大数据面试反馈反思

面试反馈反思：

面试公司：特斯联-大数据开发工程师

1：yarn?

参考博客(Hadoop MapReduceV2(Yarn) 框架简介)：

https://www.ibm.com/developerworks/cn/opensource/os-cn-hadoop-yarn/

答题思路：解决MapReduce1.0版本的JobTracker/TaskTracker难于扩展的问题，解耦它的资源调度和任务的调度，因此产生了yarn，既MapReduce2.0版本：MapReduceV2或者叫Yarn

一个应用程序是如何在yarn上执行的？

2：Spark的运行模式

Local(N):N表示线程数

Spark on standalone

standalone-client:

tandalone-client:

spark on yarn

yarn-Client:

yarn-cluster:

3：机器学习：梯度下降

求解机器学习算法的模型参数，梯度下降（Gradient Descent）是最常采用的方法之一，另一种常用的方法是最小二乘法

在机器学习算法中，在最小化损失函数时，可以通过梯度下降法来一步步的迭代求解，得到最小化的损失函数，和模型参数值

面试公司：多牛传媒-多牛传媒spark工程师

多牛传媒面试题解析：

三人面试（人事，spark开发，部门总监），面试时间1.5小时

人事让介绍自己，为什么想换工作

答：请参考自我介绍，换工作的理由：1：找一个更大的平台，能够有更大的发展空间；2：准备买房结婚了，希望找一个薪资待遇更好的工作

总监问sparksql和sparkstreaming哪个比较熟

答：都还行，SparkSql的DataFrame或者DataSet和SparkStreaming的DStream都是基于SparkCore的，最终都会转化为Spark task执行。我们可以交流一下本质的东西SparkCore，而SparkCore的核心又是RDD。

开发问可以说一下sparkshuffle吗？

答：Spark的shuffle也是一处理问题的思想：分而治之。shuffle一般称为洗牌，一般会有Shuffle Write阶段和Shuffle Read阶段。在Spark中实现Shuffle的方式有两种，一种是HashShuffle，一种是SortShuffle。Ｓｈｕｆｆｌｅ的性能是影响Ｓｐａｒｋ应用程序性能的关键。Ｓｈｕｆｆｌｅ发生在Ｓｔａｇｅ之间，Ｓｔａｇｅ中用的ｐｉｐｌｉｎｅ的计算模式。

HashShuffle又有实现又有2种机制：

1：HashShuffle的普通机制，画图，HashShuffle的普通机制的问题

2：HashShuffle合并机制，画图，解决了的问题

SortShuffle实现也有2种机制：

1：SortShuffle的普通机制，出现的问题，画图

2：SortShuffle的ByPass机制，细节。

Spark Shuffle的数据位置定位和拉取数据的组件：

主：MapOutputTrackerMaster---存在Driver进程中

从：MapOutputTrackerWorker---存在Executor进程中

BlockManager组件：块管理者

BlockManagerMaster：存在Driver中

1：DiskStore：负责磁盘的管理

2：MemStore：负责内存的管理

3：ConnectionManager：负责连接其他的BlockManagerSlave

4：BlockTransforService ：负责数据的传输

Spark Shuffle的调优点：

1：Shuffle的选择

2：缓冲区的大小

3：拉去的数据量的大小

4：间隔时间重试次数

开发问缓存这块熟悉吗，介绍缓存级别

答：Spark的缓存机制是Spark优化的一个重要点，它将需要重复使用或者共用的RDD缓存在内存中，可以提高Spark的性能。

Spark的底层源码中使用StorageLevel来表示缓存机制，其中包括：使用内存，使用磁盘，使用序列化，使用堆外内存。在他的半生对象中基于这几种方式提供了一些实现：不使用缓存，Memory_Only

Disk_only,offHeap,分别都有相应的序列化，副本，组合的实现提供选择。持久化的级别StorageLevel可以自定义，但是一般不自定义。如何选择RDD的缓存级别的本质是在内存的利用率和CPU的利用率之间的权衡。

