Spark常用面试题

**1、spark的有几种部署模式，每种模式特点？**

1）本地模式   Spark本质是一套SDK，可以不依赖任何集群服务，因此可以在本地，起多个线程的方式来运行。将Spark应用以多线程的方式直接运行在本地，一般都是为了方便调试，本地模式分三类   local：只启动一个executor   local[k]:启动k个executor   local[ \* ]:启动跟cpu数目相同的 executor。

2）standalone模式   分布式部署集群（master-worker服务运行模式），自带完整的服务，资源管理和任务监控是Spark自己监控，这个模式也是其他模式的基础。

3）Spark on yarn模式   分布式部署集群，资源和任务监控交给yarn管理，但是目前仅支持粗粒度资源分配方式，包含cluster和client运行模式，cluster适合生产，driver运行在集群子节点，具有容错功能，client模式其dirver运行在统一的客户端提交节点。

4）Spark On Mesos模式。  官方推荐这种模式（当然，原因之一是血缘关系）。正是由于Spark开发之初就考虑到支持Mesos，因此，目前而言，Spark运行在Mesos上会比运行在YARN上更加灵活，更加自然。

用户可选择两种调度模式之一运行自己的应用程序：（1）粗粒度模式（Coarse-grained Mode）：每个应用程序的运行环境由一个Dirver和若干个Executor组成，其中，每个Executor占用若干资源，内部可运行多个Task（对应多少个vcore）。应用程序的各个任务正式运行之前，需要将运行环境中的资源全部申请好，且运行过程中要一直占用这些资源，即使不用，最后程序运行结束后，回收这些资源。（2）细粒度模式（Fine-grained Mode）：鉴于粗粒度模式会造成大量资源浪费，Spark On Mesos还提供了另外一种调度模式：细粒度模式，这种模式类似于现在的云计算，思想是按需分配。  
**2、Spark为什么比mapreduce快？**

（参考我的视频回答）  
**3，简单说一下hadoop和spark的shuffle相同和差异？**

1. 从 high-level 的角度来看，两者并没有大的差别。都是将 mapper（Spark 里是 ShuffleMapTask）的输出进行 partition，不同的 partition 送到不同的 reducer（Spark 里 reducer 可能是下一个 stage 里的 ShuffleMapTask，也可能是 ResultTask）。Reducer 以内存作缓冲区，边 shuffle 边 aggregate 数据，等到数据 aggregate 好以后进行 reduce() （Spark 里可能是后续的一系列操作）。
2. 从 low-level 的角度来看，两者差别不小。Hadoop MapReduce 是 sort-based，进入 combine() 和 reduce() 的 records 必须先 sort。这样的好处在于 combine/reduce() 可以处理大规模的数据，因为其输入数据可以通过外排得到（mapper 对每段数据先做排序，reducer 的 shuffle 对排好序的每段数据做归并）。目前的 Spark 默认选择的是 hash-based，通常使用 HashMap 来对 shuffle 来的数据进行 aggregate，不会对数据进行提前排序。如果用户需要经过排序的数据，那么需要自己调用类似 sortByKey() 的操作；如果你是Spark 1.1的用户，可以将spark.shuffle.manager设置为sort，则会对数据进行排序。在Spark 1.2中，sort将作为默认的Shuffle实现。
3. 从实现角度来看，两者也有不少差别。Hadoop MapReduce 将处理流程划分出明显的几个阶段：map(), spill, merge, shuffle, sort, reduce() 等。每个阶段各司其职，可以按照过程式的编程思想来逐一实现每个阶段的功能。在 Spark 中，没有这样功能明确的阶段，只有不同的 stage 和一系列的 transformation()，所以 spill, merge, aggregate 等操作需要蕴含在 transformation() 中。如果我们将 map 端划分数据、持久化数据的过程称为 shuffle write，而将 reducer 读入数据、aggregate 数据的过程称为 shuffle read。那么在 Spark 中，问题就变为怎么在 job 的逻辑或者物理执行图中加入 shuffle write 和 shuffle read的处理逻辑？以及两个处理逻辑应该怎么高效实现？Shuffle write由于不要求数据有序，shuffle write 的任务很简单：将数据 partition 好，并持久化。之所以要持久化，一方面是要减少内存存储空间压力，另一方面也是为了 fault-tolerance。

