J Supercomput (2014) 68:488–507
DOI 10.1007/s11227-013-1050-4

Towards a framework for large-scale multimedia data storage and processing on Hadoop platform

Wei Kuang Lai · Yi-Uan Chen · Tin-Yu Wu · Mohammad S. Obaidat

Published online: 5 December 2013

Abstract Cloud computing techniques take the form of distributed computing by utilizing multiple computers to execute computing simultaneously on the service side. To process the increasing quantity of multimedia data, numerous large-scale multimedia data storage computing techniques in the cloud computing have been developed. Of all the techniques, Hadoop plays a key role in the cloud computing. Hadoop, a computing cluster formed by low-priced hardware, can conduct the parallel computing of petabytes of multimedia data. Hadoop features high-reliability, high-efficiency, and high-scalability. The numerous large-scale multimedia data computing techniques include not only the key core techniques, Hadoop and MapReduce, but also the data collection techniques, such as File Transfer Protocol and Flume. In addition, distributed system configuration allocation, automatic installation, and monitoring platform building and management techniques are all included. As a result, only with the integration of all the techniques, a reliable large-scale multimedia data platform can be

Prof. Mohammad S. Obaidat is a Fellow of IEEE and Fellow of SCS.

W. K. Lai · Y.-U. Chen
Department of Computer Science and Engineering, National Sun Yat-sen University,
Kaohsiung, Taiwan, ROC
e-mail: wklai@cse.nsysu.edu.tw

Y.-U. Chen
e-mail: angus@cht.com.tw

T.-Y. Wu (✉)
Department of Computer Science and Information Engineering, National Ilan University,
Ilan, Taiwan, ROC
e-mail: tyw@niu.edu.tw

M. S. Obaidat
Department of Computer Science and Software Engineering, Monmouth University,
W. Long Branch, Monmouth, NJ 07764, USA
e-mail: Obaidat@monmouth.edu

offered. In this paper, we introduce how cloud computing can make a breakthrough by proposing a multimedia social network dataset on Hadoop platform and implementing a prototype version. Detailed specifications and design issues are discussed as well. An important finding of this article is that we can save more time if we conduct the multimedia social networking analysis using Cloud Hadoop Platform rather than using a single computer. The advantages of cloud computing over the traditional data processing practices are fully demonstrated in this article. The applicable framework designs and the tools available for the large-scale data processing are also proposed. We show the experimental multimedia data including data sizes and processing time.

Keywords Cloud computing · Hadoop · MapReduce · BigTable · High performance computing

1 Introduction

Large-scale multimedia data storage and processing techniques—making use of multiple computers to process terabytes or even petabytes of data in a parallel computer efficiently—are playing an increasingly significant role in cloud computing nowadays. Among the current network services and telecommunication services, only a few of them practically apply cloud computing techniques; to put it simple, most current network and telecommunication services are provided on the basis of a single server. As a result, the quality of the service depends mainly on the class of the computer server. On the condition that only high-priced servers can provide high-class service, undoubtedly, it will be costly to build a server equipped with a high and complete capacity for the data storage. Besides, a robust server is required to speed up the processing of the large-scale multimedia data; however, the speed of the data processing can never match up to the growth of the data. Providing it takes 3 days to process the data accumulated in 1 day, the most updated and useful information cannot be analyzed and applied. What is worse, the ever-increasing data can never be processed in time. In view of this, cloud computing has been devised due to the thriving development of the Internet and the mobile devices. To share and retrieve data and get service more conveniently, the users transfer the data from the personal computers to the online storage services. To provide more reliable and high-speed services, the service providers urgently require a new service system. To serve the purpose, Google developed many cloud computing techniques and frameworks. For example, MapReduce provides integrated computing resources in distributed computing to reduce the computing time. Google File System (GFS) integrates a large amount of distributed storage space into a reliable storage medium. BigTable provides highly-efficient distributed database. Of the frameworks mentioned above holds a striking characteristic—using the new framework [1,2], the service developers just need to focus on the service development without considering how to put the data into the distributed database and how to appropriately allocate the computing resources. The newly-developed techniques and frameworks, in which the allocation and distribution of the data and computing would be processed by the cloud computing framework, considerably increase the speed of the service development [3].

Large-scale multimedia data storage operations originated from Google proposes three software techniques—one is the Google File System, abbreviated as GFS [4], second is the MapReduce [5] and the third is the BigTable [6,7], a distributed database. Later, the Apache Hadoop [8], an open-source code framework, implements Google's GFS and MapReduce algorithm. As a best-known and widely-used large-scale multimedia data storage operation technique, Hadoop is not only a distributed file system used for storage, but also a framework designed to execute distributed applications. In addition to the core techniques in storage operations, the file and data collection and other monitor-related techniques also play the key parts in building a large-scale multimedia data storage operation platform. File Transfer Protocol (FTP) is commonly used for file and data collection. Besides, Cloudera also provides Flume [9] to act as a tool for the uploading of the large-scale multimedia data collection. Monitor techniques include such parts as the hardware configuration allocation, automatic installation and monitoring. To efficiently achieve the advantage of the large-scale multimedia data platform, the complete large-scale multimedia data storage operation techniques should include not only the distributed processing, but also the transmission of management techniques [10].

2 Core techniques

This section will introduce the core techniques in large-scale multimedia data storage operation. Hadoop, the most mature and widely-used framework, combined with MapReduce, a distributed computing technique, and the distributed database, HBase, will be applied to achieve the greatest efficiency.

2.1 Hadoop

Processing of large-scale multimedia data has been a very important topic in computer science and practical application. To tackle the problem of the storage and the processing of large-scale multimedia data, relative data base techniques are being eliminated by and by. Other techniques such as NoSQL and Not Only SQL are currently taking the role of relative data base techniques to enhance the data processing efficiency and its flexibility. During the transitional period, Hadoop has gradually become a crucial role in large-scale multimedia data processing.

As an open-source distributed computing platform, Hadoop combined with multiple-computer resources forms a distributed file system. Hadoop, a distributed file system, was developed by Apache Software Foundation. Hadoop basically is a platform built by a computer cluster composed of general PCs. The main advantage of Hadoop is its capability to handle and store the increasing large-scale multimedia data. With many nodes parallelizing data processing, Hadoop can gain the responses at a higher speed. In dealing with the error occurring to a node, Hadoop would obtain the backup data in no time and deploy the computing resources [11,12].