一般默认选择的是Memory_only,其次是Memery_only_Ser,再次是Memory_only_and_Disk

至于怎么选择你得自己权衡。

Ｓｐａｒｋ　ｓｈｕｆｆｌｅ本身也实现了缓存机制，有利于提高ｓｈｕｆｆｌｅ的性能

在外层的封装就是catch()和Persist()方法了，catch()的persist()的简化版本，如何选择持久化的级别就是刚刚说的了。

说一下cache和checkpoint的区别

答：要知道区别，首先要知道实现的原理和使用的场景

catche的就是将共用的或者重复使用的RDD按照持久化的级别进行缓存

checkpoint的是将业务场景非常长的逻辑计算的中间结果缓存到ＨＤＦＳ上，它的实现原理是
首先找到ｓｔａｇｅ最后的ｆｉｎａｌＲＤＤ，然后按照ＲＤＤ的依赖关系进行回溯，找到使用了ｃｈｅｃｋＰｏｉｎｔ的ＲＤＤ
然后标记这个使用了ｃｈｅｃｋＰｏｉｎｔ的ＲＤＤ
重新的启动一个线程来将ｃｈｅｃｋＰｏｉｎｔ之前的ＲＤＤ缓存到ＨＤＦＳ上面
最后将ＲＤＤ的依赖关系从ｃｈｅｃｋＰｏｉｎｔ的位置切断
知道了实现的原理和使用场景后我们就很容易的知道了ｃａｔｃｈ和ｃｈｅｃｋｐ的区别了

总监出好了sql题

A	B	C
1	0.5	9
0.9	1.5	2
3	0.7	0.8

统计这个表每列的数字大于1的个数结果如下：

A	B	C
2	1	2

答：用casewhen 和sum：

```
select sum(case when a>1 then 1 else 0 end) a,
            sum(case when b>1 then 1 else 0 end) b,
            sum(case when c>1 then 1 else 0 end) c,
            sum(case when d>1 then 1 else 0 end) d,
            sum(case when e>1 then 1 else 0 end) e,
            sum(case when f>1 then 1 else 0 end) f
  from A
```

开发出了个sparkcore业务的题,求出每个用户各次支付时间的间隔，如用户支付了三次，就要求出两条这个用户支付间隔时间

用户id	支付时间
User1	T11
User2	T2
User1	T3
User2	T8
User1	T9

答：
ｇｒｏｕｐＢＹＫｅｙ－ｆｌａｔＭａｐ＋ｆｏｒｅａｃｈ
先按用户分组，排序后，循环时间，并求出间隔，并输出，开发问用什么输出，我说flatmap

总监问列式存储有哪些类型
答：
列式存储和行式存储的区别？按行对象存储和以列作为文件存储
答：ORC、Parquet
列式存储格式详解：http://blog.csdn.net/yu616568/article/details/51868447

问spark运行模式local local[] local[*]分别是什么
答：
该模式被称为Local[N]模式，是用单机的多个线程来模拟Spark分布式计算，通常用来验证开发出来的应用程序逻辑上有没有问题。
其中N代表可以使用N个线程，每个线程拥有一个core。如果不指定N，则默认是1个线程（该线程有1个core）。
如果是local[*]，则代表
Run Spark locally with as many worker threads as logical cores on your machine：
在本地运行Spark，与您的机器上的逻辑内核一样多的工作线程
关于逻辑内核的意义：
这样的处理器是双核处理器，而不是真正的四核处理器。只不过其内部有两个物理核心，而且由于这样的处理器使用了超线程技术，所以每个核心是两个线程，所以两个物理核心就是四个线程，也就形成了四个逻辑处理器，所以在操作系统的设备管理器里面看到的CPU数量是实际物理CPU数量的两倍。举个例子就是 i3 就是2个内核4个逻辑处理器，而i7是真正意义上的四核处理器
开发问了一个逻辑题：父亲有三个女儿加起来一共13岁，只有一个女儿超过了5岁，且三个女儿年龄相乘是父亲的年龄，问父亲合适的年龄
答：
1*1*11=11，1*2**10=20，1*3*9=27，1*4*8=32，1*5*7=35，{1*6*6=36}，{2*2*9=36}，2*3*8=48，2*4*7=56，2*5*6=60，3*3*7=63，3*4*6=72，3*5*5=75，4*4*5=80
给了第一个答案 7 5 1发现不对，两个大于5了 之后给了个8 4 1 ，并解释肯定要有个1岁的，不然父亲的年龄会太大

