Research of Performance of Distributed Platforms Based on Clustering Algorithm

Di Jian, Yanfeng Peng*
School of control and Computer Engineering, North China Electric Power of University, Baoding, Hebei Province, China, 071000.

* Corresponding author. Tel.: +86 15733227880; email: pengyanfeng2012@126.com
Manuscript submitted July 23, 2015; accepted September 7, 2015.
doi: 10.17706/jcp.11.3.195-200

Abstract: With the deep development and application of Internet technology, data need to be processed more and more, when dealing with large amounts of data. Spark is a versatile high-performance and parallel computing framework, which can be applied to data mining. This paper is based on the parallelization of platforms' K-means algorithm, by building a YARN cluster environment and making experiments to analyze performance of two distributed platforms, and finally find out that the match of Spark and YARN shows more effective on clustering results and consumes less time on the execution of programs, so it's more suitable for cluster analysis of big data.

Key words: Clustering algorithm, distributed platforms, research of performance.

1. Introduction

With the development of informational society, technologies and platforms base on distributed storage and parallel computing of big data become more and more mature, and gradually get promoted and applied, which provides a good technical means and supplies platforms for different industries solving the problems of large data applications.

The algorithms on cluster analysis can be divided into Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Model-Based Methods [1]. These algorithms can achieve a good clustering effect, among these algorithms K-means is most widely used and has the relatively simple idea which belongs to Partitioning Methods. In cluster analysis, K-means is an algorithm which is simple, fast, higher efficient for large data sets, scalable and suitable for mining large data sets, and its time complexity becomes nearly linear. This paper will use the machine learning library MLlib of Spark and the resource manager YARN scheduling to execute tasks in parallel, by building a cluster environment to implement the parallelization of K-means algorithm in order to improve the practicality of data mining and the efficiency and of cluster analysis.

2. Research of Cloud Computing Framework Spark

Spark is a versatile high-performance and parallel computing framework developed by UC Berkeley AMP lab, which is similar to and has the advantage of Hadoop MapReduce. But what is different from MapReduce is that Job intermediate output results can be saved in memory, thus, eliminating the need to frequently read and write HDFS, so Spark can be better suited for data mining and machine learning algorithms that

need iteration [2].

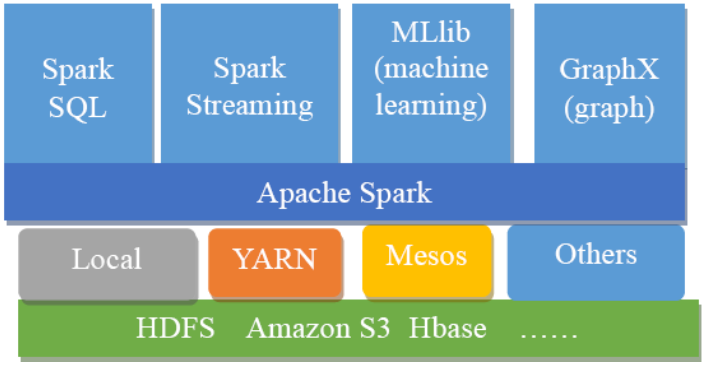

Fig. 1. Graph of ecosystem structure of Spark.

The architecture of Spark is shown in Fig. 1. The main components of the Spark architecture are as follows [3]:

- Spark SQL is equal to the function of Hadoop's Hive. It can convert SQL commands into MapReduce programs of Spark.
- Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Kafka, Flume, Twitter, ZeroMQ, Kinesis or TCP sockets which can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window [4].
- MLlib is Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives.
- GraphX is a new component in Spark for graphs and graph-parallel computation. At a high level, GraphX extends the Spark RDD by introducing a new Graph abstraction [5]: a directed multigraph with properties attached to each vertex and edge.

3. Research of MLlib and K-means

3.1. MLlib

MLlib is the library of Spark which has implemented common machine learning algorithms [6], and includes the relevant tests and data generators. Now, MLlib supports four common machine learning problems: binary classification, regression, clustering and collaborative filtering. MLlib currently supports K-means which is regarded as one of the most widely used clustering algorithms, but needs to set the number of clusters and iterations in advance, and then data can be clustered.

3.2. K-means

Clustering belongs to unsupervised learning. K-means, which is known as one of the top ten mining algorithms and proposed by Macqueen in 1967 [7], is a popular method of cluster analysis in data mining. The basic idea of this algorithm is that data points of data sets randomly are divided into K groups, the mean value of each group is treated as the center point, calculating the distance of each data point to the center point according to the similarity association rules of data points, regrouping the data points, calculating again a new mean of each group as a new center point. After the iterative calculation, the mean value of K groups' center points limits to some value. If this, the iteration will be stopped.

Assuming that the data set is divided into K clusters, the algorithm is described as follows [8]:

- Randomly select k points as the initial center point of each cluster;

- Calculate the distance of each point to the K initial center points, and classify this point into the cluster that the distance of this point to some center point is min;
- Calculate again the center point of each class;
- Calculate the standard measurement function, if the function is limited to some value, the algorithm terminates, otherwise go back to step 2 to continue.

