## 使用Yarn本地部署Hadoop

1. 先启动Hadoop，执行start-dfs.sh后，查看下进程只有SecondaryNameNode、DataNode、NameNode
2. 修改配置文件mapred-site.xml

cd /usr/local/hadoop/etc/hadoop

mv mapred-site.xml.template mapred-site.xml

将内容编辑如下

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

1. 修改配置文件yarn-site.xml，将内容编辑如下

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

1. 重新启动Hadoop，并启动Yarn

start-dfs.sh

start-yarn.sh

mr-jobhistory-dameon.sh start historyserver

开启历史服务器，才能在Web中查看任务运行情况

再查看进程，多出NodeManager和ResourceManager两个进程

如果以后在伪分布式模式下启动Hadoop的时候不想启动Yarn，务必把配置文件mapred-site.xml重命名为mapred-site.xml.template

1. 通过Yarn提交MapReduce作业

命令和以前的类似

## 使用Yarn本地部署Spark

1. 修改配置文件/usr/local/spark/conf目录下的spark-env.sh

export HADOOP\_CONF\_DIR=/usr/local/hadoop/etc/Hadoop

这样Spark就可以跑在YARN上了，也没必要启动spark的master和slaves服务，因为是靠yarn进行任务调度，所以直接提交任务即可

1. spark-shell运行在YARN上

执行spark-shell --master yarn-client

1. 可能存在的问题

由于是在虚拟机上运行，虚拟内存可能超过了设定的数值。在执行命令spark-shell --master yarn-client时可能报错

解决方法：先停止YARN服务，然后修改yarn-site.xml，增加以下内容

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

<description>Whether virtual memory limits will be enforced for containers</description>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>4</value>

<description>Ratio between virtual memory to physical memory when setting memory limits for containers</description>

</property>

重新启动YARN，执行spark-shell --master yarn-client

在浏览器中访问locahost:8088，可以看到Spark shell应用程序在运行

1. 通过yarn提交spark作业

命令类似如下

./spark-submit –master yarn  --class org.apache.spark.examples.SparkPi  ../lib/spark-examples-XXX-hadoopXXX.jar 100

注意：如果是自己编写的程序，不要在程序中将sparkcontext指定为local