**1、**在HDFS服务中，为了保证NameNode高可用的角色不包括：DataNode **2、**NameNode在启动时自动进入安全模式，在安全模式阶段，说法错误的是：允许用户对文件系统进行读写操作  **3、**在集群中配置HDFS的副本数为3，设置数据块大小为128M，此时上传一份64M的数据文件，该数据文件占用HDFS空间大小为：384M **4、**对YARN角色在集群中的作用描述正确的是：集群资源管理、集群任务调度与管理 **5、**YARN服务包含：ResourceManager、NodeManager、ApplicationMaster、Container

**6、**分布式计算框架：MapReduce、SPARK、Tez；不属于分布式的框架：MATLAB **7、**关于外表和托管描述正确的是：删除外表只会删除Inceptor上的元数据不会删除数据文件，删除托管两者都会被删除 **8、**对分桶描述正确的：分桶表通过改变数据的存储分布，对查询起到一定的优化作用。**9、**关于inceptor excutor资源配置的说法正确的有：Excutor资源配置fixed和ratio两种模式；Excutor内核数配置的是每个excutor所使用的逻辑core数量；Excutor内核数和内存配置比例一般为1 core：2G memory

**10、**假设有查询语句：SELECT Sex, Region, COUNT(ID), AVG (Salary) FROM Employee WHERE Department = ‘IT' GROUP BY Sex, Region ORDER BY Sex, Region；通过holodesk的cube和index手段对这种过滤率和聚合率高的业务进行优化，正确建表：CREATE TABLE Employee TBLPROPERTIES ( 'cache' = ‘RAM', 'holodesk.index' = ‘Department', 'holodesk.dimension' = 'Sex, Region’ )

**11、Hyperbase**全局索引的正确描述:核心是倒排表;全局索引概念是对应Rowkey这个“一级”索引;全局索引使用B+树检索数据

**12、Hyperbase**分布式存储的最小单元是：Region **13、**有关 Hyperbase 说法正确的是：这题说法都不对 **14、**关于 StreamSQL 的概念描述正确的是：Stream 是数据流；Application 是一个或多个 streamjob 的集合 **15、**某交通部门通过使用流监控全市过往 24 小时各个卡口数据，要求每分钟更新一次，原始流为 org\_stream，以下实现正确的：CREATE STREAM traffic\_stream AS SELECT \* FROM original\_stream STREAMWINDOW w1 AS (length '24' hour slide '1' minute); **16**、以下不是 Zookeeper 的功能是：存储大量数据；是 Zookeeper 的功能的是：配置管理、集群管理、分布式锁 **17**、以下服务需要与 zookeeper 进行通信的是：Active ResourceManager **18**、关于 flume 和 sqoop 对比的描述，不正确的是： flume 主要采集流式数据而 sqoop 主要用来迁移规范化数据；flume 和 sqoop 都是分布式处理任务 **19**、有关使用 sqoop 抽取数据（数据迁移）的原理的描述不正确：sqoop 抽取数据是个多节点并行抽取的过程，因此 map 的个数设置的越多性能越好 **20、**sqoop 抽取数据时需要做一些数据转换的工作，说法不正确： --hive-drop-import-delims 用来设置在 hdfs 生成的文件的存储形式为列存储 **21、**有关 flume 的描述不正确：flume 和 sqoop 功能相似，因此可以相互替代。**22、**flume 不支持的 sink：memory sink **23、**对 ElasticSearch 描述不正确：ElasticSearch 数据存储在 HDFS 上；主节点(master node)进行集群的管理，只负责集群节点添加和删除 **24、**不属于 kafka 应用场景的是：关系型数据库和大数据平台之间的数据迁移 **25、**TDH 提供哪几种认证模式：所有服务使用简单认证模式——所有服务都无需认证即可互相访问；所有服务都启用 Kerberos 认证，用户要提供 Kerberos principal 和密码（或keytab）来访问各个服务；所有服务都启用 Kerberos 同时 Inceptor 启用 LDAP 认证 **26、**对各组件的运维页面描述不正确：通过 Resource Manager 的 8180 对 YARN 上运行的任务进行监控 **27、**Inceptor server 服务无法启动时，该如何查看日志： 查看 Inceptor server 所在节点/var/log/inceptorsql\*/目录下的 hive-server2.log 日志 **28、**对 Hadoop 组件的应用场景描述正确：Hive 主要用于构建大数据数仓，主要做批处理、统计分析型业务；Hbase 主要用于检索查询的 OLTP 业务；ElasticSearch 主要用于全文检索的关键字查询业务；Spark Streaming 主要用于实时数据的业务场景 **29、**不属于管理角色：Node Manager。 **30、**不属于集群预安装工作：配置集群安全模式 **31、**与HDFS 有关的正确说法：规划 HDFS 集群时，建议 Active NameNode 和 Standby NameNode 分配在不同机架上 **32、**有关 YRAN 中角色的描述不正确：NodeManager 负责调度当前节点的所有 ApplicationMaster **33、**Spark 与 MapReduce 对比，突出的优势不包括： Spark 可以运行在 YARN 之上而 MapReduce 不能

**34、**导入数据经常会用到 LOAD 命令，描述错误的是：源数据文件存放于 hdfs 上，通过 load 命令加载数据文件，数据文件将被复制到表目录下 **35、**tableA 有 10G 的数据，tableB 有 100G 的数据，两个表通过共有的 id 列做关联查询 name 列，以下方式可以优化计算效率：建表时将 tableA 和 tableB 根据 id 字段分相同数量的桶 **36、**不属于 Hyperbase 存储模型单位：region **37、**有关 Minor Compact 的描述正确：把多个 HFile 合成一个 **38、** stream 的描述不正确：定义 Derived stream 后 stream 当即根据转换规则进行变形 **39、**某公司有部门 A、部门 B…，各部门的源数据都取自于企业总线，要求部门内部共享数据源，部门间做到资源隔离，以下设计合理的有：每个部门起一个 application 管理本部门的 streamjob **40、**Zookeeper 服务描述正确的：它是分布式应用程序协调服务；Zookeeper 通过选举机制确定 leader，有且仅有一个 **41、**hue 可以实现 hdfs的： 创建目录、上传文件、直接查看文件、更改权限 **42、**通过 oozie workflow 调度 sqoop 任务：确保对应的 jdbc 驱动正确上传到 hdfs 上；Sqoop 导入的 hdfs 目录必须前提不存在 **43、**sqoop 的参数说法不正确：--query 是执行 sqoop 操作的必需参数

