**北京理工大学硕士论文**

**题目：浅谈NoSQL**

**姓名： 李楠**

**学号： 2220140563**

**院系： 软件学院**

**指导老师：赵小林**

目录

[一、 引言 3](file:///W:\grd\GrdClass\DB\数据库作业.docx#_Toc373007659)

[二、 NoSQL相对与传统关系型数据库的优点 4](file:///W:\grd\GrdClass\DB\数据库作业.docx#_Toc373007660)

[三、 NoSQL理论基础 6](file:///W:\grd\GrdClass\DB\数据库作业.docx#_Toc373007661)

[四、 NoSQL的分类及应用 9](file:///W:\grd\GrdClass\DB\数据库作业.docx#_Toc373007662)

[五、 关于NoSQL的思考 10](file:///W:\grd\GrdClass\DB\数据库作业.docx#_Toc373007663)

[六、 参考文献说明 11](file:///W:\grd\GrdClass\DB\数据库作业.docx#_Toc373007664)

**浅谈NoSQL**

**李楠**

北京理工大学 软件学院

**摘要：**随着web2.0技术的日益普及，在大数据的网络环境下，数据有了新的内容和特征，传统的关系型数据库已经不能很好的适应。NoSQL打破了传统的关系模型，以一种更自由的方式存储数据。本文介绍了NoSQL相对与传统关系型数据库的优点及其理论基础，简单的介绍了目前NoSQL的几种分类应用，最后阐述了NoSQL存在的问题和未来的发展趋势。

**关键词**：关系数据库，NoSQL

**1引言**

随着互联网web2.0网站的兴起，数据有了新的特征，在海量数据的并发性读写，存储方面，传统关系型数据库暴露出难以克服的缺点，业界为了解决上述问题，推出了新类型的“NoSQL”数据库。

NoSQL是非关系型数据库，是一种与传统的关系型数据库系统完全不同的数据库系统，关系型数据库中的表都是存储一些格式化的数据结构，这样的结构可以便于表与表之间进行连接等操作。而非关系型数据库以键值对存储，它的结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，这样就不会局限于固定的结构，可以减少一些时间和空间的开销。它的数据存储格式可以是松散的、通常不支持Join操作并且易于横向扩展，由于非关系型数据库其本身的特点得到了非常迅速的发展。

**2 NoSQL相对与传统关系型数据库的优点**

**2.1传统数据库在现在所面临的问题**

关系数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。关系模型是由埃德加•科德于1970年首先提出的。

**2.1.1 High performance——对数据库高并发读写的需求**

Web2.0时代，网站通常要根据用户的个性化定制实时生成页面，例如现在流行的新浪微博等。网站几乎要实时地为用户提供信息。该类应用对数据库提出了很高的并发负载要求，传统的关系型数据库应付上万次SQL查询还勉强顶得住，但是应付上万次SQL写数据请求，硬盘IO就已经无法承受了。

**2.1.2Huge Storage——对海量数据的高效率存储和访问的需求**

Web2.0时代，网站信息不在是单方面的提供，而是交互性的提供，用户也不再仅仅是数据的接受方，同时也是数据的提供方，用户提供的信息是海量的。类似新浪微博，qq空间等SNS类型的网站，可能每天都会产生千万级的数据。如果在关系型数据库里的一张存有亿级记录的数据表里作SQL查询，耗费时间巨大。虽然可通过分库、分表等方法切分数据，部分地解决查询问题，但也带来了诸如加重程序开发的复杂度和数据备份以及数据库扩容的复杂度等问题。

**2.1.1 High Scalability && High Availability——对数据库的高可扩展性和高可用性的需求**

在基于web的架构当中，数据库是最难进行横向扩展的，当一个应用系统的用户量和访问量与日俱增的时候，你的数据库却没有办法像web server和app server那样简单的通过添加更多的硬件和服务节点来扩展性能和负载能力。

**2.2 NoSQL的优势**

**2.2.1数据模式**

区别于关系型数据库固定的二维表元组模式，NoSQL没有严格的数据模式，通常存储的是一对键值或数组，其结构不确定，在系统运行中可以动态更改。NoSQL的这种松耦合、可扩展的数据模式，有利于存储在日益广泛的Web应用中日趋重要的半结构化和非结构化数据。

**2.2.2可扩展性**

从根本上说，NoSQL是伴随着分布式系统而产生的，在分布式系统下性能良好，支持横向扩展，能够很好的适应现代Web应用迅猛发展带来的海量数据。

**2.2.3事务完整性**

关系型数据库通过ACID保证数据的完整性，ACID特性也是关系型数据库事务完整性最高级别的黄金标准。ACID，分别代表原子性、一致性、隔离性和持久性。NoSQL数据库优先考虑的是数据库的性能。这是NoSQL项目在企业中难以普及的原因。但另一方面，很多Web实时系统对事务完整性、读一致性要求并不高，某些时候对写一致性要求也不高。

