《Hadoop 开发 - 就业技能》

章节	Ch01 - 大数据及 Hadoop 概述
题目 1)	**什么是大数据**
	大数据是指无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合。大数据技术，是指从各种各样类型的数据中，快速获得有价值信息的能力。适用于大数据的技术，包括大规模并行处理（MPP）数据库，数据挖掘电网，分布式文件系统，分布式数据库，云计算平台，互联网，和可扩展的存储系统。
题目 2)	**大数据的特点**
	大数据具有 4 个基本特征： 一是数据体量巨大。百度资料表明，其新首页导航每天需要提供的数据超过 1.5PB（1PB=1024TB），这些数据如果打印出来将超过 5 千亿张 A4 纸。有资料证实，到目前为止，人类生产的所有印刷材料的数据量仅为 200PB。 二是数据类型多样。现在的数据类型不仅是文本形式，更多的是图片、视频、音频、地理位置信息等多类型的数据，个性化数据占绝对多数。 三是处理速度快。数据处理遵循“1 秒定律”，可从各种类型的数据中快速获得高价值的信息。 四是价值密度低。以视频为例，一小时的视频，在不间断的监控过程中，可能有用的数据仅仅只有一两秒。
题目 3)	**Haddop 概念**
	Hadoop 是一个开源的可运行于大规模集群上的分布式文件系统和运行处理基础框架 Hadoop 擅长于在廉价机器搭建的集群上进行海量数据(结构化与非结构化)的存储与离线处理。 Hadoop 就是一门用来处理大数据的技术
题目 4)	**Hadoop 的核心组件**
	1.海量存储——HDFS(Hadoop 分布式文件系统，Hadoop Distributed File System) 分布式易扩展，廉价易得，高吞吐量，高可靠性 2.分布式并行计算——资源调度(Yarn)+编程模型(MapReduce) 大容量高并发，封装分布式实现细节，大大提高分析效率
题目 5)	**Hadoop 体系的核心项目**
	1. HDFS: Hadoop 分布式文件系统(Distributed File System) - HDFS (Hadoop Distributed File System) 2. MapReduce：并行计算框架，0.20 前使用 org.apache.hadoop.mapred 旧接口，0.20 版本开始引入 org.apache.hadoop.mapreduce 的新 API 3. HBase: 类似 Google BigTable 的分布式 NoSQL 列数据库。(HBase 和 Avro 已经于 2010 年 5 月成为顶级 Apache 项目) 4. Hive：数据仓库工具，由 Facebook 贡献。 5. Zookeeper：分布式锁设施，提供类似 Google Chubby 的功能，由 Facebook 贡献。 6. Avro：新的数据序列化格式与传输工具，将逐步取代 Hadoop 原有的 IPC 机制。

	7. Pig: 大数据分析平台，为用户提供多种接口。 8. Ambari：Hadoop 管理工具，可以快捷的监控、部署、管理集群。 9. Sqoop：于在 HADOOP 与传统的数据库间进行数据的传递。

章节	**Ch02 - 分布式文件系统 HDFS**
题目 1)	**简述分布式文件系统**
	多台计算机联网协同工作(有时也称为一个集群)就像单台系统一样解决某种问题，这样的系统我们称之为分布式系统。 分布式文件系统是分布式系统的一个子集，它们解决的问题就是数据存储。换句话说，它们是横跨在多台计算机上的存储系统。存储在分布式文件系统上的数据自动分布在不同的节点上。分布式文件系统在大数据时代有着广泛的应用前景，它们为存储和处理来自网络和其它地方的超大规模数据提供所需的扩展能力。
题目 2)	**HDFS 是什么**
	HDFS 即 Hadoop 分布式文件系统（Hadoop Distributed Filesystem)，以流式数据访问模式来存储超大文件，运行于商用硬件集群上，是管理网络中跨多台计算机存储的文件系统。 HDFS 不适合用在：要求低时间延迟数据访问的应用，存储大量的小文件，多用户写入，任意修改文件。
题目 3)	**HDFS 数据块**
	HDFS 上的文件被划分为块大小的多个分块，作为独立的存储单元，称为数据块，默认大小是 64MB。 使用数据块的好处是： 1.一个文件的大小可以大于网络中任意一个磁盘的容量。文件的所有块不需要存储在同一个磁盘上，因此它 们可以利用集群上的任意一个磁盘进行存储。 2.简化了存储子系统的设计，将存储子系统控制单元设置为块，可简化存储管理，同时元数据就不需要和块 一同存储，用一个单独的系统就可以管理这些块的元数据。 3.数据块适合用于数据备份进而提供数据容错能力和提高可用性。
题目 4)	**HDFS 的三种节点**
	1.Namenode：HDFS 的守护进程，用来管理文件系统的命名空间，负责记录文件是如何分割成数据块，以及这些数据块分别被存储到那些数据节点上，它的主要功能是对内存及 IO 进行集中管理。 2.Datanode：文件系统的工作节点，根据需要存储和检索数据块，并且定期向 namenode 发送他们所存储的块的列表。 3.Secondary Namenode：辅助后台程序，与 NameNode 进行通信，以便定期保存 HDFS 元数据的快照。
题目 5)	**HDFS 读数据步骤详解**
	1）客户端向 namenode 发起 RPC 调用，请求读取文件数据。 2）namenode 检查文件是否存在，如果存在则获取文件的元信息（blockid 以及对应的 datanode 列表)。 3）客户端收到元信息后选取一个网络距离最近的 datanode，依次请求读取每个数据块。客户端首先要校检文件是否损坏，如果损坏，客户端会选取另外的 datanode 请求。

