Indian Journal of Science and Technology, *Vol 8(18), DOI: 10.17485/ijst/2015/v8i18/75864, August 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Bigdata Platform Design and Implementation Model

Kyoo-sung Noh[1] and Doo-sik Lee[2*]

[1]Business Administration, Sunmoon University, Asan, 336-708, South Korea;
ksnoh@sunmoon.ac.kr
[2]IMCLOUD Corporation, Seoul, 133-070, South Korea;
dslee@imcloud.co.kr

Abstract

Bigdata software platform technology using Hadoop Ecosystem is the essential element and also underlying technology for application software or service implementation of bigdata analysis. It is required to have bigdata platform technology that can ensure the scalability, reliability and high performance of system for processing and analyzing a variety of bigdata related tasks including structured data, unstructured data, semi-structured data, etc. Bigdata platform can process large amounts of data in parallel unlike those conventional application software solutions and it is an easily scalable system. Its technical components include collection (Flume and Sqoop), storage (Hadoop and NoSQL), search (Solr), analysis (R Analysis), visualization (Node.js), scheduler (Oozie), etc. The purpose of this study is to propose an optimized bigdata platform implementation model through S/W configuration based on open source.

Keywords: Bigdata, Bigdata Visualization, Flume, Hadoop, MAP/REDUCE, Oozie, Platform, R Analysis, Sqoop, SQL on Hadoop

1. Introduction

Bigdata platform is software that uses computing power based on distributed parallel method in order to process large amounts of data. It consists of such technical components as data collection, storage, processing, search, analysis and visualization. In addition, bigdata platform implementation model is a technical method to guarantee the scalability, reliability and high performance of platform; thus, it is an essential element for building all kinds of bigdata systems. This thesis studied the model to implement bigdata software platform using Hadoop Ecosystem and the implementation method for each function[1]. In this model, Flume and Sqoop were utilized for data collection and Hadoop was utilized for data storage. As for data processing, MAP/REDUCE processing method using Hive and Job scheduling using Oozie were utilized. As for data visualization, web-service visualization method based on asynchronous event using Node.js is proposed[2].

2. Implementation of Bigdata Platform

2.1 Software Architecture

Bigdata platform consists of the following elements: collection in large amounts, storage, processing, search and visualization. Each function is operated by an independent software engine and mutually operated in various manners with the other functions simultaneously. Bigdata platform collects and processes data from the primary data sources in which data is first generated and then outputs the final analysis results using storage, analysis

*Author for correspondence

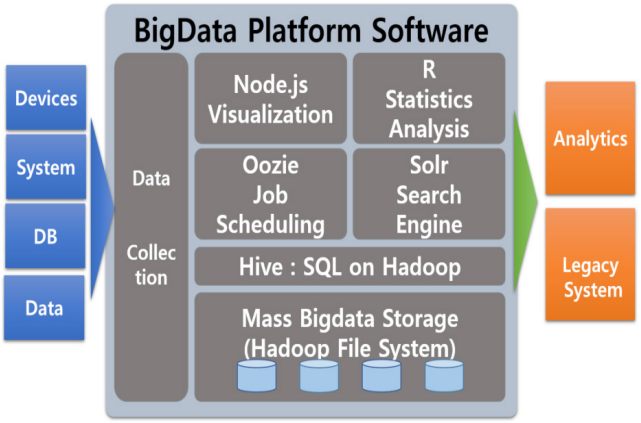

Figure 1. Bigdata platform S/W architecture.

and search processing. The process of outputting requires visualized analytical feature.

2.2 Applied Open Source Technique

As Table 1 shows, the Bigdata platform is configured with the connection to various open sources based on Hadoop. Those open sources called Hadoop Ecosystem is mainly consisted of the integration of connection and function of data based on Hadoop. Flume engine is utilized as for log data collection processing and Sqoop is utilized as for data collection from RDMBS. HIVE processes data stored in Hadoop based on MAP/REDUCE method and also have the function to process the stored data in Hadoop through SQL method as managing data schema by itself. Solr is a search engine based on Lucene, which conducts the required indexing task for search in batch by using the original data of Hadoop. When a key word is entered in the search server, it performs the role of searching data promptly by configuring a search ranking through the already created indexing task. R engine is an engine to analyze the data based on statistics; thus, it distinguishes the pattern and characteristics of data by using various statistical functions. Bigdata visualization is a suitable method for bigdata processing; thus, it configures web-services by using such visualization techniques as HTML5, ECMA Script, JQuery, JavaScript, JSON, etc. In particular, it configures web-services by using Node.js that processes scripts of clients at a server through an asynchronous inter-printer method.