**4、spark工作机制是什么样的？**

① 构建Application的运行环境，Driver创建一个SparkContext

② SparkContext向资源管理器（Standalone、Mesos、Yarn）申请Executor资源，资源管理器启动StandaloneExecutorbackend（Executor）

③ Executor向SparkContext申请Task

④ SparkContext将应用程序分发给Executor

⑤ SparkContext就建成DAG图，DAGScheduler将DAG图解析成Stage，每个Stage有多个task，形成taskset发送给task Scheduler，由task Scheduler将Task发送给Executor运行

⑥ Task在Executor上运行，运行完释放所有资源

**5、spark的优化怎么做？**

Spark调优比较复杂，但是大体可以分为三个方面来进行

1）业务层面的调优：能过滤掉的数据，优先过滤，减少后续数据处理过程的IO。根据业务需要优化处理流程，原本可以3个步骤处理完的流程，就不要用5个步骤；

2）应用程序层面的调优：代码逻辑中避免耗时的额外IO操作，比如数据计算过程中需要读取额外的磁盘IO，可以考虑优先将这部分数据通过广播变量进行存储，在使用时直接读取广播变量；

3）JVM层面的调优：设置合适的资源量，设置合理的JVM，启用高效的序列化方法如kyro，增大off head内存等等；

**6、数据本地性是在哪个环节确定的？**

具体的task运行在哪台机器上，dag划分stage的时候确定的

**7、RDD的弹性表现在哪几点？**

1）自动的进行内存和磁盘的存储切换；

2）基于Lineage的高效容错；

3）task如果失败会自动进行特定次数的重试；

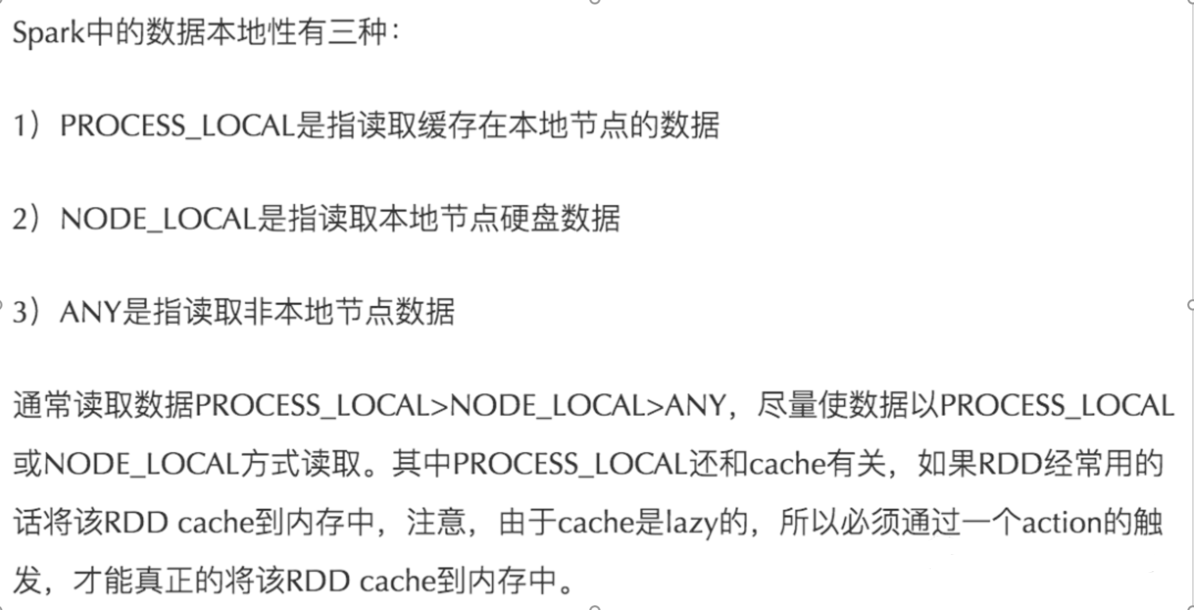
4）stage如果失败会自动进行特定次数的重试，而且只会计算失败的分片；

5）checkpoint和persist，数据计算之后持久化缓存；

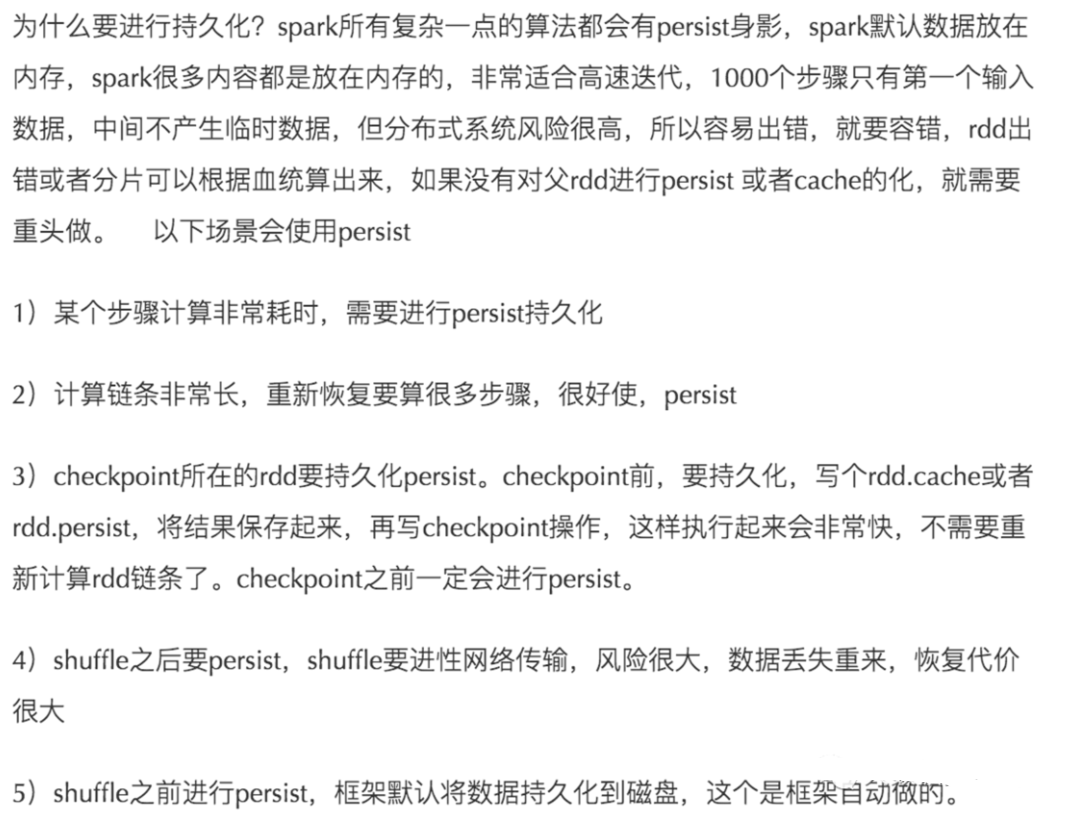
