**Hadoop2.0基础实验课问答题**

1. 请从技术的角度谈谈Hadoop出现的历史背景。
   * 系统瓶颈：存储容量，读写速率，计算效果
   * Google:MapReduce,BigTable,GFS
   * 成本，软件容错，简化并行分布式计算
2. 请从整体架构的角度说明一下Hadoop2.0比较Hadoop1.0有哪些重要的改进。
   * Hadoop 1.0 中的资源管理静态资源配置，资源无法共享， 资源划分粒度过大，没引入有效的资源隔离机制。
   * Hadoop 2.0 调度器则按照任务实际需求为其精细地分配对应的资源量，还引入了基于cgroups的轻量级资源隔离方案，这大大降低了同节点上任务间的相互干扰，而Hadoop 1.0仅采用了基于JVM的资源隔离，粒度非常粗糙。
   * Hadoop 1.0，内核主要由HDFS和MapReduce两个系统组成
   * Hadoop 2.0，内核主要由HDFS、MapReduce和YARN三个系统组成
3. HDFS是典型的Master/Slave架构，请对NameNode和DataNode的关系及二者的作用给出解释。
   * namenode
     1. 管理文件系统文件的元数据信息（包括文件名称、大小、位置、属性、创建时间、修改时间等等）
     2. 维护文件到块的对应关系和块到节点的对应关系
     3. 维护用户对文件的操作信息（文件的增删改查）
   * datanode主要是存储数据的
4. “SecondNameNode是NameNode的一个热备”这句话对吗？请说明你的理由。
   * 不是。snn是HDFS架构中的一个组成部分，
     1. 用来保存namenode中对HDFS metadata的信息的备份，并减少namenode重启的时间。
     2. hadoop：没有对任何一次的当前文件系统的snapshot进行持久化，对HDFS最近一段时间的操作list会被保存到namenode中的一个叫Editlog的文件中去。当重启namenode时，除了 load fsImage意外，还会对这个EditLog文件中 记录的HDFS操作进行replay，以恢复HDFS重启之前的最终状态。而SecondaryNameNode，会周期性的将EditLog中记录的对HDFS的操作合并到一个checkpoint中，然后清空 EditLog。所以namenode的重启就会Load最新的一个checkpoint，并replay EditLog中记录的hdfs操作，由于EditLog中记录的是从 上一次checkpoint以后到现在的操作列表，所以就会比较小。
     3. 如果没有snn的这个周期性的合并过程，那么当每次重启namenode的时候，就会 花费很长的时间。而这样周期性的合并就能减少重启的时间。同时也能保证HDFS系统的完整性。
     4. snn并不能分担namenode上对HDFS交互性操作的压力。尽管如此，当namenode机器宕机或者namenode进程出问题时，namenode的daemon进程可以通过人工的方式从snn上拷贝一份metadata来恢复HDFS文件系统。
5. 在HDFS的元数据管理过程中，fsImage和Edits Log分别起到什么作用？
   * fsimage：文件是文件系统元数据的一个永久性检查点，包含文件系统中的所有目录和文件idnode的序列化信息。
   * 文件系统的写操作首先把它记录在edit中。
6. 请详细说明NameNode元数据管理中一次checkpoint过程的完整步骤。
   * 辅助Namenode请求主Namenode停止使用edits文件，暂时将新的写操作记录到一个新文件中，如edits.new。
   * 辅助Namenode节点从主Namenode节点获取fsimage和edits文件（采用HTTP GET）
   * 辅助Namenode将fsimage文件载入到内存，逐一执行edits文件中的操作，创建新的fsimage文件
   * 辅助Namenode将新的fsimage文件发送回主Namenode（使用HTTP POST）
   * 主Namenode节点将从辅助Namenode节点接收的fsimage文件替换旧的fsimage文件，用步骤1产生的edits.new文件替换旧的edits文件（即改名）。同时更新fstime文件来记录检查点执行的时间
7. Name Node和DataNode数据的保存位置各自存放在哪里？该位置信息在哪个配置文件中定义？
   * dfs.namenode.name.dir
   * dfs.datanode.data.dir
   * hdfs-site.xml
8. 请详细说明HDFS文件的读取过程。
   * 使用HDFS提供的客户端开发库Client，向远程的Namenode发起RPC请求；
   * Namenode会视情况返回文件的部分或者全部block列表，对于每个block，Namenode都会返回有该block拷贝的DataNode地址；
   * 客户端开发库Client会选取离客户端最接近的DataNode来读取block；如果客户端本身就是DataNode,那么将从本地直接获取数据.
