面向 WEB2.0 的高效率数据系统—CSDS

CSDS – High Efficient Data System Based WEB2.0

董江明　唐劲维　张　凡　（搜狐公司 北京 100034）

摘　要： 目前 WEB2.0 产品的性能往往取决于数据系统，但是现有的关系型数据库很难同时满足低成本和高效能的需求。本文主要研究采用索引与存储分离的方式设计数据系统，用不同的服务器完成不同的工作，通过网络传送完成数据组装。开发上分别采用 B + tree 和 BerkeleyDB 为核心来实现索引部分和存储部分，通过前端逻辑封装将这两个物理独立部分联结为一个整体。以目前的原型产品测试结果表明，该系统可以提供比 MySQL 更大的存储容量和更快的索引速度，可以满足论坛类产品的需要。

关键词： 数据库 数据系统 B + tree 算法 BerkeleyDB 数据库

互联网软件开发，与传统企业软件开发有着巨大的差异，其特点表现在：

·低成本。通常互联网开发选用的操作系统、数据库及相关组件，以免费开源为首选，服务器方面也多采用廉价 PC 服务器。

·高访问量。门户级应用产品，每日的访问次数通常都在百万以上，甚至达到数千万次，高峰期每秒需要进行上千次处理。

目前互联网发展进入 WEB2.0 时代，对动态互动性产品的需求非常强烈，而数据库系统往往是瓶颈所在。MySQL 在存储量大变更频繁的情况下效率存在不足；Oracle、SQLserver 虽然可以通过集群提高效率，但是其价格昂贵代价较高。从国内情况看，Discuz、PhpWind 等论坛产品虽然应用广泛，但由于数据库的制约，很难使单一应用突破百万 PV/日的规模；而 SOHU、天涯、猫扑等提供千万 PV/日规模服务的专用论坛产品，则面临着成本过高的压力。

CSDS 系统（common separated data system）的设计思想是：以通用的索引和存储相互独立的数据系统来取代关系型数据库，从而提高整体性能。索引部分（简称 Cdex）采用 B + tree[1] 算法为基础进行开发，独占服务器运行；存储部分（简称 Cdata）基于开源项目 BekeleyDB[2] 开发，采用开源项目 MemCached[3] 做缓冲，与接口部分共同独占服务器；系统接口与逻辑部分采用 PHP 语言开发，向开发者提供操作类库和操作 API，部署于前端服务器上。其中，BekeleyDB 经过了公司其他研发团队改进，在大数据存储量情况下的效率得到增强。

CSDS 具有初步的数据库系统特征：支持数据结构、数据冗余度小易扩充、较高数据与程序独立性、共享程度高[4]。CSDS 是由定制索引与 key – value 对存储两个完全独立部分构成，由接口逻辑部分提供数据关联，只能支持基本的二维表。虽然接口采用类 SQL 语言格式，但是系统本身并不满足关系数据模型的要素[5]。因此，CSDS 的定位是简单高效的数据系统，而不是完备的关系型数据库。

1　需求分析与试验环境

整体需求以搜狐社区为需求评估原型：峰值访问量 6000 万 PV/日，总用户量 1100 万，日发帖量 40 万，年新增数据存储 200G。据此，对 CSDS 系统设计指标为：

（1）每组服务器（含索引 1 台存储 1 台）可存储历史数据 100G，支持访问量 500 万次/天。

（2）可通过多组服务器堆叠，实现对访问量 1 亿次/天的支持能力。

（3）数据系统除满足论坛的特定需求外，还应具有通用性，满足向其他产品扩展的潜在需求。

针对需求的 CSDS 系统设计分析：

(1) B + tree 效率分析:B + tree 是非常成熟的排序算法,SourceForge 的 B + tree 开源项目早在 2003 就完成了最终的稳定版本[3]。采用 1000 万数据构造的模型进行测试,每秒可执行 10 万次以上的操作。

(2) BerkeleyDB 与 MemCached 性能分析:BerkeleyDB 是开源产品,后被甲骨文公司购买开发出一系列高性能数据库相关产品[4]。根据公司研发中心数据报告显示:BerkeleyDB 单台服务器可存储 2G 条数据、总容量 200G,每秒可提供操作约 100 次。MemCached 是成熟的开源项目,实际测试表明可支持每秒 2000 次操作。用 BekeleyDB 改进版本和 MemCached 进行组合而构成的存储模块,在读取命中率 80%、更新率 5% 的情况下,理论每秒可操作数为:

(1) (5% /BDB 操作数 + 95% * (80% /MemCached 操作数 + 20% /BDB 操作数) = 345 次/秒。

(2) 逻辑层封装,主要消耗在 XML 数据解析。根据测试,CPU 消耗远小于设计要求。

(3) 由于索引和存储单独设计,多组服务器间既可以隔离操作也可以交叉操作,没有堆叠限制。

(4) 数据存储内容可以灵活组织,索引可以根据不同需求动态创建,满足通用性和扩展性需求。

MySQL 是互联网中广泛使用的一种关系数据库系统,在使用 MyISAM static 数据表格式时不能支持可变长的数据列;使用 MyISAM Dynamic 数据表格式时会造成碎片增多速度变慢[6]。与 MySQL 做对比,CSDS 系统性能稳定,不会随数据长度及总量变化而性能下降;同时,Cdex 与 Cdata 物理独立设计,并发性更强。

本文的实验环境在网通公司土城机房,服务器采用标准 PC Server(HP 360G5),操作系统为 RedhatEnterpriseAS4,服务器配置为 Xeon3GHz * 2/4GMEM/76GDisk。

2 CSDS 系统的设计与实现

2.1 系统框架设计

CSDS 系统的整体模块框架分为三个部分:索引部分(简称 Cdex)、存储部分(简称 Cdata)、逻辑与接口部分。整体项目中还有其他外部服务支持,不在本文讨论范围(图 1)。

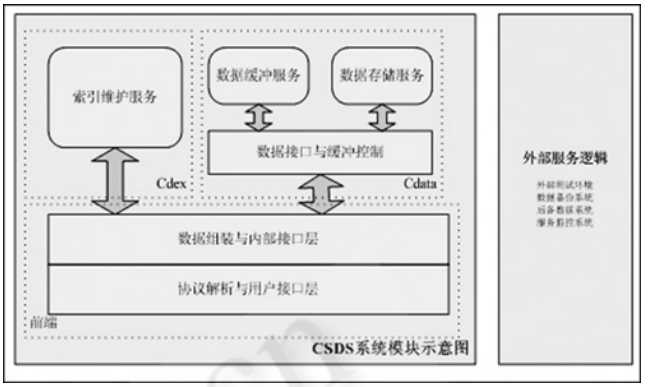

图 1　CSDS 系统模块示意图

CSDS 的部署上,Cdex 独占服务器,Cdata 独占服务器,逻辑与接口部分部署在前端服务器上。整体项目中还有其他外部服务器支持,不在本文讨论范围(图 2)。

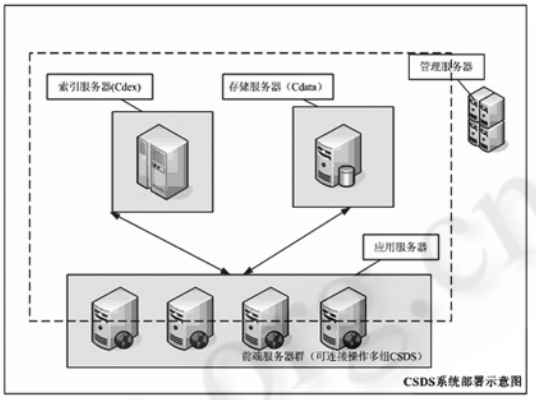

图 2　CSDS 系统部署示意图

2.2 索引模块(Cdex)设计

动态多级索引是一种常见的索引结构,通常使用 Btree 和 B + tree 数据结构实现[7]。B + tree 是 Btree 的变种,指向数据块的指针只存储在叶子节点中,可以得到级数较少而容量更高的索引。

Cdex 采用 C 语言开发,每个服务进程内部采用多线程模式并发运行,绑定一个 IP 端口提供符合 HTTP1.1 协议的对外服务。

性能指标为 存储 2000 万条数据,每秒执行 2000 次排序或查询操作。

2.2.1 Cdex 内部数据结构

索引模块内部数据结构分两部分:

数据表。单条数据长度确定,采用数组存放,保存需索引的字段和数据内容字段。

索引树。根据数据表中需要做索引的字段创建，采用 B + tree 结构，叶子节点指针指向数据表条目地址。图 3 描述了一个创建了 B – T、Z – T 两个索引的数据状态：

