

# Nosql概述

2020年7月16日 13:37

为什么使用nosql

一句话：大数据时代的到来！

nosql处理大数据的情况

什么是nosql（不仅仅是sql）

not only sql

泛指非关系型数据库，随着web2.0的诞生！传统的关系型数据库很难对付！尤其是超大规模的高并发的社区！

暴露出来难以克服的问题。

nosql的特点

数据之间没有关系，很好扩展

大数据量高性能（redis一秒写8万次，读取11万，性能高）

数据类型是多样的。不需要事先设计数据库

传统的RDBM

-结构化组织

-sql

-数据和关系都存在单独的表里

-严格的一致性

-基础的事务

nosql

没有固定的查询语言

CAP定理和BASE

高性能，高可用，高可扩展性

大数据时代的3V和3高

3v描述问题的。3高解决问题的

1、海量Volume

2、多样Variety

3、实时Velocity

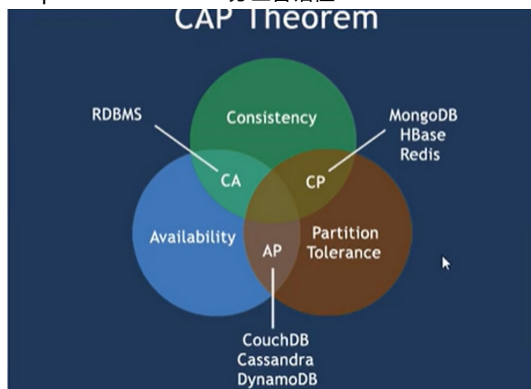
三高，高并发，高可扩，高性能

CAP理论

C: consistency强一致性

A: availability高可用性

P: partition tolerance分区容错性



CAP理论：3进2。一个分布式系统不可能同时满足一致性，可用性和分区容错性三个需求。

最多同时满足两个

因此，根据 CAP 原理将 NoSQL 数据库分成了满足 CA 原则、满足 CP 原则和满足 AP 原则三 大类：

CA - 单点集群，满足一致性，可用性的系统，通常在可扩展性上不太强大。

CP - 满足一致性，分区容忍必的系统，通常性能不是特别高。

AP - 满足可用性，分区容忍性的系统，通常可能对一致性要求低一些。

CA-----MYSQL

CP-----REDIS

BASE就是为了解决关系数据库强一致性引起的问题而引起可用性降低而提出的解决方案

基本可用Bastic Available

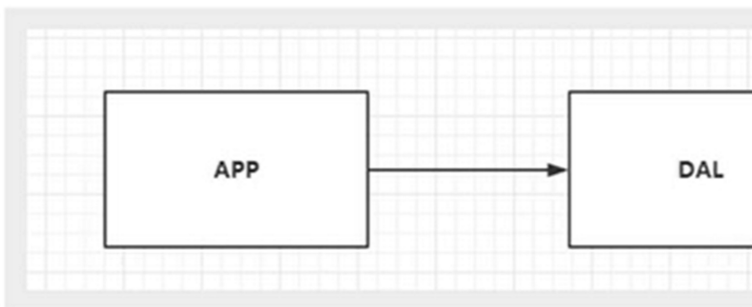
软状态soft

最终一致 Eventually consistent

它的思想是通过让系统放松对某一时刻的要求来换取系统整体伸缩性和性能上的改观。因为大型系统往往由于地域分布和极高性能的要求，不可能采用分布式事务来完成这些指标，要想获得这些指标，必须采用另外一种方式来完成。BASE就是解决这个问题方法。

它的思想是通过让系统放松对某一时刻数据一致性的要求来换取系统整体伸缩性和性能上改观。为什么这么说呢，缘由就在于大型系统往往由于地域分布和极高性能的要求，不可能采用分布式事务来完成这些指标，要想获得这些指标，我们必须采用另外一种方式来完成，这里BASE就是解决这个问题的办法

## 1、单机MySQL的年代！



90年代，一个基本的网站访问量一般不会太大，单个数据库完全足够！

那个时候，更多的去使用静态网页 Html ~ 服务器根本没有太大的压力！

思考一下，这种情况下：整个网站的瓶颈是什么？

1、数据量如果太大、一个机器放不下了！

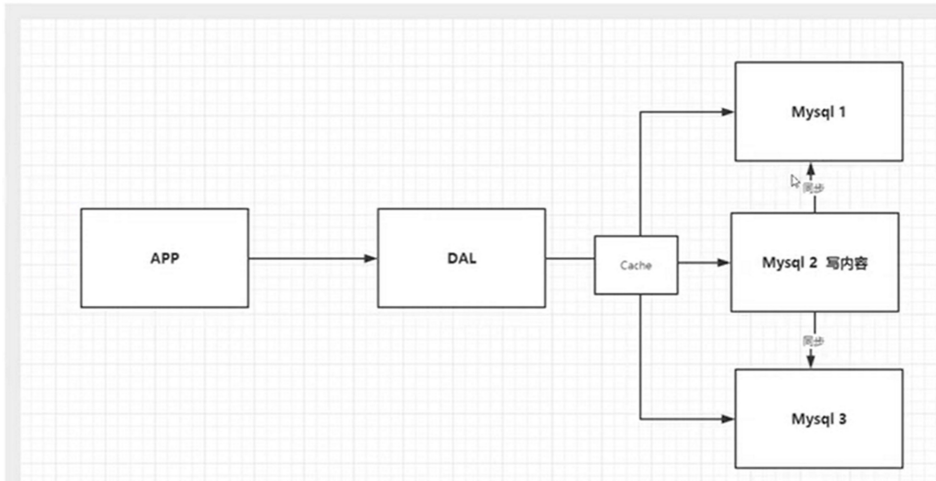
2、数据的索引（B+ Tree），一个机器内存也放不下

3、访问量（读写混合），一个服务器承受不了~ I

只要你开始出现以上的三种情况之一，那么你就必须要晋级！

## 2、Memcached (缓存) + MySQL + 垂直拆分

网站80%的情况都是在读，每次都要去查询数据库的话就十分的麻烦！所以说我们希望减轻数据的压力，我们可以使用缓存来保证效率！



nosql四大类

KV键值对

新浪：redis

美团：redis+tair

阿里，百度：redis+memecache

文档型数据库（bson格式和json一样）

mongodb（一般要掌握）分布式存储的数据库，是nosql和关系型数据库的交集

列存储数据库

Hbase

分布式文件系统

图形关系数据库

不是存图形的，放的是关系，比如朋友圈社交网络，广告推荐

neo4j, infogrid

分类	Examples举例	典型应用场景	数据模型	优点	缺点
键值 (key-value) [3]	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。[3]	Key 指向 Value 的键值对, 通常用 hash table 来实现[3]	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[3]
列存储数据库 [3]	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 [3]	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 [3]	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方案。[3]