

四 Hadoop 运行模式

Hadoop 运行模式包括：本地模式、伪分布式模式以及完全分布式模式。

Hadoop 官方网站：<http://hadoop.apache.org/>

4.1 本地运行模式

4.1.1 官方 grep 案例

- 1) 创建在 hadoop-2.7.2 文件下面创建一个 input 文件夹

```
[atguigu@hadoop101 hadoop-2.7.2]$ mkdir input
```

- 2) 将 hadoop 的 xml 配置文件复制到 input

```
[atguigu@hadoop101 hadoop-2.7.2]$ cp etc/hadoop/*.xml input
```

- 3) 执行 share 目录下的 mapreduce 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hadoop jar
```

```
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar grep input output 'dfs[a-z.]+'
```

- 4) 查看输出结果

```
[atguigu@hadoop101 hadoop-2.7.2]$ cat output/*
```

4.1.2 官方 wordcount 案例

- 1) 创建在 hadoop-2.7.2 文件下面创建一个 wcinput 文件夹

```
[atguigu@hadoop101 hadoop-2.7.2]$ mkdir wcinput
```

- 2) 在 wcinput 文件下创建一个 wc.input 文件

```
[atguigu@hadoop101 hadoop-2.7.2]$ cd wcinput
```

```
[atguigu@hadoop101 wcinput]$ touch wc.input
```

- 3) 编辑 wc.input 文件

```
[atguigu@hadoop101 wcinput]$ vi wc.input
```

在文件中输入如下内容

```
hadoop yarn
```

```
hadoop mapreduce
```

```
atguigu
```

```
atguigu
```

```
保存退出:: wq
```

4) 回到 `hadoop` 目录 `/opt/module/hadoop-2.7.2`

5) 执行程序:

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop jar
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount wcinput wcoutput
```

6) 查看结果:

```
[atguigu@hadoop101 hadoop-2.7.2]$ cat wcoutput/part-r-000000
atguigu 2
hadoop 2
mapreduce 1
yarn 1
```

4.2 伪分布式运行模式

4.2.1 启动 HDFS 并运行 MapReduce 程序

1) 分析:

- (1) 配置集群
- (2) 启动、测试集群增、删、查
- (3) 执行 `wordcount` 案例

2) 执行步骤

(1) 配置集群

(a) 配置: `hadoop-env.sh`

Linux 系统中获取 `jdk` 的安装路径:

```
[atguigu@hadoop101 ~]# echo $JAVA_HOME
/opt/module/jdk1.8.0_144
```

修改 `JAVA_HOME` 路径:

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

(b) 配置: `core-site.xml`

```
<!-- 指定 HDFS 中 NameNode 的地址 -->
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://hadoop101:9000</value>
</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-2.7.2/data/tmp</value>
</property>
```

(c) 配置: hdfs-site.xml

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

(2) 启动集群

(a) 格式化 NameNode (第一次启动时格式化, 以后就不要总格式化)

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs namenode -format
```

(b) 启动 NameNode

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start namenode
```

(c) 启动 DataNode

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start datanode
```

(3) 查看集群

(a) 查看是否启动成功

```
[atguigu@hadoop101 hadoop-2.7.2]$ jps
```

```
13586 NameNode
```

```
13668 DataNode
```

```
13786 Jps
```

(b) 查看产生的 log 日志

当前目录: /opt/module/hadoop-2.7.2/logs

```
[atguigu@hadoop101 logs]$ ls
```

```
hadoop-atguigu-datanode-hadoop.atguigu.com.log
```

```
hadoop-atguigu-datanode-hadoop.atguigu.com.out
```

```
hadoop-atguigu-namenode-hadoop.atguigu.com.log
```

```
hadoop-atguigu-namenode-hadoop.atguigu.com.out
```

```
SecurityAuth-root.audit
```

```
[atguigu@hadoop101 logs]# cat hadoop-atguigu-datanode-hadoop101.log
```

(c) web 端查看 HDFS 文件系统

<http://192.168.1.101:50070/dfshealth.html#tab-overview>

注意：如果不能查看，看如下帖子处理

<http://www.cnblogs.com/zlsich/p/6604189.html>

(4) 操作集群

(a) 在 hdfs 文件系统上**创建**一个 input 文件夹

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -mkdir -p /user/atguigu/input
```