其实NoSQL数据库仅是关系数据库在某些方面（性能，扩展）的一个弥补，单从功能上讲，NoSQL的几乎所有的功能，在关系数据库上都能够满足，所以选择NoSQL的原因并不在功能上。因此我们一般会把NoSQL和关系数据库结合使用，各取所长。

关系型数据库存储了数据之间存在的或潜在的关系，适合于企业数据的存储于查询；而NoSQL数据库更看重数据的存储，适合于现在的Web应用。

**3 NoSQL理论基础**

**3.1 相关理论**

**3.1.1 CAP理论**

CAP理论最早由Eric Brewer教授提出，Seth Gilbert和Nancy Lynch予以证明。CAP理论归纳了分布式系统的三个特性：一致性：系统中的所有数据备份，在同一时刻都是同一值。可用性：每个操作总能在确定的时间内返回，即系统随时都是可用的。分区容忍性：在出现网络分区的情况下，例如断网，分离的系统也能正常运行。

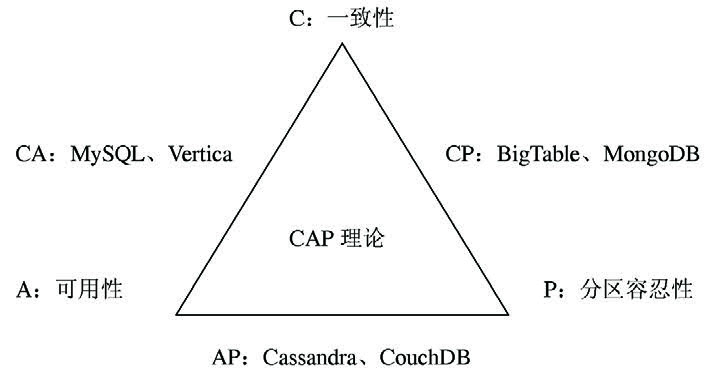


图1 CAP

CAP理论指出，分布式系统不可能同时实现所有这三个特性，系统必须做出权衡，至少牺牲一样以成全其他两样。图1对侧重点不同的数据库产品进行了说明。

**3.1.2最终一致性**

NoSQL中通常有两个层次的一致性：强一致性和最终一致性。强一致性指集群中的所有机器状态同步保持一致；最终一致性可以允许短暂的数据不一致，但数据最终会保持一致。根据CAP理论，强一致性无法实现与可用性、分区容忍性同步，而最终一致性是考虑用户体验的折中办法，也是与传统的RDBMS最大的不同。

**3.1.3 BASE思想**

BASE思想基于CAP理论，实际上是CAP理论中AP的衍伸，是对一致性进行概化处理，这是因为在越来越多的应用和实际案例中，公认为可用性和分区容忍性比一致性更需要严格设计。这些应用设计普遍倾向于降低一致性，突出可用性和数据冗余机制，也就是强调有序地将数据分散于不同节点中。

**3.1.4分布式系统**

分布式系统是建立在网络之上的软件系统，具有高度的内聚性和透明性。内聚性是指每一个数据库分布节点高度自治，有本地的数据库管理系统。透明性是指每一个数据库分布节点对用户的应用来说都是透明的，看不出是本地还是远程。在分布式数据库系统中，用户感

觉不到数据是分布的，即用户不须知道关系是否分割、有无复本、数据存于哪个站点以及事务在哪个站点上执行等。

**3.2 NoSQL基本数据类型**

数据模型是定义数据如何输入和输出的一种模型。其主要作用是为信息系统提供数据的定义和格式。数据模型是数据库系统的核心和基础，现有的数据库系统都是基于某种数据模型而建立起来的。

**3.2.1基于键-值的数据模型**

Key-Value存储是最简单的NoSQL存储，是非结构化的数据存储模式。它将以一种算法把“键”映射到相应的“值”（数据），而不关心数据的内容。开发者需要自行组织和定义“值”的数据格式，并进行解析。这种存储系统不支持任何非“键”的查询，Dynamo是最典型的键-值存储系统。

**3.2.2面向文档的数据模型**

面向文档的存储系统的代表有CouchDB和MongoDB。文档存储的数据一般用j son或类似j son的格式，存储内容是文档型的。MongoDB的文档数据以bson格式存储，CouchDB的文档数据以json格式存储，文档可以存储列表、键-值对以及层次结构复杂的文档。文档型存储的灵活性和复杂性是一把双刃剑：一方面，开发者可以任意组织文档结构；另一方面，应用层的查询需求会变得比较复杂。