Table 1. Open sources used in bigdata platform

Type	Open Source	Main Function
Collection	Flume Sqoop	-Log data collection and processing, -Data of RDBMS collection
Storage	Hadoop	-Save original data inHDFS, -Parallel distributedprocessing Map/Reduce, -Data copy and Autorecovery
Processing	Hive Oozie	- SQL query processing, - Job scheduling
Search	Solr	- Indexing - Searching
Analytics	R Mahout	-Data analytics based onstatistical functions -Data analytics based on machine learning
Visualization	Node.js	- web service forvisualization usingjavascript
Streaming Processing	Storm, S4,Spark	- Real time data streaming
Management	Zookeeper Ambari	- Hadoop Name node clustering - Hadoop Data node management

2.3 Bigdata Collection

2.3.1 Log Collection Engine Flume

It collects data using Flume engine as for data collection and processing (Figure 2). Flume is suitable for collecting large amounts of logs generated at the security or network equipment. Log data generated at the equipment is generally made up of Syslog data type and Flume supports for Syslog data type. Flume collects data by using TCP or UDP standard protocols and then saves it in Hadoop at a high speed. When a collection function is for large amounts, Flume can generate and configure numerous Flume agents in terms of software. Flume Agents can create even more Flume agents when a server has more physical memories because it is software that makes several memories reside in a server. In addition, Flume agents may be configured with multiple physical servers. Flume agents are configured internally with the three elements including source, channel and sink. Source set data collection source, whereas channel generates data temporarily in memories and hard disk and then saves it through sink. Sink sets a place where data is saved. In generally, it is set with Hadoop; however, data can be sent again to source of other Flume. In conclusion, Flume engine has the functions of scaling out large amounts of log data in parallel by saving them in Hadoop smoothly and also collecting and processing data very flexibly.

2.3.2 RDBMS Data Collection Engine Sqoop

Sqoop engine is used as for data collection from database. Sqoop retrieves data from database, the conventional RDBMS such as Oracle, MySQL, MS-SQL, etc., by using SQL gate. Then, it saves data in Hadoop. Sqoop is a command-line interface engine that retrieves data at a high speed through JDBC connection, which is the RDMBS connection technique standard. In particular, Sqoop provides the function of processing data in parallel at Hadoop DataNode server by MAP/REDUCE method when saving data in Hadoop. In addition, Sqoop may also have the function of sending data in Hadoop to RDBMS in the opposite direction. Any failure occurred during data transfer process has the ability to process exceptional cases.

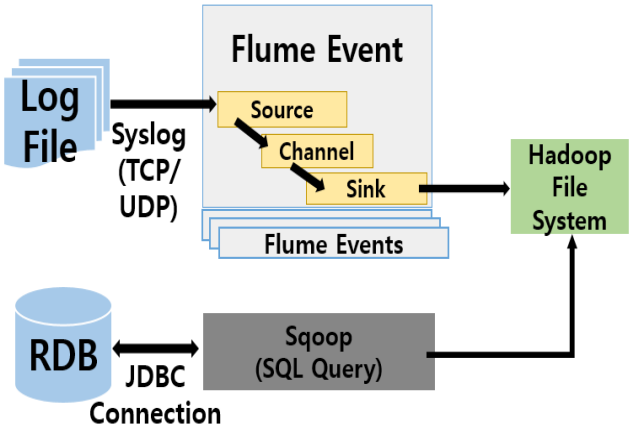

Figure 2. Bigdata collection process.

2.4 Bigdata Storage

2.4.1 Data Replication Function of Hadoop

As Figure 3 shows, Hadoop is bigdata storage technology based on HDFS (Hadoop File System)[3,4]. Hadoop saves data physically in a 64 MB data block unit[5]. This data block is randomly saved at multiple physical DataNode servers. Multiple DataNode servers configure a Ring-Node in a logical way and multiple physical Data Node servers are operated as though they are one data storage place[6]. In particular, Hadoop provides the automatic data replication function when saving data. This function is the technology to keep multiple data all the time as the already replicated data block automatically recovers data in another DataNode server when some of the 64MB data storage places have a physical failure. It provides the function to ensure that data will never be broken. The replication function has the three equivalent data blocks by default; however, replication numbers can be set by a number desired by users. Hadoop automatic recovery function does not require back-up; as a result, it is the data storage technique to allow for infinite extension while increasing the number of servers.

