@Linux123 rpm -aq|grep mariadb

rpm -e --nodeps mariadb-libs

yum install perl -y

yum install net-tools -y

Mysqld password改为12345678

查找MYSQL root口令：

grep password /var/log/mysqld.log

#设置MySQL密码口令

set global validate\_password\_policy=0;

-- 创建用户设置口令、授权、刷新

CREATE USER 'hive'@'%' IDENTIFIED BY '12345678';

GRANT ALL ON \*.\* TO 'hive'@'%';

FLUSH PRIVILEGES;

mysql -u hive -p12345678

cp mysql-connector-java-5.1.46.jar /opt/lagou/servers/hive-2.3.7/lib/

启动hive前需要

root@linux121:start-dfs.sh

start-yarn.sh

hive的日志在/tmp/root/下

在 vi 中输入特殊字符：

(Ctrl + v) + (Ctrl + a) => ^A

(Ctrl + v) + (Ctrl + b) => ^B

(Ctrl + v) + (Ctrl + c) => ^C

显示特殊字符

cat -A /home/hadoop/data/s1.dat

把当前的Profile文件考到121上同级目录：

scp profile linux121:$PWD

nohup hive --service metastore &

systemctl stop firewalld 关闭防火墙

lsof -i:9083 查看端口9083连接情况：

[root@linux121 hadoop]# lsof -i:9083

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

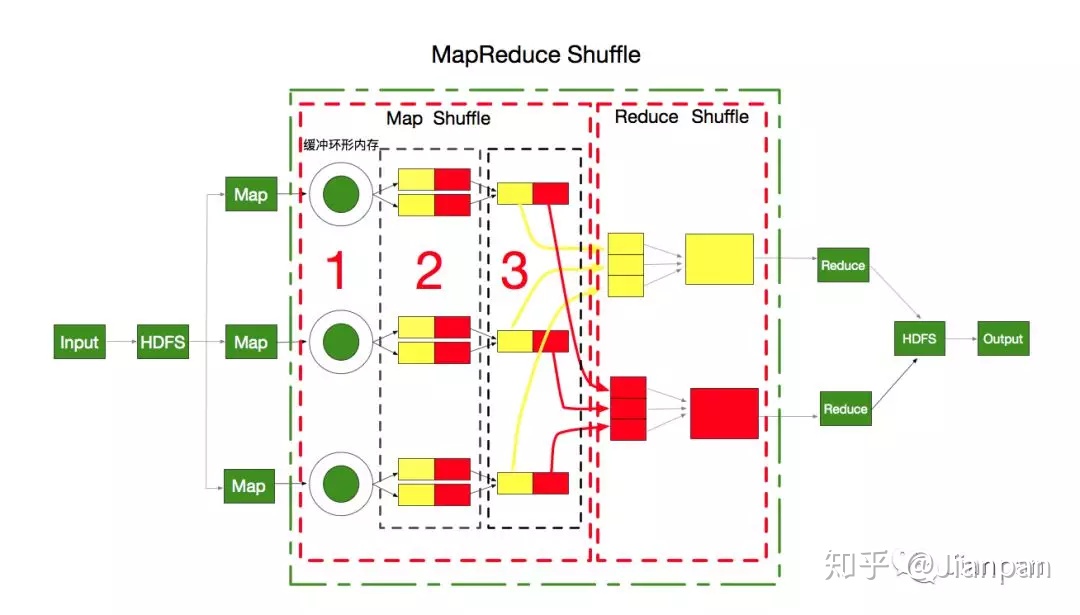
java 4231 root 513u IPv4 59719 0t0 TCP \*:emc-pp-mgmtsvc (LISTEN)

java 4231 root 514u IPv4 59724 0t0 TCP linux121:emc-pp-mgmtsvc->linux122:56886 (ESTABLISHED)

beeline也是客户端，类似于hive

/opt/lagou/servers/hive-2.3.7/conf/core-site.xml

scp hdfs-site.xml linux122:$PWD



如图绿色虚线框框所示：从map( )的输出到reduce( )的输入，中间的过程被称为shuffle过程。Shuffle 过程分为Map端跟Reduce端的Shuffle过程：  
**Map端流程：**

