大数据系统开发考试内容

1. **大数据基本概念**
2. 大数据特征

4V:

1. 体量：非结构化数据的超大规模增长占总数据量的80~90%
2. 多样性
3. 价值密度
4. 速度
5. 大数据面临的挑战

数据形式多样性，大规模，处理速度

实现低密度、高价值

1. 关系数据库与NoSQL数据库的差异

一、关系型数据库与NoSql数据库的比较

1、并发支持

关系型数据库：通过事务和锁来支持并发，高并发情况下，执行效率较低。

NoSql：打破了传统关系型数据库范式的约束和事务一致性，因此并发性能高。

2、存储与查询

关系型数据库：利用关系表方式存储数据，通过SQL查询

NoSql：分布式数据库，支持海量数据，多为键值方式存储数据，还有表格/列、文档、图像、对象和多值等存储方式；查询方式也多种多样，查询效率高。

3、扩展方式

关系型数据库：主要通过提升硬件配置等向上扩展方式来提升性能（垂直扩展）。

NoSql：增加数据库节点向外扩展（水平扩展）。

4、索引方式

关系型数据库：B树、哈希等

NoSql：键值索引

5、应用领域

关系型数据库：通用领域

NoSql：特定应用领域，如要求海量数据，高并发，高性能，伸缩性强，可容忍最终一致性的系统。

二、NoSql数据库的类型

按存储类型分有

1、键值对

代表是Redis

2、表格/列存储

关系型数据为按行存储。按列存储特别适合一个字段非常多的表，每次查询我们只从其中挑一些字段的情况。为啥呢，因为按行存储的话，尽管我们只获取小部分字段，但系统却不得不先将整行读出来，然后再挑出我们要的字段返回。返回内容的确不多，但背后消耗的性能却不少。因为数据是存放在页里面的，而页的尺寸大小固定，比如通常就是8K、16K等等，如果一行的字段很多，那么一页里包含的记录数就少，为了替我们找到指定的数据，返回的页就要很多，如果这些页没有在内存中，那么还要从磁盘里调度，呵呵，这时硬盘灯就在狂闪，怎么能不慢！

