BRICS
2022 CHINA

Hadoop Distributed Environment Construction

Senior Lecturer： Zhou Qiao

Chongqing Technology And Business Institute

2022 BRICS Skills Competition
(BRICS Future Skills Challenge)

CONTENTS

01. (Introduction)

02. (Technical Preparation)

03. (Installation Process)

What is Hadoop?

- □ Hadoop： a distribution file system framework, lead by Apache.
- □ It’s a software framework capable of distributed processing of large amounts of data.
- □ Hadoop was named after its creator's (Doug , Cutting's) child's stuffed elephant.

Features in Hadoop

- □ Scalable
  - □ Hadoop can be trusted for its ability to store and process data in bits.
- □ Economical
  - □ It distributes data and processing across clusters of commonly used computers
    - □ These clusters can number into the thousands of nodes.

Features in Hadoop

- □Efficient
  - □By distributing data, Hadoop can process it in parallel on the nodes where the data resides.
- □Reliable
  - □Hadoop automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures.

Hadoop core project

- □Hadoop Distributed File System, HDFS.
- □MapReduce programming model

- □Hadoop's HDFS and MapReduce provide users with a distributed infrastructure where the underlying details of the system are transparent.
- □For Hadoop clusters, it can be divided into two categories: Master and Slave .

- □An HDFS cluster include： a NameNode and several DataNodes.

- □The MapReduce framework include： a JobTracker running on the master node and a TaskTracker running on each cluster slave node.

HDFS provides support for file operations and storage during MapReduce task processing.

On the basis of HDFS, MapReduce realizes tasks such as distribution, tracking, and execution, and collects the results. The two interact to complete the main tasks of Hadoop distributed clusters.

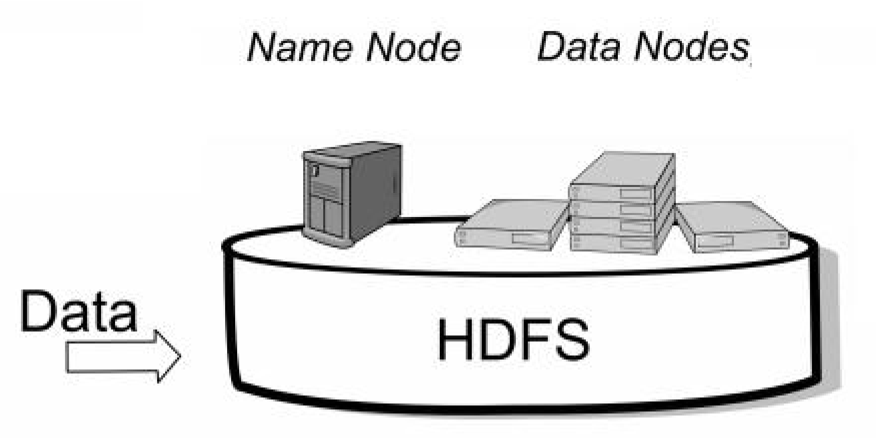
What is HDFS?

- Hadoop Distributed File System （HDFS）

Hadoop Distributed File System (HDFS) is the primary storage system used by Hadoop applications.

HDFS creates multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, extremely rapid computations.

Functions of the main components of HDFS



NameNode	DataNode
• store metadata	• Store file content
•Metadata is kept in memory	•File contents are saved on disk
• Save the mapping relationship between file and block, block and datanode	•Maintains the mapping relationship between block id and datanode local file

Files in HDFS

Files in HDFS

- The file is divided into blocks (block, the default size is 128M), in blocks, each block has multiple copies stored on different machines, the number of copies can be specified when the file is generated (default 3)
- NameNode is the master node, which stores metadata of files such as file name, file directory structure, file attributes (generation time, number of copies, file permissions), as well as the block list of each file and the DataNode where the block is located, etc.
- DataNode stores file block data in the local file system, as well as the checksum of the block data
- Files can be created, deleted, moved, or renamed, and the contents of the file cannot be modified after the file is created, written, and closed.

What is MapReduce Programming Model

- □ MapReduce is a programming model and an associated implementation for processing and generating large data sets
- □ Put forward by Google
- □ Schema of map and reduce functions

  map: input → list(k, v)

  reduce: (k,list(v)) → output
- □ This model can express the work of realistic world
- □ More than 1000 MapReduce procedures are run on Google racks everday

What is MapReduce

□MapReduce is a distributed computing model proposed by Google, which is mainly used in the search field to solve the computing problem of massive data.

