

数据一 课堂笔记

目录

数据一 课堂笔记.....	1
----- day01 -----.....	2
一、主机配置命令.....	2
二、liunix 基本结构介绍.....	2
三、修改配置虚拟机网络.....	3
----- day02 -----.....	4
一、Liunix 的常用命令.....	4
二、linux 定时任务.....	5
----- day03 -----.....	6
一、ntp 服务器时间同步 - 需要关闭防火墙.....	6
二、安装 Java 环境变量.....	7
三、安装 Hadoop 环境变量.....	8
四、克隆虚拟机.....	10
五、配置 ssh 免密登录.....	10
六、格式化 HDFS.....	11
----- day04 -----.....	12
一、从代码层看 hadoop.....	12
二、启动 hadoop.....	12
三、block 的概念.....	12
四、HDFS shell 命令.....	12
----- day05 -----.....	14
—————周总结考核.....	14
—————构建 Maven 项目.....	15
一、HDFS 基本架构关系图解.....	16
二、HDFS 文件读写过程图解.....	16
hdfs 流式写入过程.....	17
hdfs 文件读操作.....	18
三、HDFS 基础理论知识.....	19

----- day01 -----

一、主机配置命令

一、修改主机名 `vi /etc/sysconfig/network`

二、修改 ip 地址映射 `vi /etc/hosts`

三、`ctrl + c` 结束当前终端的前台进程

四、重启 `reboot` 、`init 6`

五、关机 `halt` 、`shutdown -h now` 、`init 0`

六、`tab` 键 ----> 命令的补全和提示功能

`space` 向下一屏

`b` 向上一屏

`d` 向下半屏

`u` 向上半屏

七、帮助命令

1. 内部命令 `shell` 自带开机会加载到内存中的

`help` 可以查看内部命令的帮助

2. 外部命令 其他应用程序提供的

`man` 可以查看外部命令的帮助

二、linux 基本结构介绍

八、Linux 目录结构

<code>dr-xr-xr-x.</code>	2	root	root	4096	May	31	04:46	bin
<code>dr-xr-xr-x.</code>	5	root	root	1024	May	31	04:47	boot
<code>drwxr-xr-x.</code>	18	root	root	3740	Jun	1	03:17	dev
<code>drwxr-xr-x.</code>	102	root	root	4096	Jun	1	03:19	etc
<code>drwxr-xr-x.</code>	3	root	root	4096	May	31	04:49	home
<code>dr-xr-xr-x.</code>	10	root	root	4096	May	31	04:41	lib
<code>dr-xr-xr-x.</code>	9	root	root	12288	May	31	04:41	lib64
<code>drwx-----.</code>	2	root	root	16384	May	31	04:35	lost+found
<code>drwxr-xr-x.</code>	2	root	root	4096	Sep	23	2011	media
<code>drwxr-xr-x.</code>	3	root	root	4096	May	31	04:50	mnt
<code>drwxr-xr-x.</code>	3	root	root	4096	May	31	04:52	opt
<code>dr-xr-xr-x.</code>	148	root	root	0	Jun	1	03:17	proc
<code>dr-xr-x---</code>	24	root	root	4096	Jun	1	03:18	root
<code>dr-xr-xr-x.</code>	2	root	root	12288	May	31	04:50	sbin
<code>drwxr-xr-x.</code>	7	root	root	0	Jun	1	03:17	selinux
<code>drwxr-xr-x.</code>	2	root	root	4096	Sep	23	2011	srv
<code>drwxr-xr-x.</code>	13	root	root	0	Jun	1	03:17	sys
<code>drwxrwxrwt.</code>	16	root	root	4096	Jun	1	03:23	tmp
<code>drwxr-xr-x.</code>	13	root	root	4096	May	31	04:36	usr
<code>drwxr-xr-x.</code>	21	root	root	4096	May	31	04:44	var

在 linux 中所有以 `.` 开头的文件都是隐藏目录

1. **bin** : 存放二进制可执行文件(`ls,cat,mkdir` 等), 常用命令一般都在这里。

2. **dev** : 用于存放设备文件。`/dev/null` “黑洞”, 所有写入该设备的信息都将消失。

`/dev/zero` 是类 Unix 操作系统中的一个特殊文件, 用来提供一个空字符文件, 其一个典型的应用就是提供字符

流进行数据存储初始化

3. **home** : 存放所有用户文件的根目录, 是用户主目录的基点。不包括 **root** 用户
- 1). 进入家目录的方式 `cd ~` `cd /home/jingyue`
4. **mnt** : 系统管理员安装临时文件系统的安装点, 系统提供这个目录是让用户临时挂载其他的文件系统。
5. **root** : **root** 目录是超级用户的目录。
6. **sbin** : 一般是命令的扩展命令目录
7. **tmp** : **Linux** 开关机时自动维护的临时目录, 所以不要把文件创建或安装在这个目录下
8. **usr** : **usr** 是个很重要的目录, 通常这一文件系统很大, 因为所有程序安装在这里。

