Design and Realization of Hadoop Distributed Platform

Jing Wang[a], Haibo Luo[b],Yongtang zhang[c]

Neusoft Institute, Guangdong, China

[a]wj@neusoft.com, [b]luo-hb@neusoft.com, [c]zhangyongtang@neusoft.com

Keywords: Internet Applications; Hadoop; Big Data; Cloud Computing; Virtual Machine

Abstract. Hadoop provides a reliable shared storage and analysis system to meet the needs of the mass data of Internet users. We have designed the architecture of Hadoop distributed development platform. And under the environment of Ubuntu, the Hadoop distributed development platform is realized. The aim is to provide a parallel cloud computing system to a user in a template based and reusable way. Through the experimental test, the platform has the reliability and practicability, which provides the basis for the development of large data or cloud computing application services.

Hadoop Introduction

The rapid development of Internet has led to huge amounts of data distributed storage and computing needs. In 2004, Google published two papers "*The Google File System*" and "*MapReduce: Simplied Data Processing on Large Clusters*", which is to extend and improve its own search System. In 2005, Doug Cutting [1] took examples by technologies of Google two papers, and build a layer on Nutch to control the distributed processing, redundancy, automatic failure recovery and load balancing problem, that is the Hadoop [2]. At present, Hadoop can be regarded as the defacto standard in the field of big data in industry, while it is mainly represented by Yahoo, Facebook, Adobe, EBay, IBM, Last.Fm, and LinkedIn at aboard, and took Baidu, Tencent, Alibaba, Huawei, China Mobile, and Pangu search as primary at home.

Hadoop is an open-sourcing efficient platform for cloud computing infrastructure, it is not only widely used in the field of cloud computing, and it can also support the search engine services, as the underlying infrastructure system of search engine. At the same time it is more and more favored in huge amounts of data processing, data mining, machine learning, scientific computing, and other fields. In essence, the rapid development of the Internet has led to huge amounts of data distributed storage and computing needs, and Hadoop just provides a very good solution for these needs [2].

Hadoop distributed platform architecture

Hadoop work pattern and overall architecture

Hadoop uses PC cluster for storage and computing, and it uses the Master/Slave architecture, running NameNode, JobTracker on the Master, deploying DataNode and TaskTracker on the Slave, using DataNode process in NameNode process guidance cluster to manage data storage of its own nodes. JobTracker processes command and running management and TaskTracker processes on the cluster to realize the operation control and scheduling, therefore it objectively form the working mode of command and management on Master, and implementation of computing and storage on Slave.

Hadoop's overall architecture is shown in Fig.1. Computer cluster generally run on Linux. Hadoop is based on Java, relying on Java virtual machine. The programming framework of HDFS and MapReduce are two cores of Hadoop. HDFS is responsible for data storage on cluster, which is the data reading and writing foundation of NameNode and DataNode. MapReduce programming framework is a programming model, which provides runnable and manageable programming for JobTracker and TaskTracker. SecondaryNameNode is a snapshot of NameNode, which is generally not at the same computer with Master. Hadoop can view running state, operation progress and cluster information through browser.

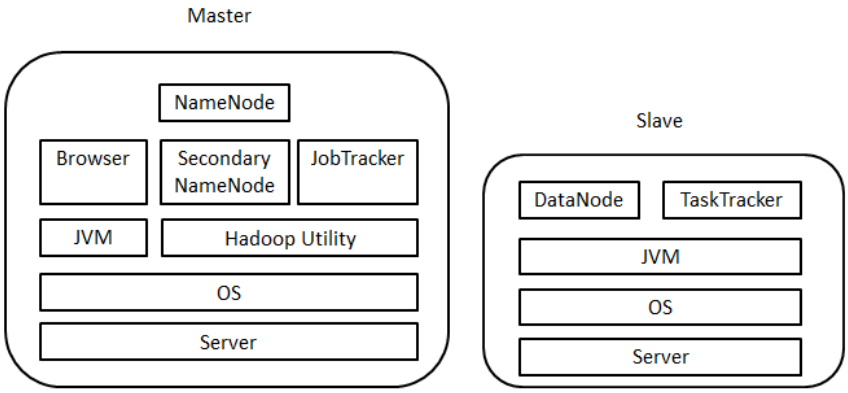

Fig. 1 Hadoop architecture [3]

HDFS architecture

When the size of the data set exceeds the storage capacity of a separate physical computer, it is necessary to conduct partition for it and stored it on several separate computers. The file system of across multiple computer storage in management network is called distributed file system (DFS). HDFS of Hadoop can store large files by streaming data access patterns, and run on the cluster.

