The Implementation of a Hadoop Ecosystem Portal with Virtualization Deployment

Chao-Tung Yang[1(✉)], Chien-Heng Wu[2], Wen-Yi Chang[2], Whey-Fone Tsai[2], Yu-Wei Chan[3], Endah Kristiani[4], and Yuan-Ping Chiang[1]

[1] Department of Computer Science, Tunghai University, No.1727, Sec.4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan
ctyang@thu.edu.tw, benjrevive@gmail.com

[2] High Performance Computing and Applications National Center, High-Performance Computing National Applied Research Laboratories, Hsinchu 30076, Taiwan
garywu@narlabs.org.tw, 0303106@nchc.narl.org.tw, 9303115@narlabs.org.tw

[3] College of Computing and Informatics, Providence University, 200, Sec.7, Taiwan Boulevard, Shalu Dist, Taichung City 43301, Taiwan
ywchan@gm.pu.edu.tw

[4] Department of Computer Science, Department of Industrial Engineering and Enterprise Information, Tunghai University, No.1727, Sec.4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan
endahkristi@gmail.com

Abstract. The requirements of research, analysis, processing and storing of big data are more and more important because big data is increasingly vital for development in the fields of information technology, finance, medicine, etc. Most of the big data environments are built on Hadoop or Spark. However, the constructions of these kinds of big data platform are not easy for ordinary users because of the lacks of professional knowledge and familiarity with the system. To make it easier to use the big data platform for data processing and analysis, we implemented the web user interface combining the big data platform including Hadoop and Spark. Then, we packaged the whole big data platform into the virtual machine image file along with the web user interface so that users can construct the environment and do the job more quickly and efficiently. We provide the convenient web user interface, not only reduce the difficulty of building a big data platform and save time but also provide an excellent performance of the system. And we also made the comparison of performance between the web user interface and the command line using the HiBench benchmark suit.

Keywords: Big data platform · Portlet · Virtualization · Hadoop Spark

F. Xhafa et al. (Eds.): 3PGCIC 2018, LNDECT 24, pp. 116–127, 2019.
https://doi.org/10.1007/978-3-030-02607-3_11

1 Introduction

Big Data is becoming more and more important today. Due to the rapid development of computers, networks, and information services, a large amount of data has been generated [1,2]. Many industries see it as an important resource. The relevant researches, development, storage, applications, and environments are constantly expanding and updating because of the development of big data. It is a good time for people who are willing to get into the field of big data because there are more and more resources available.

Although there are many resources about big data and many applications available, ordinary users may have some problems using big data tools at present. The issues may be how to prepare the environment suitable for big data, how to set up the whole environment. Because of the need for big data research, we want to address these situations that are present. We want to do the studies and develop a way to simplify the pre-operation and installation process of the big data platform. The idea that can deploy the big data platform directly in the existing environment, does not require the dedicated devices, let users choose the environment they are familiar to, makes users operating the tool intuitively, reduces the chance of error and makes the different types of jobs executing together. Besides, we want to make the file management, job status monitoring and job scheduling easier and the capability for advanced users to add functions by themselves.

In this work, we implemented the web user interface applying to Hadoop ecosystem with Virtualization Development. The specific purposes of our work are:

- To package the web user interface along with the whole Hadoop ecosystem into a virtual machine image file.
- To develop a web user interface for this system in modular and allows users to modify or add the desired functions based on their needs by introducing Liferay Portal into our system.
- To measure the average time of VM Image on PC and Notebook.
- To compare the performance of sorting, word count, and terasort.
- To compare the performance between using the portal and the command line of Hadoop, and Spark in our system.

The rest of this work is organized as follows. In Sect. 2, we will describe the background information and previous studies related to our work, including big data, Hadoop ecosystem, the portal and the portlet, and the virtualization technology. In Sect. 3, the system architecture and the implementation are introduced. Section 4 shows the experiments done in our system, including the experimental environment and the experimental results. Finally, we summarized our work and the future work in Sect. 5.

2 Background Review and Related Works

In this chapter, the background information related to our work, including big data, Hadoop ecosystem, portlet, and the virtualization technology are introduced.

2.1 Big Data

Big data means the data sets which are hard to be processed with traditional methods or tools owing to the large volume or the high complexity. Big data can be the data collected from sensors, log files generated while servers are running, or the user behavior recodes and posted information on the Internet.

The term was defined as the combination of 3Vs: Volume, Velocity and Variety [3]. These are the generic big data properties. Nowadays, the big data system definition is extended to the 7Vs [4].

