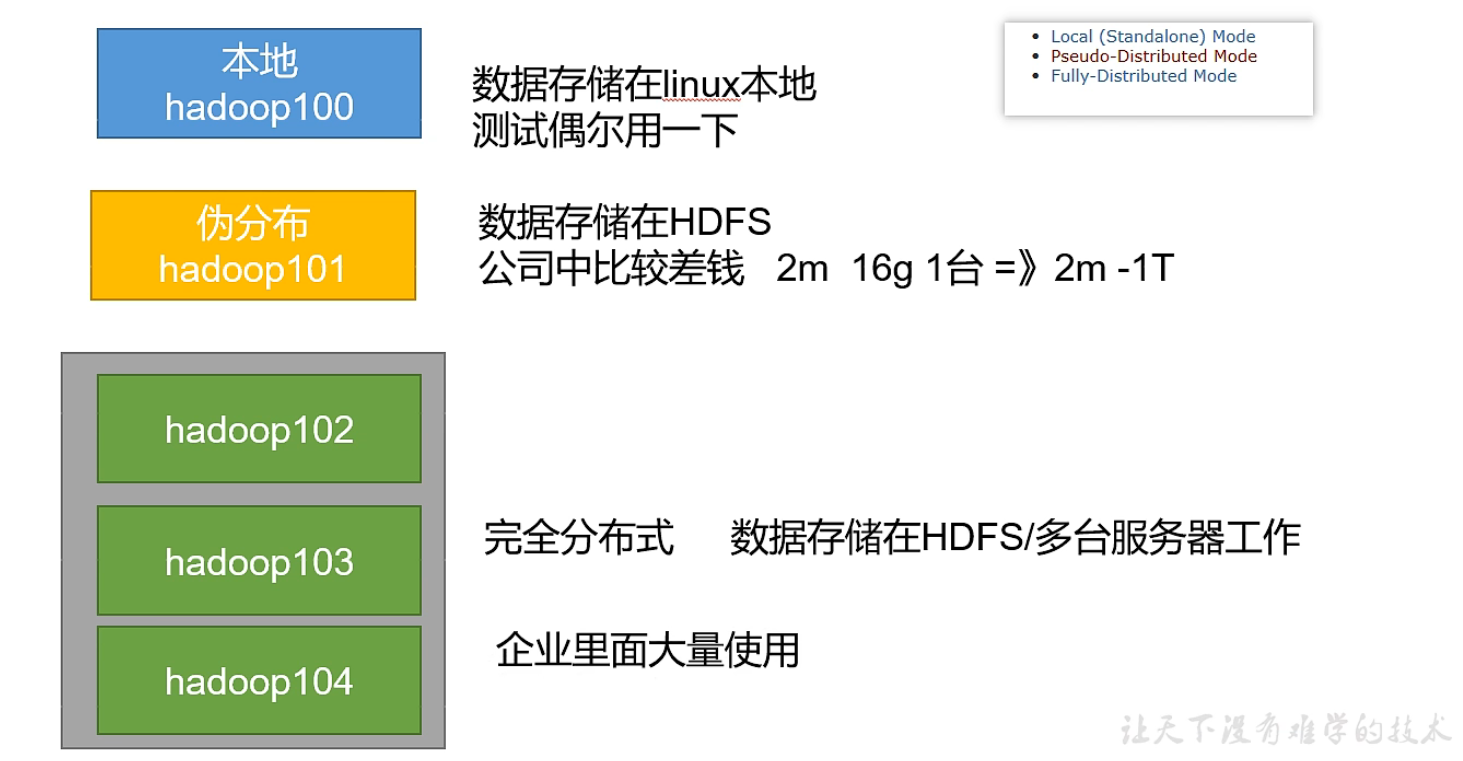
一 Hadoop部署方式有三：

本地模式部署

伪分布式部署

完全分布部署



注意！运行前注意在etc中的hadoop-env.sh中将自己的java路径附其之上

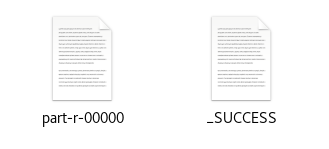
一、1 本地模式部署测试

测试hadoop本地部署，统计单词指令：

hadoop jar /opt/module/Hadoop/hadoop-3.3.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar wordcount /opt/test\_Hadoop/input/LocalMode\_test\_wordcount /opt/test\_Hadoop/output/ LocalMode\_test\_wordcount