(b) 将测试文件内容**上传**到文件系统上

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -put wcinput/wc.input  
/user/atguigu/input/
```

(c) **查看**上传的文件是否正确

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -ls /user/atguigu/input/  
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -cat /user/atguigu/  
input/wc.input
```

(d) 运行 mapreduce 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hadoop jar  
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /user/atguigu/input/  
/user/atguigu/output
```

(e) 查看输出结果

命令行查看：

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -cat /user/atguigu/output/*
```

浏览器查看

Browse Directory

/user/atguigu/output							Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	atguigu	supergroup	0 B	2017/12/1 上午11:05:18	1	128 MB	._SUCCESS
-rw-r--r--	atguigu	supergroup	38 B	2017/12/1 上午11:05:18	1	128 MB	part-r-00000

(f) 将测试文件内容**下载**到本地

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop fs -get /user/atguigu/  
output/part-r-00000 ./wcoutput/
```

(g) **删除**输出结果

```
[atguigu@hadoop101 hadoop-2.7.2]$ hdfs dfs -rm -r /user/atguigu/output
```

4.2.2 YARN 上运行 MapReduce 程序

1) 分析:

- (1) 配置集群 yarn 上运行
- (2) 启动、测试集群增、删、查
- (3) 在 yarn 上执行 wordcount 案例

2) 执行步骤

(1) 配置集群

(a) 配置 yarn-env.sh

配置一下 JAVA_HOME

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

(b) 配置 yarn-site.xml

```
<!-- reducer 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>
```

(c) 配置: mapred-env.sh

配置一下 JAVA_HOME

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

(d) 配置: (对 mapred-site.xml.template 重新命名为) mapred-site.xml

```
[atguigu@hadoop101 hadoop]$ mv mapred-site.xml.template mapred-site.xml
```

```
[atguigu@hadoop101 hadoop]$ vi mapred-site.xml
```

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

(2) 启动集群

(a) 启动前必须保证 namenode 和 datanode 已经启动

(b) 启动 resourcemanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start resourcemanager
```

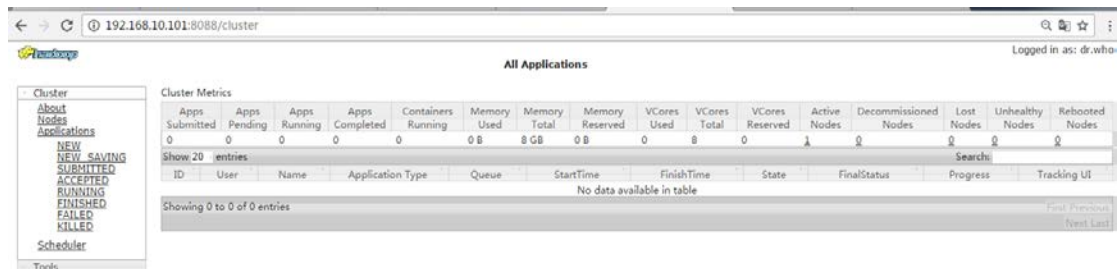
(c) 启动 nodemanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start nodemanager
```

(3) 集群操作

(a) yarn 的浏览器页面查看

<http://192.168.1.101:8088/cluster>



(b) 删除文件系统上的 output 文件

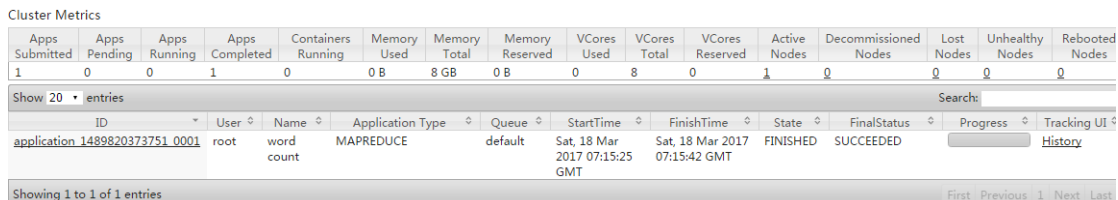
```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -rm -R /user/atguigu/output
```

(c) 执行 mapreduce 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hadoop jar
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /user/atguigu/input
/user/atguigu/output
```