Spark 怎么设置垃圾回收机制 ？
答：Spark中各个角色的JVM参数设置：http://blog.csdn.net/wuxb_2000/article/details/52870198
1)Driver的JVM参数：
GC方式，如果是yarn-client模式，默认读取的是spark-class文件中的JAVA_OPTS；如果是yarn-cluster模式，则读取的是spark-default.conf文件中的spark.driver.extraJavaOptions对

应的参数值。
(2)Executor的JVM参数：
GC方式，两种模式都是读取的是spark-default.conf文件中的spark.executor.extraJavaOptions对应的JVM参数值。
一台节点上以root用户执行一个spark 程序，以其他非root用户也同时在执行一个spark程序，这时以spark用户登录，这个节点上，使用Jps 会看到哪些线程？
答：单独的用户只能看自己的进程
Linux下Zk、Hbase、Redis的相关命令？
答：
Hbase的：
Redis：
http://blog.csdn.net/u011642663/article/details/50975675

zookeeper：
1. 启动ZK服务:　　sh bin/zkServer.sh start
2. 查看ZK服务状态: sh bin/zkServer.sh status
3. 停止ZK服务:　　sh bin/zkServer.sh stop
4. 重启ZK服务:　　sh bin/zkServer.sh restart
1. 显示根目录下、文件： ls / 使用 ls 命令来查看当前 ZooKeeper 中所包含的内容
2. 显示根目录下、文件： ls2 / 查看当前节点数据并能看到更新次数等数据
3. 创建文件，并设置初始内容： create /zk "test" 创建一个新的 znode节点“ zk ”以及与它关联的字符串
4. 获取文件内容： get /zk 确认 znode 是否包含我们所创建的字符串
5. 修改文件内容： set /zk "zkbak" 对 zk 所关联的字符串进行设置
6. 删除文件： delete /zk 将刚才创建的 znode 删除
7. 退出客户端： quit
8. 帮助命令： help
Zk如何删除节点？
答：
delete　/dubbo
rmr
delete
get
ls /
Redis 中如何向Spark存东西一条一条插，还是一堆一堆插数据，，怎么建立连接?
答：使用foreach和foreachpartition的区别
Kafka　如何手动取数据？
答：
实际项目中数据落在哪里？hdfs　hive　hbase？
答：HDFS，hive，hbase也是基于HDFS存储的，hive是即时SQL查询，Hbase相当于数据库
Linux命令用法 ？
答：常用的linux命令用法：https://www.cnblogs.com/crazylqy/p/5818745.html
Scala 里边的对象与java对比有什么不同　数据类型？
答：scala是多范式的编程语言，支持面向对象和函数式编程。scala中一切皆对象，没有基本的数据类 型。Scala与Java具有相同的数据类型
(Byte,Short,Int,Long，Float，Double，Char，String，Boolean，Unit，NullNothin，Any，AnyRef)，具有相同的内存占用和精度，Scala中没有类似Java中那样的原始类型。
Scala中有一些比较特殊的对象：
单例(*Singleton*)对象是一个通过使用`object`关键字，当有一个与单例(*singleton*)对象同名的类时，它被称为伴生(*companion*)类，单例(*singleton*)对象调用伴生对象
样列类：scala Case类和对象

Awk 用法　大量列如何指定某一列？
答：
Awk编程参考：
http://man.linuxde.net/awk
http://blog.jobbole.com/109089/
awk中一行为一个记录：
NR==2，指定第二行，NR（Number of Record，记录数，awk中默认一行为一个记录）
练习：
1：
打印文件的第一列：awk'{print $1}' filename
打印文件的前二列：awk'{print $1,$2}' filename
打印完文件的第一列，然后打印第二列：awk'{print $1 $2}' filename
打印总行数：awk'END{print NR}' filename
打印第一行：awk'NR==1{print}' filename
打印第二行第三列输入到另外文件：cat 1.txt|awk 'NR==2{print $3}' > 2.txt

128G如何对5T数据中的ip做word count?
答：大数据处理的一般思路请参考：https://www.cnblogs.com/CheeseZH/p/5283390.html
开发问了算法怎么评估？
问这个时间复杂度是一个具体的时间还是什么？
算法的怎么评价好坏，这个时间复杂度怎样能评价这个算法？
答：
参考知乎：https://www.zhihu.com/question/19747115
最简单直接的办法， 是考察算法在特定条件下（时间、空间、其他开销、特定的计算机、特定的输入）解决问题的能力：
能解决， 那么这个算法就是可行的；
不能解决， 那么这个算法就是白搭。
如果要找寻通用的评价方法， 也应该根据算法的

