

第三阶段面试题

大数据阶段

1. 请列出正常工作的 **hadoop** 集群中 **hadoop** 都需要启动哪些进程，他们的作用分别是什么？

--namenode =>HDFS 的守护进程，负责维护整个文件系统，存储着整个文件系统的元数据信息，有 image+edit log namenode 不会持久化存储这些数据，而是在启动时重建这些数据。

--datanode =>是具体文件系统的工作节点，当我们需要某个数据，namenode 告诉我们要去哪里找，就直接和那个 DataNode 对应的服务器的后台进程进行通信，由 DataNode 进行数据的检索，然后进行具体的读/写操作

--secondarynamenode =>一个冗余的守护进程，相当于一个 namenode 的元数据的备份机制，定期的更新，和 namenode 进行通信，将 namenode 上的 image 和 edits 进行合并，可以作为 namenode 的备份使用

--resourcemanager =>是 yarn 平台的守护进程，负责所有资源的分配与调度，client 的请求由此负责，监控 nodemanager

--nodemanager => 是单个节点的资源管理，执行来自 resourcemanager 的具体任务和命令

2. 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 datanode

3. **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 默认 3 份

4. Hadoop 常见输入格式是什么？

- 三种广泛使用的输入格式是：
- 文本输入：Hadoop 中的默认输入格式。
- Key 值：用于纯文本文件
- 序列：用于依次读取文件

5. 说一下 namenode 的重要性。

namenode 的作用在 Hadoop 中非常重要。它是 Hadoop 的大脑，主要负责管理系统上的分配块，还为客户提出请求时的数据提供特定地址。

6. Flume 的 Source，sink，Channel 的作用？ Source 有什么类型？

- (1) Source 组件是专门用来收集数据的，可以处理各种类型、各种格式的日志数据，包括 avro、thrift、exec、jms、spooling directory、netcat、sequence generator、syslog、http、legacy
- (2) Channel 组件对采集到的数据进行缓存，可以存放在 Memory 或 File 中。
- (3) Sink 组件是用于把数据发送到目的地的组件，目的地包括 Hdfs、Logger、avro、thrift、ipc、file、Hbase、solr、自定义。

7. 说一下 Flume 的事务机制？

Flume 使用两个独立的事务分别负责从 Source 到 Channel(put)，以及从 Channel 到 Sink 的事件传递(take)。

比如 spooling directory source 为文件的每一行创建一个事件，一旦事务中所有的事件全部传递到 Channel 且提交成功，那么 Source 就将该文件标记为完成。

同理，事务以类似的方式处理从 Channel 到 Sink 的传递过程，如果因为某种原因使得事件无法记录，那么事务将会回滚。且所有的事件都会保持到 Channel 中，等待重新传递。

8. 简述 mapreduce 的运行原理

先将文件进行分割后，进行 map 操作，后面进行 shuffle 操作，分为 map 端 shuffle 和 reduce 端 shuffle，map 输出结果放在缓冲区，当缓存区到达一定阈值时，将其中数据 spill（也就是溢写）到磁盘，然后进行 partition, sort, combine 操作，这样多次 spill 后，磁盘上就会有多个文件，merge 操作将这些文件合并成一个文件，reduce 端 shuffle 从 map 节点拉取数据文

件，如果在内存中放得下，就直接放在内存中，每个 map 对应一块数据，当内存占用量达到一定程度时，启动内存时 merge，把内存中的数据输出到磁盘的一个文件上。如果在内存中放不下的话，就直接写到磁盘上。一个 map 数据对应一个文件，当文件数量达到一定阈值时，开始启动磁盘文件 merge，把这些文件合并到一个文件中。最后，把内存中的文件和磁盘上的文件进行全局 merge，形成一个最终端文件，做为 reduce 的输入文件。当然 merge 过程中会进行 sort,combine 操作。