 6）数据调度弹性，DAG TASK调度和资源无关；

7）数据分片的高度弹性。

**8、 Spark的数据本地性有哪几种？**



**9、Spark为什么要持久化，一般什么场景下要进行persist操作？**



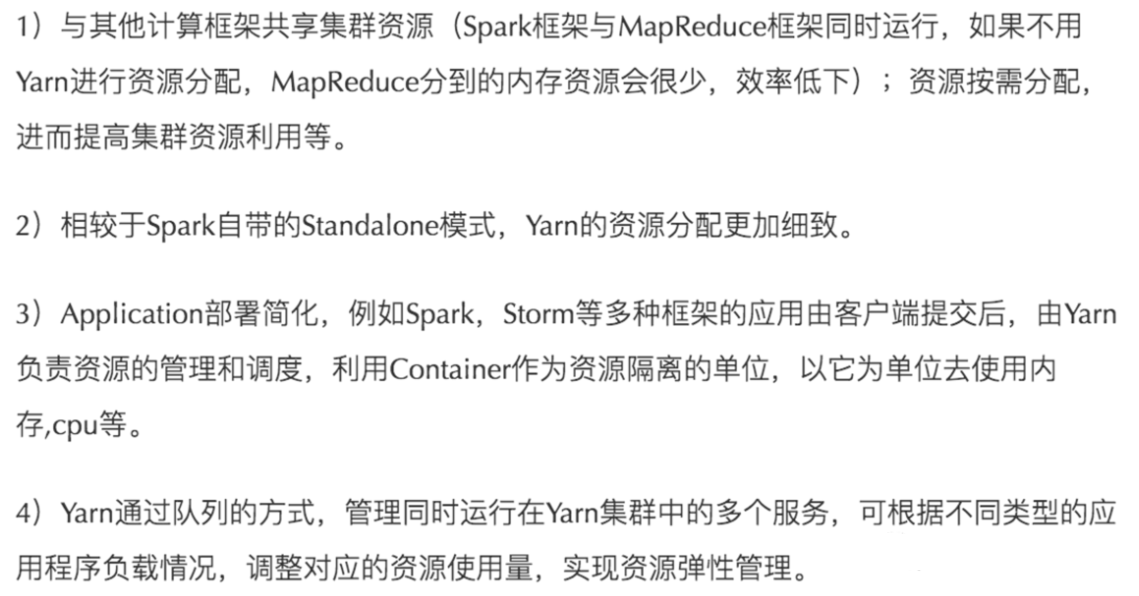
**10、描述Yarn执行一个任务的过程？**

1）客户端client向ResouceManager提交Application，ResouceManager接受Application并根据集群资源状况选取一个node来启动Application的任务调度器driver（ApplicationMaster）。

2）ResouceManager找到那个node，命令其该node上的nodeManager来启动一个新的 JVM进程运行程序的driver（ApplicationMaster）部分，driver（ApplicationMaster）启动时会首先向ResourceManager注册，说明由自己来负责当前程序的运行。



