The Implications from Benchmarking Three Big Data Systems

Jing Quan[1] Yingjie Shi[2] Ming Zhao[3] Wei Yang[4]

[1]School of Software Engineering, University of Science and Technology of China, Hefei, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[3]School of Computing and Information Sciences, Florida International University, Florida, USA
[4]School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

sea11424@mail.ustc.edu.cn shiyingjie@ict.ac.cn ming@cs.fiu.edu qubit@ustc.edu.cn

***Abstract*—Along with today's data explosion and application diversification, a variety of hardware platforms for data centers are emerging and are attracting interests from both industry and academia. The existing hardware platforms represent a wide range of implementation approaches, and different hardware have different strengths. In this paper, we conduct comprehensive evaluations on three representative data center systems based on BigDataBench, which is a benchmark suite for benchmarking and ranking systems running big data applications. Then we explore the relative performance of the three implementation approaches with different big data applications, and provide strong guidance for the data center system construction.**

Through our experiments, we has inferred that a data center system based on specific hardware has different performance in the context of different applications and data volumes. When we construct a system, we can take into account not only the performance or energy consumption of the pure hardwares, but also the application-level characteristics. Data scale, application type and complexity should be considered comprehensively when researchers or architects plan to choose fundamental components for their data center system.

I. Introduction

We can hardly stop the world creating data, and it is not easy to estimate how much data is produced every day. According to **IDC** (International Data Corporation), the total amount of global data is expected to grow to 2.7 zettabytes during 2012. This is an increase of 48% from 2011 [1]. The rapid growth of data volumes brings big challenges to the construction of data center platforms, and makes lots of existing solutions no longer applicable. In response to this trend, both academia and industry are looking for innovative ways to deal with large-scale data. The distributed cluster system is a kind of solutions which is adopted by many organizations. However, data volumes are growing to the point where companies are being forced to strengthen their infrastructure, for example, by adding more nodes or replacing with more powerful hardware. The infrastructure costs and energy consumption skyrocket along this way. Then, how to select proper hardware for data center construction, and how to find a balance between processing capacity and energy costs become the questions which are urgently to be solved.

Because of architectural difference, configuration variance, and other factors, different hardware have their own processing specialities, which would make them show diverse behaviors when dealing with types of applications. For example, Xeon series are equipped with higher processing frequency, which makes Xeon commonly applied to some scenarios involving complex calculations. While, Atom series usually appear in some scenarios which treat low power as the key. Architects are searching for a approach to take good use of different kinds of hardware resources to construct a balanced system. And finding appropriate hardware fundamental components for the workloads counts for much more in data center construction.

In this paper, we analyze the processing performance of different big data systems through driving them with various workloads, and then to find some implications on the relationship between applications and data center system. We select a big data benchmark suite – **BigDataBench** [2] to evaluate three kinds of big data systems – **Xeon**, **Atom**, and **Tilera**. Through experimental results collected, we find that the merits of a system is not only determined by hardware equipment, but also relative to the application type and the data volume running on it. A system may be going to be favorable for some applications, and show both better energy performance and processing capacity when handling certain workloads within specific range of data volume. So, when architects choose fundamental components for their systems, they can not only consider the performance or energy consumption of hardware itself, but also need to regard the application type, data volume and the complexity of the application the system will run for.

The remainder of the paper is organized as follows. Section 2 introduces our evaluation methodology. Section 3 elaborates the basic analysis on hardware platforms and workloads. Section 4 discusses the experimental results based on three big data systems, and carries on the comparative analysis to the processing result. Section 5 concludes the paper.

II. Evaluation Methodology

In this section, we will give the brief information about the hardware platforms we evaluate, the Benchmark we use, and

978-1-4799-1293-3/13/$31.00 ©2013 IEEE

TABLE I
THE BASIC CONFIGURATION OF XEON

CPU Type	Intel ®Xeon E5310
CPU Core	4 cores @ 1.6GHz
L1 I/D Cache	32KB
L2 Cache	512KB

TABLE II
THE BASIC CONFIGURATION OF ATOM

CPU Type	Intel ®Atom D510
CPU Core	2 cores @ 1.66GHz
L1 I/D Cache	24KB
L2 Cache	512KB

the metrics we choose.

A. Experimental Platforms

We choose platforms based on their popularity in industry and academia. Xeon and Atom are widely used in many academic institutions and internet service providers, and Tilera is a kind of burgeoning CPU. What's more, these three platforms have great differences on their architecture. The basic informations of them are showed in TABLE I, TABLE II, and TABLE III. We construct different-sized system using these hardware, then deploy Hadoop on them. Hadoop is an open-source software for reliable, scalable, distributed computing [3], and it is adopted in data center system construction by many companies, such as Facebook [4], IBM [5], NREL [6], etc. The Hadoop version we used is Hadoop-1.0.2. And three clusters are supplied with same Hadoop setting referring to the guidance on Hadoop official website.

