目录

[1. 实验一：hadoop全分布配置 2](#_Toc49350165)

[1.1. 实验目的 2](#_Toc49350166)

[1.2. 实验要求 2](#_Toc49350167)

[1.3. 实验环境 2](#_Toc49350168)

[1.4. 实验过程 2](#_Toc49350169)

[1.4.1. 实验任务一：在Master节点上安装Hadoop 2](#_Toc49350170)

[1.4.2. 实验任务二：配置core-site.xml文件参数 3](#_Toc49350171)

[1.4.3. 实验任务三：配置hdfs-site.xml文件参数 4](#_Toc49350172)

[1.4.4. 实验任务四：配置mapred-site.xml 5](#_Toc49350173)

[1.4.5. 实验任务五：配置yarn-site.xml 6](#_Toc49350174)

[1.4.6. 实验任务六：Hadoop其它相关配置 8](#_Toc49350175)

# 实验一：hadoop全分布配置

## 实验目的

完成本实验，您应该能够：

* 掌握hadoop全分布的配置
* 掌握hadoop全分布的安装
* 掌握hadoop配置文件的参数意义

## 实验要求

* 熟悉hadoop全分布的安装
* 了解hadoop配置文件的意义

## 实验环境

本实验所需之主要资源环境如表1-1所示。

表1-1 资源环境

|  |  |
| --- | --- |
| **服务器集群** | 3个以上节点，节点间网络互通，各节点最低配置：双核CPU、8GB内存、100G硬盘 |
| **运行环境** | CentOS 7.4 |
| **大数据平台** | H3C教学与实践管理系统 |
| **服务和组件** | 完成前面章节的实验，其他服务及组件根据实验需求安装 |

## 实验过程

### 实验任务一：在Master节点上安装Hadoop

#### 步骤一：解压缩hadoop-2.7.1.tar.gz安装包到/usr目录下

#root用户下删除原先安装的单机hadoop

[root@master ~]$ rm -rf /usr/local/src/hadoop/

[root@master ~]$ ls /usr/local/src/

**java**

#解压hadoop安装包

[root@master ~]# tar -zxvf /opt/software/hadoop-2.7.1.tar.gz -C /usr/local/src/

#### 步骤二：将hadoop-2.7.1文件夹重命名为hadoop

[root@master ~]# mv /usr/local/src/hadoop-2.7.1 /usr/local/src/hadoop

#### 步骤三：配置Hadoop环境变量

#在文件末尾添加以下配置信息

[root@master ~]# vi /etc/profile

注意：此处配置信息在第二章已配置，没配置则添加内容如下

# set java environment

export JAVA\_HOME=/usr/local/src/java

export PATH=$PATH:$JAVA\_HOME/bin

# set hadoop environment

export HADOOP\_HOME=/usr/local/src/hadoop

export PATH=$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH

#### 步骤四：使配置的Hadoop的环境变量生效

[root@master ~]# su - hadoop

[hadoop@master ~]# source /etc/profile

[hadoop@master ~]# exit

#### 步骤五：执行以下命令修改hadoop-env.sh配置文件。

[root@master ~]# cd /usr/local/src/hadoop/etc/hadoop/

[root@master hadoop]# vi hadoop-env.sh

#在文件末尾添加以下配置信息

export JAVA\_HOME=/usr/local/src/java

### 实验任务二：配置core-site.xml文件参数

执行以下命令修改core-site.xml配置文件。

[root@master hadoop]# vi core-site.xml

#在文件中<configuration>和</configuration>一对标签之间追加以下配置信息

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://master:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>file:/usr/local/src/hadoop/tmp</value>

</property>

</configuration>

如没有配置hadoop.tmp.dir参数，此时系统默认的临时目录为：/tmp/hadoop-hadoop。该目录在每次Linux系统重启后会被删除，必须重新执行Hadoop文件系统格式化命令，否则Hadoop运行会出错。

