电子商务环境下基于Bigtable的海量数据存储系统设计及分析

王献美[1]，吴迪冲[2]，朱泽飞[3]，李仁旺[1]，张华[4]

（1. 浙江理工大学 先进制造研究所，浙江 杭州 310018；2. 浙江财经学院 工商管理学院，浙江 杭州 310018；
3. 杭州电子科技大学 机械学院，浙江 杭州 310018；4. 浙江理工大学 现代纺织装备技术教育部工程研究中心，浙江 杭州 310018）

摘 要：在分析互联网海量存储各种NOSQL数据库、电子商务环境下数据特点及现有数据模型的基础上，根据Bigtable的理论，提出了海量存储系统（HSS）存储数据模型，并对系统进行了设计，给出了系统架构及各个数据和控制流程。最后分析系统可能存在的瓶颈，给出测试性能数据，满足性能要求。

关键词：电子商务；海量数据存储；Bigtable

中图分类号：TP392 **文献标识码**：A **文章编号**：1000-436X(2011)9A-0233-05

Based on the Bigtable under the e-commerce environment huge storage system design and analysis

WANG Xian-mei[1], WU Di-chong[2], ZHU Ze-fei[3], LI Ren-wang[1], ZHANG Hua[4]

(1. Advanced Manufacturing Research Institute, Zhejiang Sci-Tech University, Hangzhou 310018, China;
2. College of Business Administration, Zhejiang University of Finance and Economics, Hangzhou 310018, China;
3. School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China; 4. Engineering Research Center of Modern Textile and Equipment Technology, Ministry of Education, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Based on the analysis of various NOSQL database for mass storage of Internet, characteristics of data on e-commerce environment and the data storage model, according to the theory of Bigtable, the mass storage system (HSS) storage data model was put forward, and the system was designed, the system structure and various data and control process were given. At last, the possible existence of system bottlenecks were analyzed, test performance data was given which shows the system design meets the performance requirements.

Key words: e-commerce; huge storage; Bigtable

1 引言

随着互联网的进一步普及，互联网应用的用户进一步增长，对现有的很多架构提出了挑战。这在存储系统方面尤为突出。首先，互联网应用需要存储的数据直接由用户生成，而不再是由应用提供者提供。随着用户量的增长，用户生成的数据量也随之增长，对系统能支撑的数据量提出了更高的要求，进而对底层的存储系统的可扩展性要求也越来越高。

随着移动互联网用户的增多，用户在线时间大大增长，这对服务的可用性提出了更高的要求，任何时刻服务的不可用，将影响大量的用户体验。

用户对服务的用户体验要求也越来越高，用户体验的一个重要因素便是响应时间，底层存储系统的响应时间非常关键。对数据存储有以下3个方面

收稿日期：2011-07-05

基金项目：浙江省自然科学基金资助项目（R6080403, Z6090572）；浙江省重点科技创新团队基金资助项目（2011R09015）
Foundation Items: The Natural Science Foundation of Zhejiang Province (R6080403, Z6090572); Zhejiang Province Key Science Technology and Innovation Team (2011R09015)

的要求：对数据库高并发读写的需求、对海量数据的高效率存储和访问的需求、对数据库的高可扩展性和高可用性的需求。

海量的网络数据需要功能强大的管理系统进行组织和存储，Google公司采用分布式文件系统存储海量的网络数据，从而提供强大的互联网搜索能力。GFS[1]是其用于底层数据存储的分布式文件系统，Bigtable[2]系统运行在GFS上实现对网络数据的结构化管理。Bigtable系统的运行环境是普通微机组成集群环境，却管理着Google公司的数PB级的数据。Zvents公司仿照Bigtable开发了Hypertable系统，此系统参考了前者的实现原理，但是仍然存在部分缺陷。Yahoo公司利用Bigtable的数据模型实现了Hbase系统，此系统基于Java语言，存储速度受到一定影响。Cassandra[3,4]是一个混合型的非关系的数据库，类似于Google的Bigtable。其主要功能比Dynomite[5]（分布式的Key-Value存储系统）更丰富，但支持度却不如文档存储MongoDB[6](介于关系数据库和非关系数据库之间的开源产品，是非关系数据库当中功能最丰富最像关系数据库的。)但是现有的系统数据模型很难迁移到NoSQL数据模型中。

