



干货 | SQL 与 NoSQL还在傻傻分不清?



知乎用户...

微信公众号: Next_Academy

关注

118 人赞同了该文章

上一节我们认识了数据库，了解了数据库事务是什么，索引是如何提升数据库性能的，现在我们来学习下大家常说的一些数据库，MySQL、mongoDB、kv等等这些又有什么区别。本文中，SQL 与 NoSQL 代表关系型数据库与非关系型数据库，当然，SQL \neq 关系型数据库，这里用作简写。

1. 关系型数据库

1.1 什么是关系型数据库

关系数据库 (Relational database)，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。

- **数据库**：包括一个或多个表
- **表 (关系 Relation)**：是以列和行的形式组织起来的数据的集合
- **列 (属性 Attribute)**：在数据库中经常被称为字段
- **行 (值组 Tuple)**：在数据库中经常被称为记录

我们可以理解为：关系型数据库，是指采用了关系模型来组织数据的数据库。

例如，可能有一个有关作者信息的名为 authors 的表 (关系 Relation)。每列 (值组 Tuple) 都包含特定类型的信息，如作者的姓氏。每行 (属性 Attribute) 都包含有关特定作者的所有信息：姓、名、住址等等。

在关系型数据库当中一个表就是一个关系，一个关系数据库可以包含多个表。

关系型数据库的主要代表：SQL Server, Oracle, MySQL, PostgreSQL。

1.2 关系型数据库优点

- **事务一致性**：通过事务处理保持数据的一致性
- **复杂查询**：支持SQL，可以进行 JOIN 等复杂查询
- **容易理解**：二维表结构是非常贴近逻辑世界的一个概念，关系模型相对网状、层次等其他模型来



- **易于维护**：丰富的完整性(实体完整性、参照完整性和用户定义的完整性)大大减低了数据冗余和数据不一致的概率

1.3 关系型数据库缺点

- **读写性能**：在数据量达到一定规模时，由于关系型数据库的系统逻辑非常复杂，为了维护一致性，使得其非常容易发生死锁等的并发问题，所以导致其读写速度下滑非常严重
- **表结构更新**：表结构可以在被定义之后更新，但是如果有比较大的结构变更的话就会变得比较复杂
- **高并发**：网站的用户并发性非常高，往往达到每秒上万次读写请求，对于传统关系型数据库来说，硬盘I/O是一个很大的瓶颈
- **海量数据**：对于关系型数据库来说，在一张包含海量数据的表中查询，效率是非常低的

2. 非关系型数据库

2.1 什么是非关系型数据库

非关系型数据库 (NoSQL) 是对不同于传统的关系数据库的数据库管理系统的统称。

当代典型的关系数据库在一些数据敏感的应用中表现了糟糕的性能，例如为巨量文档创建索引、高流量网站的网页服务，以及发送流式媒体。关系型数据库的典型实现主要被调整用于执行规模小而读写频繁，或者大批量极少写访问的事务。

NoSQL 的结构通常提供弱一致性的保证，如最终一致性，或交易仅限于单个的数据项。

NoSQL 提出另一种理念，例如，以键值对存储，且结构不固定，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。

NoSQL 与 SQL 存在许多显著的不同点，其中最重要的是 NoSQL 不使用 SQL 作为查询语言。其数据存储可以不需要固定的表格模式，也经常避免使用 SQL 的 JOIN 操作，一般有水平可扩展性的特征。

非关系型数据库包括：

临时性键值存储：memcached、Redis=

永久性键值存储：ROMA、Redis

面向文档的数据库：MongoDB、CouchDB

面向列的数据库：Cassandra、HBase

主要代表：MongoDB, Redis, CouchDB

2.2 非关系型数据库分类

由于非关系型数据库本身天然的多样性，以及出现的时间较短，相比关系型数据库，非关系型数据库非常多，并且大部分都是开源的。

非关系型数据库严格上不是一种数据库，应该是一种数据结构化存储方法的集合。依据结构化方法以及应用场合的不同，主要分为以下几类：

- **面向高性能并发读写的 key-value 数据库**：key-value数据库的主要特点即使具有极高的并发读写性能，Redis, Tokyo Cabinet, Flare 就是这类的代表
- **面向海量数据访问的面向文档数据库**：这类数据库的特点是，可以在海量的数据中快速的查询数据，典型代表为 MongoDB 以及 CouchDB
- **面向可扩展性的分布式数据库**：这类数据库想解决的问题就是传统数据库存在可扩展性上的缺陷，这类数据库可以适应数据量的增加以及数据结构的变化



非关系型数据库的出现，多是源于关系型数据库的性能不足，故非关系型数据库的优点也很明显：

- **读写性能**：无需经过 SQL 层的解析，读写性能很高。主要例子有Redis，由于其逻辑简单，而且纯内存操作，使得其性能非常出色，单节点每秒可以处理超过10万次读写操作；
- **简单的扩展**：基于键值对，数据没有耦合性，容易扩展。典型例子是 Cassandra，由于其架构是类似于经典的 P2P，所以能通过轻松地添加新的节点来扩展这个集群；
- **存储格式多**：支持key-value形式、文档形式、图片形式，而关系型数据库则只支持基础类型；
- **低廉的成本**：这是大多数分布式数据库共有的特点，因为主要都是开源软件，没有昂贵的 License成本；

2.4 非关系型数据库缺点

- **不提供对 SQL 的支持**：如果不支持 SQL 这样的工业标准，将会对用户产生一定的学习和应用迁移成本
- **支持的特性不够丰富**：现有产品所提供的功能都比较有限，大多数 NoSQL 数据库都不支持事务，也不像 MS SQL Server 和 Oracle 那样能提供各种附加功能，比如 BI 和报表等
- **现有产品的不够成熟**：大多数产品都还处于初创期，和关系型数据库几十年的完善不可同日而语

3. 关系型数据库与 NoSQL

3.1 NoSQL 使用场景

并不是任何场景，NoSQL 都要优于关系型数据库。NoSQL 使用场景常见如下：

数据库表 schema 经常变化

比如在线商城，维护产品的属性经常要增加字段，这就意味着 ORMMapping 层的代码和配置要改，如果该表的数据量过百万，新增字段会带来额外开销（重建索引等）。

数据库表字段是复杂数据类型

对于复杂数据类型，比如 SQL Sever 提供了可扩展性的支持，像 xml 类型的字段。DB 层对 xml 字段很难建高效索引，应用层又要做从字符流到 dom 的解析转换。NoSQL 以 json 方式存储，提供了原生态的支持，在效率方便远远高于传统关系型数据库。

高并发数据库请求

此类应用常见于 web2.0 的网站，很多应用对于数据一致性要求很低，而关系型数据库的事务以及大表 JOIN 反而成了“性能杀手”。

海量数据的分布式存储

海量数据的存储如果选用大型商用数据，如 Oracle，那么整个解决方案的成本是非常高的，要花很多钱在硬件上。NoSQL 分布式存储，可以部署在廉价的硬件上，是一个性价比非常高的解决方案。

3.2 NoSQL 和关系数据库结合

一般把 NoSQL 和关系数据库进行结合使用，各取所长，需要使用关系特性的时候我们使用关系数据库，需要使用 NoSQL 特性的时候我们使用 NoSQL 数据库，各得其所。NoSQL 数据库是关系数据库在某些方面（性能，扩展）的一个弥补。

举个简单的例子吧，比如用户评论的存储，评论大概有主键 id、评论的对象 aid、评论内容 content、用户 uid 等字段。我们能确定的是评论内容 content 肯定不会在数据库用 where content=' ' 查询，评论内容也是一个大文本字段。那么我们可以把主键 id、评论对象 aid、用户 id 存储在数据库，评论内容存储在 NoSQL，这样数据库就节省了存储 content 占用的磁盘空间，从而节省大量 IO，对 content 也更容易做 Cache。

另外，可使用 NoSQL 作为缓存服务器。MySQL + Memcached 的架构中，Memcached 这类内存缓存服务器缓存的数据大小受限于内存大小，如果用 NoSQL 来代替 Memcached 来缓存数据库的话，就可以不再受限于内存大小。虽然可能有少量的磁盘IO读写，可能比 Memcached 慢一点，但是完全可以用来缓存数据库的查询操作。

4. 参考

- 维基百科-SQL
- 维基百科-NoSQL
- 为什么使用 NoSQL: NoSQL 与 SQL 的区别
- 关系型数据库和非关系型数据库的特性以及各自的优缺点
- 关系型数据库和非关系型数据库区别、oracle与mysql的区别

结束语: 本节主要针对关系型数据库和非关系型数据库的特性和区别, 由于在数据库相关的经验还有些缺乏, 对于数据库的理解, 还有后台存储的设计, 以及在不同场景下的方案使用, 或许还差的有些远呢。

文章来源: 腾讯工程师 王贝珊

发布于 2019-04-23 10:15

数据库 程序员

推荐阅读

我以为自己MySQL够厉害了, 直到看到了alibaba面试题

前言众所周知, 简历上 “了解 = 听过名字; 熟悉 = 知道是啥; 熟练 = 用过; 精通 = 做过东西” 相信大家对于MySQL的索引都不陌生, 索引(Index)是帮助MySQL高效获取数据的数据结构。索引作为MySQL...

编程大白

我以为自己MySQL够厉害了, 直到看到了alibaba面试题

前言众所周知, 简历上 “了解 = 听过名字; 熟悉 = 知道是啥; 熟练 = 用过; 精通 = 做过东西” 相信大家对于MySQL的索引都不陌生, 索引(Index)是帮助MySQL高效获取数据的数据结构。索引作为MySQL...

李超朋



MySQL 工作、底层原理一篇就够了!

里奥ii

发表于J

4 条评论

切换为时间排序

写下你的评论...



谢希晟Ross

2021-02-14

写的真的很好, 我们prof都不一定能解释的这么清楚

1



吴彪

03-14

纯小白提问: 把主键 id、评论对象 aid、用户 id 存储在数据库, 评论内容存储在 NoSQL。那当想知道某个用户的评论内容时, 要怎么关联呢

赞



MoFisher

2021-08-27

简洁明了, 感谢

赞



alando

2020-08-17

好清晰, 手动点赞