Journal of Computers Vol. 31 No. 2, 2020, pp. 127-140
doi:10.3966/199115992020043102012

Big Data Analysis Based on Hadoop Cluster and Spark Cluster on Linux Platform

Kangxu Liu, Guangming Li*

School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
liukangxu@mail.sdu.edu.cn, gmli@sdu.edu.cn

Received 26 June 2018; Revised 8 October 2018; Accepted 18 December 2018

Abstract. In recent years, big data is widely mentioned worldwide, and its analysis technology has become one of the most popular research subjects. This paper adopts a practical case (movie datasets) to present big data development process. We propose a solution that combines Hadoop cluster and Spark cluster on Linux platform. The proposed idea lies on that Hadoop cluster stores data files and Spark cluster processes data files on Linux platform. In addition, deployed Linux platform can be used to process big data files in various different fields. What's more, Scala in Spark cluster plays a vital role in big data analysis technology, and it can achieve data mining of important information. Finally, the feasibility of the scheme is verified by an actual case analysis.

Keywords: big data analysis, data mining, Hadoop, Linux platform, Scala, Spark

1 Introduction

Big data is not only a technical hot word, but also a social wave in the era of information explosion. Big data has already permeated through every aspect of the society, and the big data analysis technology has been playing an increasingly important role in our daily life. The big data is not only a large amount of data, but more important is the complexity of the data [1]. The complexity of big data is reflected in two aspects: rapid change and complex type. Sometimes, it is significant to note that the small data in big data should also be deserved attention [2]. The properties of big data are as follows: massive-scale data (petabyte level), fast data flow, a variety of data types, and low value density. Nowadays, big data is ubiquitous, and it is indispensable in every aspect of life. However, it also has some problems of data security and privacy [3-4] in reality operation, such as identity card number, phone number, home address and so on. Therefore, we should pay more attention to protect the privacy information of users in the actual big data analysis.

There has been various big data analysis system that proposed in previous literature. Yi et al. [5] proposed a hybrid framework for data analysis under the background of big data. This method improves the speed and efficiency of data processing, but the data mining algorithm is too complicated for the application. Lin et al. [6] presented joint operation framework that combines R and Hadoop to solve the data analysis problems. The framework could greatly reduce the time of data calculation for data mining, but the R language is not one of the current popular computer programming languages, and it does not have been extensive popularization. Disha et al. [7] used Hadoop MapReduce framework to solve the data analysis problems, and this framework can reduce execution time of comparison performance graph. Besides, it also presented the way of protecting the privacy of users. What's more, Naïve Bayes Classifier model is used to predict the movie review, but this model is too complicated in actual big data analysis. Gulzar et al. [8] used Spark cluster to help users to mine relevant important data in big data analysis system, and it can support interactive ad-hoc analytics. Yang et al. [9] presented a hypergraph partitioning algorithm for data analysis in the era of big data. The performance of the algorithm is better, and the quality of data processing is high. What's more, this algorithm has a low communication cost,

* Corresponding Author

and it can maintain a balanced workload across CPUs. However, the idea is still in the experimental stage and has not been applied to the actual application. Feller et al. [10] used Hadoop HDFS (Hadoop distributed file system) cloud for big data storage. By comparing with the traditional data storage methods, it is found that cloud environment can not only improve efficiency, but also save a lot of money cost. Ji [11] presented the big data analysis system in the medical industry, which proves the value of the data mining. Many healthcare industry problems can be solved by big data analysis technology. Therefore, the big data analysis technology is one of the trends in future development.

Based on the above discussions, in order to solve big data analysis questions, this paper adopts solution that pseudo-distributed Hadoop cluster [12-14] should be configured on Linux platform, and then the Spark cluster [15-16] is deployed on Hadoop cluster. Finally, datasets of cloud storage in Hadoop HDFS use the Scala [17] in Spark cluster to achieve data mining.

Compared with other researchers' methods, the major contributions of this paper are as follows: to begin with, this paper adopts combination of Hadoop cluster and Spark cluster to process big data. The advantage of cluster combination is to effectively improve the efficiency and save big data processing time. Data files are stored in the Hadoop HDFS, and Spark cluster complete data files processing. It is rather than a single storing and processing data on Hadoop cluster in previous works. It is the future development trend of big data analysis that Hadoop cluster combines with the Spark cluster. Besides, deployed Linux platform has extensive applicability, and it can be used to process big data files in all kinds of fields, such as health care, business analysis, national security, food safety, and so on. What's more, Scala in Spark cluster is used to realize data mining. Aiming at solving the same one question, Scala only uses 10 lines of code, while Java needs 200 lines of code. Generic paradigm, regular expression, and higher-order function are introduced into Scala to solve the code redundancy questions, and this language is one of the most popular languages to process big data.

The rest of this paper is organized as follows. Section 2 starts with a brief review of related work, and then the proposed method is described. Linux platform's deployment of Hadoop and Spark are designed in Section 3. Big data case analysis is presented in Section 4. Conclusions and remarks on possible further work are given finally in Section 5.

2 Related Work

The related work mainly includes two parts: Linux platform's deployment and big data case analysis, which will be detailed described in Section 3 and Section 4.

Compared with Linux operation system, the windows operation system has disadvantage of lower compatibility. The Hadoop official only provides binary files for windows operation system, and some components need to be compiled by users. Therefore, most of the clusters are more appropriate to be deployed on Linux platform rather than windows platform.

Aiming at platform deployment, pseudo-distributed Hadoop cluster should be configured on Linux platform, and then Spark cluster should be deployed on Hadoop cluster. The core of the Hadoop cluster is MapReduce and HDFS. MapReduce is a programming paradigm that can be used to process distributed data. MapReduce can be divided into three stages to process distributed data, the first stage is map (the same operation is applied to each target in the set), the second stage is shuffle (the same key of key-value pair is saved into the same set), and the third stage is reduce (the traversing set elements return a comprehensive result). Each stage has inputs and outputs of key-value pairs, and the example of processing distributed data using Hadoop MapReduce is shown in Fig. 1. The function of HDFS is to split a large file into a small data block for solving the data files storage problem. NameNode is primarily used to store metadata (address information), and DataNode is used to store data block. Master-slave structure of HDFS is shown in Fig. 2.