To be fair to comparison, we guarantee at least one dimension of two systems to be the same. For Xeon and Atom, we construct the cluster on them with one master and seven slaves, while, for Tilera, the system is consisted of one master and one slaves. There are 4 cores in a Xeon CPU with one hardware thread per core, and 2 cores in an Atom CPU with two hardware threads per core. So, the Xeon and Atom system have the same hardware thread number. The Tilera CPU we used has 36 tiles (core is named as tile in Tilera), we closed 8 cores of it to make this platform have same core number as Xeon.

B. Benchmark Selection

We attempt to find some underlying relations between workloads and different big data systems in this paper, especially the impacts of the application type, and data volume on different platforms. So, we need a benchmark, which can provide typical applications, and can offer data sets of any volume. BigDataBench meets these all.

BigDataBench is a big data benchmark suite from web search engines. It provides six kinds of applications which are typically employed in search engines, including Sort, Grep, Wordcount, Naive Bayes, SVM and search [7]. Because the search application are effected by data volume as well as submission interval which we don't care about in this paper, we don't adopt this application in our experiments. And then, applications like Sort, Grep, Wordcount, have disparate execution behaviors compared with Naive Bayes and SVM – Naive Bayes and SVM have higher computation complexity than the others. This will help us obtain more comprehensive results. BigDataBench also provides a data generation tool to overcome difficulties of obtaining real big data. This tool generates big data based on small-scale data while preserving the key characteristics of real data. This can make our experimental results more reasonable.

TABLE III
THE BASIC CONFIGURATION OF TILERA

CPU Type	Tilera ®TilePro36
CPU Core	36 cores @ 500MHz
L1 I/D Cache	16KB/8KB
L2 Cache	64KB

C. Metrics

We call for some metrics which can be directly perceived, and be compared and gotten easily. Also, these metrics should reflect the integrated processing capacity of a data center. We evaluate the performance of data center systems from two aspects – processing capacity and energy consumption, and the corresponding metrics we adopt are the **DPS** (Data processed per second) and the **DPJ** (Data processed per joule) [8]. These two metrics aren't inclined to the computing power of the CPU or bandwidth of I/O. They place emphasis on estimating the processing capacity of the whole system. We record the volume and run time of workloads, and use a power meter to measure energy consumption, then calculate the DPS/DPJ according to formulas 1 and 2 below:

$$DPS = \frac{Data\ Input\ Size}{Run\ Time} \tag{1}$$

$$DPJ = \frac{Data\ Input\ Size}{Energy\ Comsumption} \tag{2}$$

III. ELABORATION OF HARDWARE PLATFORMS AND WORKLOADS

In this part, we will analyze some basic details of hardware and workloads to lay the foundation for the analysis in section 4.

A. Hardware Analysis

1) Xeon: The hardware configuration of Xeon is showed in TABLE I. In the field of industry, Xeon usually is used to deal with resource-intensive applications, like JAVA or PHP, which is different from Atom. Xeon E5310 is based on Intel Core Microarchitecture [9] which also treats energy efficiency as a design goal. The chip we used has 4 cores per CPU with 1.6GHz basic frequency, supporting multiple arithmetic and many instruction set such as MMX, SSE and etc. TDP (Thermal Design Power) of the chip is 80W. This model does not support Hyper-Threading technology which means it has one hardware thread per core. It adopts the Intel SpeedStep technology to adjustment the power according to the demand.

2) Atom: Atom is often used in some lightweight tasks, like web, Apache, and some real-time applications. All of these tasks split a problem into many pieces in order to leverage each core's processing power. The configuration information is showed in TABLE II. Atom D510 is based on Intel Pine Trail architecture [10], supporting hyper-threading technology. The TDP of Atom is 13W. Atom processor is based on a micro-architecture different from mainstream ones, which uses the In-Order Execution to reduce power consumption. However, this is bound to affect processing performance. Therefore, Atom core introduce the hyper-threading (HT) technology, which means the core is equipped with dual-threaded executions. The adoption of HT technology improves parallelism and make up for the lack of powerful execution architecture.

3) Tilera: Tilera is a Many-Core processor for cloud applications such as memcached, media, and Hadoop [11]. In order for the tiles (cores) to communicate with each other and to I/O device, the Tile Processor Architecture provides a communication fabric called the **iMesh** which differ from **BUS** in Intel Architecture. Tilera also aims at lower power efficiency to adapt to massive-scale clusters. The core number of Tilera series ranges from 16 to 100. The information of Tilera we used is showed in TABLE III. It has 36 cores on chip with integrated cache, supported by 2 dimensional on-chip mesh network, and, the tiles (cores) can be closed if need. The TDP of TilePro36 is 16W, and Tilera does not support floating point operations.

B. Workloads Analysis

We choose five out of six applications from BigDataBench. For each application, the core operations are different. From paper [13] and [14], I/O wait time means the time spent by CPU waiting for I/O operations to complete. A high percentage of I/O wait time means that the application has I/O operations frequently, which further indicates that the application is an I/O intensive workload. Starting from this point, we describe each application.

