北京交通大学
BEIJING JIAOTONG UNIVERSITY




《大数据概论》
大数据存储与管理

鲍鹏
软件学院

本章内容

- 非关系型数据库概述
- 非关系型数据相关理论
- 典型的非关系型数据库
- 关系型数据库与非关系型数据库
- NewSQL的出现

时代背景

- 传统关系数据库管理系统（RDBMS）面临的问题
  - 很难支持分布式集群。
  - RDBMS对增删改查操作一视同仁，无法适应“频繁读和增加，不频繁修改”的特性。
  - 传统存储模式增大了运维的复杂性和扩展难度，无法满足数据库存储模式频繁变更的需求。

NoSQL是什么

- NoSQL是一种不同于关系数据库的数据库管理系统方案，是非关系型数据库的通称。

- 非关系数据库包括：
  - 键值数据库
  - 列数据库
  - 文档数据库
  - 图数据库

NoSQL的特点

- 可扩展性
- 灵活的数据模型
- 与云计算紧密结合

本章内容

- 非关系型数据库概述
- 非关系型数据相关理论
- 典型的非关系型数据库
- 关系型数据库与非关系型数据库
- NewSQL的出现

非关系型数据库相关理论

- NoSQL需要满足以下需求
  - 横向扩展
  - 高可用
  - 模式自由

- NoSQL三大基石理论：
  - CAP理论
  - BASE理论
  - 最终一致性

CAP理论

- 分布式是非关系型数据库的必要条件。
  - Consistency 一致性：在分布式环境中，多点数据是一致的。
  - Availability 可用性：快速获取数据，可以在确定的时间内返回操作结果。
  - Tolerance of Network Partition 分区容忍性：当出现网络分区的情况下，分离的系统依旧可以正常运行。

CAP理论

- 一个分布式系统不可能同时满足CAP三个需求，最多只能同时满足其中两个。

CAP理论

- CA：强调一致性(C)和可用性(A)
  - MySQL、SQLServer和PostgreSQL等
- CP：强调一致性(C)和分区容忍性(P)
  - Neo4J、BigTable和HBase等
- AP：强调可用性(A)和分区容忍性(P)
  - Dynamo、Riak、CouchDB等

忠告：不要将精力浪费在如何设计满足三者的完美分布式系统，而是应该进行取舍。

ACID理论

- BASE理论与数据库事务的ACID原则联系紧密，ACID原则是指：
  - Atomicity 原子性
    - 指事务必须是原子工作单元，对于其数据修改，要么全部执行，要么全都不执行
  - Consistency 一致性
    - 指事务完成时，必须使所有数据都保持一致状态
  - Isolation 隔离性
    - 指由并发事务所做的修改必须与任何其他并发事务所做的修改隔离
  - Durability 持久性
    - 指事务完成后，对系统的影响是永久性的

BASE理论

- BASE模型不同于ACID模型，牺牲一致性(C)，获得可用性(A)或分区容忍性(P)。
  - Basically Available（基本可用）：允许分区失败。
  - Soft state（软状态）：状态可以有一段时间不同步。
  - Eventually consistent（最终一致性）：最终数据是一致就可以，而不是每时每刻都一致。

BASE理论主要强调基本的可用性。

BASE理论

- Basically Available（基本可用）
- 什么是基本可用？
  - 响应时间上的损失
    - 正常情况下的搜索引擎0.5秒即返回给用户结果，而基本可用的搜索引擎可以在2秒作用返回结果。
  - 功能上的损失
    - 在一个电商网站上，正常情况下，用户可以顺利完成每一笔订单。但是到了大促期间，为了保护购物系统的稳定性，部分消费者可能会被引导到一个降级页面。

BASE理论

- Soft state（软状态）

- 什么是软状态？

  - 微观：允许系统中的数据存在中间状态，并认为该状态不影响系统的整体可用性。

  - 宏观：允许系统在多个不同节点的数据副本存在数据延时。

BASE理论

- Eventually consistent（最终一致性）
  - 经过一段时间后能够访问到更新后的数据。
  - 最终一致性的分类：
    - 因果一致性
    - 写一致性
    - 会话一致性
    - 单调读一致性
    - 单调写一致性