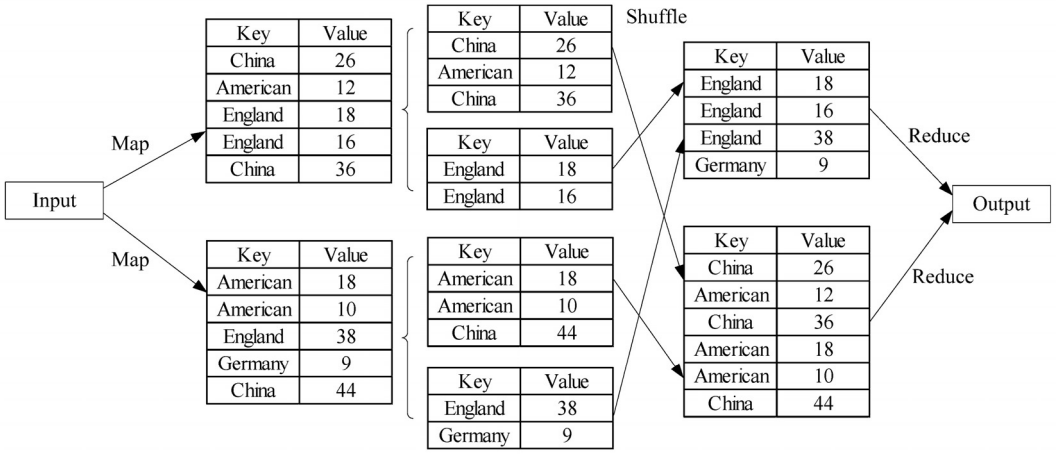


Fig. 1. Example of Hadoop MapReduce

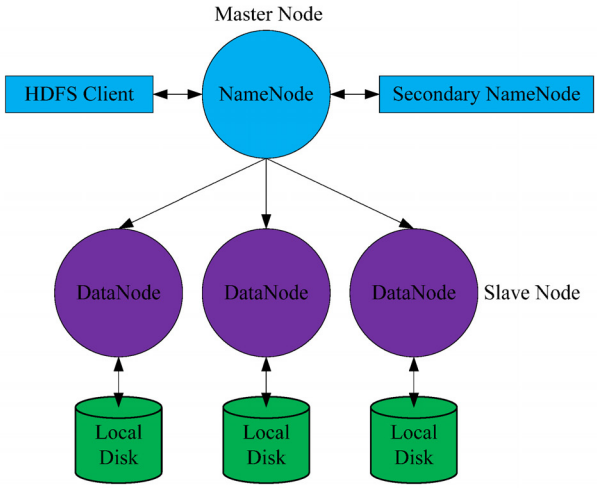


Fig. 2. Master-slave structure of HDFS

However, the limitations of Hadoop MapReduce framework for processing data are as follows. Firstly, it only supports two operations: Map operation and Reduce operation. Secondly, iterative computation (machine learning and graph computing) has lower computational efficiency. Thirdly, it is not suitable for an interactive processing (data mining). Fourthly, MapReduce programming is not sufficiently flexible. Based on the above analysis, aiming at processing data, the Spark cluster is a good supplement to the Hadoop cluster. Spark is running programs up to 100 times faster than Hadoop MapReduce in memory, or 10 times faster than that on disk [18]. The reason why Spark speed is so fast is that it is based on memory calculation and DAG (directed acyclic graph) algorithm. Otherwise, the Spark provides extensive API (application programming interface) for development and perfectly integrates with Hadoop HDFS. Therefore, this paper adopts that Spark cluster completes the calculation work of data.

The core ideas of big data are as follows: data are immobile, and calculation is mobile, and data are high-concurrency process. The solutions that data are processed by high-concurrency system architecture are as follows. Firstly, clusters (static resource clusters and application cluster) that multiple servers have the same functions mainly play the role of diversion. Secondly, different services will be placed on different servers for improving the processing speed of the data request, and processing one data request may requires multiple servers.

The mainly languages of big data processing include R, Scala, Python, and Java. Compared with other computer programming languages, Scala is characterized by the following. Firstly, Scala runs in the JVM (Java virtual machine), and it is also a language that drives Spark and Kafka and succeeds in combinations of function paradigm and object-oriented paradigm. Secondly, Scala can randomly access the “Java ecosystem”, and it includes many useful programming functions (pattern matching and sample

case) that are considered to be more concise than standard Java. Thirdly, Scala also includes a convenient REPL (Read-Eval-Print-Loop) to achieve interactive development and analysis, like as R and Python. Therefore, Scala is chosen for big data processing in this case, and Scala is very suitable for using in Spark cluster that focus on data transformation and mapping. Finally, Scala in Spark cluster relies on the IDEA (IntelliJ IDEA) software to realize big data development.

3 Linux Platform's Deployment of Hadoop and Spark

3.1 Hadoop Cluster Configuration

Firstly, VMware Workstation 10 [19] should be installed Linux system (CentOS 6). Due to three Linux virtual machines needed totally, we construct one master node and two slave nodes, which are named as master, slave1, and slave2, respectively. Then, master and slave nodes are assigned the internet protocol address. Secondly, the files (/etc/hostname, /etc/host) need to be modified in three different Linux virtual machines. Besides, the three virtual machines should configure user permission authentication. Thirdly, three Linux virtual machines need to be installed JDK (Java development kit) and configure environment variables and files. Finally, SSH (secure shell) needs to be configured in three Linux virtual machines to realize non-password login between nodes.

