

# NoSQL 非关系型数据库的发展及应用初探

黄贤立

(淮阴师范学院计算机科学与技术学院 江苏 淮安 223300)

**【摘要】:** 随着 Web 2.0 技术的发展,网络环境下的数据库应用出现了一些新的变化。NoSQL 是在这种新变化之下产生的一种非关系型数据库产品的总称。文章分析了 NoSQL 的起源与发展,通过与传统关系型数据库产品的对比,指出了 NoSQL 的优缺点,讨论了其应用前景。

**【关键词】:** NoSQL; 数据库; 并发; Key/Value

## 1、NoSQL 的起源与发展

NoSQL 的发展最早可以追溯到 1991 年 Berkeley DB 第一版的发布。Berkeley DB 是一个 Key/Value(键/值)类型的 Hush 数据库。这种类型的数据库适用于数据类型相对简单,但需要极高的插入和读取速度的嵌入式场合<sup>[1]</sup>。

NoSQL 得到真正的快速发展开始于 2007 年,从 2007 年到现在,先后出现了十多种比较流行的 NoSQL 产品,从 2009 年开始,国内的 NoSQL 领域也开始活跃起来,豆瓣的 BeansDB,人人网的 Nuclear 开源 NoSQL 产品以及盛大创新院的 TCDatabase 纷纷发布。

NoSQL 能够得到快速的发展,其主要背景在于 Web 2.0 技术在网络中的广泛应用。在 Web 2.0 环境下,用户对于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等,都对传统关系型数据库带来很大的困难。典型的例子是北京奥运会的订票方案,由于最早实施的是网站、电话申请,先到先得的原则。在开始订票的当天,数以百万计的用户同时涌入奥运会官方票务网站,大量的并发请求使得该网站在短短几分钟内死机,一直到当天晚上才恢复工作。而北京奥组委不得不紧急调整了预售方式,改为在规定时间内申请,之后进行摇号的方式发售,这才解决了这一问题。

## 2、传统关系型数据库在 Web 2.0 环境下存在的问题

通过上述例子可以看出,传统的关系型数据库难以适合现在的 Web 2.0 环境下可能出现的众多并发读写请求,特别是超大规模和高并发的 SNS 类型的 Web 2.0 纯动态网站更是力不从心。可以看出,传统关系型数据库面临的挑战主要有以下几点<sup>[1]</sup>:

### (1)对数据库高并发读写的需求

Web 2.0 网站要根据用户个性化信息来实时生成动态页面和提供动态信息,无法使用动态页面静态化技术,因此数据库的并发负载非常高,往往要达到每秒上万次的读写请求,此时的磁盘 I/O 根本无法承受如此之多的读写请求。

### (2)对海量数据的高效率存储和访问的需求

类似 Facebook、Twitter 和 Friendfeed 这样的 SNS 网站,每天用户产生海量的用户动态,以 FriendFeed 为例,一个月就达到了 2.5 亿条用户动态,对于关系型数据库来说,在一张 2.5 亿条记录的表里面进行 SQL 查询,效率是极其低下的。再例如大型 Web 网站的用户登录系统,如腾讯、盛大,动辄数以亿计的帐号,关系数据库也难以应付。

### (3)对数据库的高可扩展性和高可用性的需求

在基于 Web 的架构中,数据库是最难进行横向扩展的,当用户量和访问量增加时,数据库没有办法像 Web Server 那样简单的通过添加更多的硬件和服务结点来扩展性能和负载能力,对于很多需要 24 小时不间断服务的网站来说,对数据库系统的

升级和扩展往往需要停机维护。

## 3、NoSQL 和关系型数据库的区别<sup>[2]</sup>

### (1)横向和纵向扩展能力

关系型数据库通常部署在一台服务器上,通过增加处理器、内存和硬盘来升级。部署在多台服务器上的关系型数据库通过是依赖互相复制来保持数据同步。NoSQL 数据库可以部署在单服务器上,但更多的部署是成云状分布,在 NoSQL 的模式一文中<sup>[3]</sup>,作者详细介绍了 NoSQL 数据库的部署、数据分区、数据复制等问题的解决方案。

