**大 连 理 工 大 学 本 科 外 文 翻 译**

**一个支持NoSQL数据库ACID特性的中间层的解决方案**

**A middle layer solution to support ACID properties for NoSQL databases**

学 部（院）： 国家示范性软件学院

专 业： 软件工程英语强化

学 生 姓 名： 李德钊

学 号： 201292467

指 导 教 师：

完 成 日 期： 2016年4月1日

大连理工大学

Dalian University of Technology

**一个支持NoSQL数据库ACID特性的中间层的解决方案**

Ayman E. Lotfy, Ahmed I. Saleh b, Haitham A. El-Ghareeb c, Hesham A. Ali

沙特国王大学

摘要：本文的主要目的是保持关系型数据库管理系统的一致性优势和数据库事务基本要素的特性，同时通过提供一个中间层来激发非关系型的数据库（NoSQL）好处。拟议的中间层使用四个阶段提交协议,以确保：最近的数据的使用，使用恶意技术禁止别人在使用时处理数据，以及数据更新驻留在很多地方,以避免数据丢失和令人失望的事情发生。这种机制是必需的,尤其是在分布式数据库应用程序基于非关系型的数据库（NoSQL）的环境，由于允许冲突的事务继续不仅浪费有限的计算能力和降低带宽，也加剧了冲突。中间层继续跟踪所有正在运行的事务和管理与其他数据层的并发事务的执行。该解决方案将帮助增加的可伸缩性和吞吐量。最后,实验结果表明，系统的吞吐量提高了在场景和增加中间层的数量在一个事务中更新阅读的数量增加。这也符合执行许多交易相关的数据通过更新彼此相同的数据。系统的可伸缩性和可用性则不受影响，同时保证严格的一致性。

关键词：非关系型的数据库，数据库事务正确执行的四个基本要素，统一性，库，事务，并行机制。

1. **介绍**

关系数据库管理系统(RDBMs)来存储和管理数据，也保持数据之间的关系通过限制的使用，如主键和外键。以防数据的规模不大,系统的用户并不多,数据一致性的要求是强制性的,关系数据库管理系统(RDBMs)将是完美的解决方案。由于数据存储在一台机器上，没有任何挑战来管理它们之间的关系。今天的网络应用面临着一个挑战，数以百万计的用户谁是分布在世界各地，谁都期望服务总是可用的、可靠的，并具有高度的一致性。随着用户数量的增加和所产生的数据量的增加，数据必须存储在多个服务器，可以分布在不同的位置。在多台服务器上分配数据使得维护数据之间的关系变得困难。Web应用程序需要在多台服务器规模上的能力，这一挑战导致一种新趋势的出现叫关系数据库管理系统的数据库管理系统（NoSQL DBMS）。关系数据库管理系统有能力在多个节点上分配数据，提供所需的可用性水平，同时保持可扩展性在接受的水平，并忽略了数据的一致性。

分布式数据库系统的问题是，我们必须克服一种的三个基本属性：可用性、一致性和分区根据埃里克·布鲁尔教授上限定理的证明。上限定理指出，任何数据库管理系统都可以提供三个属性：一致性意味着数据库管理系统为用户提供同一版本的数据，高可用性系统在任何时候对用户的数据进行响应，并且对许多计算机的网络分区数据的容忍度。

1. **背景和基本概念**

在这一部分中，对传统的关系数据库相关的基本概念、性质以及NoSQL数据库的基础属性也将介绍。同时关系数据库和关系型数据库管理系统相关的挑战也作了介绍。

**2.1 传统的关系数据库和数据库事务正确执行的四个基本要素属性**

关系数据库是一种存储数据集合中的关系的系统。保存在关系中的数据通过主、外密钥之间的关系。典型的应用程序关系称为表，这种应用的例子是微软SQL Server、Oracle数据库管理系统，和 IBM DB2. SQL语言在关系数据库系统中使用管理数据。SQL语言的范围包括从多个节点、插入、查询数据的删除、更新等操作。由于其丰富的功能，如查询功能和事务管理，他们似乎是适合几乎每一个可能的任务。一个关系数据库管理系统的重要功能是提供增删改查操作为事务执行交易。数据库维护增删改查则难以在分布式数据维护。

