

NoSQL 数据库

pwwlin1992@gmail.com

2017 年 11 月 6 日

Contents

1 NoSQL 简介	1
2 NoSQL 兴起的原因	2
2.1 RDBMS 无法满足 Web 2.0 的需求	2
2.2 NoSQL 与 RDBMS 的对比	2
3 NoSQL 的类型	2
3.1 键值数据库	2

1 NoSQL 简介

灵活的可扩展性

RDBMS 很难实现“横向扩展”，在面对数据库负载大规模增加时，往往需要通过**升级硬件**来实现“纵向扩展”。

横向扩展：一个 DBMS 不够用，则用两个，三个。

纵向扩展：一个 DBMS 不够用，则升级硬件，使其更快。

NoSQL 在设计之初就是为了满足“横向扩展”的需求，因此天生具备良好的水平扩展能力。

灵活的数据模型

摆脱 RDBMS 的各种束缚条件，采用键/值、列族等非关系模型，允许在一个数据元素里存储不同类型的数据。

与云计算紧密融合

云计算具有很好的水平扩展能力，可以根据资源使用情况进行自由伸缩，各种资源可以动态加入或退出。NoSQL 数据库可以凭借自身良好的横向扩展能力，充分自由利用云计算基础设施，很好地融入到云计算环境中，构建基于 NoSQL 的云数据库服务。

2 NoSQL 兴起的原因

2.1 RDBMS 无法满足 Web 2.0 的需求

无法满足海量数据的管理需求

对于 RDBMS 来说，在一张 10 亿条记录的表里进行 SQL 查询，效率极其低下。

无法满足数据高并发的需求

在 Web 1.0 时代，通常采用动态页面静态化技术，事先访问数据库生成静态页面供浏览者访问，从而保证在大规模用户访问时，也能够获得较好的实时响应性能。但是在 Web 2.0 时代，各种用户都在不断地发生更新，如：购物记录、搜索记录等信息都需要实时更新，动态页面静态化技术基本无用武之地，所有信息都需要动态实时生成，这就会导致高并发的数据库访问，可能产生每秒上万次的读写请求，对于很多 RDBMS 而言都是难以承受的。

无法满足高可扩展性和高可用性的需求

在 Web 2.0 时代，不知名的网站可能一夜爆红，知名网站也有可能因为发布了热门信息而引来大量用户围观，这些都会导致对数据库读写负荷的急剧增加，需要数据库能够在短时间内迅速提升性能应对突发需求。而 RDBMS 难以水平扩展，无法像网页服务器和应用服务器那样简单通过添加更多硬件和服务节点来扩展性能和负载能力。

2.2 NoSQL 与 RDBMS 的对比

NoSQL 与 RDBMS 各有优劣，NoSQL 缺乏数学理论基础，复杂查询性能不高，不能实现事务强一致性，很难实现数据完整性。在实际应用中，公司一般会采用两者混合使用的方式，如：对于购物车这种临时性数据，采用键值存储会更加高效，而当前产品和订单信息则适合存放在关系数据库中，大量的历史订单信息则适合保存在类似 MongoDB 的文档数据库中。

3 NoSQL 的类型

3.1 键值数据库

键值数据库使用哈希表，表中的 key 可以用来定位 value，即存储和检索具体的 value。value 对数据库而言是不可见的，不能对 value 进行索引和查询，只能通过 key 进行查询，value 可以存储任意类型的数据。

在存在大量写操作的情况下，键值数据库可以比关系数据库取得明显更好的性能。因为 RDBMS 需要建立索引来加速查询，当存在大量写操作时，索引会发生频繁更新，由此会产生高昂的索引维护代价。

