数据访问层（DAL）

KISS原则是英语 Keep It Simple, Stupid 的首字母缩写

cap原则 Consistency（一致性）、 Availability（可用性）、Partition tolerance（分区容错性），三者不可得兼[

● 一致性（C）：在分布式系统中的所有数据备份，在同一时刻是否同样的值。（等同于所有节点访问同一份最新的数据副本）

● 可用性（A）：在集群中一部分节点故障后，集群整体是否还能响应客户端的读写请求。（对数据更新具备高可用性）

● 分区容错性（P）：以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择。

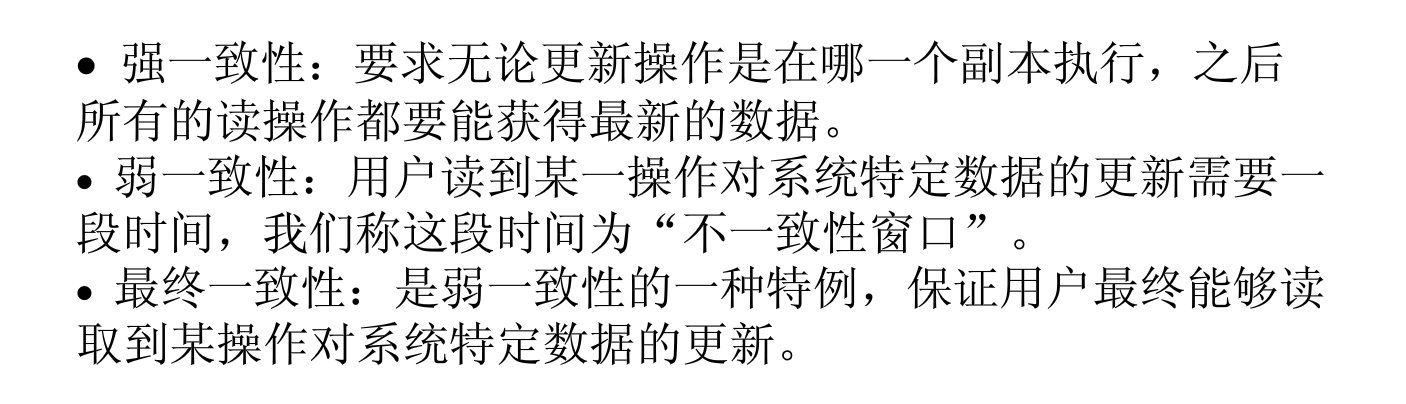


c 节点数据一致吗

a 挂了节点还能正常服务吗

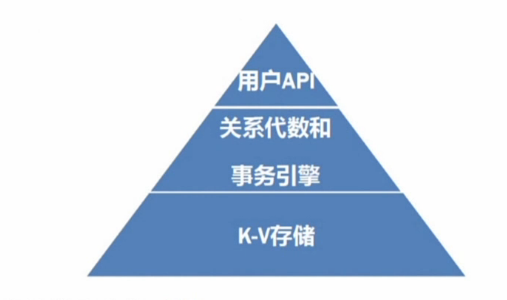
p 通信的速度够快吗

事务就是强一致性的体现





SQL数据库功能



execute,query,update(sql) rollback(transaction)

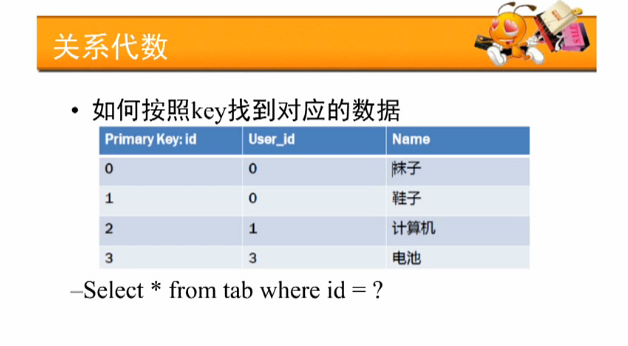
K-V ?