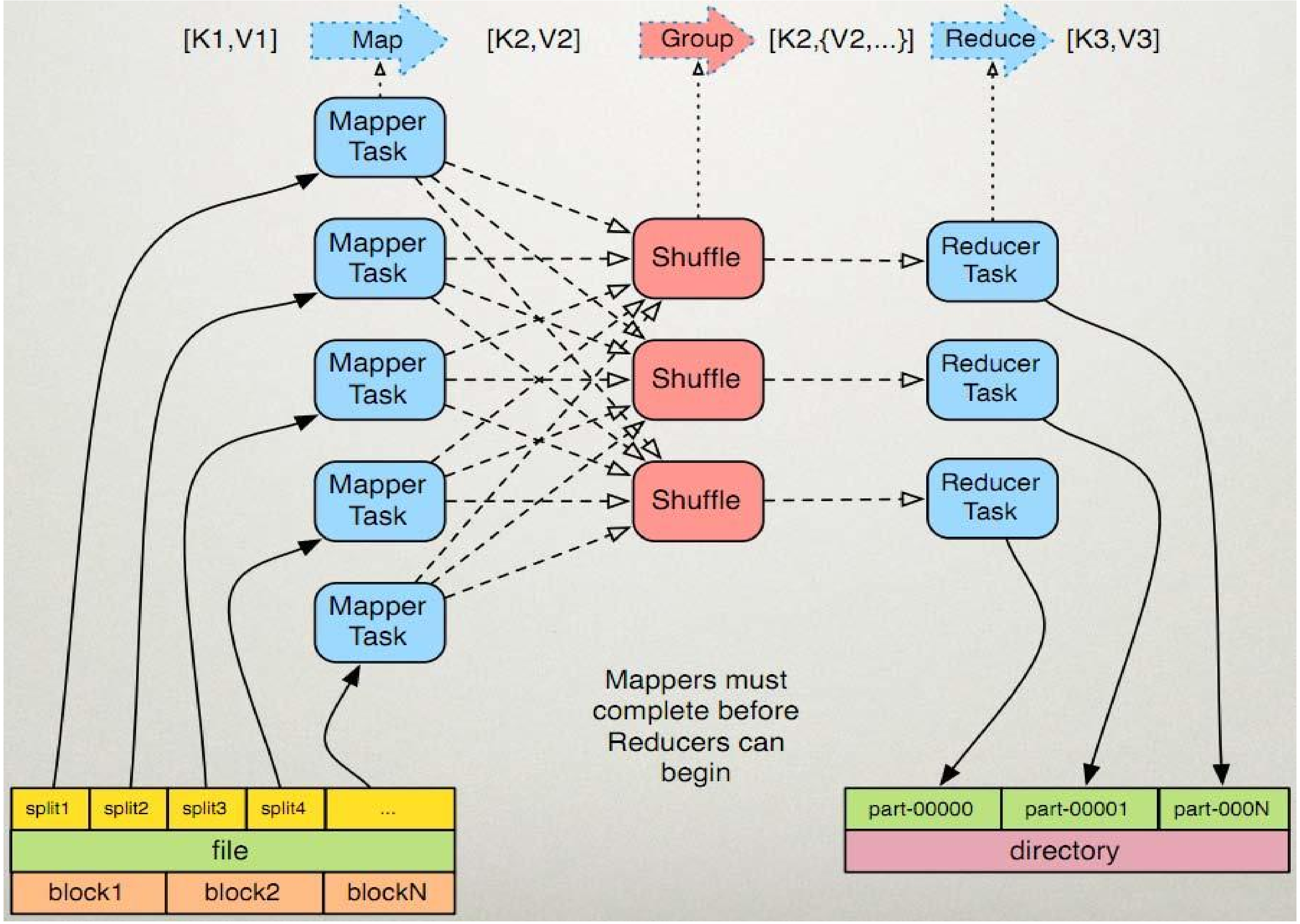
□MR consists of two stages: Map and Reduce. Users only need to implement two functions, map() and reduce(), to realize distributed computing. Users only need to focus on domain knowledge and solve domain problems. No need to consider the underlying details of distributed computing such as: network communication, task distribution, task scheduling, etc.

- □ master-slave structure
  - □ Master node, only one: JobTracker
  - □ Slave nodes, there are many: TaskTrackers
- □ JobTracker is responsible for:
  - □ Receive computing tasks submitted by customers
  - □ Allocate computing tasks to TaskTrackers for execution
  - □ Monitor the execution of TaskTracker
- □ TaskTrackers are responsible for:
  - □ Execute the computing tasks assigned by the JobTracker

Architecture of MapReduce

- □ Map is responsible for breaking up the data, and Reduce is responsible for aggregating the data. Users only need to implement the two interfaces of map and reduce to complete the calculation of terabyte-level data.
- □ The implementation of MapReduce also adopts the Master/Slave structure. Master is called JobTracker, and Slave is called TaskTracker.
- □ The computation submitted by the user is called a Job, and each Job is divided into several Tasks. JobTracker is responsible for scheduling Jobs and Tasks, and TaskTracker is responsible for executing Tasks.

Principle of Mapreduce



◆Execution steps:

1. Map task processing

1.1 Read the content of the input file and parse it into key and value pairs. For each line of the input file, it is parsed into key and value pairs. The map function is called once for each key-value pair.

1.2 Write your own logic, process the input key and value, and convert it into a new key and value output.

1.3 Partition the output key and value.

1.4 Sort and group data in different partitions by key. The values of the same key are put into a set.

1.5 (Optional) Reduce the grouped data.

- Execution steps:

2.Reduce task processing

2.1 For the output of multiple map tasks, according to different partitions, copy to different reduce nodes through the network.