2.2 Hadoop Ecosystem

2.2.1 Hadoop

Apache Hadoop [5,6] is a open-source software framework for big data processing. Owing to its reliability, scalability and distributability, Hadoop can provide processing capacity by integrating the computational resource of the thousands nodes in the cluster. The implementation of Hadoop is based on two research results published by Google, Google Distributed System (DFS) in 2003 and MapReduce programming framework in 2004. The Hadoop framework is constructed on the Hadoop Distributed File System (HDFS) and it manage jobs and resource by YARN. The MapReduce [7] programming framework is implemented to operate the distributed processing by dividing the files into the same block size and distributing them to nodes [8].

2.2.2 HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system that provides data storage with reliability, scalability and fault tolerance [9]. It is designed to be deployed on low-cost hardware. HDFS is suitable for big data applications because it provides high-throughput and streaming data access and can store data of different kinds of format. HDFS has master-slave architecture including a single namenode and multiple datanodes. In the HDFS, data is divided into some blocks and distributed to nodes. If there are some datanodes down, HDFS can recovery the data with the backups on the other working datanodes [10].

2.2.3 Spark

Spark provides the application programming interface (API) centered on the data abstraction called the resilient distributed dataset (RDD) distributed to

the nodes in the cluster with fault tolerance for programmers. Spark is designed to improve the performance of MapReduce by offering the in-memory processing. The programs run with Spark can be up to 100 times faster than Hadoop MapReduce in memory or 10 times faster on disk. The features of RDD are conducive to the implementation of iterative algorithms, accessing datasets multiple time using loops, and analyzing data interactively. For machine learning systems, the iterative algorithms are the training methods. Thus Spark is the framework suitable for machine learning.

2.3 Portal and Portlet

A portal [11,12] is a specially designed website with specific contents and appearance. It gathers resource and information together from the system or other source and presents them to users on the single entry point with consistent way. For users, they can easily obtain resource and information and use the applications from one location.

A portal is consists of several portlets. A portlet is a pluggable user interface software component. It is based on Java, can be deployed, configured and displayed in the web container. A portlet can be regarded as a miniature web application. It is managed by portlet container, used to process the requirement from the container and generate contents dynamically.

2.4 Virtualization

Virtualization [13] is the technology that can abstract, manage, and redistribute the computing resources like CPU, memory, storage, network, applications, and so on. Virtualization can make the computing resources more flexible and scalable, and reduce the costs. In this work, we use the virtual machine (VM) an application of the virtualization technology to build our system in it. We can configure the computing resource of a virtual machine and package it into a image file easy to be move or copy to another environment. A virtual machine contain the whole operating system and applications in it. The resource of a virtual machine can be reconfigured according to the resource in the environment.

3 System Architecture

The architecture and the implementation of our system are introduced in this section. Our system is based on Hadoop ecosystem and packaged into virtual machine images along with the web user interface. The bottom layer of the software part of our system is the hypervisor of the virtual machine, Oracle Virtualbox and VMware Workstation are tested in this work. And the Ubuntu desktop operating system is installed on the hypervisor. Then the Hadoop ecosystem that includes HDFS, Yarn, ZooKeeper and Spark is built in Ubuntu. The Hadoop applications and the Liferay Portal are based on Hadoop and Liferay server. Users can execute big data jobs through the portal easily.

- job submission: executing Hadoop applications with given jar file and required arguments;
- file upload: uploading the given file to the destination path on Hadoop Distributed File System;
- sequential file packaging: packaging the given files like images into a sequential file and uploading it to the destination path in Hadoop Distributed File System;
- file management: presenting the files and directories on Hadoop Distributed File System.

4 Experimental Results

In this section, we show the system and experimental results. In Sect. 4.1, we introduce the experimental environment, including the hardware specification and the software information. Our experimental results are presented in Sect. 4.2 in detail including the virtual machine deployment in different virtualization environments, functionality validation of the portlets, performance comparison between using the portlets on the web user interface and command line, and performance comparison between Hadoop and Spark.

4.1 Experimental Environment

In this work, we perform the experiments on a desktop personal computer to simulate the user operating environment. The hardware specification is shown in Table 1. The operating system on the computer is Microsoft Windows 7 SP1, and the one in our virtual machine is Ubuntu Desktop 16.04 LTS. The virtualization platforms are Oracle Virtualbox and VMware Workstation. The software includes Hadoop ecosystem, Liferay Portal, and the benchmark suite HiBench. The detail software versions are shown in Table 4.