**44、**不属于 Flume 的 Source 类型的是：file source **45、**有关 Elasticsearch 特性描述正确：分布式实时文件存储，可将每一个字段存入索引；实时分析的分布式搜索引擎；支持插件机制，分词插件、同步插件 **46、**在安装有 kerberos 服务的集群中如何切换用户：直接使用 kinit 用户名称方式进行切换 **47、** Transwarp Manager 描述不正确：在 Transwarp Manager 上能启动和停止 Transwarp Agent 角色 **48、**某电信部门有 100 亿条用户过往使用通讯记录，现需要提供客户终端根据电话号精确查询历史通讯，满足用户同时并发访问，则该表应该设计为：Hyperbase 表+全局索引 **49、**可以安装 TDH 的操作系统：SUSE SP2-SP3、CentOS 6.3-6.5、REHL 6.3-6.5 **50、**HDFS说法正确：规划 HDFS 集群时，建议 Active NameNode 和 Standby NameNode 分配在不同的机架上 **51、**NameNode 用于存储 HDFS 上数据块的元数据信息，它保存的数据形式： fsimage、editlog **52、**YARN 框架中，负责集群资源管理的组件：ResourceManager **53、**MapReduce 计算框架的特点包括：自动化并行和分布式计算、出错容忍度高、优先数据本地化计算 **54、**关于 Inceptor 数据倾斜场景正确的处理方式：导入数据期间格式转换出现错误引起 null 过多，可以通过重新清理数据解决；将一起数据倾斜的数据和剩下的数据单独运行，再通过 union 合并的方式解决 **55、**关于 inceptor 日志信息描述正确：Inceptor server 日志存放于各节点的/var/log/inceptorsql[x]/hive-server.log；可以通过 inceptor server 4040 查看 SQL 错误日志；Excutor 日志存放于 excutor 节点的/var/log/inceptorsql[x]/spark-excutor.log； ExcutorGC 日志存放于 excutor 节点的/var/log/inceptorsql[x]/spark-excutor.gc.log **56、**HMaster 功能：为 Region Server 分配 region；管理用户对 table 的增删改查操作 **57、**Minor Compact：把多个 HFile 合成一个 **58、**创建全局索引的语句： add\_global\_index 't1','index\_name’,'COMBINE\_INDEX|INDEXED=f1:q1:9|rowKey:rowKey:10, UPDATE=true'。 **59、**对**流处理计算框架**描述不正确：这题都不对。**60、 Hue** 对 hive server 的支持度描述正确：只支持 hive server2 **61、**关于 **oozie 三个编辑**器正确： workflow 是最简单的一种工作方式；coordinator 可以包含一到多个 workflow **62、Guardian** 的功能：用户管理、用户认证、审计、权限管理 **63、**一个表数据要存储在 **hyperbas**e 上，并**创建全文索引**，原表数据 10GB，HDFS配置为 3 副本，hyperbase 压缩比例按 1:3 计算，索引数据量为 20GB，ES 副本数为 1， ES 压缩比按 1:3 计算，则该表存储空间：23.33GB **64、**作为 HDFS 高可靠协调服务的共享存储：JournalNodes **65、**ResourceManager 是 YARN 的主要组成部分，有关其功能描述不正确：它直接将集群所拥有的资源按需分配给运行在 YARN 上的应用程序 **66、**当前用户提交了一个 wordcount 词频统计的任务，最后任务执行失败，可能的原因：全部 **67、**SQL 运行中如果出现 maptask 数据特别多，执行时间又很短时可以通过小文件合并来进行优化，以下是合并参数：SET ngmr.partition.automerge = TRUE；SET ngmr.partition.mergesize = n；SET ngmr.partition.mergesize.mb = m

**68、**Hyperbase 与 Inceptor 的关系:Inceptor 可以访问 Hyperbase；两者相辅相成 **69、**通过 Hue 修改 HDFS 目录或文件的权限：Hdfs 相应的权限、以 hdfs 用户登录 **70、**通过 **Oozie** 使用 ssh：Oozie 用户可以免密钥登录；Oozie 用户必须要有 bash 权限 **71、**使用 sqoop 连接关系型数据时，查看关系型数据库中有哪些表：sqoop list-tables --username root --password 111111 --connect jdbc:mysql://192.168.164.25:3306/test **72、**要将采集的日志数据作为 kafka 的数据源，则 flume sink 需要设置为：org.apache.flume.sink.kafka.KafkaSink **73、**不能保证 kafka 数据可靠性的：kafka 无法保证数据可靠性 **74、**开启 LDAP 后，应该使用哪个命令连接 Inceptor：beeline -u jdbc:hive2://$ip:10000 -n $username -p $password **75、**现有一批数据需要进行清洗，要求对其中 null 通过 update 转换为 0，删除重复的记录，添加部分新的记录，则该表应该设计为Orc 事务表

**1、集群有8个节点，每个节点有8块硬盘（默认3副本）。如果某个节点有3块盘损坏，是否可能存在数据块丢失情况；如果有3个节点发生故障，是否可能存在数据块丢失情况；并简述原因。**

第一个不会，因为副本放在三个不同的节点上。第二个有可能，如果三个副本正好是在这三个节点上，则会丢失。–第1个副本放在客户端所在节点 • 如果是远程客户端，block会随机选择节点 • 系统会首先选择空闲的DataNode节点 – 第2个副本放在不同的机架节点上– 第3个副本放在与第2个副本同一机架的不同机器上