**2.2 关系型数据库管理系统数据库和基本性能**

关系型数据库管理系统是一个总称，包括一组非关系型数据库管理系统。它提供了良好的横向扩展简单的读/写操作数据库分布在多台服务器，与传统的关系型数据库，有很少或没有规模水平的能力。关系型数据库管理系统不需要一个固定的表结构，不提供一个增删改查的支持。它提供了最终的一致性，这意味着数据将在一段时间内是一致的。一些经常使用的关系型数据库有CouchDB, Riak, Cassandra, Mnesia, BerkeleyDB, HamsterDB, MongoDB, and Redis。

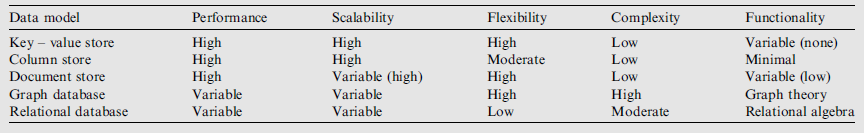
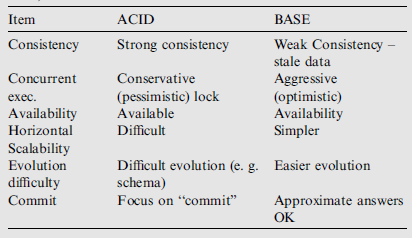
有对关系型数据库区分的各种方法，各有不同的类别和子类别。然而，大多数人会同意的基本分类是基于数据模型的。我们可以根据数据模型的列，文件分类，K值和图来区分关系型数据库。关系型数据库是基于功能的分类及其特征覆盖评估。表1总结了这些特性。

表1 数据库类型和功能。

在面向文档的数据库，每个数据库由若干集合（如关系数据库中的表）每个集合包含许多写在二进制结构化对象符号文件（BSON）格式。每个文档都有一个可以访问此文档的标识。每一个文档由键-值对组成。我们可以在特定集合中的特定文档中获取或更新特定的键。SimpleDB，MongoDB和coachdb一些面向文档的NoSQL数据库的例子。在键/值存储数据存储在一个哈希。关键是一个独特的标识符和值是各自的数据。数据结构类似于字典。插入、删除和更新操作应用于每个给定的键。一些关键-值对可以分组在桶中，该键有一个部分，确定了桶。更新和操作是在按键上完成的。列存储在行中保持数据。每行有一个唯一的标识符称为键和一个或多个列。列为自己的键-值对。该列名称不需要预先定义，所以该结构不固定。行中的列按其键（名称）的顺序保存。例如谷歌的BigTable，HBase和来自脸书的Cassandra。

关系型数据库管理系统可以提供高吞吐量。例如，列存储Hypertable，谷歌追求的BigTable方法允许本地搜索引擎Zvent存储每天十亿数据单元。关系型数据库提供有效的横向扩展，意味着数据可以驻留在多个机器上，如果空间不够用的数据，其他机器可以很容易的添加。当数据驻留在一个单独的机器并且扩展是通过多核即传播之间的负荷机器的CPU和RAM资源,动态规模水平扩展通常更容易通过添加更多的机器到现有池——垂直扩展通常局限于一台机器的能力,除此之外扩展能力通常需要停机时间和有一个上限。水平缩放的一个很好的例子是Cassandra，MongoDB。对于垂直缩放是一个很好的例子–RDS MySQL（MySQL的云版）提供了从小到大尺度的开关机垂直的一个简单的方法；这个过程经常涉及到机器的停机时间。

为了区别于数据库，关系型数据库介绍什么是称为属性即增删改查属性性能。库的方法，根据布鲁尔的增删改查属性上限定理证明的一致性和隔离的来进行区分。其基本的缩写词由以下几个特点构成：基本可用，软状态，最终一致性。布鲁尔对比增删改查的属性见表2，总结了基本属性在以下方式:一个应用程序基本上（基本上可用），不需要一直保持一致（软状态），但最终将在一些已知状态（最终一致性）相比之下增删改查属性提供了严格的一致性。严格一致性意味着所有读操作必须返回来自最新完成的写操作的相同数据。这样一个严格的一致性，不能实现在一起的可用性和分区公差根据CAP定理。

表2 布鲁尔说ACID和BASE比较

最终一致性意味着所有的读操作可能会有不同的数据来自最新完成的写操作的回报，但随着时间的推移：“在一个稳定状态”，系统最终将返回最后一个写入的值。因此，客户端可能会面临一个不一致的状态的数据更新正在进行中。例如，在一个复制的数据库更新可能会去一个节点，复制的最新版本的所有其他节点包含一个副本的修改后的数据集，这样的副本节点，最终将有最新的版本。表3总结之前的相同之处和不同之处。