2.4.2 MAP/REDUCE Distributed Parallel Processing

Hadoop provides the distributed parallel processing function in the processing phases such as data storage and query through MAP/REDUCE technique. When processing a very large data size, the required number of Hadoop DataNodes may be very high. In such case, it will take too much time if Hadoop data are processed sequentially. MAP/REDUCE technique does not process data sequentially. Instead, it generates even faster result than the sequential processing method as processing each data node in parallel and computing the processed result

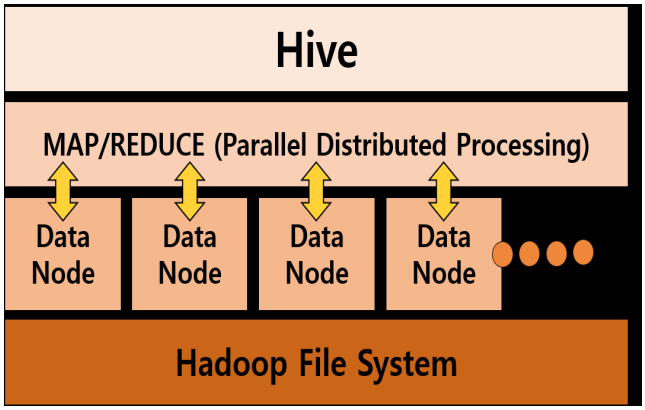

Figure 3. Bigdata parallel distributed processing and storage architecture.

for the last time. This technique is not very effective when data size is small; however, it can be very effective when data size is large.

2.5 Bigdata Processing

2.5.1 Data Structure Processing using Hive

Hadoop to save bigdata saves data in its own file system called HDFS[7]. On this account, it is required to have a way to structure and retrieve data in order to query and search original data. At this time, Hive is software engine to be used when Hadoop is installed. It manages the data schema as to the original data saved in Hadoop through the table method. It is possible to save database, table, column, etc. as to original data and query data using a simple SQL syntax. In addition, Hive provides its own MAP/REDUCE function; thus, it automatically provides distributed parallel processing function when saving data in Hadoop. As a result, Hive is the data processing engine to process data in Hadoop through SQL method while MAP and REDUCE is automatically operated.

2.5.2 Data Process Scheduling using Oozie

All the processes to collect save and process data are generated by the scenario method work-flow concept; thus, it is required to have a process scheduling to gather all of the processing units and process them at once. Oozie is in charge of an important job scheduler function in terms of processing bigdata. For instance, it is possible to schedule that three jobs are to be generated sequentially when assuming that there are three jobs (task unit) to be collected, saved and queried by Oozie. Also, it is equally possible to schedule in advance through work-flow concept so that another job will be conducted after one job is completed. The process of processing and analyzing bigdata is very complex. Also, it is consisted of various jobs. As a result, the job scheduling function is one of the required functions of Bigdata platform.

2.6 Bigdata Search

Bigdata is large sized data; thus, it cannot be queried promptly in real time. Search engine can solve this issue. The data search engine using Solr conducts indexing process as to the original data. Indexing configures a search ranking in advance, which allows users to search data promptly using those already defined keys with the batch processing method. The index data is operated based on the method that Solr engine searches data promptly using indexing data when entering search keywords or conditions at Solr. The search engine is not a real-time data

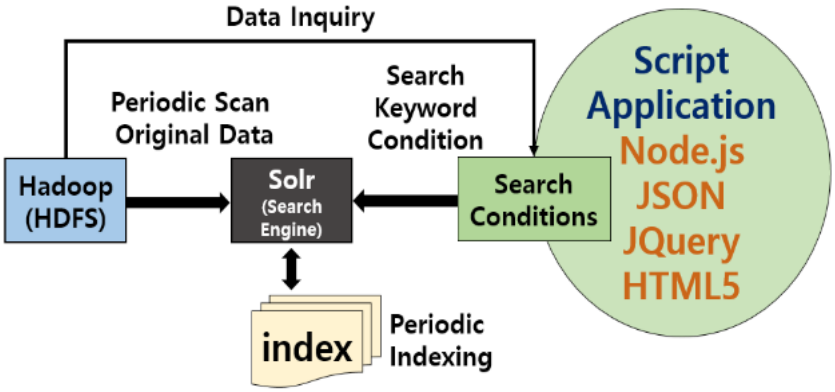

Figure 4. Bigdata search architecture based on solr.

search and it conducts indexing of data in advance so that users can feel it in real time. Thus, it is also called "Near Realtime". For instance, those search services as Google are operated based on this method. Solr engine has a similar function to those engines used in the portal services like Google and it can be applied to Hadoop (Figure 4).

