**大数据面试题**

一、多项选择题（共5道题）

1.下列哪个是 Hadoop 运行的模式

A)单机版

B)伪分布式

C)分布式

答案：a,b,c

备注：常识题。

2.关于 SecondaryNameNode 哪项是正确的?

A)它是 NameNode 的热备

B)它对内存没有要求

C)它的目的是帮助 NameNode 合并编辑日志，减少 NameNode 启动时间

D)SecondaryNameNode 应与 NameNode 部署到一个节点

答案：C

3.下面哪个程序负责 HDFS 数据存储。

A)NameNode

B)Jobtracker

C)Datanode

D)secondaryNameNode

E)tasktracker

答案：C

**4．**HDfS 中的 block 默认保存几份?   
A)3 份   
B)2 份   
C)1 份   
D)不确定   
答案：A   
提示：常识题，block默认副本数是3（在hdfs-site.xml文件中设置dfs.replication）。

**5.在hadoop中定义的主要公用InputFormats中，哪一个是默认值：**   
(A)TextInputFormat   
(B)KeyValueInputFormat   
(C)SequenceFileInputFormat

答案：A

二、分析题（共12道题）

1、fsimage和edit的区别？  
  大家都知道namenode与secondary namenode 的关系，当他们要进行数据同步时叫做checkpoint时就用到了fsimage与edit，fsimage是保存最新的元数据的信息，当fsimage数据到一定的大小事会去生成一个新的文件来保存元数据的信息，这个新的文件就是edit，edit会回滚最新的数据。

2、datanode 首次加入 cluster 的时候，如果 log 报告不兼容文件版本，那需要namenode 执行格式化操作，这样处理的原因是？  
  1）这样处理是不合理的，因为那么 namenode 格式化操作，是对文件系统进行格式化，namenode 格式化时清空 dfs/name 下空两个目录下的所有文件，之后，会在目录 dfs.name.dir 下创建文件。  
  2）文本不兼容，有可能时 namenode 与 datanode 的 数据里的 namespaceID、clusterID 不一致，找到两个 ID 位置，修改为一样即可解决。

3、简单叙述如何安装开源版本Hadoop，无需完整步骤

1.创建hadoop用户.

2.setup修改IP.

3.安装javaJdk，并且修改etc/profile文件，配置java的环境变量.

4.修改host的文件域名

5.安装SSH免密码通信。

6.解压hadoop包.

7配置conf文件下的hadoop-env.sh.core-site.sh.mapre-site.sh.hdfs-site.sh

8配置hadoop环境变量

9.hadoop格式化hadoopnamenode-format

10.启动hadoopstart-all

4、请写出以下执行的命令

(1)杀死一个job

haoopjob-list拿到job-idhadoopjob-killjob-id

(2)删除hdfs上/tmp/aaa目录

hadoopfs-rmr/tmp/aaa

(3)加入一个新的存储节点和删除一个计算节点需要刷新到集群的状态命令

加新节点时：hadoop-deamon.shstartdatenode

Hadoop-deamon.shstarttasktracker

删除节点时:hadoopmradmin-refreshnodes

Hadoopdfsadmin-refreshnodes

5、Hadoop调度器有哪几种，说明你知道的

Fairschedular: 公平调度，所有job具有相同的资源。

Fifoschedular:默认，先进先出原则。

Capacityschedular：计算能力调度器，选择占用最小、优先高级的先执行。

6、简单叙述mapreduce中combiner，partition的作用.

combiner:实现的功能跟reduce差不多，接受map的值，经过计算后给reduce，他的key，value类型跟reduce完全一样，当reduce业务复杂时可以用，不过他貌似是操作本机的数据

Partition：将输出的结果分别保存在不同的文件中。

7、用mapreduce怎么处理数据倾斜的问题？

1.增加reduce的jvm内存

2.增加reduce个数

3.customerpartition

4.其他优化的讨论.

5.reducesortmerge排序算法的讨论

6.正在实现中的hiveskewedjoin.

7.pipeline

8.distinct

8、HDFS的常用的数据压缩算法有什么?