三、修改配置虚拟机网络

- 1、物理机网络共享设置
- 2、确定 VMware 网络连接模式 NAT
- 3、虚拟机中进行网络配置

九、如何修改网卡信息（红色部分注意替换）

`ifconfig` 显示当前的网络配置信息

1. `vi /etc/udev/rules.d/70-persistent-net.rules` 修改物理地址 网络连接描述信息
2. `vi /etc/sysconfig/network-scripts/ifcfg-eth0` `ONBOOT="yes"` 启动时是否激活网卡

`TYPE=Ethernet`

`DEVICE=eth0`

`ONBOOT=yes`

`BOOTPROTO=static`

`IPADDR=192.168.137.101`

`NETMASK=255.255.255.0`

`GATEWAY=192.168.137.1`

`DNS1=192.168.137.1`

`HWADDR=00:0c:29:a8:a7:45`

十、虚拟机的网络连接方式

仅主机 : 虚拟机只能和当前的宿主机相连

桥 接 : 会创建一个实际的 `ip`, 用这个 `ip` 对外可提供放, 可访问外网上网

NAT : 会和宿主机共享一块网卡, 不会创建实际的 `ip`, 但可以利用宿主机的 `ip` 访问外网上网

----- day02 -----

一、Linux 的常用命令

1. 防火墙

- 1) service iptables status 查看状态
- 2) service iptables stop 停止
- 3) chkconfig iptables --list 查看各种运行状态下的防火墙状态
- 4) chkconfig iptables off 关闭各种运行状态下的防火墙
- chkconfig iptables on --level 234

2. ctrl + l 清空屏幕

3. find -name soft 查找一个文件或目录

4. which pwd 查找一个命令所在的位置

5. pwd 查看当前所在的工作目录

6. ls 列出目录下所有内容

- l 显示详细信息 等价 ll
- a 显示所有文件包括隐藏文件(以.开头的文件是隐藏文件)
- h 人性化展示

7. vi 编辑器

1) 进入编辑模式

i 光标当前 a 光标的下一个位置 o 下一行

esc 退出编辑模式

#yy 复制 p 粘贴 #dd 删除 u 后退 ctrl+r 前进

G 光标定位到最后一行 gg 光标定位到第一行

: 命令行模式 :set nu 显示行号 :set nonu 取消行号

:w 保存 :q 退出 ! 强制执行

ctrl + w ,s 水平分屏

ctrl + w ,v 垂直分屏

ctrl + w ,w 移动光标到下一个

ctrl + w ,c 关闭当前屏

ctrl + w ,p 切换分屏光标

/string 从上向下查找字符串 ?string 从下向上查找字符串

8. mkdir 在指定位置创建目录

- 1) -p 递归创建目录

9. rm 删除

- 1) -r 递归删除
- 2) -f 不提示强制执行

10. touch 文件目录 创建一个空文件

11. echo 在控制台输出语句

12. cat 读取一个文件内容到控制台

13. -rw-r--r--. 1 root root 0 Jun 1 09:55 test01

-rw-r--r-- 文件类型和权限
 - 文本文件
 d 目录
 c/b 设备文件
 l 连接文件
 1 连接次数
 root root 属主 属组

14.cp -r 递归拷贝 src path tarpath 将 srcpath 复制到 tarpath cp test01 test01_01
 15.mv srcpath tarpath 将 srcpath 移动到 tarpath mv test02 test02_02
 如果在当前目录下对文件 mv 相当于是对文件重命名
 16.chmod 修改文件权限 r=4 w=2 x=1 chmod 755 test_01_01
 chmod o+rx test_01_01
 17.tail 读取一个文件尾部指定行数的数据 默认显示 10 条
 tail -15 test01 tail -F test01 监控文件尾部数据变化
 -F 参数如果文件被删除又重新创建依然可以监控空数据
 -f 如果文件删除断开监控状态 -F 如果文件删除不会断开监控状态
 18.date 查看时间 date -s "2018-6-4 04:15:19" 设置时间
 date +%Y-%m-%d'
 19.kill -9 ps_id 强制关闭进程

二、linux 定时任务

service crond status 查看状态
crontab -e 配置定时任务的内容
 * * * * *
 分 时 日 月 周 ***3* commend 每下个单位的对应这个时间执行一次
 */1 **** commend 每隔当前对应的时间执行一次
 在定时任务中有时会找不到系统配置的环境 尽量把命令写成全路径
 如果有多个任务就每个任务一行
 因为 linux 是多用户多线程的操作系统,所以控制台的功能有限

date -s "2018-12-16 9:35" 查看时间

----- day03 -----

一、ntp 服务器时间同步 - 需要关闭防火墙

1. 编辑 ntpd 服务配置文件 （在时间服务器主机上，想作为时间同步的标准主机）

- a) **vi /etc/ntp.conf**
 - i. 文件中查找如下内容并修改： 修改红色部分
 - 1. 设置允许哪个网段来访问时间服务器(ntpd) 这里允许 227 网段
 - 2. restrict 192.168.227.0 mask 255.255.255.0 nomodify notrap
 - ii. 注释掉外网服务

```
#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
```
 - iii. # 添加如下内容，本机访问，无需外网时间同步时间

```
server 127.127.1.0
```