**2、请描述一个100GB文件写入HDFS的整个过程：**

1客户端发送创建文件指令给分布式文件系统。 2文件系统告知namenode • 检查权限，查看文件是否存在• EditLog增加记录• 返回输出流对象。3 客户端往输出流中写入数据，分成一个个数据包 。 4根据namenode分配，输出流往datanode写数据• 多个datanode构成一个管道pipeline，输出流写第一个，后面的转发。5每个datanode写完一个块后，返回确认信息。 6写完数据，关闭输出流。 7发送完成信号给namenode

**3、假设集群的每个节点初始有6块硬盘，运行一段时间后，每个节点又加了4块新硬盘，为了使数据在所有硬盘上分布均匀，能否通过hdfs balancer达到效果，为什么？并列出能达到效果的两种措施。** 不能，旧版本的hdfs仅支持节点间的数据平衡，新版本可通过balancer实现。措施：手动重写所有数据；将数据全部移到几个节点上，再在节点间数据平衡。

**4、请描述HDFS的高可用性实现机制：**

1.支持NameNode HA。集群可以运行两个冗余的NameNode，一个作为active节点提供服务，一个standby节点作为热备份，集群只能有一个activeNameNode为了保证standby节点与active节点之间数据的同步，两个节点会与另一组服务集群“journalNodes” (JNs)进行通信。 • 当active节点对命名空间做任何改动时，操作日志会被记录到超过半数的JournalNode上。 • standby节点会去JournalNode上读取这些日志。2.利用2N+1台JournalNode存储EditLog。3.最多容忍N台服务器挂掉。4.基于Paxos的一致性算法算法：如果要实现主备切换的自动化，还需要额外添加两个服务。5.额外一个ZooKeeper集群：• 实现NameNode故障检测 • active NameNode的选举 。 6. 每个NameNode上运行ZKFailoverController (ZKFC)服务：• NameNode健康状态的检查 • ZooKeeper会话的管理 • ZooKeeper-based选举

**5、请描述TDH平台中在Yarn上可以使用哪几种调度策略，并分别阐述各调度策略的特点。FIFO Scheduler（先进先出调度器）**：（策略）将所有任务放入一个队列，先进队列的先获得资源，排在后面的任务只有等待。(缺点)－资源利用率低，无法交叉运行任务。－灵活性差。**Capacity Scheduler（容量调度器）**：(核心思想)：提前做预算，在预算指导下分享集群资源。(调度策略)：－集群资源由多个队列分享。－每个队列都要预设资源分配的比例（提前做预算）。－空闲资源优先分配给“实际资源/预算资源”比值最低的队列－队列内部采用FIFO调度策略。(特点):－层次化的队列设计：子队列可使用父队列资源。－容量保证：每个队列都要预设资源占比，防止资源独占。－弹性分配：空闲资源可以分配给任何队列，当多个队列争用时，会按比例进行平衡。－支持动态管理：可以动态调整队列的容量、权限等参数，也可动态增加、暂停队列。－访问控制：用户只能向自己的队列中提交任务，不能访问其他队列。－多租户：多用户共享集群资源。**Fair Scheduler（公平调度器）**：（调度策略)：－多队列公平共享集群资源。－通过平分的方式，动态分配资源，无需预先设定资源分配比例。－队列内部可配置调度策略：FIFO、Fair（默认）。资源抢占+队列权重。

**6、请简述bulkload的作用和操作步骤（包括原理的步骤和使用sqlbulkload的步骤）**Hyperbase支持bulk load的入库方式，它是利用Hyperbase的数据信息按照特定格式存储在hdfs内这一原理,直接在HDFS中生成持久化的HFile数据格式文件,然后上传至合适位置,即完成巨量数据快速入库的办法。配合 mapreduce完成，高效便捷,而且不占用 region资源,增添负载,在大数据量写入时能极大的提高写入效率,并降低对 Hyperbase节点的写入压力。通过使用先生成HFile,然后再 BulkLoad到 hyperbase的方式来替代之前直接调用HTable OutputFormat的方法有如下的好处：(1)消除了对 Hyperbase集群的插入压力(2)提高了job运行速度,降低了Job的执行时间·**步骤**：1.对数据预先分region。2.根据预分region建表。3.把Hfile数据load到Hyperbase（只是文件的移动）。4.对Index数据预分Region&rebuild\_index·**使用sql 步骤**：• 第1步：将数据集上传至HDFS• 第2步：为HDFS中的数据集创建Inceptor外表。• 第3步：对外表（第2步创建）预分Region，获取Split Key。第4步：在Inceptor中创建Hyperdrive表（HBase表的二维映射表），利用Split Key（第3步获取）对HBase表预分Region。• 第5步：在Inceptor中使用SQL BulkLoad语句，将外表中的数据导入Hyperdrive表。

**7、写出一下场景下的优化思路**（1）、假设在Inceptor上执行任务，发现Map Task数量多、执行时间短，应采取哪种措施来提升性能？**答**：对数据块进行合并：Automerge（碎片自动合并）（2）、请简述在Inceptor中大表与大表做join、大表与小表做join时分别有哪些优化手段？**答**：大表与大表的普通JOIN（这里所说的普通JOIN并不单指语句中的JOIN操作，而是指它的实现方法）：实现普通JOIN的过程是这样的：扫描过滤两张表的数据（Map Stages），然后通过Shuffle将Key哈希值相同的数据分发到各个节点，在各节点内部执行JOIN（Reduce Stages）。MapJoin是一种针对大表与小表JOIN的特殊实现方式，在大小表数据量悬殊的情况下能有效的提升JOIN执行效率，一般受优化开关或者Hint控制启动。