2.2 HDFS

The core architecture is the Hadoop Distributed File System, abbreviated as HDFS, and the MapReduce (Fig. 1). HDFS is a distributed file system, which is used for data

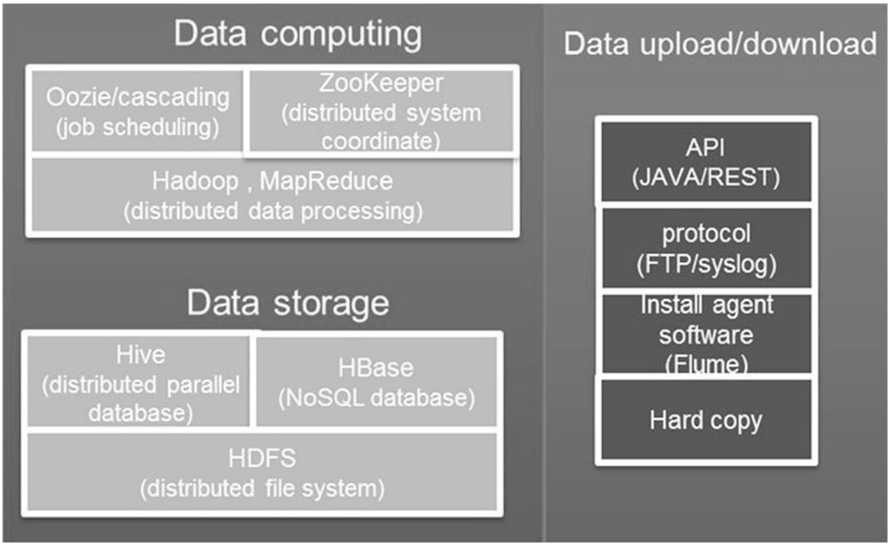

Fig. 1 Large-scale data cloud computing platform—storage and computing

storage and has high fault-tolerance. Even when parts of the Data Nodes break down, HDFS can still work ordinarily. HDFS is a master/slave architecture. In the general deployment, only one NameNode is executed on each master and one DataNode is executed on each slave.

Responsible for the file system dispatch, the NameNode divides the files into numerous blocks, which are pre-set to be 64 or 128 MB. After dividing the files into blocks, the NameNode backs up three replicas of blocks and allocates them to different DataNodes. As shown in Fig. 2, the first file is divided into numerous blocks: $\{1, 2, 5\}$, and the blocks are allocated to different DataNodes, which are monitored and managed by the NameNode. When the file is read and written, the client inquires the NameNode about the file location and then retrieves the actual file blocks from the DataNode. If any one of the DataNodes is damaged or off-line, the NameNode backs up the file blocks from other DataNodes to maintain the three-replica-block fault-tolerance mechanism. Further, HDFS architecture is also characterized by the feature that it can be deployed on the low-priced hardware devices without particularly requiring high-priced and stable computers. Moreover, Hadoop can dynamically increase new DataNodes to expand its capacity and computing ability. In addition to the fault tolerant mechanism, the NameNode can make use of the Secondary NameNode to back up data regularly to achieve the protective mechanism [2].

2.3 MapReduce

MapReduce is a parallel computing algorithm and its processing techniques can be divided into two stages: Map and Reduce. In the Map process, the large distributed stored data would be computed in a distributed manner. In the Reduce process, the

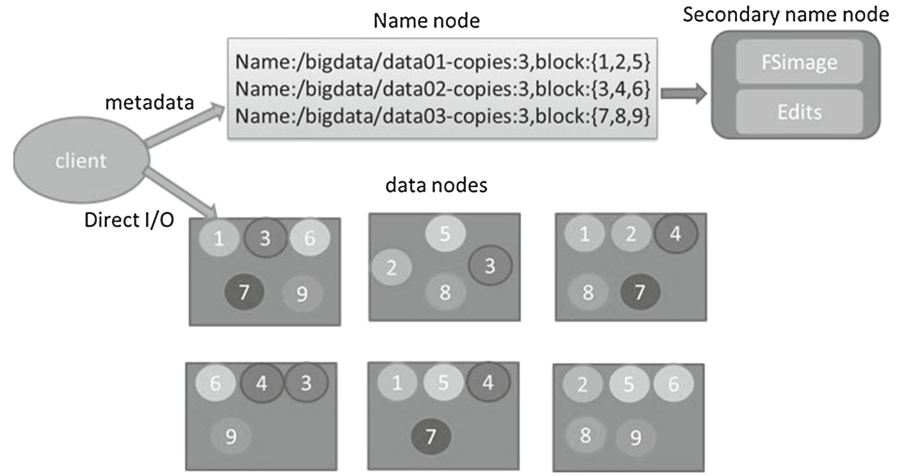

Fig. 2 HDFS architecture

analyzed data would be archived. The execution procedure is shown as Fig. 3. Through Hadoop, (1) The programs fork a NameNode (also Master) and several DataNodes (also Workers), (2) The Master assigns the task that a Worker has to take; Workers execute the Map process or the Reduce process. When a Worker completes the assigned task, it moves on to the newly-assigned task. (3) The Worker that is assigned to execute the Map process retrieves the data waiting to be processed from the files, which have been divided into numerous blocks. (4) The Worker then writes the processed metafiles, which wait to be read in the Reduce process, into the hardware of the host computer. (5) After the metafiles are retrieved, the Reduce process would be executed. (6) The result is written into the output files. To summarize, MapReduce applies HDFS to save the data-transfer time and further achieve higher processing efficiency using local computers to process the local data. All the tasks are executed by the DataNodes. Since each DataNode is equipped with part of the file blocks, it can start the execution task without searching the files first, which in turn speeds up the overall processing performance [13,14].

2.4 HBase

As a distributed database structured on HDFS [16], HBase provides high availability and high-efficiency. In addition, HBase can be expanded in capacity and efficiency easily. Different from the general Relational Database, HBase employed Row and Column to construct an index value, used to save and retrieve data (Table 1). Hence, querying in the HBase is much like using Map Container. Another characteristic of HBase is that each data has a Timestamp. Thus, multiple data can be stored on one column according to different time periods when the data is uploaded. Moreover, Hive, a distributed database developed from HBase by Facebook, provides a mechanism similar to SQL to save and retrieve the large-scale data stored on the Hadoop platform (Table 2).