   * 读取完当前block的数据后，关闭与当前的DataNode连接，并为读取下一个block寻找最佳的DataNode；
   * 当读完列表的block后，且文件读取还没有结束，客户端开发库会继续向Namenode获取下一批的block列表。
   * 读取完一个block都会进行checksum验证，如果读取datanode时出现错误，客户端会通知Namenode，然后再从下一个拥有该block拷贝的datanode继续读。
9. 请详细说明HDFS文件的写入过程。
   * 使用HDFS提供的客户端开发库Client，向远程的Namenode发起RPC请求；
   * Namenode会检查要创建的文件是否已经存在，创建者是否有权限进行操作，成功则会为文件创建一个记录，否则会让客户端抛出异常；
   * 当客户端开始写入文件的时候，开发库会将文件切分成多个packets，并在内部以数据队列”data queue”的形式管理这些packets，并向Namenode申请新的blocks，获取用来存储replicas的合适的datanodes列表，列表的大小根据在Namenode中对replication的设置而定。
   * 开始以pipeline（管道）的形式将packet写入所有的replicas中。开发库把packet以流的方式写入第一个datanode，该datanode把该packet存储之后，再将其传递给在此pipeline中的下一个datanode，直到最后一个datanode，这种写数据的方式呈流水线的形式。
   * 最后一个datanode成功存储之后会返回一个ack packet，在pipeline里传递至客户端，在客户端的开发库内部维护着”ack queue”，成功收到datanode返回的ack packet后会从”ack queue”移除相应的packet。
   * 如果传输过程中，有某个datanode出现了故障，那么当前的pipeline会被关闭，出现故障的datanode会从当前的pipeline中移除，剩余的block会继续剩下的datanode中继续以pipeline的形式传输，同时Namenode会分配一个新的datanode，保持replicas设定的数量。
10. 请写出对HDFS目录进行递归删除以及从本地递归复制一个目录到HDFS的shell命令。
    * hadoop fs -rm -r /user
    * hadoop fs -put ~/dir /dir
11. 请说明dfsadmin管理命令都能实现哪些功能。  
    获取HDFS状态信息，和管理操作
12. 在安装Hadoop2.0时，需要在$HADOOP\_HOME下创建哪几个子目录？说明这些目录的用途。
    * tmp 临时文件
    * name目录只是存放HDFS文件系统的元数据
    * data目录是要存储HDFS的文件内容
    * logs 日志文件
13. 需要修改的YARN的配置文件总共有7个，请分别说明这些文件的名称和用途。
    * yarn.nodemanager.aux-services mapreduce\_shuffle
    * yarn.nodemanager.aux-services.mapreduce.shuffle.class org.apache.hadoop.mapred.ShuffleHandler
    * yarn.resourcemanager.address fqiyou01:8032
    * yarn.resourcemanager.scheduler.address fqiyou01:8030
    * yarn.resourcemanager.resource-tracker.address fqiyou01:8031
    * yarn.resourcemanager.admin.address fqiyou01:8033
    * yarn.resourcemanager.webapp.address fqiyou01:8088
14. 在首次使用HDFS之前，为什么要进行HDFS的格式化操做？
    * 让存储空间明白该按什么方式组织存款数据
15. 请对Pig的功能做一个简要的说明。
    * 探索大型数据集的脚本语言
16. Hive的结构可分为4部分，请分别说明它们的名称和作用。