map:get value by key(java hash map,treemap)



NOSQL 原始结构,达到高性能





query type:

iter all

o[n]: all iterate and compare

secondary index(存在数据倾斜 全表扫描问题,回表)

combined index(multikey -> primary key)

# 事务

关键：锁

目的：方便人脑理解，人脑理解-串行



查询-更新

线程独占 锁

本质：加锁 去锁，保证执行顺序而又尽可能地快

四种隔离级别

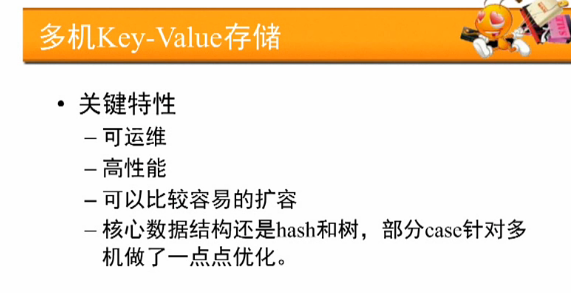


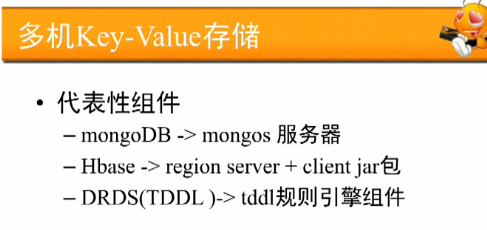


数据库的好坏很大程度取决于sql->执行计划

单机有io上限

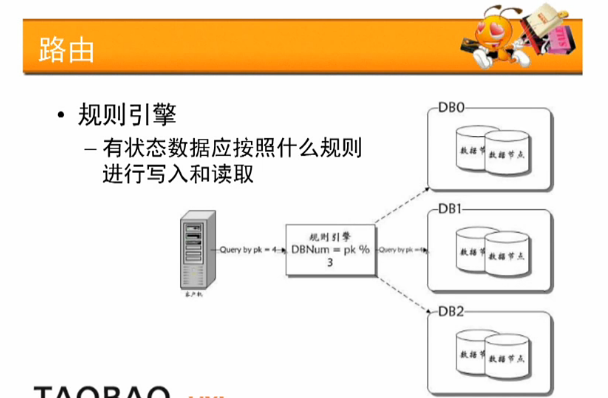
以上为单机数据库

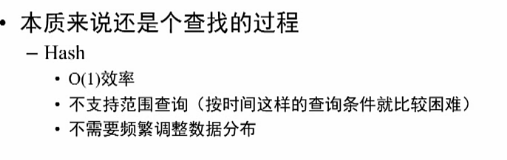


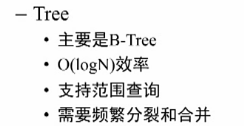


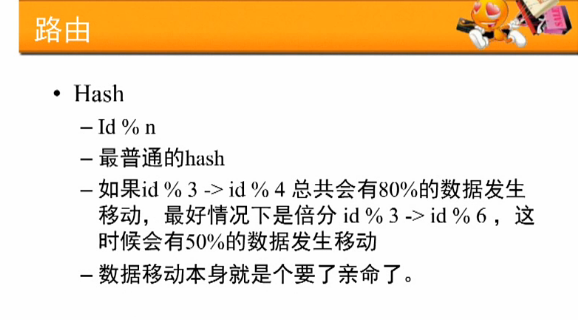