- **Sort:** Sort simply uses the MapReduce framework to sort records within a directory. It is a representative I/O intensive application.
- **Wordcount:** Wordcount reads text files and counts how often the words occur. It is a representative CPU intensive application, accompanied with lighter network and disk load.
- **Grep:** There are two processes in grep, the first step is to find the words provided by users, and count the number of occurrences, the next step is to sort the words by their occurrences. In general, Grep is a CPU intensive application.
- **Naive Bayes:** Naive Bayes is a simple probabilistic classifier which applied the Bayes' theorem with strong (naive) independence assumptions. We simply select the classification step as our workloads rather than the training step. The main process is to calculate the probability, then decide the classification according to the existing model. It belongs to CPU intensive application.
- **SVM:** SVM is supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. SVM is the most complex application in all fives. The main process of it is vector calculus, and it is a typical CPU intensive application.

TABLE IV
DETAILS OF DIFFERENT ALGORITHMS

Application	Time Complexity	Characteristics
Sort	$O(n \times log_2 n)$	Integer comparison
WordCount	O(n)	Integer comparison and calculation
Grep	O(n)	String comparison
Naive Bayes	$O(m \times n)$	Floating-point computation
SVM	$O(n^3)$	Floating-point computation

IV. RESULTS AND ANALYSIS

In this part, we will report the experimental data collected in general, then analyze the performance difference of these big data systems through comparison.

A. General Description

To decide proper dataset to drive these systems, we initially run the experiments using datasets with 500MB and 100GB, to get approximate processing capacity of system. During the experiments, we find that the SVM is the most time-consuming application, and the variation of its DPS/DPJ is very small, so we determine to run the SVM with maximum 25G on Xeon and Atom. From paper [8], we know that the floating point operations in SVM is the most among five applications. Tilera has no floating-point unit. It will change the floating point operation into some basic operations, then it will cost more time. Considering this, we don't adopt the SVM in the Tilera testing. In order to aggrandize the stability of our results, for every experiment, we run at least two times, taking the average value to eliminate bias caused by uncertainty factors as possible. Normally, the run time of the same workload are similar, however, there are existing some abnormal results, such as one execution time is much higher than the others. These may be caused by Hadoop abnormal execution – having too many failed tasks, or the interference from the operating system self-check. For these kinds of results, we don't take it into our mean value calculation.

Fig. 1 to Fig. 6 display the final DPS and DPJ of three big data systems. From these figures, we have following observations. Firstly, Grep has the highest DPS/DPJ value, then Wordcount, Sort, and Naive Bayes. SVM has the smallest ones. Secondly, for most applications, along with an increase in the amount of input data, the DPS/DPJ first rise, and then keep stable, thereamong, the curve of Sort appears relatively obvious downtrend when the data volume arrive to a certain value, and the variation of SVM curve is minimal. Last, the profile of the DPS and the DPJ curve of one system are similar. Further, we find that the performance indicated by the

