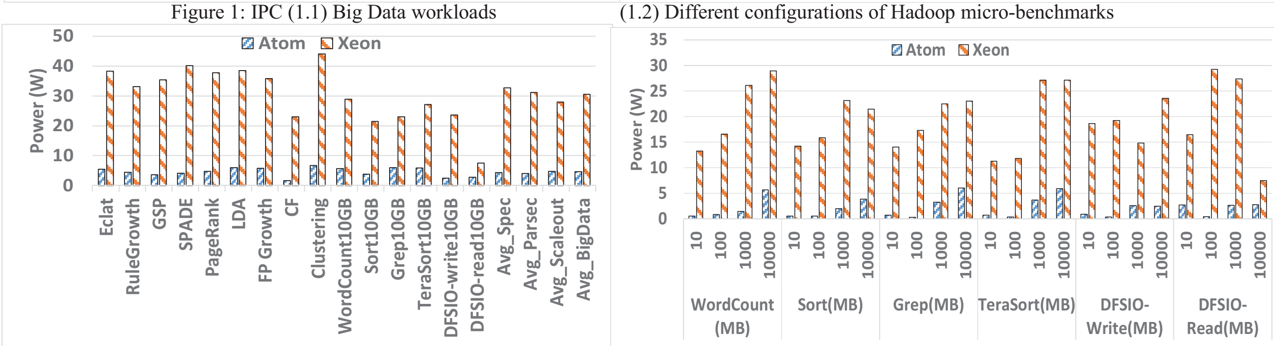

Figure 2: Power (2.1) Big Data workloads (2.2) Different configurations of Hadoop micro-benchmarks

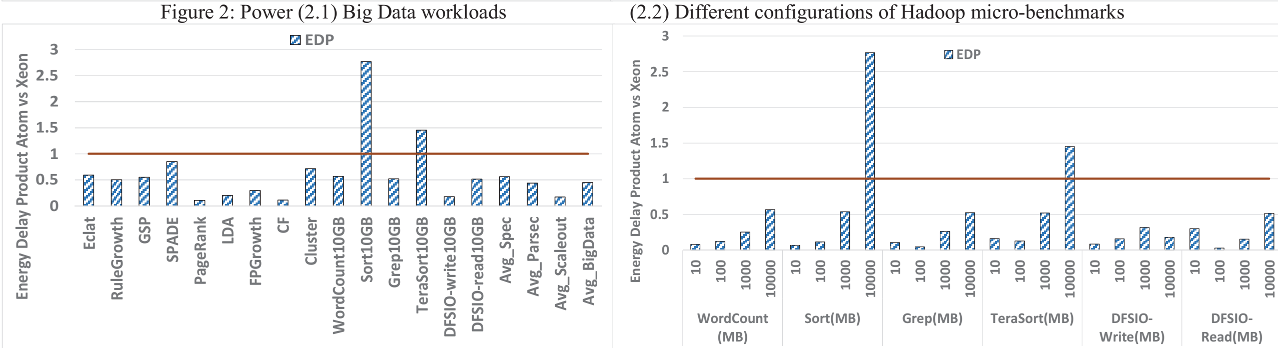

Figure 3: EDP Analysis (3.1) Big Data workloads (3.2) Different configurations of Hadoop micro-benchmarks

2) Energy-Efficiency Analysis

Based on the results of power consumption for both platforms, we have evaluated the trade-off between power and performance by investigating the EDP metric. Figure 3.1 illustrates EDP ratio for little vs. big cores. The EDP results show that little core is noticeably more efficient than big core in almost all the applications. Figure 3.2 presents the data sensitivity analysis of Hadoop micro-benchmarks. The results show that for some applications such as sort and terasort, as the size of data increases (10000MB), big core is becoming more efficient than the little core. Complex memory subsystem in big core along with higher processing capacity (2X more than little core) allows big core to be more efficient in hiding the cost of high I/O communication in these applications as the size of data grows.

Observation**-** The results illustrate that overall little core is more efficient in terms of EDP than big cores. However, as the size of the data grows big core is becoming more efficient across a number of big data applications.