**3.2.3面向列的数据模型**

这种数据模型的特点是列式存储，即每一行数据的各项被存储到不同的列中，这些列的集合称为列簇，它可以对大量行少数列进行读取和更新。代表系统有BigTable和HBase等。BigTable是Google为了有效管理海量大规模结构化数据而设计的分布式存储系统，它可以用来处理PB级的海量数据并能分布在数千台普通服务器上。HBase和Cassandra的数据模型都借鉴自Google的BigTable。

**3.2.4图结构数据模型**

图结构存储是NoSQL的另一种存储形式。基于图理论的图数据库使用节点、属性和边的概念，其中节点类似于面向对象编程中的对象概念，代表人、商业、账户等任意项的实体属性存储与节点有关的信息，例如使用Wikipedia作为节点，那么它的属性可以是网页、引用材料或者以W开头的单词，具体选择取决于实际应用；边被用来连接节点与节点或者节点与属性，表示两者之间的关系，最重要的信息存储在边上。Neo4J和HyperGraphDB是当前最流行的图结构数据库。

**4 NoSQL的分类及应用**

NoSQL是多种非关系型数据库的集合,根据应用场景的不同可将这些非关系数据库概括为以下３种类型。

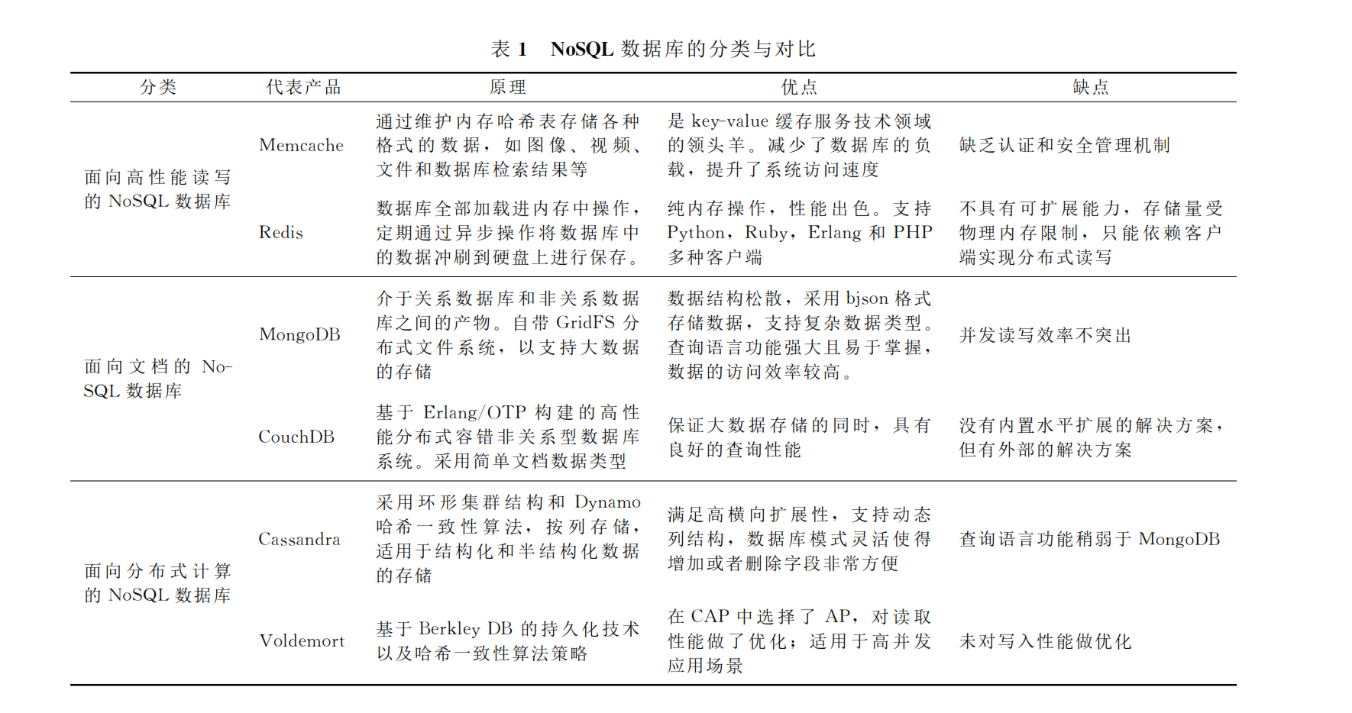
**4.1面向高性能读写的NoSQL数据库**

面向高性能读写的NoSQL数据库具有较为出色的读写性能。在一般大型网站平台的构建中，通常使用面向高性能读写的NoSQL数据库，代表产品主要有Memcached和Redis数据库。