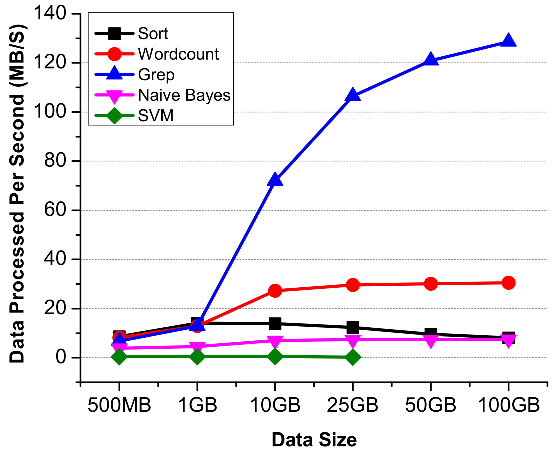

Fig. 1. The DPS of Xeon Platform

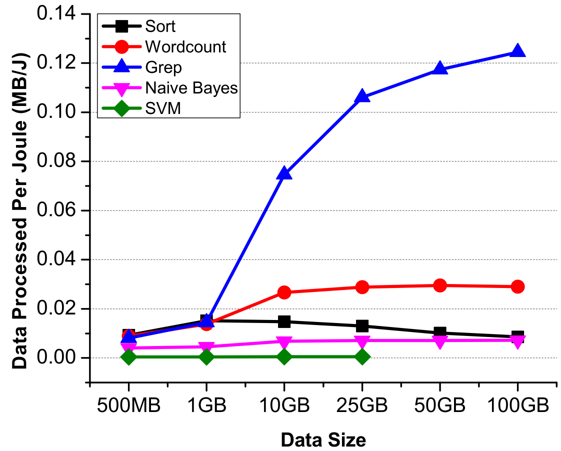

Fig. 2. The DPJ of Xeon Platform

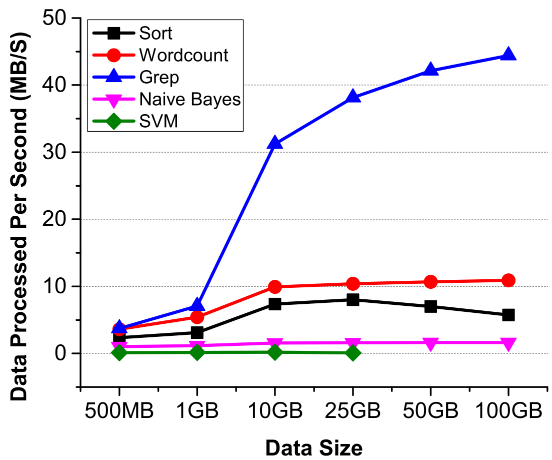

Fig. 3. The DPS of Atom Platform

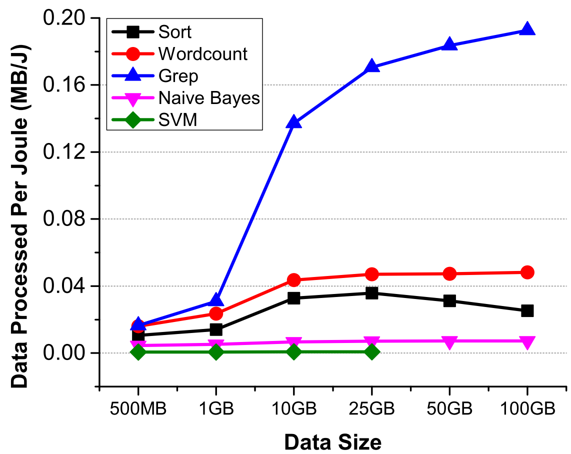

Fig. 4. The DPJ of Atom Platform

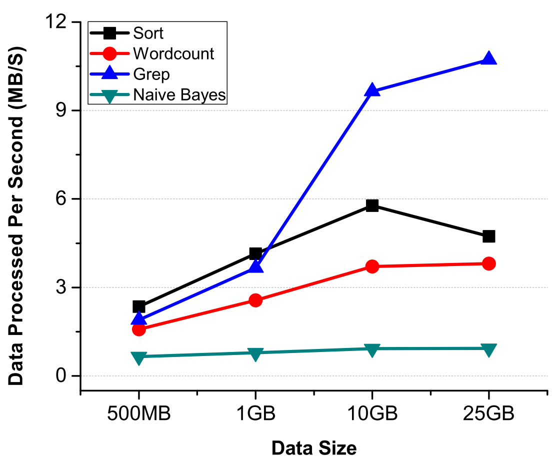

Fig. 5. The DPS of Tilera Platform

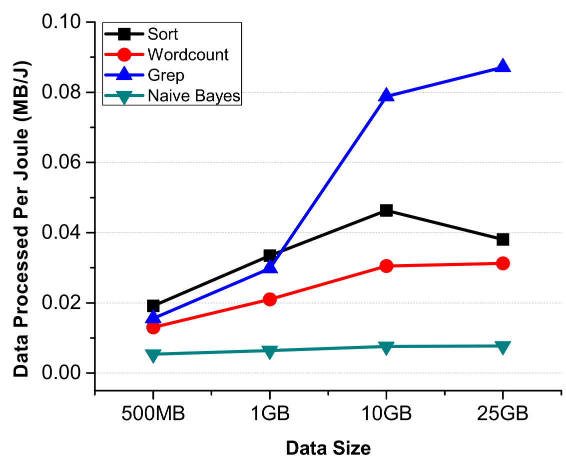

Fig. 6. TThe DPJ of Tilera Platform

DPS/DPJ and the growth rate of the curves are closely linked with the complexity of the application. TABLE IV records the computation complexity of the five applications. Basically, the easier the application is, the higher its DPS/DPJ are. Sort is a typical I/O-intensive application, and its processing capability trend is mostly impacted by the I/O operations. The large percentage of I/O wait time elongates the Sort's execution time, then reduces the processing capability. While, for Wordcount and Grep, they are not I/O-intensive application, when the system's resource is fully used, the processing capability is unchanged. For the complex applications like SVM, Naive Bayes, the changes of the DPS/DPJ is not obvious with the increasing of data volume, this may be caused by the usage of CPU. For this kind of applications, CPU is always treated as bottleneck, the computational resources are fully used when even running small dataset. Observing the format 1 and 2, we know that, for the DPS/DPJ, execution time is the most influential factor. This explains why the curve of the DPS and the DPJ are similar. Then we dissect the performance difference through comparison.

B. The Comparison of Xeon and Atom

Both of Xeon and Atom platforms consist of one master and seven slaves, and the hardware thread number of two systems are the same (the details of hardware can be found in TABLE I and TABLE II).

