文章编号：1005-8451（2015）08-0020-04

# 互联网历史订单集群的设计与实现

梅巧玲，刘文韬，杨立鹏，王　拓
（中国铁道科学研究院　电子计算技术研究所，北京　100081）

**摘　要：**针对历史订单的查询流程，将内存数据库技术和大数据Hadoop技术无缝结合，实现旅客快速地查询到更多的历史订单信息。不仅为旅客查询订单提供便利，也为分布式大数据技术在客票系统中的广泛推广打下坚实的基础。

**关键词：**Hadoop；订单查询；历史数据；消息；分布式
**中图分类号：**U293.22：TP39　**文献标识码：**A

## Internet history ordering clusters

MEI Qiaoling, LIU Wentao, YANG Lipeng, WANG Tuo
( Institute of Computing Technologies, China Academy of Railway Sciences, Beijing　100081, China )

**Abstract:** According to the query processing of history orders, the paper combined the memory database technology with large data Hadoop technology, implemented quick query more history orders for passengers, provided convenience for passengers to order query, laid a solid foundation for distributed large data technology in the Railway Ticketing and Reservation System.

**Key words:** Hadoop; order-query; history data; message; distributed

从2011年6月互联网售票系统上线以来，注册用户已达1.2亿多，十一国庆节期间网站最高日售票量达636.8万张。截止到目前为止，互联网用户的订单数据已达几十亿条左右，互联网售票系统面临前所未有的并发访问压力和海量数据存储压力。相比于国内和国外主流的电商网站，现有的铁路互联网售票系统受到架设在铁路既有客票系统之上等因素的影响，铁路互联网售票系统与这些电商网站有着明显差别。互联网售票系统除了售票主要功能之外，还有其他业务，包括退票、改签、订单跟踪、退款详情、快递详情等，要办理或者查询这些业务，需要到“我的订单”中进行订单查询才能看到，订单查询是这些业务的入口。铁路客票预售期由原来的20天调整为60天后，未出行订单的数量根据历史数据的分析预估增加1.5倍，这样的数据量对现有的分布式内存数据库是一种挑战。基于上面因素的考虑，引入了分布式Hadoop数据库技术，将12306中“我的订单”查询功能分为“未出行订单”和“历史订单”。旅客频繁访问的未出行订单数据存放在内存数据库中，历史订单数据存放在分布式大数据Hadoop数据库中，将数据进行分布式存储，实现“表面集中、内在分散”的现象，达到并行计算快速查询的目的。

收稿日期：2014-12-11
作者简介：梅巧玲 副研究员；刘文韬，副研究员。

## 1　关键技术

### 1.1　大数据Hadoop技术

Hadoop是一个对大量数据进行分布式处理的软件框架，它包括两个核心模块：HDFS和Map-Reduce。HDFS是Hadoop中的分布式文件系统，存储Hadoop生态系统中的各种文件，实现了对文件的增删改查等操作。HDFS有高容错性的特点，并且适合部署在硬件上；而且它能提供高吞吐量应用程序的访问数据，适合那些有着超大数据集的应用程序。MapReduce是一种简化的用于并行处理大数据集的分布式编程模式，让程序自动分布到一个由普通机器组成的超大集群上并发执行。

### 1.2　异构数据同步中间件

异构数据同步中间件CTMSX主要用于异构环境的数据同步。通过读取并转发关系型数据库复制服务主点到从点数据库的消息，作为数据桥模拟复制主

点数据库的响应消息序列到复制从点，以确保实时数据的完整性和一致性，完成数据库主、从点之间的数据同步，同时通过数据适配器、解析器将消息和数据实时同步转发到消息服务器 MQServer 中，由消费程序消费 MQServer 的队列，将数据操作动作同步到 GemFire 内存数据库和 Hadoop 集群中，最终实现 Sybase 数据库与 GemFire 内存数据库和 Hadoop 集群的异构数据同步功能。

### 1.3 MQServer消息机制

RabbitMQ 是由 LShift 提供的一个 AMQP（Advanced Message Queuing Protocol）的开源消息队列系统，由以高性能、健壮以及可伸缩性出名的 Erlang 写成。一端往消息队列中不断写入消息，而另一端则可以读取或者订阅队列中的消息。绑定协议，包括消息交换机（Exchange）和路由关键字（RoutingKey），定义了 Exchange 和消息队列实体 Queue 之间的关联，提供路由规则。Exchange 指定消息的路由规则，由异构系统双方（或多方）共同约定。