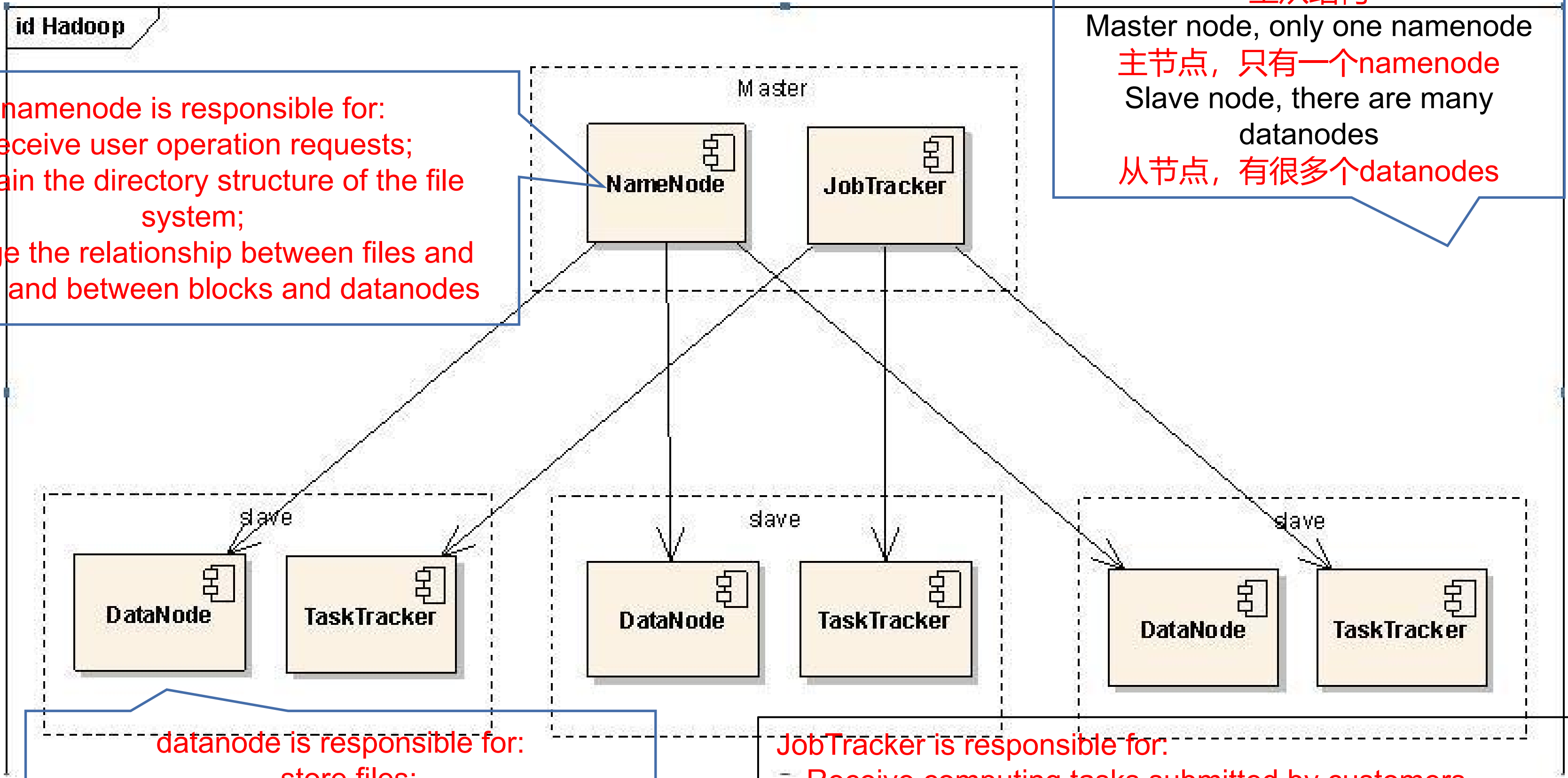
2.2 Merge and sort the outputs of multiple map tasks. Write the logic of the reduce function, process the input key and value, and convert it into a new key and value output.

2.3 Save the output of reduce to a file.

Hadoop Architecture

master-slave structure
主从结构
Master node, only one namenode
主节点，只有一个namenode
Slave node, there are many datanodes
从节点，有很多个datanodes

namenode is responsible for:
Receive user operation requests;
maintain the directory structure of the file system;
Manage the relationship between files and blocks, and between blocks and datanodes



datanode is responsible for:
store files;
The file is divided into blocks and stored on the disk;
To ensure data security, there are multiple copies of the file

JobTracker is responsible for:
- Receive computing tasks submitted by customers
  Assign computing tasks to TaskTrackers for execution
  Monitor the execution of TaskTracker

TaskTrackers are responsible for:
  Execute the computing tasks assigned by the JobTracker

CONTENTS

01. (Introduction)

02. (Technical Preparation)

03. (Installation Process)

一、 Hadoop deployment method

(Standalone mode)

(Pseudo-Distributed mode)