BASE理论

- 因果一致性
  - 进程A通知进程B它已更新数据项（因），那么进程B的后续访问将获得A写入的最新值（果）。
  - 与进程A无因果关系的进程C的访问，仍遵守一般的最终一致性规则。

BASE理论

- 写一致性
  - 因果一致性的特例。
  - 进程A自己执行一个更新操作后，自己总是可以访问到更新过的值。

BASE理论

- 会话一致性
  - 将对系统数据的访问过程框定在一个会话当中。
  - 执行更新操作之后，客户端能够在同一个会话中始终读取到该数据项的最新值。

BASE理论

- 单调读一致性
  - 如果进程已经看到过数据对象的某个值，那么任何后续访问都不会返回该值之前的值。
- 单调写一致性
  - 系统保证来自同一进程的写操作顺序执行。

本章内容

- 非关系型数据库概述
- 非关系型数据相关理论
- 典型的非关系型数据库
- 关系型数据库与非关系型数据库
- NewSQL的出现

NoSQL分类

- 键值数据库
- 列数据库
- 文档数据库
- 图数据库

键值数据库

- 键值数据库
  - 使用哈希表，键为特定值，值为指针。
  - 只能对键进行索引，值不可进行索引。
  - 值可以指向任意类型。
  - 由于不用建立索引，写操作性能优秀。
  - 由于值使用指针，扩容性强。

键值数据库

- 键值数据库的优缺点

- 优点：
  - 便于扩展，适用于云计算环境。
  - 与应用程序代码的兼容性好。
- 缺点：
  - 数据完整性约束移动至程序。
  - 目前很多键值数据存储系统之间不兼容。

键值数据库

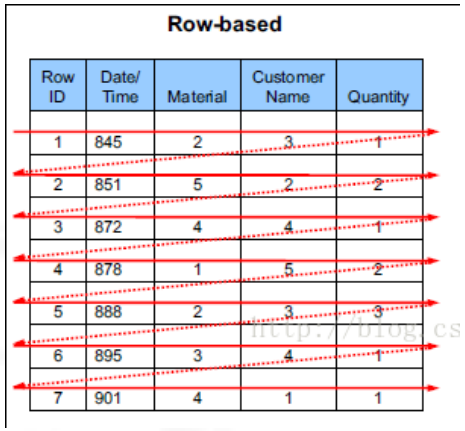
- Redis数据库
  - Redis是一个高性能的键值数据库。Redis的出现，是对关系数据库起到很好的补充作用。
  - 提供了Java，C/C++，C#，Python，Ruby，Erlang等客户端，使用方便。

- 优势：
  - 性能极高：读速度11万次/s，写速度8万次/s。
  - 丰富的数据类型：Redis支持5种数据类型操作。
  - 原子性：Redis的所有操作都是原子性的。
  - 丰富的特性：支持消息通知，key过期等等特性。

列数据库

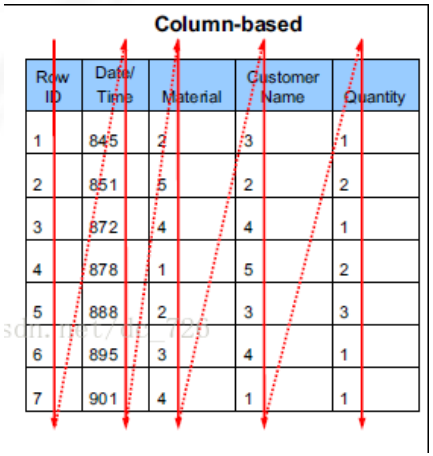
- 列数据库
  - 列族数据模型，数据库由多行构成，每行包含多个列族。
  - 同一列数据在内存空间中是存放在一起的。

列数据库



- 传统行数据库
  - 数据是按行存储
  - 没有索引的查询需要大量I/O
  - 建立索引需要花费大量时间和资源
  - 面对查询需求，数据库必须被大量膨胀才能满足性能需求

- 列数据库
  - 数据按列存储，即每一列单独存放
  - 数据即是索引
  - 之访问查询涉及的列，降低系统IO
  - 每一列可以由一个线程处理，可以并发处理
  - 同一列数据类型一致，数据特征相似，方便压缩