	4）datanode 与客户端简历 socket 连接，传输对应的数据块，客户端收到数据缓存到本地，之后写入文件 。 5）依次传输剩下的数据块，直到整个文件合并完成。

章节	**Ch03 - 分布式计算框架 MapReduce**
题目 1)	**MapReduce 定义**
	Hadoop 中的 MapReduce 是一个使用简单的软件框架，基于它写出来的应用程序能够运行在由上千个机器组成的大型集群上，并以一种可靠容错并行处理 TB 级别的数据集
题目 2)	**MapReduce 特点**
	1)MapReduce 易于编程 它简单的实现一些接口，就可以完成一个分布式程序，这个分布式程序可以分布到大量廉价的 PC 机器运行。也就是说你写一个分布式程序，跟写一个简单的串行程序是一模一样的。就是因为这个特点使得 MapReduce 编程变得非常流行。 2)良好的扩展性 当你的计算资源不能得到满足的时候，你可以通过简单的增加机器来扩展它的计算能力。 3)高容错性 MapReduce 设计的初衷就是使程序能够部署在廉价的 PC 机器上，这就要求它具有很高的容错性。比如其中一台机器挂了，它可以把上面的计算任务转移到另外一个节点上面上运行，不至于这个任务运行失败，而且这个过程不需要人工参与，而完全是由 Hadoop 内部完成的。 4)适合 PB 级以上海量数据的离线处理 它适合离线处理而不适合在线处理。比如像毫秒级别的返回一个结果，MapReduce 很难做到
题目 3)	**介绍 MapReduce 框架的容错性**
	MapReduce 最大的特点之一就是有很好的容错性，即使你的节点挂掉了 1 个、2 个、3 个，都是没有问题的， 它都可以照常来运行，把你的作业或者应用程序运行完成。不会出现某个节点挂了，你的作业就运行失败这种情况。 那么 MapReduce 到底是通过什么样的机制，使它具有这么好的容错性呢？下面我们依次来介绍一下。 1、JobTracker 很不幸，JobTracker 存在单点故障，一旦出现故障，整个集群就不可用。这个是 1.0 里面出现的问题，在 2.0 里面这个问题已经得到了解决。不过大家放心，即使在 1.0 中，MapReduce 也不会经常出现故障。它可能一年也就是出现几次故障，出现故障之后，你重启一下，再把作业重新提交就可以了，它不会像 HDFS 那样出现数据的丢失。 因为 MapReduce 是一个计算框架，计算过程是可以重现的，即使某个服务挂掉了，你重启一下服务，然后把作业重新提交，也是不会影响你的业务的。 2、TaskTracker TaskTracker 周期性的向 JobTracker 汇报心跳，如果一定的时间内没有汇报这个心跳，JobTracker 就认为该 TaskTracker 挂掉了，它就会把上面所有任务调度到其它 TaskTracker（节点）上运行。这样即使某个节点挂了，也不会影响整个集群的运行。 3、MapTask 和 ReduceTask MapTask 和 ReduceTask 也可能运行挂掉。比如内存超出了或者磁盘挂掉了，这个任务也就挂掉了。这个时候 TaskTracker 就会把每个 MapTask 和 ReduceTask 的运行状态回报

	给 JobTracker，JobTracker 一旦发现某个 Task 挂掉了，它就会通过调度器把该 Task 调度到其它节点上。这样的使任务挂掉了，也不会影响应用程序的运行
题目 4)	**Mapreduce 执行过程**
	1.mapper 开始运行，调用 InputFormat 组件读取文件逻辑切片（逻辑切片不是 block 块，切片大小默认和 block 块大小相同） 2.经过 inputformat 组件处理后，文件以<k,v>的形式进入我们自定义的 mapper 逻辑 3.mapper 逻辑中输出结果会调用 OutPutCollector 组件写入环形缓冲区。 4.环形缓冲区的存储达到默认阀值会调用 Splliller 组件将内容分区且排序（快排算法，外部排序算法）后溢写到磁盘文件中，mapper 组后结果不满环形缓冲区也会溢写到磁盘。 5.mapper 结束后磁盘中的结果小文件会合并（merge），产生大文件（分区且排序，归并算法）。 6.reducer 启动后会到不同的 map 结果文件中下载相同区号的结果文件，再合并这些来自不同 map 的结果文件，再将这些文件合并（归并算法），产生的大文件是分区且排序且分好组了的，分组调用默认的 GroupingComparator 组件。 7.reducer 把下载的所有 map 输出文件合并完成之后就会开始读取文件，将读入的内容以<k,v>的形式输入到我们用户自定义的 reducer 处理逻辑中。 8.用户逻辑完成之后以<k,v>的形式调用OutPutFormat组件输出到hdfs文件系统中去保存。
题目 5)	**MapReduce 和 YARN 的关系**
	YARN 并不是下一代 MapReduce（MRv2），下一代 MapReduce 与第一代 MapReduce（MRv1）在编程接口、数据处理引擎（MapTask 和 ReduceTask）是完全一样的， 可认为 MRv2 重用了 MRv1 的这些模块，不同的是资源管理和作业管理系统，MRv1 中资源管理和作业管理均是由 JobTracker 实现的，集两个功能于一身，而在 MRv2 中，将这两部分分开了，其中，作业管理由 ApplicationMaster 实现，而资源管理由新增系统 YARN 完成，由于 YARN 具有通用性，因此 YARN 也可以作为其他计算框架的资源管理系统，不仅限于 MapReduce，也是其他计算框架，比如 Spark、Storm 等， 通常而言，我们一般将运行在 YARN 上的计算框架称为“X on YARN”，比如“MapReduce On YARN”，“Spark On YARN”，“Storm On YARN”等