After the above basic configuration of Linux platform, Hadoop needs to be installed for master node by the Linux command (tar -zxvf hadoop-2.7.1.tar.gz). Besides, the corresponding configuration files (hadoop-env.sh, yarn-env.sh, slaves, core-site.xml, hdfs-site.xml, and mapred-site.xml) need to be modified by system requirements. Afterwards, configuration successful Hadoop in master node is copied to slave nodes by the Linux command of scp (secure copy). Finally, Hadoop cluster in master node can be started by the Linux command (sbin/start-all.sh). If Hadoop cluster is successful start-up, NameNode can be seen in master node by the Linux command of jps (Java virtual machine process status tool). DataNode also can be seen in slave nodes by the Linux command of jps.

3.2 Spark Cluster Configuration

When Hadoop cluster is successfully configured on Linux platform, the Spark cluster should be deployed on Hadoop cluster. Firstly, Spark needs to be unzipped and installed by the Linux command (tar-zxvf Spark-1.2.0-bin-hadoop2.4.tgz). Secondly, Spark needs to be configured with the environment variable files (Hadoop_Conf_Dir, Yarn_Conf_Dir, and HDFS_Conf_Dir). Finally, Spark cluster in master node can be started by the Linux command (sbin/start-all.sh). If the Spark cluster is properly installed, the work information of the slave nodes can be seen in master node by the browser (URL: http://master: 8080), as shown in Fig. 3.

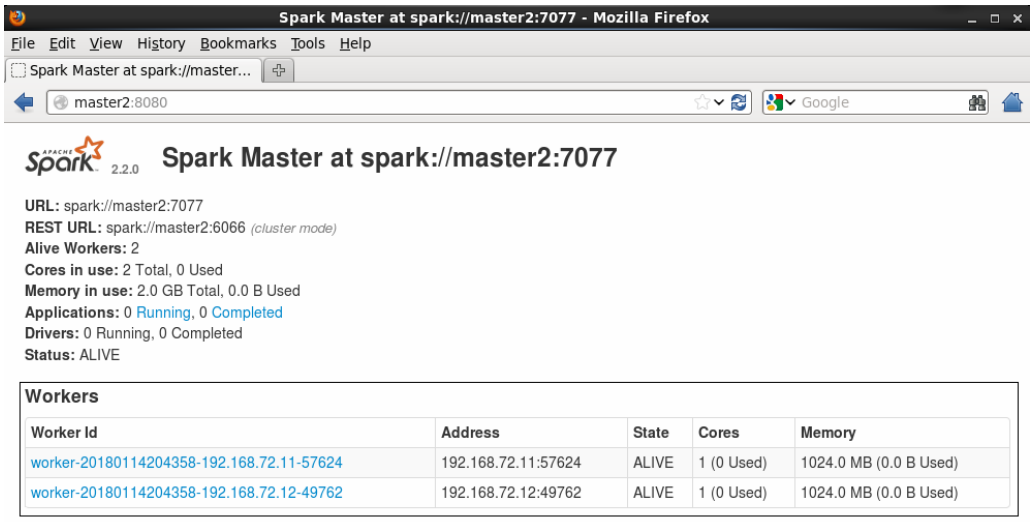


Fig. 3. Information of Spark cluster

Operation modes of Spark cluster include: Local mode, Standalone mode, Yarn mode, and Mesos mode. Firstly, Local mode is generally applied to the system test phase. Secondly, Standalone mode is default operation mode of Spark cluster, and it is a classic Master-Slave mode. Besides, one key factor is that standalone mode only needs to start the HDFS, and it not needs other clusters to manage entire cluster resources. Thirdly, Yarn mode or Mesos mode needs additional Hadoop clusters to manage entire cluster resources, and Mesos mode has been seldom used in domestic. Therefore, this paper adopts Standalone mode of Spark cluster to perform data processing.

The flow chart of big data processing by Spark cluster in Standalone mode is shown in Fig. 4. The components of Spark cluster include Driver (including Main, SparkConf, SparkContext, and RDD), Master (running in the master node), Worker (running in the slave node), Executor (specific task), and task. The first four are process, and the last one is thread. When Spark cluster detects the big data need to be processed, TaskScheduler (task allocation algorithm) is informed to prepare start-up by DAGScheduler (stage partitioning algorithm), and then TaskScheduler needs to be registered from master node. The purpose of registration to master node is that an application needs to be executed and needs to be allocated resources. After the master node receives the registration information, it communicates with worker node, and it requires worker node to start-up the corresponding Executor. Finally, when Executor is start-up, it needs to be un-registered with TaskScheduler, and TaskScheduler is informed that the current application is executed by which Executor. The purpose of reverse registration is that TaskScheduler can process specific tasks in Executor.

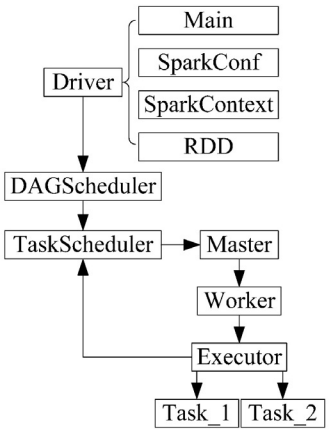


Fig. 4. Flow chart of big data processing

4 Big Data Case Analysis

When Linux platform is successfully deployed, a specific case (movies datasets) is chosen to realize the big data analysis. The experimental design mainly includes two steps: preparation of experimental data and preparation of experimental questions. The source of experimental data gets from the open source datasets of MovieLens system [20]. The setting of experimental questions is based on user actual requirements. The purpose of the experiments verifies that deployed Linux platform can be widely used in big data processing in various scenarios.

The flow chart of big data analysis system is shown in Fig. 5. Flume1 and Flume2 are used to collect data information. Flume3 is used to integrate collected data information from the web servers. If data does not need real-time processing, it can be solved by off-line batch processing. Data mining is completed by data processing at minute level. Otherwise, the Kafka is adopted to complete data cache, and data are processed by SparkStreaming/Strom for real-time processing. Data mining is completed by data processing at millisecond level. Finally, the useful big data information will be saved into the database (Redis, Hbase, and MySQL) and displayed to the users by Java-Web technology. In this paper, it adopts the way of real-time processing for realizing data mining of movies datasets.