9. 数据倾斜的原因？

key 分布不均匀
业务数据本身的特性
SQL 语句造成数据倾斜

10. 如何解决数据倾斜？

1.hive 设置 hive.map.aggr=true 和 hive.groupby.skewindata=true

2.有数据倾斜的时候进行负载均衡，当选项设定为 true,生成的查询计划会有两个 MR Job。第一个 MR Job 中，Map 的输出结果集合会随机分布到 Reduce 中，每个 Reduce 做部分聚合操作，并输出结果，这样处理的结果是相同 Group By Key 有可能被分发到不同的 Reduce 中，从而达到负载均衡的目的；第二个 MR Job 在根据预处理的数据结果按照 Group By Key 分布到 Reduce 中(这个过程可以保证相同的 Group By Key 被分布到同一个 Reduce 中)，最后完成最终的聚合操作。

3.SQL 语句调整：

- 1.选用 join key 分布最均匀的表作为驱动表。做好列裁剪和 filter 操作，以达到两表 join 的时候，数据量相对变小的效果。
- 2.大小表 Join： 使用 mapjoin 让小的维度表（1000 条以下的记录条数）先进内存。在 Map 端完成 Reduce。
- 3.大表 Join 大表：把空值的 Key 变成一个字符串加上一个随机数，把倾斜的数据分到不同的 reduce 上，由于 null 值关联不上，处理后并不影响最终的结果。
- 4.count distinct 大量相同特殊值：count distinct 时，将值为空的情况单独处理，如果是计算 count distinct，可以不用处理，直接过滤，在做后结果中加 1。如果还有其他计算，需要进行 group by，可以先将值为空的记录单独处理，再和其他计算结果进行 union。

11. Hive 中的拍讯关键字有哪些？

sort by ： 不是全局排序，其在数据进入 reducer 前完成排序

order by ： 会对输入做全局排序，因此只有一个 reducer(多个 reducer 无法保证全局有序)。只有一个 reducer,会导致当输入规模较大时，需要较长的计算时间。

cluster by ： 当 distribute by 和 sort by 的字段相同时，等同于 cluster by.可以看做特殊的 distribute + sort

distribute by ： 按照指定的字段对数据进行划分输出到不同的 reduce 中

12. 海量数据分布在 100 台电脑中，想个办法高效统计出这批数据的 TOP10

方案 1:

- 1.在每台电脑上求出 TOP10，可以采用包含 10 个元素的堆完成(TOP10 小，用最大堆，TOP10 大，用最小堆)。
- 2.比如求 TOP10 大，我们首先取前 10 个元素调整成最小堆，如果发现，然后扫描后面的数据，并与堆顶元素比较，如果比堆顶元素大，那么用该元素替换堆顶，然后再调整为最小堆。
- 3.最后堆中的元素就是 TOP10 大。

方案 2

- 1.求出每台电脑上的 TOP10 后，然后把这 100 台电脑上的 TOP10 组合起来，共 1000 个数据
- 2.再利用上面类似的方法求出 TOP10 就可以了。

13. Hive 中追加导入数据的 4 种方式是什么？请写出简要语法

从本地导入：`load data local inpath '/home/1.txt' (overwrite)into table student;`

从 Hdfs 导入：`load data inpath '/user/hive/warehouse/1.txt' (overwrite)into table student;`

查询导入：`create table student1 as select * from student;`(也可以具体查询某项数据)

查询结果导入：`insert (overwrite) into table staff select * from track_log;`

14. Hive 导出数据有几种方式？如何导出数据

1.用 insert overwrite 导出方式

导出到本地：

```
insert overwrite local directory '/home/robot/1/2' row format delimited fields terminated by '\t' select * from staff;(递归创建目录)
```

导出到 HDFS

```
insert overwrite directory '/user/hive/1/2' row format delimited fields terminated by '\t' select * from staff;
```

2.Bash shell 覆盖追加导出

例如：`$ bin/hive -e "select * from staff;" > /home/z/backup.log`

3.Sqoop 把 hive 数据导出到外部

15. hive 内部表和外部表区别

- 1.创建表时：创建内部表时，会将数据移动到数据仓库指向的路径；若创建外部表，仅

记录数据所在的路径，不对数据的位置做任何改变。

2.删除表时：在删除表的时候，内部表的元数据和数据会被一起删除，而外部表只删除元数据，不删除数据。这样外部表相对来说更加安全些，数据组织也更加灵活，方便共享源数据。

16. 分区和分桶的区别？

分区

1.是指按照数据表的某列或某些列分为多个区，区从形式上可以理解为文件夹，比如我们要收集某个大型网站的日志数据，一个网站每天的日志数据存在同一张表上，由于每天会生成大量的日志，导致数据表的内容巨大，在查询时进行全表扫描耗费资源非常多。

2.那其实这个情况下，我们可以按照日期对数据表进行分区，不同日期的数据存放在不同的分区，在查询时只要指定分区字段的值就可以直接从该分区查找

分桶

1.分桶是相对分区进行更细粒度的划分。

2.分桶将整个数据内容安装某列属性值得 hash 值进行区分，如要按照 name 属性分为 3 个桶，就是对 name 属性值的 hash 值对 3 取模，按照取模结果对数据分桶。

3.如取模结果为 0 的数据记录存放到一个文件，取模为 1 的数据存放到一个文件，取模为 2 的数据存放到一个文件

17. 请说明什么是传统的消息传递方法？

·排队：在队列中，一组用户可以从服务器中读取消息，每条消息都发送给其中一个人。

·发布-订阅：在这个模型中，消息被广播给所有的用户。

18. 请说明 Kafka 相对传统技术有什么优势？

快速:单一的 Kafka 代理可以处理成千上万的客户端，每秒处理数兆字节的读写操作。

可伸缩:在一组机器上对数据进行分区和简化，以支持更大的数据

持久:消息是持久性的，并在集群中进行复制，以防止数据丢失。

设计:它提供了容错保证和持久性

19. 解释 Kafka 的用户如何消费信息？

在 Kafka 中传递消息是通过使用 sendfile API 完成的。它支持将字节从套接口转移到磁盘，通过内核空间保存副本，并在内核用户之间调用内核。

20. 解释如何提高远程用户的吞吐量？

如果用户位于与 broker 不同的数据中心，则可能需要调优套接口缓冲区大小，以对长

网络延迟进行摊销。

21 解释一下，在数据制作过程中，你如何能从 Kafka 得到准确的信息？

在数据中，为了精确地获得 Kafka 的消息，你必须遵循两件事：在数据消耗期间避免重复，在数据生产过程中避免重复。

这里有两种方法，可以在数据生成时准确地获得一个语义：

- 1.每个分区使用一个单独的写入器，每当你发现一个网络错误，检查该分区中的最后一条消息，以查看您的最后一次写入是否成功
- 2.在消息中包含一个主键(UUID 或其他)，并在用户中进行反复制

22 解释如何减少 ISR 中的扰动？broker 什么时候离开 ISR？

ISR 是一组与 leaders 完全同步的消息副本，也就是说 ISR 中包含了所有提交的消息。ISR 应该总是包含所有的副本，直到出现真正的故障。如果一个副本从 leader 中脱离出来，将会从 ISR 中删除。

23 Kafka 为什么需要复制？

Kafka 的信息复制确保了任何已发布的消息不会丢失，并且可以在机器错误、程序错误或更常见些的软件升级中使用。

24 如果副本在 ISR 中停留了很长时间表明什么？

如果一个副本在 ISR 中保留了很长一段时间，那么它就表明，跟踪器无法像在 leader 收集数据那样快速地获取数据。

25 请说明如果首选的副本不在 ISR 中会发生什么？

如果首选的副本不在 ISR 中，控制器将无法将 leadership 转移到首选的副本。

26 有可能在生产后发生消息偏移吗？

在大多数队列系统中，作为生产者的类无法做到这一点，它的作用是触发并忘记消息。broker 将完成剩下的工作，比如使用 id 进行适当的元数据处理、偏移量等。

作为消息的用户，你可以从 Kafka broker 中获得补偿。如果你注视 SimpleConsumer 类，你会注意到它会获取包括偏移量作为列表的 MultiFetchResponse 对象。此外，当你对 Kafka 消息进行迭代时，你会拥有包括偏移量和消息发送的 MessageAndOffset 对象。

27 hbase 的特点是什么？

- (1) Hbase 一个分布式的基于列式存储的数据库,基于 Hadoop 的 hdfs 存储, zookeeper 进行管理。
- (2) Hbase 适合存储半结构化或非结构化数据, 对于数据结构字段不够确定或者杂乱无章很难按一个概念去抽取的数据。
- (3) Hbase 为 null 的记录不会被存储。
- (4) 基于的表包含 rowkey, 时间戳, 和列族。新写入数据时, 时间戳更新, 同时可以查询到以前的版本。
- (5) hbase 是主从架构。hmaster 作为主节点, hregionserver 作为从节点。

28 Hbase 和 hive 有什么区别？

Hive 和 Hbase 是两种基于 Hadoop 的不同技术--Hive 是一种类 SQL 的引擎, 并且运行 MapReduce 任务, Hbase 是一种在 Hadoop 之上的 NoSQL 的 Key/value 数据库。当然, 这两种工具是可以同时使用的。就像用 Google 来搜索, 用 faceBook 进行社交一样, Hive 可以用来进行统计查询, HBase 可以用来进行实时查询, 数据也可以从 Hive 写到 Hbase, 设置再从 Hbase 写回 Hive。

29 描述 Hbase 的 rowKey 的设计原则。

Rowkey 长度原则

Rowkey 是一个二进制码流, Rowkey 的长度被很多开发者建议说设计在 10~100 个字节, 不过建议是越短越好, 不要超过 16 个字节。

原因如下:

(1) 数据的持久化文件 HFile 中是按照 KeyValue 存储的, 如果 Rowkey 过长比如 100 个字节, 1000 万列数据光 Rowkey 就要占用 $100 \times 1000 \text{ 万} = 10 \text{ 亿}$ 个字节, 将近 1G 数据, 这会极大影响 HFile 的存储效率;

(2) MemStore 将缓存部分数据到内存, 如果 Rowkey 字段过长内存的有效利用率会降低, 系统将无法缓存更多的数据, 这会降低检索效率。因此 Rowkey 的字节长度越短越好。

(3) 目前操作系统都是 64 位系统, 内存 8 字节对齐。控制在 16 个字节, 8 字节的整数倍利用操作系统的最佳特性。

Rowkey 散列原则

如果 Rowkey 是按时间戳的方式递增, 不要将时间放在二进制码的前面, 建议将 Rowkey 的高位作为散列字段, 由程序循环生成, 低位放时间字段, 这样将提高数据均衡分布在每个 Regionserver 实现负载均衡的几率。如果没有散列字段, 首字段直接是时间信息将产生所有新数据都在一个 RegionServer 上堆积的热点现象, 这样在做数据检索的时候负载将会集中在个别 RegionServer, 降低查询效率。Rowkey 唯一原则
必须在设计上保证其唯一性。

30 描述 Hbase 中 scan 和 get 的功能以及实现的异同.

HBase 的查询实现只提供两种方式:

- 1、按指定 RowKey 获取唯一一条记录, get 方法 (org.apache.hadoop.hbase.client.Get) Get 的方法处理分两种: 设置了 ClosestRowBefore 和没有设置的 rowlock .主要是用来保证行的事务性, 即每个 get 是以一个 row 来标记的. 一个 row 中可以有很多 family 和 column.
- 2、按指定的条件获取一批记录, scan 方法(org.apache.Hadoop.hbase.client.Scan) 实现条件查询功能使用的就是 scan 方式.
 - 1)scan 可以通过 setCaching 与 setBatch 方法提高速度(以空间换时间);
 - 2)scan 可以通过 setStartRow 与 setEndRow 来限定范围([start, end)start 是闭区间, end 是开区间)。范围越小, 性能越高。
- 3)、scan 可以通过 setFilter 方法添加过滤器, 这也是分页、多条件查询的基础。

31 请描述 Hbase 中 scan 对象的 setCache 和 setBatch 方法的使用.

为设置获取记录的列个数, 默认无限制, 也就是返回所有的列.每次从服务器端读取的行数, 默认为配置文件中设置的值。

32 请详细描述 Hbase 中一个 Cell 的结构

HBase 中通过 row 和 columns 确定的为一个存储单元称为 cell。

Cell: 由{row key, column(=<family> + <label>), version}唯一确定的单元。cell 中的数是没有类型的, 全部是字节码形式存储。

33 请描述如何解决 Hbase 中 region 太小和 region 太大带来的冲突.

Region 过大会发生多次 compaction, 将数据读一遍并重写一遍到 hdfs 上, 占用 io, region 过小会造成多次 split, region 会下线, 影响访问服务, 调整 hbase.hregion.max.filesize 为 256m.

34 HBase 中实现了两种 compaction 的方式: minor and major. 这两种 compaction 方式的区分是什么?

- 1、Minor 操作只用来做部分文件的合并操作以及包括 minVersion=0 并且设置 ttl 的过期版本清理, 不做任何删除数据、多版本数据的清理工作。
- 2、Major 操作是对 Region 下的 HStore 下的所有 StoreFile 执行合并操作, 最终的结果是整理合并出一个文件。

35 spark 在什么场景比不上 MapReduce?

Spark 在内存中处理数据，需要很大的内存容量。如果 Spark 与其它资源需求型服务一同运行在 YARN 上，又或者数据块太大以至于不能完全读入内存，此时 Spark 的性能就会有大的降低，此时 Spark 可能比不上 MapReduce。当对数据的操作只是简单的 ETL 的时候，Spark 比不上 MapReduce。

36 RDD 五大特性?

1、RDD 是由一系列的分区组成。2、操作一个 RDD 实际上操作的是 RDD 的所有分区。3、RDD 之间存在各种依赖关系。4、可选的特性，key-value 型的 RDD 是通过 hash 进行分区。5、RDD 的每一个分区在计算时会选择最佳的计算位置。

37 spark 能都取代 hadoop 吗?

Spark 是一个计算框架，它没有自己的存储，它的存储还得借助于 HDFS，所以说 Spark 不能取代 Hadoop,要取代也是取代 MapReduce

38 Spark 的特点?

1) 快速：基于内存的计算比 MapReduce 快 100 倍，基于磁盘快 10 倍。
2) 易用：编写一个 spark 的应用程序可以使用 Java, Scala, Python, R，这就使得我们的开发非常地灵活。并且，对比于 MapReduce,spark 内置了 80 多个高级操作，这使得开发十分高效和简单。
3) 运行范围广：spark 可以运行在 local、yarn、mesos、standalone、kubernetes 等多种平台之上。它可以访问诸如 HDFS, Cassandra, HBase, [S3](#) 等多种多样的数据源。
4) 通用：spark 提供了 SparkSQL、SparkStreaming、GraphX、MLlib 等一系列的分析工具。

39 大数据流式处理框架对比：Storm vs Spark Streaming?

a. Spark Streaming 最低可在 0.5 秒~2 秒内做一次处理，而 Storm 最快可达到 0.1 秒，在实时性和容错性上，Spark Streaming 不如 Storm。
b. Spark Streaming 的集成性优于 Storm, 可以通过 RDD 无缝对接 Spark 上的所有组件，还可以很容易的与 kafka, flume 等分布式框架进行集成。
c. 在数据吞吐量上，Spark Streaming 要远远优于 Storm。
综上所述，Spark Streaming 更适用于大数据流式处理。

40 cache()和 persist()方法的区别?

cache()在源码底层调用的是 persist()。

41 hdfs 写文件的步骤？

- (1)client 向 NameNode 申请上传.../xxx.txt 文件
- (2)NN 向 client 响应可以上传文件
- (3)Client 向 NameNode 申请 DataNode
- (4)NN 向 Client 返回 DN1,DN2,DN3
- (5)Client 向 DN1,DN2,DN3 申请建立文件传输通道
- (6)DN3,DN2,DN1 依次响应连接
- (7)Client 向 DN1 上传一个 block，DN1 向 DN2,DN3 冗余文件

42 hdfs 读取文件步骤？

- (1)client 向 NN 请求下载.../xxx.txt 文件
- (2)NN 向 client 返回文件的元数据
- (3)Client 向 DN1 请求访问读数据 blk_1
- (4)DN1 向 Client 传输数据
- (5)Client 向 DN2 请求访问读数据 blk_2
- (6)DN2 向 Client 传输数据

