I/O Characterization of Big Data Workloads in Data Centers

Fengfeng Pan[1,2(✉)], Yinliang Yue[1], Jin Xiong[1], and Daxiang Hao[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
panfengfeng@ncic.ac.cn

Abstract. As the amount of data explodes rapidly, more and more organizations tend to use data centers to make effective decisions and gain a competitive edge. Big data applications have gradually dominated the data centers workloads, and hence it has been increasingly important to understand their behaviour in order to further improve the performance of data centers. Due to the constantly increased gap between I/O devices and CPUs, I/O performance dominates the overall system performance, so characterizing I/O behaviour of big data workloads is important and imperative.

In this paper, we select four typical big data workloads in broader areas from the BigDataBench which is a big data benchmark suite from internet services. They are Aggregation, TeraSort, Kmeans and PageRank. We conduct detailed deep analysis of their I/O characteristics, including disk read/write bandwidth, I/O devices utilization, average waiting time of I/O requests, and average size of I/O requests, which act as a guide to design highperformance, low-power and cost-aware big data storage systems.

1 Introduction

In recent years, big data workloads [20] are more and more popular and play an important role in enterprises business. There are some popular and typical applications, such as TeraSort, SQL operations, PageRank and K-means. Specifically, TeraSort is widely used for page or document ranking; SQL operations, such as join, aggregation and select, are used for log analysis and information extraction; PageRank is widely used in search engine field; and Kmeans is usually used as electronic commerce algorithm.

These big data workloads run in data centers, and their performance is critical. The factors which affect their performance include: algorithms, hardware including node, interconnection and storage, and software such as programming model and file systems. This is the reason for why several efforts have been made to analyse the impact of these factors on the systems [17,20]. However, data access to persistent storage usually accounts for a large part of application time because of the ever-increasing performance gap between CPU and I/O devices.

J. Zhan et al. (Eds.): BPOE 2014, LNCS 8807, pp. 85–97, 2014.
DOI: 10.1007/978-3-319-13021-7_7

With the rapid growth of data volume in many enterprises and a strong desire for processing and storing data efficiently, a new generation of big data storage system is urgently required. In order to achieve this goal, a deep understanding of big data workloads present in data centers is necessary to guide the big data systems design and tuning.

Some studies have been conducted to explore the computing characteristics of big data workloads [16,18], meanwhile, lots of work also have been done to depict the storage I/O characteristics of enterprise storages [5,9]. However, to the best of our knowledge, none of the existing research has understood the I/O characteristics of big data workloads, which is much more important in the current big data era. So understanding the characteristics of these applications is the key to better design the storage system and optimize their performance and energy efficiency.

In this paper, we choose four typical big data workloads as mentioned above, because they have been widely used in popular application domains [1,20], such as search engine, social networks and electronic commerce. Detailed information about I/O metrics and workloads are shown in Sect. 3.2. Through the detailed analysis, we get the following four observations. First, the change of the number of task slots has no effects on the four I/O metrics, but increasing the number of task slots appropriately can accelerate the process of application execution. Second, increasing memory can alleviate the pressure of disk read/write, and effectively improve the I/O performance when the data size is large. Third, the compression of intermediate data mainly affects the MapReduce I/O performance and has little influence on HDFS I/O performance. However, compression consumes some CPU resource which may influence the job's execution time. Fourth, the I/O pattern of HDFS and MapReduce are different, namely, HDFS's I/O pattern is large sequential access and MapReduce's I/O pattern is small random access, so when configuring storage systems, we should take several factors into account, such as the number of devices and the types of devices.

The rest of the paper is organized as follows: Sect. 2 discusses related work. Section 3 describes the experimental methodology. Section 4 shows the experimental results. Section 5 briefly brings up future work and concludes this paper.

2 Related Work

Workloads characterization studies play a significant role in detecting problems and performance bottlenecks of systems. Many workloads have been extensively studied in the past, including enterprise storage systems [7,10], web server [15,19], HPC cluster [14] and network systems [12].

2.1 I/O Characteristics of Storage Workloads

There have been a number of papers about the I/O characteristics of storage workloads [8,11,15].

Kavalanekar et al. [15] characterized large online services for storage system configuration and performance modeling. It contains a set of characteristics, including block-level statistics, multi-parameter distributions and rankings of file access frequencies. Similarly, Delemitrou et al. [11] presented a concise statistical model which accurately captures the I/O access pattern of large-scale applications including their spatial locality, inter-arrival times and type of accesses.