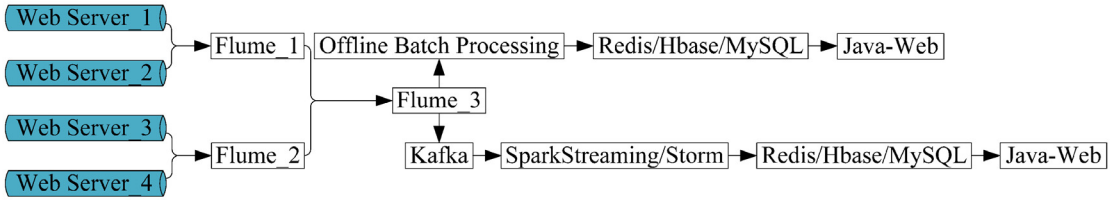


Fig. 5. Flow chart of big data analysis system

Aiming at the real-time data processing of the same magnitude, task execution time of single cluster (Hadoop cluster or Spark cluster) system approximately is 8-10s in previous research work. However, task execution time of combination cluster (Hadoop cluster and Spark cluster) system approximately is 1-2s in this paper. Compared with the traditional deployed platform, the task execution time shortens 87.5%, and the accuracy rate of task execution can reach 99.9%. Otherwise, this paper designs five experiments from different angles, such as task complexity, integration and extraction of key information, and different combinations of key value pairs. The experimental results further verify the superiority and feasibility of this system.

4.1 Data Structure

The three data files respectively are users.dat, movies.dat, and ratings.dat. These files contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens. The data structure of users.dat is as follows: UserID::Gender::Age::Occupation:: Zip-code, and it should be noted that the double colon (::) is used to split the different types of data information. Gender is denoted by “M” for male and “F” for female. Age is chosen from the following ranges: 1: “Under 18”, 18: “18-24”, 25: “25-34”, 35: “35-44”, 45: “45-49”, 50: “50-55”, 56: “56+”. Occupation is chosen from the following choices: 0: “other” or not specified, 1: “academic/educator”, 2: “artist”, 3: “clerical/admin”, 4: “college/grad student”, 5: “customer service”, 6: “doctor/health care”, 7: “executive/ managerial”, 8: “farmer”, 9: “homemaker”, 10: “K-12 student”, 11: “lawyer”, 12: “programmer”, 13: “retired”, 14: “sales/marketing”, 15: “scientist”, 16: “self-employed”, 17: “technician/ engineer”, 18: “tradesman/craftsman”, 19: “unemployed”, 20: “writer”. Partial data information of users.dat is shown in Fig. 6, and it totally contains 6,040 pieces of data. The data structure of movies.dat is as follows: MovieID::Title::Genres. Genres are pipe-separated and are selected from the following genres: Action, Adventure, Animation, Children’s, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western. Partial data information of movies.dat is shown in Fig. 7, and it totally contains 3,952 pieces of data. The data structure of ratings.dat is as follows: UserID::MovieID:: Rating::Timestamp. UserIDs range between 1 and 6,064, and MovieIDs range between 1 and 3,952. Ratings are made on a 5-star scale, and each user has at least 20 ratings. Timestamp is represented in seconds since the epoch as returned by time (2). Partial data information of ratings.dat is shown in Fig. 8, and it totally contains 1,000,208 pieces of data.

```
3660  3660::M::35::7::15137
3661  3661::M::35::12::08852
3662  3662::M::35::16::08753
3663  3663::F::35::9::17055
3664  3664::F::35::1::94043
3665  3665::M::25::0::95903
3666  3666::M::45::0::68144
3667  3667::F::1::10::10706
3668  3668::F::45::7::33324
3669  3669::M::25::14::60148
3670  3670::M::45::14::55123
3671  3671::M::25::4::98102
3672  3672::M::35::5::08406
3673  3673::M::25::2::10003
3674  3674::M::25::20::29205
3675  3675::M::35::7::06680
3676  3676::F::35::12::48109
3677  3677::F::25::12::06906
3678  3678::M::25::0::06880
3679  3679::M::25::4::68108
3680  3680::F::25::1::94939
```

Fig. 6. Partial data information of users.dat

```
3880::Ballad of Ramblin' Jack, The (2000)::Documentary
3881::Bittersweet Motel (2000)::Documentary
3882::Bring It On (2000)::Comedy
3883::Catfish in Black Bean Sauce (2000)::Comedy|Drama
3884::Crew, The (2000)::Comedy
3885::Love & Sex (2000)::Comedy|Romance
3886::Steal This Movie! (2000)::Drama
3887::Went to Coney Island on a Mission From God... Be Back by Five (1998)::Drama
3888::Skipped Parts (2000)::Drama|Romance
3889::Highlander: Endgame (2000)::Action|Adventure|Fantasy
3890::Back Stage (2000)::Documentary
3891::Turn It Up (2000)::Crime|Drama
3892::Anatomy (Anatomie) (2000)::Horror
3893::Nurse Betty (2000)::Comedy|Thriller
3894::Solas (1999)::Drama
3895::Watcher, The (2000)::Crime|Thriller
3896::Way of the Gun, The (2000)::Crime|Thriller
3897::Almost Famous (2000)::Comedy|Drama
3898::Bait (2000)::Action|Comedy
3899::Circus (2000)::Comedy
3900::Crime and Punishment in Suburbia (2000)::Comedy|Drama
```

Fig. 7. Partial data information of movies.dat

```
4272::2498::2::965301944
4272::2808::3::965301999
4272::1876::4::965301859
4272::260::4::965301462
4272::1126::2::965301318
4272::1127::5::965301664
4272::2105::4::965301892
4272::2116::3::965301715
4272::2117::3::965301638
4272::1196::4::965301522
4272::2140::3::965301715
4272::674::2::965301892
4272::1527::4::965301892
4272::1544::4::965301969
4272::1573::3::965301747
4272::1580::3::965301664
4272::1584::5::965301607
4272::2530::3::965301999
4272::2533::2::965301833
4272::3190::3::965302031
4272::1590::3::965302100
```

Fig. 8. Partial data information of ratings.dat

Finally, datasets are saved in Hadoop HDFS cloud by the Linux command (Hadoop fs –put users.dat /input/movies, Hadoop fs –put movies.dat /input/movies, and Hadoop fs –put ratings.dat /input/movies) on Linux platform, and datasets can be loaded from Hadoop HDFS cloud by the Linux command (hdfs: //master:9000/input/movies/xxx.dat) in Spark cluster. The specific work of data mining on selected datasets will be presented in the following subsections.