**11、Spark on Yarn 模式有哪些优点**

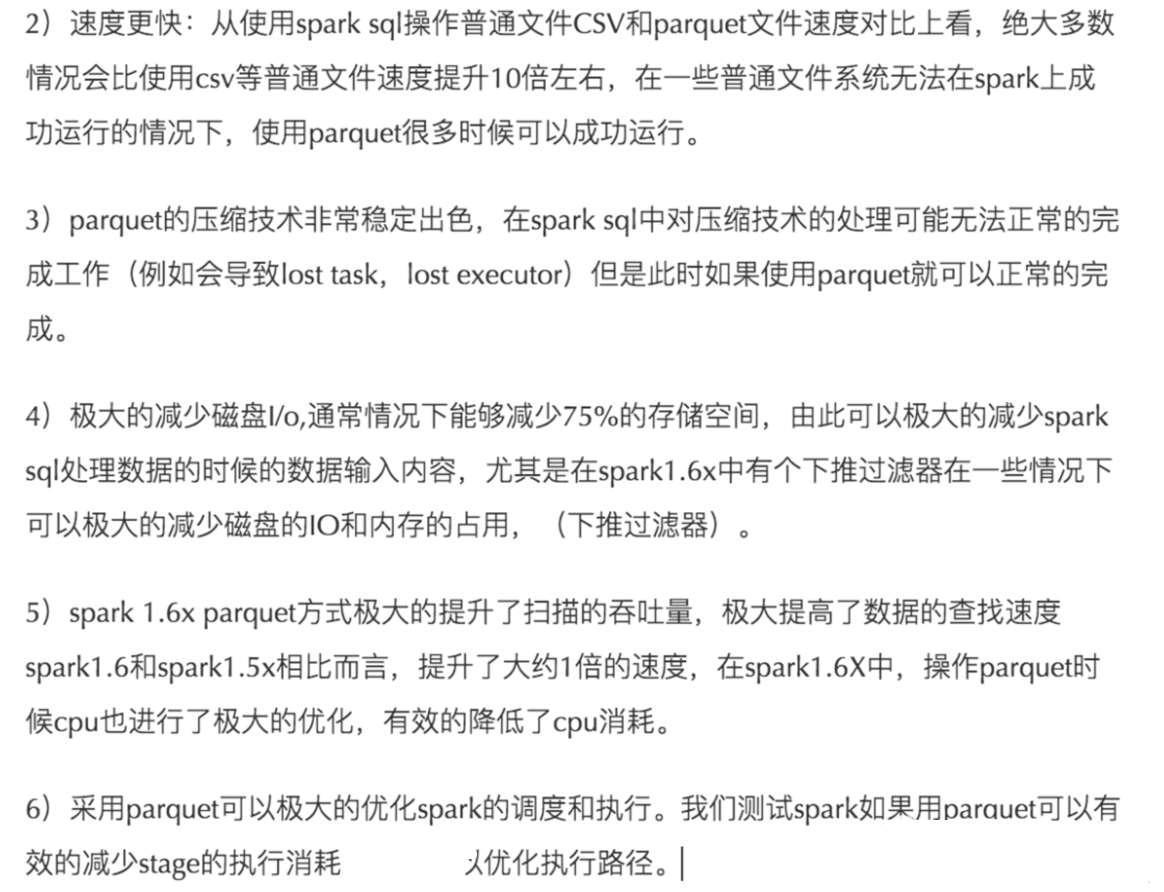


**12、谈你对container的理解？**

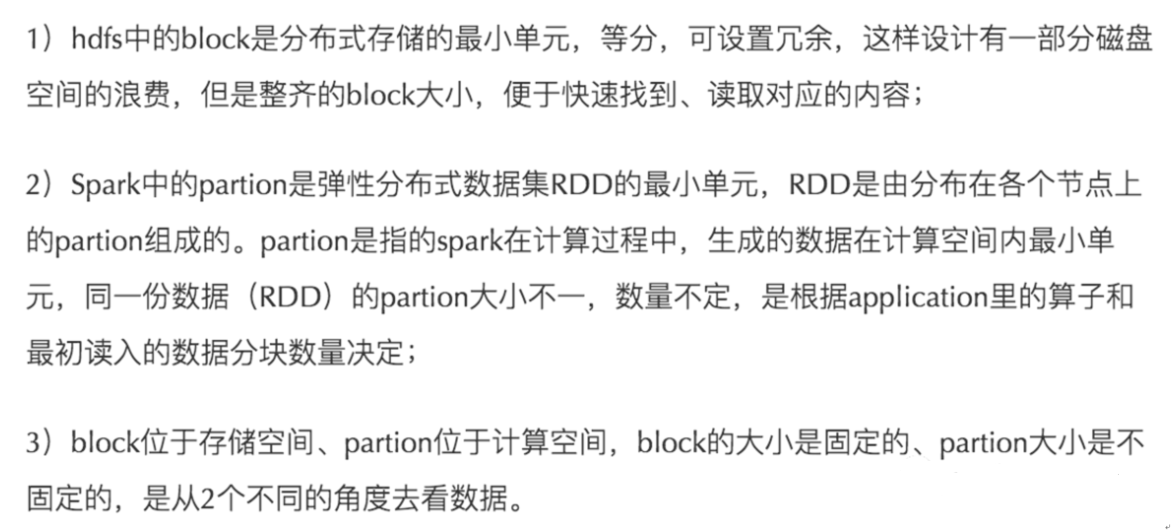


**13、Spark使用parquet文件存储格式能带来哪些好处？**

1）如果说HDFS是大数据时代分布式文件系统首选标准，那么parquet则是整个大数据时代文件存储格式实时首选标准。



**14、介绍parition和block有什么关联关系？**



**15、Spark应用程序的执行过程是什么？**



**16、不需要排序的hash shuffle是否一定比需要排序的sort shuffle速度快？**

不一定，当数据规模小，Hash shuffle快于Sorted Shuffle数据规模大的时候；当数据量大，sorted Shuffle会比Hash shuffle快很多，因为数量大的有很多小文件，不均匀，甚至出现数据倾斜，消耗内存大，1.x之前spark使用hash，适合处理中小规模，1.x之后，增加了Sorted shuffle，Spark更能胜任大规模处理了。

**17、Sort-based shuffle的缺陷?**

1）如果mapper中task的数量过大，依旧会产生很多小文件，此时在shuffle传递数据的过程中reducer段，reduce会需要同时大量的记录进行反序列化，导致大量的内存消耗和GC的巨大负担，造成系统缓慢甚至崩溃。

2）如果需要在分片内也进行排序，此时需要进行mapper段和reducer段的两次排序。

**18、spark.storage.memoryFraction参数的含义,实际生产中如何调优？**

1）用于设置RDD持久化数据在Executor内存中能占的比例，默认是0.6,，默认Executor 60%的内存，可以用来保存持久化的RDD数据。根据你选择的不同的持久化策略，如果内存不够时，可能数据就不会持久化，或者数据会写入磁盘；

2）如果持久化操作比较多，可以提高spark.storage.memoryFraction参数，使得更多的持久化数据保存在内存中，提高数据的读取性能，如果shuffle的操作比较多，有很多的数据读写操作到JVM中，那么应该调小一点，节约出更多的内存给JVM，避免过多的JVM gc发生。在web ui中观察如果发现gc时间很长，可以设置spark.storage.memoryFraction更小一点。  
