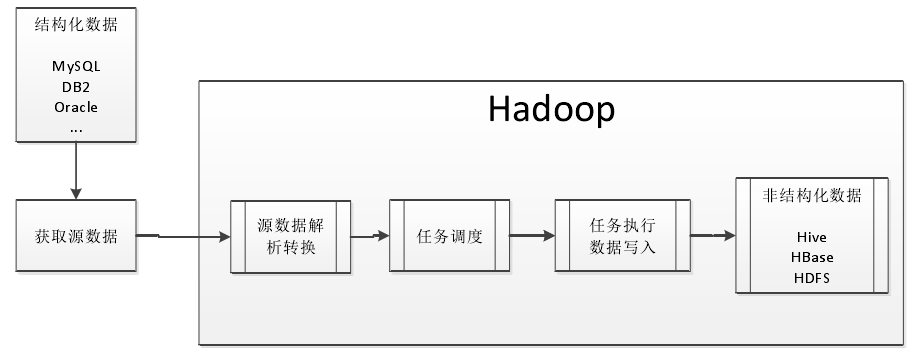
3200/28937 4237/32233 21565

# 《基于Hadoop平台的大数据迁移与查询方法研究及应用》

1. 当代社会，伴随着信息技术的快速发展，全球时时刻刻都在产生数据，面对如此大量的数据，各领域、各企业对“大数据”的处理需求不断的提高，而如何科学的管理这些数据、如何快速查询或者更新数据信息成为一个亟待解决的问题。
2. 关系型数据库以交互式访问方式处理数据集、处理数据大小一般为GB级，其结构为静态模式、非线性线性横向扩展，更适合持续更新的数据集。而随着数据量越来越大、数据库中的数据类型不断的变化、各种非结构化数据逐渐成为了需要存储和处理的大数据的重要组成部分，仅仅使用传统数据库或数据分析软件(如SAS\SPAA)已经不能很好地解决大数据带来的问题，无法满足经济有效的存储、分析和访问数据。而在众多解决大数据量和处理效率问题上，Apache基金会设计实现的Hadoop在分布式计算和大量数据处理方面脱颖而出。
3. Hadoop平台以用于存储数据的文件系统HDFS和处理大数据集的并行计算框架MapReduce两部分为底层基础，此外，还提出了一些高层的抽象工具，如按列存储的数据仓库Hive，按列存储的数据库HBase，用以检索大数据集的Pig，以及在DBMS与HDFS之间进行数据传输的工具Sqoop等。
4. Hadoop平台相对于关系数据库具有用向外扩展架构代替向上扩展、处理类型为key/value的形式而非传统表形式、用MapReduce框架编程实现SQL语言查询、处理状态可为离线批量处理四个方面的特点。
5. Hadoop并行处理框架核心HDFS和MapReduce在集群上分别实现了大数据的分布式存储和并行处理。Hadoop通过HDFS为用户提供具有高容错性和高伸缩性的海量数据的分布式存储，并通过MapReduce为用户提供逻辑简单、底层透明的并行处理框架。
6. Hadoop是近年来IT界研究热点之一，大数据存储及处理是Hadoop的技术研究的主要两个方面。
7. DBInputFormat为Hadoop新版本提供的数据迁移工具，可以让Hadoop与多种关系数据库之间进行数据导入导出。它链接数据库的方式为主流的JDBC方式，让程序员更为方便的进行数据链接操作。DBInputFormat实现数据交换的原理。
8. Sqoop是一种使关系数据库和Hadoop之间数据能够相互转换的命令行应用程序。
9. 数据迁移是数据在存储类型、格式或不同机器（computer）之间相互转换的过程。
10. 通常情况下，执行数据迁移编程方式都是自动化的过程，这样既可避免重复而单一的劳动，又可以提高生产效率。
11. 程序化的数据迁移可能涉及多个阶段，且至少会包括两点：一是数据抽取，这是指数据从旧系统中读取的过程，二是数据装载，是指数据被写入到新系统中的过程。
12. 数据迁移需要保证迁移成本低于原系统的维护成本，还需要降低大量数据因结构变化导致的数据错误或丢失的风险；数据迁移必须保证既不影响原系统的正常运行又能够在迁移之后提高系统运行效率。
13. 本文综合考虑这些因素，对数据迁移工具进行对比，分析其迁移的具体实现；然后结合水路规费征稽系统（本次研究的课题来源），设计一个有针对性的数据迁移方案。
14. 本文主要讨论的理论体系为Hadoop的核心，基于Hadoop平台的数据仓库Hive、数据库HBase，可用于数据库和HDFS之间高效数据交换的工具。
15. 在此基础上，需要进一步对Hadoop数据交换原理、Hadoop作业调度器及MapReduce作业并行工作机制进行研究，进而将原系统关系数据库中的数据迁移到Hadoop平台中。
16. 本文首先从现有的数据管理系统中获取源数据，然后进行数据迁移，重点研究Hadoop平台下结构化数据迁移方案的设计与实现。由于MapReduce可以实现更优的并发性、体现优化数据转换的能力，因此，针对结构化迁移提出基于MapReduce架构的迁移方案。

