Research on Spark Big Data Recommendation Algorithm under Hadoop Platform

Zubang Huang
East China University of Technology, Nanchang 330000, China
m319.mai@qq.com

Abstract. Hadoop is the main platform for big data mining. It can build Yarn, Mahout, Storm, Graph Lab and other frameworks on its platform. Spark framework is a common task scheduling framework, which not only has powerful processing capabilities, but also has Data processing timeliness, which integrates functions such as machine learning, graph calculation, and online learning, adopts a unified processing mechanism, and the speed is dozens or even hundreds of times that of traditional data processing methods. The paper establishes a Spark framework based on the Hadoop platform, and verifies that the Spark framework's processing mechanism for big data can meet the speed requirements of the increasingly value-added big data applications.

1. Introduction
The Hadoop platform is a distributed architecture developed by the open source organization Apache Foundation and is the foundation platform for big data. The Spark framework is called fast data. It is a memory-based programming model. It cannot put the intermediate iterative process on the disk, and the direct data is not executed in memory, which greatly improves its execution speed. Spark is a new tool for big data mining. The advantages of Spark are not only reflected in the fast, but also in the powerful integration capabilities, its integration capabilities are reflected in its powerful big data capabilities. Spark's framework is divided into four major modules: Spark SQL-RDD (basic unit of data execution), MLlib (machine learning), Graph (graph calculation) and Spark Streaming (real-time processing) [1]; these four parts of data processing the units are all RDDs, so the entire framework forms the consistency of big data processing for various application scenarios. The Spark framework is a simple, powerful and efficient framework that combines machine learning, graph computing and online learning.

2. Hadoop installation and deployment
As a practical standard implementation of the Map Reduce model, Hadoop has been widely adopted by many organizations to store and compute large data sets. It consists of two components: 1) Hadoop Distributed File System (HDFS); 2) Hadoop Map Reduce engine. The Map Reduce model is based on two user-defined functions, map and reduce, which compute the data records represented by key-value pairs. The mapping function extracts the relevant features of each key-value pair, and the approximate subtraction function uses these features to obtain the desired result [2].
Tab. 1 Hadoop, Spark framework configuration

|                  | Hadoop | Spark          |
|------------------|--------|----------------|
| HDFS block size  | 128 MB | 128 MB         |
| Copy factor      | 3      | 3              |
| The heap size of the mapper/about reducer | 3.3GB | Actuator heap size | 18.8GB |
| Per node mapper  | 4      | Number of workers per node | 1 |
| Decremented per node | 4 | Worker core number | 8 |
| Shuffle (scrambling) parallel replication | 20 | |
| IO sort MB       | 600MB  |                |
| IO sort overflow percentage | 80% | |

2.1. **Hardware preparation**

Connect the PC to a LAN via a hub using an idle PC in the lab room. This platform selects 4 PCs first. Due to the applicability of Hadoop and Spark, it is easy to add new nodes to the cluster. Each PC is equipped with a virtual machine [1-2] and is equipped with an Ubuntu 32bit operating system with 4GB of RAM and 20GB of storage. Set the network connection mode of the virtual machine to bridge mode and set a static IP address for the virtual machine to ensure that the virtual machines can ping each other. In this way, the interconnection between machines is achieved. Select one of the PCs as the master (name) and the other three as slaves (data nodes). To facilitate management of the cluster, establish Hadoop users on each PC and give root privileges to Hadoop users. Put the operations on Hadoop and Spark under the Hadoop user [3].

2.2. **Installing JDK**

Hadoop is a distributed computing framework developed in the Java language, so each node in the cluster needs to have a JDK installed. Download the JDK Ubuntu version jdk-8u25-linux-i586.tar.gz from the official website to the desktop, create a new java folder under /usr /local /, and put jdk-8u25-linux-i586.Tar.gz from the desktop Copy it to the java folder and extract the file.

2.3. **Configuring ssh password-free login**

Hadoop clusters need to communicate via the ssh [4] password-free service at runtime. Ubuntu comes with a ssh client and needs to download the ssh server itself. In the case of networking, enter the following command to install the ssh server:

```
$ sudo apt-get install open ssh-server
$ sudo apt-get update
```

When using ssh to log in to another node of the cluster, you need to enter the password of the node. In order to avoid communication, you need to configure ssh password-free login between clusters.

3. **Spark cluster construction**

After uploading Spark-1.6.1-bin-hadoop2.6.tgz and extracting the installation package to Linux, use the command tar -zxvf spark-1.6.1-bin-hadoop2.6.tgz -C /home/zkpk to extract the installation package to specify the location [5].