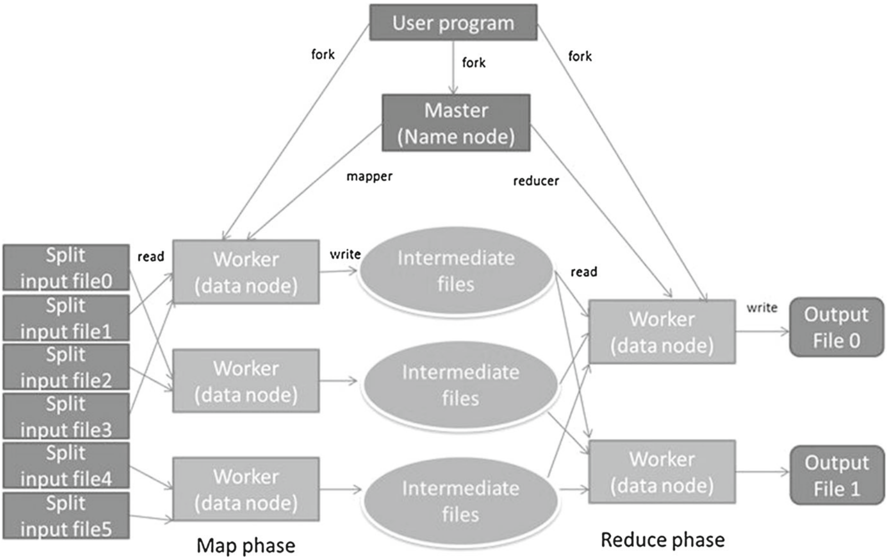

Fig. 3 MapReduce procedure map

Table 1 HBase data model logical table

Row key	Timestamp	Content	Classify		
			News	Stock	Sports
CNN news	tl	"gun event"	"society"		
	t3	"Announced the election"	"political"		
CNN stock	t2	"High revenue"		"cloud computing"	
ESPN sports	t4	"win 20 games"			"baseball"

Table 2 HBase data model actual table

Row key	Column families : column qualifier	Timestamp	Value
CNN news	Content :	tl	"gun event"
CNN news	Classify : news	tl	"society"
CNN stock	Content :	t2	"High revenue"
CNN stock	Classify : stock	t2	"cloud computing"

3 File collection and uploading techniques

3.1 FTP

FTP protocol has become the most widely-used approach to collect data. To analyze files on Hadoop, we can use HDFS over FTP, which is a FTP service developed to

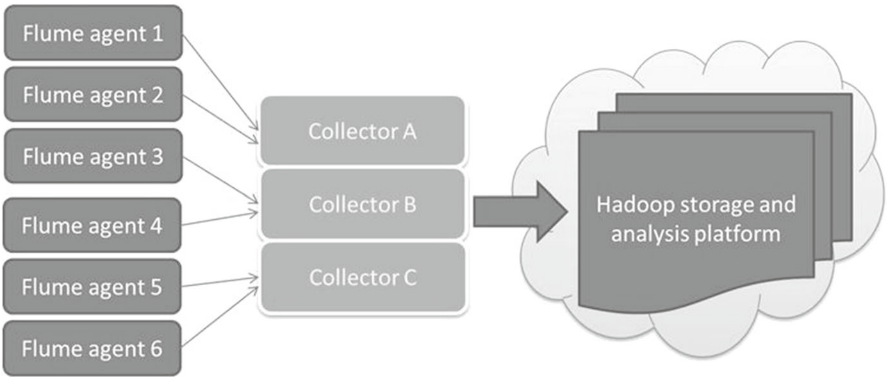

Fig. 4 Flume deployment architecture map

be added to Hadoop. Users can directly make use of FTP clients to get connected to HDFS and access data on FTP server. Plus, the large-scale multimedia data from the computer, which generates the information, such as the proxy server and mail server, can be set up to get uploaded to the Hadoop platform via FTP. Uploading information via FTP has the advantage that it can resume uploading even when the network is disconnected [15].

3.2 Flume

Flume, mainly used to collect the log files generated from the system server, is an efficient and reliable file collection software. Flume can collect the log files generated from each server; it integrates the files and then transfers the files to the storage space such as the Hadoop HDFS. The architecture of Flume is quite simple and scalable. The main concept of Flume is the Data Flow [9]. Figure 4 demonstrates Flume's procedures of collecting a series of log files from servers.

The deployment of Flume is to compose a three-level architecture via multiple logical nodes and Flume Master. The first level is the agent level; the agents are usually installed on the computer which generate the log files, and serve as the source of log files in the Flume architecture. Next, the agents transfer the data to the second level—the collector level. The collectors integrate the separate data flows and then transfer the data to the third level—the storage level.

4 Large-scale data platform structuring techniques

To construct a large-scale data platform, we require different managing and monitoring techniques. In this section, we take Hadoop as an example to demonstrate the constructing techniques, including hardware configuration allocation, automatic installation, and monitoring (Fig. 5).

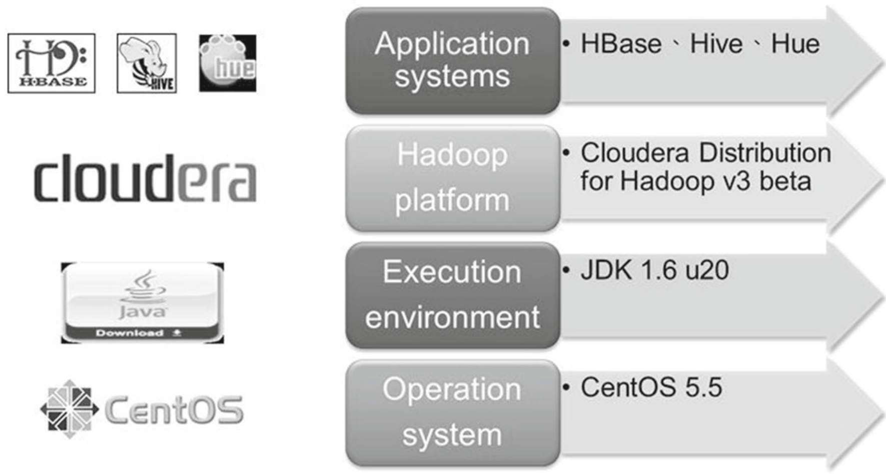

Fig. 5 Hadoop experiment platform installing software

4.1 The platform equipment

The cloud computing platform structured in the study is a general type. There are one hundred computers included and their specification is listed as follows:

CPU: The computing frequency is Intel Xeon 2.26 GHz. A single CPU contains four cores and each computer is equipped with 2 CPUs. Therefore, there are eight cores in total.
Memory: 4 GB ECC RAM. Each computer is equipped with 4 slims and the memory space is 16 GB.
Hardware: The data storage hardware with the capacity of 1 TB. Each computer is equipped with 4 storage hardware.