本文在分析现有电子商务环境数据的特点和数据模型的基础上，兼顾了NoSQL的高扩展性和传统关系型数据库在数据结构表达上的便利性，提出海量存储系统HSS存储数据模型，给出了HSS的系统架构，并给出其数据查询、事务和动静态数据合并流程，最后分析系统可能存在瓶颈。

2 电子商务环境下数据分析

电子商务环境数据的特点，本文某大型电子商务网站收藏夹应用数据进行分析得出。其数据特性如表1所示，从业务产生数据特性来看，其特点是小文件。

表1 收藏夹应用数据特性

数据项	数据量	数据项	数据量
收藏信息	4.1G条	业务访问量	30M PV/日
收藏信息存储量	400GB	收藏信息TopQPS	1K
新增收藏信息/日	12M条	收藏itemTopQPS	8K
修改收藏信息/日	5M条	收藏信息(AV)	0.1K
收藏item	390M条	收藏item(AV)	0.3K
收藏item的存储量	100GB	修改收藏item/日	6M条
新增收藏item/日	0.5M条		

从表1中，可以得出电子商务环境下的数据特点对数据存储提出了高数据量、高并发度、高可用性、多表事务、范围查询和海量历史数据处理等要求。通过分析现有基于Bigtable[7,8]理论的开源NoSql实现，结合电子商务环境中的数据特点，得出HSS其功能如表2所示。

表2 海量数据存储系统功能描述

编号	功能描述
1	多表事务
2	单项、多项、范围查询
3	高可扩展性（扩展能力）
4	高可用性（故障恢复）
5	SSD支持
6	本地实时备份、异地准实时备份
7	增量dump

3 数据模型

HSS支持应用的概念（这类似于数据库中的schema），一个应用可以创建多张table，table中包含一系列的列。table之间支持基于列的关联（join）关系。应用在使用前需要先创建数据模型，并支持在运行过程中动态修改数据模型。

3.1 row key

每张表都需要指定一个row key，row key的最大长度可以在创建时指定，其内容为二进制字符串（binary string）。row key在一张表内需要确保唯一（这和关系型数据库中的主键类似），HSS在内部存储时，数据按照row key排序。一张表的数据会根据row key动态地切分，切分后的单位为tablet，tablet由startKey和endKey指定其负责的数据范围。

HSS的row key还支持split属性。该属性指定了row key内容的一个前缀，当数据在被动态切分时，系统会确保split后前缀内容相同的row key所对应的数据不会被切分至多个tablet中。

某电子商务网站的收藏夹应用中，用户可以收藏某个店铺，也可以收藏某件商品。用户的一条收藏记录可以由{userId, object type , object id}唯一确定，所以将其作为row key，分别表示收藏者的用户id，被收藏对象的类型（店铺或者商品）和被收藏对象的id，其中，userid为一个8byte的整型。为了确保同一个用户的收藏在动态切分时分布在同一个tablet中，则可以指定row key在8byte的位置split。

3.2　column

一张表可以创建多个列，HSS 中列支持的数据类型如表 3 所示。

表 3　　HSS 中列支持的数据类型

类型	说明
int	整数，范围为[2^{63}, 2^{63}]
varchar	字节流，可以指定其最大长度
datetime	表示自 1970-1-1 00:00:00 到现在的秒数
precise_datetime	表示自 1970-1-1 00:00:00 到现在的微秒数
create_time	和 precise_datetime 类似，该列的值由数据新增时系统自动生成，不支持修改
modify_time	和 precise_datetime 类似，该列的值由数据更新时系统自动生成，不支持修改

3.3　关联

为了简化对有关联的数据的查询操作，HSS 支持在创建表时指定表之间的关联关系。当客户端查询时，服务器端根据表之间的关联关系定义，自动将相应的数据合并，然后将合并后的数据结果返回给客户端。

比如下面的关联关系定义：

Join=rowkey[8,16]%collect_item_info:item_name$item_name,item_price$new_price

该关联关系定义了当前表和 collect_item_info 表之间的关联关系，关联关系由该表 rowkey 的 8~16byte 与 collect_item_info 的 rowkey 指定。当查询该表时，将使用 collect_item_info 中的 item_name 和 new_price 分别和当前表的 item_name 和 item_price 合并，合并后的结果返回给用户。合并操作不修改 HSS 中存储的数据，只影响返回给用户的结果。

4　海量数据存储系统设计

4.1　系统架构

该系统主要包括 5 个组件如图 1 所示：维护系统元数据的 rootserver（RS）、服务更新操作的 updateserver（US）、存储静态数据的 chunkserver（CS）、服务查询请求的 mergeserver（MS）和为应用提供服务接口的客户端。