### (2)列, key/value 存储, 数组(Tuples)存储

关系型数据库通常是由表或视图里的字段构成(固定的结构,用各种操作相互关联)。NoSQL 数据库通常存储的是一对键值或数组(Tuples),其结构不固定,只是一个有顺序的数据队列。

### (3)数据的内存和硬盘使用

关系型数据库通常是驻留在一个硬盘内或一个网络存储空间里。SQL 查询或存储过程操作会把数据集提取到内存空间里。一些(并不是全部)NoSQL 数据库可以直接在硬盘上操作,也可以通过内存来加快速度。

## 4、NoSQL 存在的不足

与传统的关系型数据库相比, NoSQL 非关系型数据库在 Web 2.0 的高并发实时环境下有着一定的优势,但也存在着一些问题,主要体现在<sup>[2]</sup>:

### (1)NoSQL 很难实现数据的完整性

由于在 NoSQL 项目中很难实现数据的完整性,而在企业应用中数据完整性又必不可少。因此,目前的 NoSQL 项目很难在企业中普及开来。需要注意的是,这一不足在 Web 2.0 的很多应用中并不重要,因为很多 Web 实时系统并不要求严格的数据库事务,对读一致性要求很低,有些场合对写一致性要求也不高。

### (2)缺乏强有力的技术支持

到目前为止, NoSQL 项目都是开源的,他们缺乏供应商技术人员的支持。

### (3)开源数据库从出现到被用户接受需要一个漫长的过程。

### (4)关系型数据库在设计时更能够体现实际。

## 5、NoSQL 的应用

尽管还存在着一些不足之处,但我们更应该看到 NoSQL 在 Web 2.0 环境下的应用前景。例如,在我们所设计的一个类似于 ACM 竞赛的在线测试系统中,如果采用关系型数据库进行存储,记录数等于用户数与用户数回答题目的乘积,当用户很多的时候,数据的记录数会非常的多,对系统的性能造成影响。在系统设计时我们引入了 Oracle Berkeley DB,将 Berkeley DB 与 My-SQL 相结合,对于用户注册信息、题目本身的信息等,仍然采用传统的 My-SQL 关系型数据库,而对于注册用户对于问题的解答结果,则采用 Berkeley DB 进行存储,从而(下转第 45 页)

质的角色,供超级用户选择。这样每个操作员在处理各种业务时,只能按照本角色具有的权限对数据进行操作,这样就可以保证审核、批复数据的合法性,只有建设业主才具有成为系统管理员的可能。

## 2.2 严格控制使用系统的用户

公路建设过程是一个动态的管理过程,随着项目的开工和竣工,相应的监理单位、承包单位是不断变化的,项目开工参建单位入场时,在系统管理员的授权下,开始使用系统,某一单位离场时,系统管理员收回授权,此单位不能够再使用本系统。严格控制无关用户使用系统,将系统与外界隔离,这就需要严格的把好身份验证关。通过对常用的身份验证方式分析(见表1),结合"用户名+口令"和"IC卡验证"两种方法的特点,采用系统分发密钥文件和"用户名+口令"结合的方式,来验证用户的身份是一种比较好的解决方案。密钥文件控制使用单位是否合法,在使用单位合法的情况下,用"用户名+口令"用来验证登录用户是否合法,并据此分配操作权限。

验证方法	应用成本	方便性	安全性	使用特点
用户名+口令	低	好	低	网络环境中,密码很容易散播、泄漏。
IC卡验证	高	差	高	IC卡技术成熟,有标准可循,成本居高不下,并且携带不便。
利用计算机的特征码验证	低	差	高	由于其管理复杂,灵活性较差,现在仅限于特殊的应用方面。
人体特征码验证	高	差	高	需要专门的读取录入设备,使用仅限于安全性要求极高的场合。

常见的身份验证方法比较表(表1)

密钥文件验证机理:用户在每次登陆系统时,由应用服务器产生一个随机数发给客户端,客户端用此随机数作运算的输入返一个运算结果给服务器,服务器端作相同的计算。并把计算结果与客户端上传的结果相比较,结果正确即可进入下一步的操作。全过程可简单由以下关系式来表示:

$$\text{client}(S, K) = \text{Server}(S, K)$$

其中S代表由服务器提供的随机字符串,而K则代表密钥,client(S, K)代表是插在客户端的client所进行的运算,Server(S, K)代表提服务器程序的运算。而等式两端的K并不在客户端出现,也未直接在网上发送,这就保证了此认证方案的可行性。

