**大数据面试题**

**一、分析题（共20道题，每题4分，共80分**）

**1、列举Hadoop 运行的模式**

单机版 、伪分布式、分布式

**2、SecondaryNameNode 的作用？哪个程序负责 HDFS 数据存储？**

SecondaryNameNode的目的是帮助 NameNode 合并编辑日志，减少 NameNode 启动时间；

Datanode负责 HDFS 数据存储。

3、**在hadoop中定义的主要公用InputFormats中，默认值是？**

**默认值是**TextInputFormat。

**4、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。

**5、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。

**6、HDfS 中的 block 默认保存几份，在什么文件中设置dfs.replication？**

3份，在hdfs-site.xml文件中设置dfs.replication

**7、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

**8、spark streming在实时处理时会发生什么故障，如何停止，解决**

答：和Kafka整合时消息无序：

修改Kafka的ack参数，当ack=1时，master确认收到消息就算投递成功。ack=0时，不需要收到消息便算成功，高效不准确。sck=all，master和server都要受到消息才算成功，准确不高效。

StreamingContext.stop会把关联的SparkContext对象也停止，如果不想把SparkContext对象也停止的话可以把StremingContext.stop的可选参数stopSparkContext设为flase。一个SparkContext对象可以和多个streamingcontext对象关联。只要对前一个stremingcontext.stop(stopsparkcontext=false),然后再创建新的stremingcontext对象就可以了。

**9、mysql，mongodb，rides的端口。**

面试数据库介绍的再好，不知到默认端口，也证明你没有经验。mysql：3306，mongdb：27017，rides：6379。

**10、yarn的理解：**

YARN是Hadoop2.0版本引进的资源管理系统，直接从MR1演化而来。   
核心思想：将MR1中的JobTracker的资源管理和作业调度两个功能分开，分别由ResourceManager和ApplicationMaster进程实现。

ResourceManager：负责整个集群的资源管理和调度 ApplicationMaster：负责应用程序相关事务，比如任务调度、任务监控和容错等。 YARN的出现，使得多个计算框架可以运行在同一个集群之中。 1. 每一个应用程序对应一个ApplicationMaster。 2. 目前可以支持多种计算框架运行在YARN上面，比如MapReduce、storm、Spark、Flink

**11、为什么经常在Hadoop集群中删除或添加节点？**  
Hadoop最重要的功能是普通硬件。 但是，这会导致Hadoop集群中频繁的Datanode崩溃。  
Hadoop的另一个显着特点是数据量的快速增长使得易于扩展。  
因此，由于上述原因，管理员在Hadoop集群中添加/删除DataNode

**12、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。

**13、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！

**14、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。

**15、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

**16、jps命令在Hadoop中做了什么？**  
jps命令帮助我们检查Hadoop守护进程是否正在运行。 因此，它显示了在计算机上运行的所有Hadoop守护程序。 守护进程是Namenode，Datanode，ResourceManager，NodeManager等。

**17、./ etc / hosts中配置了什么以及它在设置Hadoop集群时的作用是什么？**./etc/hosts文件包含该主机的主机名及其IP地址。 它还将IP地址映射到主机名。 在hadoop集群中，我们将所有主机名（主服务器和从服务器）的IP地址存储在./etc/hosts中。 因此，我们可以轻松使用主机名而不是IP地址

**18、Hadoop的CLASSPATH如何在启动或停止Hadoop守护进程中发挥重要作用？**  
CLASSPATH包含启动/停止Hadoop守护程序所需的包含jar文件的所有目录。

**19、RDD中reduceBykey与groupByKey哪个性能好，为什么？**

　　reduceByKey：reduceByKey会在结果发送至reducer之前会对每个mapper在本地进行merge，有点类似于在MapReduce中的combiner。这样做的好处在于，在map端进行一次reduce之后，数据量会大幅度减小，从而减小传输，保证reduce端能够更快的进行结果计算。

　　groupByKey：groupByKey会对每一个RDD中的value值进行聚合形成一个序列(Iterator)，此操作发生在reduce端，所以势必会将所有的数据通过网络进行传输，造成不必要的浪费。同时如果数据量十分大，可能还会造成OutOfMemoryError。

通过以上对比可以发现在进行大量数据的reduce操作时候建议使用reduceByKey。不仅可以提高速度，还是可以防止使用groupByKey造成的内存溢出问题。

**20、Hibernate缓存包括哪几类，简述下特点和区别**

Hibernate一级缓存和Hibernate二级缓存：  
1）. Hibernate一级缓存又称为“Session的缓存”，它是内置的，不能被卸载。由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。在第一级缓存中，持久化类的每个实例都具有唯一的OID。  
2）.Hibernate二级缓存又称为“SessionFactory的缓存”，由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory不会启用这个插件。  
当Hibernate根据ID访问数据对象的时候，首先从Session一级缓存中查；查不到，如果配置了二级缓存，那么从二级缓存中查；如果都查不到，再查询数据库，把结果按照ID放入到缓存删除、更新、增加数据的时候，同时更新缓存。

**二、设计题（共3道题，共20分**）  
  **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）（5分）**  
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。（5分）**  
  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。（10分）**

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。