平均时间复杂度
平均空间复杂度
最坏时间复杂度
最坏空间复杂度
输入与时间开销的分布关系
特定输入下的时间空间开销
所用的资源开销（如并行处理）
算法运行的特殊要求（如硬件支持）
来综合判断，不能一概而论。
关于时间复杂度和空间复杂度（参考博客）：http://blog.csdn.net/booirror/article/details/7707551/
问数据结构熟悉哪些？图 树 链表 还有几个 你说一下
答：请参考博客：http://dongxicheng.org/structure/structure-algorithm-summary/
  常用的数据结构和算法（上）：http://www.jianshu.com/p/230e6fde9c75
    常用的数据结构和算法（下） ：http://www.jianshu.com/p/42f81846c0fb
    参考书籍结构和算法思维导图
总监让我挑一个项目介绍
答：机器学习的项目：推荐系统

总监问一jar是干啥的
答：是引用第三方jar包的
问怎么解决jar包的重复引用
答：（迟疑的口吻）把jar包copy到worker上？
问你们公司没解决过这个问题啊
答：直接将需要使用的jar包放在集群的节点上面就可以了啊

我要23，人事说要高了，被怼了
我说二十一二是底线，人事说：没得说？，我说：嗯

hive怎么解决数据倾斜的问题？
参考博客：https://www.cnblogs.com/ggjucheng/archive/2013/01/03/2842860.html
本质：使map的输出数据更均匀的分布到reduce中去，是我们的最终目标
数据倾斜的原因：
key分布不均匀
业务数据本身的特性
建表时考虑不周
某些SQL语句本身就有数据倾斜
表现的形式：
任务进度长时间维持在99%（或100%），查看任务监控页面，发现只有少量（1个或几个）reduce子任务未完成。因为其处理的数据量和其他reduce差异过大。单一reduce的记录数与平均记录数差异过大，通常可能达到3倍甚至更多。 最长时长远大于平均时长。
解决方案：
1：参数调节：
hive.map.aggr=true：
Map 端部分聚合，相当于Combiner
hive.groupby.skewindata=true：
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的 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 再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。
2：参数调节：
如何Join：
关于驱动表的选取，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，以达到两表做join的时候，数据量相对变小的效果
大小表Join：
使用map join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.
大表Join大表：
把空值的key变成一个字符串加上随机数，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结果
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，在最后结果中加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。
group by维度过小：
采用sum() group by的方式来替换count(distinct)完成计算。
特殊情况特殊处理：
在业务逻辑优化效果的不大情况下，有些时候是可以将倾斜的数据单独拿出来处理。最后union回去。
如果确认业务需要这样倾斜的逻辑，考虑以下的优化方案：
总结：
1、对于join，在判断小表不大于1G的情况下，使用map join
2、对于group by或distinct，设定 hive.groupby.skewindata=true
3、尽量使用上述的SQL语句调节进行优化

hadoop数据倾斜的问题？
原因：shuffle的操作中,这些过程需要按照key值进行数据汇集处理，如果key值过于集中，在汇集过程中大部分数据汇集到一台机，这就导致数据倾斜
表现：

触发shuffle的常见算子distinct、groupByKey、reduceByKey、aggregateByKey、join、cogroup、repartition等。要解决数据倾斜的问题，首先要定位数据倾斜发生在什么地方，首先是哪个stage，直接在Web UI上看就可以，然后查看运行耗时的task

解决方案：使map的输出数据更均匀的分布到reduce中去，是我们的最终目标

(1)：设置一个 hash 份数 N，用来对条数众多的 key 进行打散

(2)：重写partition的函数

(4). 在加个combiner函数，加上combiner相当于提前进行reduce,就会把一个mapper中的相同key进行了聚合，减少shuffle过程中数据量，以及reduce端的计算量。这种方法可以有效的缓解数据倾斜问题，但是如果导致数据倾斜的key 大量分布在不同的mapper的时候，这种方法就不是很有效了。

