Hadoop

# 一、Hadoop的HelloWord

## 1、安装hadoop，运行grep和wordcount程序

|  |
| --- |
| 0、解压，并配置环境变量  1、运行wordcount程序  hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount wcinput/ wcoutput  2、运行grep程序  hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar grep input output 'dfs[a-z.]+' |

## 2、配置伪分布式环境，HDFS系统，HDFS的操作

根据需求修改Hadoop-env.sh，配置java\_home.默认是去系统的环境变量JAVA\_HOME

vim $Hadoop\_home /etc/hadoop/core-site.xml

|  |
| --- |
| <configuration>  <!-- 指定HDFS中的NameNode的地址 -->  <property>  <name>fs.defaultFS</name>  <!—这里要注意host与ip的映射 -->  <value>hdfs://hadoop101:8020</value>  </property>  </configuration> |
|  |

vim $Hadoop\_home/etc/hadoop/hdfs-site.xml

|  |
| --- |
| <configuration>  <!-- 指定HDFS副本的数量 -->  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  </configuration> |

第一次要初始化

|  |
| --- |
| bin/hdfs namenode –format |

启动namenode

|  |
| --- |
| sbin/hadoop-daemon.sh start namenode |

启动datanode

|  |
| --- |
| sbin/hadoop-daemon.sh start datanode |

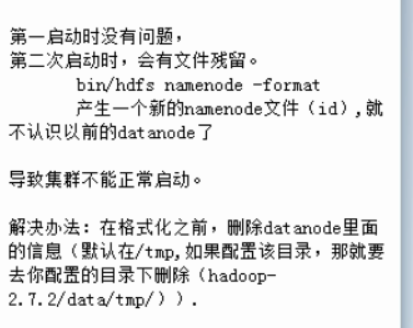
jps可查看namenode和datanode是否启动成功

都启动成功后，可访问<http://ip:50070> 查看web端的管理页面

## 3、HDFS操作，并在hdfs上运行mr程序

|  |
| --- |
| 1. 创建目录   hadoop fs -mkdir -p /user/hadoop/input   1. 查看目录   hadoop fs -ls -R /   1. 上传文件   hadoop fs -put wcinput/wc.input /user/hadoop/input |
| 1. 在hdfs上运行mapreduce程序（输出路径必须是没有创建的目录）   hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /user/hadoop/input/ /user/hadoop/output |

## 4、启动集群的问题



## 5、在yarn上运行mapreduce程序

1、配置$hadoop\_home/etc/yarn-env.sh中的java\_home.

2、配置$hadoop\_home/etc/ mapred-env.sh

中的java\_home.

3、配置配置$hadoop\_home/etc/ yarn-site.xml

|  |
| --- |
| <!-- reducer的获取数据的方式 -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <!-- 指定yarn的resourceManager的地址 -->  <property>  <name>yarn.resourcemanager.hostname</name>  <value>hadoop101</value>  </property> |

4、配置配置$hadoop\_home/etc/mapred-site.xml(对mapred-site.xml.tempate重命名)

|  |
| --- |
| <!--指定mr程序运行在yarn上-->  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property> |

5、集群启动

|  |
| --- |
| 命令都在hadoop\_home/sbin下  hadoop-daemon.sh start namenode  hadoop-daemon.sh start datanode  yarn-daemon.sh start resourcemanager  yarn-daemon.sh start nodemanager  查看集群 浏览器输入 ip:8088 |

6、运行

|  |
| --- |
| ./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /user/hadoop/wordcount/input /user/hadoop/wordcount/output |

## 6、修改hadoop文件存储位置

1、默认是存在/tmp下

2、修改

|  |
| --- |
| <!--修改数据存储位置-->  <property>  <name>hadoop.tmp.dir</name>  <value>/opt/module/hadoop-2.7.2/data/tmp</value>  </property> |

3、重启各node

|  |
| --- |
| 1、关闭各node,(命令在$hadoop\_home/sbin下)  hadoop-daemon.sh stop namenode  hadoop-daemon.sh stop datanode  yarn-daemon.sh stop NodeManager  yarn-daemon.sh stop nodemanager  yarn-daemon.sh stop resourcemanager  2、重启前，需要先删除之前的数据、pid和日志  rm –fr /tmp/Hadoop-root  rm -fr /tmp/hadoop-root-namenode.pid  rm –fr $HADOOP\_HOME/logs/\*  3、格式化(命令在$hadoop\_home/bin下)  hdfs namenode -formate  4、启动各node(命令在$hadoop\_home/sbin下)  hadoop-daemon.sh start namenode  hadoop-daemon.sh start datanode  yarn-daemon.sh start resourcemanager  yarn-daemon.sh start nodemanager |