注意：不能提前创建好输出路径，否则报错

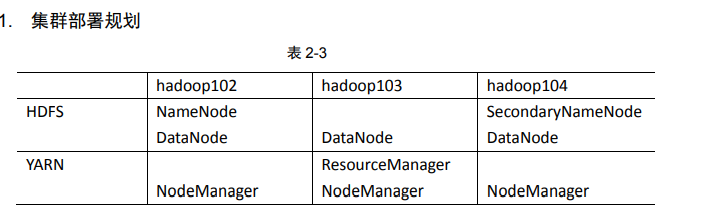
运行结果如下：



可以见得part-r-00000就是单词统计的结果

一、2 完全分布式部署

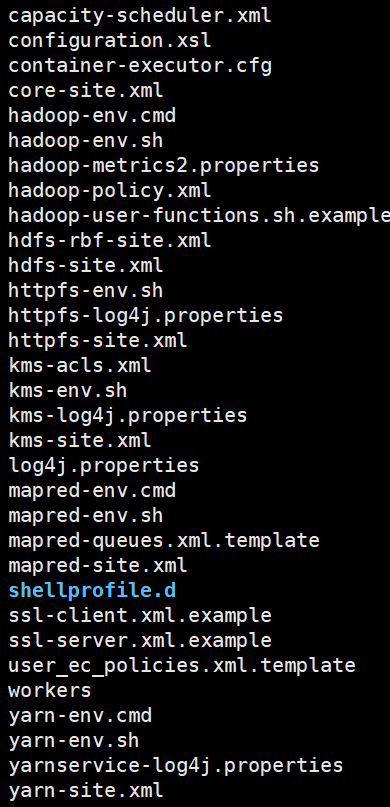
A 集群规划部署



因为NameNode与SecondaryNameNode功能相近，故不能在同一个机器上，否则失去备份意义

其次ResourceManager作为一个主要功能同样占据较多内存空间，故不将其与同样占据内存空间较多的NameNode与SecondaryNameNode放在一处

B首先配置相应文件



对etc/hadoop目录下文件内容进行分析：

a 默认配置文件：

如果从官网上下载源码后，会有：

core-default.xml

hdfs-default.xml

yarn-default.xml

mapred-default.xml

文件都在share/hadoop路径下，相应jar包中：

hadoop-common-xxx.jar/core-default.xml

hadoop-hdfs-xxx.jar/hdfs-default.xml

hadoop-yarn-common-xxx.jar/yarn-default.xml

hadoop-mapreduce-client-core-xxx.jar/mapred-default.xml

b 四大核心配置文件：(通常默认配置文件都配好了相关内容，只有需要修改其中某一默认配置值的的时候，才需要修改自定义配置文件，更改相应属性值)

1 core-site.xml

<!-- 指定 HDFS 中 NameNode 的地址 -->

<property>

<name>fs.defaultFS</name>

<value>hdfs://Hadoop102:9000</value> <!-- 可以使用8020 9020等端口，这个端口可以理解为是hadoop内部通讯所占用的端口 -->