(5). 局部聚合加全局聚合。第二种方法进行两次mapreduce，第一次在map阶段对那些导致了数据倾斜的key 加上1-n的随机前缀，这样之前相同的key 也会被分到不同的reduce中，进行聚合，这样的话就有那些倾斜的key进行局部聚合，数量就会大大降低。然后再进行第二次mapreduce这样的话就去掉随机前缀，进行全局聚合。这样就可以有效地降低mapreduce了。不过进行两次mapreduce，性能稍微比一次的差些。

一下科技面试总结：要会一句话概述，逻辑清晰，引导面试官向你熟悉的方向问

数据结构和算法方面：

1：手写有序数组的二分查找，手写单链表的反转？

答案：看复习面试思路-目标中的常见的面试算法的首写

2：如果链表的实现方式中hash的值有冲突的话，怎么解决？如果解决以后怎么解决再链表的常数次的查询？

答案：使用链表来存储重复的hash值，如何对链表进行常数次的查找，需要将链表+随机数再hash

Java方便：

1：ConcurrentHashMap是怎么实现的？

答案：concurrent包中线程安全的哈希表，采用分段锁，可以理解为把一个大的Map拆分成N个小的HashTable，根据key.hashCode()来决定把key放到哪个HashTable中。

在ConcurrentHashMap中，就是把Map分成了N个Segment，put和get的时候，都是现根据key.hashCode()算出放到哪个Segment中。

大数据方面：

1：HDFS的读写流程细节？HDFS中的fsimage里面存储的是什么信息？副本的存放策略？

答：

HDFS的读写流程：

写流程：

1. 客户端创建 DistributedFileSystem 对象.

2. DistributedFileSystem 对象调用元数据节点，在文件系统的命名空间中创建一个新的文件，元数据节点首先确定文件原来不存在，并且客户端有创建文件的权限，然后创建新文件，并标识为“上传中”状态，即可以看见，但不能使用。

3. DistributedFileSystem 返回 DFSOutputStream，客户端用于写数据

4. 客户端开始写入数据， DFSOutputStream 将数据分成块，写入 data queue（Data queue 由 Data Streamer 读取），并通知元数据节点分配数据节点，用来存储数据块（每块默认复制 3 块）。分配的数据节点放在一个 pipeline 里。 Data Streamer 将数据块写入 pipeline 中的第一个数据节点。第一个数据节点将数据块发送给第二个数据节点。第二个数据节点将数据发送给第三个数据节点。 注意：并不是第一个数据节点完全接收完 block 后再发送给后面的数据节点，而是接收到一部分就发送，所以三个节点几乎是同时接收到完整的 block 的。 DFSOutputStream 为发出去的数据块保存了 ack queue，等待 pipeline 中的数据节点告知数据已经写入成功。如果 block 在某个节点的写入的过程中失败：关闭 pipeline，将 ack queue 放至 data queue 的开始。已经写入节点中的那些 block 部分会被元数据节点赋予新的标示，发生错误的节点重启后能够察觉其数据块是过时的，会被删除。失败的节点从 pipeline 中移除， block 的其他副本则写入 pipeline 中的另外两个数据节点。元数据节点则被通知此 block 的副本不足，将来会再创建第三份备份

5. ack queue 返回成功。

6. 客户端结束写入数据，则调用 stream 的 close 函数， 最后通知元数据节点写入完毕

总结：

客户端切分文件 Block，按 Block 线性地和 NN 获取 DN 列表（副本数），验证 DN 列表后以更小的单位流式传输数据，各节点两两通信确定可用， Block 传输结束后， DN 向 NN 汇报 Block 信息， DN 向 Client 汇报完成， Client 向 NN 汇报完成，获取下一个 Block 存放的 DN 列表，最终 Client 汇报完成， NN 会在写流程更新文件状态。

读流程：

1. 客户端(client)用 FileSystem 的 open()函数打开文件

2. DistributedFileSystem 调用元数据节点，得到文件的数据块信息。对于每一个数据块，元数据节点返回保存数据块的数据节点的地址。

3. DistributedFileSystem 返回 FSDataInputStream 给客户端，用来读取数据。

4. 客户端调用 stream 的 read()函数开始读取数据（也会读取 block 的元数据）。DFSInputStream 连接保存此文件第一个数据块的最近的数据节点（优先读取同机架的 block）。