Totally, the capacity is 4 TB. Altogether, the total number of CPUs is $100 \times 4 = 400$ cores; the total storage hardware capacity is 100×4 TB $= 400$ TB. The fault-tolerance mechanism of Hadoop backs up the files into 3 replicas, so that the actual hardware capacity is 200 TB, and the total memory space is 16 G $\times$ 100 $=$ 1,600 GB. As the result of the power consumption test shows, the power consumption, while the computer is idle, is about 130–150 W and the power consumption in executing operations is about 175 W. When 100 computers are formed into a computing cluster, the total power consumption while idle is 225–250 kW, and the total power consumption in operation is about 280 kW.

4.2 The hardware configuration

The problems mainly encountered, while building the Hadoop cluster and achieving higher computer efficiency, are the space allocation, the hardware configuration specification, the network planning, and the power configuration. Of all the factors mentioned above, the hardware configuration and the network planning have a significant influence on the computing efficiency. The Hadoop is designed to be capable of operating on the low-priced computers without especially requiring high-priced and stable computers. The main reason is that Hadoop contains a fault-tolerance mechanism. Therefore, even when any of the bottom computers breaks down, the top-level task execution will not be influenced. Based on Hadoop's low requirements on the computer specification, it is only necessary to purchase the mainstream, reasonable and low-priced computers on the market. For example, an economical and top-scale Hadoop parallel computing cluster can be built with the current economic model of computers equipped with two 4-core CPUs and 16–32 G RAM. Because Hadoop makes use of the local CPU processors of each computer to process the data at the local hardware without transmitting the data to other processors, each computer is suggested to be equipped with multiple hardware to fulfill CPU's high-speed processing ability and further to achieve the high computing efficiency of each computer. Each computer is also suggested to be equipped with 4–12 hardware. The specific computer configuration is closely related to the tasks of the top-level application programs. The factors that the task is I/O bound or CPU bound programs, or that the computing bottleneck of the algorithm lies in the data retrieval or CPU computing ability, may influence the number of the hardware each computer should be equipped with. The computers in the cluster must rely on a switch to achieve communication with each other. Currently, the low-priced GE (Gigabyte Ethernet) interface is the first choice. However, how to connect the switch series and arrange enough bandwidth is another issue. The bandwidth is also closely related to the attribute of the top-level application program. For sorting tasks, a large quantity of package exchange among computers is needed; therefore, the network bandwidth is very likely to become the bottleneck of the computing efficiency in this condition. For example, if each computer can process 100 MB/s data, there are 60 computers and every 30 computers are put in a single switch; then how much network bandwidth between the two switches should be reserved? On the condition that every computer can process 100 MB/s data on average, that means there is 3 GB data on each switch. With half of the data set on the local switch and the other half set on the second switch, the average amount of the data transmission between the two switches is 1.5 GB/s, the equivalence of 12 GB/s. However, the amount is the highest data-transmission rate in theory; the data-transmission amount can be significantly decreased by means such as good program design or some filter designs at the map stage. Therefore, in the cluster of the general scale, the bandwidth of 4–6 GB/s would be adopted. There are various ways that can be used to meet the bandwidth requirements between switches. For example, we can adopt the switch with 10E (Ethernet) interface, which is simple, but costs high. Or we can adopt the switch-stacked technique to build an economical, but high-transmission-efficiency framework. Currently, there have been many models of switches with switch-stacked techniques on the market.

The study mainly focuses on the performance of the computing efficiency analysis; therefore, the speed of CPU and the number of the CPU cores must be increased. Besides allocating the tasks to each computer, we can also make good use of the multiple-core parallel computing to enhance the computing efficiency. In addition to enhancing the computing speed of CPU, the main processing efficiency bottleneck is the hardware IO. Therefore, it is suggested to use the higher-speed SSD (Solid-state Drive) hardware to reduce the influence of the hardware IO on the overall performance. The application developed on the platform requires shorter response time; it belongs to the short-term batch work required to be processed rapidly. For example, the timely processing of the database, HBase, is an application in the Hadoop platform, which requires high-efficiency computing performance. Also, the requirements should contain enough memory space. It is suggested that the ratio of (hardware number): (CPU core number) : (Memory GB) should be 1:2:4. If we use the ratio 1:2:4 to allocate the memory space, the bottleneck of the hardware IO would be reduced to minimum. If we also use the multiple-core to conduct the parallel computing task, we would meet the application requirements of high-speed computing.

4.3 Automatic installation

In the general process of installing the operation systems, users are required to fill in the installing content, the installing kits and the system set-ups in order. However, the practice often causes the bottleneck in the operation process even though the Hadoop DataNode is deployed fast and on a large scale. In view of this, if the installing content, the installing kits and the system set-ups in the installing process can be written into the configuration and stored at the designated areas in advance for the installation program of the operation system to read, then according to the configuration, the installation program would complete the installation of Hadoop DataNode automatically (Fig. 6). By using the automatic installation process, there will be many advantages listed as follow:

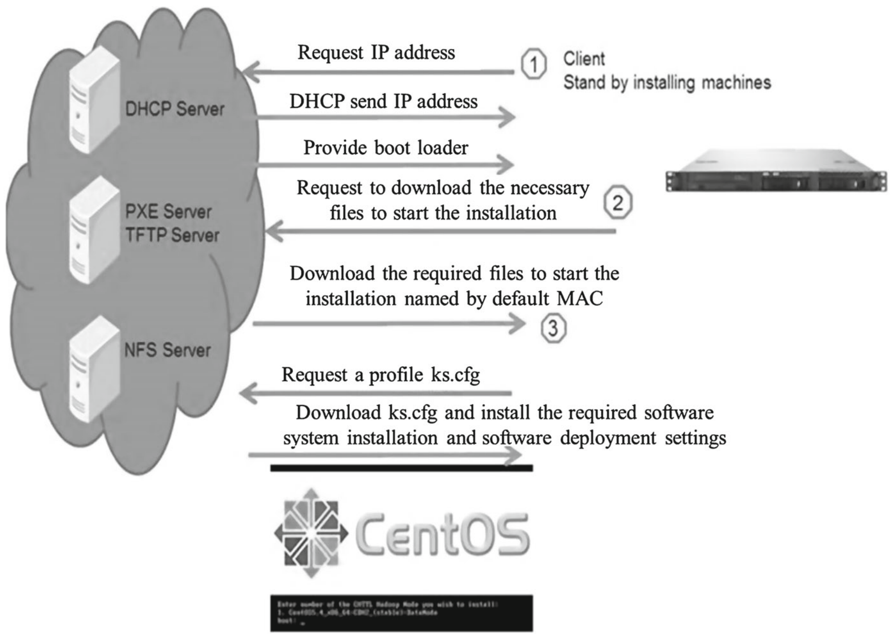

Fig. 6 Automatic installation architecture

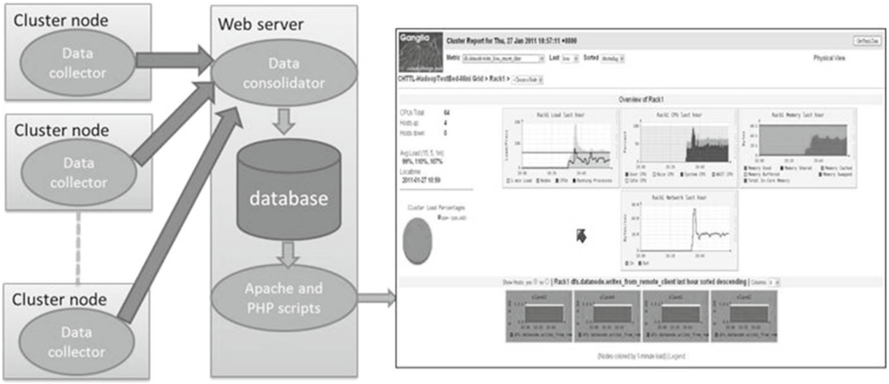

Fig. 7 Ganglia: system monitoring

1. Avoid the human error in the installation process.
2. Be able to deploy computers automatically on a large scale.
3. Shorten the installation time.

The most common automatic deployment tools of the operation systems include the Kickstart of PXE RedHat Linux and Ubnutu Linux, the Auto YaST of Suse Linux and RIS, sysprep, and answer file of Windows.

4.4 Platform monitoring

With the development of the large-scale multimedia data computing platform, getting a deeper understanding of the characteristics of the distributed computing platform is necessary. To monitor the performance of the hardware equipment while executing computing, to realize whether the service works normally, and to know and predict in advance the potential use and planning of the platform, we need a distributed monitoring system to monitor the large-scale multimedia data computing platform. Likewise, we take Hadoop for example. Ganglia [17], originally designed to monitor the distributed computing nodes, is an open-source code project developed by UC Berkeley (BSD-licensed). As shown in Fig. 7, the webpage is a monitoring webpage of the computing platform. It can dynamically observe the real-time CPU usage, RAM usage, hardware usage, and the network transmission conditions of every machine through the webpage, which enables the managers to monitor the operation condition of Hadoop platform conveniently and can also be helpful in debugging programs and tuning performance (Fig. 7).

5 The multimedia social network dataset using Hadoop cloud computing framework

The traditional multimedia social network analysis is greatly limited due to the computing ability of the traditional web server and the storage space. By applying the cloud computing framework, we can shorten the processing and analyzing time, utilize multi-dimensions and analyze non-structured raw data. For example, the traditional webserver is capable enough to process hundreds of Tera bytes of data in the dimension of up to billions of users' data on the multimedia social network. The study makes use of Hadoop cloud computing implementation and online analysis framework to break out the traditional limit and to investigate its feasibility, advantages and disadvantages.

5.1 The implementation environment

The initial target of the implementation is to find a model in a short time to verify the techniques. The main focus is firstly put on data uploading, preliminary summary and multiple-level summary. The next focus is set on building Meta Data, Dimension Form, and simplifying the details of the query nodes. Later, the focus would be transferred to data compression, updating maintenance, and dynamic query. The multimedia data source of the study is from "Facebook social network user analysis." The time range of the multimedia data is the three-year history data of Facebook. The general dimensions include month, age, gender, number of friends, application service, message service, and so on. The large dimensions are based on the condition that each user is a unit and builds a dimension on the scale of 10,000. The long-period history data, combined with the large dimensions, can present the usage analysis trend of the multimedia social network users, and the analysis trend of personal messages, which highlight the advantages of cloud computing.

5.2 The cloud computing online analysis and processing framework

As shown in Fig. 8, the study adopts the Hadoop cloud computing framework. After the raw data is uploaded to the Hadoop file system, HDFS, the format of Hive is applied to increase the speed of query. The traditional online analysis stored in the structure of multidimensional online analytical processing MOLAP [18–20] aims at higher-speed data storage and retrieval. However, the cloud computing framework cannot control the file displacement to achieve high-speed storage and retrieval. Thus, MOLAP loses

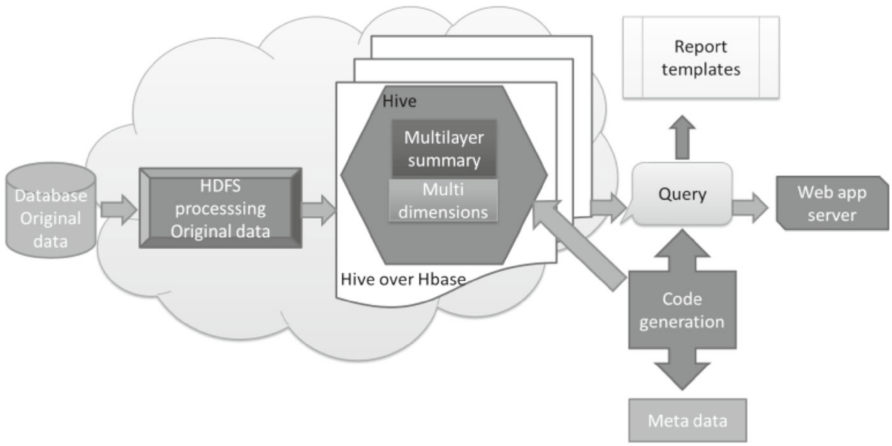

Fig. 8 Cloud computing online analysis and processing framework

its advantages in the cloud computing framework. Relatively, the “dimension-value” framework of relational online analytical processing ROLAP coincides with the “key value” of MapReduce. ROLAP [21], in coordination with the indexing of HBase, can fulfill the highest computing and querying efficiency. Therefore, the study adopts the ROLAP framework, storing the data blocks in HBase. If we do not adopt HBase and store the data blocks in HDFS, it will take more than 20 s to simply query and retrieve the archived data, which cannot achieve the reasonable response time [22].

Besides, the study adopts Hive, the query semantic of which is similar to that of SQL; that is, the summarized computing can be completed with one single instruction.

Otherwise, if we do not take Hive but have to write the MapReduce program to control the data storage and retrieval of Hbase, and the summarized computing, it would be more complicated and may produce more errors.

