

新型数据库技术概论

蒙卓

(上海工程技术大学机械工程学院, 上海 201620)

摘要：关系型数据库自 1980 年诞生以来，现已成熟，而为了应对 Web2.0 下的数据增长速度快，可伸缩性高的新型数据库以非 SQL、去中心化、分布式、通信协议简单为发展方向，本文以豆瓣公司研发的 BeansDB 为例进行论述。

关键词：数据库 NoSQL BeansDB

The Overview of Database's Tendency

Meng Zhuo

(College of Mechanical Engineering, Shanghai University of Engineering Science, Shanghai
201620, China)

Abstract : Since the Relational Database Management System (RDBMS) developed in 1980 and nearly completed today which is manipulated by Structured Query Language (SQL) becomes mainstream type of database. The other database characterized by NoSQL (None SQL), embedded, distributed, simplified communicate protocol due to the explosion of data. Douban Inc's BeansDB, one of the top domestic databases in China, is themed by this article.

Key words : Database NoSQL BeansDB

0 引言

数据库，可以被视为能够进行自动查询和修改的数据集。数据库有很多种类型，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。数据库存在多种模型。而应用于大型数据储存的数据库一般为网状数据库 (Network Database)、关系数据库 (Relational Database) 以及面向对象式数据库 (Object-Oriented Database)。此外也有应用在 LDAP (轻量级数据访问协议) 的层次结构式数据库 (Hierarchical Database)。SQL 是高级的非过程化编程语言，它允许用户在高层数据结构上工作。它不要求用户指定对数据的存放方法，也不需要用户了解其具体的数据存放方式。而它的界面，能使具有底层结构完全不同的数据库系统和不同数据库之间，使用相同的 SQL 作为数据的输入与管理。它以记录项目 [records] 的合集 (set) 作为操纵对象，所有 SQL 语句接受项集作为输入，回提交的项集作为输出，这种项集允许一条 SQL 语句的输出作为另一条 SQL 语句的输入，所以 SQL 语句可以嵌套，这使它拥有极大的灵活性和强大的功能。在多数情况下，在其他编程语言中需要用一大段程序才可实践的一个单独事件，而其在 SQL 上只需要一个语句就可以被表达出来。这也意味着用 SQL 可以写出非常复杂的语句，但是又非常冗余。在面对大量的数据迸发时，这样的 SQL 不仅对计算机的负载加大，而且实时性也不佳，没有抗数据破坏的能力。对于需要经常修改数据的 WEB2.0 站点来说，这样的缺陷会导致用户体验变差，由此 BeansDB 应运而生。

1 Key-Value 架构

Key-Value (键-值) 架构是所有 NoSQL 的共通点和基本点，这一架构历史非常长，著名的 BerkeleyDB 就是使用这一架构，Key 唯一代表一个数据对象，对该数据对象的读写操通过 Key 来完成，Value 即要寻找

的数据记录值。整个数据库只有两个列，因此访问速度较关系型数据库快。缺点是搜索算法单一，初期架设和转移困难。

2 BeansDB 架构

豆瓣的 BeansDB 采用了 Amazon 公司的 Dynamo 架构，如图 1 所示，并对其进行了优化，这种架构是完全的分布式，去中心化的，存储层同样也是分布式的。这样即使数据中心遭到毁灭性打击，在其他节点的恢复下，系统一样能正常工作。BeansDB 的可扩展性和可用性采用的都比较成熟的技术，与 Dynamo 不同的是数据分区并不是用一致性哈希(consistent hashing)方式进行复制，而是用最终内容上的一致性哈希值进行校验，利用数据对象的版本化实现一致性。复制时因为更新产生的一致性问题的维护采取类似 quorum 的机制以及去中心化的复制同步协议。同 Dynamo 一样 BeansDB 是完全去中心化的系统，人工管理工作很小。BeansDB 特点为总是可写和可以根据应用类型优化，设备价格低廉却有着巨大的数据吞吐量和快速迸发能力。同 Key-Value 架构一样，BeansDB 也有 Key 值。而 BeansDB 增加了节点(node)：例如自带硬盘的主机。每个节点有三个 Java 写的组件：请求协调器(request coordination)、成员与失败检测、本地持久引擎(local persistence engine)，本地持久引擎支持不同的存储引擎。BeansDB 上最主要的底层引擎是 Berkeley Database Transactional Data Store，其他还有 BDB Java Edition、MySQL 以及一致性内存 Cache。实例(instance)：每个实例由一组节点组成。除此之外 BeansDB 还有容易扩展的特性，即可以在不中断服务的情况下进行容量扩展。