(d) 查看运行结果

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -cat /user/atguigu/output/*
```



4.2.3 配置历史服务器

1) 配置 mapred-site.xml

```
[atguigu@hadoop101 hadoop]$ vi mapred-site.xml
```

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

```
<value>hadoop101:10020</value>
</property>
<property>
  <name>mapreduce.jobhistory.webapp.address</name>
  <value>hadoop101:19888</value>
</property>
```

2) 查看启动历史服务器文件目录:

```
[atguigu@hadoop101 hadoop-2.7.2]$ ls sbin/ | grep mr
mr-jobhistory-daemon.sh
```

3) 启动历史服务器

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/mr-jobhistory-daemon.sh start historyserver
```

4) 查看历史服务器是否启动

```
[atguigu@hadoop101 hadoop-2.7.2]$ jps
```

5) 查看 jobhistory

<http://192.168.1.101:19888/jobhistory>

4.2.4 配置日志的聚集

日志聚集概念: 应用运行完成以后, 将日志信息上传到 HDFS 系统上。

开启日志聚集功能步骤:

(1) 配置 yarn-site.xml

```
[atguigu@hadoop101 hadoop]$ vi yarn-site.xml
```

```
<!-- 日志聚集功能使能 -->
<property>
  <name>yarn.log-aggregation-enable</name>
  <value>true</value>
</property>
<!-- 日志保留时间设置 7 天 -->
<property>
  <name>yarn.log-aggregation.retain-seconds</name>
  <value>604800</value>
</property>
```

(2) 关闭 nodemanager 、 resourcemanager 和 historymanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh stop resourcemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh stop nodemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/mr-jobhistory-daemon.sh stop historyserver
```

(3) 启动 nodemanager 、resourcemanager 和 historymanager

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start resourcemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/yarn-daemon.sh start nodemanager
```

```
[atguigu@hadoop101 hadoop-2.7.2]$ sbin/mr-jobhistory-daemon.sh start historyserver
```

(4) 删除 hdfs 上已经存在的 hdfs 文件

```
[atguigu@hadoop101 hadoop-2.7.2]$ bin/hdfs dfs -rm -R /user/atguigu/output
```

(5) 执行 wordcount 程序

```
[atguigu@hadoop101 hadoop-2.7.2]$ hadoop jar
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar
wordcount
/user/atguigu/input /user/atguigu/output
```

(6) 查看日志

<http://192.168.1.101:19888/jobhistory>

192.168.10.101:19888/jobhistory

JobHistory

Retired Jobs

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
2017.03.18 17:54:46 CST	2017.03.18 17:54:51 CST	2017.03.18 17:55:01 CST	job_1489830500161_0001	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 17:20:31 CST	2017.03.18 17:20:39 CST	2017.03.18 17:20:50 CST	job_1489827711073_0001	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 16:21:38 CST	2017.03.18 16:21:42 CST	2017.03.18 16:21:52 CST	job_1489820373751_0003	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 15:29:57 CST	2017.03.18 15:30:02 CST	2017.03.18 15:30:12 CST	job_1489820373751_0002	word count	root	default	SUCCEEDED	1	1	1	1
2017.03.18 15:15:25 CST	2017.03.18 15:15:31 CST	2017.03.18 15:15:42 CST	job_1489820373751_0001	word count	root	default	SUCCEEDED	1	1	1	1

MapReduce Job job_1489830500161_0001

Job Overview

Job Name: word count
User Name: root
Queue: default
State: SUCCEEDED
Uberized: false
Submitted: Sat Mar 18 17:54:46 CST 2017
Started: Sat Mar 18 17:54:51 CST 2017
Finished: Sat Mar 18 17:55:01 CST 2017
Elapsed: 9sec
Diagnostics:
Average Map Time: 2sec
Average Shuffle Time: 2sec
Average Merge Time: 0sec
Average Reduce Time: 0sec

ApplicationMaster	Attempt Number	Start Time	Node	Logs
1		Sat Mar 18 17:54:49 CST 2017	hadoop.atguigu.com:8042	logs

