# 什么是大数据

## 基本概念

在互联网技术发展到现今阶段，大量日常、工作等事务产生的数据都已经信息化，人类产生的数据量相比以前有了爆炸式的增长，以前的传统的数据处理技术已经无法胜任，需求催生技术，一套用来处理海量数据的软件工具应运而生，这就是大数据！

换个角度说，大数据是：

1. 有海量的数据
2. 有对海量数据进行挖掘的需求
3. 有对海量数据进行挖掘的软件工具（hadoop、spark、storm、flink、tez、impala......）

## 大数据在现实生活中的具体应用

电商推荐系统：基于海量的浏览行为、购物行为数据，进行大量的算法模型的运算，得出各类推荐结论，以供电商网站页面来为用户进行商品推荐

精准广告推送系统：基于海量的互联网用户的各类数据，统计分析，进行用户画像（得到用户的各种属性标签），然后可以为广告主进行有针对性的精准的广告投放

# 什么是hadoop

hadoop中有3个核心组件：

分布式文件系统：HDFS —— 实现将文件分布式存储在很多的服务器上

分布式运算编程框架：MAPREDUCE —— 实现在很多机器上分布式并行运算

分布式资源调度平台：YARN —— 帮用户调度大量的mapreduce程序，并合理分配运算资源

# hdfs整体运行机制

hdfs：分布式文件系统

hdfs有着文件系统共同的特征：

1. 有目录结构，顶层目录是： /
2. 系统中存放的就是文件
3. 系统可以提供对文件的：创建、删除、修改、查看、移动等功能

hdfs跟普通的单机文件系统有区别：

1. 单机文件系统中存放的文件，是在一台机器的操作系统中
2. hdfs的文件系统会横跨N多的机器
3. 单机文件系统中存放的文件，是在一台机器的磁盘上
4. hdfs文件系统中存放的文件，是落在n多机器的本地单机文件系统中（hdfs是一个基于linux本地文件系统之上的文件系统）

hdfs的工作机制：

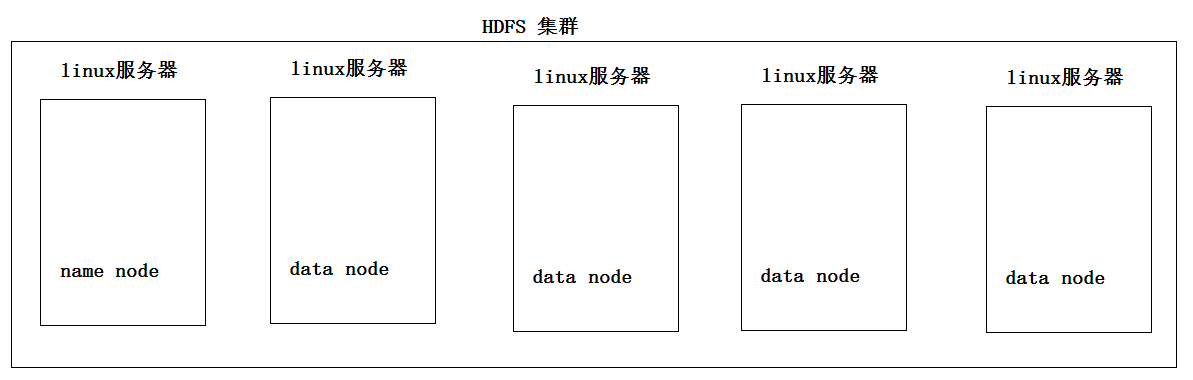
1. 客户把一个文件存入hdfs，其实hdfs会把这个文件切块后，分散存储在N台linux机器系统中（负责存储文件块的角色：data node）<准确来说：切块的行为是由客户端决定的>
2. 一旦文件被切块存储，那么，hdfs中就必须有一个机制，来记录用户的每一个文件的切块信息，及每一块的具体存储机器（负责记录块信息的角色是：name node）