2.7 Bigdata Analysis

2.7.1 The Data Analysis using R Engine

There are several ways to analyse bigdata[8]. The analytical techniques for bigdata include statistical technique, datelining technique, mechanical learning technique, etc. In particular, R is frequently used for analysing bigdata in recent years. R is an analytical engine with statistical technique that aggregates the statistical function libraries. R provides RServe that can be used in servers other than R studio to be used in PC. RServe ensures that data pattern can be compared and analysed with statistical techniques using data searched through original data or Solr saved in Hadoop. However, users should create a necessary rule for analysis rather than it is automatically analysed. This rule can become a pattern for comparing data or a comparison of history. Moreover, it can become specific information for detecting the special attributes of data. R is an engine to be used when designing a statistical function and using an actual statistical function, which would be used in the analysis logic after users plan algorithms and analytical logics to create analytical rules. Bigdata allows for associated analysis or integrated analysis by combining data lacking mutual relationship rather than comparing the equivalent data. R engine provides statistical functions that can be effectively used for these analytical techniques.

2.7.2 The Data Analysis of Machine Learning using Mahout

Data analysis can be performed by using not only the statistical technique, but also Mahout. Mahout can implement the machine learning algorithm by utilizing the data processing techniques such as Collaborative Filtering, Classification, Clustering and Dimensionality Reduction. Mahout supports the libraries that support the distributed process and the expansion to perform the machine learning. Collaboration filtering supports User-Based, Item-Based and Model-Based Methods. User-based method recommends the item by finding similar users. Item-based method recommends after calculating the similarity between items. In other words, the understanding level of similarity between the user and the evaluation data is calculated. Clustering method automatically groups similar items in various data. This method divides a large data to small data or groups small data to a large data. Categorization method tags and groups documents and, then, calculates various statistics related to the characteristics of tagged document. This method may characterized by anything that may include words, frequencies of words and parts of speech related to the tagged document. Categorization method may impact the result in a great deal depending on what algorithm is used.

2.7.3 The Bigdata Analysis Process

Collected big data can be utilized only after being processed and analysed. For the big data analysis, first, what is the issue under the question should be found and data related to such issue should be collected. The collected data can be used for the data analysis process to draw the result only after works of classification, process and transference. The analysis result can be difficult to obtain with one trial. Additional data analysis works can be repeated for several times by verifying the result and determining the value of analysis result. The process of big data analysis may defer in each analysis process so that the effective result can be acquired only by repeated performances of various methods (Figure 5).

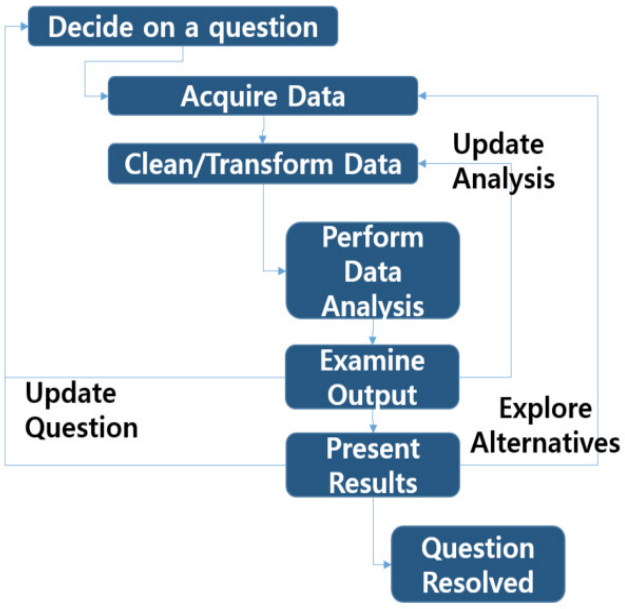

Figure 5. Bigdata analytics process.