Task Type	Total	Complete
Map	1	1
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1



```
Log Type: stderr
Log Length: 222
log4j:WARN No appenders could be found for logger (org.apache.hadoop.ipc.Server).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#mconfig for more info.

Log Type: stdout
Log Length: 312
Java HotSpot(TM) Server VM warning: You have loaded library /opt/module/hadoop-2.5.0/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c libfile', or link it with '-z noexecstack'.

Log Type: syslog
Log Length: 34561
Showing 4096 bytes of 34561 total. Click here for the full log.
rr.JobHistoryEventHandler: Copying hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/root/.staging/job_1489830500161_0001/job_1489830500161_0001_1.jhist to hdfs://hadoop.atguigu.com:8020/tmp/hadoop
2017-03-18 17:55:02,058 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Copied to done location: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/
2017-03-18 17:55:02,060 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Copying hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/root/.staging/job_1489830
2017-03-18 17:55:02,082 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Copied to done location: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/
2017-03-18 17:55:02,086 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Moved tmp to done: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/done_i
2017-03-18 17:55:02,088 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Moved tmp to done: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/done_i
2017-03-18 17:55:02,090 INFO [eventHandlingThread] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Moved tmp to done: hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/history/done_i
2017-03-18 17:55:02,090 INFO [Thread-64] org.apache.hadoop.mapreduce.jobhistory.JobHistoryEventHandler: Stopped JobHistoryEventHandler. super: stop()
2017-03-18 17:55:02,092 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Setting job diagnostics to
2017-03-18 17:55:02,093 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: History url is http://hadoop.atguigu.com:19888/jobhistory/job/job_1489830500161_0001
2017-03-18 17:55:02,106 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Waiting for application to be successfully unregistered
2017-03-18 17:55:03,112 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Final Stats: PendingReds:0 ScheduledMaps:0 ScheduledReds:0 AssignedMaps:0 AssignedReds:1 CompletedMap
2017-03-18 17:55:03,113 INFO [Thread-64] org.apache.hadoop.mapreduce.v2.app.MRAppMaster: Deleting staging directory hdfs://hadoop.atguigu.com:8020/tmp/hadoop-yarn/staging/root/.staging/job_148983050016
2017-03-18 17:55:03,118 INFO [Thread-64] org.apache.hadoop.ipc.Server: Stopping server on 56227
2017-03-18 17:55:03,120 INFO [IPC Server listener on 56227] org.apache.hadoop.ipc.Server: Stopping IPC Server listener on 56227
2017-03-18 17:55:03,122 INFO [TaskHeartbeatHandler PingChecker] org.apache.hadoop.mapreduce.v2.app.TaskHeartbeatHandler: TaskHeartbeatHandler thread interrupted
```

4.2.5 配置文件说明

Hadoop 配置文件分两类：默认配置文件和自定义配置文件，只有用户想修改某一默认配置值时，才需要修改自定义配置文件，更改相应属性值。

(1) 默认配置文件：存放在 hadoop 相应的 jar 包中

[core-default.xml]

hadoop-common-2.7.2.jar/ core-default.xml

[hdfs-default.xml]

hadoop-hdfs-2.7.2.jar/ hdfs-default.xml

[yarn-default.xml]

hadoop-yarn-common-2.7.2.jar/ yarn-default.xml

[mapred-default.xml]

hadoop-mapreduce-client-core-2.7.2.jar/ mapred-default.xml

(2) 自定义配置文件：存放在\$HADOOP_HOME/etc/hadoop

core-site.xml

hdfs-site.xml

yarn-site.xml

mapred-site.xml

4.3 完全分布式运行模式

分析：

1) 准备 3 台客户机（关闭防火墙、静态 ip、主机名称）

- 2) 安装 jdk
- 3) 配置环境变量
- 4) 安装 hadoop
- 5) 配置环境变量
- 6) 配置集群
- 7) 单点启动
- 8) 配置 ssh
- 9) 群起并测试集群

4.3.1 虚拟机准备

详见 3.1 章。

4.3.2 编写集群分发脚本 xsync

1) scp:secure copy 安全拷贝

(1) scp 定义:

scp 可以实现服务器与服务器之间的数据拷贝。

(2) 案例实操

(a) 将 hadoop101 中/opt/module 目录下的软件拷贝到 hadoop102 上。

```
[atguigu@hadoop101 /]$ scp -r /opt/module/* atguigu@hadoop102:/opt/module
```