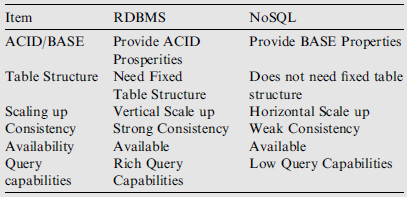


表3 RDBMS和NOSQL数据库不同之处

**2.3 关系型数据库的相关问题**

面对关系型数据库管理系统有很多的挑战。一个重要的挑战是如何添加某种程度的数据一致性以及提供增删改查的属性。作为数据存储在多个服务器，有一个挑战，如何做快速搜索存储在关系型数据库管理系统等分布式数据，以及如何从分布式数据获取商业情报信息。“商业智能和关系型数据库”和“蜂巢：仓库解决方案随着使用映射-规约模式框架和分布式搜索大型的关系型数据库”的文章地址前面的挑战。如何在多节点的分布式数据，关系型数据库面临的一个挑战。文章如“'Nosql和Hadoop基于Oracle云”、“数据存储运行的10条规则”等等这些挑战。

**2.3 分布式数据与并发控制**

分布数据有两种：在多个服务器上共享分布式的不同数据，因此每个服务器都充当数据子集的单一来源。而另一个则是复制数据在多个服务器上的复制，每一位数据都可以在多个地方找到。一个系统可以使用其中的一种或者两个技术。复制有两种形式：主从复制，这使得一个节点的权威副本，处理更新，而奴隶与主机同步，并可以处理读取。另一种称为多主复制。与主从复制相比，它允许将写入任何节点和节点坐标来同步数据的副本。主从复制减少更新冲突的机会，但多主复制避免将所有写入到一个单独的点上。

如果它们属于不同的交易访问相同的数据项目，至少有一个是写操作的操作冲突。当一个客户端在另一个客户端读取数据的同时更新相同的数据时，写的冲突发生在同一时间，同时读取-写入冲突时发生。保证一致性的方法有两种。首先是悲观的方法，其锁定数据记录以防止冲突。二是乐观的态度，执行任何交易，即使违反完整性规则，如可串行化，它检测到冲突后的固定。

1. **相关工作**

近年来，关系型数据库已由关系型存储系统，这通常灵活一致性保证在更多的可扩展性和可用性。不同的解决方案和介绍高度一致，提供基于关系型数据库的增删改查操作，键值存储。一种方法是在数据存储区中实现事务支持。这是复杂的，是难以实现的，而不影响可扩展性和可用性。在我们所提出的解决方案中使用的另一种方法是使用中间层作为客户端和数据库管理系统之间的接口。该层将支持酸特性和并发控制。另一种方法是为每个数据存储定义一个事务访问协议。

该协议提供了一个事务和数据存储的抽象接口，使客户端应用程序访问与事务语义的数据。还建议的协议维护的数据存储的可扩展性和可靠的访问的优点。但是，这种方法需要从数据库管理系统和客户端本身发送元数据。来自客户的数据将不能准确。

谷歌超大卖场是一个事务性的索引记录管理在大型表数据的顶部。超大卖场支持ACID事务跨多个数据项。然而,程序员必须手动链接数据项分为分层组，并且每个事务只能访问一个组。而使用Paxos协议建立一个可扩展的，一致的和高度可用的数据存储提供的只是一个单一的项目一致性保证。

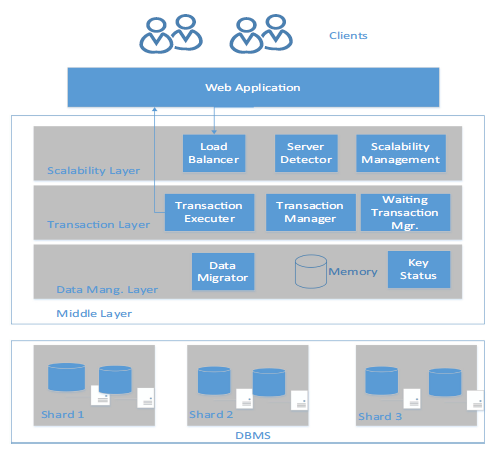


图1拟定的中间层的概述。

公共开放策略服务（COPS）是一个键值存储，交付这整个广域一致性模型。公共开放策略服务的一个重要贡献是它的可扩展性,可以执行因果键存储到整个集群之间的依赖关系,而不是一个单独的服务器。公共开放策略服务的核心方法是跟踪和明确检查是否在本地集群中暴露的关键之间的因果依赖关系的关键是在暴露之前执行写的操作。公共开放策略服务使用得到的交易，以获得一个一致的看法，没有锁定或阻止多个键。它使用复制协议优化，以实现更大的性能，同时支持本地多项目交易。