2. 启动 ntp 服务

- a) 查看防火墙状态 确保防火墙关闭 (service iptables stop)
 - i. [root@hadoop-2 ~]# service iptables status
 - ii. iptables: Firewall is not running.
- b) 查看 ntp 服务状态
 - i. [root@hadoop-1 /]# service ntpd status
 - ii. ntpd is stopped
- c) 启动 ntp 服务

```
service ntpd start / service ntpd restart
[root@hadoop-1 /]# service ntpd start
Starting ntpd: [ OK ]
```

3. 测试 ntp 时间同步服务功能 、 在其他机器，如 102 机器 ； 同步时间

- a) 登录其他机器
 - i. 比如 192.168.137.102 hadoop-2 机器进行操作
- b) 修改本机时间，修改错误了
 - i. [root@hadoop-2 ~]# date -s "2018-12-16 18:46"
- c) 执行 ntp 时间同步指令（与 101 ip 主机同步，因为 101 启动了 ntpd 服务）
 - i. [root@hadoop-2 ~]# ntpdate -u 192.168.137.101 (ntp 服务器的地址)
- d) 查看时间验证是否同步成功
 - i. [root@hadoop-2 ~]# date

**、 102 机器、通过 crontab 定时进行时间同步

```
[root@hadoop-2 ~]# crontab -e
```

编辑后自动保存到路径：/var/spool/cron/

文件添加以下内容：表示每 2 分钟与 101 机器同步一次

`*/2 * * * * /usr/sbin/ntpdate -u 192.168.137.101`

`13.server 0.cn.pool.ntp.org`

`fudge 0.cn.pool.ntp.org stratum 10` 当前服务器等级如果需要向其他服务器同步时间不要设置成 0

`ntpstat`

`ntpq -p`

二、安装 Java 环境变量

1. 配置 java jdk 环境变量

- a) Ftp == filezilla 工具
 - i. 如何安装
 - ii. 如何连接
 - 1. ip name password port
- b) Java JDK 环境
 - i. 通过 ftp 工具把 windows 里边的 jdk 复制到 远程机器
 - 1. **注意：不要放到 root 目录下，这里建议 `/usr/soft`**
 - ii. 访问、登陆远程机器，找到 那个文件
 - 1. **`cd /usr/soft`**
 - iii. 解压文件
 - 1. **`tar -zxvf` 要解压的 `jdk.tar.gz` 文件**
 - 2. **`cd` 解压后的 `jdk` 文件夹**
 - 3. **`pwd`**
 - 4. **复制显示的 `jdk` 路径 备用**
 - iv. 配置环境变量
 - 1. 用 `vi` 编辑环境变量的配置文件
 - a) **`vi /etc/profile`**
 - b) 打开后在 `profile` 最下边 添加 三个 `export` 开头的配置
 - i. **`export JAVA_HOME=/usr/soft/jdk1.8.0_40`**
 - ii. **`export PATH=$PATH:$JAVA_HOME/bin:`**
 - iii. **`export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar`**
 - v. 生效
 - 1. **`source /etc/profile`**
 - vi. 测试
 - 1. `java -version`
 - 2. `javac`

2. Hadoop 环境

- a) ftp 拷贝同上
- b) hadoop 环境配置
 - i. ftp 拖拽同上
 - ii. 访问 `cd` 同上
 - iii. 解压同上

iv. 配置 环境变量文件

1. 用 vi 编辑环境变量的配置文件

a) vi /etc/profile

b) 打开后在 profile 最下边

i. 添加

1. export HADOOP_HOME=/usr/soft/hadoop-2.7.1

ii. 更新

1. export PATH=\$PATH:\$JAVA_HOME/bin:\$HADOOP_HOME/bin:\$HADOOP_HOME/sbin

2. 生效

a) source /etc/profile

3. 测试

a) java -version

b) javac

三、安装 Hadoop 环境变量

1. 解压 hadoop 安装包

/usr/soft/hadoop*****

2. 配置 hadoop 环境变量同上，（以下仅作参考，以自己实际地址为准）

c) export HADOOP_HOME=/usr/soft/myhadoop/hadoop-2.7.1

d) export PATH=\$PATH:\$JAVA_HOME/bin:\$HADOOP_HOME/bin:\$HADOOP_HOME/sbin

3. 修改 hadoop 配置文件

hadoop-env.sh

slaves （这个文件单独添加所有机器的 对应 ip 和 hosts 名）

core-site.xml

hdfs-site.xml

yarn-site.xml

mapred-site.xml （特殊需要 cp，然后编辑）

hadoop-env.sh 环境变量的配置文件

文件中找到如下两行对应信息，修改等号后边红色部分，为自己的 java 和 hadoop 路径

1) export JAVA_HOME=/usr/soft/myhadoop/jdk1.8.0_40

2) export HADOOP_CONF_DIR=/usr/soft/myhadoop/hadoop-2.7.1/etc/hadoop

slaves

文件的配置 放值 hadoop 工作节点的目录 主机名 每台机器一行

hadoop-1

hadoop-2

hadoop-3

3.hadoop 的加载顺序是先加载 -default ---> -site ???

