

太原站

KING 大加面对面

— 数据库平替之高可用保障业务零中断 —

从“单点崩溃”到“永不掉线”： 数据库高可用的生存法则

分享嘉宾：薛晓刚

01 高可用

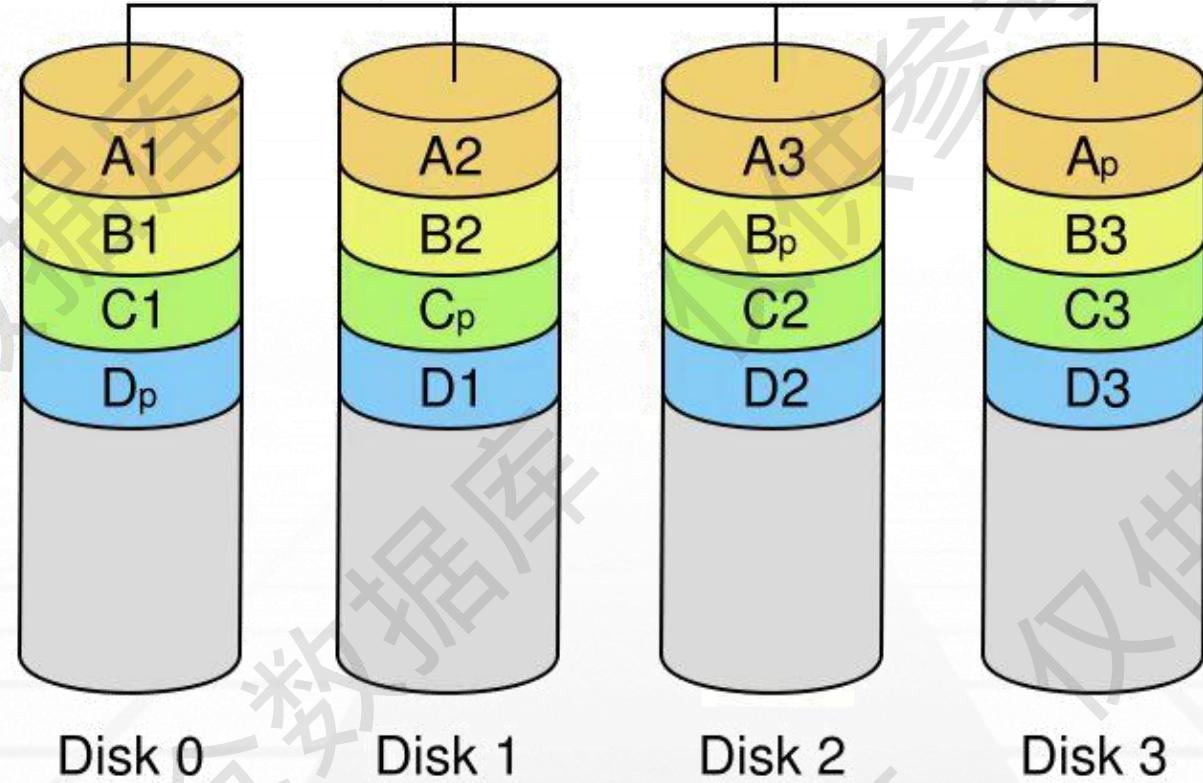