### 1.4 数据库复制技术

复制服务器（Replication Server）维护多个数据库中的复制数据，同时确保数据的完整性和一致性。使用复制系统中的数据库的客户机可以在本地访问数据，减少网络负载和中央计算机系统的复制。Replication Server 采用一种基本的“发布 - 预订”模式来实现跨网络的数据复制。用户“发布”主点数据库中的可用数据，其它用户“预订”这些数据以便将其发送到复制点数据库。

### 1.5 分布式内存数据库

内存数据库是一个较新的研究领域，全部数据常驻内存，在 CPU 进行运算时，存取数据只需与内存进行 I/O 操作，效率非常高。采用分布式内存数据库集群后，根据用户指定的数据存储规则，不同范围的记录由不同的物理节点进行存储，并行查询和并行处理也会随之分布到不同的物理节点上，最终由控制节点完成汇总。内存数据库集群支持动态水平扩展节点，由于本身具有备份机制，即使单点故障，也不会影响到整个集群。同时还支持数据持久化，如果整个集群发生异常时，数据也不会丢失，可以从硬盘的持久的文件中恢复。

## 2 历史订单集群架构设计及实现

### 2.1 历史订单集群设计思路

历史订单查询，指的是当天之前（不包括当天）的订单信息，旅客除了由于铁路原因可以进行原退操作外，更多的是查询功能，因此可以将订单相关数据划分为两部分数据：内存数据和磁盘数据，将访问频繁的数据量少的数据放到内存中，为了保证高可靠性，可在磁盘上做持久化配置；将访问较少的数据量多的数据保存到磁盘上，将所需数据加载到内存后，进行相关逻辑判断，这样既能满足大数据的存储，也能加快查询效率。

在历史订单集群框架中，内存数据库集群在初始启动时只有少量的基础数据，在关系型数据库和 Hadoop 集群中各有一份全量数据。当用户登录互联网售票系统后，在“我的订单”功能中，点击历史订单查询时，分级存储系统自动将这些被立即访问的数据从二级存储设备 Hadoop 集群加载到一级存储设备分布式内存数据库中。上述数据的搬迁过程对于用户来说是完全透明的。

### 2.2 历史订单集群系统实现

历史订单集群框架主要包括 3 个模块：数据产生模块（传统 Sybase 关系型数据库）、订单相关数据同步模块（包括复制服务器系统、CTMSX 和 MQServer）、存储运算模块（包括分布式内存数据库和 Hadoop 集群），系统架构如图 1 所示。

#### 2.2.1 数据产生模块

第 1 个模块是数据产生模块，订单相关的所有数据是在关系型数据库中产生的，关系型数据库是通过用户名来区分的一个数据库集群组成，每个用户产生的订单数据存放在各个节点上，从上到下纵列的下端是上端的备份节点。

#### 2.2.2 数据同步模块

第 2 个模块是数据同步模块，复制服务器负责侦听数据库的数据变化，将捕获到的相关订单数据变化发送给异构数据同步中间件 CTMXS，CTMSX 再将变化的数据以消息的方式发给 MQServer。MQserver 作为透明的消息转发中间模块，消费端接收到变化的