Hadoop中常用的压缩算法有bzip2、gzip、lzo、snappy，其中lzo、snappy需要操作系统安装native库才可以支持

9、说一说你对zookeeper的认识

ZooKeeper提供了一个简化并支持特定功能的分布式文件系统接口，加上数据同步，变更通知，客户端Cache等辅助机制。

基于这样的接口，用户可以自己在此之上构建逻辑，来实现各种分布式系统工作所需的各种功能，如配置管理，名称服务，Master选举，等。

10、Hive内部表与外部表的区别？

1、在导入数据到外部表，数据并没有移动到自己的数据仓库目录下，也就是说外部表中的数据并不是由它自己来管理的！

2、在删除表的时候，Hive将会把属于表的元数据和数据全部删掉；而删除外部表的时候，Hive仅仅删除外部表的元数据，数据是不会删除的！

11、Hadoop中job和Tasks之间的区别是什么?

答案：job是工作的入口，负责控制、追踪、管理任务，也是一个进程，包含maptask和reducetask。Tasks是map和reduce里面的步骤，主要用于完成任务，也是线程。

12、Map阶段结束后，Hadoop框架会处理：Partitioning,shuffle和sort,在这个阶段都会发生了什么?

　 答案：MR一共有四个阶段，splitmapshuffreduce在执行完map之后，可以对map的输出结果进行分区，

- 分区：这块分片确定到哪个reduce去计算(汇总)

- 排序：在每个分区中进行排序，默认是按照字典顺序。

- Group：在排序之后进行分组

三、设计题（共3道题）  
  1、采集nginx产生的日志，日志的格式为user  ip   time  url   htmlId  每天产生的文件的数据量上亿条，请设计方案把数据保存到HDFS上，并提供一下实时查询的功能（响应时间小于3s）  
A、某个用户某天访问某个URL的次数  
B、某个URL某天被访问的总次数   
实时思路是：使用Logstash + Kafka + Spark-streaming + Redis + 报表展示平台  
离线的思路是：Logstash + Kafka + Elasticsearch +  Spark-streaming + 关系型数据库  
A、B、数据在进入到Spark-streaming 中进行过滤，把符合要求的数据保存到Redis中

2、有 10 个文件，每个文件 1G，每个文件的每一行存放的都是用户的 query，每个文件的query 都可能重复。要求你按照 query 的频度排序。 还是典型的 TOP K 算法。  
  解决方案如下：   
    1）方案 1：   
    顺序读取 10 个文件，按照 hash(query)%10 的结果将 query 写入到另外 10 个文件（记为）中。这样新生成的文件每个的大小大约也 1G（假设 hash 函数是随机的）。 找一台内存在 2G 左右的机器，依次对用 hash\_map(query, query\_count)来统计每个query 出现的次数。利用快速/堆/归并排序按照出现次数进行排序。将排序好的 query 和对应的 query\_cout 输出到文件中。这样得到了 10 个排好序的文件（记为）。 对这 10 个文件进行归并排序（内排序与外排序相结合）。   
    2）方案 2：   
    一般 query 的总量是有限的，只是重复的次数比较多而已，可能对于所有的 query，一次性就可以加入到内存了。这样，我们就可以采用 trie 树/hash\_map等直接来统计每个 query出现的次数，然后按出现次数做快速/堆/归并排序就可以了。   
    3）方案 3：   
    与方案 1 类似，但在做完 hash，分成多个文件后，可以交给多个文件来处理，采用分布式的架构来处理（比如 MapReduce），最后再进行合并。

**3、在2.5亿个整数中找出不重复的整数，注，内存不足以容纳这2.5亿个整数。**

方案1：采用2-Bitmap(每个数分配2bit，00表示不存在，01表示出现一次，10表示多次，11无意义)进行，共需内存内存，还可以接受。然后扫描这2.5亿个整数，查看Bitmap中相对应位，如果是00变01，01变10，10保持不变。所描完事后，查看bitmap，把对应位是01的整数输出即可。

方案2：也可采用与第1题类似的方法，进行划分小文件的方法。然后在小文件中找出不重复的整数，并排序。然后再进行归并，注意去除重复的元素。