## 7、配置ssh无秘登录

|  |
| --- |
| 1、root的家目录：/root  hadoop(其它用户)的家目录：/home/Hadoop  2、进入到家目录，生成秘钥（公钥、私钥）  ssh-keygen -t rsa |
| 3、将公钥拷贝到要登录的服务器  ssh-copy-id hadoop103 |
|  |

## 8、配置rsync同步工具

|  |
| --- |
| 1、语法  rsync –r v l $pdir/$fname $user@hadoop103:$pdir  命令 参数 要拷贝文件的路径/文件名 目的用户@主机：目的目录 |

## 9、rsync集群分发脚本

|  |
| --- |
| #!/bin/bash  #循环向多个ip同步目录  #1获取输入参数个数，如果没有，直接退出  pcount=$#  if((pcount==0));then  echo no args;  exit;  fi  #2获取文件名称  p1=$1  fname=`basename $p1`  echo fname=$fname  #3获取上级目录的绝对路径  pdir=`cd -P $(dirname $p1);pwd`  echo pdir =$pdir  #4获取当前用户的名称  user=`whoami`  #5循环  for((host=100;host<104;host++)); do  echo --------------------hadoop$host-------  rsync -rvl $pdir/$fname $user@hadoop$host:$pdir  done |

## 10、xcall在各分布式主机执行命令

|  |
| --- |
| #!/bin/bash  #循环向多个ip同步目录  #1获取输入参数个数，如果没有，直接退出  pcount=$#  if((pcount==0));then  echo no args;  exit;  fi  for((host=100;host<104;host++)); do  echo --------------------hadoop$host-------  ssh hadoop$host $@  done |
|  |

## 11、基于伪分布式搭建集群

### 1、配置

|  |
| --- |
| 0、需要配置的文件    1、集群内容    2、1修改 Hadoop-env.sh中的JAVA\_HOME  2、vim core-site.xml 配置namenode  <!-- 指定HDFS中的NameNode的地址 -->  <property>  <name>fs.defaultFS</name>  <!-- 这里要注意host与ip的映射 -->  <value>hdfs://hadoop102:8020</value>  </property>  <!--修改数据存储位置-->  <property>  <name>hadoop.tmp.dir</name>  <value>/opt/module/hadoop-2.7.2/data/tmp/hadoop-${user.name}</value>  </property>  3、hdfs-site.xml  <!-- 指定HDFS副本的数量 -->  <property>  <name>dfs.replication</name>  <value>3</value>  </property>  <!-- 指定HDFS 辅助节点的位置 -->  <property>  <name>dfs.namenode.secondary.http-address</name>  <value>hadoop104:50090</value>  </property>  4、vim slaves  hadoop102  hadoop103  hadoop104  5、1修改 yarn-env.sh中的JAVA\_HOME  5、vim yarn-site.xml  <!-- reducer的获取数据的方式 -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <!-- 指定yarn的resourceManager的地址 -->  <property>  <name>yarn.resourcemanager.hostname</name>  <value>hadoop103</value>  </property> |
| 6、1修改 mapred-env.sh中的JAVA\_HOME  6、vim mapred-site.xml |
| <!--指定mr程序运行在yarn上-->  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  7、使用xsync hadoop/ 将配置同步到其它主机。  并用xcall cat /opt/module/hadoop-2.7.2/etc/hadoop/slaves查看是否同步成功 |

### 2、启动集群

|  |
| --- |
| 1、第一次启动时，需要格式化namenode  在namenode(102)所在集群使用，命令 bin/hdfs namenode -format  2、启动namenode 和datanode  在namenode(102)所在集群使用，命令sbin/start-dfs.sh  3、启动yarn  在resourceManager(103)所在集群使用，命令sbin/start-yarn.sh |
|  |

3、重启集群

|  |
| --- |
| sbin/stop-yarn.sh  sbin/stop-dfs.sh  xcall -fr data/ logs/  rm -fr data/ logs/  bin/hdfs namenode -format  sbin/start-dfs.sh  sbin/start-yarn.sh |