HDFS architecture is as shown in Fig.2. NameNode is responsible for survival confirmation, stop control, file read/write control, data block copy control as well as data block address location of DataNode. DataNode is responsible for specific data storage, copy each data block to several copies, transfer to different DataNode for storage through network. DataNode accepts the Client and NameNode scheduling, according to the need to store and retrieve the target data blocks, and regularly send their storage blocks list to NameNode. DataNode receives read/write request of Client program, and at the same time, according to the instructions of NameNode to complete the establishment, delete, and copy of file blocks. Client is responsible for slicing file, accessing to HDFS, interacting with NameNode to get file location information, and interacting with DataNode to read and write data.

HDFS provides multiple modes so as to applications can access, such as Java API, as well as Java API of C language packaging. In addition it provides graphic interface tools under Windows, such as Hadoop4Win (UNIX environment via Cygwin simulation) and Hadoop-eclipse-plug-in.

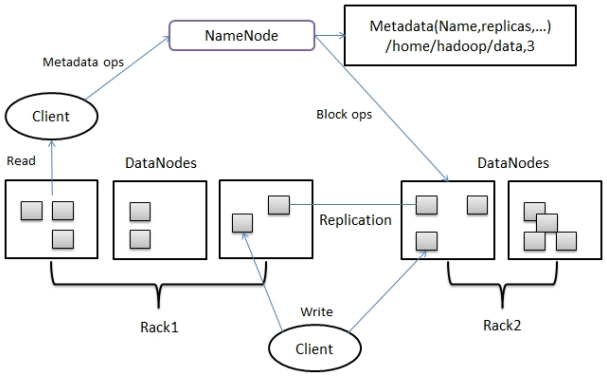

Fig.2 HDFS architecture [3]

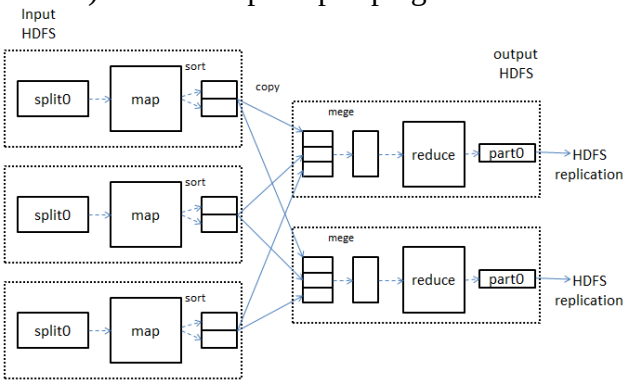

Fig.3 MapReduce task of data flow

MapReduce framework

MapReduce is a kind of distributed computing framework, which is suitable for analysis and calculation of huge amounts of data. Map function is used for operating data set to produce intermediate results in form of key-value pairs. Reduce function reduce key-value pairs to get the final results. It is suitable for parallel data processing in the distributed environment.

In MapReduce model, data flow is processed as shown in Fig.3. The ordered Map outputs needs to be sent to running Reduce task node through the network. Data are merged in Reduce terminal, and then processed by Reduce function defined by the user. Reduce outputs are usually stored in HDFS to achieve reliable storage. The dashed frame represents node, the dashed arrow represents data transmission within node, and solid arrow indicates data transmission between different nodes.

Hadoop distributed platform construction

Hadoop has three kinds of installation modes: stand-alone mode, pseudo-distributed mode and fully distributed mode. Stand-alone mode is mainly used for development and debugging application logic on MapReduce program. Pseudo-distributed mode adds code debugging function on stand-alone mode to run HDFS, and can interact with other daemons. Fully distributed mode supports cluster.

This paper implements the fully distributed platform construction. The platform construction is realized on personal computers and servers respectively. Personal computer configuration: processor, Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz; Memory, 8GB; Hard disk 1T. Software: Windows8.1, VMware Workstation12.0, Hadoop-2.6.0, Ubuntu14.04 (64 bits), OpenJDK1.7 (official suggested JDK of Sun). The server's configuration omits.

The fully distributed node planning is as shown in Fig. 4.