>>> 网络

>>> 存储

>>> 数据库

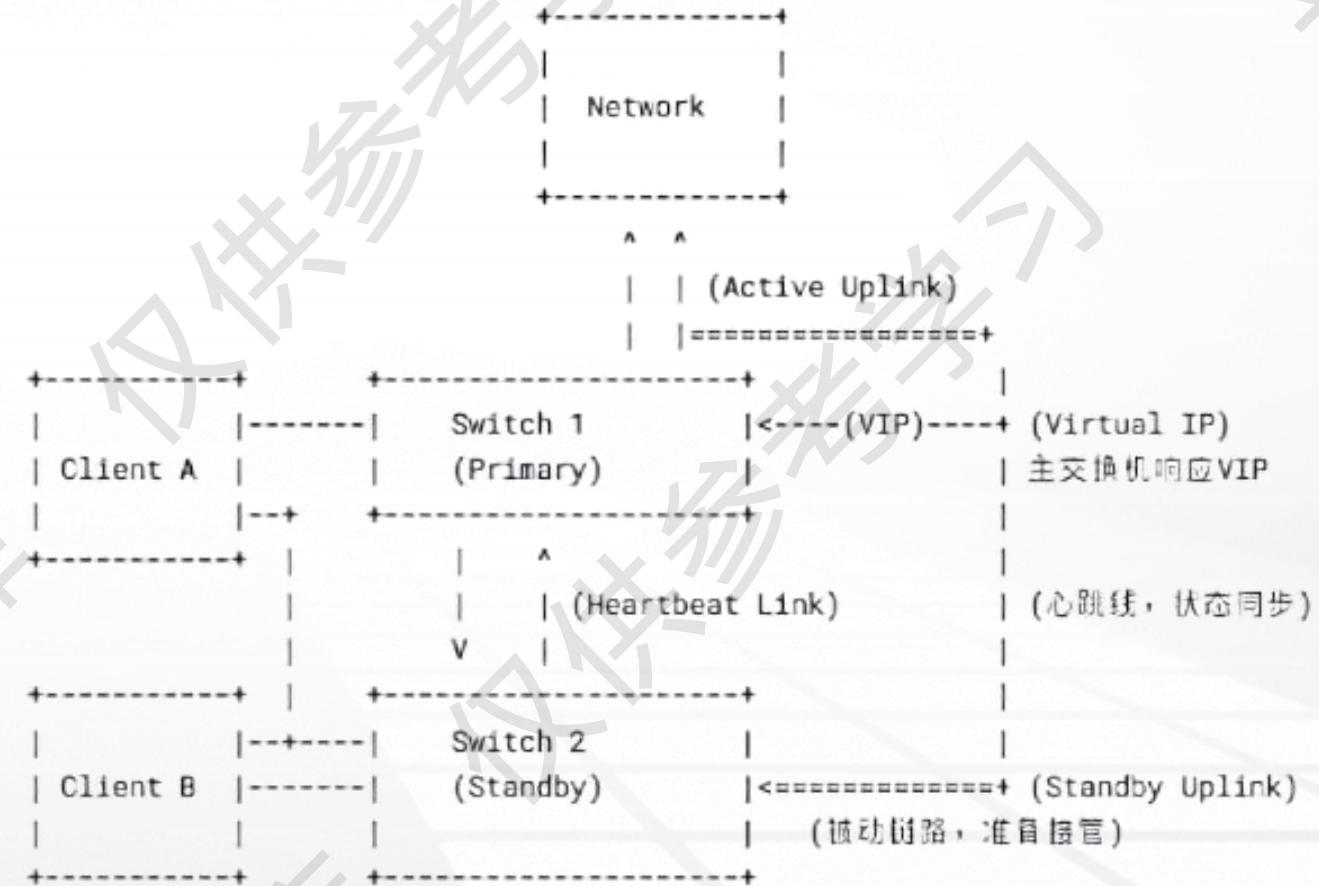
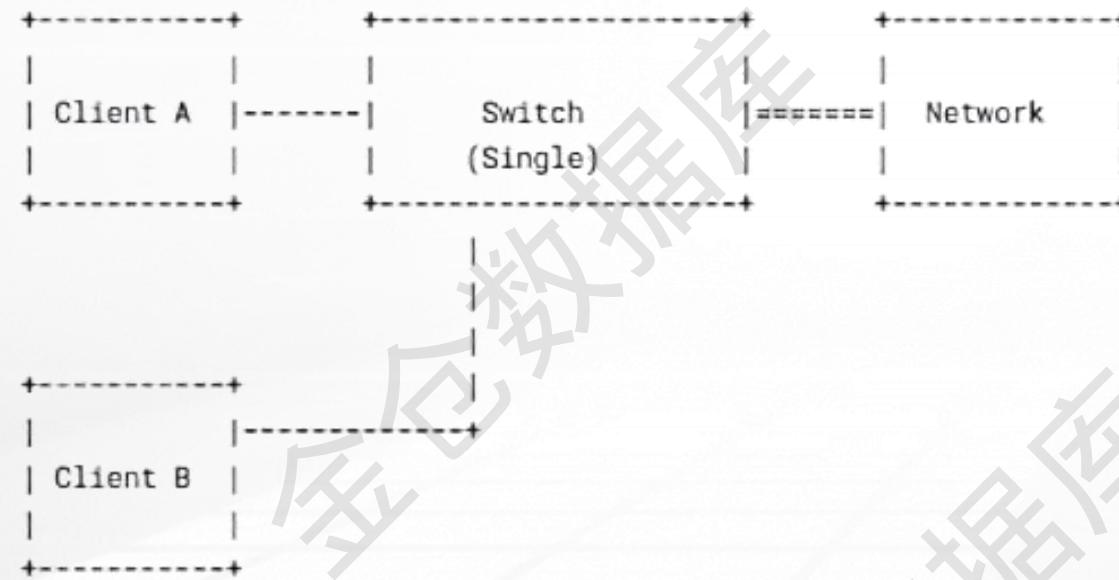
>>> 机房