core-site.xml 找到对应配置内容 进行复制粘贴；红色修改为本机 hosts 名

<!--指定 hdfs 的主端口 namenode 要放在哪台机器-->

<!--配置默认文件系统的名称-->

<!--9000 随意指定的端口 默认端口是 8020-->

<property>

HDFS 对外提供服务的位置

<name>fs.defaultFS</name>

<value>hdfs://**hadoop-1**:9000</value>

</property>

<!--流缓冲区大小-->

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

hdfs-site.xml 找到对应配置内容 进行复制粘贴；**红色修改为本机 hosts 名**

<property>

指定副本数 小于等于 datanode 数

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

在本地(linux)磁盘上存放 namenode 数据的目录

<name>dfs.namenode.name.dir</name>

<value>/usr/soft/name</value>

</property>

<property>

在本地(linux)磁盘上存放 datanode 数据的目录

<name>dfs.datanode.data.dir</name>

<value>/usr/soft/data</value>

</property>

<property>

配置 secondarynamenode 的 Web url

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

<value>**hadoop-1711-001**:50090</value>

</property>

yarn-site.xml 找到对应配置内容 进行复制粘贴；**红色修改为本机 hosts 名**

<property>

配置服务类型 mapreduce_shuffle

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

<value>mapreduce_shuffle</value>

</property>

<property>

指定 resourcemanager 启动在哪台

<name>yarn.resourcemanager.hostname</name>

<value>**hadoop-1711-001**</value>

</property>

mapred-site.xml (需要通过 mapred-site.xml.template, 拷贝然后修改)

cp mapred-site.xml.template mapred-site.xml

vi mapred-site.xml

<property>

把 mapreduce 的计算依赖给 yarn 框架

```

        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
    程序操作历史日志的位置
        <name>mapreduce.jobhistory.address</name>
        <value>hadoop-1711-001:10020</value>
    </property>
    <property>
    web url 访问历史服务的地址
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>hadoop-1711-001:19888</value>
    </property>
    <property>
    存放多少条历史服务
        <name>mapreduce.jobhistory.joblist.cache.size</name>
        <value>20000</value>
    </property>

```

四、克隆虚拟机

1. 右键--->管理--->克隆--->虚拟机中当前状态--->创建完整克隆--->
 配置安装包---->完成
 克隆完检查 1.ip 2.主机名和 ip 映射 3.防火墙是否关闭

- 1、修改 hosts ip
 - a) [root@hadoop-1 hadoop]# vi /etc/hosts
- 2、修改主机名称
 - a) [root@hadoop-1 hadoop]# vi /etc/sysconfig/network
- 3、查看 ip 和 mac 地址
 - a) [root@hadoop-1 hadoop]# ifconfig
- 4、查看 网络情况
 - a) [root@hadoop-1 hadoop]# vi /etc/udev/rules.d/70-persistent-net.rules
- 5、修改配置网络 ip 文件
 - a) [root@hadoop-1 hadoop]# cd /etc/sysconfig/network-scripts/
 - b) [root@hadoop-1 network-scripts]# vim ifcfg-eth0
 - c) service network restart
 - d) ping www.baidu.com

五、配置 ssh 免密登录

- 1.编辑秘钥的配置文件


```
[root@hadoop-2 ~]# vim /etc/ssh/sshd_config
```
- 2.找到下边内容，对应去掉前面的#


```

RSAAuthentication yes
PubkeyAuthentication yes
            
```

AuthorizedKeysFile .ssh/authorized_keys

3.service sshd restart 重启 ssh 服务

3.1 配置 sshd 服务开机启动，保证 CRT 可以正常连接

chkconfig sshd on

4.设置 ssh 的密码是无密码并且使用 rsa 非对称加密生成公私钥

设置生成 非对称 秘钥

[root@hadoop-1 ~]# ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa

重复 三遍 分别把秘钥 拷贝给 hadoop-1 hadoop-2 hadoop-3

[root@hadoop-1 ~]# ssh-copy-id hadoop-3

分别尝试登陆连接

[root@hadoop-1 ~]# ssh hadoop-1

退出登陆连接

[root@hadoop-3 ~]# exit

六、格式化 HDFS

1. 格式化执行

如果有错误，仔细查看错误提示文件，参考以上配置重新修改；

[root@hadoop-1 ~]# hadoop namenode -format

2. 启动两个服务，也可以单独使用

[root@hadoop-1 ~]# start-dfs.sh

[root@hadoop-1 ~]# start-yarn.sh

启动后查看 第一种 验证

[root@hadoop-1 ~]# jps

2787 NameNode

3605 Jps

3064 SecondaryNameNode

3304 NodeManager

3209 ResourceManager

2907 DataNode

启动后查看 第二种 验证

可以访问代表节点 3 的集群表示正常；

主机访问 hdfs 端口

<http://192.168.137.101:50070>

从属机器访问 yarn 端口

<http://192.168.137.101:8088>

3. 停止服务方法

stop-... 停止

[root@hadoop-1 ~]# stop-dfs.sh

[root@hadoop-1 ~]# stop-yarn.sh

----- day04 -----



hadoop 的作者 Doug Cutting 图标

一、从代码层看 hadoop

1. Hadoop Common: 其他组件的公共组件
 2. Hadoop Distributed File System (HDFS™): 分布式的文件系统
 3. Hadoop YARN: 资源调度框架
 4. Hadoop MapReduce: 数据的计算框架
- 从使用的框架 有 2,3,4