Granola是一个事务协调基础设施构建可靠的分布式存储应用程序。它提供了一个强有力的一致性模型,同时大大降低事务协调开销。Granola引入了一个特定的支持一种新的独立的分布式事务,可序列化没有锁的开销,没有中断写冲突。Granola使用一种新颖的基于时间顺序分布式事务协调机制,提供低延迟和高吞吐量。它使用复制协议优化来实现更大的性能,同时支持本地多产品交易。可扩展的事务跨异构的泛关系型数据库的键值数据存储定义了客户端API，这定义了客户端协调事务管理协议可插入数据存储抽象层使其跨多个数据存储处理事务。它定义了客户端协调事务协议,使有效的多事务跨异构分布式应用程序的键值存储。它还定义了一个数据存储的实现，提供了相应的接口支持多事务。但这种方法取决于客户端时钟也会给非准确的结果，它定义了一个中央层的所有客户数据到达那里。这一层是一个让人失望的单点项。过滤器与快照隔离实现多键交易语义。这取决于中央容错时间戳服务称为时间戳甲骨文生成时间戳帮助协调事务和锁协议来实现隔离。锁定协议依赖于读写操作在检查记录以检查每个记录相关联的锁定域。它不利用测试和设置操作的关键值存储这种技术不适合客户端应用遍及较高延迟提供广域网。没有死锁检测或避免的实施进一步限制了它的使用，在这些类型的网络。

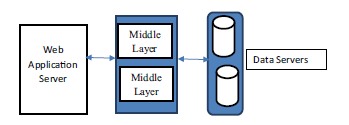


图2 实验建设

1. **提出的解决方案**

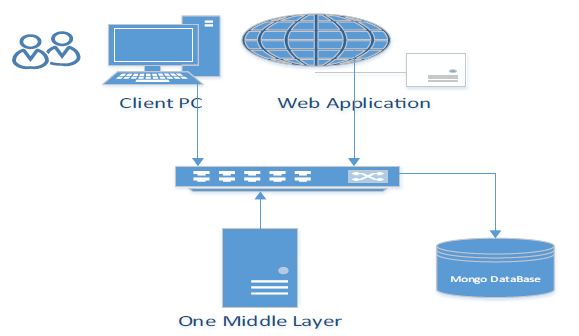


图3 实验Ⅰ的建设

根据上限定理证明任何数据库管理系统可以满足三个之中有两个的属性的一致性、可用性和分区。关系数据库提供了一致性和可用性与增删改查属性而关系型数据库提供可用性和分区的基础属性。本文的目的是改善一致性和提供增删改查属性和关系型数据库的并发事务执行。

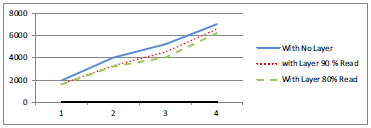


图4 吞吐量与请求数:一个中间层。

其中一个解决方法是修改的关系型数据库引擎本身，但这种解决方案将取决于数据库引擎的改进使它不适用于其他的数据库引擎。它还需要得到源代码是不可用的一些引擎也将代码的理解的努力。

另一个解决方案是基于关系型数据库引擎，为用户的应用如Web应用和关系型数据库引擎像云TPS之间的接口在大厦一层。这个中间层将支持客户如增删改查属性和一致性事务性质和维护的关系型数据库如可用性和分区由底层的关系型数据库引擎已经提供的动机。中间层将取决于我们如何在基础数据库引擎上获取或更新数据。这么小的修改，我们可以使它适合很多非关系型数据库引擎。此外，当事务管理是重要的时候，我们可以把它打开，当事务管理是不重要的情况下，把它关闭，通过一个中间层，这样我们可以更加灵活控制系统。

参考文献：略