列数据库

- 列数据库 vs 行数据库

	行数据库	列数据库
优点	1. 数据被保存在一起 2. Insert/Update容易	1. 查询时只有涉及到的列会被读取 2. 投影（projection）高效 3. 任何列都能作为索引
缺点	1. 选择时即使只涉及某几行，所有数据也会被读取	1. 选择完成时，被选择的列要重新组装 2. Insert/Update较为麻烦

列数据库

- HBase（Hadoop Database）数据库
  - HBase是高性能、面向列、可伸缩的分布式存储系统，它可在廉价PC上搭建起大规模结构化存储集群。

- 优势：
  - 容量巨大：可对单表存储百亿或更多的数据。
  - 稀疏性：对于空列，并不占用存储空间。
  - 列存储：索引速度快。
  - 扩展性：可横向扩展，不断向集群添加服务器来提供存储空间和性能。
  - 高可靠性：基于HDFS的多副本机制。

文档数据库

- 文档数据库
  - 文档模型：文档是数据库的最小单位。
  - 结构灵活：文档型数据库不要求太严格的数据格式，一个集合中文档和文档之间的字段可以不一致。
  - 分布和弹性：扩展性强。
  - 查询语言：有自己的查询语言和API。

文档数据库

- 文档数据库的例子

id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas


```
{
   "_id": ObjectId("5146bb52d8524270060001f3"),
   "age": 25,
   "city": "Los Angeles",
   "email": "mark@abc.com",
   "user_name": "Mark Hanks"
}
{
   "_id": ObjectId("5146bb52d8524270060001f2"),
   "age": 31,
   "city": "Dallas",
   "email": "richard@abc.com",
   "user_name": "Richard Peter"
}
```

文档数据库

- 文档数据库的优点
  - 数据模式的直观性
  - 使用JSON
  - 模式的灵活性

文档数据库

- Mongodb数据库
  - MongoDB支持的数据结构是类似json的bson格式，因此可以存储比较复杂的数据类型。
  - Mongo支持的查询可以实现关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

- 优势：
  - 面向集合存储，易存储对象类型的数据。
  - 文件存储格式为BSON（一种JSON的扩展）。
  - 使用高效的二进制数据存储，包括大型对象。

图数据库

- 图数据库
  - 图数据库使用图结构作为数据模型存储数据。
  - 图数据库专门用于管理具有高度相互关联的数据。
  - 应用于社交网络、模式识别、推荐系统等

图数据库

- 图数据库的例子

表1 东汉末年人物表

ID	名字	性别	籍贯	...
1	曹操	男	沛国谯县	...
2	大乔	女	庐江郡皖县	...
3	关羽	男	河东解良	...
...	...	...	...	...

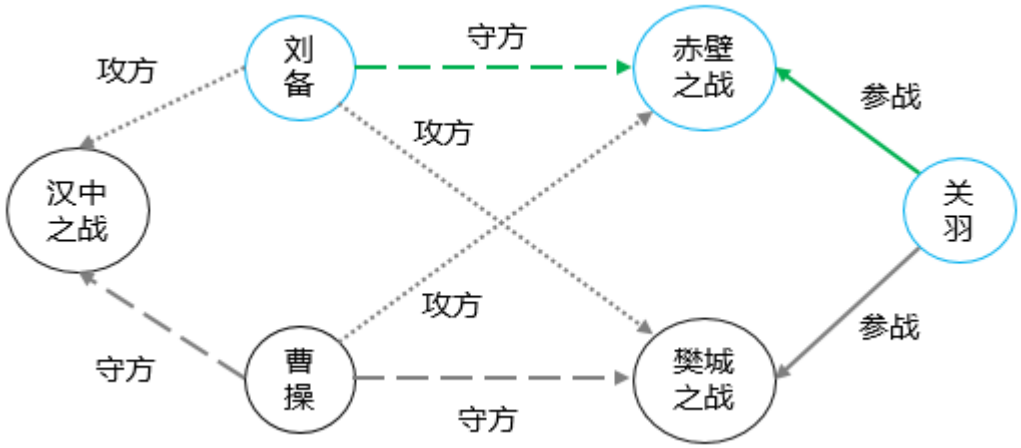
表2 东汉末年战役表

ID	战役	年份	攻方	守方
1	赤壁之战	208	曹操	孙权、刘备
2	汉中之战	217-219	刘备	曹操
3	樊城之战	219	刘备	曹操、孙权
...	...	...	...	...