该配置文件中主要的参数、默认值、参数解释如下表4-5所示：

表4-5 core-site.xml配置文件主要参数

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 参数名 | 默认值 | 参数解释 |
| 1 | fs.defaultFS | file:/// | 文件系统主机和端口 |
| 2 | io.file.buffer.size | 4096 | 流文件的缓冲区大小 |
| 3 | hadoop.tmp.dir | /tmp/hadoop-${user.name} | 临时文件夹 |

### 实验任务三：配置hdfs-site.xml文件参数

执行以下命令修改hdfs-site.xml配置文件。

[root@master hadoop]# vi hdfs-site.xml

#在文件中<configuration>和</configuration>一对标签之间追加以下配置信息

<configuration>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/src/hadoop/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/src/hadoop/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

</configuration>

对于Hadoop的分布式文件系统HDFS而言，一般都是采用冗余存储，冗余因子通常为3，也就是说，一份数据保存三份副本。所以，修改dfs.replication的配置，使HDFS文件的备份副本数量设定为3个。

该配置文件中主要的参数、默认值、参数解释如下表4-4所示：

表4-4 hdfs-site.xml配置文件主要参数

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 参数名 | 默认值 | 参数解释 |
| 1 | dfs.namenode.secondary.http-address | 0.0.0.0:50090 | 定义HDFS对应的HTTP服务器地址和端口 |
| 2 | dfs.namenode.name.dir | file://${hadoop.tmp.dir}/dfs/name | 定义DFS的名称节点在本地文件系统的位置 |
| 3 | dfs.datanode.data.dir | file://${hadoop.tmp.dir}/dfs/data | 定义DFS数据节点存储数据块时存储在本地文件系统的位置 |
| 4 | dfs.replication | 3 | 缺省的块复制数量 |
| 5 | dfs.webhdfs.enabled | true | 是否通过http协议读取hdfs文件，如果选是，则集群安全性较差 |

### 实验任务四：配置mapred-site.xml

在“/usr/local/src/hadoop/etc/hadoop”目录下有一个mapred-site.xml.template，需要修改文件名称，把它重命名为mapred-site.xml，然后把mapred-site.xml文件配置成如下内容。

执行以下命令修改mapred-site.xml配置文件。

[root@master hadoop]# cp mapred-site.xml.template mapred-site.xml

[root@master hadoop]# vi mapred-site.xml

#在文件中<configuration>和</configuration>一对标签之间追加以下配置信息

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>master:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>master:19888</value>

</property>

</configuration>

该配置文件中主要的参数、默认值、参数解释如下表4-6所示：

表4-6 mapred-site.xml配置文件主要参数

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 参数名 | 默认值 | 参数解释 |
| 1 | mapreduce.framework.name | local | 取值local、classic或yarn其中之一，如果不是yarn，则不会使用YARN集群来实现资源的分配 |
| 2 | mapreduce.jobhistory.address | 0.0.0.0:10020 | 定义历史服务器的地址和端口，通过历史服务器查看已经运行完的Mapreduce作业记录 |
| 3 | mapreduce.jobhistory.webapp.address | 0.0.0.0:19888 | 定义历史服务器web应用访问的地址和端口 |

### 实验任务五：配置yarn-site.xml

执行以下命令修改yarn-site.xml配置文件。

[root@master hadoop]# vi yarn-site.xml

#在文件中<configuration>和</configuration>一对标签之间追加以下配置信息

<configuration>

<property>

<name>yarn.resourcemanager.address</name>

<value>master:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>master:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>master:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>master:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>master:8088</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