该系统将表的数据动态切分为 tablet，tablet 的数据分为动态和静态 2 部分。静态的数据存放在 CS 上，所有对数据的修改都存储在 US 中。US 的修改定期同步到 CS，CS 将 US 的更新和本地的静态数据合并，生成合并后的新数据。

tablet 的信息由 RS 维护，客户端在初始化时会请求 RS，获取 US 的地址信息。客户端的更新请求（包括新增、修改和删除）都直接访问 US。查询请求时客户端根据相应的 rowkey 向 RS 查询其对应的 tablet 信息，RS 返回相应的 MS 地址，客户端根据返回的信息请求相应的 MS 获取数据。

MS 收到请求时，根据 rowkey 从 RS 获取相应的 tablet 信息，该信息中包括负责该 tablet 的 CS 列表，MS 请求相应的 CS，获取静态数据，然后根据返回的数据，请求 updateserver 获取相应的更新数据，将更新数据和静态的数据合并，将合并后的结果返回给客户端。

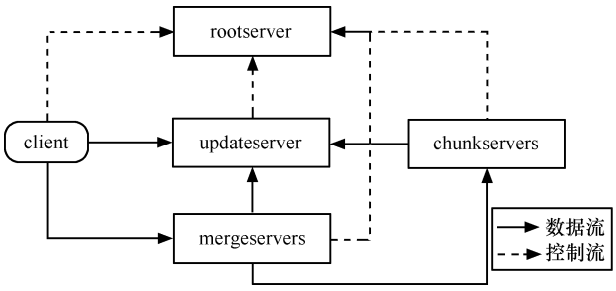


图 1　系统架构

系统各个服务组件描述如表 4 所示。

表 4　　系统各个服务组件

服务组件	功能	备份机制	特点
RS	schema/B+树节点[8]/机器管理	实时热备	高可用性
US	实时内存修改	实时热备	多表事务、高可用性
CS	SSTable 查询，每天数据合并	多副本	高扩展性、高可用性
MS	chunkserver+updateserver 查询结果合并、排序、分页		

4.2　查询流程

客户端查询流程如图 2 所示：

① client 向 MS 发送查询请求；

② MS 向 RS 发起定位请求；

③ RS 向 MS 返回定位信息；

④ MS 根据定位信息，向对用的 CS 发起查询请求；

⑤ CS 返回静态数据；

⑥ MS 向 US 发起查询请求；

⑦ US 向 MS 返回动态数据；

⑧ MS 将⑤和⑦返回的数据进行合并、排序，并返回结果给 client。

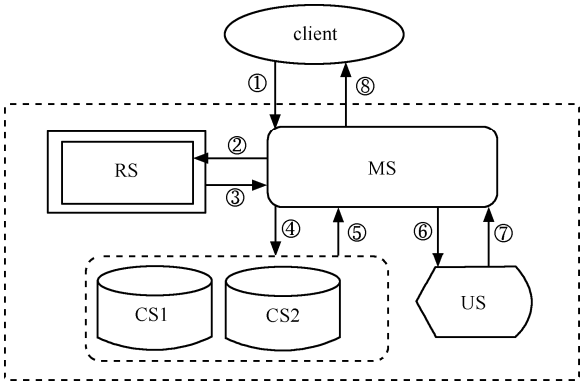


图2 数据查询流程

4.3 事务流程

US采用支持COW（copy on write）的B+树作为存储结构，新增和更新操作只对US进行操作，其主要流程如图3所示：client发起事务请求，updateserver执行并返回事务结果。

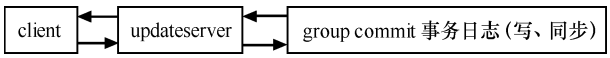


图3 数据新增流程

4.4 静动态数据合并流程

静动态数据合并流程如下：

①US开启新内存表并冻结已有内存表；

② 新修改写入新内存表；

③ CS：当前静态数据+冻结内存表→新的静态数据；

④ 切换到新的静态数据；

⑤ 查询请求：CS+新旧内存表。

合并数据的时机选择在系统访问较低的时刻。

5 系统分析

5.1 安全性分析

数据存储系统安全性主要从数据一致性、故障容灾分析，而本方案支持多表事务，记录事务日志，使其数据安全性，rootserver和updateserver实现实时热备、准实时异地热备、chunkserver多副本，确保各种异常情况下的数据安全。

