International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 5 Issue 6 ǁ June. 2017 ǁ PP. 42-47

Design and Research of Hadoop Distributed Cluster Based on Raspberry

Liang Chen[1], Liang Sun[2]
(College of Automotive Engineering, Shanghai University of Engineering Science, China)

ABSTRACT : Based on the cost saving, this Hadoop distributed cluster based on raspberry is designed for the storage and processing of massive data. This paper expounds the two core technologies in the Hadoop software framework - HDFS distributed file system architecture and MapReduce distributed processing mechanism. The construction method of the cluster is described in detail, and the Hadoop distributed cluster platform is successfully constructed based on the two raspberry factions. The technical knowledge about Hadoop is well understood in theory and practice.

Keywords: Distributed cluster, Hadoop, HDFS, MapReduce, Raspberry

I. INTRODUCTION

In the era of big data, how to store and manage these massive data becomes a urgent problem to be solved. Hadoop Distributed Clusters are a specific cluster designed to store and analyze massive amounts of unstructured data. It is essentially a computing cluster, that is, distribute the different data and process the data . In the Big data processing, Hadoop has been able to use a very wide range, mainly because of its data extraction, data distortion and loading and other advantages are very obvious. [1]In this paper, do research into the main component of Hadoop distributed file system HDFS and programming model MapReduce, do the practice of deploying a raspberry-based cloud computing storage model, laying a good foundation for the follow-up study and research Big data.

II. INTRODUCTION TO HADOOP

Hadoop is an open source project under the Apache organization, is a reference to the Google cloud computing 3 core technology GFS, Mapreduce and Bigtable design ideas developed a distributed cloud computing platform. The Hadoop distributed data processing architecture consists of many elements, including HDFS, MapReduce, Pig, Hive, HBase and so on. HDFS can support millions of large distributed file systems; MapReduce is used for parallel computing of very large data sets; Pig can perform a series of processes such as loading data, converting data formats and storing final results; Hive plays data warehouses in Hadoop Role; HBase is used in Hadoop to support large sparse table column storage data environment. The core of the Hadoop framework is HDFS distributed data storage and MapReduce data parallel processing mechanism.[2-6]

2.1 HDFS

HDFS is at the bottom of the Hadoop software framework. It mainly stores files on all storage nodes in the cluster, with characteristics of high transmission and high fault tolerance. It supports data in the file system in the form of streams, which can manage massive data. Hadoop cluster mainly consist of the manager (NameNode) and the workers (DataNode) two types of nodes, and respectively NameNode and DataNode mode. NameNode is primarily responsible for managing file system namespaces and controlling access to external clients. DataNode responds commands from HDFS clients to read and write requests and NameNode to create, delete, and copy blocks.

In a Hadoop cluster, the NameNode is unique, but there are a large number of DataNode nodes that return the DataNode node information to the client when the client sends a request to the NameNode node. The actual I / O transaction does not go through the NameNode, and the NameNode simply maps the operation of the file to the DataNode. HDFS files are divided into a lot of blocks, each file block will be copied multiple copies, respectively, into a different DataNode node. The most common strategy is to use three copy blocks: two data blocks are stored in different nodes of the same rack, and one copy block is stored in a node of another rack . The DataNode node sends heartbeat information to the NameNode node at regular intervals, and the NameNode determines whether the file block in the DataNode node is normal based on this information. If there is an exception, NameNode will be repaired to repair, to ensure the security of the data.

2.2 Mapreduce

MapReduce's distributed processing mechanism is another core technology of Hadoop. It is located on top of HDFS and consists of JobTracker and TaskTracker.

JobTracker is a MapReduce application that is launched on a single main system, which is unique in the Hadoop cluster and is primarily responsible for controlling the objects of the MapReduce application. After the application is submitted, it provides the input and output directories contained in HDFS. TaskTracker can have several, it is mainly the implementation of JobTracker various commands, and the status of the local node to the JobTracker.[7-8]

The MapReduce client program starts a job and fragments it, while sending a job request to the JobTracker. The client program copies the resources required to run the job to HDFS. After receiving the job, JobTracker places it in a job queue and waits for the job scheduler to schedule it. TaskTracker will periodically send a message to JobTracker. If TaskTraeker is ready to run a new task, JobTracker creates a Map task for each slice based on the input slice information and assigns the task to TaskTracker. For a Map task, JobTracker chooses a TaskTracker that takes the nearest input file to execute, thus saving network data transfer time; for a Reduce task, JobTracker simply selects one from the Reduce task list to execute. Hadoop through this mobile computing program rather than mobile data, to avoid the cluster and the system between the large number of frequent data movement, improve the processing speed of the machine.