4.2 Quantity Distribution Analysis

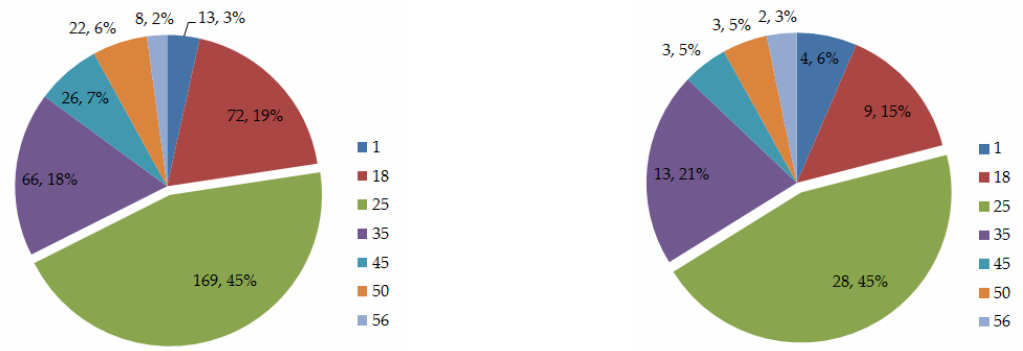
Experiment 1: What are the viewers who have seen the movie of "Lord of the Rings, The (1978)" quantity distribution of age and gender?

Firstly, datasets should be loaded from Hadoop HDFS cloud. Then, the movieId of "Lord of the Rings, The (1978)" should be obtained from the movies.dat. Besides, all user information (userId, (gender, age)) can be obtained from users.dat. Secondly, movieId should be matched userId from ratings.dat. Thirdly, the information (userId, (movieId, (gender, age)) can be obtained by join operation from users.dat and ratings.dat. Finally, it can get experimental results of quantity distribution of age and gender. Task execution time of data mining is 1.298597s, as shown in Fig. 9.

```
18/05/07 08:21:38 INFO DAGScheduler: Job 1 finished: foreach at BigData1.scala:35, took 1.298597 s
((F,50),3)
((F,45),3)
((M,56),8)
((M,50),22)
((F,35),13)
((F,18),9)
((M,18),72)
((M,25),169)
((M,1),13)
((M,35),66)
((F,25),28)
((F,56),2)
((F,1),4)
((M,45),26)
18/05/07 08:21:38 INFO SparkUI: Stopped Spark web UI at http://192.168.199.159:4040
18/05/07 08:21:38 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

Fig. 9. Results and task execution time of experiment 1

From the above experimental results, as shown in Fig. 10, we can draw following conclusions. For male viewers, people who have seen the film at age group between 25 and 34 have the largest percentage with 45% of the total of males. It reaches up to 169 people. People who have seen the film at age group over 56 have the minimum percentage with 2% of the total of males. It has only 8 people, as shown in Fig. 10(a). For female viewers, people who have seen the film at age group between 25 and 34 have the largest percentage with 45% of the total of females. It reaches 28 people. People who have seen the film at age group over 56 have the minimum percentage with 3% of the total of females. It has only 2 people, as shown in Fig. 10(b). In sum up, whether it is a male or a female viewer, the largest percentage are distributed at age group between 25 and 34, and the minimum percentage are distributed at age group over 65. The number of male viewers (376) is far beyond the female viewers (62) in the movie of "Lord of the Rings, The (1978)".



(a) male quantity distribution and (b) female quantity distribution

Fig. 10. Quantity distribution

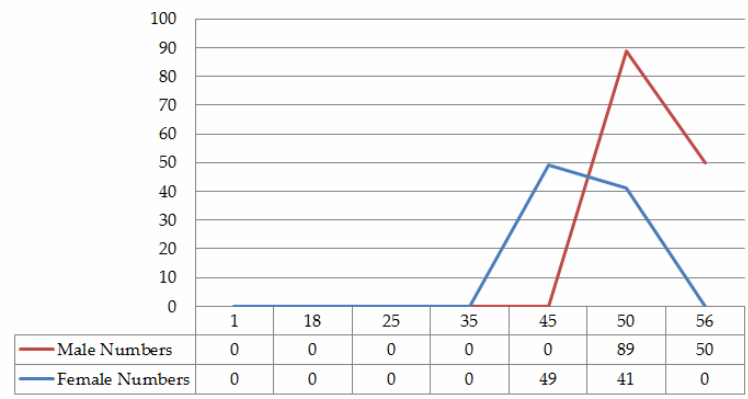
Experiment 2: What are the viewers who have seen the top three most popular movies in 1995 quantity distribution of age and gender?

Firstly, datasets should be loaded from Hadoop HDFS cloud. Secondly, moiveId of 1995 films should be obtained from movies.dat. What's more, the information (userId, movieId, ratings) in 1995 movies should be filtered from ratings.dat, and average rating should be calculated for getting the top three most popular movies in 1995. Based on the above algorithm analysis, the top three most popular movies in 1995 are "Mr. Holland's Opus (1995)", "Leaving Las Vegas (1995)", and "Ghost in the Shell (Kokaku kidotai) (1995)", respectively. Thirdly, the information (userId, (movieId, (gender, age)) of the top three most popular movies in 1995 can be obtained by join operation from users.dat and ratings.dat. Finally, it can get experimental results of quantity distribution of age and gender. Task execution time of data mining is 1.114402s, as shown in Fig. 11.

