搭建SPARK开发环境

任务概述

本章任务，主要是要建立对Spark引擎的一个整体认识，宏观上了解Spark的体系结构，知道Spark的应用场景，解决了什么问题。

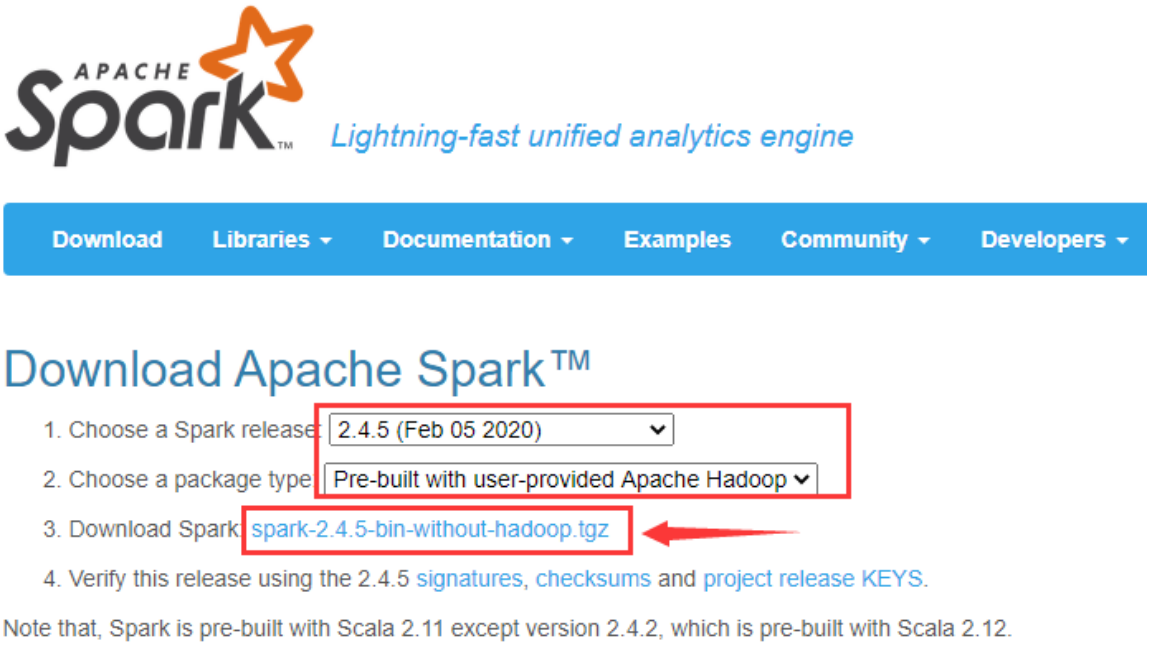
同时，还需掌握如何搭建Linux集群和Spark集群。

在基础设施建立完毕后，还需要知道如何在单机模式、伪分布式模式、集群模式上运行Spark应用。

1.下载spark

spark有提供与hadoop相对应的预版本，下载时请根据自己需要进行选择，这里我们选择不依赖于具体hadoop的spark版本，在使用时我们再根据自己hadoop版本设置相应的环境参数 。

spark目前最新的稳定版是2.4.5,版本3.0还是预览发行状态。




```
wget http://mirror.bit.edu.cn/apache/spark/spark-2.4.5/spark-2.4.5-bin-without-hadoop.tgz
```

注意：原连接使用的是https协议，下载速度可能很慢，这里改为使用http协议，速度会快很多，如果http协议被禁止不能使用，请改回https协议进行下载。

2.解压spark文件到/opt,并建立相应的soft links文件

```
tar zxvf spark-2.4.5-bin-without-hadoop.tgz
ln -sf  lspark-2.4.5-bin-without-hadoop  spark
ln -sf /opt/spark  /usr/local/spark

```

3.检查java及相关环境变量设置，这里我们仍以hadoop身份来运行spark，当然也可以单独指定一个帐号来运行，如果是单独一个运行帐号，后续在使用hdfs时，需要注意对hdfs分布式文件系统设置相应的权限。

```
java -version
rpm -qa | grep java   ##----> 1.8.0  jdk
env | grep JAVA
env | grep  HOME

```

环境变量设置
vi .bash_profile 或 vi .bashrc

```
# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java
JRE_HOME=/usr/lib/jvm/jre
export JAVA_HOME
export JRE_HOME
export HADOOP_HOME=/opt/hadoop

#export SPARK_HOME=/usr/local/spark
export SPARK_HOME=/opt/spark
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.8.1-src.zip
export PYSPARK_PYTHON=python3

PATH=$PATH:$HOME/bin:$JAVA_HOME/bin:$JAVA_JRE/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$SPARK_HOME/bin
export PATH

alias vi=vim
```