3、为了保证数据的安全性，hdfs可以将每一个文件块在集群中存放多个副本（到底存几个副本，是由当时存入该文件的客户端指定的）

综述：一个hdfs系统，由一台运行了namenode的服务器，和N台运行了datanode的服务器组成！

# 搭建hdfs分布式集群

### 4.1 hdfs集群组成结构：



### 4.2 安装hdfs集群的具体步骤：

**一、首先需要准备N台linux服务器**

学习阶段，用虚拟机即可！

先准备4台虚拟机：1个namenode节点 + 3 个datanode 节点

**二、修改各台机器的主机名和ip地址**

主机名：hdp-01 对应的ip地址：192.168.33.61

主机名：hdp-02 对应的ip地址：192.168.33.62

主机名：hdp-03 对应的ip地址：192.168.33.63

主机名：hdp-04 对应的ip地址：192.168.33.64

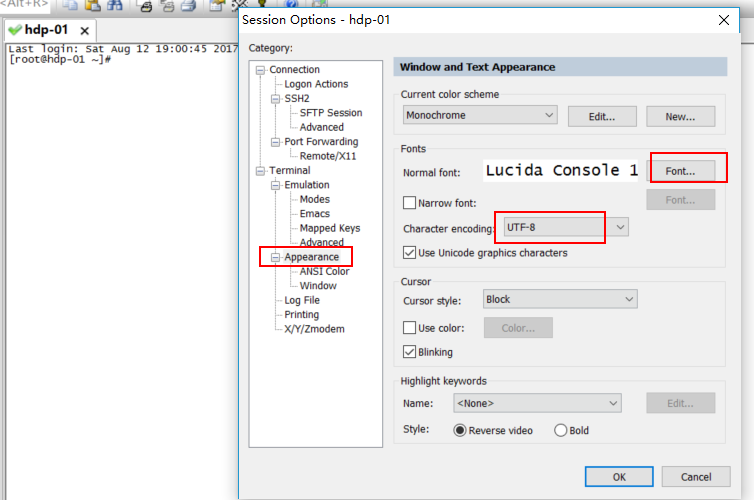
三、从windows中用CRT软件进行远程连接

在windows中将各台linux机器的主机名配置到的windows的本地域名映射文件中：

c:/windows/system32/drivers/etc/hosts

|  |
| --- |
| 192.168.33.61 hdp-01  192.168.33.62 hdp-02  192.168.33.63 hdp-03  192.168.33.64 hdp-04 |

用crt连接上后，修改一下crt的显示配置（字号，编码集改为UTF-8）：



**四、配置linux服务器的基础软件环境**

* 防火墙

关闭防火墙：service iptables stop

关闭防火墙自启： chkconfig iptables off

* 安装jdk：（hadoop体系中的各软件都是java开发的）

1. 利用alt+p 打开sftp窗口，然后将jdk压缩包拖入sftp窗口
2. 然后在linux中将jdk压缩包解压到/root/apps 下
3. 配置环境变量：JAVA\_HOME PATH

vi /etc/profile 在文件的最后，加入：

|  |
| --- |
| export JAVA\_HOME=/root/apps/jdk1.8.0\_60  export PATH=$PATH:$JAVA\_HOME/bin |

1. 修改完成后，记得 source /etc/profile使配置生效
2. 检验：在任意目录下输入命令： java -version 看是否成功执行
3. 将安装好的jdk目录用scp命令拷贝到其他机器
4. 将/etc/profile配置文件也用scp命令拷贝到其他机器并分别执行source命令

* 集群内主机的**域名映射**配置

在hdp-01上，vi /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  192.168.33.61 hdp-01  192.168.33.62 hdp-02  192.168.33.63 hdp-03  192.168.33.64 hdp-04 |

然后，将hosts文件拷贝到集群中的所有其他机器上

scp /etc/hosts hdp-02:/etc/

scp /etc/hosts hdp-03:/etc/

scp /etc/hosts hdp-04:/etc/

|  |  |
| --- | --- |
| 补充  提示: | 如果在执行scp命令的时候，提示没有scp命令，则可以配置一个本地yum源来安装   1. 先在虚拟机中配置cdrom为一个centos的安装镜像iso文件 2. 在linux系统中将光驱挂在到文件系统中（某个目录） 3. mkdir /mnt/cdrom 4. mount -t iso9660 -o loop /dev/cdrom /mnt/cdrom 5. 检验挂载是否成功： ls /mnt/cdrom 6. 3、配置yum的仓库地址配置文件 7. yum的仓库地址配置文件目录： /etc/yum.repos.d 8. 先将自带的仓库地址配置文件批量更名：      1. 然后，拷贝一个出来进行修改        1. 修改完配置文件后，再安装scp命令： 2. yum install openssh-clients -y |