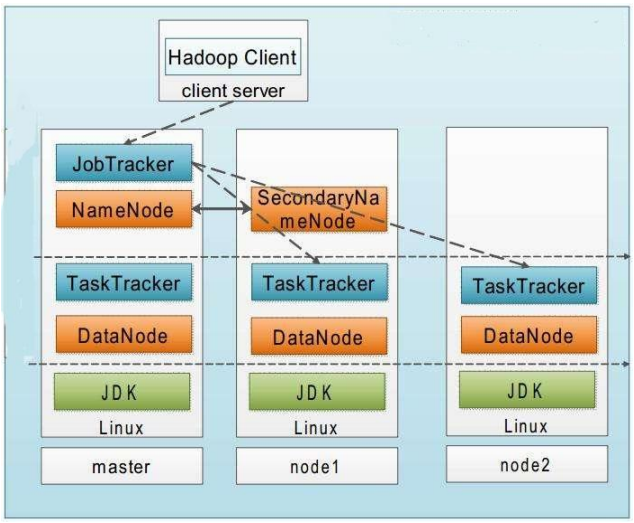

Fig. 1 Hadoop distributed cluster topology

III. ENVIRONMENT AND NODE PLANNING OF HADOOP DISTRIBUTED CLUSTERS

3.1 System environment requirements

The main system environment for building a distributed cluster with three raspberries is shown in Table 1.

3.2 Server node planning

Deploying a Hadoop distributed cluster environment requires at least three servers and is the minimum requirement to run a minimal Hadoop cluster. The specific server nodes are planned in cluster deployment as shown in Table 2.

Table.1 System Requirements for Distributed Clusters

Serial number	System or software	version
1	operating system	Ubuntu 16.01 LTS
2	Hadoop	1.0.4
3	JAVA	JDK1.8.0

Table.2 Server node planning for distributed clusters

Serial number	Server host name	Server IP address	RAM	hard disk	Function
1	master	192.168.8.163	1G	8G	NameNode JobTracker
2	slaver1	192.168.8.164	1G	8G	DataNode TaskTracker
3	slaver2	192.168.8.165	1G	8G	DataNode TaskTracker

IV. HADOOP INSTALLATION AND CONFIGURATION

4.1 Modify hostname and hosts

According to Table 2 to modify the host name of the three hosts, modify the system need to restart, edit / etc / hostname,modify the host name were master, slaver1, slaver2. Modify /etc/hosts, all hosts add the following:

192．168．8．163 master

192．168．8．164 slaver1

192．168．8．165 slaver2

4.2 SSH settings

In the Hadoop cluster, the communication between different machines is realized through SSH. The purpose of SSH configuration is to ensure the smooth flow of the network. When the machine requests communication instruction, it is no longer necessary to enter the password to establish the SSH trusted certificate.

4.2.1 Generate a password pair on the master node

[hadoop@master local# ssh-keygen -t rsa

[hadoop@master loca1]#cat hadoop／.ssh／id_rsa.pub >> /hadoop/.ssh／authorized_keys

4.2.2 Copy the public key to slaver1 and slaver2 /root/.ssh directory and install

[hadoop@slaver1.ssh]# mkdir /hadoop/.ssh／

[hadoop@slaver2.ssh]# mkdir /hadoop/.ssh／

[hadoop@master.ssh]# scp /hadoop/.ssh／id_rsa.pub hadoop@slaver1: /hadoop/.ssh

[hadoop@master.ssh]# scp /hadoop/.ssh／id_rsa.pub hadoop@slaver2: /hadoop/.ssh

[hadoop@slaver1.ssh]# cat id_rsa.pub >> authorized_keys

[hadoop@slaver2.ssh]# cat id_rsa.pub >> authorized_keys

4.2.3 Verify that there is no password for SSH login

hadoop@master:~$ ssh slaver1

Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.1.19-v7+ armv7l) Documentation: https://help.ubuntu.com/

Last login: Sat Jun 3 14:07:26 2017 from 192.168.8.163

hadoop@slaver1:~$

4.3 Install and configure Hadoop

Unzip Hadoop-1.0.4.tar.gz into the / usr / local / hadoop directory.

4.3.1 Configure the master and slaver core-site.xm1 files

The core-site.xml configuration file is located in the conf subdirectory of the hadoop run directory: /usr/local/hadoop/hadoop-l.0.4/conf.Modify the master's core-site.xml file. Change 1ocalhost to master. The main contents are as follows:

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</propeJ 此 y>
<properly>
<name>hadoop.tmp.dir</name>
<value>/usr/local/hadoop/hadoop-1.0.4.1/tmp<lvalue>
</property>
<configuration>

At the same time, respectively, open two slaver1 and slaver2 node core-site.xml, localhost domain name changed to "master".