```
18/05/07 08:31:37 INFO DAGScheduler: Job 2 finished: foreach at BigData5.scala:60, took 1.114402 s
((F,50),41)
((F,45),49)
((M,56),50)
((M,50),89)
((F,35),87)
((F,18),100)
((M,18),286)
((M,25),578)
((M,1),24)
((M,35),215)
((F,25),207)
((F,56),18)
((F,1),10)
((M,45),81)
18/05/07 08:31:37 INFO SparkContext: Invoking stop() from shutdown hook
18/05/07 08:31:37 INFO SparkUI: Stopped Spark web UI at http://192.168.199.159:4040
18/05/07 08:31:37 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

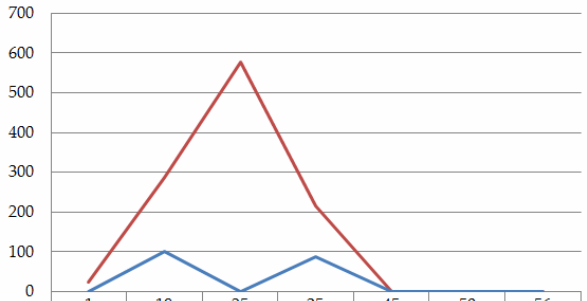
Fig. 11. Results and task execution time of experiment 2

Aiming at the movie of "Mr. Holland's Opus (1995)", the number of male viewers (139) is beyond the female viewers (90). The gap approximately is 50 people, as shown in Fig. 12. Aiming at the movie of "Leaving Las Vegas (1995)", the number of male viewers (1103) who have seen the film at four age groups is far more than the female viewers (187) who have seen the film at two age groups, as shown in Fig. 13. Aiming at the movie of "Ghost in the Shell (Kokaku kidotai) (1995)", however, male viewers who have seen the movie are only at age group between 45 and 55. It has only 81 people, as shown in Fig. 14. The number of female viewers (235) is far beyond the male viewers (81) who have seen the movies.



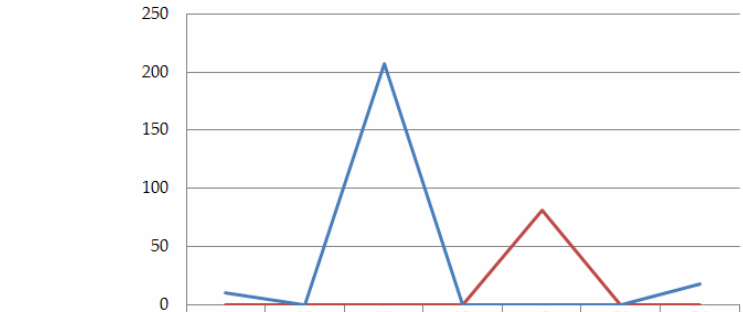
	1	18	25	35	45	50	56
Male Numbers	0	0	0	0	0	89	50
Female Numbers	0	0	0	0	49	41	0

Fig. 12. Analysis of "Mr. Holland's Opus (1995)"



	1	18	25	35	45	50	56
Male Numbers	24	286	578	215	0	0	0
Female Numbers	0	100	0	87	0	0	0

Fig. 13. Analysis of "Leaving Las Vegas (1995)"



	1	18	25	35	45	50	56
Male Numbers	0	0	0	0	81	0	0
Female Numbers	10	0	207	0	0	0	18

Fig. 14. Analysis of "Ghost in the Shell (Kokaku kidotai) (1995)"

Based on the above figures, we can draw conclusions that the first two movies are more popular with male viewers. However, the last one movie is more popular with female viewers.

4.3 Preference Analysis

Experiment 3: What are the top ten favorite movies of male viewers at age group between 18 and 24?

Firstly, datasets should be loaded from Hadoop HDFS cloud. Secondly, a broadcast variable should be created and saved the userId of male viewers at age group between 18 and 24 from users.dat. Secondly, the average rating should be calculated for getting top ten favorite movies of male viewers. Thirdly, the information (movieId, (title, avg)) of all movies can be obtained from movies.dat. Finally, it can get experimental results of the top ten favorite movies of male viewers at age group between 18 and 24. Task execution time of data mining is 0.040943s, as shown in Fig. 15.

```
18/05/07 08:23:35 INFO DAGScheduler: Job 2 finished: collect at BigData2.scala:39, took 0.040943 s
18/05/07 08:23:35 INFO SparkContext: Invoking stop() from shutdown hook
(42 Up (1998),5.0)
(Arguing the World (1996),5.0)
('Night Mother (1986),5.0)
(Black Sunday (La Maschera Del Demonio) (1960),5.0)
(Actor's Revenge, An (Yukinojo Henge) (1963),5.0)
(Young Doctors in Love (1982),5.0)
(I Am Cuba (Soy Cuba/Ya Kuba) (1964),5.0)
(Sanjuro (1962),5.0)
(Nobody Loves Me (Keiner liebt mich) (1994),5.0)
(Modulations (1998),5.0)
18/05/07 08:23:35 INFO SparkUI: Stopped Spark web UI at http://192.168.199.159:4040
18/05/07 08:23:35 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

Fig. 15. Results and task execution time of experiment 3

The top ten favorite movies of male viewers at age group between 18 and 24 are as follows: "42 Up (1998)", "Arguing the World (1996)", "Night Mother (1986)", "Black Sunday (La Maschera Del Demonio) (1960)", "Actor's Revenge, An (Yukinojo Henge) (1963)", "Young Doctors in Love (1982)", "I Am Cuba (Soy Cuba/Ya Kuba) (1964)", "Sanjuro (1962)", "Nobody Loves Me (Keiner liebt mich) (1994)", and "Modulations (1998)".

From the above analysis, we can get an apparent conclusion that the male viewers prefer to view the documentary, drama, and comedy films at age group between 18 and 24.

4.4 Rating Analysis

Experiment 4: What are the top ten movies with the highest rating (5.0) for male and female viewers?

Firstly, datasets should be loaded from Hadoop HDFS cloud. Secondly, average rating of all movies should be calculated from ratings.dat. Thirdly, movieId of top ten rating movies should be filtered from

movies.dat, and then the movieId should be matched title from movies.dat. Finally, it can get experimental results of top ten movies with the highest rating (5.0) for male and female viewers. Task execution time of data mining is 0.054050s, as shown in Fig. 16.