**8、请描述一个100GB文件写入Hyperbase表的整个过程（使用bulkload方式实现）**工作流内容：1从其现有源提取数据。例如,如果您的数据位于MSQL数据库中,则可以运行 mysqldump命令。使用的进程取于数据。如果数据已经是TsV或CSV格式,请跳过此步骤,并使用随附的 ImportS实用程序将您的数据处理到HFie中。2将您的数据处理成HFe格式。3每个输出文件夹中的每个区域都会创建一个HFile,输入数据几乎全部被重写,因此需要的可用磁盘空间应至少为原始数据集大小的两倍。例如,对于100GB的 mysqldump输出,应在HDFs中至少具有不低于200GB的可用磁盘空间。可以在此进程结束时删除原始输入文件。4将文件加载到 Hyperbase.使用 LoadIncrementalHFiles命令(通常称作 completebulkload工具),并向其传递一个用于在HDFS中查找文件的URL。

**9、请列出TDH下的4大组件（Inceptor、Hyperbase、StreamSQL、Discover）的特性以及适用场景。Inceptor适用场景**：统计分析、批处理、图计算和图检索、交互式统计分析。基于Hadoop的数据仓库产品，支持SQL2003、存储过程、Oracle/DB2/ Teradata方言,分布式事务，通过标准测试TPC-H/ TPC-DS证明产品对性能的极限优化与提升，混合负载管理与SLA管控。**Hyperbase**设计目标：海量数据量存储、高并发操作、数据随机读写操作、数据强一致性；特点：大（单个表可以有数十亿行，上百万列）；无模式:每行有任意多的列,列可以动态增加,不同行可以有不同的列,列的类型没有限制；面向列族:面向列族的存储和权限控制,支持列族独立检索；稀疏:空(null)列不占存储空间,表可以非常稀疏；数据多版本:每个单元中的数据可以有多个版本数据类型单一:所有数据都是byte数组,没有类型**Streamsql**：流式sql，StreamSQL的计算运行于流计算引擎Transwarp Slipstream之上，该引擎混合了事件驱动和微批处理，因此既可以支持有低延迟需求的任务也可以处理高吞吐任务，能够应对不同类型业务。**Discover**：分布式统计学习和机器学习算法库、同时提供R语言交互式数据挖掘和数据展现功能。机器学习与数据挖掘平台，支持SQL/R/Python等开发接口，提供分布式机器学习算法，提供丰富的行业模版

**10、请描述高并发检索和综合搜索的场景特点，这两种场景应使用哪种技术来做支撑，并指出数据和索引各自的存储位置。**

1、高并发检索：海量数据存储，高并发操作，数据随机读写操作，数据强一致性。技术：使用hbase，数据可以存在hbase中，也可以存在hdfs中，索引表存在hbase中。 2、综合搜索：分布式实时文档存储，每个字段都可以被索引与搜索，分布式实时分析搜索引擎，支持高扩展，支持结构化和非结构数据。技术：使用Elasticsearch，数据和索引都存储在ES中。

**11、请以WordCount为例描述MapReduce的运行过程，并列出Spark相较MapReduce的优势**：**运行过程**：在WordCount中，Map首先处理的数据经过分片获取输入，以键值对的形式，然后通过Map的切割，切割成一个个单词，然后把每个单词的计数标记为1，并且写到环形内存缓冲区中，排序、合并，写到分区中。在Reduce段，每个Reduce把每个Map处理的数据中同一个patition的数据拷贝过来，并且经过排序、合并，数据格式类似<Hello,list(1,3,2)>，形成Reudce数据输入。最后才写到HDFS或其他渠道中。·一个split（分片）起一个map任务（通常为HDFS的一个bock大小、计算跟着数据走）·map输出的记录由一个个键值对构成,分区排序写入本地磁盘·map数据到 Reduce过程(shue)中,同一个key的所有vaue会发到同一个 reduce·reduce接受到的每个key包含一组 value,将vaue合并然后写到HDF上

Spark优势：基于内存计算（RDD）、基于DAG优化任务流程、易于部署，更低的框架开销、丰富的API支持，如Scala/java/Python

**12、请列举出平台支持的5种存储格式/引擎的表，并详细描述各自的存储特点、使用场景、支持的操作以及是否支持分区分桶。**

**一、Text表：**TEXT表即文本格式的表，是Inceptor默认的表格式。在数据量大的情况下，TEXT表的统计和查询性能都比较低；TEXT表也不支持事务处理，所以通常用于将文本文件中的原始数据导入Inceptor中。针对不同的使用场景，用户可以将**其中的数据放入ORC表或Holodesk表中。二、ORC表：**ORC表即ORC格式的表。优化的列式存储，轻量级索引，压缩比高，只支持insert。– Inceptor中数仓离线分析的主要表类型，可由Text表生成– 在Inceptor中，ORC表还分为ORC事务表和非事务表。**三、事务表：**– 支持事务处理和更多增删改语法（INSERT VALUES/UPDATE/DELETE/MERGE），所以如果您需要对表进行事务处理，应该选择使用ORC事务表。– ORC事务表是ORC分桶表的一个特例 –表属性必须有TBLPROPERTIES ("transactional"="true")–数据量特别大，建议使用分区的ORC事务表 **四、HoloDesk表：**支持交互式数据搜索、分布式内存/SSD列式存储；特性：多维度自由组合、并发度高；应用场景：增强在交互分析中Ad-hoc query的高效性、支持流应用insert & update & delete，支持分桶 **五、Hyperbase表：**定义：Hadoop Database、Google Bigtable的开源实现、key/vaue系统、分布式数据库；设计目标：海量数据量存储、高并发操作、数据随机读写操作、数据强一致性；支持明细查询、影像检索、文档检索、全局索引、全文索引、智能索引技术，支持海量结构化数据存储

**1)大数据基本特征:** 数据规模巨大、数据类型多样、生成和处理速度极快、价值巨大但密度较低。

**2)Hadoop经历了几个发展阶段，各有什么特点:** 三个，前hadoop从无到有；hadoop从有到成熟，后hadoop做了推广。3.0

**3)大数据技术体系大致分为几层？每层包含哪些技术:** 数据源层(来自电子商务、社交网、智能硬件)•数据采集层:处理结构化数据的sqoop、transporter,处理半结构化或非机构化数据的flume、kafka•数据存储管理层:HDFS分布式文件系统、HBASE、Mongodb等数据库•资源管理层:YARN等资源管理系统以及TCOS等容器化集群操作系统•通用计算层:mapreduce批处理计算框架和spark core高性能计算框架•数据分析层:spark sql数据仓库sql引擎、spark mllib人工智能、spark Streaming流处理引擎、search搜索引擎、slipstream实时流处理引擎等•数据展现层:ECharts、D3、Cboard等,以及使用zookeeper进行分布式协调服务.

