[1-2]  NoSQL，泛指非关系型的数据库。随着互联网[web2.0](http://baike.baidu.com/view/733.htm" \t "http://baike.baidu.com/_blank)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](http://baike.baidu.com/subview/8258/5896174.htm" \t "http://baike.baidu.com/_blank)类型的web2.0纯[动态网](http://baike.baidu.com/view/528572.htm" \t "http://baike.baidu.com/_blank)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。NoSQL数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战，尤其是大数据应用难题。

虽然NoSQL流行语火起来才短短一年的时间，但是不可否认，现在已经开始了第二代运动。尽管早期的堆栈代码只能算是一种实验，然而现在的系统已经更加的成熟、稳定。不过现在也面临着一个严酷的事实：技术越来越成熟——以至于原来很好的NoSQL数据存储不得不进行重写，也有少数人认为这就是所谓的2.0版本。这里列出一些比较知名的工具，可以为大数据建立快速、可扩展的存储库。

**中文名**

非关系型数据库

**外文名**

NoSQL=Not Only [SQL](http://baike.baidu.com/view/34.htm" \t "http://baike.baidu.com/_blank)

**全    称**

Not Only [SQL](http://baike.baidu.com/view/34.htm" \t "http://baike.baidu.com/_blank)

**类    别**

非关系型的数据库

**应用领域**

[计算机](http://baike.baidu.com/view/3314.htm" \t "http://baike.baidu.com/_blank)、[软件](http://baike.baidu.com/subview/37/6030295.htm" \t "http://baike.baidu.com/_blank)、[数据库](http://baike.baidu.com/view/1088.htm" \t "http://baike.baidu.com/_blank)

## 目录

1. 1 [基本含义](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "1)
2. 2 [NoSQL数据库的四大分类](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "2)
3. 3 [NoSQL数据库的四大分类表格分析](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "3)
4. 4 [共同特征](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "4)
5. 5 [适用场景](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "5)
6. 6 [发展现状](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "6)
7. 7 [挑战](http://baike.baidu.com/link?url=ieqeDkUhKwY6XAdnJAXKOgWH8BpKymf9EoKxz2h3xM4ROwqxVvIlbYRt0pP_z9hOCTa32p37BPdJeVG_hwh0Aq" \l "7)

## 基本含义

NoSQL(NoSQL = Not Only SQL )，意即“不仅仅是[SQL](http://baike.baidu.com/view/34.htm" \t "http://baike.baidu.com/_blank)”，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，相对于铺天盖地的[关系型数据库](http://baike.baidu.com/view/495393.htm" \t "http://baike.baidu.com/_blank)运用，这一概念无疑是一种全新的思维的注入。

## NoSQL数据库的四大分类

**键值(***[Key-Value](http://baike.baidu.com/view/6728383.htm" \t "http://baike.baidu.com/_blank)***)存储[数据库](http://baike.baidu.com/view/1088.htm" \t "http://baike.baidu.com/_blank)**

这一类数据库主要会使用到一个[哈希表](http://baike.baidu.com/view/329976.htm" \t "http://baike.baidu.com/_blank)，这个表中有一个特定的键和一个指针指向特定的数据。Key/value模型对于IT系统来说的优势在于简单、易部署。但是如果[DBA](http://baike.baidu.com/subview/67156/5112091.htm" \t "http://baike.baidu.com/_blank)只对部分值进行查询或更新的时候，Key/value就显得效率低下了。[3]  举例如：Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB.

**列存储数据库。**

这部分数据库通常是用来应对分布式存储的海量数据。键仍然存在，但是它们的特点是指向了多个列。这些列是由列家族来安排的。如：Cassandra, HBase, Riak.

**文档型数据库**

文档型数据库的灵感是来自于Lotus Notes办公软件的，而且它同第一种键值存储相类似。该类型的数据模型是版本化的文档，半结构化的文档以特定的格式存储，比如JSON。文档型数据库可 以看作是键值数据库的升级版，允许之间嵌套键值。而且文档型数据库比键值数据库的查询效率更高。如：CouchDB, MongoDb. 国内也有文档型数据库SequoiaDB，已经开源。

**图形(Graph)数据库**

图形结构的数据库同其他行列以及刚性结构的SQL数据库不同，它是使用灵活的图形模型，并且能够扩展到多个服务器上。NoSQL数据库没有标准的查询语言(SQL)，因此进行数据库查询需要制定数据模型。许多NoSQL数据库都有REST式的数据接口或者查询API。[2]  如：Neo4J, InfoGrid, Infinite Graph.