二、启动 hadoop

1. start-dfs.sh 启动 HDFS
 - 1) hdfs 访问的 WEB url 主机 + 50070 端口
 - 2) namenode 服务 HDFS 的主节点负责维护所有文件,是文件元数据(描述信息)
 - 3) datanode 服务 实际的数据负责维护实际的数据
 - 4) SecondaryNameNode 定期的改变 namenode 中的信息,它会拷贝 namenode,所以相当 nn 的冷备份。
2. start-yarn.sh 启动 yarn 负责资源调度的框架
 - 1) ResourceManager yarn 中的主节点负责分发资源和创建容器NodeManager 一般和 datanode 是伴随的服务,负责维护实际数据

三、block 的概念

1. 一个文件切分开存放,每一部分叫一个 block。
2. 默认切分规则是 128M 对应一个 block。
3. 实际的文件大小按照文件的实际值。

四、HDFS shell 命令

1. hdfs dfs -help 帮助
2. -cat 查看文本文件的内容
3. -chmod 修改权限 使用创建的那个用户 hdfs dfs -chmod o+w /test
4. -copyFromLocal == -put

	linux 的路径	hdfs 的路径
hdfs dfs -copyFromLocal	in_use.lock	/
hdfs dfs -put in_use.lock	/	

5. `-copyToLocal == -get`

	hdfs 的路径	linux 的路径
<code>hdfs dfs -copyToLocal</code>	<code>/in_use.lock</code>	<code>./</code>
<code>hdfs dfs -get</code>	<code>/in_use.lock</code>	<code>./</code>

6. `-cp` `hdfs dfs -cp /LICENSE.txt /test/license` `-p` 递归拷贝

7. `-ls` `hdfs dfs -ls /` 查看目录下的文件信息

8. `-mkdir` `hdfs dfs -mkdir /test` 指定位置创建目录 `-p` 递归创建

9. `-appendToFile`

由于 HDFS 上的文件修改不能支持很好,只能在文件末尾添加数据

linux 下的文件	HDFS 上的文件
------------	-----------

<code>hdfs dfs -appendToFile</code>	<code>in_use.lock</code>	<code>/test/in_use.lock</code>
-------------------------------------	--------------------------	--------------------------------

10. `-text` 高级的查看文本文件的内容

11. `-rm` `hdfs dfs -rm /test/123.rar` 删除文件 `-f` 不询问强制删除 `-R` 递归删除

`hdfs dfs -rm -R -f /test` 不支持参数的组合使用

12. `-mv` `hdfs dfs -mv /Text.zip /output180401_2` 移动文件

13. `rm` 删除文件 如果需要加入回收站可以 修改 如下配置文件

`core-site.xml`

扩展回收站

`<property>`

`<name>fs.trash.interval</name>`

`<value>1440</value>`

`</property>`

还原回收站文件 `-mv` 到其他目录即可

----- day05 -----

—————周总结考核

周考内容:

- 01、使用的 Linux 的名称、属于哪个主流发行版本的分支?
cnetos 6.5; 红帽系列的社区版本
- 02、查看 /usr/soft 下的详细信息, 包括隐藏文件的命令是什么?
ls -la
- 03、同时创建一个多级目录、在/usr/soft/huagong (soft huagong 都不存在)
mkdir -p /usr/soft/huagong (递归创建 -p)
- 04、简述 NAT 模式、桥接模式、仅主机模式的作用?
NAT 没有独立 ip、与宿主机想用相同网卡设施;
桥接有独立 ip 相当于独立机器;
仅主机、只能和宿主机相连, 不能访问互联网和局域网;
- 05、Apache hadoop 的四大核心模块儿是什么?
hadoop common: 公共组件、代码底层公用组件
hadoop hdfs: 分布式文件系统
mapreduce:
yarn:
- 06、core-site.xml 哪个属性配置 HDFS 主节点? (NameNode)
fs.defaultFS
- 07、如何启动 HDFS? start-dfs.sh
- 08、如何启动 YARN? start-yarn.sh
- 09、搭建集群为何要配置 SSH 免密钥登录?
hadoop 集群运行中需要通过 ssh 协议相互发送信息。
- 10、HDFS 中 block 的概念是什么?
如果文件存储在 hdfs 上, 存储的单位叫做 block。
- 11、block 的默认大小多少? 128M
- 12、50070 访问的功能是? hdfs 浏览器【查看】端口; 只能查看
- 13、8088 访问的功能是? yarn 浏览器【查看】端口; 只能查看
- 14、HDFS 启动后有哪些进程被启动? (jps 可以看到)
NameNode 存放文件描述信息位置;
DataNode 存放实际数据的内容; 可以有多个;
- 15、yarn 启动后有哪些进程被启动?
ResourceManager 一个集群只有一个, 在主节点上;
NodeManager 根据多少个 DataNode 出现多少个 NodeManager