4.3.2 Configure the mapred-site.xml file for master and slaver

Configure the mapred-site.xml file for master and slaver. Change localhost to "master". Some of the contents are as follows:

<configuration>
<property>
<name>mapred.joh.tracker</name>
<value> master:9001<lvalue>
</property>
<configuration>

4.3.3 Configure masters and slavers files

The masters and slaves files are also located in the conf subdirectory of the hadoop directory. First use the vi command to modify the master host masters file, the text of localhost can be modified to master. And then repair the master host slaves file, modify the text content to:

master
slaver1
slaver2

Then, the master configuration of the masters and slaves files were copied to the slaver1 and slaver2 Hadoop installation directory under the conf folder. Operation is as follows:

[hadoop@master conf]$ scp masters hadoop@slaver1:/usr/local/hadoop/hadoop-1.0.4.1/conf/masters
[hadoop@master conf]$ scp masters hadoop@slaver2:/usr/local/hadoop/hadoop-1.0.4.1/conf/masters
[hadoop@master conf]$ scp slavers hadoop@slaver1:/usr/local/hadoop/hadoop-1.0.4.1/conf/ slavers
[hadoop@master conf]$ scp slavers hadoop@slaver2:/usr/local/hadoop/hadoop-1.0.4.1/conf/ slavers

Finally, go to the two slaver nodes and check that the contents of the masters and slaves files are correct.

At this point, Hadoop's distributed cluster environment is configured successfully.

V. DISTRIBUTED CLUSTER STARTUP AND VIEWING

5.1 Start Hadoop

First format the cluster's file system with the master node: [hadoop @ master] $ hadoop namenode -format

After formatting is complete, start the Hadoop cluster: [hadoop @ master] $ start-all.sh

Use the jps command to see if each node's process started successfully. In the master node run: JobTracker, TaskTracker, Jps, DataNode, SecondaryNameNode, NameNode and other processes. In the slave node running: TaskTracker, DataNode, Jps and other processes. The above information indicates that the various services of each node in the distributed cluster are started normally. You can view the running status of a Hadoop distributed cluster from a Web page.

5.2 View the running status

5.2.1 View the cluster status on the Namenode node

hadoop@master:~$ hadoop dfsadmin -report
Configured Capacity: 7725277184 (7.19 GB)
Present Capacity: 3303890944 (3.08 GB)
DFS Remaining: 3303849984 (3.08 GB)
DFS Used: 40960 (40 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

Datanodes available: 1 (1 total, 0 dead)
Name: 192.168.8.164:50010
Decommission Status : Normal
Configured Capacity: 7725277184 (7.19 GB)
DFS Used: 40960 (40 KB)
Non DFS Used: 4421386240 (4.12 GB)
DFS Remaining: 3303849984(3.08 GB)
DFS Used%: 0%
DFS Remaining%: 42.77%
Last contact: Sat Jun 03 15:53:39 CST 2017

5.2.2 View the cluster status through the browser.

Enter the address: http: //192.168.8.163: 50070, see the distributed cluster status.
Enter the address: http: //192.168.0.163: 50030, see the distributed cluster server node information.

