

# Nosql 数据库与关系型数据库对比读后感

19301112-蔡明宇

本文主要分析了 NoSQL 数据库与传统关系型数据库在 Web2.0 时代下的不同应用场景下各自的优缺点，同时也将二者做了简要的对比分析，介绍了非关系型数据库的多种数据存储模式的分类以及不同的应用场景。

当前，社会广泛应用的数据库系统分为两大类：**RDBMS**（关系型数据库）与 **No SQL**（非关系型数据库）。关系型数据库自上世纪 80 年代初期就被提出并得到应用，至今广泛应用于社会多个领域方面，它是基于关系模型来存储数据的数据库。关系型数据库，是指采用了关系模型来组织数据的数据库，其以行和列的形式存储数据，以便于用户理解，关系型数据库这一系列的行和列被称为表，一组表组成了数据库。用户通过查询来检索数据库中的数据，而查询是一个用于限定数据库中某些区域的执行代码。关系模型可以简单理解为二维表格模型，而一个关系型数据库就是由二维表及其之间的关系组成的一个数据组织。而非关系型数据库，**NoSQL**，则泛指非关系型的数据库。随着互联网 web2.0 网站的兴起，传统的关系数据库在处理 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，出现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，特别是大数据应用难题。

目前，较为主流的传统关系型数据库有 Oracle、DB2、My SQL、Microsoft SQL Server、Microsoft Access 等，它们在处理数据量不高，单一类型数据的时候是可以正常的进行存储管理，但当数据量增加到一定数量级，数据类型增多的时候，若未进行及时有效的存储前规划，则传统关系型数据库在存储数据时就会出现并发度不高，访问查询效率较低以及可扩展性不足等一系列问题，所以，在某些场景下，非关系型数据库针对不同的数据类型以及不同的应有场景也有不同的数据库存储模式。因此 **No SQL** 凭借自身优点特性在当代社会与工作中越来越受欢迎。接下来，具体分析这两种类型的数据库。

## 1. 关系型数据库

关系型数据库中的表结构是一种基于关系模型的二维表格形式，其模型形式基于行和列来存储管理数据。一个关系型数据库组成包括多张表以及多表之间的关系。用户可以通过 SQL 语句来进行查询访问数据库中的数据。传统的关系型数据库采用表格的储存方式，数据以行和列的方式进行存储，要读取和查询都十分方便。关系型数据库的存储结构也很简单，关系型数据库按照结构化的方法存储数据，每个数据表都必须对各个字段定义好（也就是先定义好表的结构），再根据表的结构存入数据，这样做的好处就是由于数据的形式和内容在存入数据之前就已经定义好了，所以整个数据表的可靠性和稳定性都比较高，但带来的问题就是一旦存入数据后，如果需要修改数据表的结构就会十分困难。采用了结构化查询语言（即 SQL）来对数据库进行查询，SQL 早已获得了各个数据库厂商的支持，成为数据库行业的标准，它能够支持数据库的 CRUD（增加，查询，更新，删除）操作，具有非常强大的功能，SQL 可以采用类似索引的方法来加快查询操作。强调 ACID 规则（原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）），可以满足对事务性要求较高或者需要进行复杂数据查询的数据操作，而且可以充分满足数据库操作的高性能和操作稳定性的要求。并且关系型数据库十分强调数据的强一致性，对于事务的操作有很好的支持。关系型数据库可以控制事务原子性细粒度，并且一旦操作有误或者有需要，可以马上回滚事务。

## 2. 非关系型数据库

主要分为以下的几种类型：

### （1）键值(Key-Value)存储数据库

这一类数据库主要会使用到一个哈希表,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。但是如果数据库管理员 (DBA) 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。举例如: Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB。

#### (2) 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。

#### (3) 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,在处理网页等复杂数据时,文档型数据库比传统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。

#### (4) 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。如: Neo4J, InfoGrid, Infinite Graph。

### 3. NoSQL 的特点

No SQL 数据库具有以下特点: 易扩展性: No SQL 是非关系型数据库,其包含多种数据存储的数据库设计模式,但所有的设计模式有一个共性特征是摒弃传统关系型数据库的关系特性后,其存储数据间不存在关系,此特征优势在于提升了在架构层面上的可扩展性能力。海量数据处理,相比于关系型数据库, No SQL 具有高性能并发读写的特点, No SQL 数据库的 Cache 是具有细粒度特性,因此从该层面来讲 No SQL 的性能就要高出很多,由于其无关系特性, No SQL 数据库在大数据处理场景中,其读写性能具有非常高的效率。同时,具有灵活的数据模型和高可用性, No SQL 数据库中的某些模型一方面可以在低影响性能的场景下,能快速便捷地实现高可用性,另一方面能够采用复制相应模型的方式实现高可用性。

### 4. 二者相比的优缺点

关系型数据库与 No SQL 数据库在使用情景下有着很大的区别,不同的应用场景对应着需要不同的数据库模式,只有彼此相互正确对应才能更好的发挥数据库管理的优势。

关系型数据库首先是便于维护,关系型数据库中的表结构是二维数组的形式,格式相同,方便进行 SQL 语句操作。而 No SQL 不支持 SQL 结构化查询语言,如果无法支持 SQL 这样的工业标准,将会导致用户学习与场景应用的成本增加。其次关系型数据库支持复杂操作,它可以实现一个表以及多个表之间的复杂查询操作,使初学者容易上手学习 SQL 语句。而 No SQL 提供特性支持的丰富度不足,目前较为流行的非关系型数据库都无法支持事务处理,且能提供的功能服务有限,相比较一些关系型数据库能够提供一些附加功能。但 NoSQL 的优点也是关系型数据库难以企及的,数据存储结构灵活,传统关系型数据库由于表结构相对固定,只能通过数据定义语言 (DDL 语句) 进行表结构的修改,且可扩展性较低。而 No SQL 数据库有多样的存储机制,一方面数据格式灵活度高无固定表结构,另一方面可扩展性高。比如文档存储模式, Key-Value 键值对存储模式,图存储模式。同时,可扩展性高,传统关系型数据库由于其自身表结构固定的特点不易于对数据进行分片,对数据的横向扩展很难支持,但 No SQL 数据库支持数据横向扩展,同时也支持对大数据量以及多类型数据的存储管理,最后就是,支持数据的最终一致性: No SQL 数据库是基于 CAP 模型的,同时也是支持 BASE 分布式理论,目的是达到数据的最终一致性。

由于关系型数据库与 No SQL 数据库使用的场景不同，因此两者在实际生活工作中是一种互相补充的关系。当关系型数据库在某些场景中不能有效的应用时，这时就可以依靠关系型数据库应用在此场景中。二者在生活应用中相互补充，我们各取所需，发挥两种数据库的优点进行使用，才能更充分地发挥出二者的作用。