Basic trends of two systems are showed in Fig 1 to 4. Then Fig. 7 and Fig. 8 demonstrate the **DPS** and the **DPJ** comparison of Xeon and Atom. Obviously, there is a big gap between Xeon and Atom on processing capacity. The application type and data volume do an enormous influence on the DPS for both platforms. For Sort, the distance between the DPS of Xeon and Atom comes closer as the volume of data increases, yet, for Wordcount, Grep, Naive Bayes, it comes farer with the increasing process. SVM shows relatively stable state. We can also see this from TABLE V. TABLE V shows the ratio of Xeon's DPS/DPJ and Atom's. We can know that the processing capacity of Xeon basically keeps about 2 to 4 times higher than Atom at most of the time. For Sort, Wordcount, Grep, the ratio changes variably under different data size, and for Naive Bayes and SVM, the variation of ratio varies in a small range. The reason why the DPS/DPJ present such trend may like this. Sort is a I/O intensive application mentioned above. The increment of data volume will give rises to the stress of I/O, which weaken the effects from the CPU processing capacity. To Wordcount, Grep, Naive Bayes, when using small dataset, CPU resources are not fully utilized, so the difference between the two DPS is small. After increasing the amount of data, CPU performance distance is reflected. The basic frequency of Xeon and Atom core are 1.6 GHz and 1.66 GHz, and their hardware thread number on per processor is the same. The performance difference mainly comes from the pipeline structure. For Xeon, it supports Out-Of-Order execution while Atom don't, and the work speed is optimized through the pipelining design which leads to better processing capacity. For SVM, the CPU utilization are

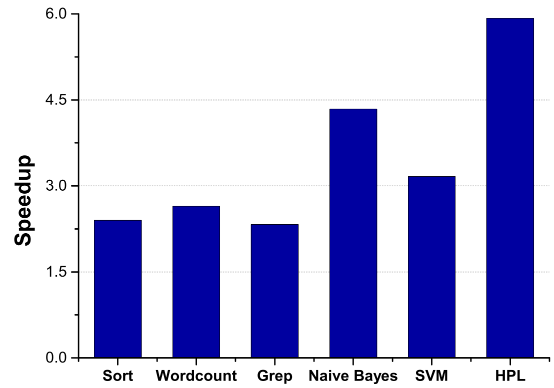

Fig. 11. The Speedup between Xeon and Atom

high from the beginning to both platforms which differ from the Sort application. When using 500M dataset, the CPU utilization of Sort only 59% on Atom, while SVM arrives to 87%. So, to the SVM, the volume of dataset doesn't have a big influence to the two systems' processing capacity distance.

In the meantime, we take the energy consumption into consideration. Atom is designed primarily for energy conservation. From the Fig. 8, to Sort, Wordcount, Grep, when the size of dataset reaches to a certain value (above 1GB or 10GB for our experimental platforms), power consumption advantages of Atom are reflected. This is because of the Xeon system hasn't been fully used under small data volume, and its energy consumption isn't high in this situation. To Naive Bayes, Atom does not show the energy strength. This is caused by too long execution time. Combined with TABLE V, we see the DPS of SVM on Xeon is more than 3 times higher than Atom's, and the DPJ of Atom is only 1.5 times higher than Xeon's all the time, which implies that the Atom is not suitable for such applications either. To further explain this, we adopt the HPL (High Performance Computing Linpack Benchmark) testing. We divide the execution time on Atom by the execution time on Xeon to get the speedup. The Fig. 11 shows the result. From the figure, we learn that the HPL has the highest speedup, then Naive Bayes and SVM. The larger value signifies that it will take more time to deal with this application. For example, it may take 4 times if using Atom rather than Xeon, even Xeon cost more energy, it take less time, the ratio of energy and time still may be lower. And that is the reason why the Atom don't reflect the energy advantage when handling the complex applications.

C. The Comparison of Xeon and Tilera

The Tilera processor we use has 36 tiles (cores). We close 8 tiles (cores) of it to guarantee the core number of two systems to be the same. Fig. 5 and Fig. 6 are the DPS and the DPJ of Tilera system. By observing the results, we find that on Xeon and Atom, the DPS of Wordcount is larger than Sort, while on Tilera, the situation is just the opposite. As we talked