C. DPS-DPJ Analysis

In this section, we evaluate the data processing capability of big and little cores for various data sizes running Hadoop micro-benchmarks. We report the data processed per second (DPS) and the data processed per joule (DPJ) metrics to compare the data processing capacity and efficiency of the two architectures. The results are reported in Figure 4.1 and Figure 4.2. For most applications, with an increase in the data size the DPS first rises rapidly to a peak and then declines slightly. The data size at which the peak DPS occurs varies across applications and architectures. The peak DPS occurs in terasort and sort at only 1000MB of size, while in other applications occurs in at least an order of magnitude larger data size. The reason is that sort and terasort are I/O intensive applications and the rise in data size exacerbates the I/O cost to an extent that it diminishes the benefit of high processing capacity.

The DPS difference between big and little cores is becoming larger for CPU intensive applications such as grep and wordcount as the size of data increases. However, this is not the case for I/O intensive applications such as sort and DFSIO-read as the I/O cost becomes the dominant performance bottleneck and the processing capacity of the processor; i.e. big vs. little become a less important factor. Overall, for small data size, below 1000MB per node, the two architectures, big and little, have almost similar processing capacity in terms of DPS, and it is only for large data sizes that the DPS gap between the two becomes clear.

Similar to DPS, for DPJ, in all applications we observe a rise in data processing efficiency on big core as the size of

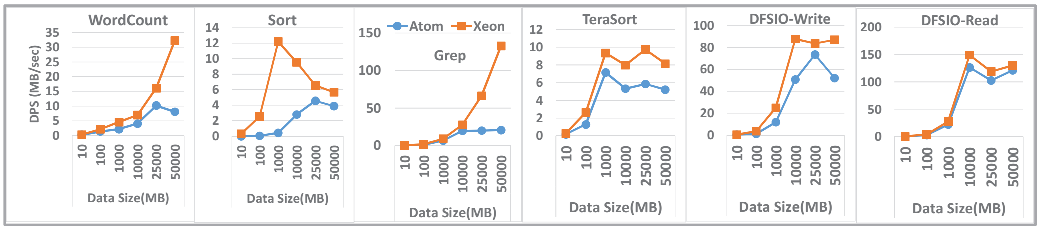

Figure 4.1: DPS Analysis of different configurations of Hadoop micro-benchmarks

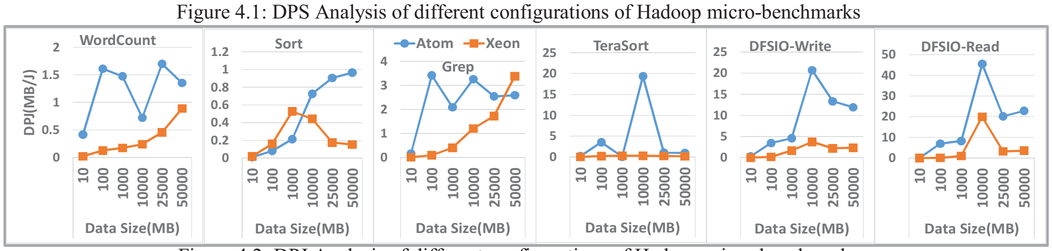

Figure 4.2: DPJ Analysis of different configurations of Hadoop micro-benchmarks

data increases However in I/O intensive applications such as *terasort*, DFSIO-read and write, the rise in DPJ on Xeon is insignificant. For CPU intensive applications including *wordcount* and *grep* there is a significant rise in DPJ on Xeon as the size of data increases. Overall, for I/O intensive applications such as sort, *terasort*, *DFSIO-write and DFSIO- -read*, little core is noticeably more efficient than big core. However, in CPU intensive micro-benchmarks, *WordCount and Grep*, the DPJ gap between big and little core reduces with the increase in data size. It is also interesting to observe that the DPJ of Xeon can exceed Atom in a number of applications and across various data sizes.