</property>

<!-- 指定 Hadoop 运行时产生文件的存储目录 -->

<property>

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

<value>/opt/module/Hadoop/data3.3.4</value>

</property>

<!-- 配置HDFS网页登录使用的静态用户为h1,保证了在web端也可以控制删除等操作 -->

<property>

<name>hadoop.http.staticuser.user</name>

<value>h1</value>

</property>

2 hdfs-site.xml

<!-- 指定 Hadoop NameNode节点web端访问地址 -->

<property>

<name>dfs.namenode.http-address</name>

<value>hadoop102:9870</value>

</property>

<!-- 指定 Hadoop 2nn节点web端访问地址 -->

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>hadoop104:9868</value>

</property>

3 yarn-site.xml

<!-- Reducer 获取数据的方式 -->

<!-- 在默认配置里这项设置为控制，但是官方也推荐使用mapreduce\_shuffle -->

<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>

<!-- 环境变量的继承，新增HADOOP\_MAPRED\_HOME -->

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_HOME,PATH,LANG,TZ,HADOOP\_MAPRED\_HOME</value>

</property>

4 mapred-site.xml

hadoop2.0时期这个文件还是需要另存备份的，在3.0中就不需要了，直接修改即可

<!-- 指定 MR 运行在 Yarn 上, 默认运行在本地 -->

<property>

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

<value>yarn</value>

</property>

总结：

core-site.xml 部署NameNode以及hadoop存储目录

hdfs-site.xml 指定web访问地址

yarn-site.xml 指定YARN的ResourceManager地址

mapred-site.xml 指定MapReduce运行位置（yarn）

c workers

所拥有集群节点，有几个节点配置几个节点

Hadoop102

Hadoop103

Hadoop104

目前这个脚本不允许有空格，空行等…

C

a集群第一次启动需要给NameNode进行初始化：

hdfs namenode -format

如果没报错，则格式化成功，回看生成的data文件，在深层目录/opt/module/Hadoop/data/dfs/name/current会出现VERSION版本信息文件：

#Wed Dec 28 16:53:49 CST 2022

namespaceID=691931482

clusterID=CID-7a1ee2e9-ad01-4175-a2e7-e08a6e85df54

cTime=1672217629042

storageType=NAME\_NODE

blockpoolID=BP-93665123-192.168.126.102-1672217629041

layoutVersion=-66

b 启动集群

在Hadoop102 start-dfs.sh 启动NameNode

在Hadoop103 start-yarn.sh 启动ResourceManager

该指令在sbin/ 目录下

一定要注意在哪台节点启动的集群，因为我设置Hadoop103为RM，那么就要在Hadoop103启动RM

c 使用 jps 指令查看正在运行的java进程

jps（Java Virtual Machine Process Status Tool）是JDK提供的一个可以列出正在运行的Java虚拟机的进程信息的命令行工具，它可以显示Java虚拟机进程的执行主类（Main Class，main()函数所在的类）名称、本地虚拟机唯一ID（LVMID，Local Virtual Machine Identifier）等信息。另外，jps命令只能显示它有访问权限的Java进程的信息。

d 使用常用命令上传hadoop集群目录以及文件

hadoop fs -mkdir [-p] /test 创建目录

hadoop fs -put /本地文件 /hadoop集群目录 上传文件

hadoop fs -rm xxx 删除目录加-r 强制删除加-f

e 上传过后，可以见得，数据应该存储在了我们创建的data目录下，我们访问Hadoop102的data目录，可见 /opt/module/Hadoop/data/dfs/data/current 目录下同样有关于datanode 的节点信息：

#Wed Dec 28 16:56:07 CST 2022

storageID=DS-074c7006-d174-4e83-a2db-475cc415d88a

clusterID=CID-7a1ee2e9-ad01-4175-a2e7-e08a6e85df54

cTime=0

datanodeUuid=09e78466-4dad-4f54-95cc-2a666bbad7ce

storageType=DATA\_NODE

layoutVersion=-57

访问Hadoop103的data目录，可见其datanode信息：

#Wed Dec 28 16:56:07 CST 2022

storageID=DS-9b0b8de7-e095-4190-abda-e8ab9a5b0225

clusterID=CID-7a1ee2e9-ad01-4175-a2e7-e08a6e85df54

cTime=0

datanodeUuid=48eb9834-10a6-4646-9fa8-d8ecacc2b1eb

storageType=DATA\_NODE

layoutVersion=-57

继续追踪，在Hadoop102的/opt/module/Hadoop/data/dfs/data/current/BP-255517332-192.168.126.102-1672214419621/current/finalized/subdir0/subdir0目录下，发现了刚上传的文件blk\_1073741826，同样，查看Hadoop103与Hadoop104 data下相应文件目录，发现blk\_1073741826与blk\_1073741826，更加确认了hadoop集群的可靠性，上传一个，三节点俱具备了该信息！同时在第一个文件上传时，生成了每个节点**不同的**datanode的相关信息。

