**NoSQL 产品介绍**

NoSQL，指的是非关系型的数据库。随着互联网[web2.0](http://baike.baidu.com/view/733.htm)网站的兴起，传统的关系数据库在应付web2.0网站，特别是超大规模和高并发的SNS类型的web2.0纯动态网站已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。

**简介**

　　NoSQL(NoSQL = Not Only SQL )，意即反SQL运动，是一项全新的数据库革命性运动，早期就有人提出，发展至2009年趋势越发高涨。NoSQL的拥护者们提倡运用非关系型的数据存储，相对于目前铺天盖地的[关系型数据库](http://baike.baidu.com/view/495393.htm)运用，这一概念无疑是一种全新的思维的注入。

[编辑本段](http://baike.baidu.com/view/2677528.html?tp=7_11)**现今状况**

　　现今的[计算机体系结构](http://baike.baidu.com/view/1363593.htm)在数据存储方面要求具备庞大的水平扩展性①，而NoSQL致力于改变这一现状。目前Google的 BigTable 和Amazon 的Dynamo使用的就是NoSQL型数据库。

　　NoSQL项目的名字上看不出什么相同之处，但是，它们通常在某些方面相同：它们可以处理超大量的数据。

　　这场革命目前仍然需要等待。的确，NoSQL对大型企业来说还不是主流，但是，一两年之后很可能就会变个样子。在NoSQL运动的最新一次聚会中，来自世界各地的150人挤满了CBS Interactive的一间会议室。分享他们如何推翻缓慢而昂贵的关系数据库的暴政，怎样使用更有效和更便宜的方法来管理数据。

　　“关系型数据库给你强加了太多东西。它们要你强行修改对象数据，以满足RDBMS （relational database management system，[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm)）的需要，”在NoSQL拥护者们看来，基于NoSQL的替代方案“只是给你所需要的”。

　　1.水平扩展性(horizontal scalability)指能够连接多个软硬件的特性,这样可以将多个服务器从逻辑上看成一个实体。

[编辑本段](http://baike.baidu.com/view/2677528.html?tp=7_11)**我们为什么要使用NOSQL非关系数据库?**

　　随着互联网[web2.0](http://baike.baidu.com/view/733.htm)网站的兴起，非关系型的数据库现在成了一个极其热门的新领域，非关系数据库产品的发展非常迅速。而传统的关系数据库在应付web2.0网站，特别是超大规模和高并发的SNS类型的web2.0纯动态网站已经显得力不从心，暴露了很多难以克服的问题，例如：

　　1、High performance - 对数据库高并发读写的需求

　　web2.0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，所以基本上无法使用[动态页面静态化](http://baike.baidu.com/view/2510440.htm)技术，因此数据库并发负载非常高，往往要达到每秒上万次读写请求。关系数据库应付上万次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。其实对于普通的BBS网站，往往也存在对高并发写请求的需求。

　　2、Huge Storage - 对海量数据的高效率存储和访问的需求

　　对于大型的SNS网站，每天用户产生海量的用户动态，以国外的Friendfeed为例，一个月就达到了2.5亿条用户动态，对于关系数据库来说，在一张2.5亿条记录的表里面进行SQL查询，效率是极其低下乃至不可忍受的。再例如大型web网站的用户登录系统，例如腾讯，盛大，动辄数以亿计的帐号，关系数据库也很难应付。

　　3、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。对于很多需要提供24小时不间断服务的网站来说，对[数据库系统](http://baike.baidu.com/view/7809.htm)进行升级和扩展是非常痛苦的事情，往往需要停机维护和数据迁移，为什么数据库不能通过不断的添加服务器节点来实现扩展呢？

　　在上面提到的“三高”需求面前，关系数据库遇到了难以克服的障碍，而对于web2.0网站来说，关系数据库的很多主要特性却往往无用武之地，例如：

　　1、数据库事务一致性需求

　　很多web[实时系统](http://baike.baidu.com/view/344431.htm)并不要求严格的数据库事务，对读一致性的要求很低，有些场合对写一致性要求也不高。因此数据库事务管理成了数据库高负载下一个沉重的负担。

　　2、数据库的写实时性和读实时性需求

　　对关系数据库来说，插入一条数据之后立刻查询，是肯定可以读出来这条数据的，但是对于很多web应用来说，并不要求这么高的实时性。

　　3、对复杂的SQL查询，特别是多表关联查询的需求

　　任何大数据量的web系统，都非常忌讳多个大表的关联查询，以及复杂的数据分析类型的复杂SQL报表查询，特别是SNS类型的网站，从需求以及产品设计角度，就避免了这种情况的产生。往往更多的只是单表的主键查询，以及单表的简单条件[分页](http://baike.baidu.com/view/159980.htm)查询，SQL的功能被极大的弱化了。