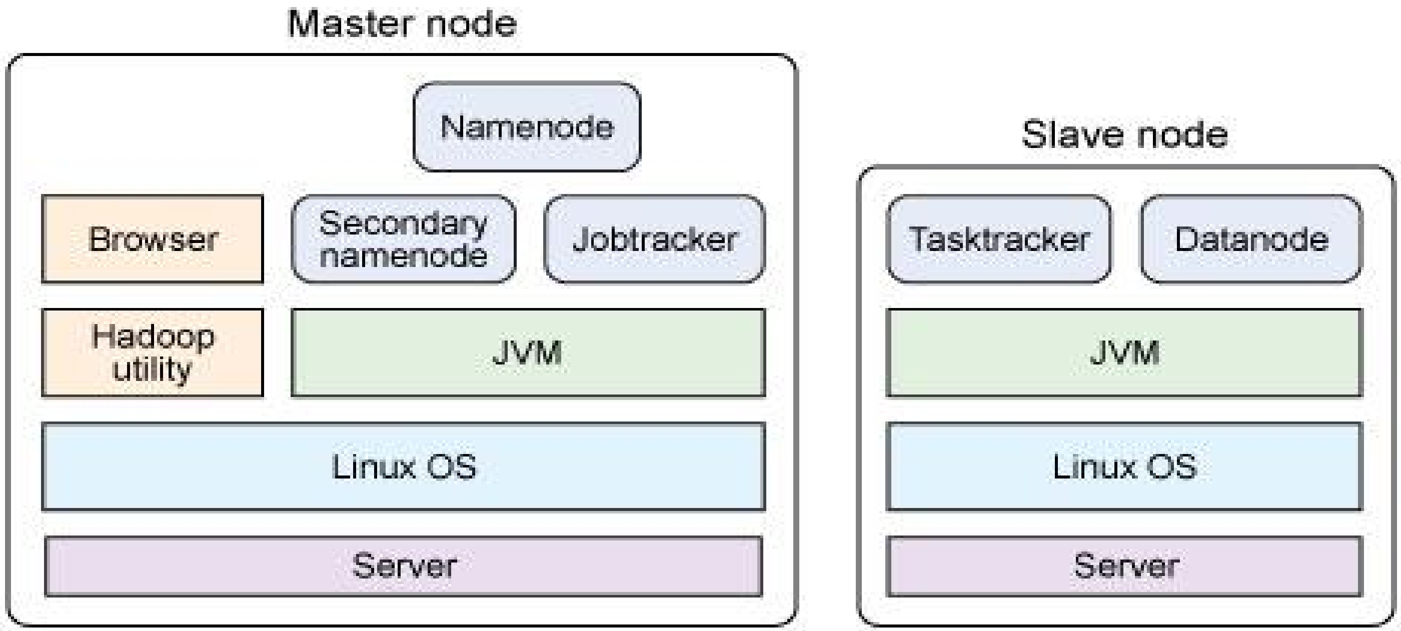
(Fully distributed mode)

(1) Standalone mode: Also known as stand-alone mode, in this mode, there is no need to run any daemons. In general, this mode is used for debugging during the learning or development phase.

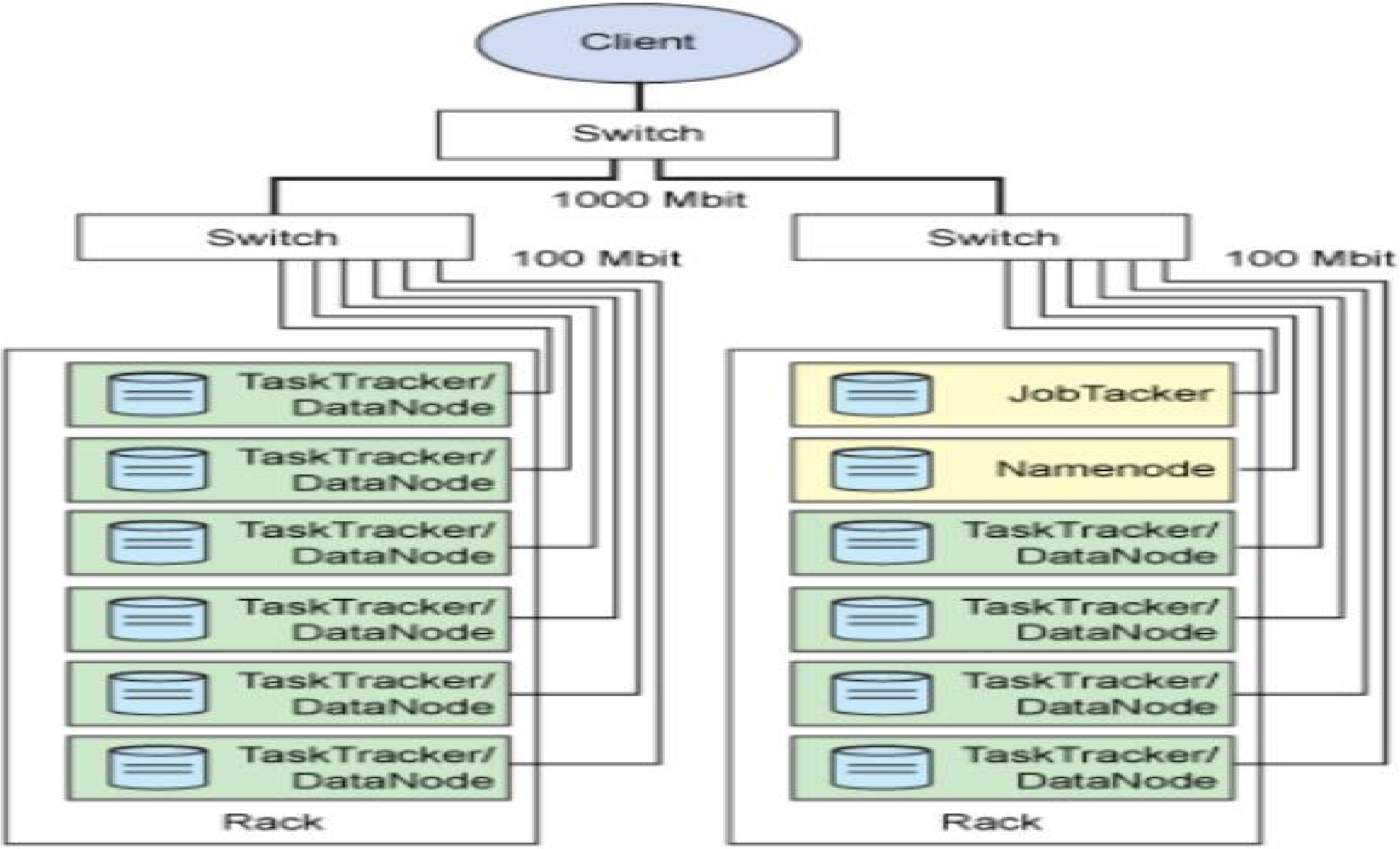
□ (2) Pseudo-distributed mode: The daemon of the Hadoop program runs on a node. Usually, the pseudo-distributed mode is used to debug the code of the Hadoop distributed program and whether the program is executed correctly. The pseudo-distributed mode is one of the fully distributed modes. special case.