单机->多机 by 路由

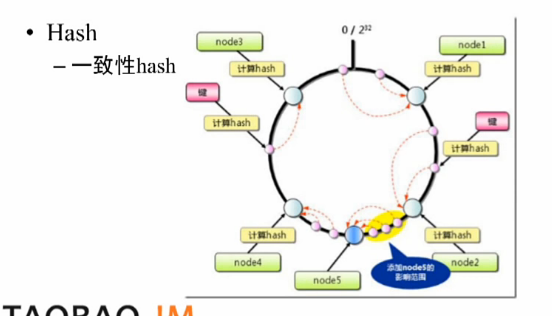
hash and b-tree

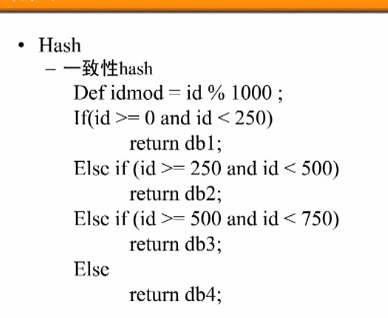


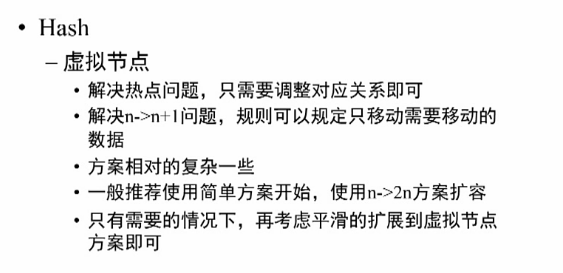


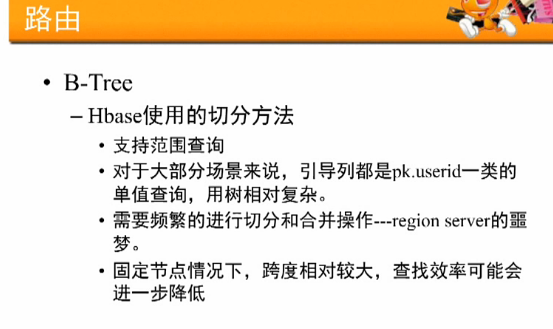




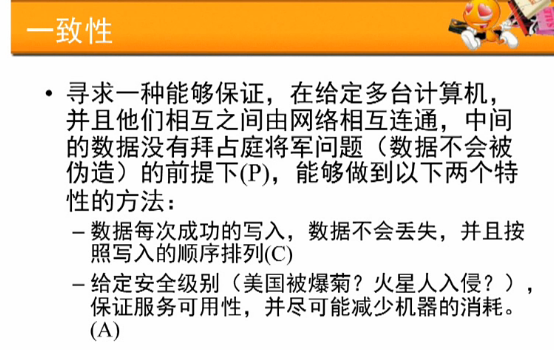








一致性 源自论文，权衡点

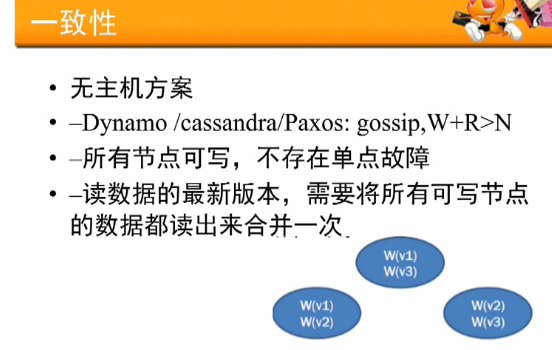


A：定义了给定时间内返回结果

# Consistency(一致性), 数据一致更新，所有数据变动都是同步的 Availability(可用性), 好的响应性能 Partition tolerance(分区容错性) 可靠性

分区容错性：理解为在存在网络分区的情况下，仍然可以接受请求（满足一致性和可用性)。这里的网络分区是指由于某种原因，网络被分成若干个孤立的区域，而区域之间互不相通。还有一些人将分区容错性理解为系统对节点动态加入和离开的能力，因为节点的加入和离开可以认为是集群内部的网络分区。