　　因此，关系数据库在这些越来越多的应用场景下显得不那么合适了，为了解决这类问题的非关系数据库应运而生。

　　NoSQL 是非关系型数据存储的广义定义。它打破了长久以来关系型数据库与ACID理论[大一统](http://baike.baidu.com/view/302413.htm)的局面。NoSQL 数据存储不需要固定的表结构，通常也不存在连接操作。在大数据存取上具备关系型数据库无法比拟的性能优势。该术语在 2009 年初得到了广泛认同。

　　当今的应用体系结构需要数据存储在横向伸缩性上能够满足需求。而 NoSQL 存储就是为了实现这个需求。Google 的BigTable与Amazon的Dynamo是非常成功的商业 NoSQL 实现。一些开源的 NoSQL 体系，如Facebook 的Cassandra， Apache 的HBase，也得到了广泛认同。从这些NoSQL项目的名字上看不出什么相同之处：Hadoop、Voldemort、Dynomite，还有其它很多。

[编辑本段](http://baike.baidu.com/view/2677528.html?tp=7_11)**NoSQL与关系型数据库设计理念比较**

　　关系型数据库中的表都是存储一些格式化的数据结构，每个元组字段的组成都一样，即使不是每个元组都需要所有的字段，但数据库会为每个元组分配所有的字段，这样的结构可以便于表与表之间进行连接等操作，但从另一个角度来说它也是关系型数据库性能瓶颈的一个因素。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。

[编辑本段](http://baike.baidu.com/view/2677528.html?tp=7_11)**特点**

**它们可以处理超大量的数据**。

**它们运行在便宜的PC服务器集群上**。

　　PC集群扩充起来非常方便并且成本很低，避免了“sharding”操作的复杂性和成本。

**它们击碎了性能瓶颈**。

　　NoSQL的支持者称，通过NoSQL架构可以省去将Web或Java应用和数据转换成SQL友好格式的时间，执行速度变得更快。

　　“SQL并非适用于所有的程序代码，” 对于那些繁重的重复操作的数据，SQL值得花钱。但是当数据库结构非常简单时，SQL可能没有太大用处。

**没有过多的操作**。

　　虽然NoSQL的支持者也承认关系数据库提供了无可比拟的功能集合，而且在数据完整性上也发挥绝对稳定，他们同时也表示，企业的具体需求可能没有那么多。

**Bootstrap支持**

　　因为NoSQL项目都是开源的，因此它们缺乏供应商提供的正式支持。这一点它们与大多数开源项目一样，不得不从社区中寻求支持。

[编辑本段](http://baike.baidu.com/view/2677528.html?tp=7_11)**缺点**

　　但是一些人承认，没有正式的官方支持，万一出了差错会是可怕的，至少很多管理人员是这样看。

　　“我们确实需要做一些说服工作，但基本在他们看到我们的第一个原型运行良好之后，我们就能够说服他们，这是条正确的道路。”

　　此外，nosql并未形成一定标准，各种产品层出不穷，内部混乱，各种项目还需时间来检验

[编辑本段](http://baike.baidu.com/view/2677528.html?tp=7_11)**NoSQL数据库开源软件**

**Membase**

　　Membase 是 NoSQL 家族的一个新的重量级的成员。Membase是开源项目，源代码采用了Apache2.0的使用许可。该项目托管在GitHub.Source tarballs上，目前可以 下载beta版本的Linux二进制包。该产品主要是由North Scale的memcached核心团队成员开发完成，其中还包括Zynga和NHN这两个主要贡献者的工程师，这两个组织都是很大的在线游戏和社区网络空间的供应商。

　　Membase容易安装、操作，可以从单节点方便的扩展到集群，而且为memcached（有线协议的兼容性）实现了即插即用功能，在应用方面为开 发者和经营者提供了一个比较低的门槛。做为缓存解决方案，Memcached已经在不同类型的领域（特别是大容量的Web应用）有了广泛的使用，其中 Memcached的部分基础代码被直接应用到了Membase服务器的前端。

　　通过兼容多种编程语言和框架，Membase具备了很好的复用性。在安装和配置方面，Membase提供了有效的图形化界面和编程接口，包括可配置 的告警信息。

　　Membase的目标是提供对外的线性扩展能力，包括为了增加集群容量，可以针对统一的节点进行复制。 另外，对存储的数据进行再分配仍然是必要的。

　　这方面的一个有趣的特性是NoSQL解决方案所承诺的可预测的性能，类准确性的延迟和吞吐量。通过如下方式可以获得上面提到的特性：

　　◆ 自动将在线数据迁移到低延迟的存储介质的技术（内存，固态硬盘，磁盘）

　　◆ 可选的写操作一一异步，同步（基于复制，持久化）

　　◆ 反向通道再平衡[未来考虑支持]

　　◆ 多线程低锁争用

　　◆ 尽可能使用异步处理

　　◆ 自动实现重复数据删除

　　◆ 动态再平衡现有集群

　　◆ 通过把数据复制到多个集群单元和支持快速失败转移来提供系统的高可用性。

**MongoDB**

　　MongoDB是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似json的bjson格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。它的特点是高性能、易部署、易使用，存储数据非常方便。