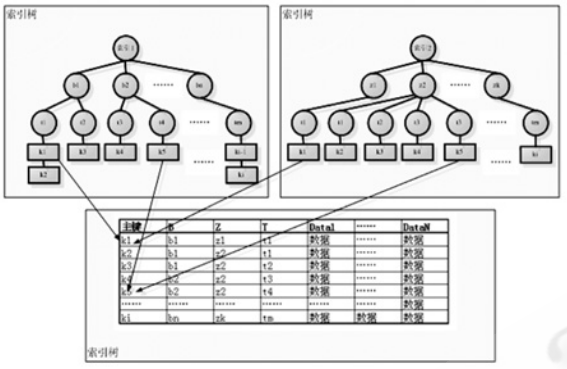

图 3　Cdex 数据状态示例

2.2.2　Cdex 功能与接口设计

Cdex 对外采用 TCP 端口提供服务，接口符合 HTTP1.1 协议。数据上行采用类 SQL 语法，易于掌握和记忆；数据下行采用 XML 对执行结果和数据进行封装。

主要功能命令包括：创建数据表、创建索引、插入数据、删除数据、修改数据、查询数据等。

2.3　存储模块（Cdata）设计

存储模块内部分两个组成部分：存储部分和缓冲部分。存储部分基于 BerkeleyDB，开发了内部访问接口；缓冲部分采用 MemCached，增加业务数据封装与解析逻辑。Cdata 内部将存储、缓冲、协议解析整合成为体，对外提供符合 HTTP1.1 协议的服务端口。

性能指标为：可存储 200G 数据（>2 亿条），缓冲命中 80% 时每秒可进行 300 次存取操作。

2.3.1　Cdata 内部结构设计

Cdata 内部分为三个组成部分：1，基本存储部分，采用 BerkeleyDB 改进版，以 Key – Value 方式存储；2，数据缓冲部分，采用 Memcached，以 Key – Value – Expiretime 方式存储；3，数据接口与缓冲控制，采用 Apache 实现 HTTP 的外部接口，PHP 程序控制读取与写入。

2.3.2　Cdata 接口与功能

接口采用 HTTP 方式，支持 GET 和 POST 模式数据提交。单条数据最大支持 32Mbyte。如果超出，可以在前端接口 API 中进行切割和拼装。返回结果采用 XML 进行封装。

主要接口功能：插入数据、更新数据、删除数据、读取数据等。

数据存储时采用直存同时更新缓冲的策略，读取时先检查是否命中，如果不命中则更新缓冲后返回（图 4）。

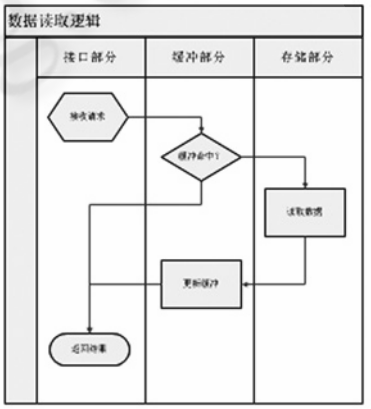

图 4　数据读取逻辑

2.4　接口与逻辑模块设计

接口与逻辑模块用 PHP 语言开发，与前端应用共同部署在 apache 服务器上。通过协议解析层，提供简单易用的操作函数，便于应用开发者进行开发；内部接口层通过网络连接操作索引服务与存储服务，并将数据进行封装/解析操作，返回给使用者需要的数据。

2.4.1　数据组装与内部接口层

在这个层里面，用类封装了对 Cdex 和 Cdata 的基本操作方法，包括插入、删除、读取、修改等。Cdex 和 Cdata 的数据采用 XML 格式，为便于网络传输进行了编码。在本层中需要完成对数据的编码解码工作。

2.4.2　协议解析与用户接口层

这个层里面，通过继承 Cdex、Cdata 的操作类，提供可以直接供前端业务使用的 API 操作类。接口代码封装结构见图 5。

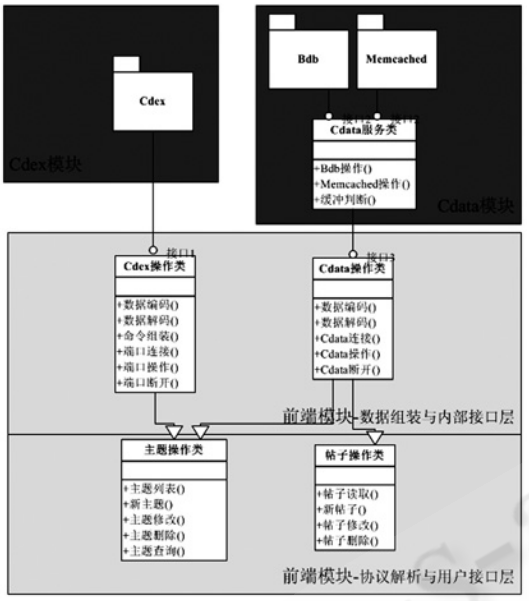

图 5 接口代码封装结构

2.4.3 锁与事务模拟及其他

为了提高整体效率,CSDS 并未提供事务功能。为满足事务操作需求,提供了锁的操作,用以模拟事务操作。锁通过 Cdata 模块的 Memcached 实现,在 Cdata 操作类中提供了对锁的操作方法。用户可以根据业务特点,将锁操作封装在主题操作类或帖子操作类的特定操作函数中。