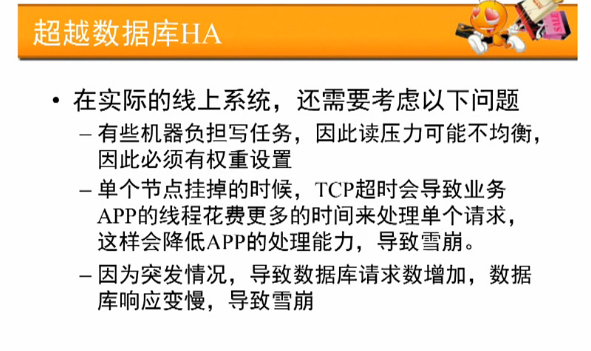
　　Partition Tolerance的意思是，在网络中断，消息丢失的情况下，系统照样能够工作。 以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择

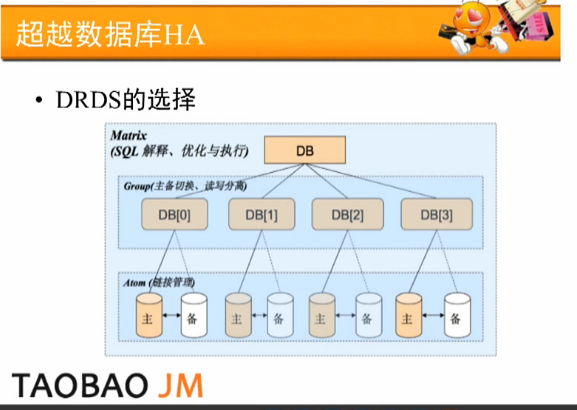


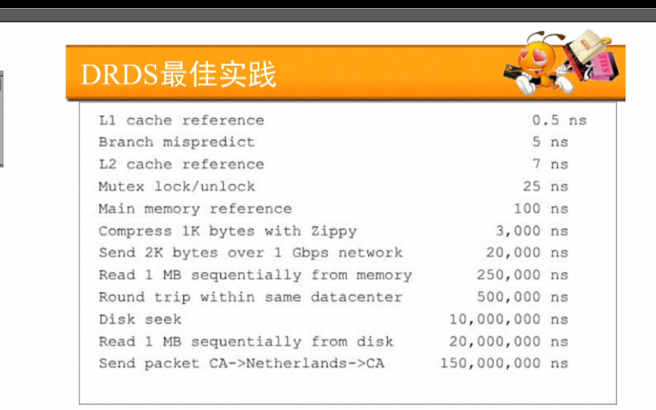
写冲突：

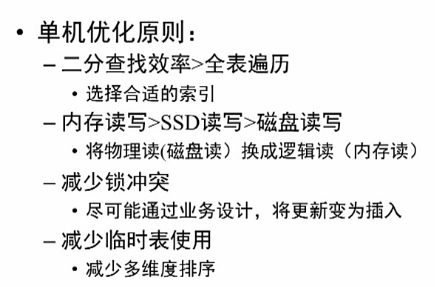


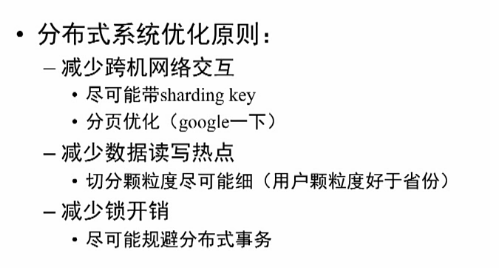
有主机方案性能要高，因为无主机方案需要处理写冲突

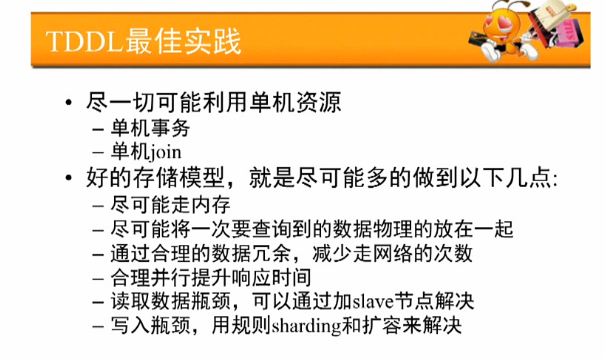












mysql5.1

100~500W 写入有拐点

尽量单表数据不要超过1000W

5.5以前DBA锁表