构建 Maven 项目

1. 新建项目(Eclipse / MyEclipse)
2. 选择 Maven
 - ① org.apache.maven.archetypes maven-archetype-quickstart 1.1
3. 导入配置依赖路径 settings.xml 文件
 - ① Windows - Preferences - Maven - User Settings
 - ② user setting s (open file); browse 选择 settings.xml 文件
4. 编辑 pom.xml 配置文件

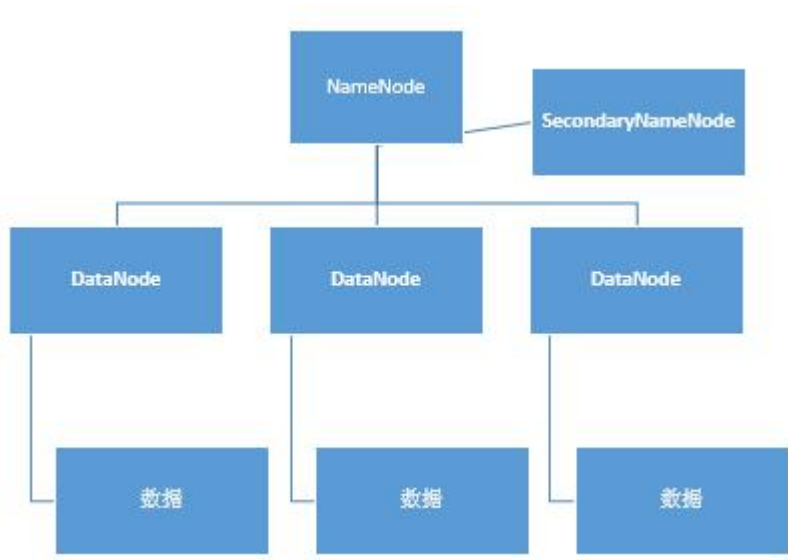
添加 eclipse maven HDFSapi 依赖

```
<dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-client</artifactId>
    <version>2.7.1</version>
</dependency>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
            </configuration>
        </plugin>
    </plugins>
</build>
```

5. 测试项目运行

```
//import java.net.URI;
//import org.apache.hadoop.conf.Configuration;
//import org.apache.hadoop.fs.FileSystem;
public class App
{
    public static void main( String[] args ) throws Exception
    {
        System.out.println( "Hello World!" );
        Configuration conf = new Configuration();
        FileSystem fileSystem = FileSystem.get(new URI("hdfs://hadoop-1:9000"), conf, "root");
        fileSystem.close();
    }
}
```

一、HDFS 基本架构关系图解



参考: <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#Introduction>

HDFS (Hadoop Distributed File System) Hadoop 分布式文件系统

基本结构分 NameNode、SecondaryNameNode、DataNode 这几个

NameNode: 是 Master 节点, 有点类似 Linux 里的根目录。管理数据块映射; 处理客户端的读写请求; 配置副本策略; 管理 HDFS 的名称空间;

Secondary NameNode: 保存着 NameNode 的部分信息 (不是全部信息 NameNode 宕掉之后恢复数据用), 是 NameNode 的冷备份; 合并 fsimage 和 edits 然后再发给 namenode。(防止 edits 过大的一种解决方案)