Usually, the adopted objective function is the squared error criterion function:

$$\varphi = \sum_{i=1}^{k} \sum_{x_i \in C_j} \|x_i - c_i\|^2$$

In this formula, xi represents a data object, c_i represents the cluster centroid of c_j, φ represents the sum of squared error of all objects in the datasets [9]. The objective function adopts the Euclidean distance, of course, other distance functions can also be used as a similarity measure method.

4. Results and Analysis of Experiments

4.1. Hardware Configuration

In this experiment, the cluster environment is built on the experimental room, cluster is made of ten servers, virtual machines are installed on a test machine so that experiments are carried out smoothly, each virtual machine's configuration is shown below in Table 1.

Table 1. Hardware Configuration of Clusters

Name	Number	Detailed configuration
Namenode	1	Intel core i5 3210M, 2.5GHz, 8G RAM, 500GHardware
Datanode	9	Intel core i5 3210M, 2.5GHz, 8G RAM, 500GHardware
Network	1	1000Mbps

4.2. Software Configuration

Ubuntu virtual machines are installed in VMware Workstation in this experiment. Spark, Hadoop (including Yarn, HDFS), Mesos are built in each virtual machine, and the underlying file system adopts HDFS. Each machine software configuration is shown in Table 2.

Table 2. Software Configuration of Clusters

Name of Software	Detailed configuration
Operating System	Ubuntu14.04 Virtual Machine
Java	JDK1.7
Hadoop(including YARN and HDFS)	Hadoop2.5.1
Spark	Spark1.1.1
Mesos	Mesos0.20.0

For building the platform and making experiments conveniently, we change the name and IP address of each machine, and configure the connections between each node via ssh without a password. IP of Master we configure is 192.168.44.131, IP of Slave1 to Slave10 is successively 192.168.44.132 ~ 140. The experimental cluster topology is shown in Fig. 2, nodes are connected to each other through a switch:

4.3. Launching Applications and Running Spark on YARN

YARN is a platform of resource management [10], it has been integrated into Hadoop2.5.1 version. This

application is developed in IntelliJ IDEA (or other IDE such as eclipse) which has installed the plugin Scala, and uses Scala language, MLlib, and the machine learning library of Spark. Run the make command in IDEA and the application can be packed a jar package, then submit the application to the YARN cluster with the spark-submit command.

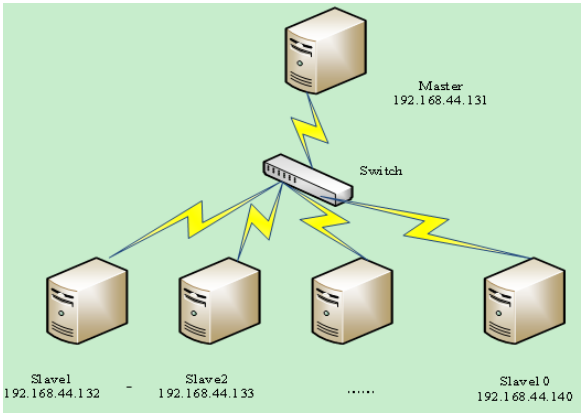

Fig. 2. Graph of cluster topology.

4.4. Experiment and Analysis

Experiment 1: Comparison and analysis of distributed platforms' performance. The following experiment uses the of K-means square error criterion function [11], $\varphi = \sum_{i=1}^{k} \sum_{x_i \in C_j} \|x_i - c_i\|^2$, by submitting an application of Spark and Hadoop to calculate WSSSE (Within Set Sum of Squared Error) [12]. the experimental results are shown in Fig. 3:

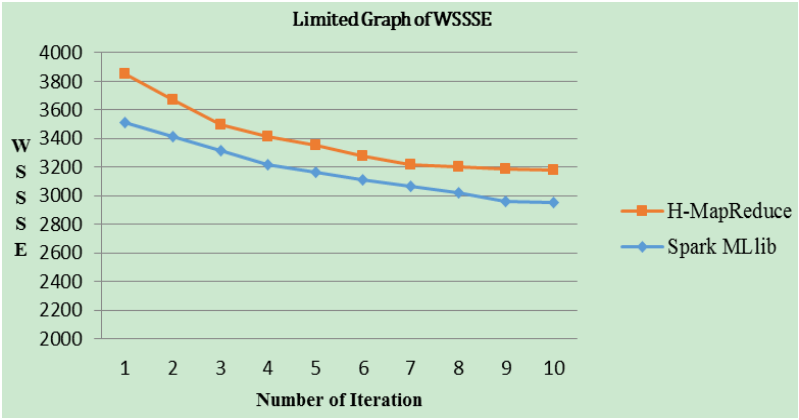

Fig. 3. Limited Graph of WSSSE.

The results from Fig. 3 can see that, with the increasing number of iterations, WSSSE of K-means clustering based on Spark Mllib is smaller than that based on Hadoop MapReduce, it indicats that the clustering results of Spark limits faster and more efficiently.

