Atitit nosql的艺术 attilax著作

目录

[1. 1.5NoSQL数据库的类型 1](#_Toc30333)

[1.1. 1.5.1键值（Key/Value）存储 1](#_Toc27033)

[1.2. 1.5.2面向文档的数据库 1](#_Toc24354)

[1.3. 1.5.3面向列的数据库 1](#_Toc11230)

[1.4. 图形(Graph)数据库 1](#_Toc12245)

[2. 为什么需要NoSQL？ 2](#_Toc20188)

[3. 共同特征 3](#_Toc29873)

[3.1. Schema-less 3](#_Toc12135)

[3.2. Schemafree 3](#_Toc7122)

[4. 第2章NoSQL的数据一致性 4](#_Toc23899)

[4.1. 2.1传统关系数据库中的ACID 4](#_Toc12049)

[4.2. 2.2CAP理论 4](#_Toc14065)

[4.3. 2.4数据一致性模型与BASE 4](#_Toc10361)

[5. 第3章NoSQL的水平扩展与其他基础知识 4](#_Toc30680)

[6. 第2篇列式NoSQL系统 Hadoop 5.8Zookeeper 5](#_Toc8077)

[7. 第4篇文档型NoSQL系统 MongoDB 5](#_Toc8679)

[8. 图数据库　23 5](#_Toc15769)

[9. 其他nosql数据库 7](#_Toc5294)

[10. 选型产品 7](#_Toc6560)

[11. Ref 9](#_Toc4648)

Atitit nosql的概念与attilax的理解

# 1.5NoSQL数据库的类型

## 1.5.1键值（Key/Value）存储

## 1.5.2面向文档的数据库

## 1.5.3面向列的数据库

## ****图形(Graph)数据库****

# ****为什么需要NoSQL？****

关系数据库面临下列挑战：

并不适用于数据类型多样化（比如图像、视频和文本）的大量（PB级）数据。

无法扩展、支持庞大的数据量。

无法纵向扩展，受制于内存和处理器的功能。

无法横向扩展，受制于依赖缓存的读取和写入操作。

分片（将数据库分成几个部分，存储在不同的节点）引起操作问题（比如管理共享式故障）。

复杂的RDBMS模型

一致性限制了RDBMS的可扩展性。

# ****共同特征****

对于NoSQL并没有一个明确的范围和定义，但是他们都普遍存在下面一些共同特征：

## Schema-less

不需要预定义模式：不需要事先定义数据模式，预定义表结构。数据中的每条记录都可能有不同的属性和格式。当插入数据时，并不需要预先定义它们的模式。

无共享架构：相对于将所有数据存储的存储区域网络中的全共享架构。NoSQL往往将数据划分后存储在各个本地服务器上。因为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，从而提高了系统的性能。

弹性可扩展：可以在系统运行的时候，动态增加或者删除结点。不需要停机维护，数据可以自动迁移。

分区：相对于将数据存放于同一个节点，NoSQL数据库需要将数据进行分区，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。

异步复制：和RAID存储系统不同的是，NoSQL中的复制，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，而不会被网络传输引起迟延。缺点是并不总是能保证一致性，这样的方式在出现故障的时候，可能会丢失少量的数据。

BASE：相对于事务严格的ACID特性，NoSQL数据库保证的是BASE特性。BASE是最终一致性和软事务。

NoSQL数据库并没有一个统一的架构，两种NoSQL数据库之间的不同，甚至远远超过两种关系型数据库的不同。可以说，NoSQL各有所长，成功的NoSQL必然特别适用于某些场合或者某些应用，在这些场合中会远远胜过关系型数据库和其他的NoSQL。

## Schemafree

Collection中可以包含具有不同schema的文档记录

# 第2章NoSQL的数据一致性

## 2.1传统关系数据库中的ACID

## 2.2CAP理论

## 2.4数据一致性模型与BASE

2.2.1NoSQL系统是分布式系统  
2.2.2CAP理论阐述  
2.3AP的例子--DNS系统  
2.3.1DNS系统  
2.3.2DNS域名解析过程  
2.3.3DNS系统是最终一致性的  
2.4数据一致性模型与BASE  
2.4.1数据一致性模型  
2.4.2BASE（BasicallyAvailable，Soft-state，Eventualconsistency）  
2.5数据一致性实现方法  
2.5.1Quorum系统NRW策略  
2.5.2时间戳策略  
2.5.3向量时钟