    * 用户接口主要有三个：CLI，Client 和 WUI。其中最常用的是CLI，Cli启动的时候，会同时启动一个Hive副本。Client是Hive的客户端，用户连接至Hive Server。在启动 Client模式的时候，需要指出Hive Server所在节点，并且在该节点启动Hive Server。 WUI是通过浏览器访问Hive。
    * Hive将元数据存储在数据库中，如MySQL、derby。Hive中的元数据包括表的名字，表的列和分区及其属性，表的属性（是否为外部表等），表的数据所在目录等。
    * 解释器、编译器、优化器完成HQL查询语句从词法分析、语法分析、编译、优化以及查询计划的生成。生成的查询计划存储在HDFS中，并在随后有MapReduce调用执行。
    * Hive的数据存储在HDFS中，大部分的查询、计算由MapReduce完成（包含\*的查询，比如select \* from tbl不会生成MapRedcue任务）。
17. 请对Hive和Hadoop的关系做一个简要的说明。
    * hive是Hadoop的一个组件，作为数据厂库，hive的数据是存储在Hadoop的文件系统中的，hive为Hadoop提供SQL语句，是Hadoop可以通过SQL语句操作文件系统中的数据。hive是依赖Hadoop而存在的。
18. 请对Hive和RDBMS在整体上做一个对比说明。
    * 查询语言。由于 SQL 被广泛的应用在数据仓库中，因此，专门针对 Hive 的特性设计了类 SQL 的查询语言 HQL。熟悉 SQL 开发的开发者可以很方便的使用 Hive 进行开发。
    * 数据存储位置。Hive 是建立在 Hadoop 之上的，所有 Hive 的数据都是存储在 HDFS 中的。而数据库则可以将数据保存在块设备或者本地文件系统中。
    * 数据格式。Hive 中没有定义专门的数据格式，数据格式可以由用户指定，用户定义数据格式需要指定三个属性：列分隔符（通常为空格、”\t”、”\x001″）、行分隔符（”\n”）以及读取文件数据的方法（Hive 中默认有三个文件格式 TextFile，SequenceFile 以及 RCFile）。由于在加载数据的过程中，不需要从用户数据格式到 Hive 定义的数据格式的转换，因此，Hive 在加载的过程中不会对数据本身进行任何修改，而只是将数据内容复制或者移动到相应的 HDFS 目录中。而在数据库中，不同的数据库有不同的存储引擎，定义了自己的数据格式。所有数据都会按照一定的组织存储，因此，**RDBMS数据库加载数据的过程会比较耗时**。
    * 数据更新。由于 Hive 是针对数据仓库应用设计的，而数据仓库的内容是读多写少的。因此，**Hive 中不支持对数据的改写和添加**，所有的数据都是在加载的时候中确定好的。而数据库中的数据通常是需要经常进行修改的，因此可以使用 INSERT INTO … VALUES 添加数据，使用 UPDATE … SET 修改数据。
    * 索引。之前已经说过，Hive 在加载数据的过程中不会对数据进行任何处理，甚至不会对数据进行扫描，因此也没有对数据中的某些 Key 建立索引。Hive 要访问数据中满足条件的特定值时，需要暴力扫描整个数据，因此访问延迟较高。由于 MapReduce 的引入， Hive 可以并行访问数据，因此即使没有索引，对于大数据量的访问，Hive 仍然可以体现出优势。数据库中，通常会针对一个或者几个列建立索引，因此对于少量的特定条件的数据的访问，数据库可以有很高的效率，较低的延迟。由于数据的访问延迟较高，决定了 **Hive 不适合在线数据查询**。
    * 执行。Hive 中大多数查询的执行是通过 Hadoop 提供的 MapReduce 来实现的（类似 select \* from tbl 的查询不需要 MapReduce）。而数据库通常有自己的执行引擎。
    * 执行延迟。之前提到，Hive 在查询数据的时候，由于没有索引，需要扫描整个表，因此延迟较高。另外一个导致 Hive 执行延迟高的因素是 MapReduce 框架。由于 MapReduce 本身具有较高的延迟，因此在利用 MapReduce 执行 Hive 查询时，也会有较高的延迟。相对的，数据库的执行延迟较低。当然，这个低是有条件的，即数据规模较小，当数据规模大到超过数据库的处理能力的时候，Hive 的并行计算显然能体现出优势。**hive执行延迟高，只有在数据规模达到一定程度后，其查询的高效才能弥补其高延迟的劣势。**
    * 可扩展性。由于 Hive 是建立在 Hadoop 之上的，因此 Hive 的可扩展性是和 Hadoop 的可扩展性是一致的（世界上最大的 Hadoop 集群在 Yahoo!，2009年的规模在 4000 台节点左右）。而数据库由于 ACID 语义的严格限制，扩展行非常有限。目前最先进的并行数据库 Oracle 在理论上的扩展能力也只有 100 台左右。