**4.2面向文档的NoSQL数据库**

面向文档的NoSQL数据库，在保证大数据存储的基础上具有良好的查询性能。其数据一般采用json格式的文档存储。灵活的json格式使得可以对特定字段建立索引，为实现关系型数据库的部分功能提供了可能。代表产品主要有MongoDB和CouchDB数据库。

**4.3面向分布式计算的NoSQL数据库**

面向分布式计算的NoSQL数据库具有良好的横向扩展能力，要求在不停止服务的情形下，增加更多的节点；对任意１个节点写操作的数据会同步复制到其他节点上。代表产品主要有Voldemort数据库和Cassandra。

**5 关于NoSQL的思考**

NoSQL是一种与RDBMS完全不同的数据库管理系统，它的数据是松散的，易于横向扩展。克服了传统数据库在大量数据写入处理方面的暴露的缺点。总的来说，在设计上，NoSQL非常关注对数据高并发地读写和对海量数据的存储等，与关系型数据库相比，它们在架构和数据模型方面做了“减法”，而在扩展和并发等方面做了“加法”。大部分支持分布式集群，在某些特定场景下，能补充关系型数据库的缺点，但是由于NoSQL目前使用的模型还没有完善的数学基础，不像SQL这种基于关系代数和关系演算的查询结构有着坚实的数学保证，即使一个结构化的查询本身很复杂，但是它仍然能够获取满足条件的所有数据。这是NoSQL系统较为混乱的主要原因。

在传统的RDBMS的领域，分析数据，构建数据模型是存储数据前的必须工作。然而在实际中需求是随着时间的推移不断变化的，虽然在传统的关系型数据库中支持一定程度的重构造，但如果应用变化太大，重构也无济于事。NoSQL数据库打破了这种数据模型的限制，允许在一个数据单元中存入其想要的任何结构，数据单元间的联系是扁平的，一般也不受模型的限制。但在注重模式自由的同时，也需要注意管理数据的完整性。

RDBMS通常采用纵向扩展的方式解决数据库负载增加带来的性能不足，即购买性能更好的服务器代替旧服务器。这一措施针对负载缓慢增加的情况是比较有效的，然而如果负载一直增加，不可能每次都更换服务器，横向扩展的思想应运而生。所谓横向扩展就是把负载均衡的分到不同的主机上。虽然RDBMS也提供了横向扩展的功能，但对程序来说是半透明的，在横向扩展的时候，会大量的修改程序，甚至会停机，而NoSQL数据库在设计之初就考虑到了横向扩展，它对程序来说是透明的，可以随时添加节点、删除节点。相对而言，RDBMS有着强大的商业支持。Oracle、Microsoft和IBM等世界级的数据库厂商提供了完善的服务。相比而言，大部分的NoSQL数据库都是开源项目，虽然也有一些商业公司的支持，但支持的效果不可预计。如果出现故障，通常只能自己解决。

其实NoSQL数据库仅是关系数据库在某些方面（性能，扩展）的一个弥补，单从功能上讲，NoSQL的几乎所有的功能，在关系数据库上都能够满足，所以选择NoSQL的原因并不在功能上。因此我们一般会把NoSQL和关系数据库结合使用，各取所长。关系型数据库存储了数据之间存在的或潜在的关系，适合于企业数据的存储于查询；而NoSQL数据库更看重数据的存储，适合于现在的Web应用。

总的来说，NoSQL提供了一个在web2.0时代海量数据存储与查询的解决方案，打破了传统数据库存储的观念，解决了在海量存储，快速访问等问题，在分布式上也很好满足了云计算时代数据存储的要求。随着各种NoSQL的产品不断涌现，NoSQL的实际应用也在迅速增加。但新生的NoSQL不太完美，并不能完全取代RDBMS，在应用中可根据需要作为RDBMS的一个很好的补充。在数据完整性要求相对较高的企业应当以使用传统型数据库，在Web实时系统对事务完整性、读一致性要求并不高，某些时候对写一致性要求也不高时，可以使用NoSQL数据库。

**参考文献说明**

[1]吕明育,李小勇.NoSQL和关系数据库的比较分析[J].微型电脑应用,2011,27(10).

[2]陆嘉恒.大数据挑战与NoSQL数据库技术[M].

[3]陈莉莹.双锴.NoSQL数据库综述[J].中国科技论文在线.

[5]潘凡.从MySQL到MongoDB——视觉中国的NoSQL之路[J].程序员,2010(6):79-81.

[6]黄贤立.NoSQL非关系型数据库的发展及应用初探[J].福建电脑,2010,(7):30～32.

[5]范凯.NoSQL数据库综述[J].程序员,2010,(6):76～78.

[7]杨卫华.一切为了分布式——2009年Web后端技术回顾[J].程序员,2010,(2):71～73.

[8]RickyHo.NoSQL的模式[J].程序员,2010,(2):95～99.