因此，我们总结NoSQL数据库在以下的这几种情况下比较适用：1、数据模型比较简单；2、需要灵活性更强的IT系统；3、对数据库性能要求较高；4、不需要高度的数据一致性；5、对于给定key，比较容易映射复杂值的环境。

## NoSQL数据库的四大分类表格分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **分类** | **Examples举例** | **典型应用场景** | **数据模型** | **优点** | **缺点** |
| **键值（key-value）[3]** | Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB | 内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等等。[3] | Key 指向 Value 的键值对，通常用hash table来实现[3] | 查找速度快 | 数据无结构化，通常只被当作字符串或者二进制数据[3] |
| **列存储数据库[3]** | Cassandra, HBase, Riak | 分布式的文件系统 | 以列簇式存储，将同一列数据存在一起 | 查找速度快，可扩展性强，更容易进行分布式扩展 | 功能相对局限 |
| **文档型数据库[3]** | CouchDB, MongoDb | Web应用（与Key-Value类似，Value是结构化的，不同的是数据库能够了解Value的内容） | Key-Value对应的键值对，Value为结构化数据 | 数据结构要求不严格，表结构可变，不需要像关系型数据库一样需要预先定义表结构 | 查询性能不高，而且缺乏统一的查询语法。 |
| **图形(Graph)数据库[3]** | Neo4J, InfoGrid, Infinite Graph | 社交网络，推荐系统等。专注于构建关系图谱 | 图结构 | 利用图结构相关算法。比如最短路径寻址，N度关系查找等 | 很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群方案。[3] |

## 共同特征

对于NoSQL并没有一个明确的范围和定义，但是他们都普遍存在下面一些共同特征：

不需要预定义模式：不需要事先定义数据模式，预定义表结构。数据中的每条记录都可能有不同的属性和格式。当插入数据时，并不需要预先定义它们的模式。

无共享架构：相对于将所有数据存储的存储区域网络中的全共享架构。NoSQL往往将数据划分后存储在各个本地服务器上。因为从本地磁盘读取数据的性能往往好于通过网络传输读取数据的性能，从而提高了系统的性能。

弹性可扩展：可以在系统运行的时候，动态增加或者删除结点。不需要停机维护，数据可以自动迁移。

分区：相对于将数据存放于同一个节点，NoSQL数据库需要将数据进行分区，将记录分散在多个节点上面。并且通常分区的同时还要做复制。这样既提高了并行性能，又能保证没有单点失效的问题。

异步复制：和RAID存储系统不同的是，NoSQL中的复制，往往是基于日志的异步复制。这样，数据就可以尽快地写入一个节点，而不会被网络传输引起迟延。缺点是并不总是能保证一致性，这样的方式在出现故障的时候，可能会丢失少量的数据。

BASE：相对于事务严格的ACID特性，NoSQL数据库保证的是BASE特性。BASE是最终一致性和软事务。

NoSQL数据库并没有一个统一的架构，两种NoSQL数据库之间的不同，甚至远远超过两种关系型数据库的不同。可以说，NoSQL各有所长，成功的NoSQL必然特别适用于某些场合或者某些应用，在这些场合中会远远胜过关系型数据库和其他的NoSQL。

## 适用场景

NoSQL数据库在以下的这几种情况下比较适用：1、数据模型比较简单；2、需要灵活性更强的IT系统；3、对数据库性能要求较高；4、不需要高度的数据一致性；5、对于给定key，比较容易映射复杂值的环境。

## 发展现状

[计算机体系结构](http://baike.baidu.com/view/1363593.htm" \t "http://baike.baidu.com/_blank)在数据存储方面要求具备庞大的水平扩展性，而NoSQL致力于改变这一现状。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的经验，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](http://baike.baidu.com/view/495393.htm" \t "http://baike.baidu.com/_blank)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" \t "http://baike.baidu.com/_blank)）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”。

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。

## 挑战

尽管大多数NoSQL数据存储系统都已被部署于实际应用中，但归纳其研究现状，还有许多挑战性问题。

已有key-value数据库产品大多是面向特定应用自治构建的，缺乏通用性；

已有产品支持的功能有限（不支持事务特性），导致其应用具有一定的局限性；

已有一些研究成果和改进的NoSQL数据存储系统，但它们都是针对不同应用需求而提出的相应解决方案，如支持组内事务特性、弹性事务等，很少从全局考虑系统的通用性，也没有形成系列化的研究成果；

缺乏类似关系数据库所具有的强有力的理论(如armstrong公理系统)、技术（如成熟的基于启发式的优化策略、两段封锁协议等）、标准规范（如SQL语言）的支持。

目前，HBase数据库时安全特性最完善的NoSQL数据库产品之一，而其他的NoSQL数据库多数没有提供内建的安全机制，但随着NoSQL的发展，越来越多的人开始意识到安全的重要，部分NoSQL产品逐渐开始提供一些安全方面的支持。

随着云计算、互联网等技术的发展，大数据广泛存在，同时也呈现出了许多云环境下的新型应用，如社交网络网、移动服务、协作编辑等。这些新型应用对海量数据管理或称云数据管理系统也提出了新的需求，如事务的支持、系统的弹性等。同时云计算时代海量数据管理系统的设计目标为可扩展性、弹性、容错性、自管理性和“强一致性”。目前，已有系统通过支持可随意增减节点来满足可扩展性；通过副本策略保证系统的容错性；基于监测的状态消息协调实现系统的自管理性。“弹性”的目标是满足Pay-per-use 模型，以提高系统资源的利用率。该特性是已有典型NoSQL数据库系统所不完善的，但却是云系统应具有的典型特点；“强一致性”主要是新应用的需求。[4]