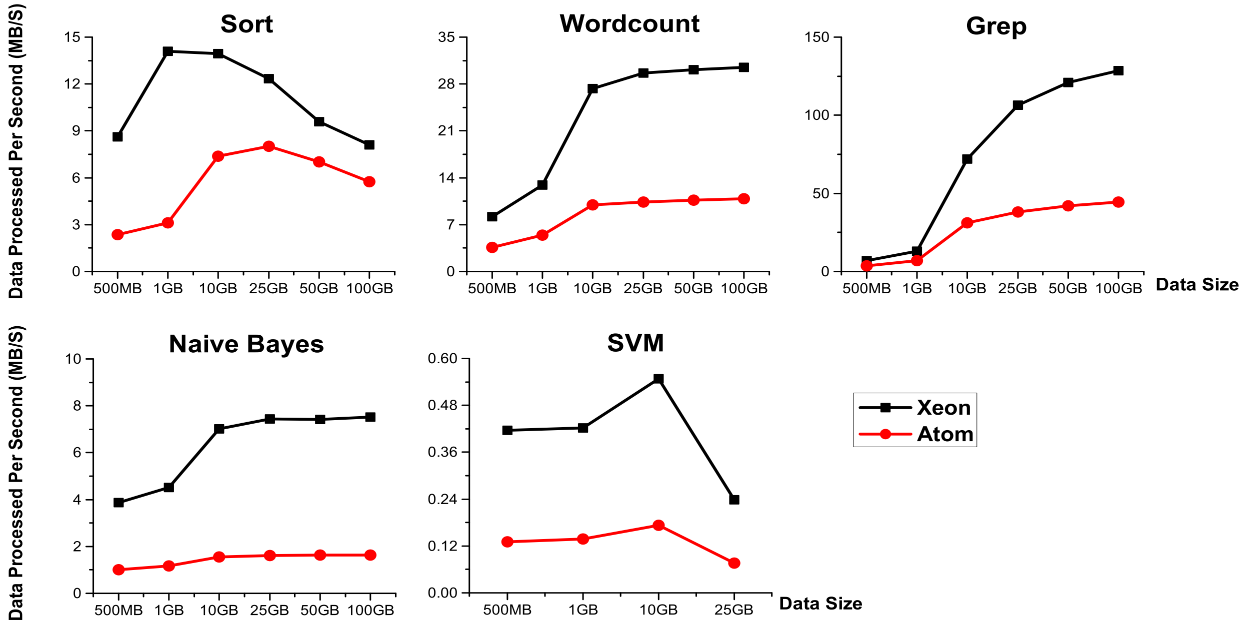

Fig. 7. The Comparison of DPS between Xeon and Atom

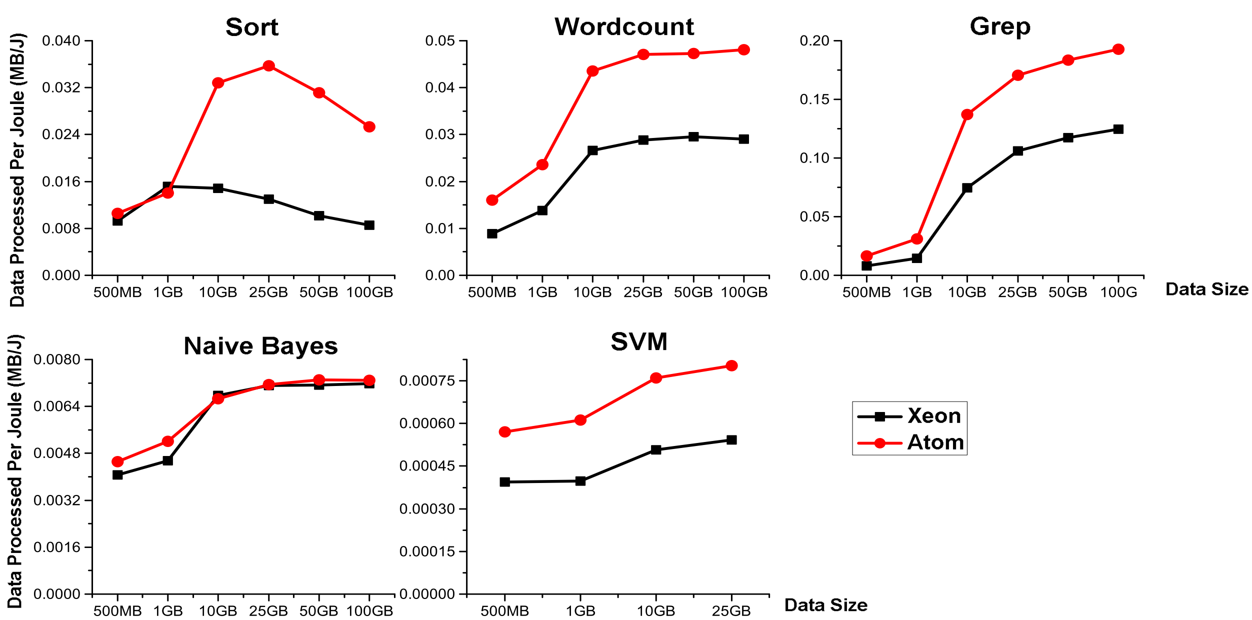

Fig. 8. The Comparison of DPJ between Xeon and Atom

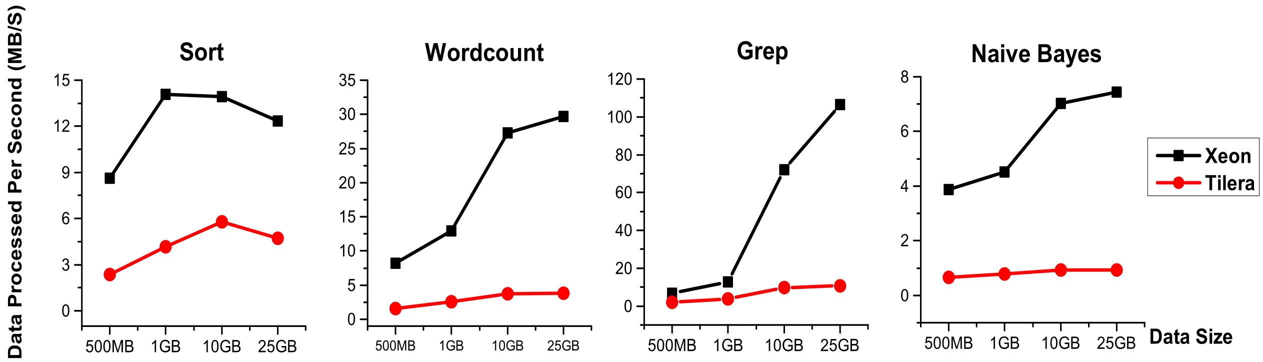

Fig. 9. The Comparison of DPS between Xeon and Tilera

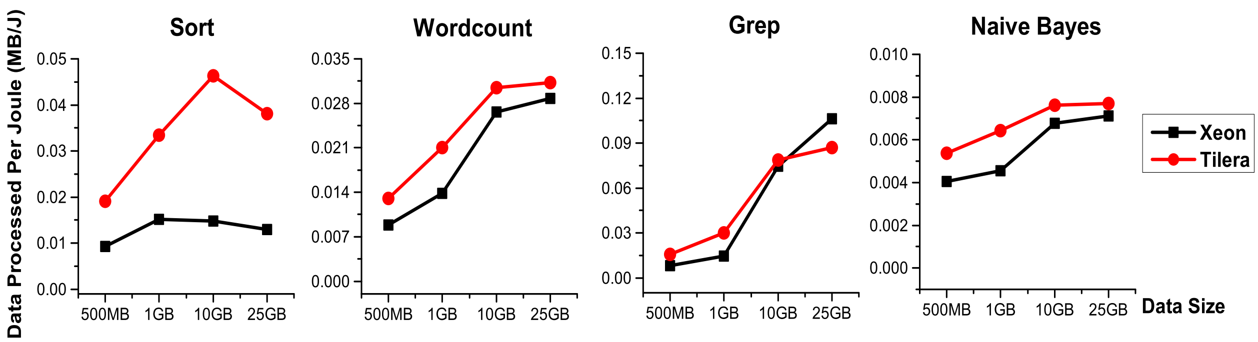

Fig. 10. The Comparison of DPJ between Xeon and Tilera

TABLE V
THE RATIO OF XEON/ATOM

		500MB	1GB	10GB	25GB	50GB	100GB
Sort	DPS	3.67	4.51	1.89	1.54	1.36	1.40
	DPJ	0.87	1.08	0.45	0.36	0.32	0.33
Wordcount	DPS	2.27	2.38	2.74	2.84	2.82	2.79
	DPJ	0.55	0.58	0.61	0.61	0.62	0.60
Grep	DPS	1.83	1.82	2.30	2.79	2.87	2.89
	DPJ	0.48	0.46	0.54	0.62	0.63	0.64
Naive Bayes	DPS	3.83	3.89	4.52	4.64	4.54	4.58
	DPJ	0.89	0.87	1.01	0.99	0.97	0.90
SVM	DPS	3.19	3.06	3.17	3.14		
	DPJ	0.69	0.64	0.66	0.67		

TABLE VI
THE RATIO OF XEON/TILERA

		500MB	1GB	10GB	25GB
Sort	DPS	3.67	3.39	2.41	2.60
	DPJ	0.48	0.45	0.31	0.34
Wordcount	DPS	5.19	5.04	7.35	7.78
	DPJ	0.67	0.65	0.87	0.92
Grep	DPS	3.60	3.52	7.45	9.93
	DPJ	0.51	0.48	0.94	1.21
Naive Bayes	DPS	5.91	5.78	7.59	7.94
	DPJ	0.75	0.70	0.89	0.92

above, the Sort is a typical I/O intensive application, so the cache size will make a big difference on its execution time. Tilera integrate 36 tiles (we use 28 tiles), and each tile are equipped with 16KB instruction cache, 8KB data