(b) 将 hadoop101 服务器上的/opt/module 目录下的软件拷贝到 hadoop103 上。

```
[atguigu@hadoop103 opt]$ scp -r atguigu@hadoop101:/opt/module/*  
hadoop103:/opt/module
```

(c) 在 hadoop103 上操作将 hadoop101 中/opt/module 目录下的软件拷贝到 hadoop104 上。

```
[atguigu@hadoop103 opt]$ scp -r hadoop101:/opt/module/*  
hadoop104:/opt/module
```

2) rsync

rsync 远程同步工具，主要用于备份和镜像。具有速度快、避免复制相同内容和支持符号链接的优点。

rsync 和 scp 区别：用 rsync 做文件的复制要比 scp 的速度快，rsync 只对差异文件做更新。scp 是把所有文件都复制过去。

(1) 查看 rsync 使用说明

```
man rsync | more
```

(2) 基本语法

```
rsync -rvl      $pdir/$fname      $user@hadoop$host:$pdir
```

命令 命令参数 要拷贝的文件路径/名称 目的用户@主机:目的路径

(3) 选项说明

选项	功能
-r	递归
-v	显示复制过程
-l	拷贝符号连接

(4) 案例实操

把本机/opt/software 目录同步到 hadoop102 服务器的 root 用户下的/opt/目录

```
[atguigu@hadoop101 opt]$ rsync -rvl /opt/software/* hadoop102:/opt/software/
```

3) 脚本需求分析：循环复制文件到所有节点的相同目录下。

(1) 原始拷贝：

```
rsync -rvl      /opt/module      root@hadoop103:/opt/
```

(2) 期望脚本：

xsync 要同步的文件名称

(3) 在/home/atguigu/bin 这个目录下存放的脚本，atguigu 用户可以在系统任何地方直接执行。

4) 脚本实现：

(1) 在/home/atguigu 目录下创建 bin 目录，并在 bin 目录下 xsync 创建文件，文件内容如下：

```
[atguigu@hadoop102 ~]$ mkdir bin
```

```
[atguigu@hadoop102 ~]$ cd bin/
```

```
[atguigu@hadoop102 bin]$ touch xsync
```

```
[atguigu@hadoop102 bin]$ vi xsync
```

```
#!/bin/bash
#1 获取输入参数个数，如果没有参数，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 获取上级目录到绝对路径
pdir=`cd -P $(dirname $p1); pwd`
echo pdir=$pdir

#4 获取当前用户名称
user=`whoami`

#5 循环
for((host=103; host<105; host++)); do
    echo ----- hadoop$host -----
    rsync -rvl $pdir/$fname $user@hadoop$host:$pdir
done
```

(2) 修改脚本 xsync 具有执行权限

```
[atguigu@hadoop102 bin]$ chmod 777 xsync
```

(3) 调用脚本形式： xsync 文件名称

```
[atguigu@hadoop102 bin]$ xsync /home/atguigu/bin
```

4.3.3 集群配置

1) 集群部署规划

	hadoop102	hadoop103	hadoop104
HDFS	NameNode		SecondaryNameNode
	DataNode	DataNode	DataNode
YARN		ResourceManager	
	NodeManager	NodeManager	NodeManager

2) 配置集群

(1) 核心配置文件

```
core-site.xml

[atguigu@hadoop102 hadoop]$ vi core-site.xml
```

```
<!-- 指定 HDFS 中 NameNode 的地址 -->
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://hadoop102:9000</value>
</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-2.7.2/data/tmp</value>
</property>
```

(2) hdfs 配置文件

hadoop-env.sh

[atguigu@hadoop102 hadoop]\$ vi hadoop-env.sh

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

hdfs-site.xml

[atguigu@hadoop102 hadoop]\$ vi hdfs-site.xml

```
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>

<property>
  <name>dfs.namenode.secondary.http-address</name>
  <value>hadoop104:50090</value>
</property>
```

(3) yarn 配置文件

yarn-env.sh

[atguigu@hadoop102 hadoop]\$ vi yarn-env.sh

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

yarn-site.xml

[atguigu@hadoop102 hadoop]\$ vi yarn-site.xml

```
<!-- reducer 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>
```