```
(787,Gate of Heavenly Peace, The (1995))
(989,Schlafes Bruder (Brother of Sleep) (1995))
(1830,Follow the Bitch (1998))
(3172,Ulysses (Ulisse) (1954))
(3233,Smashing Time (1967))
(3280,Baby, The (1973))
(3382,Song of Freedom (1936))
(3607,One Little Indian (1973))
(3656,Lured (1947))
(3881,Bittersweet Motel (2000))
18/05/07 08:26:43 INFO Executor: Finished task 0.0 in stage 3.0 (TID 3). 708 bytes result sent to driver
18/05/07 08:26:43 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 3) in 31 ms on localhost (executor driver) (1/1)
18/05/07 08:26:43 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
18/05/07 08:26:43 INFO DAGScheduler: ResultStage 3 (foreach at BigData3.scala:33) finished in 0.047 s
18/05/07 08:26:43 INFO DAGScheduler: Job 1 finished: foreach at BigData3.scala:33, took 0.054050 s
18/05/07 08:26:43 INFO SparkContext: Invoking stop() from shutdown hook
18/05/07 08:26:43 INFO SparkUI: Stopped Spark web UI at http://192.168.199.159:4040
18/05/07 08:26:43 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

Fig. 16. Results and task execution time of experiment 4

The top ten movies with the highest rating (5.0) for male and female viewers are as follows: “Gate of Heavenly Peace, The (1995)”, “Schlafes Bruder (Brother of Sleep) (1995)”, “Follow the Bitch (1998)”, “Ulysses (Ulisse) (1954)”, “Smashing Time (1967)”, “Baby, The (1973)”, “Song of Freedom (1936)”, “One Little Indian (1973)”, “Lured (1947)”, and “Bittersweet Motel (2000)”.

Therefore, whether it is a male or a female viewer, we can recommend them to these top ten films with the highest rating (5.0) when they do not know what they want to view.

4.5 Viewing Frequency Analysis

Experiment 5: What are the top ten movies that female viewers who have seen the most?