**4)ApacheHadoop项目包含哪些子项目？简述一下它们的功能: HDFS**:hadoop分布式文件系统,在开源大数据体系中,地位无可替代;**MapReduce**:面向批处理的分布式计算框架,将mapreduce程序分为map和reduce两个阶段;**Spark**:高性能分布式通用计算框架;**YARN**:分布式资源管理系统,专注于资源管理和作业调度;**Docker**:开源的应用容器引擎,打包应用及依赖包到一个可移植的容器中,然后发不到任意一台linux上;**Hive**:为提供简单的数据操作而设计的下一代分布式数据仓库,它提供了简单的类似于sql的语法的hiveQL语言进行数据查询.**HBase**:分布式nosql数据库,支持海量存储;ElastisSearch:开源的分布式全文搜索引擎,基于Lucene实现全文的快速存储、搜索、分析.

**5)Spark包含哪些组件？简述一下它们的功能: Spark core**:核心计算框架,实现spark的基本功能,包括任务调度、内存管理、错误恢复与存储系统交互等,以及RDD API的定义;**spark sql**:结构化数据查询;**spark streaming**:用来对实时数据进行流式计算的组件;**spark mlib**:机器学习的功能库,提供多种学习算法,还提供了模型评估、数据导入等功能;**spark graphx**:用来操作图的程序库,可用于并行的图计算.

**6)HDFS架构哪几种角色？各自承担什么功能? ActiveNameNode（AN）**:管理命名空间、元数据、block副本策略；**StandbyNameNode（SN）**：AN宕机，快速升级为新的Active，同步元数据；**NameNode元数据文件**:fsimage(元数据检查点镜像文件)+edits(编辑日志文件)；**DataNode**：存储Block和数据校验和、定期汇报状态。

**7)为什么HDFS不合适存储大量的小文件?** 小文件过多，过多占用namenode内存，浪费block。文件过小，寻道时间大于数据读写时间，不符HDFS的设计。

**8)Block副本的放置策略是什么?** 副本1：放Client所在节点(对于远程client,系统会随机选择节点)；副本2：放不同的机架节点上；副本3：放与第二个副本同一机架的不同节点上；副本N：随机选择；节点选择：同等条件下优先空闲节点.

**9)HDFS离开安全模式的条件是什么?（只读）**Block上报率**：**DataNode上报的可用Block个数/NameNode元数据记录的Block个数。**当Block上报率>=阈值时，**HDFS才能离开安全模式，默认阈值为0.999。

**10)HDFS是如何实现高可用的?** 1)ActiveNN与StandbyNN主备切换2)利用QJM（法定人数-奇2n+1）实现元数据高可用3)利用ZooKeeper实现Active节点选举

YARN

**1)简述YARN与MapReduce的关系:** MapReduce计算框架+资源管理系统；JobTracker既做资源管理，又做任务调度；YARN将JobTracker的资源管理、任务调度功能分离；

**2)为什么要设计ApplicationMaster角色?** 管理应用程序实例；向ResourceManager申请任务执行所需的资源；任务调度和监管。

**3)Zookeeper在YARN中承担了哪些功能？**Active节点选举；恢复ActiveRM的原有状态信息。

4)在项目实践中，**如何部署**YARN的ResourceManager、NodeManager和HDFS的NameNode、DataNode？计算跟着数据走：YARN中的NodeManager（计算节点）和HDFS的DataNode（数据节点）要一一对应的部署。数据的位置暴露给计算框架，可以直接从当地节点拿数据，因为任何一个数据都会送到离他最近的一个节点去。ResourceManager、NodeManager这种管理节点需要独立部署。因为数据节点挂载上去会影响管理节点。

**5)队列在资源调度中起什么作用？**队列的状态，可以使RUNNING或者STOPPED.如果队列是STOPPED状态，那么新应用不会提交到该队列或者子队列。同样，如果root被设置成STOPPED，那么整个集群都不能提交任务了。现有的应用可以等待完成，因此队列可以优雅的退出关闭。

**6)容量调度器与公平调度器的区别是什么？**容量：提前做预算，在预算指导下分享集群资源空闲资源优先分配给“实际资源/预算资源”比值最低的队列。公平：动态分配资源，无需预先设定资源分配比例。

**7)容量调度器严格按预设比例分配资源吗？**弹性分配：空闲资源可以分配给任何队列，当多个队列争用时，会按比例进行平衡。

**8)简述公平调度器中队列权重和资源抢占的含义**。**资源抢占：**终止其他队列的任务，使其让出所占资源，然后将资源分配给占用资源量少于最小资源量限制的队列。**队列权重：**当队列中有任务等待，并且集群中有空闲资源时，每个队列可以根据权重获得不同比例的空闲资源。

分布式计算框架

**1)简述MRSplit与HDFSBlock的关系。**两者大小一样，但split是逻辑切片，block是物理切块。Split（切片）是逻辑概念，划分方式主要由程序设定，和Block没有关系，但是为了便于从HDFS中取数据，所以默认一个Block大小等于一个Split大小。两者大小一样，但split是逻辑切片，block是物理切块。

**2)为什么MapReduce要求输入输出必须是key-value键值对？**从直观的角度讲，其中的key很关键，在MR的Shuffle中是要对key做排序的，以方便进行计算。更深层次，更核心的原因是，MR作为一个通用的计算框架来说，你的数据结构的类型是要通用的，key-value就被选作为通用的数据类型，不管什么方式的数据传输进来都可以被简单的转换为key-value的形式。