2.2 Characteristics of Big Data Workloads

Big data workloads have been studied in recent years at various levels, such as job characterization [16–18], storage systems [5,9].

Kavulya et al. [16] characterized resource utilization patterns, job patterns, and source of failure. This work focused on predicting job completion time and found the performance problems. Similarly, Ren et al. [18] focused on not only job characterization and task characterization, but also resource utilization on a Hadoop cluster, including CPU, Disk and Network.

However, the above researches on big data workloads focused on job level, but not on the storage level. Some studies have provided us with some metrics about data access pattern in MapReduce scenarios [4,6,13], bur these metrics are limited, such as block age at time of access [13] and file popularity [4,6]. Abad et al. [5] conducted a detailed analysis about some HDFS characterization, such as file popularity, temporal locality, request arrival patterns, and then figure out the data access pattern of two types of big data workloads, namely batch and interactive query workloads. But this work concentrates on HDFS's I/O characterization, does not study the intermediate data, and also this work only involves two types of workloads.

In this paper, we focus on I/O characteristics of big data workloads, the difference between our work and the previous work is that first, we focus on the I/O behaviour of individual workload and choose four typical workloads in broader areas, including search engine, social network, e-commerce, which are popular in the current big data era; Second, we analyze the I/O behavior of both HDFS and MapReduce intermediate data from different I/O characteristics, including disk read/write bandwidth, I/O devices' utilization, average waiting time of I/O requests, and average size of I/O requests.

3 Experimental Methodology

This section firstly describes our experiment platform, and then presents workloads used in this paper.

3.1 Platform

We use an 11-node (one master and ten slaves) Hadoop cluster to run the four typical big data workloads. The nodes in our Hadoop cluster are connected through 1 Gb ethernet network. Each node has two Intel Xeon E5645 (Westmere)

Table 1. The detailed hardware configuration information

CPU Type	Intel R Xeon E5645
# Cores	6 cores@2.4 G
# threads	12 threads
Memory	32 GB, DDR3
Disk	**6 disks** (one disk for system, three disks for HDFS data, the other two disks for MapReduce intermediate data)
	Disk Model Seagate: ST1000NM0011
	Capacity: 1TB
	Rotational Speed: 7200 RPM
	Avg. Seek/Rotational Time: 8.5 ms/4.2 ms
	Sustained Transfer Rate: 150 MB/s

Table 2. The detailed software configuration information

Hadoop	1.0.4
JDK	1.6.0
Hive	0.11.0
OS Distribution and Linux kernel	Centos 5.5 with the 2.6.18 Linux kernel
TeraSort	BigDataBench2.1 [2]
SQL operations	BigDataBench2.1
PageRank	BigDataBench2.1
Kmeans	BigDataBench2.1

processors, 32 GB memory and 7 disks(1TB). A Xeon E5645 processor includes six physical out-of-order cores with speculative pipelines. Tables 1 and 2 shows the detailed configuration information.

3.2 Workloads and Statistics

We choose four popular and typical big data workloads from BigDataBench [20]. BigDataBench is a big data benchmark suite from internet services and it provides several big data generation tools to generate various types and volumes of big data from small-scale real-world data while preserving their characteristics. Table 3 shows the description of the workloads, which is characterized in this paper.

Iostat [3] is a well-used monitor tool used to collect and show various system statistics, such as CPU times, memory usage, as well as disk I/O statistics. In this paper, we mainly focus on the disk-level I/O behaviour of the workloads, and we extract information from iostat's report, and the metrics which we focus on are shown in Table 4.

Table 3. The description of the workloads

Workloads	Performance bottleneck	Scenarios	Input Data size
TeraSort (TS)	I/O bound	Page ranking; document ranking	1 TB
Aggregation (AGG)	CPU bound	Log analysis; information extraction	1 TB
K-means (KM)	CPU bound in iteration; I/O bound in clustering	Clustering and Classification	512 GB
PageRank (PR)	CPU-bound	Search engine	512 GB

Table 4. Notation of I/O characterization

I/O characterization	Description	Notes
rMB/s and wMB/s	The number of megabytes read from or written to the device per second	Disk Read or Write Bandwidth
%util	Percentage of CPU time during which I/O requests were issued to the device	Disk utilization
await (ms)	The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them	average waiting time of I/O request = await - svctm
svctm (ms)	The average service time for I/O requests that were issued to the device	
avgrq-sz (the number of sectors)	The average size of the requests that were issued to the device. And the size of sectors is 512B	average size of I/O request