5. Data 从数据节点读到客户端。当此数据块读取完毕时， DFSInputStream 关闭和此数据节点的连接，然后连接此文件下一个数据块的最近的数据节点。

6. 当客户端读取完毕数据的时候，调用 FSDataInputStream 的 close 函数。

7. 在读取数据的过程中，如果客户端在与数据节点通信出现错误，则尝试连接包含此数据块的下一个数据节点。失败的数据节点将被记录，以后不再连接。

总结：

客户端和 NN 获取一部分 Block（获取部分 block 信息，而不是整个文件全部的 block 信息，读完这部分 block 后，再获取另一个部分 block 的信息）副本位置列表，线性地和 DN 获取 Block，最终合并为一个文件，在 Block 副本列表中按距离择优选取。

fsimage里面存储的元数据：

metadata 存储到磁盘文件名为” fsimage”， Block 的位置信息(副本的位置信息)不会保存到 fsimage，由 DataNode实时汇报汇报。

metadata 信息包括：文件 owership、 permissions、文件大小、时间、 Block 列表、 Block 偏移量和位置信息（副本位置由 DataNode 汇报，实时改变，不会持久化）等。

副本的存放策略：

第一个副本：放置在上传文件的 DN；如果是集群外提交，则随机挑选一台磁盘不太
满， CPU 不太忙的节点。
第二个副本：放置在于第一个副本不同的机架的节点上。
第三个副本：与第二个副本相同机架的节点。
更多副本：随机节点
2：HDFS的机架感知？
答案：
根据副本的存放策略，HDFS是如何知道多个不同node是否在同一个机架上呢 ？
在namenode启动时如果net.topology.script.file.name配置的参数不为空，表示已经动机架感知，当datanode注册时和heartbeat时，会把datanode的ip作为参数传入，返回信息为此datanode的机架信息。如果没有参数配置，datanode统一为默认的机架/default-rack
3：如果Client节点就在HDFS中的一台DataNode节点上，副本的数据又是如何存储的？
答案：存放在当前的DN上，其他的和副本的存放的策略一样，第二个副本存放在和第一个副本不同的机架上的节点上，第三个副本存放在同第二个副本相同的机架的不同的节点上
4：Hbase有了解吗？Hbase的架构等
答案：详细查看Hbase的的思维导图
5：Spark的提交方式？
答案：不管是提交到yarn上面还是提交到standalone上都分为Client的方式提交和Cluster的方式提交
6：client和Cluster的提交方式的区别？
答案：
Driver program 所在的节点不一样
提交的网络流量激增的问题：频繁的提交Client会产生大量的网络流量激增，Cluster则分散了这种流量激增的问题
应用的场景不一样：适合调试，debug，一个适合真实的生产环境
7：task任务的实现接口？应该问的就是Spark的任务调度和资源调度
参考博客：http://blog.csdn.net/pelick/article/details/41866845
项目方面：
1：项目的模型训练和项目的准确度是多少？
答案：
训练集：测试集=2：8
准确率：85%
其他的一些杂项：
项目组多少人？怎么分工的？薪水多少？项目中你负责那一块？
答案：
8个人，2个左右的数据清洗，2个左右的数据挖掘，1个负责集群的维护，1个数据的可视化，其他的是小组的领导和需求分析
薪水：16k，7的基本，5的绩效，4的奖金，没有一金
一览科技面试总结：
数据结构和算法方面：
手写冒泡排序和二分查找？
使用shell实现wordCount？
答案：cat 11.txt | tr ' ' '\n' | sort -k 1 | uniq -c | awk '{print $2"\t"$1}' |sort -k2 -nr |head
如何将一个标题等在一千万数据中进行进行Top10的推荐？
答案：标题向量化，数据清洗和降维，计算相似度，推荐
能不能使用lucence实现二次开发？
答案：如果需要，可以研究走做
mysql的全连索引等吧？
答案：
以前用过，但是很长时间不用了
mysql经典面试题：http://blog.csdn.net/u013252072/article/details/52912385
项目方面：
推荐系统项目的架构？
答：lambda架构，给他画了基本的架构图然后讲解了。

房互网:
1:合个项目的讲解
2:hive
3:hbase读写缓存机制
4:kafka
5:flume
6:sqoop
写sql的题目(网上的原题)：http://blog.csdn.net/baolibin528/article/details/46774015