按列存储的话，这种情况就舒服多了。

什么表要用列存储呢？比如说，这个表用来存储一些什么环保水质、水文气象的指标，我去，那些个指标，多到你怀疑人生。

国内南大通用的GBase就是列存储的NOSQL数据库，天津的好像。

3、文档

一条记录就是一个文档。那么这个说是键值对存储方式也无不可。

代表是芒果DB（MongoDb）。它最像关系型数据库了。用javascript 来查询，酷吧。

4、图

代表Neo4j。

常用于知识图谱，人工智能范畴吧。

5、对象和多值存储

未知

三、NoSql的缺点

1、成熟度不够，大量关键特性有待实现

2、开源数据库产品的支持力度有限

3、数据挖掘与BI支持不足，现有许多系统无法直接使用NoSql

4、NoSql属新兴事物，专家和人才较少

1. 大数据关键技术

数据采集:ETL工具

数据存取:关系数据库;NoSQL;SQL等

基础架构支持:云存储;分布式文件系统等

计算结果展现:云计算;标签云;关系图等

1. 大数据处理流程

数据采集->数据存储->数据处理->数据挖掘与分析->数据可视化与决策

1. 大数据系统架构

数据感知层、数据文件系统层(HDFS)、数据仓库与计算层(Hadoop、Spark)、接口层(Api)、分析工具层、模型与可视化层、业务应用层

1. **Hadoop**
2. 什么是Hadoop

Hadoop是一个能够对大量数据进行分布式处理的软件框架

Hadoop是一个能够让用户轻松架构和使用的分布式计算平台。用户可以轻松地在Hadoop上开发和运行处理海量数据的应用程序

1. Hadoop的核心架构

Hadoop的核心，就是HDFS和MapReduce。HDFS为海量数据提供了存储，而MapReduce为海量数据提供了计算框架。

HBase MapReduce

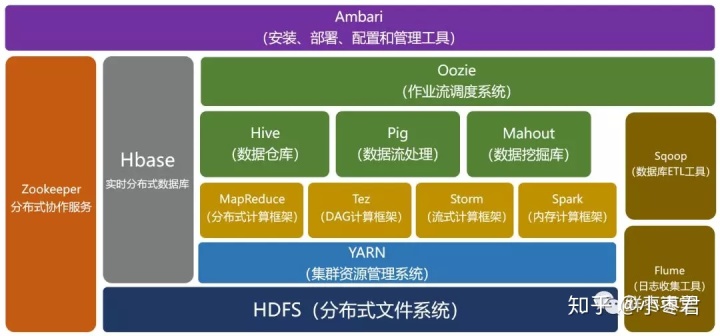
（YARN）

HDFS

Hadoop

2.0 版本新增加了YARN（资源管理框架）

1. Hadoop生态圈



1. Hadoop的优点

高可靠性。Hadoop按位存储和处理数据的能力值得人们信赖 。

高扩展性。Hadoop是在可用的计算机集簇间分配数据并完成计算任务的，这些集簇可以方便地扩展到数以千计的节点中 。

高效性。Hadoop能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此处理速度非常快 。

高容错性。Hadoop能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配 。

低成本。与一体机、商用数据仓库以及QlikView、Yonghong Z-Suite等数据集市相比，Hadoop是开源的，项目的软件成本因此会大大降低 。

1. Hadoop处理大数据的优势
2. **什么是HDFS**
3. HDFS的目标

硬件故障

流式数据访问

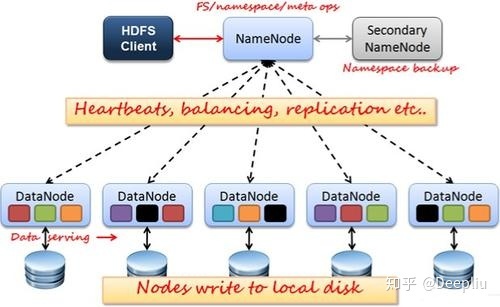
大数据集

简单一致模型

移动计算比移动数据更经济

异构软硬件平台间的可移植性

1. HDFS体系结构



1. HDFS块副本

每一个文件可以配置块副本数量，默认是3。

副本的作用是防止因DataNode故障而导致数据丢失，除此之外块副本还可以提高块可读取的节点。

每个文件可以在写入时指定这个文件块的副本数量，也可以在未来修改某个文件的块副本数量，文件块的副本数量配置作为块元数据的一部分保存在NameNode中。

1. HDFS元数据

目录树结构。

文件与block之间的关系

block与DataNode的关系

1. HDFS读写流程
2. HDFS访问方式
3. **什么是HBase**
4. HBase的特点

易扩展

海量存储

高可靠性

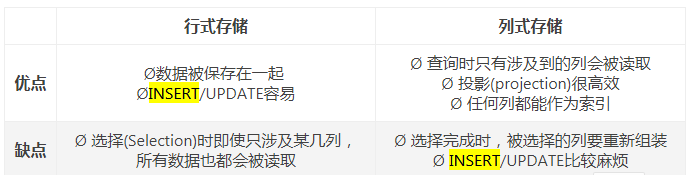
稀疏性

1. HBase处理大数据
2. HBase体系结构

HMaster RegionSever ZooKeeper

1. HBase数据模型

RowKey ColumnFamily

1. 行式存储与列式存储的区别
2. HBase日志

WAL 先写日志，再写内存

1. HBase读写流程
2. **MapReduce**
3. MapReduce是什么？
4. MapReduce体系架构与处理流程
5. MapReduce分区
6. MapReduce如何在分布环境中处理词频统计、倒排索引
7. **实验部分**
8. Hadoop本地、伪分布与全分布部署模式
9. HDFS上传文件（原理与分析命令）

Hadoop dfs -put /uset /s

1. 配置文件及具体的使用方法
2. MapReduce的主要接口

Mapper、Reducer

1. MapReduce实现倒排索引（原理及代码分析）
2. MapReduce如何实现移动计算？
3. HBase写入记录的过程（原理与分析代码）
4. Hadoop进行开发搜索引擎的优势

**注：不要求写代码、安装、部署等相应的命令，给出代码、命令能够识别与分析，例如，能够识别Mapper接口方法、该接口的作用及使用方法（例如，分析代码）。**