此外，还有一个神奇的现象：

当我将data目录的生成文件修改为dataxxx(即非data字符时)，格式化后并没有为我只创建一个dataxxx，并且附生成了一个data目录，其下只有一个 nm-local-dir 空目录，尚不知有何作用，而dataxxx中仍有一个 nm-local-dir 空目录，并且我的namenode与datanode均保存在dataxxx中。

配置好以后可以测试一下yarn的调度功能了！

同样测试wordcount功能：hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar wordcount /test/input/LocalMode\_test\_wordcount /test/output/wordcount\_20221228 要注意不要提前创建目录

如果出现这种报错，检查一下自己的yarn-site.xml文件，其中关于环境变量的配置是不允许出现空格等非法字符的！

D 重新初始化hadoop集群：

1 停止集群运行 stop-dfs.sh

2 删除每个集群的date/和logs/ 目录（就是数据存放目录以及日志存放目录）

3 重新执行格式化命令

为什么要删除data目录？因为在data目录中，存储着namenode与datanode 的版本号与相应信息，在集群启动的过程中，namenode会检测datanode的版本信息是否与自身相匹配，不匹配则无法正常启动

E 配置历史服务器

当我们想点击yarn中单个任务的history时，会显示无法加载界面，这是由于我们没有配置历史服务器

在mapred-site.xml新增如下配置：

<!-- 历史服务器端地址 -->

<property>

<name>mapreduce.jobhistory.address</name>

<value>Hadoop102:10020</value>

</property>

<!-- 历史服务器web端地址 -->

<property>

<name> mapreduce.jobhistory.webapp.address</name>

<value>Hadoop102:19888</value>

</property>

之后手动启动 历史服务器

mapred –daemon start historyserver

jps 会增加新进程 JobHistoryServer

心得：如果我再Hadoop103启动历史服务器，则集群web显示history会失败，因为我在102配置的服务器，所以只能在102开启历史服务器

F 配置日志聚集功能

具体下就是将每个节点的logs目录汇集到web显示端

在yarn-site.xml文件新增如下配置：

<!-- 开启日志聚集功能，默认关闭 -->

<property>

<name>yarn.log-aggregation-enable</name>

<value>true</value>

</property>

<!-- 设置日志聚集服务器地址，与历史服务器web端地址相同 -->

<property>

<name>yarn.log.server.url</name>

<value>[http://Hadoop102:19888/jobhistory/logs</value](http://Hadoop102:19888/jobhistory/logs%3c/value)>

</property>

<!-- 设置日志保留时间为7天 -->

<property>

<name>yarn.log-aggregation.retain-seconds</name>

<value>604800</value>

</property>

单独停止某个节点指令：

mapred --daemon stop hostoryserver

yarn/dfs/mapred/.. –daemon start/stop xxxx

注意：比如我配置日志聚集功能，只影响到了yarn以及historyserver。那么我没必要重启集群，只需要重启yarn以及历史服务器即可。

G 集群启动脚本 ctrlhadoop.sh

#!/bin/bash

if [ $# -lt 1 ]

then

echo "No Args Input..."

exit ;

fi

case $1 in

"start")

echo " =================== 启动 hadoop集群 ==================="

echo " --------------- 启动 hdfs ---------------"

ssh hadoop102 "/opt/module/hadoop-3.1.3/sbin/start-dfs.sh"

echo " --------------- 启动 yarn ---------------"

ssh hadoop103 "/opt/module/hadoop-3.1.3/sbin/start-yarn.sh"

echo " --------------- 启动 historyserver ---------------"

ssh hadoop102 "/opt/module/hadoop-3.1.3/bin/mapred --daemon start historyserver"