Table 1. Personal computer hardware specification

CPU	Intel core i7-2600 (3.4GHz)
Memory	DDR3 RAM 12 GB
Graphics	Intel HD graphics 2000
Hard disk	1TB SATA III hard disk

Table 2. Notebook hardware specification

CPU	Intel core i7-3632QM (2.2GHz)
Memory	DDR3 RAM 8 GB
Graphics	AMD radeon HD 7500M/7600M series
Hard disk	128GB SATA III SSD
External hard drive	500GB HDD (SATA to USB 3.0)

Table 3. Configurations of virtual machine

vCPU	4 cores
vRAM	8 GB
vHDD	48 GB

4.2 Experimental Results

4.2.1 Virtual Machine Deployment in Different Virtualization Environments

We deploy the virtual machines on Windows using Oracle Virtualbox and VMware Workstation, then record each the time they took. The process of Virtual Machine image file import on VMware Workstation and the one on Oracle Virtualbox.

Table 4. Software information

Software name	Version
Oracle virtualbox	5.1.18
VMware workstation	12.5.6
Microsoft windows	7 SP1
Ubuntu	16.04 LTS
Hadoop	2.6.0 (CDH 5.5.1)
Spark	1.6.0
ZooKeeper	3.4.5 (CDH 5.5.1)
HBase	1.0.0 (CDH 5.5.1)
Liferay IDE	2.2.4 GA5
Liferay portal	6.2.5 GA6
HiBench	6.0

The size of our virtual machine image file is about 7.73 GB. The each average time of virtual machine image import is shown in Tables 5 and 6. There are two part of this experiment. One is using the desktop personal computer and the other is using a laptop with an external hard drive. To simulate the scenario of making the virtual machine portable, we put the virtual machine image file in the external hard drive and connect it to the laptop. And the configurations of the two virtualization platform of this part are set to store the vHDD on the external hard drive. The time importing the virtual machine image took is less than 10 min. It is faster than installing the operating system and all the Hadoop software by users themselves. The process of image file import is much easier than building the whole system. And there is no need to keep concentrating on the process of import. That means the virtual machine can help simplifying the process of building the system and saving time.

Table 5. Average time of VM image import on PC

Hypervisor	Time of VM image import
Oracle virtualbox (5.1.18)	∼173 s
VMware workstation (12.5.6)	∼338 s

Table 6. Average time of VM image import on laptop

Hypervisor	Time of VM image import
Oracle virtualbox (5.1.18)	∼755 s
VMware workstation (12.5.6)	∼1363 s

4.2.2 Liferay Portal Webpage

We implemented portlets that can perform Hadoop and Spark operations including jar file execution, file uploading, file management and sequential file packaging. The portlets are modular so they can be add into or remove from the Liferay Portal web page by the user. The Liferay Portal web page we implemented is shown in Fig. 1.

4.2.3 Functionality Validation of Portlets

In this experiment, we use the portlets on the web user interface to execute a wordcount job to demonstrate our function of the portlets we developed. First, we upload our sample text file to the HDFS by filling in the source file full path and the upload destination field, clicking the upload button and waiting for the result. Then we can check the directory we uploaded a file to by the HDFS browser. Second, the full path of the wordcount jar file, input text file and output destination must to be set. The execution result is shown in the text