Firstly, datasets should be loaded from Hadoop HDFS cloud. Secondly, userId of female viewers should be filtered from users.dat. Besides, userId should be matched movieId from ratings.dat. Thirdly, the top ten movies frequency that female viewers have seen the movies should be calculated, and then, the information (movieId, (title, frequency)) can be obtained from movies.dat. Finally, it can get experimental results of top ten movies that have seen most with female viewers by join operation. Task execution time of data mining is 0.206953s, as shown in Fig. 17.

```
(2396,Shakespeare in Love (1998),798)
(593,Silence of the Lambs, The (1991),706)
(2858,American Beauty (1999),946)
(2762,Sixth Sense, The (1999),664)
(260,Star Wars: Episode IV - A New Hope (1977),647)
(1196,Star Wars: Episode V - The Empire Strikes Back (1980),648)
(608,Fargo (1996),657)
(1265,Groundhog Day (1993),658)
(1210,Star Wars: Episode VI - Return of the Jedi (1983),653)
(356,Forrest Gump (1994),644)
18/05/07 09:49:46 INFO Executor: Finished task 0.0 in stage 6.0 (TID 6). 1009 bytes result sent to driver
18/05/07 09:49:46 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 6) in 79 ms on localhost (executor driver) (1/1)
18/05/07 09:49:46 INFO TaskSchedulerImpl: Removed TaskSet 6.0, whose tasks have all completed, from pool
18/05/07 09:49:46 INFO DAGScheduler: ResultStage 6 (foreach at BigData4.scala:39) finished in 0.079 s
18/05/07 09:49:46 INFO DAGScheduler: Job 2 finished: foreach at BigData4.scala:39, took 0.206953 s
18/05/07 09:49:46 INFO SparkContext: Invoking stop() from shutdown hook
18/05/07 09:49:46 INFO SparkUI: Stopped Spark web UI at http://192.168.199.159:4040
18/05/07 09:49:46 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

Fig. 17. Results and task execution time of experiment 5

From the above experimental results, we can draw a conclusion that the most popular movie for female viewers is “American Beauty (1999)”, and the viewing frequency reaches 946. The genre of this film is Comedy movie. All of the top ten movies viewing frequency have been more than 600, as shown in Fig. 18.

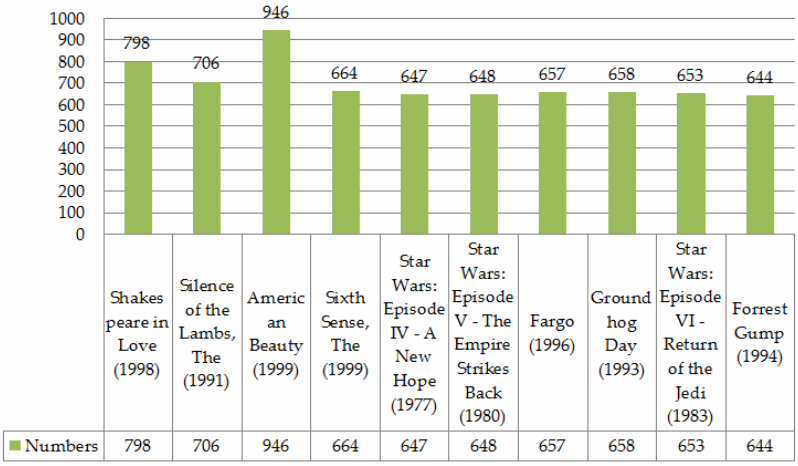


Fig. 18. Analysis of viewing frequency

5 Conclusions and Remarks on Possible Further Work

In brief, this paper adopts the way of storing data by Hadoop cluster and processing data by Spark cluster on Linux platform to solve big data analysis questions. Hadoop cluster provides features that Spark cluster does not have, such as distributed file system, but Spark cluster provides real-time memory processing for those datasets. Compared with traditional single Hadoop cluster deployment, the cluster combination deployment is suitable for big data applications. This combination becomes an extremely powerful solution for big data analysis. Based on the big data case (movie datasets) analysis, it can be verified that deployed Linux platform has stronger reliability, applicability, and generality. Therefore, the combination Hadoop cluster and Spark cluster on Linux platform can be widely used in big data analysis system.

The main contribution of this paper is combination of Hadoop cluster and Spark cluster on Linux platform and Scala in Spark cluster in order to mine data. What’s more, the advantages of using Scala in Spark cluster are as follows: code conciseness, high speed, high efficiency, and results accurate and clarity. According to the differences of cluster deployment mode, complexity of data and sample size, the data processing time-consumed is different. Based on the above five experimental results in movies datasets (size: 1,010,200 pieces of data), compared with traditional cluster deployment, the average time of task execution for data mining by combination way approximately is one second. Therefore, it can be verified that the proposed method can effectively improve the efficiency and save big data processing time.

However, the side effects of combination are as follows: Linux platform’s deployment is relatively complicated, the clusters need to be set more configuration files, and task scheduling and fault tolerance are not perfect. Therefore, we will moderate the side effects in the subsequent research work. On the one hand, we will do more work on the diversity of presentation for showing big data analysis results. For example, it is not only just by a simple program screenshots to show, but also can be graphically presented to users by Java-Web technology. On the other hand, it also can be verified the relationship between occupation and movie viewing genres, or the relationship between region and movie viewing genres in the future experiments.

Acknowledgements

This research was supported by the Key Research and Development Plan of Shandong Province under Grant No. 2015GSF120003, and Weihai Science and Technology Development Plan in 2017. The

authors would like to thank the GroupLens research group in the department of computer science and engineering at the University of Minnesota for providing open source datasets to researchers. The authors would like to thank Dr. Chengyou Wang for his help and valuable suggestions. The authors also thank the anonymous reviewers and the editor for their valuable comments to improve the presentation of the paper.

References

[1] Y. Sakurai, Y. Matsubara, C. Faloutsos, Mining and forecasting of big time-series data, in: Proc. 2015 ACM SIGMOD International Conference on Management of Data, 2015.

[2] Y.-P. Chen, S. Alspaugh, R. Katz, Interactive analytical processing in big data systems: a crossindustry study of MapReduce workloads, Proceedings of the VLDB Endowment 5(12)(2012) 1802-1813.

[3] J. Moreno, M.-A. Serrano, M.-E. Fernández, Main issues in big data security, Future Internet 8(3)(2016) 1-16.

[4] E. Damiani, Toward big data risk analysis, in: Proc. 2015 IEEE International Conference on Big Data, 2015.

[5] W.-Q. Yi, F. Teng, J.-F. Xu, Noval stream data mining framework under the background of big data, Cybernetics and Information Technologies 16(5)(2016) 69-77.

[6] C.-H. Lin, J.-C. Liu, T.-C. Peng, Performance evaluation of cluster algorithms for big data analysis on cloud, in: Proc. 2017 International Conference on Applied System Innovation: Applied System Innovation for Modern Technology, 2017.

[7] D.-N. Disha, B.-J. Sowmya, Chetan, S. Seema, An efficient framework of data mining and its analytics on massive streams of big data repositories, in: Proc. 2016 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics, 2016.

[8] M.-A. Gulzar, M. Interlandi, T. Condie, M. Kim, Debugging big data analytics in Spark with BIGDEBUG, in: Proc. 2017 ACM SIGMOD International Conference on Management of Data, 2017.

[9] W.-Y. Yang, G.-J. Wang, K.-K.-R. Choo, S.-H. Chen, HEPart: A balanced hypergraph partitioning algorithm for big data applications, Future Generation Computer Systems 83(4)(2018) 250-268.

[10] E. Feller, L. Ramakrishnan, C. Morin, Performance and energy efficiency of big data applications in cloud environments: a Hadoop case study, Parallel and Distributed Computing 79(80)(2015) 80-89.

[11] Z.-D. Ji, Applications analysis of big data analysis in the medical industry, Database Theory and Application 8(4)(2015) 107-116.

[12] T. Milo, E. Altshuler, An efficient MapReduce cube algorithm for varied data distributions, in: Proc. 2016 ACM SIGMOD International Conference on Management of Data, 2016.

[13] D.-P. Dong, J. Herbert, Content-aware partial compression for textual big data analysis in Hadoop, IEEE Transactions on Big Data 1(9)(2017) 1-14.

[14] H.-Q. Xu, Z. Li, S.-M. Guo, K.-K. Chen, CloudVista: interactive and economical visual cluster analysis for big data in the cloud, Proceedings of the VLDB Endowment 5(12)(2012) 1886-1889.

[15] Y. Huai, A. Chauhan, A. Gates, G. Hagleitner, E.-N. Hanson, O.-O. Malley, J. Pandey, Y. Yuan, R. Lee, X.-D. Zhang, Major technical advancements in Apache Hive, in: Proc. 2014 ACM SIGMOD International Conference on Management of Data, 2014.

[16] B.-D. Li, Y.-L. Diao, P. Shenoy, Supporting scalable analytics with latency constraint, Proceedings of the VLDB Endowment 8(11)(2015) 1166-1177.

[17] K. Havelund, Data automata in Scala, in: Proc. 2014 International Symposium on Theoretical Aspects of Software Engineering, 2014.

[18] Apache SparkTM Lightning-fast Unified Analytics Engine. <http://spark.apache.org/>.

[19] R.-J. Barnett, B. Irwin, Performance effects of concurrent virtual machine execution in VMware workstation 6, Advanced Techniques in Computing Sciences and Software Engineering 2(3)(2010) 329-333.

[20] F.-M. Harper, J.-A. Konstan, The movieLens datasets: history and context, ACM Transactions on Interactive Intelligent Systems 5(4)(2015) 1-19.