**3)简述Shuffle的工作原理。（必考）Map端**－Map任务将中间结果写入专用内存缓冲区Buffer（默认100M），同时进行Partition和Sort（先按“key hashcode%reducetasknumber”对数据进行分区，分区内再按key排序）。－当Buffer的数据量达到阈值（默认80%）时，将数据溢写（Spill）到磁盘的一个临时文件中，文件内数据先分区后排序。－Map任务结束前，将多个临时文件合并（Merge）为一个Map输出文件，文件内数据先分区后排序。**Reduce端**－Reduce任务从多个Map输出文件中主动抓取（Fetch）属于自己的分区数据，先写入Buffer，数据量达到阈值后，溢写到磁盘的一个临时文件中。－数据抓取完成后，将多个临时文件合并为一个Reduce输入文件，文件内数据按key排序。

**4)从编程模型的视角，MapReduce有哪些优缺点？缺点：**仅支持Map、Reduce两种语义操作；执行效率低，时间开销大；主要用于大规模离线批处理；不适合迭代计算、交互式计算、实时流处理等场景；**优点**：不容易出错，比较稳定。

**5)RDD的“弹性”主要体现在哪里？**弹性分布式数据集，**失效后自动重构**（弹性）RDD分解成partition以后，这些分区在内存中。如果RDD失效的话，可以通过transformation自动进行重构生成（弹性）。这个弹性策略导致Spark的基础很牢固。

**6)RDD宽依赖为什么又称为Shuffle依赖？**宽依赖的依赖关系中子RDD的partition要依赖于所有的父RDD。这就表明所有的父节点的任务必须同时完成之后才能启动子节点的任务，因此形成了一种强依赖关系。这种依赖就是对MapReduce的一种再写，因此必须进行Shuffle。

**7)Spark运行模式有几种？Driver的主要功能是什么？**运行模式：抽象、Local、Standalone、YARN（clientcluster）。**Driver：**一Spark程序有一Driver，一Driver创建一SparkContext，程序的main函数运行在Driver中。负责解析Spark程序、划分Stage、调度任务到Executor上执行。

**8)简述Spark的程序执行过程。**生成逻辑计划、生成物理计划、任务调度、任务执行。在Driver中先做逻辑计划，生成RDD之间的关系；然后做物理计划，把RDD分解成partition，并且划分任务，形成有向无环图；之后交付给Scheduler去做调度；最后在Executor中执行任务。

**9)DAGScheduler是如何划分Task的？**根据任务的依赖关系建立DAG。根据依赖关系是否为宽依赖，将DAG划分为不同的阶段Stage。将各阶段中的Task组成的TaskSet提交到TaskScheduler。

分布式消息队列Kafka

**1)为什么要对Consumer进行分组？为了加快读取速度，**多个Consumer可划分为一个组（Consumer Group, CG），并行消费同一个Topic。一个Topic可以被多个CG订阅，CG之间是平等的，即一个消息可同时被多个CG消费。一个CG中可以有多个Consumer，CG中的Consumer之间是竞争关系，即一个消息在一个CG中只能被一个Consumer消费。

**2)为什么Kafka分了Topic之后，还要分Partition？**Topic：是Kafka中同一类数据的集合，相当于数据库中的表。**Topic是逻辑概念**，不必关心数据存于何处。Partition（分区）：分区内消息有序存储。一个Topic可分为多个分区，相当于把一个数据集分成多份，分别存储不同的分区中，**Parition是物理概念**，每个分区对应一个文件夹，其中存储分区的数据和索引文件。

**3)PartitionLeader和Follower是如何分工合作的？**从一个分区的多个副本中选举一个Partition Leader，Leader负责读写，其他副本作为Follower从Leader同步消息。

**4*)*为什么Zookeeper不亲自负责PartitionLeader选举？**

通过Zookeeper，从Kafka集群中选举出一个Broker作为Kafka Controller Leader。Kafka Controller Leader负责管理Kafka集群的分区和副本状态，负责Partition Leader的选举。避免分区副本直接在Zookeeper上注册Watcher和竞争创建临时Znode，导致Zookeeper集群负载过重。

分布式SQL引擎 Inceptor

**5)如何定位Inceptor？它与Hive有什么区别？定位：**用于数据仓库和交互式分析的大数据平台软件；基于Hadoop的数据仓库产品；分布式通用SQL引擎；基于Hive和Spark技术打造。**区别：**与Apache Hive相比，数据分析处理速度有显著提升。补充**特点：**Hadoop领域对SQL支持最完善；支持完整分布式事务处理MVCC；优异的大数据处理和分析性能；提供便捷的SQL、PL/SQL开发调试辅助工具Waterdrop。

**6)如何理解Inceptor读时模式。含义：**数据写入数据库时，不检查数据的规范性，而是在**查询时再验证**。**特点：**－数据**写入速度快**，适合处理大规模数据。－**查询时处理尺度很宽松**，尽可能恢复各种错误。

**7)分区目的？有几种类型？如何数据导入分区表？目的：**减少不必要的全表扫描，提升查询效率。**类型：单值分区：静态、动态分区；范围分区。**单值静态：必须手动指定目标分区；单值动态：系统动态判断目标分区，动态分区再静态分区建完之后；范围分区：均需手工指定，不支持将文件直接导入范围分区。**导入：** 1.数据预处理要求：文件编码为UTF-8，\n为换行。2.将文件导入表或分区（Load导入）：仅将数据文件移动到表或分区的目录中，不会对数据进行任何处理，如分桶、排序。不支持动态，不建议Load。3.将查询结果导入表或分区（Insert导入）。**补充：**分区表将数据按分区键的键值存储在表目录的子目录中，目录名为“分区键=键值”**。**Inceptor只支持TEXT表、ORC表、CSV表和Holodesk表的分区操作。