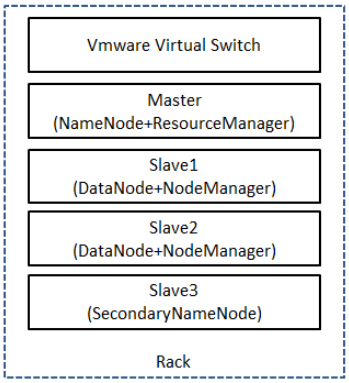

Fig.4 Node planning

Table 1 Basic configuration

```
#Add user
#useradd -m hadoop -s /bin/bash
#Set network card as fixed address
#vi /etc/network/interfaces
#Set DNS, networking is ensured (not necessary)
#vi /etc/resolvconf/resolv.conf.d/base
#Set mapping relation of host name and IP address
$sudo vi /etc/hosts
```

Basic configuration

Add Hadoop users on all nodes, setting fixed IP address, according to the Fig.4 to add host name, set the mapping relation of host name and IP address, the key operations are as shown in Table 1.

SSH configuration

Each node needs to work together in Hadoop running process, data secure communication between nodes realizes password-less authentication through SSH based on RSA algorithm [4]. Key operations are as shown in Table 2.

Java configuration

Hadoop is developed based on Java, running needs the support of Java, so install JDK on each node. Key operations are as shown in Table 3.

Table 2 SSH configuration

```
#Install ssh service, default by SSH client installed
$sudo apt-get install openssh-server
#Use RSA algorithm to generate public key of
#asymmetric algorithm, and the add public key in
#authorization
$cd  ~/.ssh/
$ssh-keygen -t rsa
$cat ./id_rsa.pub  >> ./authorized_keys
#Copy to each node (Master copies public key
#authorization Slave3 as reference)
$ scp /home/hadoop/.ssh/authorized_keys
hadoop@Slave3:/home/hadoop/.ssh/
```

Table 3 Install JDK

```
#Install jre and jdk
$sudo apt-get install openjdk-7-jre openjdk-7-jdk
#Get installation path
$dpkg -L openjdk-7-jdk | grep '/bin/javac'
#Add environment variable
$vim ~/.bashrc
#Let the environment variable takes effect
$source ~/.bashrc
#Check java installation is correct or not
$java –version
```

Hadoop configuration[4]

Install Hadoop on each node, key operations are as shown in Table 4.

Table 4 Hadoop configurations

```
$cd ~/Downloads/Soft                                  #Switch to the installation file directory
$sudo chown hadoop.hadoop hadoop-2.6.0.tar.gz         #Change the file subordinated and subordinated group
$sudo tar hadoop-2.6.0.tar.gz -C /usr/local           #Extract to the user specified directory
$cd /usr/local                                        #Switch to the specified directory
$sudo mv  ./hadoop-2.6.0 ./hadoop                     #Change directory
$sudo chown -R hadoop ./hadoop                        #Change hadoop directory subordinated and subordinated group
$cd /usr/local/hadoop                                 #Switch to the specified directory
$./bin/hadoop version                                 # Check hadoop2.6.0 version
#vi ~/.bashrc                                         #Add hadoop to environment variable
#source ~/.bashrc                                     #Let configurations take effect
$cd /usr/local/hadoop/etc/hadoop                      #Switch to configuration directory
```

#Edit core-site.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://Master/</value>
        </property>
</configuration>
```

#Edit hdfs-site.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>2</value>
        </property>
        <property>
                <name>dfs.namenode.secondary.http-address</name>
                <value>Slave3:50090</value>
        </property>
</configuration>
```

#Edit yarn-site.xml file

```
<?xml version="1.0"?>
<configuration>
   <property>
      <name>yarn.resourcemanager.hostname</name>
      <value>localhost</value>
   </property>
   <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
   </property>
</configuration>
```

#Copy template

```
$cp mapred-site.xml.template  mapred-site.xml
```

#Edit mapred-site.xml files

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

#Edit Slaves files

```
Slave1
Slave2
```

Through the above configurations, Hadoop platform has been established. You can configure Master node first, and then, in the form of complete clone to produce Slave1, Slave2 and Slave3 node.

Test

After completing the SSH password-less test, and mutual Ping test between the nodes, then test Hadoop by WordCount example, and the source code can be found in src/examples of Hadoop installation package. The processes is shown in Fig. 5. Test result is shown in Fig.6.

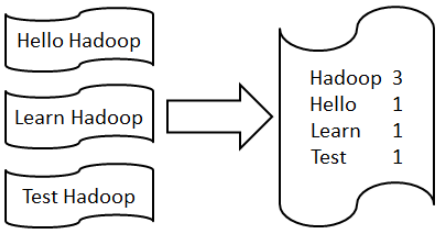

Fig.5 WordCount example

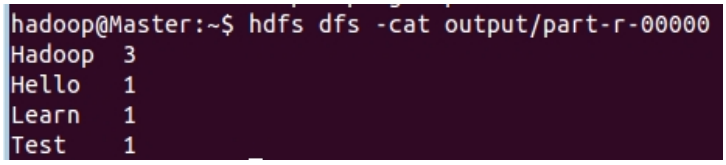

Fig.6 Test result of WordCount example

To login in as hadoop account in master node, test commands are executed in sequence, as shown in Table 5.