4 Results

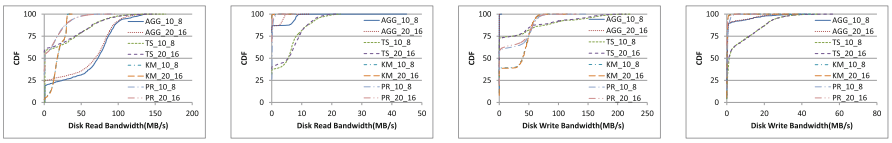
In this section, we describe HDFS/MapReduce I/O characteristics from four metrics, namely, disk read/write bandwidth, I/O devices' utilization, average waiting time of the I/O requests, and average size of the I/O requests. Through the comparison of each workloads, we can obtain the I/O characterization of HDFS and MapReduce respectively, and also the difference between HDFS and MapReudce.

In addition, different Hadoop configurations can influence the workloads' execution. So in this paper, we select three factors and analyse their impact on the I/O characterization of these workloads. The three factors are as follows. First, *the number of task slots, including map slots and reduce slots.* In Hadoop, computing resource is represented by slot, there are two types of slot: map task slot and reduce task slot. Here computing resource refers to CPU.

Second, *the amount of physical memory of each node*. As we know that memory plays an important role in I/O performance, so memory is an important factor which affects the I/O behaviour of workloads. So, the second factor we focus on is the relationship be-tween memory and I/O characterization. Third, *whether the intermediate data is compressed or not*. Compression involves two types of resources: CPU and I/O. what's the influence on the I/O behaviour of workloads when CPU and I/O resources both change. So, the final factor we focus on is the relationship between compression and I/O characterization.

4.1 Disk Read/Write Bandwidth

Task Slots. Figure 1 shows the effects of the number of task slots on the disk read/write bandwidth in HDFS and MapReduce respectively. In these experiments, each node configured 16 GB memory and the intermediate data is compressed. "10_8", "20_16" in the figures mean the number of map task slots and reduce task slots respectively.



(a) HDFS Read (b) MapReduce Read (c) HDFS Write (d) MapReduce Write

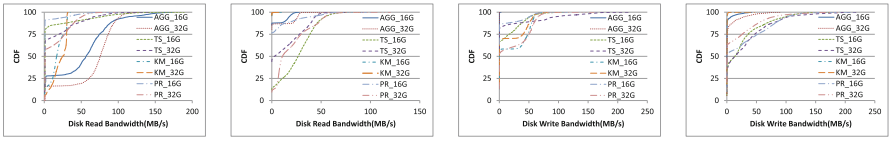
Fig. 1. The effects of the number of task slots on the Disk Read/Write Bandwidth in HDFS and MapReduce respectively

From Fig. 1 we can get the following two conclusions. First, when the number of task slots changes, there is barely any influence on the disk read/write bandwidth in both scenarios for every workload. Second, the variation of disk read/write bandwidth of different workloads in both scenarios are disparate because the data volume of each workload in different phases of execution are not the same.

In a word, there is little effect on disk read/write bandwidth when the number of task slots changes. However, configuring the number of task slots appropriately can reduce the execution time of workloads, so we should take it into account when workloads run.

Memory. Figure 2 displays the effects of the memory size on the disk read/write bandwidth in HDFS and MapReduce respectively. In these experiments, task slots configuration on each node is 10_8 and the intermediate data is not compressed. "16G", "32G" in the figures mean the memory size of node.

As Fig. 2 shows, the influence of the memory size on disk read/write Bandwidth depends on the data volume. There is a growth of disk read bandwidth



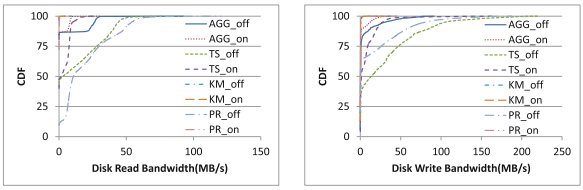
(a) HDFS Read (b) MapReduce Read (c) HDFS Write (d) MapReduce Write

Fig. 2. The effects of memory on the Disk Read/Write Bandwidth in HDFS and MapReduce respectively

in HDFS when memory increases as shown in Fig. 2(a) due to the large amount of raw data, but after handling and processing the raw data, the disk write band-width of each workloads in HDFS are different because of the final data volume. When the final data volume is small, memory has no effects on the disk write bandwidth, such as Kmeans, as shown in Fig. 2(c). This result can also be reflected in MapReduce.