表3 东汉末年人物参战表

ID	战役	参战人物ID
1	赤壁之战	1
2	赤壁之战	3
3	樊城之战	3
...	...	...

关系型数据库



图数据库

图数据库

- 图数据库的优点
  - 使用图数据库来表达现实世界的关系很直接、自然，易于建模。
  - 图数据库可以很高效的插入大量数据。
  - 图数据库可以很高效的查询关联数据。
  - 图数据库提供了针对图检索的查询语言。
  - 图数据库提供了专业的分析算法、工具。

图数据库

- Neo4j数据库
  - Neo4j是一个高性能的图数据库，它将结构化数据存储在图结构上。
  - Neo4j被视为一个高性能的图引擎，该引擎具有成熟数据库的所有特性，如：ACID、事务等。

- 优势：
  - 高效的关系遍历执行效率。
  - 基于属性图模型，支持丰富的数据语义描述。
  - 支持广泛的操作系统和最便捷的部署。

典型NoSQL数据库对比

分类	键值数据库	列存储数据库	文档数据库	图数据库
相关产品	Redis、Memcached	HBase、Riak、Cassandra	CouchDB、MongoDB	Neo4J、InfoGrid
典型应用	内容缓存	分布式文件系统	Web应用	社交网络、推荐系统等
数据模型	一系列键值对	以列簇式存储，同一列数据存放在一起	一系列键值对	图结构
优点	快速查询	查询速度快，可拓展性强，容易进行分布式扩展	数据结构要求不严格	利用图结构相关算法
缺点	存储的数据缺少结构化，条件查询困难	功能相对局限	查询性能不高，缺乏统一查询语法	需要对整个图进行计算，不利于做分布式方案

本章内容

- 非关系型数据库概述
- 非关系型数据相关理论
- 典型的非关系型数据库
- 关系型数据库与非关系型数据库
- NewSQL的出现

关系型数据库与NoSQL

- 关系型数据库 vs NoSQL

	关系型数据库	NoSQL
优点	1. 完善的关系代数理论为基础 2. 严格的标准 3. 支持ACID特性 4. 提供索引机制进行快速查询 5. 技术成熟	1. 支持超大数据存储 2. 灵活的数据模型 3. 具有较强的横向拓展能力
缺点	1. 无法支持海量数据存储 2. 数据模型死板 3. 无法支持海量数据存储 4. 事务机制影响系统性能	1. 缺乏数学理论基础 2. 复杂查询性能不高 3. 不能实现事务强一致性 4. 很难实现数据完整性 5. 技术不成熟

关系型数据库与NoSQL

- 关系型数据库与NoSQL的选择
  - 二者各有优势，也存在着不同层面的缺陷
    - 银行、超市需要关系型数据库保证数据强一致性
    - Web2.0领域则是NoSQL的主战场
  - 当今企业采用混合的方式构建数据库应用
    - 如今常用：MySQL+Redis的组合

本章内容

- 非关系型数据库概述
- 非关系型数据相关理论
- 典型的非关系型数据库
- 关系型数据库与非关系型数据库
- NewSQL的出现

NewSQL

- NewSQL是新的可扩展、高性能数据库的简称
  - 具有对海量数据的存储管理能力
  - 保持ACID特性

- NewSQL的特点
  - 支持关系数据模型
  - 使用SQL作为主要接口

NewSQL

- Spanner数据库
  - 数据同步时间10ms以内
  - 无锁读事务
  - 原子模式修改
  - 读历史数据无阻塞
- VoltDB数据库
  - 释放数据库中的缓冲池
  - 比关系型数据库快45倍
  - 每秒处理160万交易（300个CPU核心）

小作业

简答题:

阐述HDFS和HBase在Hadoop生态中的功能和联系，以及未来技术展望。

提交方式:

课程平台，word格式

提交时间:

下周二（5.24）