

本节我们讨论什么是 NoSQL 数据库。

大数据计算系统可归纳为三部分:

数据存储系统，数据处理系统，数据应用系统

数据存储架构是大数据计算的基础。

在数据存储系统中，有 4 个部分来完成不同的任务，

即数据收集与建模、分布式文件系统、分布式数据库/数据仓库和统一数据访问接口。

我们学习了数据收集和建模，分布式文件系统，现在我们看数据库。

数据库通常包括关系数据库，如 Oracle, SQL server, NoSQL 数据库，如 MongoDB, Hbase 等。

构建于文件系统之上的分布式数据库以及数据仓库，不仅实现数据的存储管理，更重要的是对上层计算引擎和应用软件的数据快速查询功能 and 数据分析服务的支持。

目前支持大数据应用的数据库产品种类繁多，其存储结构与采用的技术也各不相同。关系型数据库 RDBMS 以其数据一致性好、事务机制完善、查询语言标准化、查询效率高，技术成熟的特点一直以来是数据存储管理系统的主流，其代表产品有 Oracle，MS SQL Server 和 My SQL。在面对超大规模数据量、大量非结构化或半结构化数据的大数据计算问题时，关系型数据库面临如下挑战：

1 大数据超大规模（PB 量级）的存储管理要求系统具有很好的弹性，在分布式环境中可方便地扩展，但传统关系型数据库对数据一致性完整性的强调和集中式部署方式使得其扩展性较差，难以适应数据量爆炸式增长的场景

2 RDBMS 基于严格定义的键索引和数据表存储模式适合结构化数据的存储管理，并能够提供高效的基于 SQL 的查询机制。但对于非结构化或半结构化数据，RDMS 就难以处理，查询效率也大大降低

3 大数据计算处理要求存储结构能够很好地支持上层计算模型。

比如 Mapreduce 计算模型采用的是分治策略（Divide-and-Conquer），

即先把一个大数据集划分为多个子集，然后每个子集运行 map 程序进行处理。在完成子集的处理之后，在运行 reduce 程序完成计算结果的汇总，这就要求下层的数据存储结构能够支持这种数据集（或者数据表结构）的划分和融汇功能。

关系型数据库由于严格的数据一致性和完整性要求，难以对数据表进行这种分割处理，因此很难支持大数据计算的各种计算模型

让我们看一段视频，关于 NoSQL 数据库是如何工作的

从常用数据库，存储格式，存储规范，存储扩展，查询方式，事务，性能和成本的方面对 NoSQL 数据库和关系型数据库进行对比。

NoSQL 常用的数据库有，Hbase，MongoDB，Redis，关系型数据库常用数据库有 oracle,DB2,MySQL

存储格式，NoSQL 数据库的存储格式有，文档，键值对，图结构数据库，关系型数据库有以行和列为主的表格式

存储扩展，NoSQL 支持横向扩展，分布式的方式，关系型数据库支持纵向扩展，横向扩展能力有限

查询方式，NoSQL 支持非结构化查询，关系型数据库只支持结构化查询

对事务的支持，NoSQL 不支持事务一致性，关系型数据库支持事务一致性

NoSQL 读写性能高，关系型数据库读写性能差

NoSQL 简单易于部署，开源成本低，关系型数据库成本高，但是技术支持比较好。

近年来随着大数据处理而兴起的基于分布式文件系统的非关系型数据库，(not only SQL, NoSQL) 则以其扩展性好，能有效处理非机构化和半结构化数据，支持高并发计算模型的特点，在互联网、医疗卫生、

电子商务等领域得到了广泛应用。与传统的关系型数据库比较，这类 NoSQL 数据库存在如下共同特征

1 不需要预定义数据格式

不需要预先定义严格的数据表结构，数据的每条记录都可能有不同的属性和格式，当插入数据时，并不需要预先定义他们的格式。

2 无共享架构

NoSQL 数据库往往将数据集划分后存储在各个本地服务器上，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，从而提高了系统读写速度。

3 弹性可扩展

可以在系统运行时候动态增加或者删除节点而不需要停机维护，数据可以自动迁移。

4 数据分区

相对于将数据存放在同一个节点，NoSQL 数据库则是将数据进行分区，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

5 异步复制

NoSQL 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，而不是被网络传输引起迟延，缺点是并不总是能保证一致性。NoSQL 数据库提供的是基于 BASE 原则的最终一致性。

同步复制的优点是：一旦向用户确认，从节点可以明确保证完成了与主节点的更新同步，数据已经处于最新版本。万一主节点发生故障，总是可以在从节点继续访问最新数据。

缺点则是：如果同步的从节点无法完成确认（例如：由于从节点发生崩溃，或者网络故障，或任何其他原因），写入就不能视为成功。主节点会阻塞其后所有的写操作，直到同步副本确认已完成。

因此，把所有从节点都配置为同步复制有些不切实际。因为这样的话，任何一个同步节点的中断都会导致整个系统更新停滞不前。

实践中，如果数据库启用了同步复制，通常意味着其中某一个从节点是同步的，而其他节点这是异步模式。万一同步的从节点变得不可用或性能下降，则将另一个异步的从节点提升为同步模式。

这样可以保证至少有两个节点（即主节点和一个同步从节点）拥有最新的数据副本。这种配置有时候也叫做半同步。

7

shared nothing architecture 是一种分布式计算架构，这种架构中不存在集中存储的状态，整个系统中没有资源竞争，这种架构具有非常强的扩张性，在 web 应用中广泛使用。

对比

Shared-nothing、shared-memory、shared-disk 是并行系统最常使用的模式。

shared-memory：多个 cpu 共享同一块内存，cpu 之间通过内部通讯机制进行通讯

shared-disk：每一个 cpu 使用自己的私有内存区域，通过内部通讯机制直接访问所有磁盘系统。

和 shared-memory、shared-disk 相比，shared-nothing 优势明显，在针对多用户并行访问的时候，通过横向扩充资源，能够减少 response time。

8

数据库可以分为传统业务系统数据库，NoSQL 数据库和 NewSQL 数据库

NewSQL 是对各种新的可扩展/高性能数据库的简称，这类数据库不仅具有 NoSQL 对海量数据的存储管理能力，还保持了传统数据库支持 ACID 和 SQL 等特性。

NewSQL 是指这样一类新式的关系型数据库管理系统，针对 OLTP（读-写）工作负载，追求提供和 NoSQL 系统相同的扩展性能，且仍然保持 ACID 和 SQL 等特性（scalable and ACID and (relational and/or sql -access)）

9

NewSQL 这个词至少 2011 年开始就出现了，不过从一开始就没有特别清晰的定义，现在也差不多。不过，NewSQL 出现在 NoSQL 之后，可以认为是分布式数据库自然演变的一部分，NewSQL 尝试解决 NoSQL 碰到的阻力，NoSQL 一般解释为 Not Only SQL，实际上 NoSQL 最常见有 4 种，Key-Value Store、Document Store、Column Oriented、Graph，通常是选择 Weak Consistency（Eventual Consistency）换取更好的性能

和扩展性，通常对 SQL 支持比较弱，NewSQL 则是尽可能支持 SQL（如果说 NoSQL 只支持 10-30% SQL Like，NewSQL 支持 70% 以上 SQL，大概是这样），在 Consistency Model 上更倾向 Strong 而不是 Weak，大约是 Linearizability，还有最关键的（个人观点）扩展性上更强调写能力的扩展，MySQL MGR 或者是 1 主（可读写）15 备（只读）这种应该算不上 New，NewSQL 应该支持多个（至少 3 个以上吧）主（可读写），在数据存储层通常采用 Paxos 或 Raft 共识算法支持多份副本（顺便说一句，不要以为因为有 Multi-Paxos，采用 Paxos 的数据库实现并发性能就必然好于 Raft），而不是一主多备来支持多份副本，为了优化写能力单机存储引擎通常采用有利于写的 LSM-Tree 而不是有利于读的 B+ Tree。

Consistency Model、对 SQL 的良好支持这些方面 NewSQL 和 NoSQL 相比更接近传统的 RDBMS，同时在扩展性（需要强调写能力的扩展，这是难点）上接近 NoSQL（NoSQL 绝大部分天然支持分布式，扩展性良好）。

10

本节就讲到这里，谢谢大家