搭建Hadoop伪分布式环境

## 安装 JAVA 环境

### 安装 JDK

使用 yum 来安装1.7版本 OpenJDK：

[root@192 ~]# sudo yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel

安装完成后，输入 java 和 javac 命令，如果能输出对应的命令帮助，则表明jdk已正确安装

### 配置 JAVA 环境变量

执行命令:

编辑 ~/.bashrc，在结尾追加：

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk

保存文件后执行下面命令使 JAVA\_HOME 环境变量生效:

[root@192 ~]# source .bashrc

为了检测系统中 JAVA 环境是否已经正确配置并生效，可以分别执行下面命令:

[root@192 ~]# java -verion

Unrecognized option: -verion

Error: Could not create the Java Virtual Machine.

Error: A fatal exception has occurred. Program will exit.

[root@192 ~]# java -version

openjdk version "1.8.0\_212"

OpenJDK Runtime Environment (build 1.8.0\_212-b04)

OpenJDK 64-Bit Server VM (build 25.212-b04, mixed mode)

[root@192 ~]# $JAVA\_HOME/bin/java -version

openjdk version "1.8.0\_212"

OpenJDK Runtime Environment (build 1.8.0\_212-b04)

OpenJDK 64-Bit Server VM (build 25.212-b04, mixed mode)

若两条命令输出的结果一致，且都为我们前面安装的 openjdk-1.8.0 的版本，则表明 JDK 环境已经正确安装并配置。

## 安装 Hadoop

### 下载 Hadoop

[root@192 ~]# wget http://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz

### 安装 Hadoop

[root@192 ~]# mkdir /usr/local/hadoop

[root@192 ~]# tar -zxf hadoop-2.7.7.tar.gz -C /usr/local/hadoop/

[root@192 ~]# cd /usr/local/hadoop/hadoop-2.7.7/

[root@192 hadoop-2.7.7]# ls

bin etc include lib libexec LICENSE.txt NOTICE.txt README.txt sbin share

[root@192 hadoop-2.7.7]# mv \* /usr/local/hadoop/

[root@192 hadoop-2.7.7]# ls /usr/local/hadoop/

bin hadoop-2.7.7 lib LICENSE.txt README.txt share

etc include libexec NOTICE.txt sbin

检查Hadoop是否已经正确安装:

[root@192 ~]# /usr/local/hadoop/bin/hadoop version

Hadoop 2.7.7

Subversion Unknown -r c1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac

Compiled by stevel on 2018-07-18T22:47Z

Compiled with protoc 2.5.0

From source with checksum 792e15d20b12c74bd6f19a1fb886490

This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.7.jar

如果成功输出hadoop的版本信息，表明hadoop已经成功安装。

## Hadoop 伪分布式环境配置

Hadoop伪分布式模式使用多个守护线程模拟分布的伪分布运行模式。

### 设置 Hadoop 的环境变量

编辑 ~/.bashrc，在结尾追加如下内容：

export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

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

使Hadoop环境变量配置生效:

source ~/.bashrc

### 修改 Hadoop 的配置文件

Hadoop的配置文件位于安装目录的 /etc/hadoop 目录下，在本教程中即位于 /url/local/hadoop/etc/hadoop 目录下，需要修改的配置文件为如下两个:

/usr/local/hadoop/etc/hadoop/core-site.xml

/usr/local/hadoop/etc/hadoop/hdfs-site.xml

编辑 core-site.xml，修改<configuration></configuration>节点的内容为如下所示：

##### 示例代码：/usr/local/hadoop/etc/hadoop/core-site.xml

<configuration>

<property>

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

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

<description>location to store temporary files</description>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

同理，编辑 hdfs-site.xml，修改<configuration></configuration>节点的内容为如下所示：

##### 示例代码：/usr/local/hadoop/etc/hadoop/hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

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

</property>

<property>

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

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

</property>

</configuration>

### 格式化 NameNode

格式化NameNode:

/usr/local/hadoop/bin/hdfs namenode -format

### 启动 NameNode 和 DataNode 守护进程

启动 NameNode 和 DataNode 进程:

/usr/local/hadoop/sbin/start-dfs.sh

执行过程中会提示输入用户密码，输入 root 用户密码即可

启动进程时可能会提示出现以下错误：

19/07/10 22:11:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

提示hadoop不能加载本地库

使用命令 hadoop checknative -a查看，发现一个库都没有

[root@192 local]# hadoop checknative -a

19/07/10 22:01:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Native library checking:

hadoop: false

zlib: false

snappy: false

lz4: false

bzip2: false

openssl: false

* 修改 /etc/profile，增加以下内容：

|  |  |
| --- | --- |
| 1 2 | export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=${HADOOP\_PREFIX}/lib/native export HADOOP\_OPTS="-Djava.library.path=${HADOOP\_HOME}/lib/native" |

source /etc/profile

重新运行

[root@192 ~]# /usr/local/hadoop/sbin/start-dfs.sh

[root@192 ~]# hadoop checknative -a 查看

19/07/10 22:14:59 INFO bzip2.Bzip2Factory: Successfully loaded & initialized native-bzip2 library system-native

19/07/10 22:14:59 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library

Native library checking:

hadoop: true /usr/local/hadoop/lib/native/libhadoop.so.1.0.0

zlib: true /lib64/libz.so.1

snappy: true /lib64/libsnappy.so.1

lz4: true revision:99

bzip2: true /lib64/libbz2.so.1

openssl: false Cannot load libcrypto.so (libcrypto.so: 无法打开共享对象文件: 没有那个文件或目录)!

已经有了

检查 NameNode 和 DataNode 是否正常启动:

jps

如果NameNode和DataNode已经正常启动，会显示NameNode、DataNode和SecondaryNameNode的进程信息:

[root@192 ~]# jps

9875 Jps

4727 DataNode

4633 NameNode

4923 SecondaryNameNode

## 运行 Hadoop 伪分布式实例

Hadoop自带了丰富的例子，包括 wordcount、grep、sort 等。下面我们将以grep例子为教程，输入一批文件，从中筛选出符合正则表达式 dfs[a-z.]+ 的单词并统计出现的次数。

### 查看 Hadoop 自带的例子

Hadoop 附带了丰富的例子, 执行下面命令可以查看：

[root@192 ~]# cd /usr/local/hadoop/

[root@192 hadoop]# ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar