前言

> 我努力了这一年，不仅仅是为了逼岁月回头。

我是年前离职的，没想到这个突如其来的疫情，完全将面试升级为地狱难度，焦虑、烦躁、失眠，是过去一个月的主旋律。

自四月上旬投第一封简历开始，前一周完全是在欸打，最气的面试的小公司，还没到技术面，HR对我说了一句：“ 18届，我们最高只能给11K。” 我：“？？？？”

说实话，我不甘心，真的。毕竟在过去一年，我很少有早于凌晨睡的，每天坚持对技术进行复盘，然后不断的学习新东西，我的预期自然也远远不止于此。

从一开始的焦虑、迷茫，到对自己的技术产生的深深的怀疑。

幸亏，身边一帮小伙伴互相打气，然后还有像敖丙（为何丙丙和我一样大，能这么优秀 ┗|｀O′|┛ ）他的一些工作和生活经历给了我很多共鸣以及给了我一个努力的方向吧。

接下来，陆陆续续面试了中软国际、翼海云峰、华讯方舟、明略科技、赛意信息、浙江大华、中新赛克、华为OD、焦点科技、浩鲸、阿里云...近20家南京/杭州的大中小厂，最终成功上岸，我就大数据部分的面试题做一个总结，希望能对大家有所帮助。

一、面试准备

面试前，我花了很多时间，对项目进行了梳理，尤其在业务数仓的分层和多维数据模型设计这块。整个项目的业务流程、数据流向我用一张白纸进行了梳理，**数据收集 + 数仓建设+数据建模+数据清洗 + 数据转换+ 特征提取+算法建模+数据展示**，我觉得对自己做过或者参与的项目，在准备面试前，做一次系统的复盘，是必不可少的。

大数据技术栈这一块，可以按照B站某谷的一些视频进行复习，毕竟一些理论和架构的东西，有时是需要花时间记忆和理解的，我放一张图，大家看看自己能了解多少：

二、Hadoop

1、介绍 MapReduce 的运行过程 ， Suffer 过程

> 如果在现场，我可以手绘 MapReduce 从 InputFormat 到 OutputFormat 的流程，一边画图一边说。如果讲到环形缓冲区那里，是不是有很多调优的方式、combiner 也可以考虑讲一下。

2、Hadoop 集群的搭建过程

> 至少自己集群的配置、框架的技术选型是不是都要清楚的明明白白。

3、Hadoop 优化

> 1、HDFS 小文件的影响 、输入输入时的小文件的处理 2、Map 阶段 和 Reudce 阶段的调优 3、数据压缩（LZO \Snappy） 和 存储优化(Orcfile) 关于压缩怎么配的，几种存储格式有什么区别是不是都要搞清楚

4、Hadoop集群HA实现

5、Hadoop 调度器

> FIFO 、Capacity Scheduler（容量调度器）和 Fair Sceduler（公平调度器）三种需要区分清楚，还有在实际开发环境中，一般不用FIFO哦。

6、Hadoop 解决数据倾斜方法

> 1、提前在 map 进行 combine，减少传输的数据量 2、导致数据倾斜的 key 加盐、提升 Reducer 并行度 ...

7、Hadoop读文件和写文件流程

8、Yarn 的 Job 提交流程

> 步骤很多，理理清楚然后再由条理的进行回答。

三、Hive

1、 Hive 和关系型数据库比较

> 数据存储位置、数据更新、执行延迟、数据规模等方面来比较。

2、Hive 元数据管理

> hive表中的数据是HDFS上的文件，可是hive怎么知道这些文件的内容都对应哪个字段，对应哪个分区呢？就是hive的元数据管理着这一切。通常在hive-site.xml中的元数据库配置成MySQL，替换Derby。

3、有没有遇到数据倾斜的问题（场景、解决方式）

> 常规的数据解决方式，结合业务，随便讲个三四种不过分吧。

4、Hive 两种类型的权限控制方式

5、Hive UDF，UDTF，UDAF，窗口函数（row_number, rank,cube,rollup,lag,lead）

6、Hive 的调优

> 1、压缩存储优化 2、表设计优化 3、SQL参数优化 4、SQL语句优化 分四个方向，大概十几种优化的方式，自己都得做些了解吧

7、Hive 分区和分桶的区别，内部表和外部表的区别，怎么进行动态分区

8、Hive 几种存储方式的区别?

> ORC：行分块，列式存储，压缩快，存取快，压缩率最高，RCfile升级版。 然后再和其他三种存储方式比较一下。

四、Flume

1、Flume 组成，Put 事务，Take 事务

> Source、Channel、Sink ，想想Flume的架构。 Taildir Source、Memory Channel什么的，各自适合用在什么场景下，有什么区别。

2、Flume 自定义拦截器

> 可以在讲项目时讲，算是一个小亮点，可以自定义ETL 拦截器和区分类型拦截器等等

3、Flume Channel 选择器

> Replicating Channel Selector (default)和Multiplexing Channel Selector 这两种Selector的区别是:Replicating 会 将source过来的events发往所有channel,而 Multiplexing可以选择该发往哪些Channel。

4、Flume 调优

> 要知道flume-ng agent包括source、channel、sink三个部分，这三部分都运行在JVM上，而JVM运行在linux操作系统之上。因此，对于flume的性能调优，就是对这三部分及影响因素调优。

五、Kafka

1、Kafka 的架构

2、关于 Kafka 为什么这么快

> 顺序写入、Memory Mapped Files、零拷贝（Zero Copy）、批量发送和数据压缩。

3、Kafka 和其他消息队列的区别

4、Kafka 如何保证消息队列不丢失?

> ACK 机制 、 设置分区、关闭 unclean leader 选举等等。

5、Kafka 消息数据积压，Kafka 消费能力不足怎么处理

> 如果是 Kafka 消费能力不足，则可以考虑增加 Topic 的分区数，并且同时提升消费 组的消费者数量，消费者数=分区数。（两者缺一不可） 如果是下游的数据处理不及时：提高每批次拉取的数量。批次拉取数据过少（拉取 数据/处理时间<生产速度），使处理的数据小于生产的数据，也会造成数据积压。

6、Kafka producer consumer怎么实现at most once和exactly once（幂等计算和事务）

7、Kafka 高可用怎么实现的?

> 副本数据同步策略、ISR、OSR、Leader 选举机制（它的和Zookeeper的半数选举机制可不同哦）。

8、Kafka 数据重复

> 幂等性+ack-1+事务 Kafka 数据重复，可以再下一级：SparkStreaming、redis 或者 hive 中 dwd 层去重，去重的手段：分组、按照 id 开窗只取第一个值。

六、HBase

1、RowKey 怎么设计的?

> 三个设计原则，id+时间戳反转 什么的，结合你的业务场景讲讲。

2、描述 HBase 中 scan 和 get 的功能以及实现的异同?

3、在HBase 中，是允许设置多个列簇的，但是为什么在实际生产中会设置很少的列簇呢?

> 1、列簇的数量对flush的影响 2、列簇的数量对split的影响 3、列簇的数量对compaction的影响 4、列簇的数量对HDFS的影响 5、列簇的数量对RegionServer内存的影响 。 根据实际生产需求，能够用一个列簇解决的就尽量用一个列簇，当两个列簇的数量相差悬殊时，可以将其两个列簇的数据拆分为两个表的单个列簇。

4、HBase 的存储格式

> HBase中的每张表都通过行键按照一定的范围被分割成多个子表（HRegion），默认一个HRegion超过256M就要被分割成两个，由HRegionServer管理，管理哪些HRegion由HMaster分配。

5、HBase 的读写流程

6、HBase 的优化

> 1、预分区 2、rowkey 优化 3、减少 Column Family 数量

7、关于HBase 数据热点的问题

七、Spark

1、 Spark 有几种部署方式？请分别简要论述

> Spark 的运行模式有 Local(也称单节点模式），Standalone(集群模式），Spark on Yarn(运行在Yarn上）有 yarn-client 和 yarn-cluster 两种模式，主要区别在于：Driver 程序的运行节点。

2、Spark on yarn cluster 作业提交的流程

3、Spark 提交作业参数

4、如何理解 Spark 中的血统概念（RDD）

5、Spark 调优

> coalesce 和 repartition / BroadCast join 广播 join / 控制 Spark reduce 缓存 调优 shuffle / 使用高性能算子 等等。

6、Spark 划分任务

> RDD任务切分中间分为：Application、Job、Stage和Task，再详细讲述各自的联系。

7、Spark 宽窄依赖，reducebykey 和 groupbykey 的性能谁高?

> map、flatmap、union、filter ----> 窄依赖 groupByKey，reduceByKey，sortByKey,join 各种Bykey都是shuffle阶段 -----> 宽依赖 reducebyKey会先在本地机器上进行局部聚合，然后在移动数据，进行全局聚合 ---> 性能更好

8、分别简述 Spark 中的缓存机制（cache 和 persist）与checkpoint 机制，并指出两者的区别与联系

> 都是做 RDD 持久化的 cache:内存，不会截断血缘关系，使用计算过程中的数据缓存。checkpoint：磁盘，截断血缘关系，在 ck 之前必须没有任何任务提交才会生效，ck 过 程会额外提交一次任务。

9、Spark 的缓存级别

> memory_only、disk_only、memory_Anddisk_only 像cache() 默认 memory_only 什么的。

10、某个 task 莫名其妙内存溢出的情况

> 这种情况下去定位出问题的代码就比较容易了。我建议直接看 yarn-client 模式下本地log 的异常栈，或者是通过 YARN 查看 yarn-cluster 模式下的 log 中的异常栈。一般来说，通过异常栈信息就可以定位到你的代码中哪一行发生了内存溢出。然后在那行代码附近找找，一般也会有 shuffle 类算子，此时很可能就是这个算子导致了数据倾斜。 但是大家要注意的是，不能单纯靠偶然的内存溢出就判定发生了数据倾斜。因为自己编 写的代码的 bug，以及偶然出现的数据异常，也可能会导致内存溢出。因此还是要按照上面所讲的方法，通过 Spark Web UI 查看报错的那个 stage 的各个 task 的运行时间以及分配的数据量，才能确定是否是由于数据倾斜才导致了这次内存溢出。

11、Spark 数据倾斜

> 使用 Hive ETL 预处理数据 、 过滤少数导致倾斜的 key 、 提高 shuffle 操作的并行度 、两阶段聚合（局部聚合+全局聚合） 、 将 reduce join 转为 map join 、使用随机前缀和扩容 RDD 进行 join 等等，方法很多，大家可以再深入的了解。

12、Spark 内存溢出

> 1 加内存， 简单粗暴 2 将rdd的数据写入磁盘不要保存在内存之中 3 如果是collect操作导致的内存溢出， 可以增大 Driver的 memory 参数

13、简述 Spark 中共享变量（广播变量和累加器）的基本原理与用途

> 累加器（accumulator）是 Spark 中提供的一种分布式的变量机制，其原理类似于 mapreduce，即分布式的改变，然后聚合这些改变。累加器的一个常见用途是在调试时对作业执行过程中的事件进行计数。而广播变量用来高效分发较大的对象。

14、简述 SparkSQL 中 RDD、DataFrame、DataSet 三者的区别与联系?

15、Spark Streaming 控制每秒消费数据的速度

> 通过 spark.streaming.kafka.maxRatePerPartition 参数来设置 Spark Streaming 从 kafka 分区每秒拉取的条数。

16、Spark Streaming 背压机制

> 把 spark.streaming.backpressure.enabled 参数设置为 ture,开启背压机制后 Spark Streaming 会根据延迟动态去 kafka 消费数据,上限由 spark.streaming.kafka.maxRatePerPartition 参数控制，所以两个参数一般会一起使用。

17、SparkStreaming 有哪几种方式消费 Kafka 中的数据，它们之间的区别是什么?

> 基于 Receiver 的方式 基于 Direct 的方式 -----> 简化并行读取 高性能

八、数仓

1、数据仓库的模型设计

> 结合业务，对数仓设计的过程做个概述，例如我的就是常见的四层的一个模型，ODS、ODW... 层 ，这其中我对业务数据做了哪些操作，都得了然于心吧。

2、数仓质量怎么监控

> 数据质量管理系统，主键唯一、非空、数据波动。

3、业务建模、数据分析方法

4、有没有遇到数据倾斜的问题（场景、解决方式）

5、数仓规范设计哪些方面(字段、维度，存储压缩、数据保留机制)

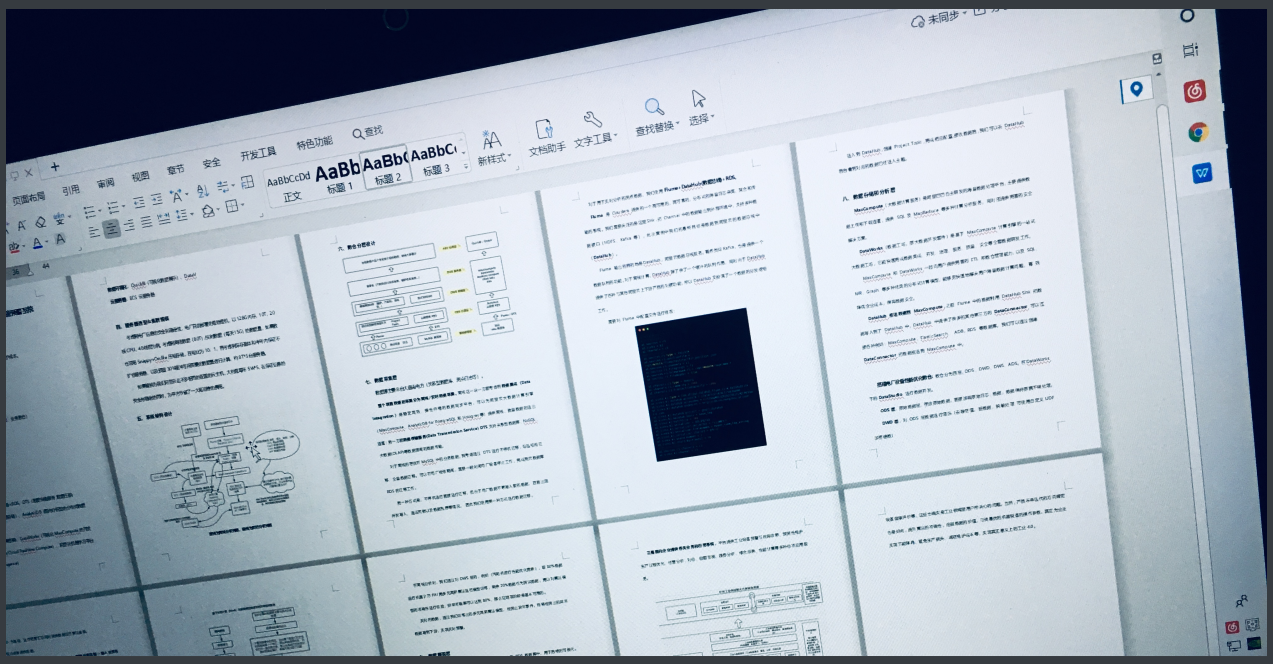
6、数仓有用到增量表、还是全量表？拉链表做过吗?

九、写在最后

毕业两年第一次跳槽，当年那只非计算机专业，误打误撞进了大数据门的小白，一路修行，磕磕绊绊。

这中间也曾焦虑过、失眠，凌晨四点爬起来Coding，又或者除夕夜那晚我还在肝面试题QAQ。时间久了也会自我怀疑，觉得自己这般努力值得吗?

但是一方面是对新技术的渴望，另一方面是来自房贷的压力，像是悬在我头上的达摩克里斯之剑，让我时刻保持清醒的头脑，不断学习。



最后，我还是确定了阿里系的Offer。凌晨肝了完了阿里面试官留下的最后的实验方案——将我的大数据项目迁移到阿里云的架构方案分析，提交过去能得到面试官的认可，也是非常庆幸的事情。

在马伯庸 《长安十二时辰》里看到一句话，非常喜欢，和大家共勉：

“祷以恒切， 盼以喜乐，苦以坚忍，必有所得”。