Compression. Figure 3 exhibits the effects of intermediate data compression on the disk read/write bandwidth in MapReduce. In these experiments, each node configured 32 GB memory and task slots configuration is 10_8. "off", "on" in the figures mean whether the inter-mediate data is compressed or not.

As Fig. 3 shows, due to the reduction of intermediate data volume with compression, the disk read/write bandwidth increase in MapReduce.



(a) MapReduce Read (b) MapReduce Write

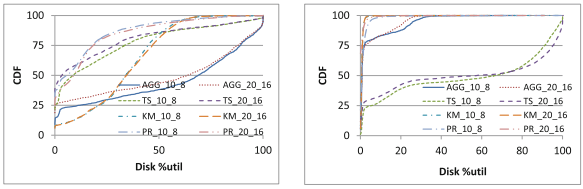
Fig. 3. The effects of compression on the Disk Read/Write Bandwidth in MapReduce.

In addition, compression has little impact on the HDFS's disk read/write bandwidth, so we do not present the result.

4.2 Disk Utilization

Task Slots. Figure 4 depicts the effects of the number of task slots on the disk utilization in HDFS and MapReduce respectively.

From Fig. 4 we can get the following two conclusions. First, the trends of workloads in disk utilization are the same when the number of task slots changes,



(a) HDFS (b) MapReduce

Fig. 4. The effects of the number of task slots on the Disk Utilization in HDFS and MapReduce respectively.

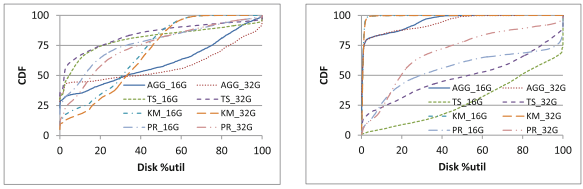
Table 5. The HDFS/MapReduce Disk %util ratio

	>90 %uitl	>95 %uitl	>99 %util
AGG	22.6 % / 0	16.4 % / 0	9.8 % / 0
TS	5.2 % / 27.2 %	3.8 % / 15.6 %	2.4 % / 5.5 %
KM	0.4 % / 0	0.3 % / 0	0.2 % / 0
PR	0.5 % / 0.1 %	0.3 % / 0.1 %	0.2 % / 0.1 %

i.e. the number of task slots has little impact on the disk utilization in both scenarios. Second, workloads have different behaviour about disk utilization in both scenarios. From the Table 5, the HDFS disk utilization of Aggregation is higher than the others, so Aggregation HDFS disk may be the bottleneck. Similarly, the MapReduce disk utilization of TeraSort is higher than the others; From the Fig. 4(b), the MapReduce disk utilization of workloads is 50 % or less at their most of execution time, so the disks are not busy, except TeraSort because of the large amount of TeraSort's intermediate data.

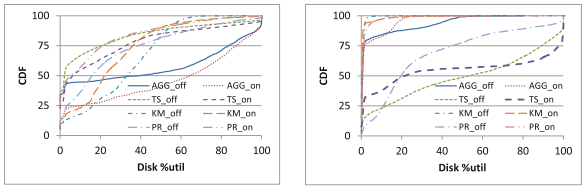
Memory. Figure 5 depicts the effects of the memory size on the disk utilization in HDFS and MapReduce respectively. As Fig. 5(a) shows, increasing memory size has no impact on the disk utilization in HDFS. However, from Fig. 5(b) we can see that there is a difference between HDFS and MapReduce. In MapReduce, the disk utilization of Aggregation and Kmeans has no changes when memory size changes because the devices are not busy before memory changes. However, the disk utilization of TeraSort and PageRank is reduced when memory increases as shown in Fig. 5(b). So, increasing memory size can help reduce the number of I/O requests and ease the bottleneck of disk.

Compression. From Fig. 6(a), we can know that when intermediate data is compressed, the HDFS's disk utilization essentially unchanges. As Fig. 6(b) shows, there is no influence on intermediate data's disk utilization of TeraSort, Aggregation and Kmeans, except PageRank in MapReduce.



(a) HDFS (b) MapReduce

Fig. 5. The effects of memory on the Disk Utilization in HDFS and MapReduce respectively.