密钥文件的管理:系统管理员在注册新单位时,为每个单位生成一对密钥(具有活动非活动状态),这对密钥的生存期为该单位进场和离场之间的时间段,系统管理员将密钥文件(公钥)以软盘为介质给刚进场的单位分发下去,系统管理员可以在该单位离场后将该单位密钥(私钥)的状态改为非活动状态。

密钥的验证过程:用户的身份验证是在每次登陆系统时进行的,其验证步骤如下:

第一步:服务器端判断密钥文件的存在与否,若存在则由服务端产生随机数并传给客户端,反之不能进行下一步工作。

第二步:客户端通过分发的密钥文件对发送的随机数进行加密,从而得到128位的摘要,然后将这个摘要返回给服务器端。

第三步:服务端将客户端发来的摘要和使用服务端密钥产生的摘要比较,相同则认为用户合法,否则非法。

第四步:根据服务器端密钥文件的标志位分析,判断该密钥是否处于活动状态,如果处于活动状态,则进行身份和口令验证,反之则认为用户非法。

第五步:通过用户登陆身份和登陆口令进行判别,如果用户的身份和用户口令相吻合,即可登陆系统。

严格进行身份认证,完成对使用系统用户的控制,将系统和外界隔离起来是保证系统性能和数据安全性的有效手段之一。

## 2.3 用户口令和用户身份的保护策略

公路工程建设计量与支付软件监理的审核和业主的批复决定着工程款的支付,监理、业主的权限一旦被盗用,其后果将非常严重。互联网是通过存储转发的包交换网,数据封闭在IP包中,但数据有可能被截获并复制,如何保护用户登陆信息尤为重要。系统在用户口令、用户名称等重要信息传输前,先随机产生一序列,对每个字符的存储位进行异或运算,加密的方法蕴含在只有解密算法方能识别的校验位中,加密后的字符和校验位一起由网上发送,在服务器端以校验位参数,通过约定的算法产生解密的数据序列,然后由此序列对数据进行解密就可以得到最终的结果。重要数据特别是用户登陆信息的加密是保护系统重要数据安全性的手段之一。

## 2.4 对运行的数据库监控和管理

对于网上运行的应用系统,除了提供保护数据的安全防范措施外,还需要防范意外,加强对数据库的监控和数据库的定期备份。如果发现不合法用户的入侵,就可以将损失减少到最小范围。系统对一些重要审核、批复的数据均设有监控管理,将其写入操作记录表中,如果发现出现不合法用户入侵,便可以查看这些数据运行记录,据此对数据进行恢复操作。

## 参考文献:

- [1]基于局域网的web动态数据库 方文华 工业计量 2001-05-26
- [2]高速公路工程建设计量支付管理数据库设计 叶智锐 工路交通科技工作者 2003-02-20
- [3]简单实用的计量所信息服务系统 刘海燕 中国计量 2008-04-10
- [4]基于web的实时控制系统研究 尹贵虎 浙江大学 2003-01-01
- [5]基于web服务的网络监控系统设计与实现 余运强 浙江大学 2005-02-01
- [6]浅谈高速公路计量软件的开发与应用前景 谷英辉 山西建筑 2008-01-10

(上接第30页)

使系统中的答题记录数减少到注册的用户数,从而大大提高检索的速度,加快系统的响应时间。

## 6、结论

可以看出,相对于传统的关系型数据库来说,NoSQL在面向较多的并发用户数及海量数据的处理方面存在着一定的优势,目前已经在国内的部分网站的建设中得到应用,如视觉中国网站的建设[4],相信在未来的若干年中,NoSQL能够发展的更为成

熟,更好的适应Web 2.0环境的要求。

## 参考文献:

- [1]范凯.NoSQL数据库综述[J].程序员.2010(6):76-78.
- [2]李莉莎.关于NoSQL的思考[J].中国传媒科技.2010(4):40-41.
- [3]Ricky Ho.NoSQL的模式[J].程序员.2010(2):95-99.
- [4]潘凡.从MySQL到MongoDB——视觉中国的NoSQL之路[J].程序员.2010(6):79-81.