(4) mapreduce 配置文件

mapred-env.sh

[atguigu@hadoop102 hadoop]\$ vi mapred-env.sh

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

mapred-site.xml

[atguigu@hadoop102 hadoop]\$ cp mapred-site.xml.template mapred-site.xml

[atguigu@hadoop102 hadoop]\$ vi mapred-site.xml

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

3) 在集群上分发配置好的 Hadoop 配置文件

[atguigu@hadoop102 hadoop]\$ xsync /opt/module/hadoop-2.7.2/

确保所传文件和文件夹都是atguigu用户的，
chown -R atguigu:hadoop /opt/module

4) 查看文件分发情况

[atguigu@hadoop103 hadoop]\$ cat /opt/module/hadoop-2.7.2/etc/hadoop/core-site.xml

4.3.4 集群单点启动

(0) 如果集群是第一次启动，需要格式化 NameNode

[atguigu@hadoop102 hadoop-2.7.2]\$ hadoop namenode -format

(1) 在 hadoop102 上启动 NameNode

[atguigu@hadoop102 hadoop-2.7.2]\$ hadoop-daemon.sh start namenode

[atguigu@hadoop102 hadoop-2.7.2]\$ jps

3461 NameNode

3531 Jps

(2) 在 hadoop102、hadoop103 以及 hadoop104 上分别启动 DataNode

[atguigu@hadoop102 hadoop-2.7.2]\$ hadoop-daemon.sh start datanode

[atguigu@hadoop102 hadoop-2.7.2]\$ jps

3461 NameNode

3608 Jps

3561 DataNode

[atguigu@hadoop103 hadoop-2.7.2]\$ hadoop-daemon.sh start datanode

[atguigu@hadoop103 hadoop-2.7.2]\$ jps

3190 DataNode

3279 Jps

[atguigu@hadoop104 hadoop-2.7.2]\$ hadoop-daemon.sh start datanode

[atguigu@hadoop104 hadoop-2.7.2]\$ jps

3237 Jps

3163 DataNode

4.3.5 SSH 无密登录配置

1) 配置 ssh

(1) 基本语法

ssh 另一台电脑的 ip 地址

(2) ssh 连接时出现 Host key verification failed 的解决方法

[atguigu@hadoop102 opt] \$ ssh 192.168.1.103

The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.

RSA key fingerprint is cf:1e:de:d7:d0:4c:2d:98:60:b4:fd:ae:b1:2d:ad:06.

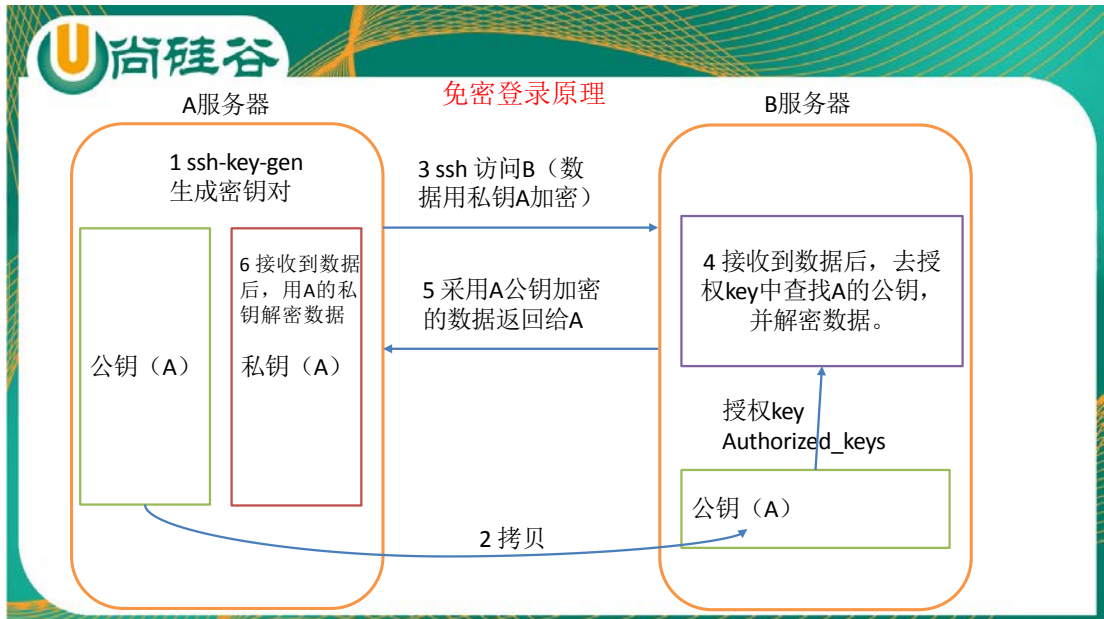
Are you sure you want to continue connecting (yes/no)?