2.8 Bigdata Visualization

2.8.1 The Implementation of Data Visualization Service utilizing Node.js

Bigdata platform software requires visualization technique to confirm the analytical results from user perspective even though it performs the roles of collecting, saving, processing, analyzing, and searching data. Although there are several visualization methods for bigdata, it is possible to visualize by creating data in the form of web-service through those standard web-script techniques such as HTML5, JQuery, JSON, Java Script, ECMA Script, etc. In particular, it is possible to configure a service by applying it to a server easily if Node.js is used. In the past, it was imperative to use a separate server-side script programming technique as for web servers unlike the script techniques to be used in clients when configuring services by visualizing data. This has been developed into a variety of middleware web-service programming technologies. In the meantime, a new technique was emerged in recent years. It allows users to use the script techniques used in clients at servers in order to configure visualized services after processing bigdata very easily. This method can be easily implemented by using Node.js. Node.js is an engine that Chrome web browser implemented a technical structure to process scripts at web servers by employing V8 engine developed by Google. It processes events asynchronously; thus, it has the web service technology to process large amounts of client web browser requests simultaneously. In addition, it can visualize bigdata analytical information at clients by using D3 (Data Driven Documents) technique. Node.js provides a convenient visualization technique; however, it requires a conjunction to the open source of Hadoop Ecosystem. Thus, it requires a module dedicated for Node.js that can be linked with Hadoop, Hive, R, Solr, Sqoop, Flume, etc. (Figure 6).

2.8.2 The Implementation of user Interface Employing D3 Technique

To visualize the big data, the user interface technology with an effective expression technique is required. D3 (Data Driven Documents) provides various components for the big data visualization and can express the data group, connected relation, map expression, chart and others. D3 technology is basically a library for development possible to be implemented based on Javascript and

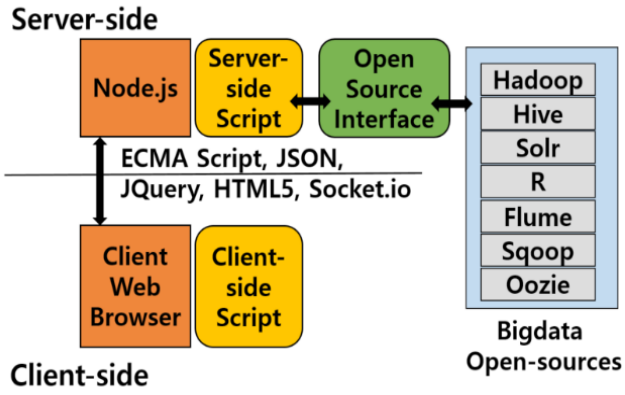

Figure 6. Bigdata visualization based on Node.js.

enables selection and application of the most suitable analysis UT according to the data analysis result.

2.9 Bigdata Real-time Streaming Processing

The Real-time Data Stream Process using Strom and S4: The big data platform software process large sized data so that the real-time data process is required. A search engine using the data query or Solr through MapReduce can rapidly call data from Hadoop where large sized data are stored. However, the alarm activated appropriately to a condition by processing the speedily processed data in real time is hard to be implemented with the existing technology. If open source technologies such as Storm or S4 are utilized, various conditions can be set on data entered in real time in the collection phase. If data appropriate to the set condition is entered, the service that can search and activate the alarm in real time can be implemented. Especially, Storm and S4 are composed of the software engines optimized to Hadoop and MapReduce to be proper for the real-time data search and process.

2.10 Bigdata Security

Then encryption of stored data, the authentication on the storage connection and the access authorization for users to data are required for the big data security. The data encryption codes the data using the encryption algorithm knows as Public Key or Private Key to store the data and decodes by using Key on the data search. In the big data platform, the data encryption process may slow the speed of decoding or searching the store data so that the security should be carefully applied by considering the performance. Also, since the data storage of Hadoop divides the partition and stores data differently for various types, the

process function for the access and connection authorization is required.