    * 数据规模。由于 Hive 建立在集群上并可以利用 MapReduce 进行并行计算，因此可以支持很大规模的数据；对应的，数据库可以支持的数据规模较小。
19. Hive的元数据为什么要保存在关系型数据库比如MySQL中？
    * Hive 不支持对数据的改写和添加
20. 请对Hive的数据存储做一个完整的描述。
    * Hive没有专门的数据存储格式，也没有为数据建立索引，用户可以非常自由的组织Hive中的表，只需要在创建表的时候告诉Hive数据中的列分隔符和行分隔符即可，然后Hive就可以解析数据了。Hive中所有的数据都存储在HDFS中，Hive包括的数据模型有Table、Externel Table、Partition、Bucket。
    * Hive中的Table和关系型数据库中的Table在概念上类似，Hive中的每一个Table都有一个相应的目录存储其数据。
    * Partition对应于关系型数据库中的Partition列的索引，但Hive中Partition的组织方式和数据库中的有很大不同。在Hive中，表的一个Partition对应表下的一个目录，所有的Partition数据都存储在对应的目录中。
    * Buckets对指定列计算hash，根据hash值分组数据，目的是为了并行处理，每一个Bucket对应一个文件。
    * Externel Table是指向已经在HDFS中存在的数据，为它创建Partition、 Externel Table和Table在元数据的组织上是相同的，而实际数据的存储差异很大。
21. MySQL的安装过程包括了哪几个主要步骤？如何启动MySQL?
    * 首先查看当前系统是否已经安装过MySQL
    * 卸载掉系统默认安装的MySQL
    * 切换到/stage目录，安装MySQL,再分别安装MySQL-client和MySQL-devel  
      启动MySQL：使用service mysql start命令
22. Hive的安装需要修改哪些配置文件？  
    hive-env.sh、hive-site.xml
23. 请给出启动Hive的命令。
    * schematool –dbType mysql –initSchema
    * hive -service metastore &
    * hive -service hiveserver2 &
24. 请对Zookeeper的功能和其在Hadoop体系中的地位做一个说明。
    * Zookeeper可以为分布式应用提供一致性的服务，功能包括：配置维护、名字服务、分布式同步、组服务等
    * Zookeeper是一个开源的分布式应用程序协调服务，是Hadoop生态系统中的基础组件。
25. 请的Zookeeper的工作原理做一个简要的说明。  
    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。
26. 请对Zookeeper的单机模式、伪分布式以及分布式集群模式之间的不同做一个说明。
    * 单机模式：Zookeeper只运行在一台服务器上，适合测试环境
    * 伪集群模式：在一台物理机上运行多个Zookeeper实例
    * 集群模式：Zookeeper运行在一个集群上，适合生产环境
27. 在集群配置模式下，如何启动Zookeeper？
    * 需要所有主机都对JAVA环境进行设置，在每台机器上的conf/zoo.cfg篇日志的文件参数相同，分别在每台机器上启动Zookeeper服务
28. Zookeeper服务器在部署时一般要求为奇数，请说明原因。
    * zookeeper有这样一个特性：集群中只要有过半的机器是正常工作的，那么整个集群对外就是可用的。所以最好使用奇数台机器，例如如果Zookeeper拥有5台机器，系统就能处理2台机器的故障了。
29. 请列举出HBase数据库的主要特点，并加以简要的说明。
    * 数据类型:HBase只有简单的字符串类型，所有类型都是交由用户自己处理，HBase只保存字符串。
    * 数据操作：HBase的操作只有简单的插入、查询、删除、清空等，表和表之间是分离的，没有复杂的表和表之间的关系，所以也不能实现表和表之间的关联等操作。
    * 存储模式：HBase是基于列存储的，每个列族都有几个文件保存。不同列族的文件是分离的。