Host key verification failed.

(3) 解决方案如下：直接输入 yes

2) 无密钥配置

(1) 免密登录原理



(2) 生成公钥和私钥:

```
[atguigu@hadoop102 .ssh]$ ssh-keygen -t rsa
```

然后敲 (三个回车)，就会生成两个文件 id_rsa (私钥)、id_rsa.pub (公钥)

(3) 将公钥拷贝到要免密登录的目标机器上

```
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop103
```

```
[atguigu@hadoop102 .ssh]$ ssh-copy-id hadoop104
```

3) .ssh 文件夹下 (~/.ssh) 的文件功能解释

(1) known_hosts : 记录 ssh 访问过计算机的公钥(public key)

(2) id_rsa : 生成的私钥

(3) id_rsa.pub : 生成的公钥

(4) authorized_keys : 存放授权过得无密登录服务器公钥

4.3.6 集群测试

1) 配置 slaves

```
/opt/module/hadoop-2.7.2/etc/hadoop/slaves
```

```
[atguigu@hadoop102 hadoop]$ vi slaves
```

```
hadoop102
hadoop103
hadoop104
```

2) 启动集群

(0) 如果集群是第一次启动，需要格式化 NameNode

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs namenode -format
```

(1) 启动 HDFS:

```
[atguigu@hadoop102 hadoop-2.7.2]$ sbin/start-dfs.sh
```

```
[atguigu@hadoop102 hadoop-2.7.2]$ jps
```

4166 NameNode

4482 Jps

4263 DataNode

```
[atguigu@hadoop103 hadoop-2.7.2]$ jps
```

3218 DataNode

3288 Jps

```
[atguigu@hadoop104 hadoop-2.7.2]$ jps
```

3221 DataNode

3283 SecondaryNameNode

3364 Jps

(2) 启动 yarn

```
[atguigu@hadoop103 hadoop-2.7.2]$ sbin/start-yarn.sh
```

注意: NameNode 和 ResourceManger 如果不是同一台机器，不能在 NameNode 上启动 yarn，应该在 ResouceManager 所在的机器上启动 yarn。

(3) web 端查看 SecondaryNameNode

(a) 浏览器中输入: <http://hadoop104:50090/status.html>

(b) 查看 SecondaryNameNode 信息。

hadoop104:50090/status.html

Hadoop
Overview

Overview

Version	2.7.2
Compiled	2017-05-22T10:49Z by root from Unknown
NameNode Address	hadoop102:9000
Started	2017/12/11 上午6:01:48
Last Checkpoint	Never
Checkpoint Period	3600 seconds
Checkpoint Transactions	1000000

Checkpoint Image URI

- file:///opt/module/hadoop-2.7.2/data/tmp/dfs/namesecondary

Checkpoint Editlog URI

- file:///opt/module/hadoop-2.7.2/data/tmp/dfs/namesecondary

Hadoop, 2015.

3) 集群基本测试

(1) 上传文件到集群

上传小文件

```
[atguigu@hadoop102 hadoop-2.7.2]$ hadoop fs -mkdir -p /user/atguigu/input
```

```
[atguigu@hadoop102 hadoop-2.7.2]$ hadoop fs -put wcinput/wc.input /user/atguigu/input
```

上传大文件

```
[atguigu@hadoop102 hadoop-2.7.2]$ bin/hadoop fs -put /opt/software/hadoop-2.7.2.tar.gz /user/atguigu/input
```

(2) 上传文件后查看文件存放在什么位置

查看 HDFS 文件存储路径