Observation-The results illustrate that the performance-efficiency indicated by DPS and DPJ is closely decided by the application type, its complexity (I/O vs. computing intensity), and the size of data. In a number of applications a clear rise in DPS and DPJ is observed as the size of data grows. These results can help to guide MapReduce scheduling decision to provide high performance-efficiency.

V. Related Work

Recently, there have been a number of efforts to understand the behavior of big data and cloud-scale applications [3, 5, 16, 17]. The most prominent big data benchmarks include HiBench, Scale-Out, BigDataBench, CloudCmp, and LinkBench [4, 5, 6, 7, 12]. The CloudSuite [3, 7] benchmark was developed for Scale-Out cloud workloads and mainly includes small data sets, e.g., 4.5 GB for Naïve Bayes. In addition, several later works make the case for low power embedded cores to improve the design efficiency and throughput of traditional server applications [1, 13].

This work is different from all above works as it primarily focuses on emerging big data applications, and performs a comprehensive system-level (power, performance, EDP, DPS, and DPJ impact) analysis of various big data applications and micro-benchmarks on little core to understand whether small low-power embedded cores are satisfying the power-performance requirement of this emerging class of applications.

VI. Conclusions

In this paper, we present a comprehensive system level analysis of running big data applications on embedded processor, a recent trend in server design which advocates the use of a low-power little core to address the power and performance-efficiency challenges. We show that while high performance big cores provide high performance-efficiency for traditional CPU applications compared to little cores, they are not power efficient to process big data applications. Presented results show that low power embedded core is noticeably more efficient for big data processing across various data sizes. We evaluate the processing capacity and efficiency under different data sizes to understand the impact of size of big data per node on power and performance efficiency. The results illustrate that the performance-efficiency indicated by DPS and DPJ is closely decided by the application type, its complexity (I/O vs. computing intensity), and the size of data. These results can help to guide MapReduce scheduling decision to provide high performance-efficiency.

References

[1] V. J. Reddi, et al. Web Search Using Mobile Cores: Quantifying and Mitigating the Price of Efficiency. In Proc. of 37th Annual ISCA, 2010.

[2] Intel VTune Amplifier XE Performance Profiler.

[3] M. Ferdman, et al. Clearing the clouds: a study of emerging scale-out workloads on modern hardware. In Proc. of the ASPLOS, Mar. 2012.

[4] S. Huang, et al. The HiBench benchmark suite: characterization of the MapReduce-based data analysis. In 26th ICDE,Mar10

[5] W. Gao, "BigDataBench: a Big Data Benchmark Suite from Web Search Engines".(ASBD 2013) in conjunction with ISCA 2013

[6] Li,Ang,et al. CloudCmp: comparing public cloud providers. ACM,'10

[7] A. Ghazal, Bigbench: Towards an industry standard benchmark for big data analytics. In: ACM SIGMOD Conference (2013)

[8] SPMF; http://www.philippe-fournier-viger.com/spmf/

[9] T. L. Willke and N. Jain. GraphBuilder – A Scalable Graph Construction Library for Apache™ Hadoop™, in Big Learning WS at NIPS. 2012.

[10] Frequent Itemset Mining Dataset Repository; http://fimi.ua.ac.be/data/

[11] Apache Mahout: scalable machine-learning and data-mining library

[12] Armstrong, Timothy G., et al. "Linkbench: a database benchmark based on the facebook social graph." Proc. of the ACM/SIGMOD ACM, 2013.

[13] D. G. Andersen, et al. FAWN: A Fast Array of Wimpy Nodes. In the Proceedings of the ACM SIGOPS 22nd SOSP, pages 1–14, 2009.

[14] Hardavellas, Nikos, et al. "Toward dark silicon in servers." IEEE Micro 31.EPFL-ARTICLE-168285 (2011): 6-15

[15] WattsUpPro power meter https://www.wattsupmeters.com/secure/index.php

[16] K. Neshatpour, et al. "Accelerating Big Data Analytics Using FPGAs". In 23rd IEEE FCCM, 2015.

[17] K. Neshatpour, Houman Homayoun, "Accelerating Machine Learning Kernels in Hadoop Using FPGAs". 15th IEEE/ACM CCGRID, 2015.