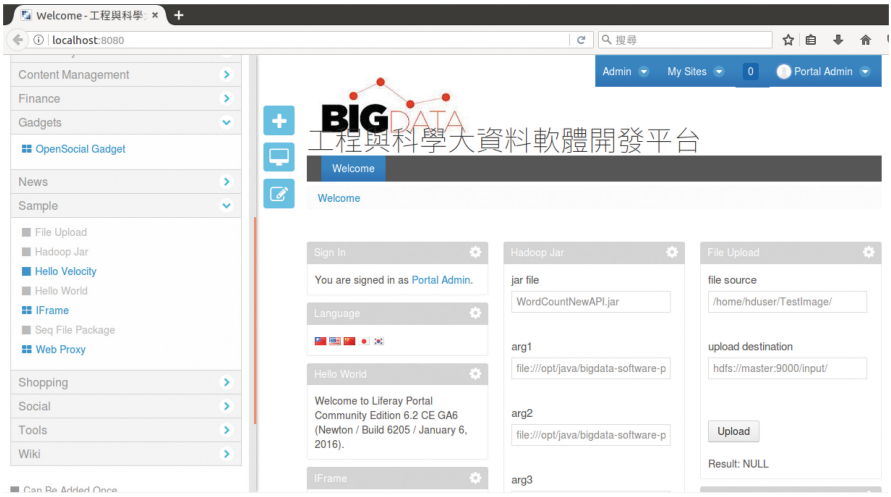

Fig. 1. Liferay Portlet web UI

field after clicking the submission button and waiting for execution finished. We also can check the output directory we set through the HDFS browser. Then we upload our sample text file to the HDFS, browse the directory on the HDFS, execute the wordcount jar application on Hadoop, and check the output result. Last, we can check the output file on the HDFS by downloading it through the HDFS browser or using the portlet of command execution.

4.2.4 Performance Comparison between the Portal and the Command Line

The following are experiments of performance comparison between using the command line and the portal web user interface. We use the Hadoop and Spark benchmark suite — HiBench to evaluate the performance of our platform. We run the three kinds of workloads, sorting, TeraSort and wordcount. In theory, the execution time of using command line should be better than or equivalent to the ones of using the portal interface. We run the benchmark with three kinds of dataset scale. The each experimental data is the average value of the five times result. The experimental data show that most of them meet the cognition. In this work, most of the differences of performance are related to the execution time. Figure 3 shows that it takes more time to sort data through the portal web user interface. We find that the differences of the sorting performance between using command line and the portal web user interface with Hadoop are higher than the one with Spark. We can get the result that it also takes more time to perform the wordcount application through the portal web user interface from Fig. 3. But the difference of the wordcount performance between using command line and the portal web user interface with Spark and the large scale dataset is quite higher than others. Figure 4 shows that it takes more time to perform the

TeraSort application through the portal web user interface but Hadoop with the small scale dataset.