# 第3章NoSQL的水平扩展与其他基础知识

3.1所有数据存放在一个服务器上  
3.2分片（Sharding）  
3.3主从复制  
3.4对等（PeerToPeer）复制  
3.5复制和分片的同时使用  
3.6数据水平扩展的方法总结  
3.7分片对数据的划分方式  
3.7.1Range-BasedPartitioning  
3.7.2Round-Robin  
3.8一致性hash算法（ConsistentHashing）  
3.8.1基本场景  
3.8.2hash算法和单调性  
3.8.3ConsistentHashing算法的原理  
3.8.4虚拟节点  
3.9磁盘的读写特点及五分钟法则  
3.9.1磁道、扇区、柱面和磁头数  
3.9.2固态硬盘（SSD）：随机读写速度快  
3.9.3内存：读写速度极快  
3.9.4五分钟法则  
3.10不要删除数据

# 第2篇列式NoSQL系统 Hadoop 5.8Zookeeper

Kv Redis

# 第4篇文档型NoSQL系统 MongoDB

.5　比较ACID和BASE——两种可靠的数据库事务方法　19  
  
2.5.1　RDBMS的事务控制——ACID　21  
  
2.5.2　非RDBMS的事务控制—BASE　22  
  
2.6　通过数据库分片获得水平扩展能力　23  
  
2.7　基于Brewer的CAP定理进行权衡　25

# 图数据库　23

简介　23  
  
3.1　图论基础　24  
  
3.1.1　节点　24  
  
3.1.2　边　25  
  
3.1.3　图的结构　25  
  
3.2　RDBMS与图数据库　26  
  
3.3　凯文?贝肯问题的六度　26  
  
3.3.1　通用图的邻接表模型　27  
  
3.3.2　通用图的覆盖路径模型　30  
  
3.3.3　真实数据的复杂关系　32

第4章　NoSQL数据架构模式　51

4.1　键值存储　52

4.1.1　什么是键值存储　52

4.1.2　使用键值存储的好处　54

4.1.3　使用键值存储　56

4.1.4　使用案例：用键值存储保存网页　59

4.1.5　使用案例：亚马逊简单存储服务（S3）　59

4.2　图存储　60

4.2.1　图存储概述　60

4.2.2　用RDF标准来连接外部数据　62

4.2.3　图存储的使用案例　63

4.3　列族（Bigtable）存储　68

4.3.1　列族存储基础　69

4.3.2　理解列族存储的键　69

4.3.3　列族存储的优点　70

4.3.4　案例研究：在Bigtable中存储分析信息　72

4.3.5　案例研究：Google地图用Bigtable存储地理信息　72

4.3.6　案例研究：使用列族存储用户偏好信息　73

4.4　文档存储　73

4.4.1　文档存储基础　74

4.4.2　文档集合　74

4.4.3　应用集合　75

4.4.4　文档存储的API　75

4.5　NoSQL架构模式的变体　78

4.5.1　定制RAM和SSD存储　78

4.5.2　分布式存储　78

4.5.3　分组的对象　79

# 其他nosql数据库

第5章　流式数据库和复杂事件　55

第7章　文本数据库　77

第8章　地图数据　89

生物特征、指纹和专业数据库　109

1. 分析型数据库　123
2. 第12章　多值数据库或NFNF数据库　145
3. 层次数据库系统和网络数据库系统　155

# 选型产品

NoSQL数据库的四大分类的分析

分类

Examples举例

典型应用场景

数据模型

优点

缺点

键值（key-value）

Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB

内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等等。

Key 指向 Value 的键值对，通常用hash table来实现

查找速度快

数据无结构化，通常只被当作字符串或者二进制数据

列存储数据库

Cassandra, HBase, Riak

分布式的文件系统

以列簇式存储，将同一列数据存在一起

查找速度快，可扩展性强，更容易进行分布式扩展

功能相对局限

文档型数据库

CouchDB, MongoDb

Web应用（与Key-Value类似，Value是结构化的，不同的是数据库能够了解Value的内容）

Key-Value对应的键值对，Value为结构化数据

数据结构要求不严格，表结构可变，不需要像关系型数据库一样需要预先定义表结构

查询性能不高，而且缺乏统一的查询语法。

图形(Graph)数据库

Neo4J, InfoGrid, Infinite Graph

社交网络，推荐系统等。专注于构建关系图谱

图结构

利用图结构相关算法。比如最短路径寻址，N度关系查找等

很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群

# Ref

《NoSQL数据库技术实战》(皮雄军)【简介\_书评\_在线阅读】 - 当当图书.html

解读NoSQL\_百度百科.html

NoSQL权威指南 (豆瓣).html

NoSQL数据库的四大分类的分析 - Don't\_Stop - 博客园.html

NoSQL数据库的四大分类的分析 - Don't\_Stop - 博客园.html