30. 在HBase上的操作都有哪些？请使用示例分别加以说明。
    * HBase表的管理，
      1. list –查看当前已有的表
      2. describe—查看表结构
      3. create—创建表
      4. drop—删除表
    * 查看HBase服务状态和版本,
      1. status—状态
      2. Version—版本
    * HBase的DML操作，
      1. put—添加数据
      2. get—查询数据
      3. delete—删除数据
31. HBase的数据在物理上是如何在HDFS中存放的？
    * 每个Column Family存储在HDFS上的一个单独文件中
    * Key和Version Number在每个Column Family中均有一份
    * 空值不会被保存
32. 随着HBase中数据的不断增加，大Region会不断被分割成更小的Region, 请说明原因。
    * Region按大小分割的，每个表开始只有一个region，随着数据增多，region不断增大， 当增大到一个阀值的时候， region就会等分会两个新的region，之后会有越来越多的 region，Region是HBase中分布式存储和负载均衡的最小单元。不同Region分布到不同RegionServer上
33. 请对HMaster, HRegionServer和HRegoin的概念和三者的关系加以详细说明。
    * HMaster:为Region server分配region,负责Region server的负载均衡,发现失效的Region server并重新分配其上的region,管理用户对table的增删改查操作
    * HRegion Server:Region server维护region，处理对这些region的IO请求,Region server负责切分在运行过程中变得过大的region
    * HRegion：Region是HBase中分布式存储和负载均衡的最小单元，不同Region分布到不同RegionServer上
    * 每台HRegion服务器都会和HMaster服务器通信，HMaster的主要任务就是要告诉每台HRegion服务器他要维护哪些HRegion。当一台新的HRegion服务器登录到HMaster服务器时，HMaster会告诉它先等待分配数据。而当一台HRegion服务器死机时，HMaster会把他负责的HRegion标记为未分配，然后再把它们分配到其他的HRegion服务器中。
34. 请说明在访问一个具体列值时，HBase是如何定位一个regoin的。
    * 每个region需要一个唯一的regionID来表示它，HRegion的表达式是：表名+开始主键+唯一ID，可以用这个标识符来区分不同的HRegion。我来定位这些HRegion，采用一个根数据表，它保存了所有的元数据表的位置，而根数据表是不能被分割的，永远只存在于一个HRegion。表中的项使用区域名作为键值，区域名有所属的表名、区域的起始行、区域的创建时间以及对整体进行MD5哈希值组成。
35. 对于Sqoop1和Sqoop2，请说明二者之间在部署与使用上的区别。
    * Sqoop1 基于客户端模式，需要在客户端安装和配置
    * Sqoop2 只需要在服务器安装和配置，连接器在一个地方进行配置，由管理员管理
36. 请分别说明Sqoop导入和导出操作的步骤
    * 导入
      1. sqoop list-databases –connect jdbc:mysql://hadoop1:3306/ –username hive –password hive
      2. sqoop import –connect jdbc:mysql://hadoop1:3306/ –username hive –password hive –table TBLS -m 1
    * 导出
      1. hive –service metastore &
      2. hive –service hiveserver2 &
      3. sqoop import –connect jdbc:mysql://hadoop1:3306/ –username hive –password hive –table SDS –hive-table Mysql2Hive –hive-import -m 1