为了纠错和调试,还提供了一些功能,如 :debug 方法,批量数据删除等。

3 CSDS 在社区论坛产品的应用实例

CSDS 系统原型开发完成后,在‘ 搜狐社区新平台项目 ”中进行实际的考察应用。

3.1 逻辑实现方案

3.1.1 名词定义 :

bid 母版面标识,zid 子版面标识,tid 标签标识,pid 主题标识

3.1.2 在 Cdex 中创建表和索引 :

Create table (pid key, bid, zid, tid, lasttime,)

Create thread (pid key, bid, lasttime) 母版索引

Create thread (pid key, zid, lasttime) 子版索引

Create thread (pid key, tid, lasttime) 标签版索引

3.1.3 定义主题信息结构,存储在 Cdata 中

KEY = ‘post_’+pid

VALUE = XML(title, pid , bid, zid, tid, lasttime, authorid, replynumber)

3.1.4 定义主题内容信息结构,按每一页一个单元存储

KEY =‘page_’+ pid + pageid,

VALUE =XML(XML(authorid, posttime, contents), XML(authorid, posttime, contents), … …)

3.1.5 发帖逻辑

生成唯一 pid,创建主题信息结构存入 Cdata,创建主题内容信息结构存入 Cdata,插入新节点到母板索引、子版索引、标签版索引

3.1.6 回帖逻辑

pid 加锁,读取并更新主题信息结构,读取并更新主题内容信息结构,更新母板索引,更新子版索引,更新标签版索引,解锁。

3.1.7 读版逻辑

根据 bid 或者 zid、tid,从对应的索引读取信息,读取主题信息结构,显示页面

3.1.8 读贴逻辑

根据 pid 和 pageid,读取主题信息结构,读取主题内容信息结构,显示页面

3.2 服务部署方案

核心服务器共 4 台服务器,包括 1 台部署 Cdex,一台部署 Cdata,两台部署 apache + php 作为前端应用。辅助服务器采用 1 台 Mysql,用于用户表、版面关联信息表、关键词屏蔽表等辅助功能的需求。

3.3 性能测试与评估

测试数据表明,每组服务器上,可以存储 1000 万个主题、2 亿个帖子。按照每天 1 万个主题、10 万个回帖计算,至少可以容纳 3 年的信息。

模拟测试结果显示,每组服务器在 100 万主题、2000 万帖子的情况下,可以支撑 500 万次/天的访问。

3.4 可能存在的风险与预防措施

对突发访问的预防 :突发高访问可能会对 Cdata 服务器产生较大压力。可以根据实际情况,在 Cdata 服务器上对并发连接数进行限制,以保证服务稳定。

历史数据超出容量:多年的数据积累,可能导致超出容量。Cdex 容量可以通过设置归档机制或快慢表方式解决;Cdata 容量可以通过增加硬盘扩大存储空间解决。

4 结果分析与实验结论

通过对 CSDS 系统的深入剖析和在“搜狐社区新平台项目”(图 6)的实际检验,证明 CSDS 系统是能够通知满足海量存储和高效率访问的数据系统,尤其是在论坛类应用上具有相对于传统关系型数据较大的性能优势。

图 6 应用 CSDS 的“搜狐社区新平台项目”

总体上来看,CSDS 系统是一种分布式的、通用的、可扩展的、高效率的数据系统,在特定的应用需求下用以取代关系型数据库,提高产品整体的存储容量和访问速度。在企业被日益庞杂的 DBA 和不断增长的硬件成本所困扰的情况下,CSDS 系统是一种全新的解决思路。

参考文献

1 开源项目“B + Tree”,组织者 Lawandco,2003,http://sourceforge.net/projects/btreeindex/

2 开源产品“BerkeleyDB”,甲骨文公司,2007,http://www.oracle.com/technology/products/berkekey - db/

3 开源项目“Memcached”,Danga Interactive 公司,2007,http://danga.com/memcached/

4 西尔伯沙茨等著,杨冬青译. 数据库系统概念第五版,机械工业出版社.

5 C. J. Date 著,数据库系统导论(第八版),中国电力出版社.

6 Michael Kofler 著,杨晓云等译. MySQL5 权威指南,人民邮电出版社.

7 Elmasri 等著,张伶等译. 数据库系统基础,中国电力出版社.