* **五、安装hdfs集群**

1、上传hadoop安装包到hdp-01

1. 修改配置文件

|  |  |
| --- | --- |
| 要点提示 | ***核心配置参数：***   1. *指定hadoop的默认文件系统为：hdfs* 2. *指定hdfs的namenode节点为哪台机器* 3. *指定namenode软件存储元数据的****本地目录*** 4. *指定datanode软件存放文件块的****本地目录*** |

hadoop的配置文件在：/root/apps/hadoop安装目录/etc/hadoop/

**1) 修改hadoop-env.sh**

*export JAVA\_HOME=/root/apps/jdk1.8.0\_60*

**2) 修改core-site.xml**

|  |
| --- |
| <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://hdp-01:9000</value>  </property>  </configuration> |

**3) 修改hdfs-site.xml**

|  |
| --- |
| <configuration>  <property>  <name>dfs.namenode.name.dir</name>  <value>/root/dfs/name</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>/root/dfs/data</value>  </property>  </configuration> |

**4) 拷贝整个hadoop安装目录到其他机器**

*scp -r /root/apps/hadoop-2.8.0 hdp-02:/root/apps/*

*scp -r /root/apps/hadoop-2.8.0 hdp-03:/root/apps/*

*scp -r /root/apps/hadoop-2.8.0 hdp-04:/root/apps/*

**5) 启动HDFS**

所谓的启动HDFS，就是在对的机器上启动对的软件

|  |  |  |
| --- | --- | --- |
| 要点  提示： | *要运行hadoop的命令，需要在linux环境中配置HADOOP\_HOME和PATH环境变量*  *vi /etc/profile*   |  | | --- | | *export JAVA\_HOME=/root/apps/jdk1.8.0\_60*  *export HADOOP\_HOME=/root/apps/hadoop-2.8.0*  *export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin* | |

**首先，初始化namenode的元数据目录**

要在hdp-01上执行hadoop的一个命令来初始化namenode的元数据存储目录

*hadoop namenode -format*

* 创建一个全新的元数据存储目录
* 生成记录元数据的文件fsimage
* 生成集群的相关标识：如：集群id——clusterID

**然后，启动namenode进程（在hdp-01上）**

*hadoop-daemon.sh start namenode*

启动完后，首先用jps查看一下namenode的进程是否存在

**然后，在windows中用浏览器访问namenode提供的web端口：50070**

http://hdp-01:50070

**然后，启动众datanode们（在任意地方）**

*hadoop-daemon.sh start datanode*

**6) 用自动批量启动脚本来启动HDFS**

1. 先配置hdp-01到集群中所有机器（包含自己）的免密登陆
2. 配完免密后，可以执行一次 *ssh 0.0.0.0*
3. 修改hadoop安装目录中/etc/hadoop/slaves（把需要启动datanode进程的节点列入）

|  |
| --- |
| hdp-01  hdp-02  hdp-03  hdp-04 |

1. 在hdp-01上用脚本：**start-dfs.sh** 来自动启动整个集群
2. 如果要停止，则用脚本：**stop-dfs.sh**

# hdfs的客户端操作

## 客户端的理解

hdfs的客户端有多种形式：

1. 网页形式
2. 命令行形式
3. 客户端在哪里运行，没有约束，只要运行客户端的机器能够跟hdfs集群联网

文件的切块大小和存储的副本数量，都是由客户端决定！

所谓的由客户端决定，是通过配置参数来定的

hdfs的客户端会读以下两个参数，来决定切块大小、副本数量：

切块大小的参数： dfs.blocksize

副本数量的参数： dfs.replication

上面两个参数应该配置在客户端机器的hadoop目录中的hdfs-site.xml中配置

|  |
| --- |
| *<property>*  *<name>dfs.blocksize</name>*  *<value>64m</value>*  *</property>*  *<property>*  *<name>dfs.replication</name>*  *<value>2</value>*  *</property>* |