**19、Spark有哪两类算子？**

Transformation（转化）算子和Action（执行）算子。

**20、如何从Kafka中获取数据？**

1）基于Receiver的方式这种方式使用Receiver来获取数据。Receiver是使用Kafka的高层次Consumer API来实现的。receiver从Kafka中获取的数据都是存储在Spark Executor的内存 中的，然后Spark Streaming启动的job会去处理那些数据。

2）基于Direct的方式这种新的不基于Receiver的直接方式，是在Spark 1.3中引入的，从而能够确保更加健壮的机制。替代掉使用Receiver来接收数据后，这种方式会周期性地 查询Kafka，来获得每个topic+partition的最新的offset，从而定义每个batch的offset的范围。当处理数据的job启动时，就会使用Kafka的简单consumer api来 获取Kafka指定offset范围的数据。

**21、map与flatMap的区别？**

map：对RDD每个元素转换，文件中的每一行数据返回一个数组对象。

flatMap：对RDD每个元素转换，然后再扁平化。将所有的对象合并为一个对象，文件中的所有行数据仅返回一个数组对象，会抛弃值为null的值。

**22、driver的功能是什么？**

1）一个Spark作业运行时包括一个Driver进程，也是作业的主进程，具有main函数，并且有SparkContext的实例，是程序的入口点；

2）功能：负责向集群申请资源，向master注册信息，负责了作业的调度，负责作业的解析、生成Stage并调度Task到Executor上。包括DAGScheduler， TaskScheduler。

**23、Spark技术栈有哪些组件，每个组件都有什么功能，适合什么应用场景？**

可以画一个这样的技术栈图先，然后分别解释下每个组件的功能和场景

1）Spark core：是其它组件的基础，spark的内核，主要包含：有向循环图、RDD、Lingage、Cache、broadcast等，并封装了底层通讯框架， 是Spark的基础。