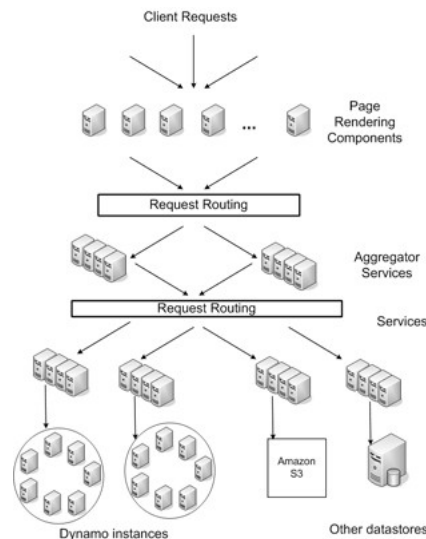


图 1.Dynamo 架构示意图

3 实际使用情况

表 1 是在 150 个请求每秒(qps)，有内容分发网络(CDN)作为缓存的情况下的负载。

数据量(1KB-20MB)	16Mx16x3	460Gx16x3
响应时间 (ms)	18/35	86/266
系统负载(%)	90	99

表 1.CDN 测试使用情况

表 2 是在 180 个请求每秒(qps)，有内存缓存(memcached)作为缓存的情况下的负载。

数据量(10B-100KB)	16MBx16x3	460Gx16x3
-----------------	-----------	-----------

本文为作业论文，仅供参考

响应时间 (ms)	3/10	21/65
系统负载(%)	90	99

表 2.内存缓存测试使用情况

4 结论

可见对于海量数据处理方面，BeansDB 架构确实有着高速响应，随时写入的特点。对于 Web2.0 的发展各大国外网络服务公司,如 Facebook, Twitter, Google 都开始采用了这样或那样的分布式的 NoSQL 的架构的服务，国内的如豆瓣和支付宝也采用了这种先进的数据库技术来应对新时代下的互联网发展。

参 考 文 献

- [1] Giuseppe DeCandia, etc. Dynamo: Amazon's Highly Available Key-value Store [OL]. <http://s3.amazonaws.com/AllThingsDistributed/sosp/amazon-dynamo-sosp2007.pdf>
- [2] 刘巍巍,徐成,李仁发.嵌入式数据库Berkeley DB的原理与应用[J].科学技术与工程,2005(5)
- [3] Adya, A., Bolosky, W. J., Castro, M., Cermak, G., Chaiken, R., Douceur, J. R., Howell, J., Lorch, J. R., Theimer, M., and Wattenhofer, R. P. 2002. Farsite: federated, available, and reliable storage for an incompletely trusted environment. SIGOPS Oper. Syst. Rev. 36, SI (Dec. 2002), 1-14.
- [4] Davis Liu. BeansDB设计与实现[OL]. <http://beansdb.googlecode.com/files/Inside%20BeansDB.pdf>
- [5] 王京谦，万莅新.开源嵌入式数据库Berkeley DB和SQLite的比较[J].单片机与嵌入式系统应用,2005, (2)
- [6] 冯大辉. Amazon的Dynamo架构[OL]. http://www.dbanotes.net/techmemo/amazon_dynamo.html
- [7] Merkle, R. A digital signature based on a conventional encryption function. Proceedings of CRYPTO, pages 369-378. Springer-Verlag,1988.
- [8] BERNSTEIN, P.A., AND GOODMAN, N. AN ALGORITHM FOR CONCURRENCY CONTROL AND RECOVERY IN REPLICATED DISTRIBUTED DATABASES. ACM TRANS. ON DATABASE SYSTEMS, 9(4):596-615, DECEMBER 1984
- [9] Satyanarayanan, M., Kistler, J.J., Siegel, E.H. Coda: A Resilient Distributed File System. IEEE Workshop on Workstation Operating Systems, Nov. 1987.
- [10] SAITO, Y., FRÖLUND, S., VEITCH, A., MERCHANT, A., AND SPENCE, S. 2004. FAB: BUILDING DISTRIBUTED ENTERPRISE DISK ARRAYS FROM COMMODITY COMPONENTS. SIGOPS OPER. SYST. REV. 38, 5 (DEC. 2004), 48-58.