□ (3) Fully distributed mode: Hadoop daemons run on independent nodes, and different nodes play different roles. In actual work application development, this mode is usually used to build enterprise-level Hadoop systems.

Single node physical structure



Physical distribution of the Hadoop cluster



二、choice of hardware

1、Need to build a cluster with several nodes?

A server is a node。
Determined by the business scenario。

The more nodes, the better the performance, but the higher the cost。

Regarding the Hadoop cluster, there is a minimum number of nodes for your reference.

Contains the following 3 nodes：
NameNode、SecondaryNameNode、DataNode

二、choice of hardware

1、Need to build a cluster with several nodes?

□The first one is used to record all data distribution, the running process is NameNode

□The second one is used to back up all data distributions. After all, when the previous server goes down, data can be restored through this server. So, the program running on this server is SecondaryNameNode

□The third one is used to store the actual data, and the running process is the DataNode

二、choice of hardware

2、How to choose and configure each service in a cluster environment?

This question is mainly about how to choose memory, CPU, storage, etc.

In a Hadoop cluster environment, the most important thing is the server run by the NameNode, because it plays the role of scheduling and coordinating the work of the entire cluster.

Here is a reference: Generally, 1GB of memory can manage millions of block files.

二、choice of hardware

For example: bolck is 128M, replica is 3, 200 clusters, 4TB data, the required Namenode memory is: 200 (number of servers) x 4194304MB (4TB data) / (128MB x 3) = 2184533.33 files = 2.18 million file, so the memory value is close to 2.2G

Because the secondary namenode is the backup node, it is also the same as the memory configuration of the datanode

二、choice of hardware

Regarding the choice of CPU, because Hadoop is a distributed computing operation, its operating model is basically intensive parallel computing. Therefore, the recommended CPU should be multi-channel and multi-core as much as possible, and if conditions permit, each node must do so.

二、choice of hardware

3、How to configure the storage size of each node in a cluster environment?

It is clear that the size of the data volume determines the overall storage size of the cluster, and also determines the scale of the entire cluster!

The main consideration is the growth of data

三、software selection

Regarding the choice of Hadoop cluster environment software, the selection mainly focuses on these software products: OS operating system, Hadoop version, JDK version, Hive version, MySQL version, etc.

1、Choice of operating system

We know that a hadoop cluster environment requires many servers.

The Linux operating system is recommended, the reason is very simple, open source and free

三、software selection

2、Hadoop version

hadoop-3.1.3 JDK 1.8

hadoop-2.10.1 JDK 1.7 or 1.8

hadoop-2.9.2 JDK 1.7 or 1.8

hadoop-2.8.2 JDK 1.7+

hadoop-2.7.1 JDK 1.7+

三、Node planning

主机名	IP地址	运行的服务	配置	操作系统
master	192.168.20.10	namenode	2核 2G 50G	Centos7.9
node1	192.168.20.11	datanode	2核 2G 50G	Centos7.9
node2	192.168.20.12	datanode	2核 2G 50G	Centos7.9

四、Prepare the software before installation

Essential software:

- VitualVox 或 vmware
- centos7.9
- jdk-8u331-linux-aarch64.tar.gz
- hadoop-2.10.1.tar.gz

Other software:

-
- CRT or Mobaxterm

CONTENTS

01. (Introduction)

02. (Technical Preparation)

03. (Installation Process)

installation steps

1. OS configuration
2. Configure password-free login
3. install jdk
4. Configure hadoop cluster
5. Start cluster and tests

一、OS configuration

Prepare three virtual machines and set ip

1.Turn off firewall and selinux

```
[root@localhost ~]# systemctl stop firewalld && systemctl disable firewalld
[root@localhost ~]# sed -
i ‘s/SELINUX=enforcing/SELINUX=disabledg’ /etc/selinux/config
[root@localhost ~]# setenforce 0
[root@localhost ~]# getenforce
```

2.Modify the hostname and set the hosts file

```
[root@master ~]# hostnamectl set-hostname master
[root@node1 ~]# hostnamectl set-hostname node1
[root@node2 ~]# hostnamectl set-hostname node2

[root@localhost ~]# cat /etc/hosts
192．168．20．10 master
192．168．20．11 node1
192．168．20．12 node2
```

二、Configure password-free login

1.Modify the hostname and set the hosts file

```
[root@master ~]# ssh-keygen
[root@master ~]# ssh-copy-id master
[root@master ~]# ssh-copy-id node1
[root@master ~]# ssh-copy-id node2
```

2.Password-free login test

```
[root@master ~]# ping master
[root@master ~]# ping node1
[root@master ~]# ping node2
```