To view results access http://master:50070 and http://master:8088/cluster through the brower，as shown in Fig. 7 and Fig.8.

Table 5 Test of WordCount Example

$ hdfs namenode –format	*#Initialize data*
$ start-all.sh	*#Start hadoop*
$ hdfs dfsadmin –report	*#Start working record demons*
$ mkdir ~/Test	*#Create Test directory*
$ cd ~/Test	*#Switch to the specified directory*
$ echo "Hello Hadoop" >> file1.txt	*#Create a text fil named file1, the content is Hello Hadoop*
$ echo "Learn Hadoop" >> file2.txt	*#Create a text file named file2, the content is Learn Hadoop*
$ echo "Test Hadoop" >> file3.txt	*#Create a text file named file2, the content is Test Hadoop*
$ hdfs dfs -mkdir -p /user/hadoop	*#Create user directory on HDFS*
$ hdfs dfs -mkdir input	*#Create input directory on HDFS*
$ hdfs dfs -put ~/Test/.txt input*	*#Put test file into input directory*
$ export HADOOP_ROOT_LOGGER=DEBUG,console	
$hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount input output	*#Execute counting*
$ hdfs dfs -ls output	*#View output file*
$ hdfs dfs -cat output/part-r-00000	*#View results*

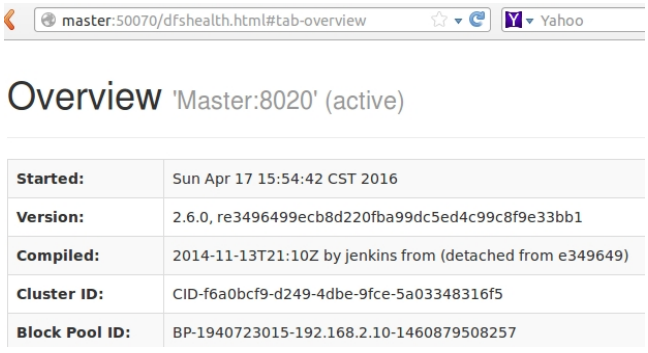

Fig.7 Hadoop running information

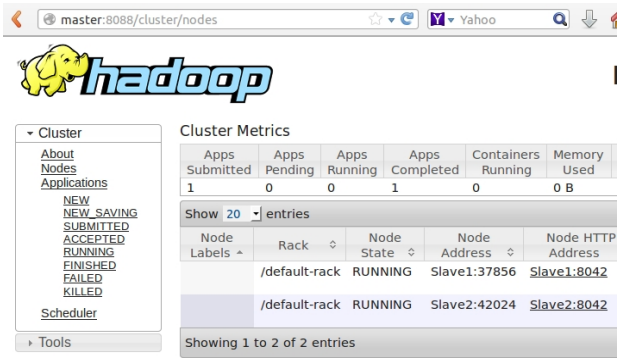

Fig.8 Hadoop cluster information

Conclusion and expectation

Hadoop provides a reliable shared storage and analysis system for users. HDFS realizes data storage, and MapReduce realizes data analysis and processing. This paper briefly introduces the development history of Hadoop, and analyzes the Hadoop distributed platform architecture, and realizes the Hadoop distributed platform based on Ubuntu environment. The platform provides the basis for the development of big data or cloud computing application service. There are many aspects can be extended in Hadoop, for example, data format, data mining, data processing, data storage, coordinating, monitoring and security.

Acknowledgements

This work was financially supported by 2016 Opening Project of Guangdong Province Key Laboratory of Big Data Analysis and Processing at the Sun Yat-sen University and 2014 Youth Innovative Talents Project (Natural Science) of Education Department of Guangdong Province(2014KQNCX248，2014KQNCX249)

References

[1] Tom White. *Hadoop: The Definitive Guide, 4th Edition*, O'Reilly Media March (2015)，p.23-26.

[2] Zhouwei, Zhai. *Hadoop core technology*. China Machine Press (2015), p. 6-16.

[3] Tiwei, Xiao. *Hadoop distributed computing platform architectural analysis and application development*. Southwest petroleum university degree thesis (2014), p.6-19.

[4] Information on http://www.powerxing.com/install-hadoop-cluster/