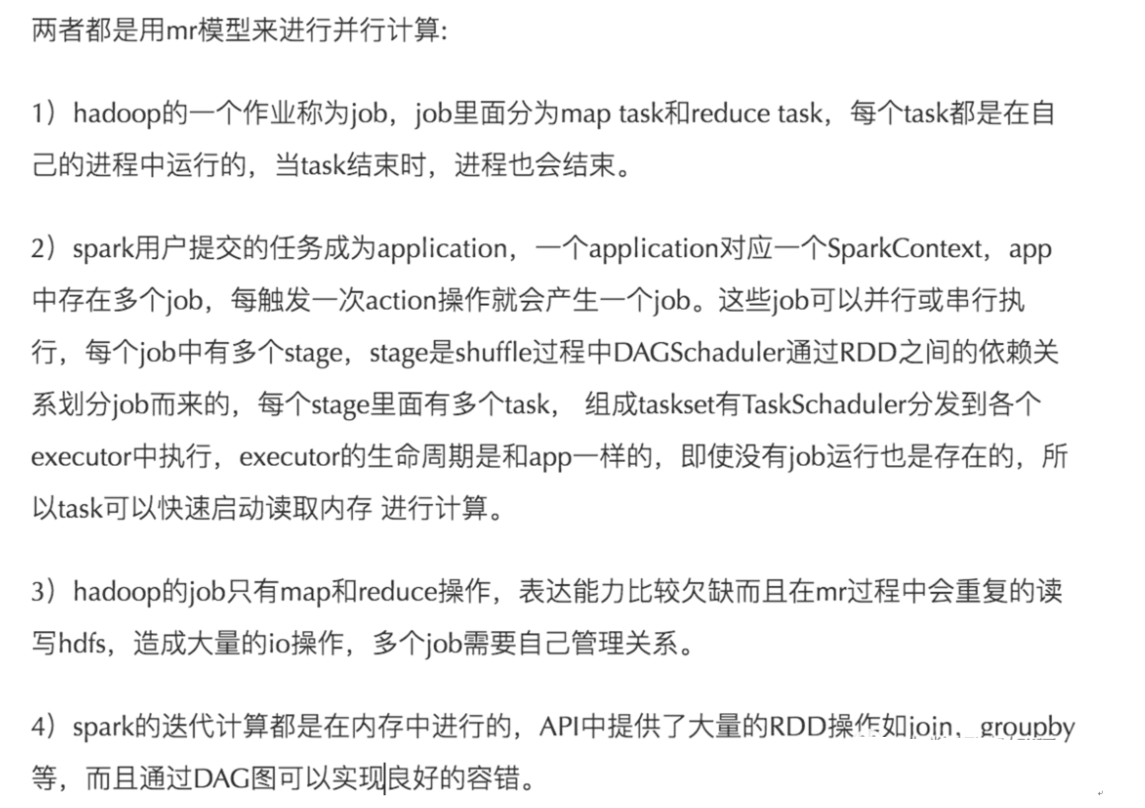
2）SparkStreaming是一个对实时数据流进行高通量、容错处理的流式处理系统，可以对多种数据源（如Kafka、Flume、Twitter、Zero和TCP 套接字） 进行类似Map、Reduce和Join等复杂操作，将流式计算分解成一系列短小的批处理作业。

3）Spark sql：Shark是SparkSQL的前身，Spark SQL的一个重要特点是其能够统一处理关系表和RDD，使得开发人员可以轻松地使用SQL命令进行外部查询， 同时进行更复杂的数据分析。

4）MLBase是Spark生态圈的一部分专注于机器学习，让机器学习的门槛更低，让一些可能并不了解机器学习的用户也能方便地使用MLbase。MLBase分为四部分：MLlib、MLI、ML Optimizer和MLRuntime。