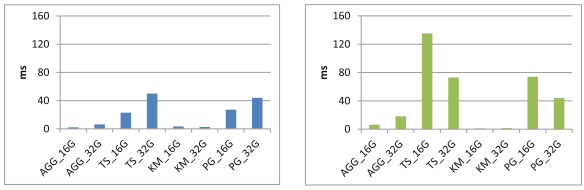
(a) HDFS (b) MapReduce

Fig. 6. The effects of compression on the Disk Utilization in HDFS and MapReduce respectively.

4.3 Average Waiting Time of I/O Request

Memory. Figure 7 shows the effects of Memory on the disk waiting time of I/O requests in HDFS and MapReduce respectively.

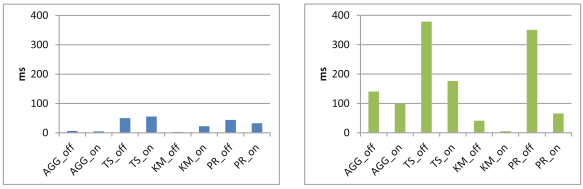
From Fig. 7, we can learn that the disk average waiting time of I/O requests of workloads varies with different memory size, in other words, the memory size has an impact on the disk waiting time of I/O requests and the MapReduce disk waiting time of I/O request is larger than the HDFS's.



(a) HDFS (b) MapReduce

Fig. 7. The effects of memory on the Disk waiting time of I/O requests in HDFS and MapReduce respectively.

Compression. Figure 8 depicts the effects of compression on the disk waiting time of I/O requests in HDFS and MapReduce respectively.



(a) HDFS (b) MapReduce

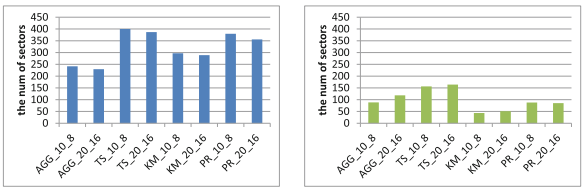
Fig. 8. The effects of compression on the Disk waiting time of I/O requests in HDFS and MapReduce respectively.

From Fig. 8, we can see that the disk average waiting time of I/O requests remains unchanged in HDFS because HDFS's data is not compressed, however, due to the reduction of intermediate data volume with compression, the disk waiting time of I/O request in MapReduce is decreased. And the MapReduce disk waiting time is larger than the HDFS's because of their different I/O mode in access pattern, i.e. HDFS's access pattern is domated by large sequential accesses, while MapReduce's access pattern is dominated by smaller random access. This result can be seen in Fig. 9.

4.4 Average Size of I/O Requests

Task Slots. Figure 9 depicts the effects of the number of task slots on the disk average size of I/O request in HDFS and MapReduce respectively.

As the task slots is a kind of computing resource, there is little impact on the disk average size of I/O requests when the number of task slots changes from the figures. Also, the average size of HDFS I/O requests is larger than the MapReduce's because they have different I/O mode in I/O granularity. In other words, HDFS's I/O granularity is larger than the MapReduce's.



(a) HDFS (b) MapReduce

Fig. 9. The effects of the number of task slots on the Disk average size of I/O request in HDFS and MapReduce respectively.

The same result also can be achieved by the effects of memory on the disk average size of I/O requests. However, due to the compression, the effects of memory on the disk average size of I/O requests is different from the effects of the number of task slots on the disk average size which is reflected in Fig. 10.

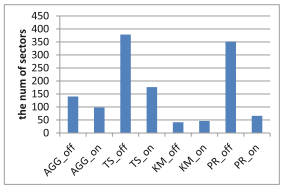


Fig. 10. The effects of compression on the Disk average size of I/O request in MapReduce.

Compression. In addition, whether the intermediate data is compressed or not has no impact on the HDFS's disk average size of I/O requests, so we do not present the result.

Figure 10 displays the effects of compression on the disk average size of I/O requests in MapReduce. It is seen from figure that as the intermediate data is compressed, the disk average size of I/O requests is decreased, and the percentage of reduction varies with the types of workloads due to the intermediate data volume. As Fig. 10 shows, there is little influence on the disk average size of I/O request when the intermediate data volume is small, such as Aggregation and Kmeans.