```
[atguigu@hadoop102 subdir0]$ pwd
/opt/module/hadoop-2.7.2/data/tmp/dfs/data/current/BP-938951106-192.168.10.107-1495462844069/current/finalized/subdir0/subdir0
```

查看 HDFS 在磁盘存储文件内容

```
[atguigu@hadoop102 subdir0]$ cat blk_1073741825
```

hadoop yarn

hadoop mapreduce

atguigu

atguigu

(3) 拼接

```
-rw-rw-r--. 1 atguigu atguigu 134217728 5 月 23 16:01 blk_1073741836
-rw-rw-r--. 1 atguigu atguigu 1048583 5 月 23 16:01 blk_1073741836_1012.meta
-rw-rw-r--. 1 atguigu atguigu 63439959 5 月 23 16:01 blk_1073741837
-rw-rw-r--. 1 atguigu atguigu 495635 5 月 23 16:01 blk_1073741837_1013.meta

[atguigu@hadoop102 subdir0]$ cat blk_1073741836>>tmp.file
[atguigu@hadoop102 subdir0]$ cat blk_1073741837>>tmp.file
[atguigu@hadoop102 subdir0]$ tar -zxvf tmp.file
```

(4) 下载

```
[atguigu@hadoop102 ~]$ cd /user/atguigu/input/hadoop-2.7.2
[hadoop102 ~]$ bin/hadoop fs -get /user/atguigu/input/hadoop-2.7.2.tar.gz ./
```

4) 性能测试集群

- (1) 写海量数据
- (2) 读海量数据

4.3.7 集群启动/停止方式

1) 各个服务组件逐一启动/停止

(1) 分别启动/停止 hdfs 组件

```
hadoop-daemon.sh start|stop namenode|datanode|secondarynamenode
```

(2) 启动/停止 yarn

```
yarn-daemon.sh start|stop resourcemanager|nodemanager
```

2) 各个模块分开启动/停止（配置 ssh 是前提）常用

(1) 整体启动/停止 hdfs

```
start-dfs.sh
```

```
stop-dfs.sh
```

(2) 整体启动/停止 yarn

```
start-yarn.sh
```

```
stop-yarn.sh
```

3) 全部启动/停止集群（不建议使用）

```
start-all.sh
```

```
stop-all.sh
```

4.3.8 集群时间同步

时间同步的方式：找一个机器，作为时间服务器，所有的机器与这台集群时间进行定时的同步，比如，每隔十分钟，同步一次时间。

配置时间同步实操：

1) 时间服务器配置（必须 root 用户）

(1) 检查 ntp 是否安装

```
[root@hadoop102 桌面]# rpm -qa|grep ntp  
ntp-4.2.6p5-10.el6.centos.x86_64  
fontpackages-filesystem-1.41-1.1.el6.noarch  
ntpdate-4.2.6p5-10.el6.centos.x86_64
```

(2) 修改 ntp 配置文件

```
[root@hadoop102 桌面]# vi /etc/ntp.conf
```

修改内容如下

a) 修改 1（设置本地网络上的主机不受限制。）

```
#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（设置为不采用公共的服务器）

```
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（添加默认的一个内部时钟数据，使用它为局域网用户提供服务。）

server 127.127.1.0

fudge 127.127.1.0 stratum 10

(3) 修改/etc/sysconfig/ntpd 文件

```
[root@hadoop102 桌面]# 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) 重新启动 ntpd

```
[root@hadoop102 桌面]# service ntpd status
```

ntpd 已停

```
[root@hadoop102 桌面]# service ntpd start
```

正在启动 ntpd:

[确定]

(5) 执行:

```
[root@hadoop102 桌面]# chkconfig ntpd on
```

2) 其他机器配置（必须 root 用户）

(1) 在其他机器配置 10 分钟与时间服务器同步一次

```
[root@hadoop103 hadoop-2.7.2]# crontab -e
```

编写脚本

```
*/10 * * * * /usr/sbin/ntpdate hadoop102
```

(2) 修改任意机器时间

```
[root@hadoop103 hadoop]# date -s "2017-9-11 11:11:11"
```

(3) 十分钟后查看机器是否与时间服务器同步

```
[root@hadoop103 hadoop]# date
```