4.编辑配置文件spark-env.sh

```
cp spark-env.sh.template  spark-env.sh
vi  spark-env.sh
```

spark-env.sh内容：

```
export JAVA_HOME=/usr/lib/jvm/java
export HADOOP_CONF_DIR=/opt/hadoop/etc/hadoop
export HADOOP_HOME=/opt/hadoop/

export SPARK_MASTER_HOST=master.lab.hwadee.com
#export SPARK_MASTER_PORT=7077
#export SPARK_MASTER_WEBUI_PORT=8080
#export SPARK_WORKER_CORES=1
#export SPARK_WORKER_MEMORY=1024M

export SPARK_PID_DIR=/var/lib/spark
export SPARK_LOG_DIR=/var/log/spark
export LD_LIBRARY_PATH=$HADOOP_HOME/lib/native/:$LD_LIBRARY_PATH
export SPARK_DIST_CLASSPATH=$(${HADOOP_HOME}/bin/hadoop classpath)

```

```
15
16
```

创建PID及日志目录并修改属主

```
mkdir /var/log/spark
chown hadoop:hadoop /var/log/spark
mkdir /var/lib/spark
chown hadoop:hadoop  /var/lib/spark
```

5 .加入节点

```
vim   slaves
datanode01.lab.hwadee.com
datanode02.lab.hwadee.com
datanode03.lab.hwadee.com

```

6.同步资料到其它节点

修改文件属主为运行帐号:

```
chown  hadoop:hadoop /opt/spark* -R
```

分发spark文件到其它节点:

a.)分发修改后登录脚本

```
 scp ~haoop/.bash_profile  hadoop@datanode01.lab.hwadee.com:~/
 scp ~haoop/.bash_profile  hadoop@datanode02.lab.hwadee.com:~/
 scp ~haoop/.bash_profile  hadoop@datanode03.lab.hwadee.com:~/
```

b.) 分发spark程序到其它节点

```
 scp -r /opt/spark-2.4.5-bin-without-hadoop
hadoop@datanode01.lab.hwadee.com:/opt
 scp -r /opt/spark-2.4.5-bin-without-hadoop
hadoop@datanode01.lab.hwadee.com:/opt
 scp -r /opt/spark-2.4.5-bin-without-hadoop
hadoop@datanode01.lab.hwadee.com:/opt
```

c.)创建相关的link文件和目录

```
ssh  datanode01  "cd /opt;ln -sf /spark-2.4.5-bin-without-hadoop hadoop;ln -
sf /opt/spark /usr/local/spark"
ssh  datanode02  "cd /opt;ln -sf /spark-2.4.5-bin-without-hadoop hadoop;ln -
sf /opt/spark /usr/local/spark"
ssh  datanode03  "cd /opt;ln -sf /spark-2.4.5-bin-without-hadoop hadoop;ln -
sf /opt/spark /usr/local/spark"
```

```
ssh  datanode01  "mkdir /var/lib/spark;chown hadoop:hadoop /var/lib/spark"
ssh  datanode01  "mkdir /var/log/spark;chown hadoop:hadoop /var/log/spark"

ssh  datanode02  "mkdir /var/lib/spark;chown hadoop:hadoop /var/lib/spark"
ssh  datanode02  "mkdir /var/log/spark;chown hadoop:hadoop /var/log/spark"

ssh  datanode03  "mkdir /var/lib/spark;chown hadoop:hadoop /var/lib/spark"
ssh  datanode03  "mkdir /var/log/spark;chown hadoop:hadoop /var/log/spark"
```

7.启动spark

```
/opt/spark/sbin/start-all.sh
```

8.测试验证:

put 文件到hdfs

vi 123.txt

```
aaaa
bbb
dd
ddd
cc
sss   sss

```

vi 456.txt

```
1111
2222
3333
4444
5555
```

```
hadoop  fs  -put  123.txt  /
hadoop  fs  -put  456.txt  /
hadoop fs -ls /
```

9.运行 spark-shell

```
val f1 = sc.textFile("hdfs://master.lab.hwadee.com/123.txt")
val f2 = spark.read.textFile("/456.txt")
f1.count()
f2.count()
f1.first()
f2.first()
```

10.提交任务到spark

```
/opt/spark/bin/spark-submit --master spark://master.lab.hwadee.com:7077 --class org.apache.spark.examples.SparkPi  /opt/spark/examples/jars/spark-examples_2.11-2.4.5.jar 2>/dev/null
```