三、Install JDK

1.Download and extract jdk to /opt/ directory

```
[root@master ~]# tar -zxvf jdk-8u191-linux-x64.tar.gz -C /opt/
```

Pay attention to user permissions Pay attention to user permissions ：

```
chown -R root:root jdk1.8.0_191
```

2.configure jdkenvironmentvariable

```
[root@master ~]# vi /etc/profile
export JAVA_HOME=/opt/jdk1.8.0_191
export PATH=$PATH:$JAVA_HOME/bin

#Immediately effective environment variables
[root@master ~]# source /etc/profile
#Test installation was successful
[root@master ~]# java -version
java version "1.8.0_191"
```

三、Install JDK

3.copy jdk file to other host

```
[root@master ~]# scp -r /opt/jdk1.8.0_191 root@node1:/opt/
[root@master ~]# scp -r /opt/jdk1.8.0_191 root@node2:/opt/
```

4.Copy environment variable configuration files t

```
[root@master ~]# scp -r /etc/profile root@node1:/etc/
[root@master ~]# scp -r /etc/profile root@node2:/etc/
# 立即生效
[root@node1 ~]# source /etc/profile
[root@node2 ~]# source /etc/profile
```

5.verify jdk

```
# 查看版本
[root@master ~]# java -version
```

四、Configure the Hadoop cluster

1.Download and extract hadoop to the /opt/ dire

```
[root@master ~]# tar -zxvf hadoop-2.10.1.tar.gz -C /opt/
[root@master ~]# ll /opt/
total 0
drwxr-xr-x. 9 1000 1000 149 Sep 14  2020 hadoop-2.10.1
drwxr-xr-x. 7   10  143 245 Oct  6  2018 jdk1.8.0_191
```

2.Configure environment variables

```
[root@master ~]# vi /etc/profile
export PATH=$PATH:$JAVA_HOME/bin:/opt/hadoop-2.10.1/sbin:/opt/hadoop-
2.10.1/bin
# 立即生效
[root@master ~]# source /etc/profile
# 查看版本
[root@master ~]# hadoop version
Hadoop 2.10.1
```

四、Configure the Hadoop cluster

3.Configure hadoop cluster

xml file corresponding to each component

common components------>core-site.xml
HDFS components------>hdfs-site.xml
MapReduce components------>mapred-site.xml
YARN components------>yarn-site.xml

Go to the configuration file directory

```
[root@master ~]# cd /opt/hadoop-2.10.1/etc/hadoop/
```

四、Configure hadoop cluster

3.Configure hadoop cluster

3.1Configure hadoop-env.sh file

```
[root@master hadoop]# vi hadoop-env.sh
export JAVA_HOME=/opt/jdk1.8.0_191  # JDK的安装路径
```

3.2Configure core-site.xml

```
[root@master hadoop]# vi core-site.xml
<configuration>
  <!-- 指定hdfs中namenode的地址 -->
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>

  <!-- 指定hadoop运行时产生文件的存储目录 -->
  <property>
    <name>dfs.tmp.dir</name>
    <value>file:///opt/hadoop-data/</value>
  </property>
</configuration>
```

四、Configure hadoop cluster

3.3Configure hdfs-site.xml file

```
[root@master hadoop]# vi hdfs-site.xml
<configuration>
  <!--Set the number of dfs replicas, the default is 3 if not set  -->
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <!--Set the port for secondname  -->
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>node1:50090</value>
  </property>
</configuration>
```

3.4Configure slaves file

The slaves file specifies which nodes run the DataNode process

```
[root@master hadoop]# vi slaves
master
node1
node2
```

四、Configure the Hadoop cluster

3.5Configure maperd-env.sh file

```
[root@master hadoop]# vi maperd-env.sh
# 找到export JAVA_HOME=在后面添加java环境变量
export JAVA_HOME=/opt/jdk1.8.0_191
```

3.6Configure maprted-site.xml

如果没有该文件复制mapred-site.xml.template文件为mapred-site.xml

```
[root@master hadoop]# cp mapred-site.xml.template
mapred-site.xml
[root@master hadoop]# vi mapred-site.xml
<configuration>
<!--指定mapreduce运行在yarn上 -->
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

四、Configure the Hadoop cluster

3.7Configure yarn-env.sh file

```
[root@master hadoop]# vi yarn-env.sh
#Find exportJAVA_HOME=Add java environment variable behind
export JAVA_HOME=/opt/jdk1.8.0_191
```

3.8Configure yarn-site.xml

```
[root@master hadoop]# vi yarn-site.xml
<configuration>
  <!--Specify the address of the ResourceManager -->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>master</value>
  </property>
  <!--Specify how the reducer gets the data -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

四、Configure the Hadoop cluster

4.Distribute the installed hadoop to other host r

```
[root@master ~]# scp -r /opt/hadoop-2.10.1 root@node1:/opt/
[root@master ~]# scp -r /opt/hadoop-2.10.1 root@node2:/opt/
```

5.Copy the environment variable configuration f

```
[root@master ~]# scp -r /etc/profile root@node1:/etc/
[root@master ~]# scp -r /etc/profile root@node2:/etc/
#Execute separately and take effect immediately
[root@node1 ~]# source /etc/profile
[root@node2 ~]# source /etc/profile
#View version
[root@node1 ~]# hadoop version
Hadoop 2.10.1
[root@node2 ~]# hadoop version
Hadoop 2.10.1
```