# 《Hadoop 架构下数据挖掘与数据迁移系统的设计与实现》

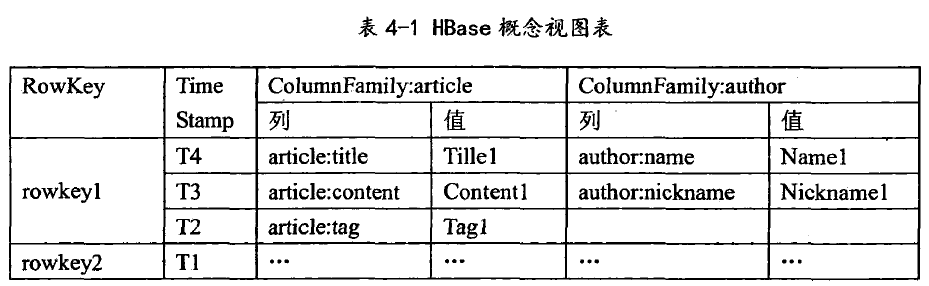
1. 在大数据平台中，需要存储所有的历史数据，这些历史交易记录可能高达几十亿条，存储大小可能高达几百TB；在业务上既要支持数据挖掘，也要支持用户对历史交易记录的查询，传统SQL数据库的存储容量和处理能力已经不能满足大数据平台的上述超大规模数据的处理需求，所以必须采用NoSQL数据库来存储上述数据。
2. HBase是Google的Bigtable分布式数据库的开源实现，HBase是一个稀疏的、分布式的、持久化存储的多维度排列的映射。
3. HBase与关系型数据库比较：
4. 查询
5. 读写特性
6. 扩容
7. 负载均衡
8. 数据版本
9. HDFS是Hadoop架构下的一个分布式文件系统，其采用主从结构体系。HDFS的集群是由一个或多个命名节点（Name Node）和多个数据节点（Data Node）组成。其中Name Node用来管理系统的命令空间以及文件的元数据，此外Name Node还负责监听Data Node和客户端的请求，执行对应操作，比如打开、关闭、重命名文件或者目录等，并且负责监控Data Node的状态。Data Node要定期向Name Node发送心跳（Heartbeat）信息，如果Data Node一段时间内没有反应，Name Node则将此Data Node标记为“死”节点，并将处理由此Data Node节点的离线所导致的副本数量不足而产生的数据复制操作。Data Node用来存储文件的数据，处理文件客户端的读写请求：创建、删除、复制数据块等。
10. HDFS数据特性：
    1. 数据可用性
    2. 数据完整性
    3. 数据一致性
11. 分层存储系统由在线业务存储平台和大数据存储平台组成，其中，在线业务平台为企业的业务系统提供了在线数据的保存和处理服务，保存了企业近期一段时间的生产和运营数据；大数据平台保存了企业业务平台所有的历史数据，是在线业务平台的备份和归档系统。在设计和实现的过程中，分层存储系统需要解决两个主要的问题：对于大数据平台中的海量数据，如何向用户提供可靠且迅捷的数据查询及分析服务；对于在线业务平台中的实时数据，如何高效且无损失地从在线业务平台迁移到大数据平台。针对上述的问题，本文后续章节提出了针对大数据平台海量数据的数据挖掘分析和在线业务平台向大数据平台迁移解决方案。
12. 在线业务平台上的数据只能保存一定时期的数据，随着数据量的增多，系统数据的维护成本较大，当在线业务平台数据量达到一定程度时，需要将数据迁移到大数据平台上做备份归档。
13. 规定多长时间进行一次数据迁移，在HBase中需要的是进行增量插入。
14. 在数据迁移的过程中可能会造成数据的损失，因此对于那些较为重要的数据，也可以采用将关系型数据库文件备份为数据库文件保存在大数据平台，需要恢复的时候再重启导入到关系型数据库中，提供数据查询。
15. 通过对电子支付中消费者和商户的消费记录进行数据挖掘，将消费者和商户分类，根据消费者的消费金额和消费频率来推销不同的产品，根据商户不同的经营情况决定是否提供贷款等。
16. 采用分布式的开源框架Hadoop，对保存在HBase数据库中的数据，通过Map Reduce进行大规模的数据分析，完成挖掘功能。数据挖掘即从数据库中大量的数据揭示出隐含的、潜在的信息的过程。
17. 通过自动化的分析数据，做出相关的归纳性的推理，帮助决策者做出相关策略。在系统需求的情况下，数据挖掘模块主要对消费者进行分类和商户进行分类。
18. 在数据迁移的过程中，系统设计需要注意两个问题：数据完整性和迁移性能。
19. 据统计，一般主流的电子支付平台每天会产生近4TB的数据，如此量级数据迁移需在数小时之内完成，迁移系统的调度设计和并发性是性能的主要决定因素。
20. 系统需要定期将在线业务平台中的结构化数据迁移到NoSQL数据库中，以便进行后续的查询和数据挖掘工作。结构化数据迁移的过程就是将在线业务平台的数据库记录导出，并转换成目标格式，保存在 NoSQL数据库中。
21. 为了获得更好的迁移性能，也可将在线业务平台的数据库记录保存成数据库文件，存放在HDFS上。