43 fsimage 和 edit 的区别？

当 NN,SN 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 数据到一定的大小时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。

44 运行 hadoop 集群需要哪些守护进程？

DataNode,NameNode,TaskTracker 和 JobTracker 都是运行 Hadoop 集群需要的守护进程。

45 hadoop 的 TextInputFormat 作用是什么，如何自定义实现？

InputFormat 会在 map 操作之前对数据进行两方面的预处理。

- 1.是 getSplits，返回的是 InputSplit 数组，对数据进行 Split 分片，每片交给 map 操作一次。
- 2.是 getRecordReader，返回的是 RecordReader 对象，对每个 Split 分片进行转换为 key-value 键值对格式传递给 map 常用的 InputFormat 是 TextInputFormat，使用的是 LineRecordReader 对每个分片进行键值对的转换，以行偏移量作为键，行内容作为值。自定义类继承 InputFormat 接口，重写 createRecordReader 和 isSplittable 方法在 createRecordReader 中可以自定义分隔符。

46 Linux 使用什么命令查看磁盘使用空间？ 空闲空间呢？

```
df -hl
```

文件系统 容量 已用 可用 已用% 挂载点

```
Filesystem Size Used Avail Use% Mounted on /dev/hda2 45G 19G 24G 44% /  
/dev/hda1 494M 19M 450M 4% /boot
```

47 进程的查看和调度分别使用什么命令？

进程查看的命令是 `ps` 和 `top`.

进程调度的命令有 `at`, `crontab`, `batch`, `kill`.

48 说一下 Zookeeper 工作原理

Zookeeper 的核心是原子广播，这个机制保证了各个 Server 之间的同步。实现这个机制的协议叫做 Zab 协议。Zab 协议有两种模式，它们分别是恢复模式（选主）和广播模式（同步）。当服务启动或者在领导者崩溃后，Zab 就进入了恢复模式，当领导者被选举出来，且大多数 Server 完成了和 leader 的状态同步以后，恢复模式就结束了。状态同步保证了 leader 和 Server 具有相同的系统状态。

49 zookeeper 是如何保证事务的顺序一致性的？

zookeeper 采用了递增的事务 id 来标识，所有的 proposal（提议）都在被提出的时候加上了 zxid，zxid 实际上是一个 64 位的数字，高 32 位是 epoch（时期；纪元；世；新时代）用来标识 leader 是否发生改变，如果有新的 leader 产生出来，epoch 会自增，低 32 位用来递增计数。当新产生 proposal 的时候，会依据数据库的两阶段过程，首先会向其他的 server 发出事务执行请求，如果超过半数的机器都能执行并且能够成功，那么就会开始执行。

50 Zookeeper 下 Server 工作状态

每个 Server 在工作过程中有三种状态：

LOOKING：当前 Server 不知道 leader 是谁，正在搜寻

LEADING：当前 Server 即为选举出来的 leader

FOLLOWING：leader 已经选举出来，当前 Server 与之同步

51 Zookeeper 同步流程

- 1、Leader 等待 server 连接；
- 2、Follower 连接 leader，将最大的 zxid 发送给 leader；

- 3、Leader 根据 follower 的 zxid 确定同步点；
- 4、完成同步后通知 follower 已经成为 uptodate 状态；
- 5、Follower 收到 uptodate 消息后，又可以重新接受 client 的请求进行服务了。

52 zookeeper 负载均衡和 nginx 负载均衡区别

zk 的负载均衡是可以调控，nginx 只是能调权重，其他需要可控的都需要自己写插件；但是 nginx 的吞吐量比 zk 大很多，应该说按业务选择用哪种方式。

53 storm 跟 spark 之间的对比

Storm 纯实时，来一条数据，处理一条数据；spark Streaming 准实时，对一个时间段内的数据收集起来，作为一个 RDD，再处理

Storm 毫秒级的延迟，Spark Streaming 秒级的延迟。

Storm 吞吐量低，Spark Streaming 吞吐量高

Storm 事务机制支持并且完善，Spark Streaming 支持，但不够完善。

Storm 健壮性/容错性非常强（Zookeeper，Acker），Spark Streaming 健壮性/容错性一般（Checkpoint，WAL）