5）GraphX是Spark中用于图和图并行计算。

**25、Mapreduce和Spark的都是并行计算，那么他们有什么相同和区别？**



**26、什么是RDD宽依赖和窄依赖？**

RDD和它依赖的parent RDD(s)的关系有两种不同的类型，即窄依赖（narrow dependency）和宽依赖（wide dependency）

1）窄依赖指的是每一个parent RDD的Partition最多被子RDD的一个Partition使用

2）宽依赖指的是多个子RDD的Partition会依赖同一个parent RDD的Partition

**27、cache和pesist的区别？**

cache和persist都是用于将一个RDD进行缓存的，这样在之后使用的过程中就不需要重新计算了，可以节省程序运行时间

1）cache只有一个默认的缓存级别MEMORY\_ONLY ，cache调用了persist，而persist可以根据情况设置其它的缓存级别；

2）executor执行的时候，默认60%做cache，40%做task操作，persist是最根本的函数，最底层的函数。

**28、 cache后面能不能接其他算子,它是不是action操作？**

cache可以接其他算子，但是接了算子之后，起不到缓存应有的效果，因为会重新触发cache。cache不是action操作。

**其他面试题推荐**

1.rdd的属性  
  
2.算子分为哪几类(RDD支持哪几种类型的操作)  
  
3.创建rdd的几种方式  
  
4.spark运行流程  
  
5.Spark中coalesce与repartition的区别  
  
6.sortBy 和 sortByKey的区别  
  
7.map和mapPartitions的区别  
  
8.数据存入Redis 优先使用map mapPartitions foreach foreachPartions哪个  
  
9.reduceByKey和groupBykey的区别  
  
10.cache和checkPoint的比较  
  
11.spark streaming流式统计单词数量代码  
  
12.简述map和flatMap的区别和应用场景  
  
13.计算曝光数和点击数  
  
14.分别列出几个常用的transformation和action算子  
  
15.按照需求使用spark编写以下程序，要求使用scala语言  
  
16.spark应用程序的执行命令是什么？  
  
17.Spark应用执行有哪些模式，其中哪几种是集群模式  
  
18.请说明spark中广播变量的用途  
  
19.以下代码会报错吗？如果会怎么解决 val arr = new ArrayList[String]; arr.foreach(println)  
  
20.写出你用过的spark中的算子，其中哪些会产生shuffle过程  
  
21.Spark中rdd与partition的区别  
  
22.请写出创建Dateset的几种方式  
  
23.描述一下RDD，DataFrame，DataSet的区别？  
  
24.描述一下Spark中stage是如何划分的？描述一下shuffle的概念  
  
25.Spark 在yarn上运行需要做哪些关键的配置工作？如何kill -个Spark在yarn运行中Application  
  
26.通常来说，Spark与MapReduce相比，Spark运行效率更高。请说明效率更高来源于Spark内置的哪些机制？请列举常见spark的运行模式？  
  
27.RDD中的数据在哪？  
  
28.如果对RDD进行cache操作后，数据在哪里？  
  
29.Spark中Partition的数量由什么决定  
  
30.Scala里面的函数和方法有什么区别  
  
31.SparkStreaming怎么进行监控?  
  
32.Spark判断Shuffle的依据?  
  
33.Scala有没有多继承？可以实现多继承么？  
  
34.Sparkstreaming和flink做实时处理的区别  
  
35.Sparkcontext的作用  
  
36.Sparkstreaming读取kafka数据为什么选择直连方式  
  
37.离线分析什么时候用sparkcore和sparksql  
  
38.Sparkstreaming实时的数据不丢失的问题  
  
39.简述宽依赖和窄依赖概念，groupByKey,reduceByKey,map,filter,union五种操作哪些会导致宽依赖，哪些会导致窄依赖  
  
40.数据倾斜可能会导致哪些问题，如何监控和排查，在设计之初，要考虑哪些来避免  
  
41.有一千万条短信，有重复，以文本文件的形式保存，一行一条数据，请用五分钟时间，找出重复出现最多的前10条  
  
42.现有一文件，格式如下，请用spark统计每个单词出现的次数  
  
43.共享变量和累加器  
  
 44.当 Spark 涉及到数据库的操作时，如何减少 Spark 运行中的数据库连接数？  
  
45.特别大的数据，怎么发送到excutor中？  
  
46.spark调优都做过哪些方面？  
  
47.spark任务为什么会被yarn kill掉？  
  
48.Spark on Yarn作业执行流程？yarn-client和yarn-cluster有什么区别？  
  
49.Flatmap底层编码实现？  
  
50.spark\_1.X与spark\_2.X区别