**8)分桶目的？如何将数据导入分桶表？（必考）目的：**通过改变数据的存储分布，提升取样、Join等特定任务的执行效率。（先分区再分桶）**导入：**按分桶键**哈希取模**的方式，将表中数据随机、均匀地分发到若干桶文件中。--分桶表在创建的时候只定义Schema，数据写入时系统不自动分桶，需要先人工分桶再写入。--写入分桶表只能通过Insert，而不能通过Load，因为Load只导入文件，并不分桶。--如果分桶表创建时定义了排序键，那么数据不仅要分桶，还要排序。--如果分桶键和排序键不同，且按降序排列，使用Distribute by Sort by分桶排序。－如果分桶键和排序键相同，且按升序排列（默认），使用Cluster by分桶排序。**补充：**• 与分区键不同，分桶键必须是表结构中的列• 分桶键和分桶数在建表时确定，不允许更改• ORC事务表必须分桶• 每个桶的文件大小应在100~200MB之间（ORC表压缩后的数据）• 先分区后分桶。

实时流处理引擎 Slipstream（事件驱动+微批处理）

**1)事件驱动模式与微批模式有什么不同？**1.相比微批模式，事件驱动模式的延迟更低，在延迟敏感的场景中表现更佳。2.微批秒级，事件驱动毫秒级**3.**微批将Input Stream按时间划分若干小数据块batch，事件驱动以单条数据被Input Stream接收为事件，逐条读取并处理。

**2)两种处理模式下的窗口变形有什么不同？**微：对一个时间窗口内的多个Batch进行计算得到新Batch的过程。＊Window Stream：通过窗口变形得到的Derived Stream。＊两个重要参数：Length和Slide，Length窗口持续时间，Slide两相邻窗口间隔时间。Length和Slide必须是Batch Duration的倍数。事件：对一个时间窗口内的多条数据进行计算得到新数据的过程。

**3)简述一下StreamJob的主要作用。**要让StreamSQL执行计划，需要有相应的Action操作来触发StreamJob。一个StreamJob启动时，StreamSQL会为每一个InputStream启动一组称为Receiver的任务来接收数据，接收来的数据经过一系列Derived Stream的变形最终被插入一张表，供用户查询。

**4)StreamSQL与普通SQL有什么区别？**1.DML语句的运行机制不同。 **普通SQL：**阻塞式运行：提交SQL后，用户需等待SQL执行结束，期间命令被持续阻塞，无法执行其他命令。**StreamSQL：**背景运行：计算任务持续在后台运行。执行StreamSQL的DML语句会立即返回结果。2.查询结果的输出不同。**普通SQL：**查询结果或者显示在Console，或者通过JDBC读取。**StreamSQL：**用户必须显式地指定查询结果输出到某个地方。后台持续运行的SQL无法直接跟Console交互。查询结果通常会插入到表中。

分布式搜索引擎 Search

**5)Search数据模型与关系数据库的对应关系？索引上：**与关系数据库的索引不同，这里是指Search的数据对象。**数据对象：**Search: index索引/ document文档/ field字段。关系：Table表/row行/column列**map映射上：**相当于关系数据库中的表结构定义（Schema）。

**6)Seach包含哪几类节点，各自负责什么工作？**主节点（MasterNode）：负责管理集群内的所有变更，如增删节点、增删索引、分配分片等，不负责文档更新和搜索。//每个集群只有一个主节点，默认情况下任何节点都可能被选为主节点。数据节点（DataNode）：负责存储数据，即文档的增删改查。//分离主节点和数据节点是一个比较好的选择，因为索引和搜索操作会消耗大量资源。客户端节点（ClientNode / 路由节点）：负责路由请求，实现集群访问的负载均衡。//集群规模大时有用，协调主节点和数据节点，据集群状态直接路由请求。

**7)简述Index、Document、Shard与副本Shard的关系。** Search以Index为单位组织数据（Document），Type是Index的逻辑分类。Document是Search的最基础数据单元，以JSON格式存储。//**Shard**是Search的数据存储单元，是数据的容器，Document保存在Shard中。Index是逻辑概念，Shard是物理概念，创建Index时会指定划分为一个或多个Shard，然后分布到集群的各节点中，Document通过“哈希取模”（Shard编号 = hash(Document id) % Shard数 量）的方式分配到不同的Shard中。Shard不可分割，但可以收缩（Shrink）。**分区副本:**Shard分为主Shard和副本Shard，后者是前者的精确复制，Index的任意一个Document都归属于一个主Shard，主Shard的数量决定了Index的最大数据量。Index建立时就必须明确主Shard数且不能修改，但副本Shard数可以随时修改。写操作只能被主Shard处理，读操作可同时被主Shard或副本Shard处理。

**8)简述Search更新文档的基本流程。**（1）客户端向Node1（路由节点）发送新建、索引或删除文档请求。（2）通过文档id确定该文档属于分片0，请求被转发到Node3，因为分片0的主分片在Node3上。（3）Node3在主分片上执行更新操作，如果成功了，Node3将请求并行转发到Node1和Node2的副本分片上，一旦所有副本分片都报告同步成功，Node3将向Node1报告更新成功，最后Node1向客户端报告成功。

分布式NewSQL数据库 Hyperbase

**1)为什么将Hyperbase表看作是一张四维表？四维表**：RowKey | 列族 | 列限定符 | 时间戳；二维表：RowKey | 列

这四维有：RowKey，列族，列限定符，时间戳。这四维被拍扁的条件是，列和列限定符合二为一变成列，同时时间戳被忽略掉，只更新最新数据。就变成了二维表：RowKey和列。

**2)为什么Hyperbase是Key-Value数据库？**1.按Key的字典序顺序存储。2.主要通过Key实现数据的增删改查，以及扫库操作。

**3)简述Table、Region、Store和StoreFile的关系。Table：**Hyperbase以“表”为单位组织数据。表由多行组成。//**Region：**系统将表水平划分（按行）为多个Region，每个Region保存表的一段连续数据。 默认每张表开始只有一个Region，随着数据不断写入，Region不断增大，当Region大小超过阀值时，当前Region会分裂成两个子Region。//**Store：**一个Region由多个Store组成，每个Store存储一个列族。Store由内存中的MemStore和磁盘中的若干StoreFile组成。//**StoreFile：**MemStore是Store的内存缓冲区，数据读写都先访问MemStore。StoreFile是MemStore的磁盘溢写文件，在HDFS中被称为HFile。当Store中的StoreFile数量超过阈值时，HRegionServer会将若干小StoreFile合并为一个大StoreFile。当Region中最大Store的大小超过阈值时，HRegionServer会将其等分为两个子Region。Client读取数据时，先找MemStore，再找StoreFile。