cache, and 64KB L2 cache. More, it provides virtual L3 cahce, which means, if misses in the L2 cache on a certain tile are satisfied by caches in other tiles, it will get this data, otherwise, Tilera will fetch data from external memory and deliver it to the requesting core. This structure makes Tilera have a more flexible cache access strategy, then makes Tilera more suitable for I/O intensive application, like Sort. Wordcount is a CPU intensive application, and Tilera is weak on CPU processing capacity. This may cause the difference comparing with Xeon and Atom.

From the Fig. 9 and Fig. 10, we obtain that the processing capacity of Xeon is better than Tilera, while Tilera has better energy performance. Nevertheless, except Sort, the DPS of Xeon is around 6 times on average than the DPS of Atom, and the DPS of Atom is slightly more than one time than that of Xeon. TABLE VI shows the ratio of Xeon's DPS/DPJ and Tilera's. This implies that the Tilera is more appropriate for I/O intensive applications rather than CPU intensive ones. Tilera combines the low-power consumption of slower clock speeds with the increased throughput of many independent cores [4],

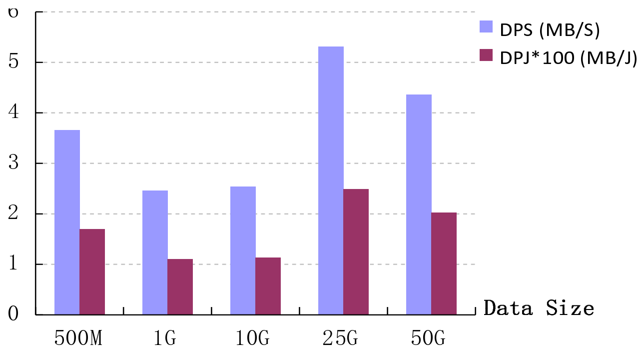

Fig. 12. The Exception of Sort on Atom Platform

which can be used to illustrate the Tilera's processing features.