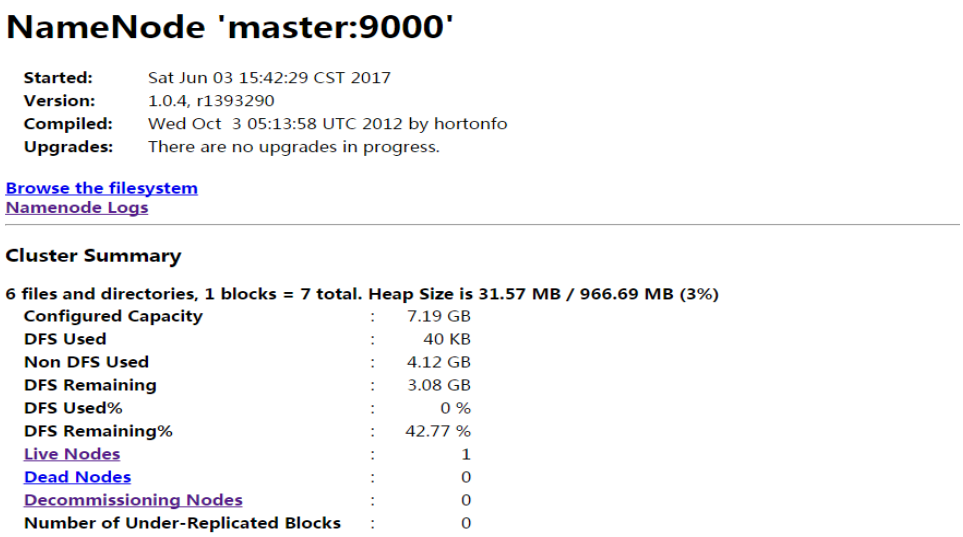

Fig. 2 Monitor Hadoop Distributed Cluster Status

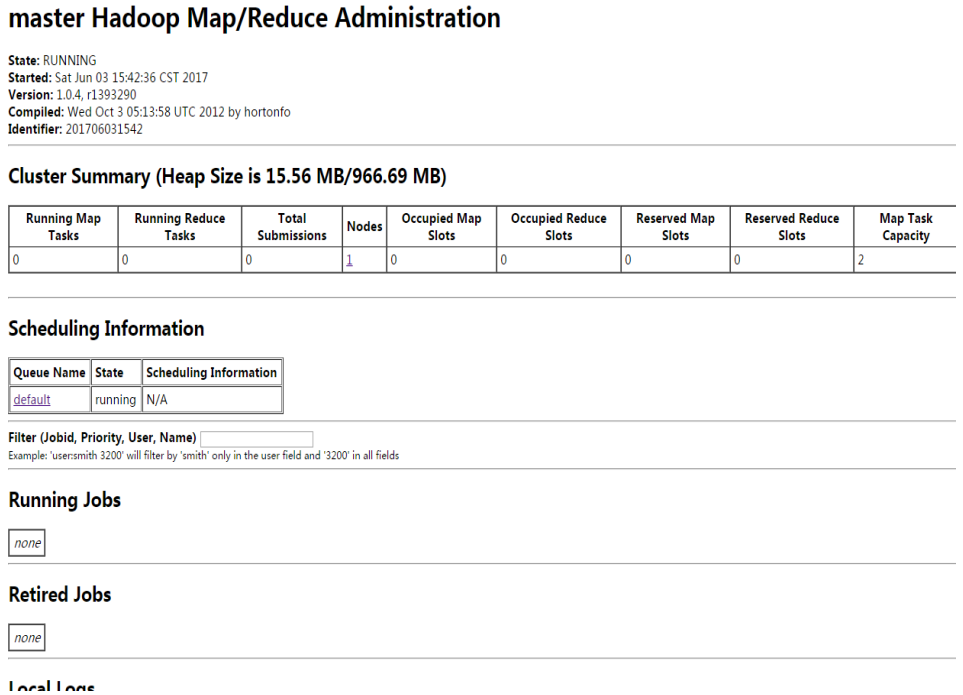

Fig. 3 Monitor server node information for Hadoop distributed clusters

VI. CONCLUSION

Hadoop is a new technology, and more and more scholars join the technology of learning and research. This paper describes the Hadoop two key technologies HDFS distributed file system architecture and MapReduce distributed processing mechanism, as well as Hadoop distributed cluster deployment of the program design and implementation process. The Hadoop distributed cluster based on raspberry deployment is an important development platform for dealing with large data information and making large data mining analysis. At the same time, based on the Hadoop distributed cluster environment, a more efficient Spark distributed computing system based on Hadoop cluster MapReduce computing framework can be constructed to provide the foundation and guarantee for large data analysis solutions.

ACKNOWLEDGEMENTS

This project is sponsored by Shanghai University of Engineering Science Innovation Fund for Graduate Students (No.E3-0903-17-01267).

REFERENCES

[1]. Rdadmlasmdlas Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System[C]// IEEE, Symposium on MASS Storage Systems and Technologies. IEEE Computer Society, 2010:1-10.
[2]. Zhai Yanlong, Luo Zhuang, Yang Kai, et al. Hadoop - based high - performance mass data processing platform [J]. Computer Science, 2013, 40 (3): 100.
[3]. Ma Lin Shan, Zhao Qingfeng, Xiao Xinguo. Hadoop based cloud mobile information service model research [J]. Information Science, 2013, 31 (4): 28.
[4]. Bass L，Kazlnan R，Ozkaya I．Open Source Systems：Grounding Research[M]．Berlin：Springer，2011：50—61
[5]. Lam C. Hadoop actual combat [M]. Han Ji in the translation. Beijing: People's Posts and Telecommunications Press, 2011.
[6]. Kang Li Yun, Wang Xiao Yue, Bai Rujiang. MapReduce principle and its main realization platform analysis [J]. Modern Library and Information Technology, 2012 (2): 60.
[7]. Dean J，Chemawat S．MapReduce：simplified data processing on large clusters [J]．Communication of the ACM，2008，51 (1)：107—113．
[8]. Li Yulin, Dong Jing. Research and improvement of MapReduce model based on Hadoop [J]. Computer Engineering and Design, 2012, 33 (8): 3110-3116.