## hdfs命令行客户端的常用操作命令

1. 查看hdfs中的**目录信息**

hadoop fs -ls /hdfs路径

1. **上传文件**到hdfs中

hadoop fs -put /本地文件 /aaa

hadoop fs -copyFromLocal /本地文件 /hdfs路径 ## copyFromLocal等价于 put

hadoop fs -moveFromLocal /本地文件 /hdfs路径 ## 跟copyFromLocal的区别是：从本地**移动**到hdfs中

1. **下载文件**到客户端本地磁盘

hadoop fs -get /hdfs中的路径 /本地磁盘目录

hadoop fs -copyToLocal /hdfs中的路径 /本地磁盘路径 ## 跟get等价

hadoop fs -moveToLocal /hdfs路径 /本地路径 ## 从hdfs中移动到本地

1. 在hdfs中**创建文件夹**

hadoop fs -mkdir -p /aaa/xxx

1. 移动hdfs中的文件（更名）

hadoop fs -mv /hdfs的路径 /hdfs的另一个路径

1. **删除**hdfs中的文件或文件夹

hadoop fs -rm -r /aaa

1. 修改文件的权限

hadoop fs -chown user:group /aaa

hadoop fs -chmod 700 /aaa

1. 追加内容到已存在的文件

hadoop fs -appendToFile /本地文件 /hdfs中的文件

1. **显示**文本文件的内容

hadoop fs -cat /hdfs中的文件

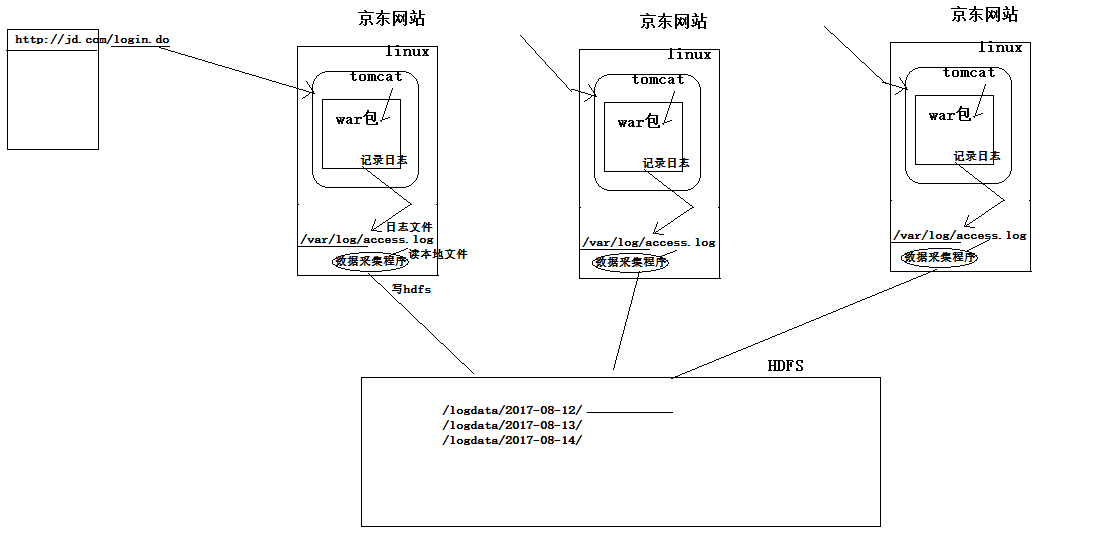
hadoop fs -tail /hdfs中的文件

补充：hdfs命令行客户端的所有命令列表

|  |
| --- |
| Usage: hadoop fs [generic options]  [-appendToFile <localsrc> ... <dst>]  [-cat [-ignoreCrc] <src> ...]  [-checksum <src> ...]  [-chgrp [-R] GROUP PATH...]  [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]  [-chown [-R] [OWNER][:[GROUP]] PATH...]  [-copyFromLocal [-f] [-p] [-l] [-d] <localsrc> ... <dst>]  [-copyToLocal [-f] [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] <path> ...]  [-cp [-f] [-p | -p[topax]] [-d] <src> ... <dst>]  [-createSnapshot <snapshotDir> [<snapshotName>]]  [-deleteSnapshot <snapshotDir> <snapshotName>]  [-df [-h] [<path> ...]]  [-du [-s] [-h] [-x] <path> ...]  [-expunge]  [-find <path> ... <expression> ...]  [-get [-f] [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  [-getfacl [-R] <path>]  [-getfattr [-R] {-n name | -d} [-e en] <path>]  [-getmerge [-nl] [-skip-empty-file] <src> <localdst>]  [-help [cmd ...]]  [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [<path> ...]]  [-mkdir [-p] <path> ...]  [-moveFromLocal <localsrc> ... <dst>]  [-moveToLocal <src> <localdst>]  [-mv <src> ... <dst>]  [-put [-f] [-p] [-l] [-d] <localsrc> ... <dst>]  [-renameSnapshot <snapshotDir> <oldName> <newName>]  [-rm [-f] [-r|-R] [-skipTrash] [-safely] <src> ...]  [-rmdir [--ignore-fail-on-non-empty] <dir> ...]  [-setfacl [-R] [{-b|-k} {-m|-x <acl\_spec>} <path>]|[--set <acl\_spec> <path>]]  [-setfattr {-n name [-v value] | -x name} <path>]  [-setrep [-R] [-w] <rep> <path> ...]  [-stat [format] <path> ...]  [-tail [-f] <file>]  [-test -[defsz] <path>]  [-text [-ignoreCrc] <src> ...]  [-touchz <path> ...]  [-truncate [-w] <length> <path> ...]  [-usage [cmd ...]] |

# hdfs的java客户端编程

HDFS客户端编程应用场景：数据采集



在windows开发环境中做一些准备工作：

1. 在windows的某个路径中解压一份windows版本的hadoop安装包
2. 将解压出的hadoop目录配置到windows的环境变量中：HADOOP\_HOME

## 开发代码

1. 将hdfs客户端开发所需的jar导入工程（jar包可在hadoop安装包中找到common/hdfs）
2. 写代码

要点：要对hdfs中的文件进行操作，代码中首先需要获得一个hdfs的客户端对象

Configuration conf = new Configuration();

FileSystem fs = FileSystem.get(new URI("hdfs://hdp-01:9000"),conf,"root");

1. 利用fs对象的方法进行文件操作

比如：

上传文件—— fs.copyFromLocalFile(new Path("本地路径"),new Path("hdfs的路径"));

下载文件——fs.copyToLocalFile(new Path("hdfs的路径"),new Path("本地路径"))

## 项目实战

### 需求描述：

在业务系统的服务器上，业务程序会不断生成业务日志（比如网站的页面访问日志）

业务日志是用log4j生成的，会不断地切出日志文件

需要定期（比如每小时）从业务服务器上的日志目录中，探测需要采集的日志文件(access.log不能采)，发往HDFS

注意点：业务服务器可能有多台(hdfs上的文件名不能直接用日志服务器上的文件名)

当天采集到的日志要放在hdfs的当天目录中

采集完成的日志文件，需要移动到到日志服务器的一个备份目录中

定期检查（一小时检查一次）备份目录，将备份时长超出24小时的日志文件清除

Timer timer = new Timer()

timer.schedual()