2.11 Bigdata Sizing

For the big data system, all servers for collection, storage, search, analysis, visualization should basically be equipped with the Scale-out feature[9]. Hadoop is basically featured for the convenient physical expansion of Data Node in one cluster and can be easily expanded for the purpose and/or the performance. The collection, search and analysis engines can perform the distributed parallel process and the MapReduce process of Hadoop by physically expanding many servers in the load to servers is excessive. By using the expansion technology of big data open source, the specific sizing method for Memory, Hard Disk, CPU and Network Speed of a server can be determined. Especially, Hadoop storage simultaneously performs roles to save the data and to process data such as MapReduce so that, while the data capacity should be considered, the Disk I/O for the data process should be considered on the size determination. For Hadoop server, according to the purpose either for light data process, excessively large data storage, frequent process of very complicated data such as MapReduce or combination of all functions, the number and the performance of Data Node and Name Node are determined. Especially, in case of many MapReduce, the minimum 25% of free space in the entire data storage is recommended to perform the disk I/O and the data process fluently.

2.12 Bigdata Management

If the storage space and the data process are largely expanded, maintenance issues occur on the server, the operating system and the software for the big data system and the operation software is essentially required to prevent any loss of stored data. Especially for the data storage Hadoop, the system stability is secured only by well maintenance of Name Node's Meta information. Zookeeper provides clustering measures to stably manage Name Node's Meta information. And Ambari provides the monitoring and control functions for easy operations of multiple Hadoop Data Nodes. For Hadoop Data Node, if the storage is expanded to an excessively large scale, very many resources are used to manage servers and the operating system itself so that effective resources and maintenances are essential.

3. Conclusion

It is required to have a variety of techniques and methods in order to process bigdata. Even though there are various techniques, those essential elements for collecting, saving, processing, searching, analyzing and visualizing data would not change over time. As a result, it is possible to configure a bigdata platform model by gathering these elements. It will be possible to conduct analytical and programming tasks more easily on top of platform if a bigdata platform is configured. On this account, an important issue is which model should be used to implement a bigdata platform. This thesis proposed an implementation model for bigdata platform based on the essentially required elements for bigdata analysis. This technique exists as a platform software technique to process bigdata since it is packaged by the open source techniques called Hadoop Ecosystem. These techniques have been consistently upgraded. The Apache Foundation has been internationally leading and developing the technology standard. Commercially, those companies and organizations such as Cloudera, HortonWorks and MAPR Technologies in the United States and IMCLOUD Corp., KT NexR, Gruter, Flamingo open source, etc. in Korea provides the distribution version of bigdata platform software to users. As the open source technology for processing bigdata, the method of processing large amounts of data more promptly and in real time and the analytical rules for each industry have been significantly developed. The importance of bigdata platform software technologies, which are proprietary technologies, are growing bigger and bigger. It is expected that the real time data processing and analytical techniques will be developed further in the future. Thus, it is required to conduct follow-up studies on this field. In addition, it is required to continue to conduct research and development on the implementation models of bigdata platform in order to improve the ability of performance, stability and data processing.

4. References

1. Simmhan Y, Aman S, Kumbhare A, Liu R. Stev: cloud-based software platform for big data analytics in smart grids. Computing in Science and Engineering. 2013; 15(4):18–34, 38–47.

2. Wang S, Su W, Zhu X, Zhang H. A Hadoop-based approach for efficient web service management. International Journal of Web and Grid Services. 2013; 9(1):5–13.
3. Kim W. Web Data Stores (aka NoSQL databases): a data model and data management perspective. International Journal of Web and Grid Services. 2014; 10(1):100–10.
4. Benzaken V, Castagna G, Nguyen K, Simeon J. Static and Dynamic Semantics of NoSQL Languages. ACM SIGPLAN Notices. 2013; 48(1):101–14.
5. Wang J, Cheng L, Wang L. Concentric layout, a new scientific data layout for matrix data–set in Hadoop file system. International Journal of Parallel, Emergent and Distributed Systems. 2013; 28(5):407–33.
6. Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA. Bigtable: A Distributed Storage System for Structured Data. ACM Transactions on Computer Systems. 2008; 26(2):4.
7. Xindong W, Xingquan Z, Gong–Qing W, Wei D. Data mining with big data. IEEE Transactions on Knowledge and Data Engineering. 2014; 26(1):97–107.
8. Steed CA, Ricciuto DM, Shipman G, Smith B. Big data visual analytics for exploratory earth system simulation analysis. Computers and Geosciences. 2013; 61:71–82.
9. Tan YS, Tan J, Chang ES, Lee B-S, Li J. Hadoop framework: impact of data organization on performance. Software. 2013; 43(11):1241–60.