Figure 1: NoSQL 与 RDBMS 的对比

比较标准	RDBMS	NoSQL	备注
数据库原理	完全支持	部分支持	RDBMS有关系代数理论作为基础 NoSQL没有统一的理论基础
数据规模	大	超大	RDBMS很难实现横向扩展，纵向扩展的空间也比较有限，性能会随着数据规模的增大而降低 NoSQL可以很容易通过添加更多设备来支持更大规模的数据
数据库模式	固定	灵活	RDBMS需要定义数据库模式，严格遵守数据定义和相关约束条件 NoSQL不存在数据库模式，可以自由灵活定义并存储各种不同类型的数据
查询效率	快	可以实现高效的简单查询，但是不具备高度结构化查询等特性，复杂查询的性能不尽人意	RDBMS借助于索引机制可以实现快速查询（包括记录查询和范围查询） 很多NoSQL数据库没有面向复杂查询的索引，虽然NoSQL可以使用MapReduce来加速查询，但是，在复杂查询方面的性能仍然不如RDBMS
比较标准	RDBMS	NoSQL	备注
一致性	强一致性	弱一致性	RDBMS严格遵守事务ACID模型，可以保证事务强一致性 很多NoSQL数据库放松了对事务ACID四性的要求，而是遵守BASE模型，只能保证最终一致性
数据完整性	容易实现	很难实现	任何一个RDBMS都可以很容易实现数据完整性，比如通过主键或者非空约束来实现实体完整性，通过主键、外键来实现参照完整性，通过约束或者触发器来实现用户自定义完整性 但是，在NoSQL数据库却无法实现
扩展性	一般	好	RDBMS很难实现横向扩展，纵向扩展的空间也比较有限 NoSQL在设计之初就充分考虑了横向扩展的需求，可以很容易通过添加廉价设备实现扩展
可用性	好	很好	RDBMS在任何时候都以保证数据一致性为优先目标，其次才是优化系统性能，随着数据规模的增大，RDBMS为了保证严格的一致性，只能提供相对较弱的可用性 大多数NoSQL都能提供较高的可用性
比较标准	RDBMS	NoSQL	备注
标准化	是	否	RDBMS已经标准化（SQL） NoSQL还没有行业标准，不同的NoSQL数据库都有自己的查询语言，很难规范应用程序接口 StoneBraker认为：NoSQL缺乏统一查询语言，将会拖慢NoSQL发展
技术支持	高	低	RDBMS经过几十年的发展，已经非常成熟，Oracle等大型厂商都可以提供很好的技术支持 NoSQL在技术支持方面仍然处于起步阶段，还不成熟，缺乏有力的技术支持
可维护性	复杂	复杂	RDBMS需要专门的数据库管理员(DBA)维护 NoSQL数据库虽然没有DBMS复杂，也难以维护

Key_1	Value_1
Key_2	Value_2
Key_3	Value_1
Key_4	Value_3
Key_5	Value_2
Key_6	Value_1
Key_7	Value_4
Key_8	Value_3

Figure 2: 键值数据库

相关产品	Redis、Riak、SimpleDB、Chordless、Scalaris、Memcached
数据模型	键/值对 键是一个字符串对象 值可以是任意类型的数据，比如整型、字符型、数组、列表、集合等
典型应用	涉及频繁读写、拥有简单数据模型的应用 内容缓存，比如会话、配置文件、参数、购物车等 存储配置和用户数据信息的移动应用
优点	扩展性好，灵活性好，大量写操作时性能高
缺点	无法存储结构化信息，条件查询效率较低
不适用情形	不是通过键而是通过值来查：键值数据库根本没有通过值查询的途径 需要存储数据之间的关系：在键值数据库中，不能通过两个或两个以上的键来关联数据 需要事务的支持：在一些键值数据库中，产生故障时，不可以回滚
使用者	百度云数据库（Redis）、GitHub（Riak）、BestBuy（Riak）、Twitter（Redis和Memcached）、StackOverFlow（Redis）、Instagram（Redis）、Youtube（Memcached）、Wikipedia（Memcached）

Figure 3: 键值数据库