**主要功能特性**：

　　◆ 面向集合存储，易存储对象类型的数据

　　“面向集合”（Collenction-Orented），意思是数据被分组存储在数据集中，被称为一个集合（Collenction)。每个 集合在数据库中都有一个唯一的标识名，并且可以包含无限数目的文档。集合的概念类似关系型数据库（RDBMS）里的表（table），不同的是它不需要定 义任何模式（schema)。

　　◆ 模式自由

　　模式自由（schema-free)，意味着对于存储在mongodb数据库中的文件，我们不需要知道它的任何结构定义。如果需要的话，你完全可以把不同结构的文件存储在同一个数据库里。

　　◆支持动态查询

　　◆支持完全索引，包含内部对象

　　◆支持查询

　　◆支持复制和故障恢复

　　◆使用高效的二进制数据存储，包括大型对象（如视频等）

　　◆自动处理碎片，以支持云计算层次的扩展性

　　◆支持RUBY，PYTHON，JAVA，C++，PHP等多种语言

　　◆文件存储格式为BSON（一种JSON的扩展）

　　BSON（Binary Serialized dOcument Format）存储形式是指：存储在集合中的文档，被存储为键-值对的形式。键用于唯一标识一个文档，为字符串类型，而值则可以是各中复杂的文件类型。

　　◆可通过网络访问

　　MongoDB服务端可运行在Linux、Windows或OS X平台，支持32位和64位应用，默认端口为27017。推荐运行在64位平台，因为MongoDB在32位模式运行时支持的最大文件尺寸为2GB。

　　MongoDB把数据存储在文件中（默认路径为：/data/db），为提高效率使用内存映射文件进行管理。

**Hypertable**

　　Hypertable是一个开源、高性能、可伸缩的数据库，它采用与Google的Bigtable相似的模型。在过去数年中，Google为在PC集群 上运行的可伸缩计算基础设施设计建造了三个关键部分。第一个关键的基础设施是Google File System（GFS），这是一个高可用的文件系统，提供了一个全局的命名空间。它通过跨机器（和跨机架）的文件数据复制来达到高可用性，并因此免受传统 文件存储系统无法避免的许多失败的影响，比如电源、内存和网络端口等失败。第二个基础设施是名为Map-Reduce的计算框架，它与GFS紧密协作，帮 助处理收集到的海量数据。第三个基础设施是Bigtable，它是传统数据库的替代。Bigtable让你可以通过一些主键来组织海量数据，并实现高效的 查询。Hypertable是Bigtable的一个开源实现，并且根据我们的想法进行了一些改进。

**Apache Cassandra**

　　Apache Cassandra是一套开源分布式Key-Value存储系统。它最初由Facebook开发，用于储存特别大的数据。Facebook目前在使用此系统。

**主要特性**：

　　◆ 分布式

　　◆ 基于column的结构化

　　◆ 高伸展性

　　Cassandra的主要特点就是它不是一个数据库，而是由一堆数据库节点共同构成的一个分布式网络服务，对Cassandra 的一个写操作，会被复制到其他节点上去，对Cassandra的读操作，也会被路由到某个节点上面去读取。对于一个Cassandra群集来说，扩展性能 是比较简单的事情，只管在群集里面添加节点就可以了。

　　Cassandra是一个混合型的非关系的数据库，类似于Google的BigTable。其主要功能比 Dynomite（分布式的Key-Value存 储系统）更丰富，但支持度却不如文档存储MongoDB（介于关系数据库和非关系数据库之间的开源产品，是非关系数据库当中功能最丰富，最像关系数据库 的。支持的数据结构非常松散，是类似json的bjson格式，因此可以存储比较复杂的数据类型。）Cassandra最初由Facebook开发，后转变成了开源项目。它是一个网络社交云计算方面理想的数据库。以Amazon专有的完全分布式的Dynamo为基础，结合了Google BigTable基于列族（Column Family）的数据模型。P2P去中心化的存储。很多方面都可以称之为Dynamo 2.0。

**和其他数据库比较，其突出特点是：**

　　◆ 模式灵活 ：使用Cassandra，像文档存储，你不必提前解决记录中的字段。你可以在系统运行时随意的添加或移除字段。这是一个惊人的效率提升，特别是在大型部 署上。

　　◆ 真正的可扩展性 ：Cassandra是纯粹意义上的水平扩展。为给集群添加更多容量，可以指向另一台电脑。你不必重启任何进程，改变应用查询，或手动迁移任何数据。

　　◆ 多数据中心识别 ：你可以调整你的节点布局来避免某一个数据中心起火，一个备用的数据中心将至少有每条记录的完全复制。

　　◆ 范围查询 ：如果你不喜欢全部的键值查询，则可以设置键的范围来查询。

　　◆ 列表数据结构 ：在混合模式可以将超级列添加到5维。对于每个用户的索引，这是非常方便的。

　　◆ 分布式写操作 ：有可以在任何地方任何时间集中读或写任何数据。并且不会有任何单点失败。