章节	**Ch04 - 分布式列式数据库 HBase（一）**
题目 1)	**Hbase 是什么**
	HBase – Hadoop Database，是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统，利用 HBase 技术可在廉价 PC Server 上搭建起大规模结构化存储集群。
题目 2)	**HBase 的特点**
	1)大：一个表可以有上亿行，上百万列。 2)面向列：面向列表（簇）的存储和权限控制，列（簇）独立检索。 3)稀疏：对于为空（NULL）的列，并不占用存储空间，因此，表可以设计的非常稀疏。 4)无模式：每一行都有一个可以排序的主键和任意多的列，列可以根据需要动态增加，同一张表中不同的行 可以有截然不同的列。 5)数据多版本：每个单元中的数据可以有多个版本，默认情况下，版本号自动分配，版本号就是单元格插入时的时间戳。 6)数据类型单一：HBase 中的数据都是字符串，没有类型。
题目 3)	**Hbase Master 作用**
	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
题目 4)	**Hbase Region Server 作用**
	Regionserver 维护 region，处理对这些 region 的 IO 请求 Regionserver 负责切分在运行过程中变得过大的 region
题目 5)	**Zookeeper 作用**
	通过选举，保证任何时候，集群中只有一个 master，Master 与 RegionServers 启动时会向 ZooKeeper 注册 存贮所有 Region 的寻址入口 实时监控 Region server 的上线和下线信息。并实时通知给 Master 存储 HBase 的 schema 和 table 元数据 默认情况下，HBase 管理 ZooKeeper 实例，比如， 启动或者停止 ZooKeeper Zookeeper 的引入使得 Master 不再是单点故障

章节	**Ch05 - 分布式列式数据库 HBase（二）**
题目 1)	**HBase 容错性**
	Master 容错：Zookeeper 重新选择一个新的 Master 无 Master 过程中，数据读取仍照常进行； 无 master 过程中，region 切分、负载均衡等无法进行； RegionServer 容错：定时向 Zookeeper 汇报心跳，如果一旦时间内未出现心跳，Master 将该 RegionServer 上的 Region 重新分配到其他 RegionServer 上，失效服务器上“预写”日志由主服务器 进行分割并派送给新的 RegionServer Zookeeper 容错：Zookeeper 是一个可靠地服务，一般配置 3 或 5 个 Zookeeper 实例
题目 2)	**Zookeeper 作用**
	1. 配置管理 2. 名字服务 3. 分布式锁 4. 集群管理
题目 3)	**Zookeeper 的特点**
	1 最终一致性：为客户端展示同一视图，这是 zookeeper 最重要的功能。 2 可靠性：如果消息被到一台服务器接受，那么它将被所有的服务器接受。 3 实时性：Zookeeper 不能保证两个客户端能同时得到刚更新的数据，如果需要最新数据，应该在读数据之前调用 sync()接口。 4 等待无关（wait-free）：慢的或者失效的 client 不干预快速的 client 的请求。 5 原子性：更新只能成功或者失败，没有中间状态。 6 顺序性：所有 Server，同一消息发布顺序一致。
题目 4)	**例举常用的 Hbase shell**
	1)查看有哪些表 list 2)创建表 create 'xt','xcf' 3)表的描述 describe 'xt' 4)向指定的列族中插入数据 put 'xt','1','xcf:col_name1','col_value1' 5)get 查询数据 get 'xt','1'

	6)扫描全部数据 scan 'xt'
题目 5)	**HBase 接受数据，如果短时间导入数量过多的话就会被锁，该怎么办？集群数 16 台，高可用性的环境**
	通过调用 Htable.setAutoFlush(false)方法可以将 htable 写客户端的自动 flush 关闭,这样可以批量写入到数据到 hbase。而不是有一条 put 就执行一次更新，只有当 put 填满客户端写缓存时，才实际向 Hbase 服务端发起请求。默认情况下 auto flush 是开启的。