Below are the procedures of building blocks. First, we make a basic summary of the raw data. Next, we make a further multiple-level summary to increase the query speed by computing any possible query about permutations and combinations, and we cross analyze data in advance and store the computing results. Last, we make use of the query program to present the query results in the form of report template. Whether in the basic data summary, the multiple-level data summary, or the query report template, the Meta Data can be built in advance by means of the automatic program-producer, Code-gen, so as to produce numerous, complicated, but regular summary semantics or query programs [23].

5.3 The experiment results

The study uploads the basic data of the mobile users for 36 months; the data is in total 4.8 GB and the number of data is about 298,000 registered users. On average, the data are about 134 MB and the number of data is about 10,000 registered users for each month. The time consumed can be divided into three stages. At the first stage, the data are input to Hadoop. At the second stage, the data are then input into Hive to process. At the last stage, the multiple-level summarized computing will be executed. At the first stage of uploading the data to Hadoop, there are three steps as follows:

1. Output the data to the text files from the database.
2. Turn the text files into the Hive format.
3. Transmit the text files from the output computers to the Hadoop host computers through the Network.

It takes about one hour to complete the first step; on average, each file takes about 1.29 min. The time consumed in inputting the files into Hive can be divided into two parts. The first part is the time consumed in creating forms; the other part is the time consumed in inputting the files into Hive. Creating forms takes about 15 s while inputting the files into Hive takes about half an hour. On average, each file takes about 1 minute. At the third stage, it takes about 8.5 h to complete the multiple-level summarized computing with 7 dimensions and 128 permutations and combinations. Further studies are recommended to compare the time differences between by using the current computing mechanism and by using the Hadoop techniques within the same data range. Besides, whether there are any response time differences of query

Table 3 Cloud computing platform configuration and time cost

Configurations	Times cost
io.sort.factor = 100	*TeraGen* = 18 min 32 s
io.sore.mb = 340	*TeraSort* = 9 min 45 s
io.filter.buffer.size = 132,012	*TeraValidate* = 1 min 20 s
speculateive.execution = true	
mapred.child.java.opts = 2,048	*TeraSort* = *Map* + *Shuffle* + *Reduce*
map.tasks.maximum = 4	*Map* = 27 s
reduce.tasks.maximum = 2	*Average shuffle* = 6 min 24 s
HADOOP_HEAPSIZE = 2,000	*AverageReduce* = 2 min 41 s
Chunk_size = 64,128 MB	
tasktracker.http.threads = 40	
mapred.reduce.parallel.copies = 20	
mapred.compress.map.output = true	
Reducer = 285	

and summarized computing between by adopting HBase or not, and whether there are performance differences between by using the Hive instructions and by adopting the self-written MapReduce programs are the issues worth further research.

5.4 The performance analysis

The study makes use of 100 computers to form a computing cluster to get the performance data. TeraSort is generally used to test the platform computing capacity; therefore, the study also adopts the test Benchmark. The results are shown in Table 3

After the Hadoop parameter adjustment and test, the shortest execution time we get is 9 min and 45 s. Hadoop parameters would make a difference to the execution results, so that different application programs should be fine-tuned in parameters to achieve the best execution results. After the measurement and computing, the results of the actual energy consumption of building Hadoop platforms can be presented as follows: Multimedia Social Network Analysis (storage and computing): 200 TB of data are stored for 5 years, 1 TB computing is executed once a week, each execution lasts for 8 h. In Fig. 9, we can find that it can save more time to conduct the multimedia social networking analysis using Cloud Hadoop Platform than using a single computer. After measuring and calculating the energy consumption of the Hadoop platform that we build, we can get the following result:

The Self-built Hadoop Cluster:
150 nodes (total − 1,200 core), 200 T storage space
The equipment cost: about 8 million and 21,000 US dollars
Electricity/Five years: 2.2 million US dollars (computers would be shut off while not used)
0.3 kW (energy consumption) $\times$ 150 node $\times$ 8 h $\times$ 3(dollar/kW) $\times$ (12 $\times$ 5)(week)= \$2,160 US dollars
=> The total cost is about two million and 900,000 US dollars.

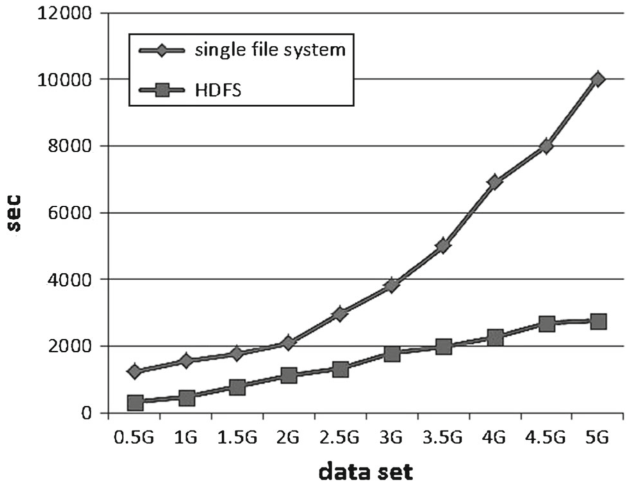

Fig. 9 Single FS VS HDFS performance comparison

Judging from the rough cost estimate of the cloud computing platform with the above-mentioned specifications, in addition to the hardware specification, the energy consumption should also be taken into consideration while building a cloud computing platform. As the estimate indicates, the cost of energy consumption of the cloud computing platform may go beyond that of building the hardware specifications of the platform. Therefore, it is suggested that suitable platforms should be built based on its specific application purpose to increase the using rate of the hardware and further to reduce the energy consumption cost after building the platform, because the sustainable development of the cloud computing platforms is achievable only on the stable and economical energy consumption.

We also use a single computer as well as multiple computers to compare and verify the performance efficiency of the Hadoop platform. In addition to the number of computers and the size of the processed multimedia, we conduct some experiments to investigate the influence of the computer grades and the number of redundant files on the performance efficiency of the Hadoop platform. Figure 10 can clearly show the performance efficiency of the Hadoop platform and the influence of other factors on the Hadoop platform. Figure 10 presents the comparison of the performance efficiency of the WordCount program using one single computer and four computers. Longitudinal axis represents the number of seconds, while the horizontal axis represents the size of the tested file. As the figure indicates, while processing small files, the single computer program has an advantage since it does not need extra network communication mechanism. However, while the processed file is larger than 80 MB (the highlighted circle), the Hadoop cluster made up of four computers is far superior to the single computer program in performance efficiency. As indicated in this experiment result, Hadoop platform is well suited to process big files.