Experiment 2: Comparison and analysis of time consumption of parallel clusters. This experiment uses the platform of Spark on YARN and Spark on Mesos, by executing the K-means algorithm while keeping the other conditions remain unchanged, separately submitting applications to YARN and Mesos, and controlling the number of data nodes to test time consumption. The results are shown in Fig. 4:

As can be seen from the results of this experiment, when clustering needs more and more computing resources, the capabilities of scheduling resource for YARN is higher than Mesos. With the number of nodes increasing, resources scheduled of YARN can become more, and processing applications consumes less time, which indicates that, the combination of Spark and YARN is more efficient in processing applications, and

superior to the combination of Spark and Mesos.

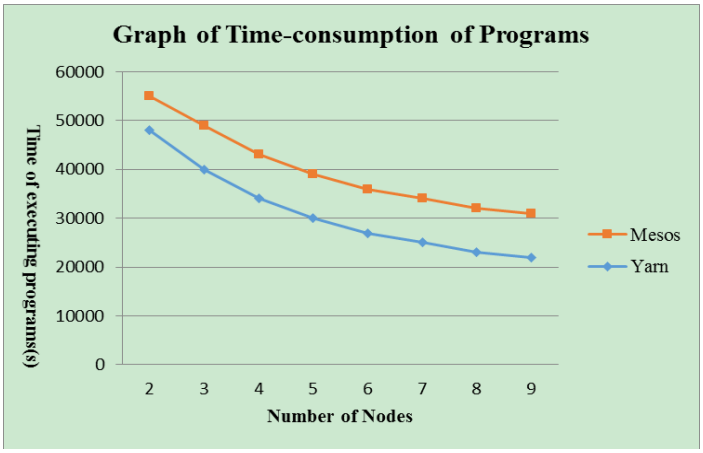

Fig. 4. Graph of time-consumption of programs.

5. Conclusion and Prospect

This article designs and implements K-means algorithm based on MLlib, uses a combination of Spark and YARN to implement the parallelization of the algorithm, and verifies the algorithm by experiments' comparison in set sum of squared errors and time consumption of programs, and by analyzing concludes that the performance based on the combination of Spark and YARN shows better. Data mining and cloud computing are the products of massive data, next we will study programming models of Spark and resource scheduling mode of YARN, learn Scala languages which is integrated features of object-oriented programming and functional programming, develop some applications which are more suitable for cluster analysis of massive data, and study some data processing platforms of higher performance which are well applied to iterative calculation for massive data.

References

[1] American Psychological Association. (2001). *Publication Manual of the American Psychological Association* (5th ed.). Washington, D.C.: Publisher.

[2] Li, A., Chen, C. Z., Long, Q. L., Liang, G. H., & Xiong, D. Y. (2014). Research of big data technology in power supply enterprise base on Spark and Shark. Beijing: China Electric (Technology).

[3] Gao, G. T., Zheng, X. Y., Song, Y. W., Zhou, X. Y., Wu, J. M., Huo, L., & Zhang, J.-L. (2013). *Research of Distributed Rendering System Based on Spark MapReduce Framework.* Beijing: Software Guide.

[4] Xing, L. G., & Lv, Q. S. (2014). Analysis of social networking features based on Spark. *Learned Journal of Pingdingshan University, 5,* 80-83.

[5] Ding, S. Y., Min, S. W., & Fan, Y. B. (2014). NetFlow traffic analysis system based on spark platform. *Telecommunications Science, 10,* 48-51.

[6] Yu, H. H. (2014). Research of plagiarism detection cloud computing framework based on Spark. *Computer CD Software and Applications, 11,* 110-112.

[7] Li, W., Cheng, H.-L., Peng, Y., Wen, M.-J., & Xiao, W.-Q. (2014). Visualized data mining platform based on the Spark. Chinese Association of Automation System Simulation Professional Committee.

[8] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281-297).

[9] Tang, Z. K. (2014). *Design and Implementation of Machine Learning Platform Based on Spark*. Amoy University.

[10] Hu, J. (2012). *The Research and Implementation of the Parallelization of the Clustering Algorithm in Cluster Environment.* East China Normal University.
[11] Hu, S. J. (2013). *Parallel Data Ming Algorithm Research in Cloud.* University of Electronic Science and Technology of China.
[12] Liang, Y. (2014). *Research on Parallelization of Data Mining Algorithm based on Distributed Platforms Spark and YARN*. Sun Yat-sen University.
[13] Qiu, R. C. (2014). *The Parallel Design and Application of the CURE Algorithm Based on Spark Platform*. South China University of Technology.

Di Jian was born in 1968. He is a master's tutor of North China Electric Power University and senior engineer, mainly studying in internet of things, SDN, big data etc. He has engaged in educational work many years, mainly teaches computer network, information security, internet of things, network protocol etc. He has published a number of papers in some journals and conferences and got a round of applause. He is one of the academic leaders of computer network, and many postgraduates are being in his door.

Yanfeng Peng was born in 1989. As a postgraduate of North China Electric Power University, he mainly studies in big data, data mining, cloud computing. He graduated from Hebei Normal University of Science and Technology in 2013 and got a degree of bachelor of science. During undergraduate time, he majors in data structure, software engineering, computer network, database technology etc.