1. 环形内存缓存区：每个split数据交由一个map任务处理，map的处理结果不会直接写到硬盘上，会先输送到环形内存缓存区中，默认的大小是100M（可通过配置修改），当缓冲区的内容达到80%后会开始溢出，此时缓存区的溢出内容会被写到磁盘上，形成一个个spill file，注意这个文件没有固定大小。
2. 在内存中经过分区、排序后溢出到磁盘：分区主要功能是用来指定 map 的输出结果交给哪个 reduce 任务处理，默认是通过 map 输出结果的 key 值取hashcode 对代码中配置的 redue task数量取模运算，值一样的分到一个区，也就是一个 reduce 任务对应一个分区的数据。这样做的好处就是可以避免有的 reduce 任务分配到大量的数据，而有的 reduce 任务只分配到少量甚至没有数据，平均 reduce 的处理能力。并且在每一个分区（partition）中，都会有一个 sort by key 排序，如果此时设置了 Combiner，将排序后的结果进行 Combine 操作，相当于 map 阶段的本地 reduce，这样做的目的是让尽可能少的数据写入到磁盘。
3. 合并溢出文件：随着 map 任务的执行，不断溢出文件，直到输出最后一个记录，可能会产生大量的溢出文件，这时需要对这些大量的溢出文件进行合并，在合并文件的过程中会不断的进行排序跟 Combine 操作，这样做有两个好处：减少每次写入磁盘的数据量&减少下一步 reduce 阶段网络传输的数据量。最后合并成了一个分区且排序的大文件，此时可以再进行配置压缩处理，可以减少不同节点间的网络传输量。合并完成后着手将数据拷贝给相对应的reduce 处理，那么要怎么找到分区数据对应的那个 reduce 任务呢？简单来说就是 JobTracker 中保存了整个集群中的宏观信息，只要 reduce 任务向 JobTracker 获取对应的 map 输出位置就可以了。具体请参考上方的MapReduce工作原理。

**Reduce端流程：**  
  
reduce 会接收到不同 map 任务传来的有序数据，如果 reduce 接收到的数据较小，则会存在内存缓冲区中，直到数据量达到该缓存区的一定比例时对数据进行合并后溢写到磁盘上。随着溢写的文件越来越多，后台的线程会将他们合并成一个更大的有序的文件，可以为后面合并节省时间。这其实跟 map端的操作一样，都是反复的进行排序、合并，这也是 Hadoop 的灵魂所在，但是如果在 map 已经压缩过，在合并排序之前要先进行解压缩。合并的过程会产生很多中间文件，但是最后一个合并的结果是不需要写到磁盘上，而是可以直接输入到 reduce 函数中计算，每个 reduce 对应一个输出结果文件。

------------------开启虚拟机

D:

cd "D:\developer\vm\_workstation"

vmrun -T ws start "E:\VMData\hdp01\hdp01.vmx" nogui

vmrun -T ws start "E:\VMData\linux122.vmx" nogui

vmrun -T ws start "E:\VMData\linux123\linux123.vmx" nogui

------------------关闭虚拟机

d:

cd "D:\developer\vm\_workstation"

vmrun -T ws stop "E:\VMData\hdp01\hdp01.vmx"

vmrun -T ws stop "E:\VMData\linux122.vmx"

vmrun -T ws stop "E:\VMData\linux123\linux123.vmx"

群起3个节点上的Hadoop，每次启动虚拟机后都得执行：

[root@linux121 hadoop-2.9.2]# start-dfs.sh

[root@linux121 hadoop-2.9.2]# mr-jobhistory-daemon.sh start historyserver

[root@linux121 hadoop-2.9.2]# jps

[root@linux123 hadoop-2.9.2]$ start-yarn.sh

[root@linux123 hadoop-2.9.2]$ jps

cd /root

hadoop jar wc.jar com.lagou.mr.wc.WordCountDriver /wcinput /wcouput

ntpd -g -q

启动Flume Agent:

$FLUME\_HOME/bin/flume-ng agent --name a2 \

--conf-file $FLUME\_HOME/conf/flume-exec-hdfs.conf \

-Dflume.root.logger=INFO,console

tail -F /tmp/root/hive.log