D. Supplementary Analysis

Above all, we can see that the Xeon cluster has definitely processing advantage in three clusters. We can also deduce that the application behaviors affect the use of system resource, which lead to the different performance when dealing with diverse workloads. From the TABLE IV, we can see the main operation of SVM and Naive Bayes are all floating-point computation. Atom which don't support OoO execution and the Tilera which has no floating computing component are hard to run such applications, even the Tilera have better mechanism for core connecting which may reduce the time costing on the machines communication.

During our experiments, for example, the Sort experiments on Atom, we found an unusual phenomenon which is showed in Fig. 12. For Sort curve, it at first appears a downward trend, then begins to emerge a upward trend. This seems to be inconsistent with the conventional. The phenomenon may be caused by data skew or Hadoop task failure. We regenerate the testing dataset and re-experiment, then it backs to normal. Hadoop is a distribute framework, the data skew will lead to the unbalance of workload for each slave in cluster, then influence execution time. We can solve this problem by regenerating new dataset, or using a cluster with single slave when testing a system.

V. Conclusion

In this paper, we have evaluated three big data systems. Based on the above analysis, we have concluded that different hardware have their own processing features. A big data system consisted of specific hardware has various performance when dealing with types of applications and data sets of different volume. In some cases, a big data system may have both better processing capacity and energy consumption. When architects plan to construct big data systems or choose fundamental components for a system, they should not only concentrate on the processing capacity and energy consumption of hardware itself, but also need to regard the application type, data volume and the complexity of the application that the system will handle with.

More specifically, through our experimental results, we have shown Xeon generally has better processing capacity accompanied with high energy consumption, especially to some light scale-out applications like Sort, Wordcount, Grep. For some complex applications like Naive Bayes, SVM, the weaker processing capability of processor causes long execution time. Even the power of them is lower than Xeon, the total energy consumption is still high, which further illustrate that the energy performance is relative to the application type. From the perspective of energy consumption, Tilera exerts a energy advantage on processing I/O intensive application like Sort, while Atom is more excellent in processing simple CPU intensive application like Wordcount and Grep.

Acknowledgment

We are grateful to our mentors for the many informative discussions that guided this work and the anonymous reviewers, who provided excellent feedback. This work is supported by the Chinese 973 project (Grant No.2011CB302502), the NSFC project (Grant No.60933003, 61202075), and the BNSF project (Grant No.4133081).

References

[1] Shirer, M., & Murray, S. (2011). IDC Predicts 2012 Will Be the Year of Mobile and Cloud Platform Wars as IT Vendors Vie for Leadership While the Industry Redefines Itself, International Data Corporation (IDC).
[2] http://prof.ict.ac.cn/BigDataBench/
[3] http://hadoop.apache.org/
[4] Berezecki M, Frachtenberg E, Paleczny M, et al. Many-core key-value store. Green Computing Conference and Workshops (IGCC), 2011 International. IEEE, 2011: 1-8.
[5] http://www-01.ibm.com/software/data/infosphere/hadoop/
[6] http://hpc.nrel.gov/systems
[7] Wanling Gao, Yuqing Zhu, Zhen Jia, Chunjie Luo, Lei Wang, Jianfeng Zhan, Yongqiang He, Shiming Gong, Xiaona Li, Shujie Zhang, and Bizhu Qiu. BigDataBench: a Big Data Benchmark Suite from Web Search Engines. The Third Workshop on Architectures and Systems for Big Data(ASBD 2013) in conjunction with The 40th International Symposium on Computer Architecture, May 2013.
[8] Chunjie Luo, Jianfeng Zhan, et al. CloudRank: Benchmarking and Ranking Private Cloud Computing System for Data Processing Applications. Institute of Computing Technology, Chinese Academy fo Sciences.
[9] Inside Intel Core Microarchitecture and Smart Memory Access. White Paper of Inter Company
[10] http://www.legitreviews.com/intel
[11] Richard Schooler. Tilera Many-Core Processors for Cloud Applications. Tilera Corp.2011/12/2
[12] Bell, S., Edwards, B., Amann, J., Conlin, R., Joyce, K., Leung, V., et al. (2008, February). Tile64-processor: A 64-core soc with mesh interconnect. In Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International (pp. 88-598). IEEE.
[13] Jia, Z., Zhou, R., Zhu, C., Wang, L., Gao, W., Shi, etc. The Implications of Diverse Applications and Scalable Data Sets in Benchmarking Big Data Systems. Lecture Node in Computer Science. 2013.
[14] Jia, Zhen. and Wang, Lei. and Zhan, Jianfeng. and Zhang, Lixin and Chunjie Luo. Characterizing Data Analysis Workloads in Data Centers. 2013 IEEE International Symposium on Workload Characterization (I-ISWC 2013).
[15] Diane Bryant, Jason Waxman. Introducing the First Datacenter Atom SOC. December 11, 2012.