Figure 11 shows the analysis of performance using the Hadoop platform with one to four computers, respectively, and using the single computer program. Due to the different levels of computer hardware specification, the hardware specification levels of

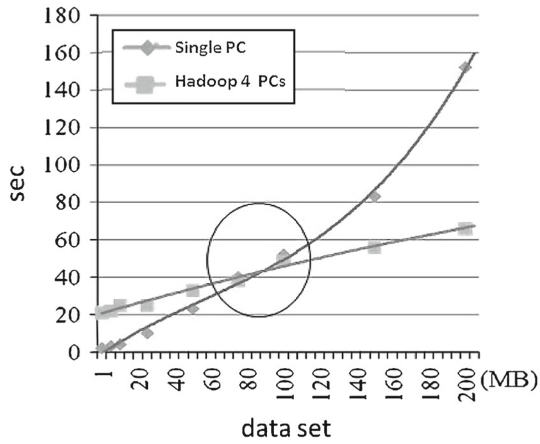

Fig. 10 Single PC VS Multiple PCs with Hadoop

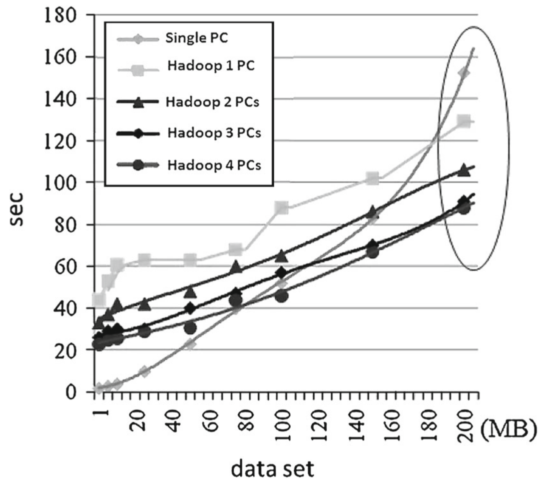

Fig. 11 The performance analysis of Single PC VS multiple PCs with Hadoop

the third and the fourth computer are far inferior to the Master and the second computer, and therefore there is better performance overall using three and four computers, but the effect is not obvious. However, as the highlighted circle in Fig. 11 shows, while large files are processed, the Hadoop cluster made up of multiple computers fulfills its full utility with the co-operation of its multiple computers, which also fully explains the main function of co-operative operation of Cloud Computing in Hadoop platform.

Figure 12 shows the result of the performance of the Hadoop platform with different levels of computer hardware specifications. As shown in Fig. 12, the level of computer hardware specification has an influence on the overall performance of the Hadoop platform. With the computer with better level of hardware specification added into the Hadoop cluster (the performance of Slave 1 is better than Slave 2 and Slave 3), the processing speed would be increased. However, if the computer with poorer hardware specification is added to the Hadoop cluster, the original performance might

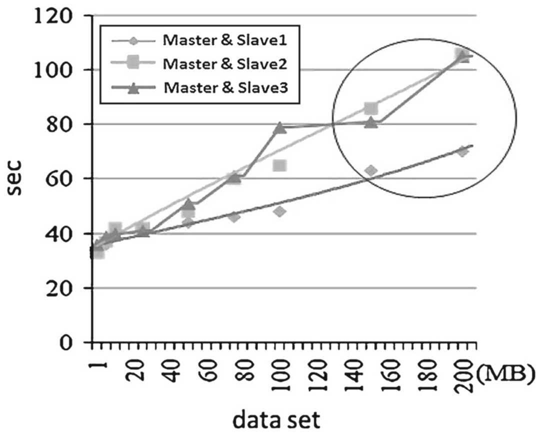

Fig. 12 Performance of the Hadoop platform with different levels of computer hardware specifications

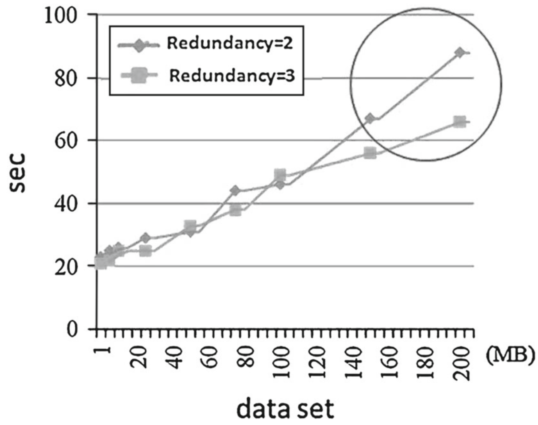

Fig. 13 The analysis of the influence of the number of redundant files on the Hadoop platform on the processing performance

be crippled. As the highlighted circle in Fig. 12 indicates, when a better computer is added, the performance would improve significantly.

In Fig. 13, the experiment aims to examine the influence of the number of redundant files on the Hadoop platform on the processing performance. Figure 13 shows the influence of Redundancy parameter on the performance of the Hadoop platform. The larger the number of redundant files is, the faster the performance of Hadoop platform is. However, more redundant files would consume more disk space. Google's default setting is making three copies. As the highlighted circle in Fig. 13 shows, while large files are processed, the processing speed with three copies is better than that with two copies.

In addition to the above performance experiments dealing with single file, we conduct experiments on big files and multiple files. The following Table 4 shows the experiment results while there are 100 processing files and each file is the 1.03 GB image file. According to the experiment results, while dealing with big files and multiple files, the Hadoop platform fulfills the function of cloud computing and thus saves processing time significantly.

Table 4 The comparison of the processing time with different PC numbers

PC numbers	Processing time (s)
1	120,897
2	45,768
3	39,686
4	35,103

Table 5 The comparison of processing time using Hadoop big data processing and traditional processing

Hadoop big data processing		Traditional processing	
Parallel processing with 100-node cluster		Sequential processing with 2 Severs (AP server+DB server)	
CPUs cores total: 400		CPUs total: 8	
Memory total: 2 TB		Memory total: 32 GB	
Disk total: 400 TB		Disk total: 1 TB	
Processing job	Processing time	Processing job	Processing time
FTP	1 h 40 min	FTP	2 h
Decompress	2 h	Decompress	2 h
Data loading into Hadoop	1 h 10 min	Data analysis	24 h
Hadoop data filter and statistical analysis	15 min	Analysis data loading into data base	46 h
Data result loading into data base	35 min	Statistical analysis of data base	3 h
Total processing time	5 h 40 min	Total processing time	55 h