5.2 性能分析

根据系统架构，对理论上可能存在性能瓶颈的项进行分析，并给出最终的性能数据。

1）内存：表5说明了更新数据全部纯内存操作，不存在瓶颈。

2）网络：表6中列举了各个执行操作不同的流量，采用多网卡或万兆网卡可以解决该性能瓶颈。

表5 内存分析

操作	数据数/条	单条数据大小/B	占用内存大小/GB
新增	1千万	1000	10
修改	1亿	100	10

表6 各个功能点网络流量

项目	流量/MB·s^{-1}
事务	10
查询	10
每天合并	30

3）磁盘：事务日志实现group commit[9]解决磁盘的IO瓶颈，系统在每次提交事务日志时，为了保证数据已经持久化到磁盘（durable），需要调用一次fsync来告知文件系统将可能在缓存中的数据刷新到磁盘。而fsync操作本身是非常消耗较多的IO资源，响应较慢：传统硬盘（10K转/分钟）大约每秒支撑150个fsync操作，SSD（Intel X25-M）大约每秒支撑1 200个fsync操作。所以，如果每次事务提交都单独做fsync操作，那么这里将是系统TPS的一个瓶颈。系统在当多个事务并发时，让多个都在等待fsync的事务一起合并为仅调用一次fsync操作。这样的一个简单优化将大大提高系统的吞吐量，这将带来5~6倍的性能提升。性能测试数据如表7所示。

表7 性能测试数据

项目	数量
机器数	6+2(chunkservers+updateservers)
性能数据	1 500TPS+2 000QPS
每天合并	耗时2h

6 结束语

本文在分析某大型电子商务环境下的数据特点和历史数据存储方式的基础上，基于Bigtable的理论，提出了HSS的数据模型，并给出了系统架构、各种数据操作流程，最后对系统的安全性及性能分析，经测试和线上运行情况表明，该系统设计的存储系管理功能完善、可靠性较强、可拓展性较好、具有较好的负载均衡能力。

参考文献：

[1] Google's Bigtable [EB/OL]. http://andrew.hitchcock.org/?post=214, 2005.

[2] CHANG F, DEAN J, GHEMAWAT S, et al. Bigtable: a distributed structured data storage system [A]. 7th OSDI[C]. 2006.305-314.

[3] CASSANDRA. Structured storage system over a p2p network by avinash lakshman, prashant malik and karthik ranganathan[EB/OL]. http://www.slideshare.net/jhammerb/data-presentations-cassandra-sigmod.

[4] Apache software foundation: the apache cassandra project[EB/OL]. http://cassandra.apache.org/,2010.

[5] Inc: amazon simple storage service (amazon S3). [EB/OL]. http://aws.amazon.com/s3/,2010.

[6] CHODOROW K, HOROWITZ E, MERRIMAN D, et al. MongoDB manual-database-commands-list of database commands[EB/OL]. http://www.mongodb.org/display/DOCS/List+of+Database+Commands,2010.

[7] 张晓清, 费江涛, 潘清. 分布式海量数据管理系统 Bigtable 主服务器设计[J]. 计算机工程与设计, 2010,31(5):1141-1143.
ZHANG X Q, FEI J T, PAN Q. Design of master in Bigtable system for massive data distributed management[J]. Computer Engineering and Design, 2010,31(5):1141-1143.

[8] GOOGLE WIKI. Overview of hypertable architecture[EB/OL]. http://code.google.com/p/hypertable/wiki/ArchitecturalOver-view,2008.

[9] AGUILERA M K, GOLAB W, SHAH M A. A practical scalable distributed B-tree[A]. Proceedings of the VLDB Endowment Vol 1, VLDB Endowment[C]. 2008.598-609.

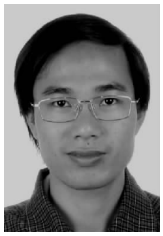
作者简介：

王献美（1979-），男，湖南邵东人，浙江理工大学博士生，主要研究方向为云制造和供应网格。

吴迪冲（1965-），女，浙江慈溪人，浙江财经学院教授，主要研究方向为先进制造技术。

朱泽飞（1963-），男，浙江天台人，博士，杭州电子科技大学教授，主要研究方向为机械电子工程。

李仁旺（1971-），男，湖南宁远人，博士，浙江理工大学教授、博士生导师，主要研究方向为数字化设计与制造。



张华（1980-），男，浙江建德人，浙江理工大学博士生，主要研究方向为物联网技术在纺织行业的应用和纺织装备关键技术。