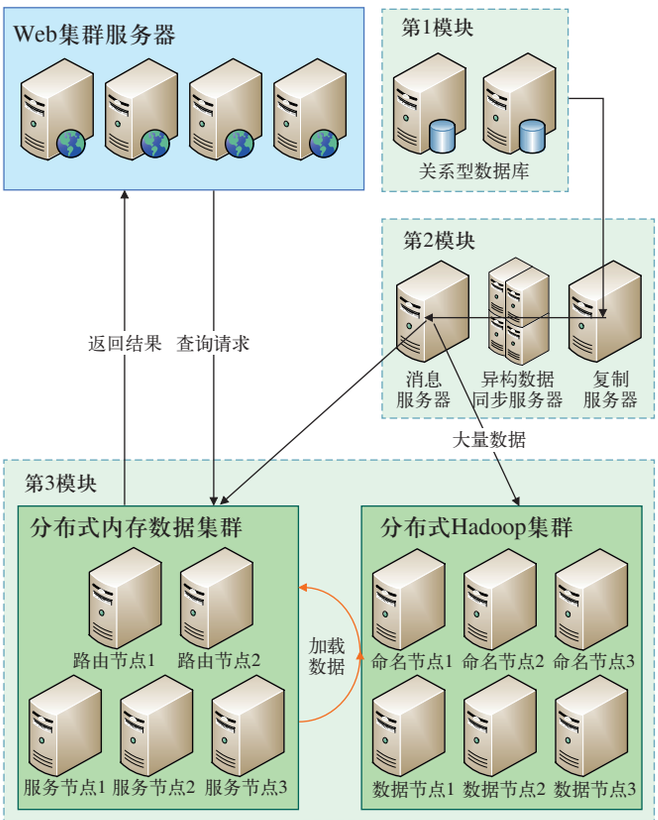


图1 历史订单集群架构图

SQL 语句，对不同的 Update、Insert 和 Delete 语句进行不同的解析处理，写入内存集群和 Hadoop 集群中。

### 2.2.3 存储运算模块

第 3 个模块是存储运算模块，它是分布式历史订单集群的核心模块，其具有数据存储和数据运算的两大重要功能，由分布式内存数据集群和分布式 Hadoop 集群组成，内存数据集群负责数据运算和存储，Hadoop 集群负责数据存储。内存数据库集群由分布在不同物理服务器上的多个节点组成，每个节点内含有若干个数据区域，每个数据区域内含有若干个数据单元，数据按照一致性 Hash 理论分散存储在不同的数据单元中。从物理逻辑上看，一个完整的数据存储容器包含了 3 个层次：多个内存数据库 Server 节点组成了整体的存储运算模块，多个数据存储区域 Region 组成了一个数据 Server 节点，多个数据存储单元 bucket 组成了一个数据存储区域 Region。集群并行调度多个节点参与运算，每个节点只进行与自己内存单元中关联数据的运算，避免跨网络的读取大量数据，最终将所有参与运算节点的运算结果数据汇总，完成订单查询的计算。活跃数据的存储与对数据的处理均在物理内存中完成。

历史订单 Hadoop 集群由两个命名节点（NameNode）和多个数据节点（DataNode）组成。NameNode 的作用是管理元数据和文件块、管理命名空间、监听并处理请求和心跳检测。DataNode 的作用是读写数据块、向 NameNode 汇报活跃状态和执行数据流水线复制。订单数据被切分成文件块的形式分散存储在分布式文件系统（HDFS，Hadoop Distributed File System）的不同 DataNode 节点上。在 HDFS 中，订单数据以注册用户名作为划分列导向存储机制的数据库（HBase）区域的原则，将数据分布在不同 HDFS DataNode 物理节点上，在读写时将自动进行校验，NameNode 上保存了每份数据的版本信息，发现不同数据节点的同一个数据块的版本不一致，就会触发恢复流程同步所有的节点数据，一旦发现数据校验错误将重新进行复制，最大程度保障了数据的准确性和一致性。尽最大程度提高查询性能，将关系型数据库中的唯一索引设置为 HBase 的 rowkey，客户端进行查询时，先通过 login_name 模糊匹配，再通过订单号等其他条件拼接 rowkey 来获取数据。Hive 数据仓库可以将结构化的数据文件映射到一张数据库表上，提供简单的 SQL 查询功能，可以将 SQL 语句转换为 MapReduce 任务运行。以 HBase 为基础，建立 Hive 到 HBase 的外部关联表，对历史订单数据进行更深层次的数据挖掘。

## 2.3 历史订单集群试验效果

互联网售票上线运营以来，高峰日售票量占客票交易总额的 68% 以上，累计销售客票约 18.1 亿张。按照高峰期的售票量统计，互联网每日产生 900 万张订单（包括自动返库和主动取消的订单），再加上与之相关联的其他数据，每日净增 3 900 万条数据。模拟并发用户 1 000 个对历史订单集群进行压力测试，订单查询请求平均值为 761 次 /s，平均响应时间 RT 为 492 ms，满足高峰时期系统的查询请求。

# 3 结束语

本文分析了铁路互联网售票系统面临的并发和存储双重压力，提出使用新的技术来尝试解决网站面

（下转 P26）