DataNode: 负责存储 client 发来的数据块 block; 执行数据块的读写操作。是 NameNode 的小弟。

热备份: b 是 a 的热备份, 如果 a 坏掉。那么 b 马上运行代替 a 的工作。

冷备份: b 是 a 的冷备份, 如果 a 坏掉。那么 b 不能马上代替 a 工作。但是 b 上存储 a 的一些信息, 减少 a 坏掉之后的损失。

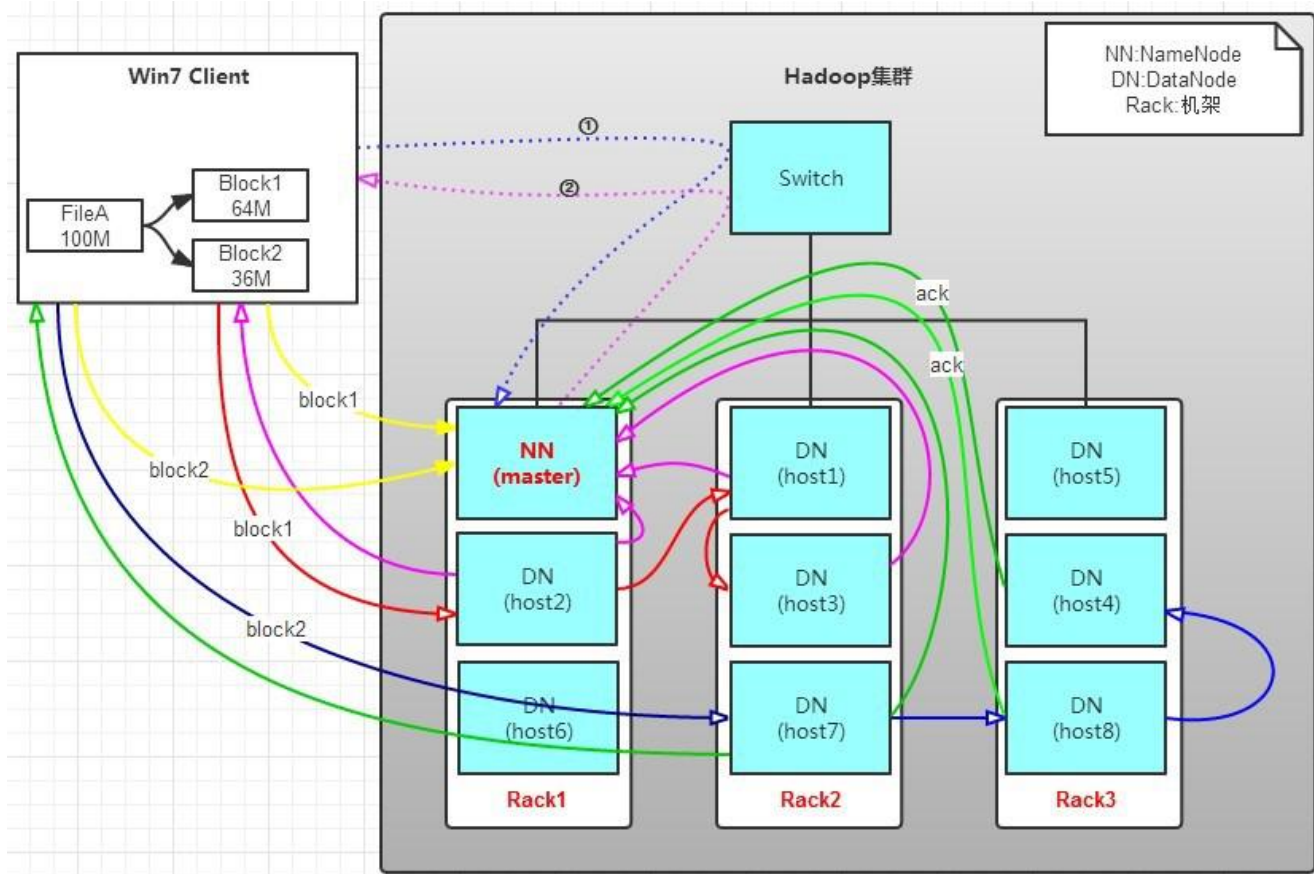
fsimage: 元数据镜像文件 (文件系统的目录树。)

edits: 元数据的操作日志 (针对文件系统做的修改操作记录)

namenode 内存中存储的是=fsimage+edits。

参考: <https://www.cnblogs.com/wxplmm/p/7239342.html>

二、HDFS 文件读写过程图解



- a. Client 将 FileA 按 64M 分块。分成两块，block1 和 Block2;
- b. Client 向 nameNode 发送写数据请求，如图蓝色虚线①----->。
- c. NameNode 节点，记录 block 信息。并返回可用的 DataNode，如粉色虚线②----->。

Block1: host2,host1,host3

Block2: host7,host8,host4

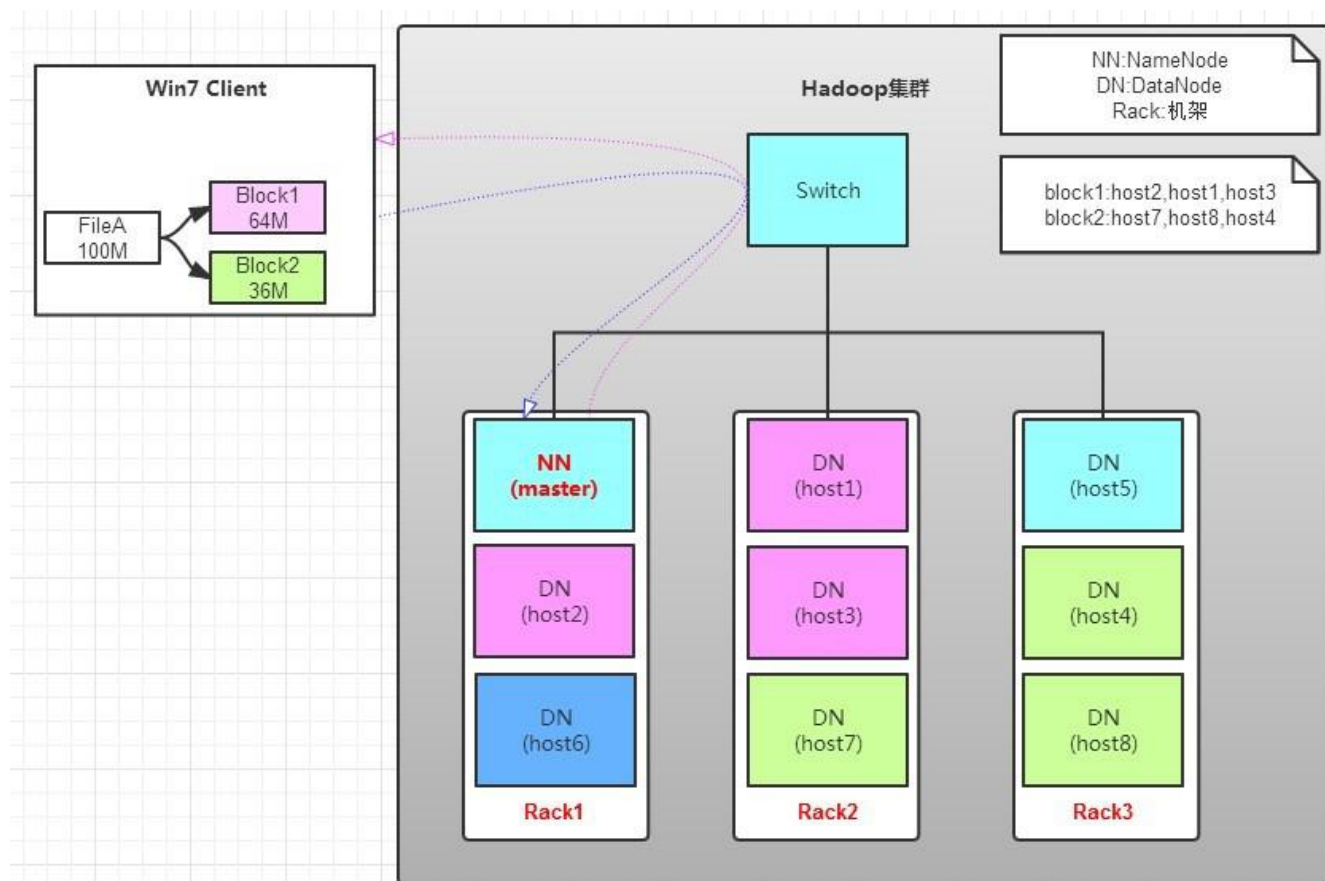
- d. client 向 DataNode 发送 block1；发送过程是以流式写入。

hdfs 流式写入过程

- 1>将 64M 的 block1 按 64k 的 package 划分;
- 2>然后将第一个 package 发送给 host2;
- 3>host2 接收完后，将第一个 package 发送给 host1，同时 client 想 host2 发送第二个 package;
- 4>host1 接收完第一个 package 后，发送给 host3，同时接收 host2 发来的第二个 package。
- 5>以此类推，如图红线实线所示，直到将 block1 发送完毕。
- 6>host2,host1,host3 向 NameNode，host2 向 Client 发送通知，说“消息发送完了”。如图粉红颜色实线所示。
- 7>client 收到 host2 发来的消息后，向 namenode 发送消息，说我写完了。这样就真完成了。如图黄色粗实线
- 8>发送完 block1 后，再向 host7，host8，host4 发送 block2，如图蓝色实线所示。
- 9>发送完 block2 后，host7,host8,host4 向 NameNode，host7 向 Client 发送通知，如图浅绿色实线所示。

10>client 向 NameNode 发送消息，说我写完了，如图**黄色粗实线**。。。这样就完毕了。

hdfs 文件读操作



读操作就简单一些了，如图所示，client 要从 datanode 上，读取 FileA。而 FileA 由 block1 和 block2 组成。

那么，读操作流程为：

a. client 向 namenode 发送读请求。

b. namenode 查看 Metadata 信息，返回 fileA 的 block 的位置。

block1: host2, host1, host3

block2: host7, host8, host4

c. block 的位置是有先后顺序的，先读 block1，再读 block2。而且 block1 去 host2 上读取；然后 block2，去 host7 上读取；

上面例子中，client 位于机架外，那么如果 client 位于机架内某个 DataNode 上，例如，client 是 host6。那么读取的时候，遵循的规律是：

优选读取本机架上的数据。

运算和存储在同一个服务器中，每一个服务器都可以是本地服务器

三、HDFS 基础理论知识

一、HDFS 的安全模式 安全模式下 HDFS 是只读

1. `hadoop dfsadmin -safemode get` 查看是否是安全模式
2. hdfs 在启动的时候回去检测各节点上的文件是否正常,进入安全模式
3. `hadoop dfsadmin -safemode leave` 关闭安全模式
4. `hadoop dfsadmin -safemode enter` 开启安全模式
5. 没有文件不会进入安全模式

二、namenode 的大小是固定的----->无法高效存储大量小文件。

三、namenode 在启动后会把数据加载到内存中

四、查看块报告

(1) `hdfs dfsadmin -report` 1 小时报告一次

五、心跳 3 秒一次 超过 10 分钟认为这个节点不可用

六、checksum 文件创建后的三周开始检测