# hdfs的核心工作原理

## namenode元数据管理要点

1. 什么是元数据？

hdfs的目录结构及每一个文件的块信息（块的id，块的副本数量，块的存放位置<datanode>）

1. 元数据由谁负责管理？

namenode

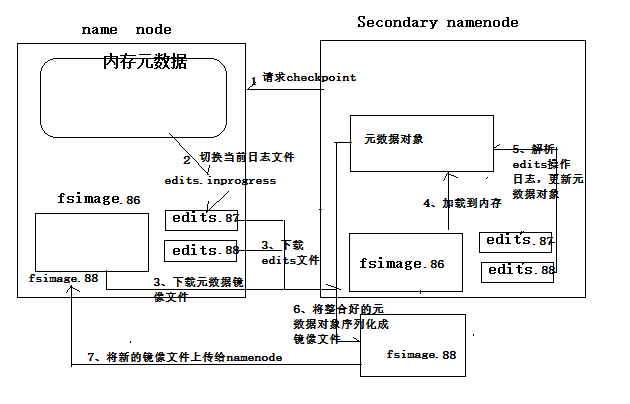
1. namenode把元数据记录在哪里？

namenode的实时的完整的元数据存储在内存中；

namenode还会在磁盘中（dfs.namenode.name.dir）存储内存元数据在某个时间点上的镜像文件；

namenode会把引起元数据变化的客户端操作记录在edits日志文件中；

|  |
| --- |
| secondarynamenode会定期从namenode上下载fsimage镜像和新生成的edits日志，然后加载fsimage镜像到内存中，然后顺序解析edits文件，对内存中的元数据对象进行修改（整合）  整合完成后，将内存元数据序列化成一个新的fsimage，并将这个fsimage镜像文件上传给namenode |
| 上述过程叫做：checkpoint操作  提示：secondary namenode每次做checkpoint操作时，都需要从namenode上下载上次的fsimage镜像文件吗？  第一次checkpoint需要下载，以后就不用下载了，因为自己的机器上就已经有了。 |



补充：secondary namenode启动位置的配置

|  |  |
| --- | --- |
| 默认值 | <property>  <name>dfs.namenode.secondary.http-address</name>  <value>0.0.0.0:50090</value>  </property> |

把默认值改成你想要的机器主机名即可

secondarynamenode保存元数据文件的目录配置：

|  |  |
| --- | --- |
| 默认值 | <property>  <name>dfs.namenode.checkpoint.dir</name>  <value>file://${hadoop.tmp.dir}/dfs/namesecondary</value>  </property> |

改成自己想要的路径即可：/root/dfs/namesecondary

## 写数据流程

## 读数据流程

# mapreduce快速上手

小案例：

统计HDFS的/wordcount/input/a.txt文件中的每个单词出现的次数——wordcount

明白了一点：可以在任何地方运行程序，访问HDFS上的文件并进行统计运算，并且可以把统计的结果写回HDFS的结果文件中；

但是，进一步思考：如果文件又多又大，用上面那个程序有什么弊端？

慢！因为只有一台机器在进行运算处理！

如何变得更快？

核心思想：让我们的运算程序并行在多台机器上执行！

# 9、mapreduce运行平台YARN

mapreduce程序应该是在很多机器上并行启动，而且先执行map task，当众多的maptask都处理完自己的数据后，还需要启动众多的reduce task，这个过程如果用用户自己手动调度不太现实，需要一个自动化的调度平台——hadoop中就为运行mapreduce之类的分布式运算程序开发了一个自动化调度平台——YARN

## 安装yarn集群

yarn集群中有两个角色：

主节点：Resource Manager 1台

从节点：Node Manager N台

Resource Manager一般安装在一台专门的机器上

Node Manager应该与HDFS中的data node重叠在一起

修改配置文件：

yarn-site.xml

|  |
| --- |
| <property>  <name>yarn.resourcemanager.hostname</name>  <value>hdp-04</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property> |

然后复制到每一台机器上

然后在hdp-04上，修改hadoop的slaves文件，列入要启动nodemanager的机器

然后将hdp-04到所有机器的免密登陆配置好

然后，就可以用脚本启动yarn集群：

sbin/start-yarn.sh

停止：

sbin/stop-yarn.sh

启动完成后，可以在windows上用浏览器访问resourcemanager的web端口：

<http://hdp-04:8088>

看resource mananger是否认出了所有的node manager节点

# 运行mapreduce程序

首先，为你的mapreduce程序开发一个提交job到yarn的客户端类（模板代码）：

* 描述你的mapreduce程序运行时所需要的一些信息(比如用哪个mapper、reducer、map和reduce输出的kv类型、jar包所在路径、reduce task的数量、输入输出数据的路径)
* 将信息和整个工程的jar包一起交给yarn

然后，将整个工程（yarn客户端类+ mapreduce所有jar和自定义类）打成jar包

然后，将jar包上传到hadoop集群中的任意一台机器上

最后，运行jar包中的（YARN客户端类）

[root@hdp-04 ~]# hadoop jar wc.jar cn.edu360.hadoop.mr.wc.JobSubmitter