;;

"stop")

echo " =================== 关闭 hadoop集群 ==================="

echo " --------------- 关闭 historyserver ---------------"

ssh hadoop102 "/opt/module/hadoop-3.1.3/bin/mapred --daemon stop historyserver"

echo " --------------- 关闭 yarn ---------------"

ssh hadoop103 "/opt/module/hadoop-3.1.3/sbin/stop-yarn.sh"

echo " --------------- 关闭 hdfs ---------------"

ssh hadoop102 "/opt/module/hadoop-3.1.3/sbin/stop-dfs.sh"

;;

\*)

echo "Input Args Error..."

;;

esac

H 查看所有集群启动情况脚本 jpsall.sh

# !/bin/bash

for host in hadoop102 hadoop103 hadoop104

do

echo ==================$host===================

ssh $host jps

done

I 特殊引出：常见的面试题之

1 常用端口号

hadoop3.x

HDFS NameNode 内部通讯端口：8020/9000/9820

HDFS NameNode 对用户的查询接口：9870

YARN查看任务运行情况：8088

历史服务器：19888

hadoop2.x

HDFS NameNode 内部通讯端口：8020/9000

HDFS NameNode 对用户的查询接口：50070

YARN查看任务运行情况：8088

历史服务器：19888

2 常用配置文件

hadoop3.x

core-site.xml

hdfs-site.xml

yarn-site.xml

mapred-site.xml

workers

hadoop2.x

core-site.xml

hdfs-site.xml

yarn-site.xml

mapred-site.xml

slaves (因为slaves-奴隶含有歧视的意思)

J 所有集群时间同步解决

因为在任务运行过程中，有时需要将任务定时启动，故所有集群节点时间必须相同(事实上不同步会报错，所以必须同步)

情况一：所有节点均能连接外网，则不需要进行时间同步

情况二：部分节点不能连接外网，则需要进行时间同步

1 检查ntp 是否安装

rpm -qa | grep ntp

NTP 是一种通过网络在计算机之间进行时钟同步的协议，它工作在 OSI 模型的应用层，通过一系列原理与算法，实现以极小的误差，将所有网络中的计算机与 UTC 同步。

2 修改ntp 配置文件

vim /etc/ntp.conf

修改内容如下

a）修改 1：（授权 192.168.1.0-192.168.1.255 网段上的所有机器可以从这台机器上查

询和同步时间）

#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap 为

restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

b）修改 2：（集群在局域网中，不使用其他互联网上的时间）

server 0.centos.pool.ntp.org iburst

server 1.centos.pool.ntp.org iburst

server 2.centos.pool.ntp.org iburst

server 3.centos.pool.ntp.org iburst 为

#server 0.centos.pool.ntp.org iburst

#server 1.centos.pool.ntp.org iburst

#server 2.centos.pool.ntp.org iburst

#server 3.centos.pool.ntp.org iburst

c）添加 3：（当该节点丢失网络连接，依然可以采用本地时间作为时间服务器为集群

中的其他节点提供时间同步）

server 127.127.1.0

fudge 127.127.1.0 stratum 10

3 修改/etc/sysconfig/ntpd 文件（这一步可配置可不配置）

vim /etc/sysconfig/ntpd

增加内容如下（让硬件时间与系统时间一起同步）

SYNC\_HWCLOCK=yes

4 查看 ntpd的运行情况并且将其开启

查看情况：

service ntpd status

开启服务：

service ntpd start

5 其它节点配置：

关闭所有节点的ntp服务和开机自启

systemctl stop ntpd

systemctl disable ntpd

6 编写linux 定时任务

crontab -e

每60min同步一次时间

\*/60 \* \* \* \* /usr/sbin/ntpdate hadoop102

副：当内存满时，可能开启安全模式

查看内存命令：df -TH

关闭安全模式命令：hadoop dfsadmin -safemode leave