

# 大数据相关作业文档

学号:20144675

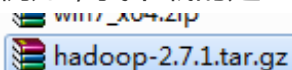
姓名:罗志翔

## 部署Hadoop平台

### 单机版Hadoop部署过程：

将下载好的hadoop文件直接解压即可完成Hadoop单机版的部署。

例如，我下载的是Hadoop2.7.1版



进入Linux系统。

注意需要配置好JAVA\_HOME

我将文件解压至 `/opt/` 目录下：命令为：

```
tar zxvf hadoop-2.7.1.tar.gz -C /opt/
```

即完成了单机版部署过程。

### 伪分布式Hadoop部署过程：

#### 1.第一步修改 `core-site.xml` 文件。

`core-site.xml` 文件位于/hadoop 文件夹下的/etc/hadoop文件夹中，我的是：

```
root@kali: /opt/hadoop-2.7.1
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali: /opt/hadoop-2.7.1# ls
bin  include  libexec  NOTICE.txt  sbin
etc  lib      LICENSE.txt  README.txt  share
root@kali: /opt/hadoop-2.7.1#
```

```
root@kali: /opt/hadoop-2.7.1/etc/hadoop# ls
capacity-scheduler.xml  httpfs-env.sh  mapred-env.sh
configuration.xml       httpfs-log4j.properties  mapred-queues.xml.template
container-executor.cfg  httpfs-signature.secret  mapred-site.xml.template
core-site.xml           httpfs-site.xml  slaves
hadoop-env.cmd          kms-acls.xml      ssl-client.xml.example
hadoop-env.sh           kms-env.sh        ssl-server.xml.example
hadoop-metrics2.properties  kms-log4j.properties  yarn-env.cmd
hadoop-metrics.properties  kms-site.xml       yarn-env.sh
hadoop-policy.xml        log4j.properties   yarn-site.xml
hdfs-site.xml            mapred-env.cmd
```

```
root@kali: /opt/hadoop-2.7.1/etc/hadoop#
```

将内容修改为如下

```
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/opt/hadoop-2.7.1/temp</value>
    <description>Abase for other temporary directories.</description>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

选中了“hadoop-2.7.1” (含有 10 项)

29, 8 底端

## 2.修改 `hdfs-site.xml` 文件.

该文件同样位于/etc/hadoop/文件夹下。

将内容修改如下

```

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/opt/hadoop-2.7.1/temp/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/data</value>
  </property>
</configuration>
"hdfs-site.xml" 32L, 1182C 已写入

```

### 3.修改完之后到 `/bin` 目录下执行如下命令：

```
./hdfs namenode -format
```

```

root@kali: /opt/hadoop-2.7.1/bin# ./hdfs namenode -format
16/12/14 00:37:15 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = kali/127.0.1.1
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.7.1
STARTUP_MSG: classpath = /opt/hadoop-2.7.1/etc/hadoop:/opt/hadoop-2.7.1/share/hadoop/common/lib/jaxb-api-2.2.2.jar:/opt/hadoop-2.7.1/share/hadoop/common/lib/commons-httpclient-3.1.jar:/opt/hadoop-2.7.1/share/hadoop/common/lib/jettison-1.1.jar:/opt/hadoop-2.7.1/share/hadoop/common/lib/commons-io-2.4.jar:/opt/hadoop-2.7.1/share/hadoop/common/lib/

```

### 4.到 `/sbin` 目录下开启 NameNode 和 DataNode 守护进程

```
./sbin/start-dfs.sh
```

```
root@kali: ~# ./opt/hadoop-2.7.1/sbin/start-dfs.sh
```

### 5.判断是否开启成功

到 `/hadoop` 安装目录下运行如下 `jps` 命令：

```

root@kali: /opt/hadoop-2.7.1# jps
2442 SecondaryNameNode
2763 DataNode
3174 Jps

```

出现上述截图的相似内容，则说明成功开启了。

## 单机版Hadoop运行Mapreduce算例

启动服务后，输入以下代码运行word count算例：

## 1.进入bin文件夹

```
cd /opt/hadoop-2.7.1/bin
```

## 2.建立input 文件夹.

```
mkdir input
```

## 3.并建立三个文件1.txt , 2.txt , 3.txt并输入一些内容

```
vim 1.txt  
hello hadoop  
:wq  
  
vim 2.txt  
hello java  
:wq  
  
vim 3.txt  
hello world  
:wq
```

## 4.在相关服务启动后在运行的hadoop中建立文件夹

```
hadoop fs -mkdir /hadoop-2.7.1/input
```

## 5.复制文件到hadoop中

```
hadoop fs -put /opt/hadoop-2.7.1p/bin/input /opt/hadoop/input
```

## 6.运行wordcount

```
hadoop jar /opt/hadoop-2.7.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar  
wordcount /opt/hadoop-2.7.1p/input /opt/hadoop-2.7.1/output
```

## 7.查看运行结果

```
./hadoop fs -cat /opt/hadoop-2.7.1/output/part-r-00000
```

8结果产生的格式为：

```
hadoop 1
```

```
hello 3
```

```
jave 1
```

```
world 1
```