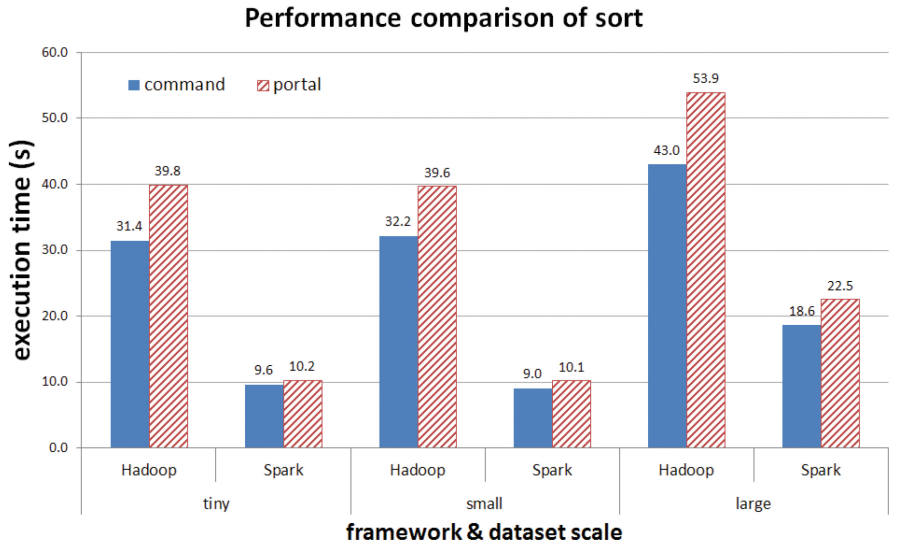

Fig. 2. Sorting performance comparison

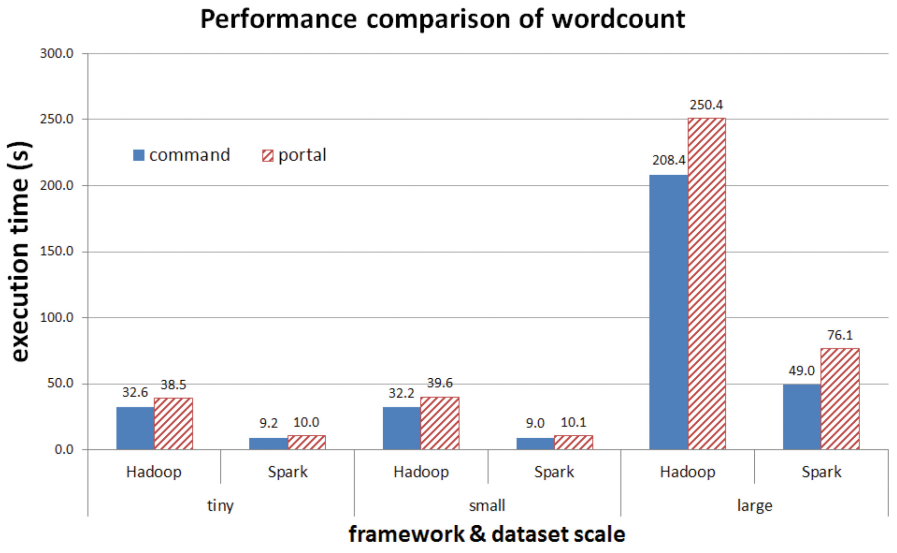

Fig. 3. Word count performance comparison

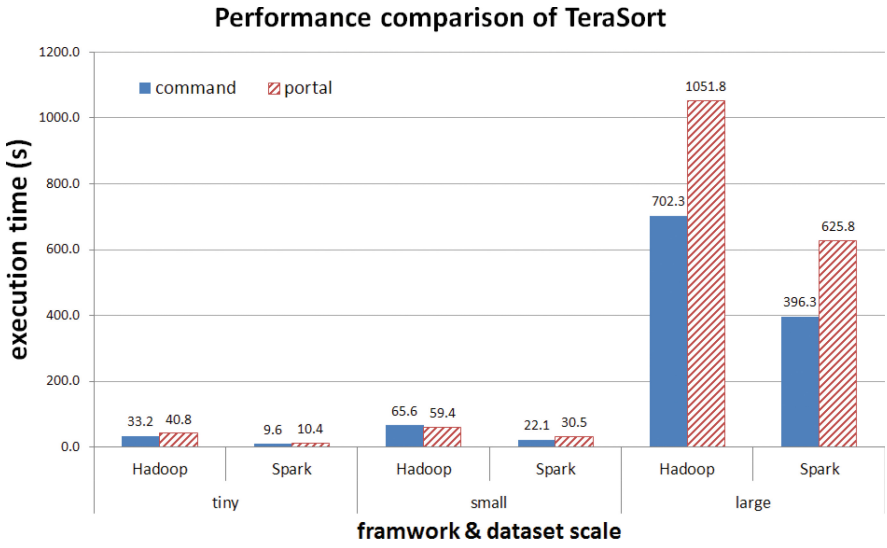

Fig. 4. Terasort performance comparison

4.2.5 Performance Comparison between Hadoop and Spark

In Figs. 5 and 6, we get that all the execution time of Spark are shorter than the one of Hadoop. While the dataset size is bigger, there are also more differences between the execution times. And we can see the differences of execution times between Hadoop and Spark. The speed of Spark is truly faster than Hadoop. In this work, the speed of Spark is at least 60 percent faster than Hadoop.

5 Conclusions

In this work, we developed and implemented a portal web user interface and portlets that facilitated the use of the Hadoop ecosystem and integrated the interface with the Hadoop ecosystem into a virtual machine image file. It provides a fast and convenient way to set up the platform for users. The processes of building the whole system were described, and the implementation of our system was presented. We have tested how quick the process of using the virtual machine image file to deploy our system on several desktop environments. We had executed the Hadoop job through the portlets we developed on the web user interface to validate the functionality. The differences in performance between using the portal web user interface and the command line to perform several works with Hadoop and Spark in our system are also tested. Executing the jobs with the large-scale dataset by using the portlets takes more time than using the command line. The differences in performance between Hadoop and Spark are also presented. The result of the performance comparisons is similar to those given in other studies before. In theory, the execution time of using command-line should be better than or equivalent to the execution time of using the portal web