## 12、上传文件到集群

|  |
| --- |
| 1、上传文件  hadoop fs -put input/xiaoxiong.txt /user/root/input  2、查看目录  hadoop fs –lsr /  3、查看内容  Hadoop fs –cat /user/root/input/xiao.txt  4、下载  hadoop fs –get /user/root/input/xiao.txt ./  5、数据真实存放数据的位置  配置的数据存放位置/dfs/data/current/BP-150611011-192.168.111.102-1554768263323/current/finalized/subdir0/subdir0    6、数据拼接还原  将文件所存在的多个块，重定向到一个文件里，对其进行解压，即可还原文件。  如 |
| cat blk\_1073741824 >> tmp  cat blk\_1073741825>> tmp  tar –zxvf tmp |
|  |

## 13、集群的启动方式

|  |
| --- |
| 1、各个服务组件逐一起动  hadoop-daemon.sh start | stop namenode | datanode | secondarynode  yarn-daemon.sh start | stop resourcemanager | nodemanager  2、各个模块分开启动（配置ssh是前提）常用  start-dfs.sh | stop-dfs.sh |
| start-yarn.sh | stop-yarn.sh |
| 3、全部启用(不建议使用)  start-all.sh | stop-all.sh |

# 二、HDFS文件系统（Hadoop Distributed File System）

## 1、概念

|  |
| --- |
|  |

## 2、命令

|  |
| --- |
|  |

## 3、HDFS客户端

|  |
| --- |
| 1、流程 |

## 4、eclipse 环境准备

|  |
| --- |
| 1、准备  2、代码    3、修改hadoop登录用户 |
| 或者在获取文件系统时，使用这种方式 |
|  |

## 5、Hdfs文件系统读写流程

|  |
| --- |
|  |
|  |

# 三、mapreduce程序

## 1、思想

|  |
| --- |
|  |

## 2、MapReduce进程

|  |
| --- |
|  |

## 3、MapReduce编程规范

|  |
| --- |
|  |
|  |

## 4、代码

|  |
| --- |
| 0、代码见 项目mapreduce0508 ：wordcount  1、统计单词数量 |
|  |
| 2、若本机安装了hadoop，并配置了环境变量，可在直接在eclipse中运行，调试。  运行时，右键Driver类，Run as > Run configuration ,配置运行参数 ，即输入路径和输出路径，如图： |
| 3、代码见 项目mapreduce0508 |

## 5、自定义序列化bean

|  |
| --- |
|  |

## 6、Shuffle机制

|  |
| --- |
| 1、概念 |
| 2、结构 |
| 3、mapreduce详细流程 |
|  |

## 6、分区

|  |
| --- |
| 分区，按要求输出到不同的的文件。流程上是在map之后。  根据分区来确定reduce的个数,最后输出文件的个数。不匹配会报错。    代码见 项目mapreduce0508 ： flow |

## 7、合并

|  |
| --- |
|  |

## 8、排序

|  |
| --- |
| 根据key排序  bean实现WriteComparable接口  并重写compareTo方法 |

## 9、分组

|  |
| --- |
| 1、决定输入reduce中的数据的一组数据，满足什么条件为一组 |

## 10、大量小文件的处理

|  |
| --- |
|  |

## 11、在map中加载缓存文件

|  |
| --- |
| 0、代码见 项目mapreduce0508 ：distributedcache  1、加载缓存    2、重写map的setup方法，缓存数据 |
| 3、代码见 |

## 12、自定义InputFormat

|  |
| --- |
| 1、主要处理过多小文件的输入map流程  2、代码见 项目mapreduce0508 ： wholefile  3、定义      4、driver中设置 |

## 13、自定义OutputFormat

|  |
| --- |
| 1、代码见 项目mapreduce0508 ： filter  2、代码片段 |

## 14、数据清洗,计数器

|  |
| --- |
| 1、功能：去除日志长度小于15的  2、代码见项目mapreduce0508 ： weblog  3、片段 |

## 15、总结

|  |
| --- |
| 1、map 数据预处理，封装bean  2、分区 根据需求，设置分区。决定最后输出文件个数  3、排序 对分区后（map后？）的数据 进行排序  4、合并 默认用hash合并。决定输入reduce中的一组数据。满足条件即为一组。减轻reduc的压力  5、reduc 最后的处理。 |

# 四、Yarn机制

|  |
| --- |
| 1、 |