该配置文件中主要的参数、默认值、参数解释如下表4-7所示：

表4-7 yarn-site.xml配置文件主要参数

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 参数名 | 默认值 | 参数解释 |
| 1 | yarn.resourcemanager.address | 0.0.0.0:8032 | ResourceManager 提供给客户端访问的地址。客户端通过该地址向RM提交应用程序，杀死应用程序等 |
| 2 | yarn.resourcemanager.scheduler.address | 0.0.0.0:8030 | 定义历史服务器的地址和端口，通过历史服务器查看已经运行完的Mapreduce作业记录 |
| 3 | yarn.resourcemanager.resource-tracker.address | 0.0.0.0:8031 | ResourceManager 提供给NodeManager的地址。NodeManager通过该地址向RM汇报心跳，领取任务等 |
| 4 | yarn.resourcemanager.admin.address | 0.0.0.0:8033 | ResourceManager 提供给管理员的访问地址。管理员通过该地址向RM发送管理命令等 |
| 5 | yarn.resourcemanager.webapp.address | 0.0.0.0:8088 | ResourceManager对web 服务提供地址。用户可通过该地址在浏览器中查看集群各类信息 |
| 6 | yarn.nodemanager.aux-services | org.apache.hadoop.mapred.ShuffleHandler | 通过该配置项，用户可以自定义一些服务，例如Map-Reduce的shuffle功能就是采用这种方式实现的，这样就可以在NodeManager上扩展自己的服务。 |

### 实验任务六：Hadoop其它相关配置

#### 步骤一：配置masters文件

执行以下命令修改masters配置文件。

#加入以下配置信息

[root@master hadoop]# vi masters

master

#### 步骤二：配置slaves文件

需要把所有数据节点的主机名写入该文件，每行一个，默认为localhost（即把本机作为数据节点），所以，在伪分布式配置时，就采用了这种默认的配置，使得节点既作为名称节点也作为数据节点。在进行分布式配置时，可以保留localhost，让Master节点同时充当名称节点和数据节点，或者也可以删掉localhost这行，让Master节点仅作为名称节点使用。

本教材Master节点仅作为名称节点使用，因此将slaves文件中原来的localhost删除，并添加slave1、slave2节点的IP地址。

执行以下命令修改slaves配置文件。

#删除localhost，加入以下配置信息

[root@master hadoop]# vi slaves

slave1

slave2

#### 步骤三：新建目录

执行以下命令新建/usr/local/src/hadoop/tmp、/usr/local/src/hadoop/dfs/name、/usr/local/src/hadoop/dfs/data三个目录。

[root@master hadoop]# mkdir /usr/local/src/hadoop/tmp

[root@master hadoop]# mkdir /usr/local/src/hadoop/dfs/name -p

[root@master hadoop]# mkdir /usr/local/src/hadoop/dfs/data -p

#### 步骤四：修改目录所属用户

执行以下命令修改/usr/local/src/hadoop目录的权限。

#查看hadoop目录权限和用户与用户组

[root@master hadoop]# ll /usr/local/src/

**total 0**

**drwxr-xr-x 11 10021 10021 171 Aug 26 15:37 hadoop**

**drwxr-xr-x 8 hadoop hadoop 255 Sep 14 2017 java**

#修改hadoop目录所属用户

[root@master hadoop]# chown -R hadoop:hadoop /usr/local/src/hadoop/

#再次查看权限

[root@master hadoop]# ll /usr/local/src/

**total 0**

**drwxr-xr-x 11 hadoop hadoop 171 Aug 26 15:37 hadoop**

**drwxr-xr-x 8 hadoop hadoop 255 Sep 14 2017 java**

#### 步骤五：同步配置文件到Slave节点

上述配置文件全部配置完成以后，需要执行以下命令把Master节点上的“/usr/local/src/hadoop”文件夹复制到各个Slave节点上，并修改文件夹访问权限。