According to Table 5, the comparison of processing time using Hadoop big data processing and traditional processing, the application of big data processing techniques significantly enhances the performance of big multimedia data analysis. But there is a similarity lying in the two methods; that is, the decompression of data takes two hours or so. To retrieve useful field information from each original file, traditional processing methods must inevitably decompress files and analyze each file one by one, while the big data processing method loads the decompressed data into Hadoop and then conducts the data filtering. In the process of researching the Hadoop data loading mechanism, it is found that Hadoop supports direct loading of Gzip or Bzip2 compression formats, but either of them is not the current compression format of the multimedia data. Therefore, the data decompression still needs to be executed. If future adjustments towards the direct compression can be made, the decompression process can be skipped and it would overall save up to 35 % of processing time. In addition, the restoration of the statistical data analyzed by Hadoop back into the database is currently conducted through the JDBC Driver of the Hive and Oracle; that is, the data is read through Hive and then written to Oracle. According to Table 5, to restore the statistical data from Hive to the database takes about 21 min. If Sqoop [24] can be utilized to restore the statistical data, the processing performance would definitely be enhanced. If Hadoop is used as the operation platform, the setting of Block Size would affect the efficiency of performance. The best settings for different computing capabilities (queries) vary. The larger the computing capabilities are, the larger the setting of Block Size must be. In this case, different settings can reach 60 % of performance gaps. Based on the current experience, on the big data processing platform, the setting with the number of Mapper ranging between 500 and 3,000 is better [25].

6 Conclusions and future work

With the development of network transmission and the enhancement of the computer computing capacity, the development of cloud computing is bound to bring us more potentials and possibilities in the network service, and provide stronger ability to store and compute data. Based on Hadoop, this study aims to introduce the large-scale multimedia data storage and computing techniques, also including MapReduce, the files and data collection techniques, and the monitoring platform building techniques. Currently, many network service companies, such as Yahoo and Facebook, use Hadoop as the core techniques to process their ever-increasing quantities of multimedia data. Other companies also integrate Hadoop and its related techniques to meet their own different demands. Such applications range from the searching engines, the advertisement benefit analysis, the user behavior analysis, to the multimedia social network analysis. The study builds a Hadoop large-scale multimedia data platform and integrates the related large-scale data applications. The large-scale multimedia data storage and computing techniques serve as a new trend of network service, and bring a new wave of prosperity for the large-scale multimedia data cloud computing. The study implements an example of applying the cloud computing to conduct the online processing and analysis. The implementation not only highlights the advantages of cloud computing over the traditional data processing practices, but also indicates the applicable framework designs and tools available for the large-scale data processing.

To work towards a higher-speed large-scale multimedia data storage and processing framework, future works are recommended to focus on comparing the time difference between using the current computing mechanism and using the Hadoop techniques within the same data range, the response time difference of query and summarized computing by adopting HBase or not, and the performance difference between using the Hive instructions and adopting the self-written MapReduce programs.

Acknowledgments Our deep gratitude goes to Chunghwa Telecom, Broadband network lab, which provided the research related platform for the test and evaluation of our research work.

References

1. Mundkur P, Tuulos V, Flatow J (2011) Disco: a computing platform for large-scale data analytics. In: Erlang'11: Proceedings of the 10th ACM SIGPLAN workshop on Erlang, New York, NY, USA, ACM, pp 84–89
2. HDFS Architecture Guide (2010) The Apache Software Foundation. http://hadoop.apache.org/hdfs/docs/r0.21.0/hdfs_design.html
3. Taylor R (2010) An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC Bioinforma 11(Suppl 12):S1

4. Ghemawat S, Gobioff H, Leung ST (2003) The Google File System. In: Proceedings of the 19th ACM Symposium on operating systems principles, pp. 20–43
5. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
6. Chang F et al (2006) Bigtable: a distributed storage system for structured data. In: 7th USENIX symposium on operating systems design and implementation (OSDI), pp 205–218
7. Welcome to Hadoop Common! (2010) Welcome to Hadoop Common!, The Apache Software Foundation. http://hadoop.apache.org/common/. Accessed 7 Sept 2010
8. Khuc V, Shivade C, Ramnath R, Ramanathan J (2012) Towards building large-scale distributed systems for twitter sentiment analysis. In: ACM symposium on applied, computing, pp 459–464
9. https://github.com/cloudera/flume
10. http://hbase.apache.org/
11. Liao I-J, Tin-Yu W (2011) A novel 3D streaming protocol supported multi mode display. J Netw Comput Appl 34(5):1509–1517
12. Tin-Yu W, Chen C-Y, Kuo L-S, Lee W-T, Chao H-C (2012) Cloud-based image processing system with priority-based data distribution mechanism. Comput Commun 35(15):1809–1818
13. Babu S (2010) Towards automatic optimization of MapReduce programs. In: Proceedings of the first ACM symposium on cloud computing (SOCC)
14. Stonebraker M, Abadi D, DeWitt DJ, Madden S, Pavlo A, Rasin A (October 2012) MapReduce and Parallel DBMSs: friends or foes. Commun ACM 53(1):64–71
15. http://www.hadoop.iponweb.net/Home/hdfs-over-ftp
16. Chaiken R, Jenkins B, Larson P-A, Ramsey B, Shakib D, Weaver S, Zhou J (2008) SCOPE: easy and efficient parallel processing of massive data sets. In: Proceedings of the very large database endowment, vol 1, no. 2, pp 1265–1276
17. http://ganglia.sourceforge.net/
18. Thoms E (1997) OLAP solutions: building multidimensional information systems, 2nd edn. Wiley, New York. ISBN 978- 0471149316
19. Lemire D (2007) Data warehousing and OLAP—a research-oriented bibliography. http://www.daniel-lemire.com/OLAP/
20. Lakshmanan LVS, Pei J, Zhao Y (2003) QCTrees: an efficient summary structure for semantic OLAP, SIGMOD
21. Eavis T, Taleb A (2012) Towards a scalable, performance-oriented OLAP storage engine. In: Lee SG, Peng Z, Zhou Z (eds) Database systems for advanced applications. Lecture Notes in Computer Science, vol 7239. Springer, Berlin, pp 185–202
22. Bennett C, Grossman RL, Locke D, Seidman J, Vejcik S (2010) MalStone: towards a benchmark for analytics on large data clouds. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (KDD)
23. Van der Aalst W (2012) Process mining: overview and opportunities. CM Trans Manag Inf Syst 3(2):7:1–7:17
24. Apache Sqoop (TM) is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. http://sqoop.apache.org/ (2013)
25. Mirajkar N, Barde M, Kamble H, Athale R, Singh K (2012) Implementation of private cloud using eucalyptus and an open source operating system. Int J Comput Sci Issues (IJCSI) 9(3):360–364