5 Conclusion and Future Work

In this paper, we have presented a study of I/O characterization of big data workloads. These workloads are typical, which are representative and common in search engine, social networks and electronic commerce. In contrast with previous work, we take into account disk read/write bandwidth, average waiting time of I/O requests, average size of I/O requests and storage device utilization, which are important for big data workloads. Some observations and implications are concluded as follows. First, task slots has little effects on the four I/O metrics, but increasing the number of task slots can accelerate the process of application execution. Second, the compression of intermediate data mainly affects the MapReduce I/O performance and has little influence on HDFS I/O performance. However, compression consumes some CPU resource which may influence the job's execution time. Third, increasing memory can alleviate the pressure of disk read/write and effectively improve the I/O performance when the data size is large. Last, HDFS data and MapReduce intermediate data have different I/O mode, which leads us to configuring their own storage systems according to their I/O mode.

Acknowledgement. This paper is supported by National Science Foundation of China under grants no. 61379042, 61303056, and 61202063, and Huawei Research Program YB2013090048.

References

1. http://www.alexa.com/topsites/global
2. http://prof.ict.ac.cn/BigDataBench/
3. http://linux.die.net/man/1/iostat
4. Abad, C.L., Lu, Y., Campbell, R.H.: Dare: adaptive data replication for efficient cluster scheduling. In: 2011 IEEE International Conference on Cluster Computing (CLUSTER), pp. 159–168 (2011)
5. Abad, C.L., Roberts, N.: A storage-centric analysis of mapreduce workloads: file popularity, temporal locality and arrival patterns. In: 2012 IEEE International Symposium on Workload Characterization (IISWC), pp. 100–109 (2012)
6. Ananthanarayanan, G., Agarwal, S.: Scarlett: coping with skewed content popularity in mapreduce clusters. In: Proceedings of the Sixth Conference on Computer Systems (2011)
7. Bairavasundaram, L.N., Arpaci-Dusseau, A.C., Arpaci-Dusseau, R.H., Goodson, G.R., Schroeder, B.: An analysis of data corruption in the storage stack. ACM Transactions on Storage (TOS) **4** (2008)
8. Kozyrakis, C., Kansal, A., Sankar, S., Vaid, K.: Server engineering insights for large-scale online services. IEEE Micro **30**, 8–19 (2010)
9. Chen, Y., Alspaugh, S., Katz, R.: Interactive analytical processing in big data systems: a cross-industry study of mapreduce workloads. In: Proceedings of the VLDB Endowment (2012)
10. Chen, Y., Srinivasan, K., Goodson, G.: Design implications for enterprise storage systems via multi-dimensional trace analysis
11. Delimitrou, C., Sankar, S., Vaid, K., Kozyrakis, C.: Decoupling datacenter studies from access to large-scale applications: a modeling approach for storage workloads. In: 2011 IEEE International Symposium on Workload Characterization (IISWC), pp. 51–60 (2011)
12. Ersoz, D., Yousif, M.S., Das, C.R.: Characterizing network traffic in a cluster-based, multi-tier data center. In: 27th International Conference on Distributed Computing Systems, ICDCS '07, p. 59 (2007)
13. Fan, B., Tantisiriroj, W., Xiao, L., Gibson, G.: Diskreduce: raid for data-intensive scalable computing. In: Proceedings of the 4th Annual Workshop on Petascale Data Storage (2009)
14. Iamnitchi, A., Doraimani, S., Garzoglio, G.: Workload characterization in a high-energy data grid and impact on resource management. In: 2009 IEEE International Conference on Cluster Computing (CLUSTER), pp. 100–109 (2009)
15. Kavalanekar, S., Worthington, B.: Characterization of storage workload traces from production windows servers. In: 2008 IEEE International Symposium on Workload Characterization (IISWC), pp. 119–128 (2008)
16. Kavulya, S., Tan, J., Gandhi, R., Narasimhan, P.: An analysis of traces from a production mapreduce cluster. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 94–103 (2010)
17. Kyrola, A., Blelloch, G., Guestrin, C.: Graphchi: large-scale graph computation on just a pc. In: Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation (2012)
18. Ren, Z., Xu, X., Wan, J., Shi, W., Zhou, M.: Workload characterization on a production hadoop cluster: a case study on taobao. In: 2012 IEEE International Symposium on Workload Characterization (IISWC), pp. 3–13 (2012)

19. Sankar, S., Vaid, K.: Storage characterization for unstructured data in online services applications. In: 2009 IEEE International Symposium on Workload Characterization (IISWC), pp. 148–157 (2009)
20. Wang, L., Zhan, J., Luo, C., et al.: Bigdatabench: a big data benchmark suite from internet services. In: 2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA), pp. 488–499 (2014)