Use the command to enter the Spark installation directory for configuration. Use the command cd conf/ to enter the conf directory, use the command: mvspark-env.sh. Template spark-env.sh to rename, and then modify the file with the command: vi spark-env.sh Open the configuration file and add the following configuration to the configuration file, where master is the host name of the primary node.

```
Export JAVA_HOME=/usr/java/jdk1.7.0_71/
Export SPARK_MASTER_IP=master
Export SPARK_MASTER_PORT=7077
```
Then save and exit. Next rename and modify the slaves. Template file. Add the location of the child node worker node in the file. The host names of the two slave nodes are slave01 and slave02. Save and exit. Finally, copy the configured Spark to other nodes. The command is as follows:

- `Scp -r spark-1.6.1-bin-hadoop2.6/ zkpk@slave01: ~`
- `Scp -r spark-1.6.1-bin-hadoop2.6/ zkpk@slave02: ~`

Go to the `/home/zkpk/spark-1.6.1-bin-hadoop2.6/sbin` directory and run: `./start-all.sh` to run jps on three nodes, respectively, with the processes master, worker, and worker. View master: 8080 in your browser [6].

![Fig. 1 Overall system architecture](image)

The secure data source includes various network and service data that can be utilized: network packet capture packets, network devices provide Net Flow data, and log files generated by various systems and services, such as DNS logs, Web service logs, and mail logs. The network message is pre-production, such as quintuple information extraction and stream information extraction, and then pushed to the Kafka distributed message system in real time. Kafka has good throughput and load balancing technology to ensure high-speed and orderly delivery of data to message consumers. Status text messages such as logs are also pushed to Kafka after being collected in real time using the Flume distributed logging system. Part of the security data can be directly saved to the distributed file system HDFS after simple processing [7]. A part of the data is distributed to the Spark computing module through Spark streaming for analysis and processing. Spark Streaming breaks down streaming data calculations into a series of short batch jobs that take advantage of Spark’s memory-based computing model to process stream data at high speed. Spark is responsible for the main data analysis and processing functions, and provides the MLlib machine learning library. MLlib can be used to build analysis models for different security issues.
4. Verify Hadoop clusters and Spark clusters

Word Count is an introductory program for big data programming. It implements statistics on the number of occurrences of each word in the input file. It can be applied to word frequency retrieval of massive text. The Word Count algorithm for realizing word frequency statistics [7-8] is a typical grouping aggregation algorithm. The algorithm first classifies the text into the intermediate value of the key-value pair, the key is the word itself, and the value is 1. Then the data is aggregated, the values with the same key are summed, and the result is output. This paper uses Word Count to verify whether the Hadoop cluster and the Spark cluster are successful [8]. The data set of the algorithm is selected from the network novel. Spark needs to initialize the data on the HDFS to Spark RDD, then generate the new RDD for the string segmentation in the RDD, and convert the newly generated RDD. Parallel computing for Key Value RDD, stipulate the newly generated Key Value RDD, and then get the result. The specific code implemented in Eclipse using Scalar language is as follows:

```scala
Object Word Count {
  def main(args: Array[String]) {
    val conf = new SparkConf()
    setAppName("Word Count");
    val sc = new SparkContext(conf)
    val lines: RDD[String] = sc.textFile(args(0));
    val words: RDD[String] = lines.flatMap(_.split(" "));
  }
}
```
5. Conclusion

In the context of large-scale network environment, this paper proposes a low-cost calculable big data security analysis and detection platform based on Hadoop and Spark computing framework for the existing single-machine security analysis and detection platform processing capability limitation. Using offline model generation and online detection to analyze large-scale network data, and real-time security analysis and detection. Compared with the traditional single server analysis and test data, it is proved by experiments that the big data security analysis platform based on Hadoop and Spark has good applicability and efficient processing ability, which can meet the requirements of analysis and detection of secure big data. In the next stage, the DNS log and network traffic data will be fully utilized on the basis of the big data security analysis platform to perform real-time botnet detection and analysis in a heterogeneous data environment.
References

[1] Liu Ping. Research on University Campus Big Data Platform Based on Hadoop and Spark. Software Engineering, Vol.5 (2018) No.12, p. 140-145.

[2] Chen Hongjun, Wu Xueqin. Analysis and Application of Spark Fast Data Recommendation Algorithm Based on Hadoop Platform. modern electronic technology, Vol. 10 (2016) No.39, p. 18-20.

[3] Hu Jun, Hu Xiande, Cheng Jiaxing. A Big Data Hybrid Computing Model Based on Spark. Journal of Computer Systems, Vol. 4 (2015) No. 24, p. 214-218.

[4] Cheng Min. calculable Big Data Analysis Platform Based on PostgreSQL and Spark [D]. Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Vol.1 (2016.) No.19, p. 167-175.

[5] Zhao Kejun, Ge Liansheng, Liu Yang, et al. Building a calculable network security analysis platform based on Hadoop and Spark. Journal of Huazhong University of Science and Technology (Natural Science), Vol. 11 (2016) No.31, p. 25-28.

[6] Yuan Wenliang. Application of Hadoop Spark in Optimizing IPTV Maintenance Support System. China New Communications, Vol. 16(2017) No.16, p. 157-163.

[7] Huang Zhen, Qian Yurong, Fan Yingying, et al. Acceleration strategy for remote sensing big data feature extraction under Spark. Computer Engineering and Design, Vol. 12 (2017) No.38, p. 3279-3283.

[8] Zhai Fuyin, Zhang Lin, He Kunpeng. Fast Query and Implementation of Massive Data Based on Hadoop Platform. Computer Knowledge and Technology, Vol. 12(2016) No.21, p. 3-5.