五、Start the cluster and verify the tests

1.The first time you start the cluster, you need t

```
[root@master ~]# hdfs namenode -format
[root@master ~]# sh start-dfs.sh
```

or **start-all.sh**

2.start yarn

```
#on the host master as we configure in yarn-site.xml
```

to start on the master, so it needs to be started separately

```
[root@master ~]#sh start-yarn.sh
```

3.jps view process

```
[root@master ~]# jps
11602 NameNode
12717 NodeManager
13069 Jps
12607 ResourceManager
```

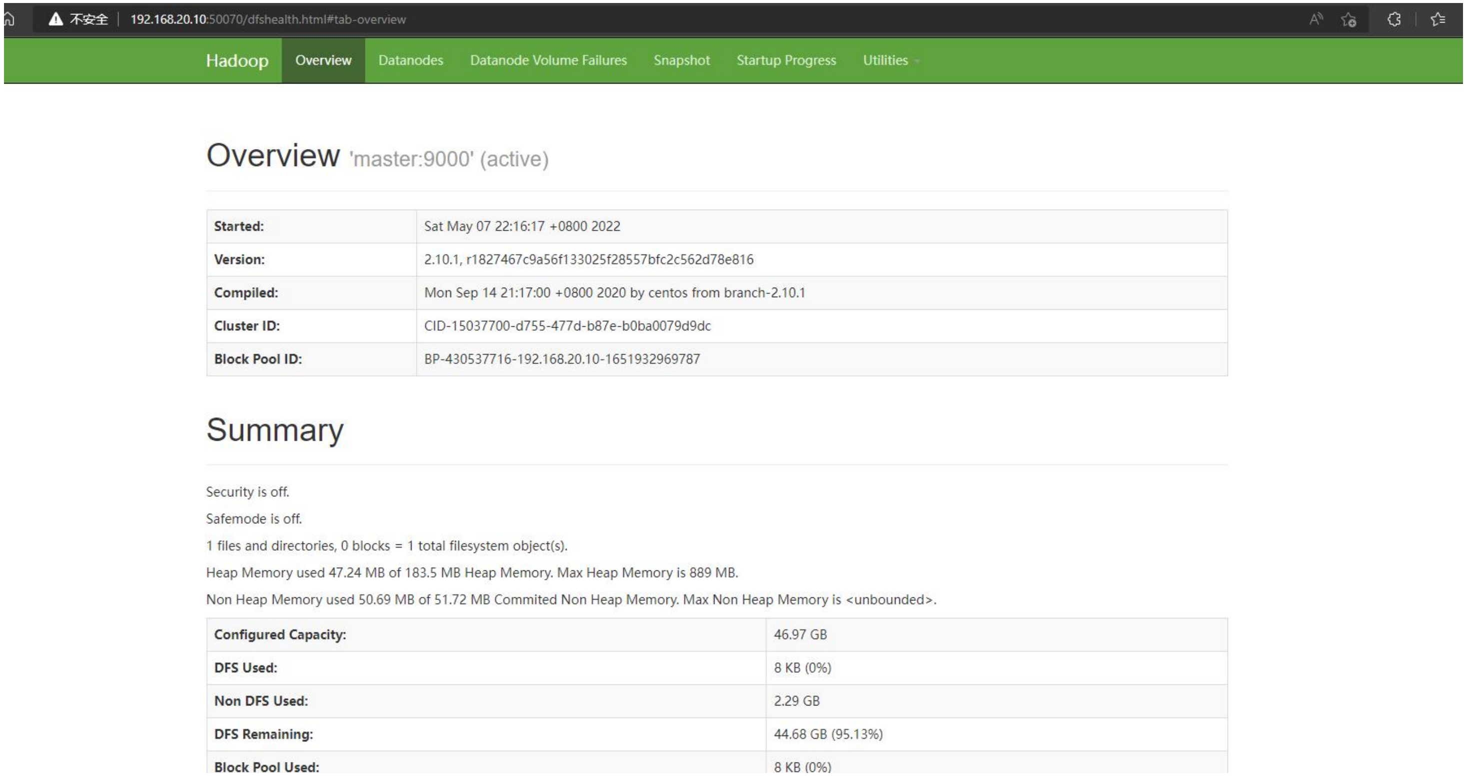
```
[root@node1 ~]# jps
3187
SecondaryNameNode
3299 NodeManager
3415 Jps
```

```
[root@node2 ~]# jps
3556 NodeManager
3672 Jps
3321 DataNode
```

五、Start the cluster and verify the tests

4.View on the web

Enter master in the address bar of the web page: 50070



Overview 'master:9000' (active)

Started:	Sat May 07 22:16:17 +0800 2022
Version:	2.10.1, r1827467c9a56f133025f28557bfc2c562d78e816
Compiled:	Mon Sep 14 21:17:00 +0800 2020 by centos from branch-2.10.1
Cluster ID:	CID-15037700-d755-477d-b87e-b0ba0079d9dc
Block Pool ID:	BP-430537716-192.168.20.10-1651932969787

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

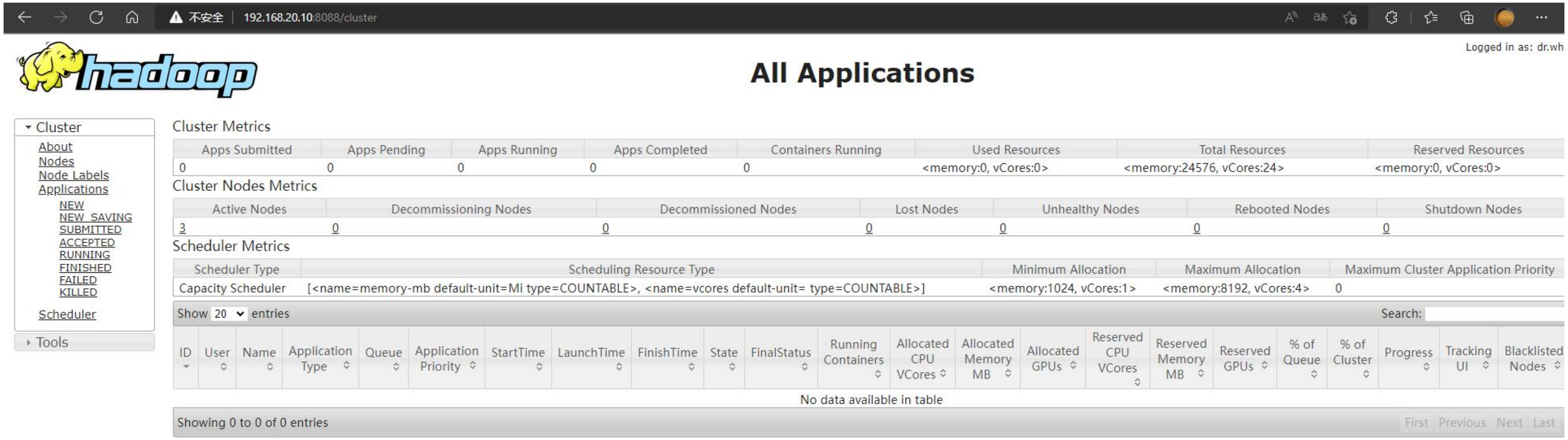
Heap Memory used 47.24 MB of 183.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 50.69 MB of 51.72 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	46.97 GB
DFS Used:	8 KB (0%)
Non DFS Used:	2.29 GB
DFS Remaining:	44.68 GB (95.13%)
Block Pool Used:	8 KB (0%)

五、Start the cluster and verify the tests

Explorer view master:8088



不安全 | 192.168.20.10:8088/cluster

Logged in as: dr.wh

All Applications

- Cluster
  - About
  - Nodes
  - Node Labels
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resources
0	0	0	0	0	<memory:0, vCores:0>	<memory:24576, vCores:24>	<memory:0, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
3	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[<name=memory-mb default-unit=Mi type=COUNTABLE>, <name=vcores default-unit= type=COUNTABLE>]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries   Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																						

Showing 0 to 0 of 0 entries

Hadoop common commands

1、 View files and folders：bin/hadoop fs -ls　HDFS目录
eg：　bin/hadoop fs -ls　/
(查看HDFS根目录下的文件目录)

2、upload files上传文件：bin/hadoop fs -put local_dir HDFS_dir
eg: bin/hadoop fs -put　~/jdk-7u25-linux-i586.gz　/
(当前根目录下的文件上传到HDFS根目录下)

3、download file下载文件：
bin/hadoop fs -get　HDFS目录　本机目录
eg:　bin/hadoop fs -get　/jdk-7u25-linux-i586.gz　~/Desktop
(将HDFS根目录下的文件下载到当前根目录下的Desktop)

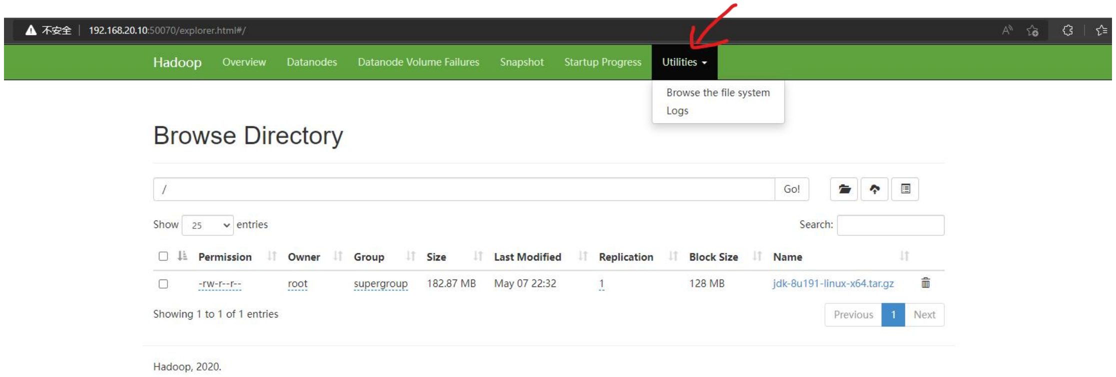
六、测试验证

1.test upload 测试上传

该测试使用Hadoop API测试

```
[root@master ~]# hadoop fs -put /root/jdk-8u191-linux-x64.tar.gz /
[root@master ~]# hadoop fs -ls /
Found 1 items
-rw-r--r--   1 root supergroup  191753373 2022-05-07 10:32 /jdk-8u191-linux-x64.tar.gz
```

网页查看：



不安全 | 192.168.20.10:50070/explorer.html#/

Browse Directory

/

Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	root	supergroup	182.87 MB	May 07 22:32	1	128 MB	jdk-8u191-linux-x64.tar.gz

Showing 1 to 1 of 1 entries

Hadoop, 2020.