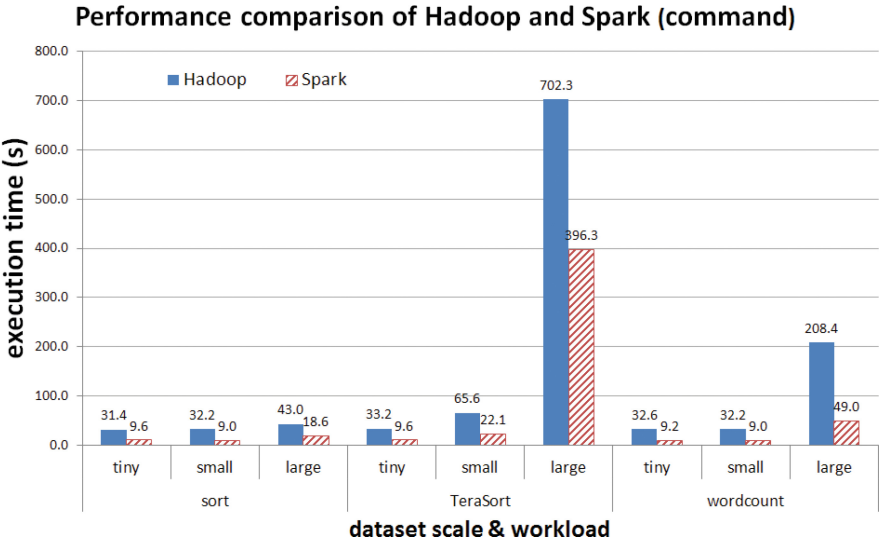

Fig. 5. Performance comparison between Hadoop and Spark (command)

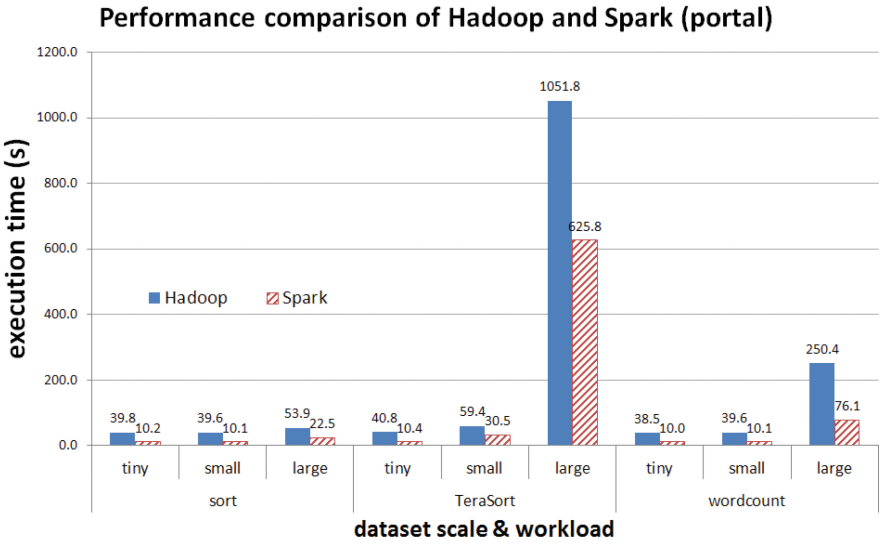

Fig. 6. Performance comparison between Hadoop and Spark (portal)

user interface. Our experimental data show the actual test comparison results. That also indicates that there is still a challenge for improvement.

Acknowledgements. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2221-E-029-008 and MOST 106-3114-E-029-003.

References

1. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
2. Yang, C.-T., Yan, Y.-Z., Liu, R.-H., Chen, S.-T.: Cloud city traffic state assessment system using a novel architecture of big data. In: 2015 International Conference on Cloud Computing and Big Data (CCBD) (2015)
3. Laney, D.: 3D data management: controlling data volume, velocity, and variety. Technical report, META Group (2001)
4. Gupta, A.: Big data analysis using computational intelligence and hadoop: a study. In: 2015 International Conference on Computing for Sustainable Global Development, INDIACom 2015, pp. 1397–1401 (2015)
5. Apache Hadoop (2014). http://hadoop.apache.org/
6. Hadoop (2017). http://en.wikipedia.org/wiki/Apache_Hadoop
7. Mapreduce (2017). https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
8. Dittrich, J., Quian, J.: Efficient big data processing in Hadoop MapReduce. Proc. VLDB Endow. **5**(12), 2014–2015 (2012)
9. Borthakur, D.: The Hadoop distributed file system: architecture and design (2007). http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf
10. Azzedin, F.: Towards a scalable HDFS architecture. In: Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013, pp. 155–161 (2013)
11. What is a Portlet - O'Reilly Media (2017). http://archive.oreilly.com/pub/a/java/archive/what-is-a-portlet.html
12. Portals and Portlets: The Basics (2017). http://editorial.mcpressonline.com/web/mcpdf.nsf/wdocs/5232/$file/5232_exp.pdf
13. Virtualization Technology & Virtual Machine Software - VMware (2017). https://www.vmware.com/il/solutions/virtualization.html