1. 传统的Hadoop自带了三种作业调度器：FIFO调度、公平调度和计算能力调度器。如果不进行特殊设置，则Hadoop采用FIFO调度。Hadoop调度器假设：
   1. 所有的节点处理工作的能力是一样的
   2. 每个任务在处理过程中速率是保持恒定不变的
   3. 在相同类别中的任务，如Map或者Reduce，需要的工作量是相同的。
   4. 调度的过程中不考虑Tasktracker的CPU利用率、磁盘IO速率等因素的影响。
2. 但是现实的集群系统往往是异构的，每个节点的性能都不同，如果计算密集型或者数据密集型的作业被调度到CPU较慢或者磁盘较慢的节点，作业的运行时间会很长。这样就影响了整体数据迁移任务执行的速度。

# 《商品比价系统中大数据迁移及数据转换技术研究》

1. 原有的数据存储工具中存储了大量的历史数据，并且为了系统运营和设计上的方便，这些历史数据并不能直接丢弃，而焉需要迁移到新的数据库中。
2. 数据迁移需求：
   1. 功能性需求：①数据库连接；②表模式的映射与迁移；③数据类型映射；④数据加载。
   2. 非功能性需求：①正确性；②完整性；③高效性
3. 数据迁移过程的实现大致可以分为三个阶段：迁移前的准备工作、数据迁移过程的实施和数据迁移后的正确性校验。在数据迁移前的准备阶段，需要在新数据库中创建数据字典；分析新数据库与旧数据库中数据结构的差异，从而建立新旧数据库表之间的转换映射关系；创建新旧数据库之间数据类型的映射，以用来处理在新数据库中不支持的数据类型；部署已有的工具或者进行工具的开发。
4. 数据的迁移流程：
   1. 建立数据库，并获取源数据库和目标数据库连接信息。
   2. 根据关系型数据库中表的连接关系，定义关系型数据库中的数据表与HBase数据库的映射方式，并记录在映射关系文件中。
   3. 根据映射关系文件中的映射方式，在HBase数据库中建立相应的数据存储表。
   4. 将关系型数据库中的数据按照映射关系迁移至建立的数据表中。
5. 在中进行数据的查询时，根据所请求的数据不同，可能出现以下三种情况：全表扫描、区间扫描以及行键扫描。
6. 在设计数据表结构时，需要尽量使用中相对性能较优的行键扫描或区间扫描方式，从而最大程度地减少数据查询时间。为了提高在实际应用中的查询效率，应当尽量将需要查询的维度信息存储在行键中，因为使用行键蹄选数据的效率最高。
7. RDBMS与HBase之间的映射关系包括表映射和列映射。
8. 数据库模型是对数据库逻辑结构与语义的描述，用来表示实体类型及实体间联系。
9. HBase中的数据都是字节流形式，没有类型，任意类型数据入库时都被转化为字节流。每一张表都至少包含一个列簇集合（不多于3个列族），列簇是固定不能改变的，只有重新改变表结构。



1. HBase与传统关系型数据库存在的一个很大区别在于，在概念视图上如果此列是空白旳，即不存在数值，那么这样的列实际是不会被存储在数据表中的。如果我们请求这些空白的单元格，就会返回null值。在存储数据的时候数据会按照时间戳进行排序。所以如果查询数据时不提供时间戳，就会返回最近版本的数据。
2. HBase和RDBMS差异：
   1. 数据类型：HBase中存储的数据最终都得转化为字节数组，其所操作的数据都只是简单的字符串类型。而关系型数据库中可以存储各种所支持的复杂的数据类型。
   2. 数据操作：HBase数据库摒弃了关系型数据库表之间的关联关系，HBase中很难实现如关系型数据那样的连接查询操作，也没有函数操作，更多的是进行插入数据，批量查询，删除等操作。
   3. 存储模式：HBase是基于列的存储模式；RDBMS是基于行模式的表格存储结构。
   4. 事务性：HBase不具有事务性，而RDBMS是具有事务性的数据库。
   5. 数据存储多版本机制：HBase提供了时间戳来存储多个版本的数据，而关系型数据库则没有。
   6. 数据维护：HBase数据表中能够维护多个版本，而传统关系型数据库的更新操作是直接修改了数据本身。
   7. 可伸缩性：HBase具有高伸缩性，可以方便地增减集群节点，并保证高容错性，传统的数据库扩展比较困难，需要重新设计进行分库分表操作。
3. 由于HBase中数据表之间不能像关系型数据库中数据表之间通过设立外键来实现关联性，因此为了满足能够在迁移后的HBase数据库上做对应的关联查询，就需要根据实际表之间的关系作相应的变换。