**4)为什么要进行Region Split和StoreFile Compaction? Region Split**：根据一定的触发条件和分裂策略，将Region划分为两个子Region的过程。**目的：**实现数据访问的负载均衡。**方法：**利用Middle Key将当前Region划分为两个等分的子Region。**条件：**当Region中最大Store的大小超过阈值时，触发Region Split。**StoreFile Compaction：**将Store中的全部或部分StoreFile合并为一个StoreFile的过程。**目的**：减少StoreFile数量，提升数据读取效率。**条件：**当Store中的StoreFile数量超过阈值，触发StoreFile Compaction。

**5)**简述HBase BulkLoad的基本过程。1.抽取：从数据源中抽取数据2.转换：利用MapReduce，将数据转换为HFile文件。－对于TSV或CSV文件，使用HBase ImportTsv工具将其转换成HFile文件－每个输出文件夹中的每个区域都会创建一个HFile文件。－HDFS中的可用磁盘空间至少为原始输入文件的两倍。例如：对于100GB的mysqldump导出文件，HDFS中至少预留不少于200GB的磁盘空间，可在任务结束后删除原始输入文件。3.加载：将HFile文件加载到HBase。－利用HBase CompleteBulkLoad工具，将HFile文件移动到HBase表的相应目录中，完成加载。

简单题 2.SQL BulkLoad的操作步骤。

第1步：将数据集上传至HDFS• 第2步：为HDFS中的数据集创建Inceptor外表。• 第3步：对外表（第2步创建）预分Region，获取Split Key。第4步：在Inceptor中创建Hyperdrive表（HBase表的二维映射表），利用Split Key（第3步获取）对HBase表预分Region。• 第5步：在Inceptor中使用SQL BulkLoad语句，将外表中的数据导入Hyperdrive表。

3.THD的4大组件的特性和使用场景。

Inceptor:**特性:**hadoop领域对sql支持最完善、支持完整分布式事务处理、优异的大数据处理和分析性能、提供便捷的sql、pl/sql开发调试辅助工具waterdrop. **使用场景:**离线分析、批处理、图计算与图检索、交互式分析

Hyperbase:**特性:**海量数据存储、线性扩展、高并发、高可用、实时随机读写、数据强一致性.**适用场景:高并发查询、半结构化/结构化/非结构化数据存储**

StreamSQL:**特性:**微批模式和时间驱动模式一体化,极高的易用性,性能提升,产品化程度高,迁移成本低. **适用场景:**包括ETL、规则报警工具在内的使用流处理的几乎所有类型的业务场景.

Discover:机器学习库,比spark mllib更加强大.

其他重点：

Sqoop是一个主要在Hadoop和关系数据库之间进行批量数据迁移的工具。Flume：是一个分布式海量数据采集、聚合和传输系统。Event：事件，最小数据传输单元，由Header和Body组成。Agent：代理，JVM进程，最小运行单元，由Source、Channel、Sink三个基本组件构成，负责将外部数据源产生的数据以Event的形式传输到目的地。映射关系：1个Source多个Channel，1个Channel多个Sink，1个Sink1个Channel。

分布式消息队列Kafka**一、**一个Topic可分为多个Partition，仅保证同一分区内消息有序存储，不保证Topic整体有序。

**二、**Kafka索引：偏移量和时间戳。

**知识回顾与总结: 一、**哈希取模在哪些技术中使用过，分别发挥什么作用? 1)MapReduce: Map任务将中间结果写入专用内存缓冲区Buffer，同时进行Partition(先按“key hashcode % reduce task number”对数据进行分区，分区内再按key排序) 2)Sqoop: 从Oracle或DB2导入数据时，利用哈希取模实现数据均匀切片 3)Hive/Inceptor:利用Select...Distributeby...Sortby(Clusterby)实现数据分桶. 4)Search / ElasticSearch :将Document分入不同的Shard

二、ZooKeeper在哪些技术和产品中使用过?分别起什么作用? 1)HDFS: NameNode HA:Active NN选举 2)YARN: ResourceManger HA: Active RM选举、存储元数据 3)Kafka:存储元数据、配置管理、Broker 动态扩展、Broker负载均衡、Controller Leader选举，以及Consumer Group变化时的Rebalance 4)Hyperbase :HMaster选举、存储元数据入口地址5)SolrCloud

三、计算框架与资源管理系统是如何协同工作的? 1)计算框架:-本质:编程模型 -负责画出分布式作业的执行图纸 2)资源管理框架:-本质:管理和调度系统 -负责按照图纸，将代码转化为基于DAG的任务集合

四、分治思想主要体现在哪些方面，分别有哪些具体应用? 1)架构层面:大数据架构分为数据来源层、数据传输层、数据存储层、资源管理层、数据计算层、任务调度层.不同层之间分工协作. 2)计算层面:运用分布式技术进行并行计算,大幅缩减计算时间.

3)数据层面:将很大的数据块打散到不同的节点上取存储.

五、大数据技术的学习在方法论层面给了我们哪些启迪? 基本方法论:发现问题--分析问题--解决问题

**发现问题**:-发现最本质、最核心的问题 -问题代表着方向(发现好的问题比后两者更重要) -大数据的核心问题是什么?数据和计算的规模与多样 VS 单体硬件+单体软件架构的瓶颈 **分析问题**:-紧紧围绕问题，充分地做功课，包括来龙去脉、相关知识、前人的工作 -梳理出多种的解决思路，形成解决问题的大思路 -大数据技术的解决思路:基于分布式理论，在架构、数据和计算等方面进行分治，用单体组成集群 **解决问题**:-不重复造轮子 -逢山开路、遇水搭桥