（1）将Master上的Hadoop，Java安装文件同步到slave1、slave2。

#同步Hadoop文件

[root@master hadoop]#cd

[root@master ~]# scp -r /usr/local/src/hadoop/ root@slave1:/usr/local/src/

[root@master ~]# scp -r /usr/local/src/hadoop/ root@slave2:/usr/local/src/

#同步Java文件

[root@master ~]# scp -r /usr/local/src/java/ root@slave1:/usr/local/src/

[root@master ~]# scp -r /usr/local/src/java/ root@slave2:/usr/local/src/

（2）将master环境变量同步到每个Slave节点上。

[root@master ~]# scp /etc/profile root@slave1:/etc/

[root@master ~]# scp /etc/profile root@slave2:/etc/

（3）删除slave节点上自带的java

#**slave1**

[root@slave1 ~]# rpm -qa | grep java

**java-1.8.0-openjdk-headless-1.8.0.131-11.b12.el7.x86\_64**

**javapackages-tools-3.4.1-11.el7.noarch**

**tzdata-java-2017b-1.el7.noarch**

**java-1.7.0-openjdk-headless-1.7.0.141-2.6.10.5.el7.x86\_64**

**java-1.7.0-openjdk-1.7.0.141-2.6.10.5.el7.x86\_64**

**java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86\_64**

**python-javapackages-3.4.1-11.el7.noarch**

删除相关文件，键入命令

[root@slave1 ~]# rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.131-11.b12.el7.x86\_64

[root@slave1 ~]# rpm -e --nodeps java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86\_64

[root@slave1 ~]# rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.141-2.6.10.5.el7.x86\_64

[root@slave1 ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.141-2.6.10.5.el7.x86\_64

#**slave2**

[root@slave2 ~]# rpm -qa | grep java

**java-1.8.0-openjdk-headless-1.8.0.131-11.b12.el7.x86\_64**

**javapackages-tools-3.4.1-11.el7.noarch**

**tzdata-java-2017b-1.el7.noarch**

**java-1.7.0-openjdk-headless-1.7.0.141-2.6.10.5.el7.x86\_64**

**java-1.7.0-openjdk-1.7.0.141-2.6.10.5.el7.x86\_64**

**java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86\_64**

**python-javapackages-3.4.1-11.el7.noarch**

删除相关文件，键入命令

[root@slave2 ~]# rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.131-11.b12.el7.x86\_64

[root@slave2 ~]# rpm -e --nodeps java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86\_64

[root@slave2 ~]# rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.141-2.6.10.5.el7.x86\_64

[root@slave2 ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.141-2.6.10.5.el7.x86\_64

（4）在每个slave节点上修改/usr/local/src/目录的权限。

#**slave1** root用户下修改权限

[root@slave1 ~]# chown -R hadoop:hadoop /usr/local/src/

[root@slave1 ~]# ll /usr/local/src/

**total 0**

**drwxr-xr-x 11 hadoop hadoop 171 Aug 26 15:43 hadoop**

**drwxr-xr-x 8 hadoop hadoop 255 Aug 26 15:44 java**

#**slave2** root用户下修改权限

[root@slave2 ~]# chown -R hadoop:hadoop /usr/local/src/

[root@slave2 ~]# ll /usr/local/src/

**total 0**

**drwxr-xr-x 11 hadoop hadoop 171 Aug 26 15:43 hadoop**

**drwxr-xr-x 8 hadoop hadoop 255 Aug 26 15:44 java**

（5）在每个Slave节点上切换到hadoop用户。

[root@slave1 ~]#su - hadoop

[root@slave2 ~]#su - hadoop

（6）使每个Slave节点上配置的Hadoop的环境变量生效。

[hadoop@slave1 ~]# source /etc/profile

[hadoop@slave2 ~]# source /etc/profile

（7）验证Slave机器上java和hadoop

[hadoop@slave1 ~]# java -version

**java version "1.8.0\_152"**

**Java(TM) SE Runtime Environment (build 1.8.0\_152-b16)**

**Java HotSpot(TM) 64-Bit Server VM (build 25.152-b16, mixed mode)**

[hadoop@slave2 ~]# java –version

**java version "1.8.0\_152"**

**Java(TM) SE Runtime Environment (build 1.8.0\_152-b16)**

**Java HotSpot(TM) 64-Bit Server VM (build 25.152-b16, mixed mode)**

[hadoop@slave1 ~]# hadoop

**Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]**

**CLASSNAME run the class named CLASSNAME**

**or**

**where COMMAND is one of:**

**………**

[hadoop@slave2 ~]# hadoop

**Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]**

**CLASSNAME run the class named CLASSNAME**

**or**

**where COMMAND is one of:**

**………**