RAID 5

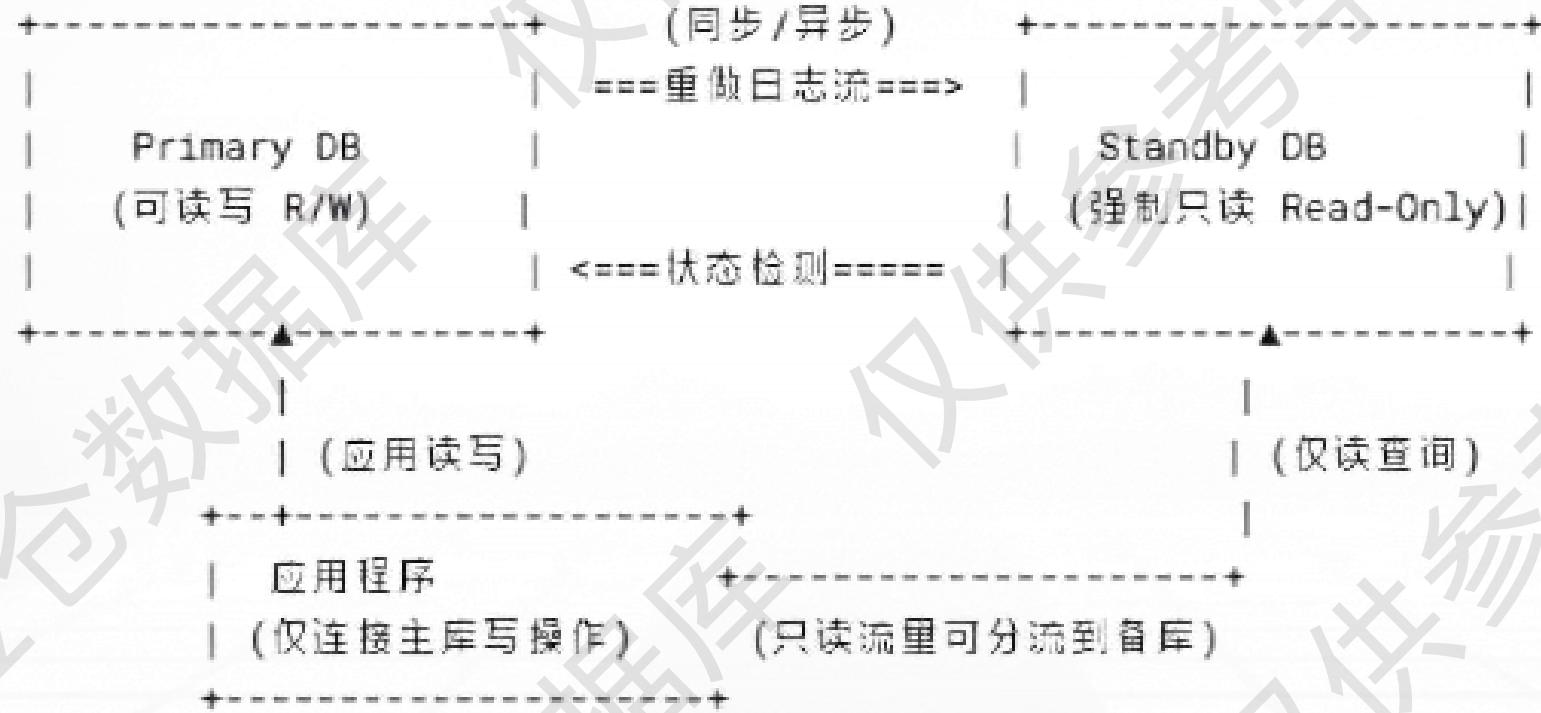


高可用--网络

KING BASE | 金仓社区

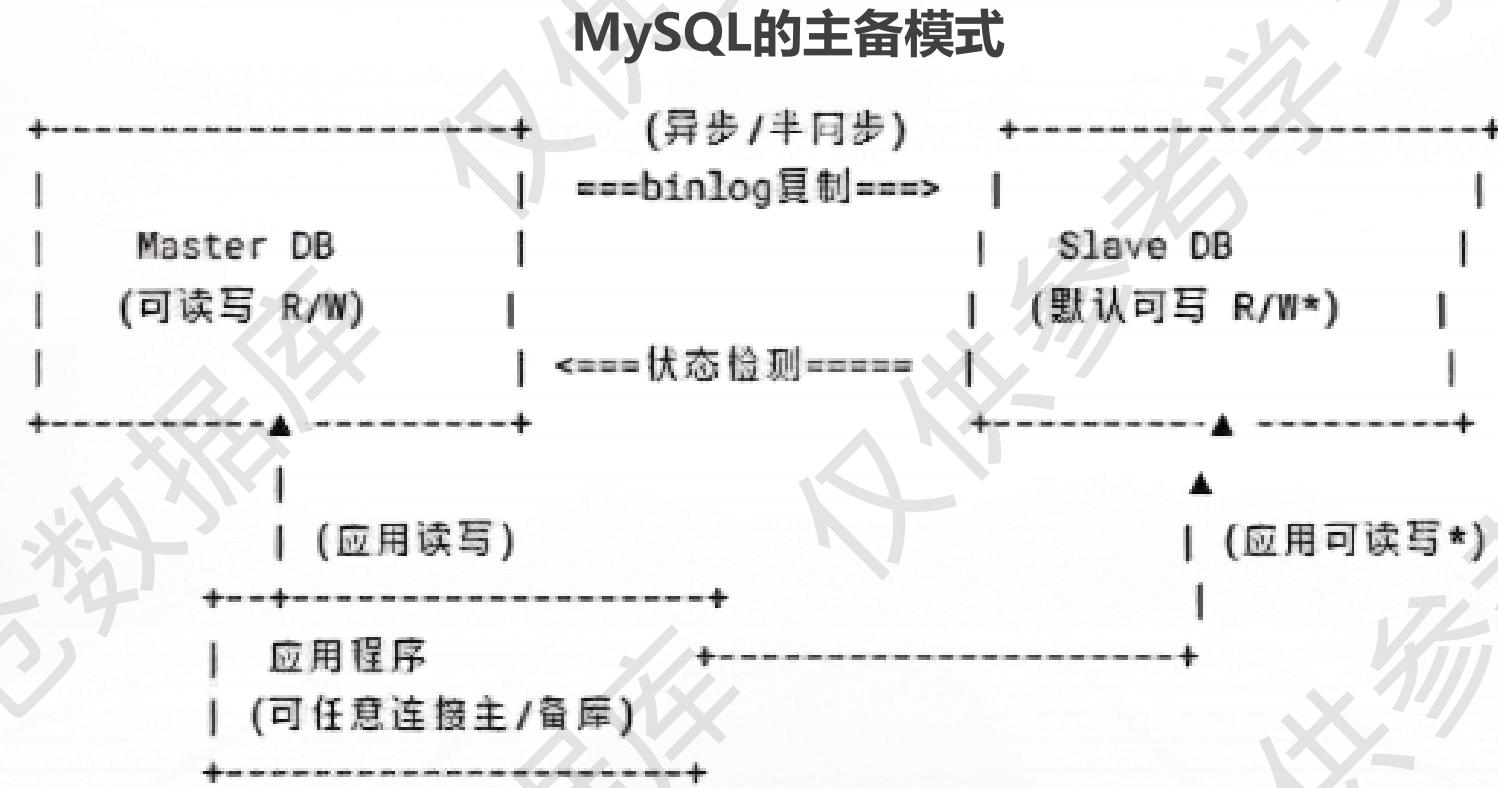


Oracle ADG (Active Data Guard) 主备模式



关键特性：

- 备库物理级别禁止写操作（即使尝试写入也会被阻止）
- 数据通过重做日志 (Redo) 实现块级同步
- 备库可开放只读查询 (用于报表等场景)



关键特性：

- 备库默认允许写操作 (*但会导致主从数据不一致！)
- 生产环境必须手动设置 `read_only=1` 限制写入
- 数据通过逻辑日志 (binlog) 实现 SQL 语句级同步
- 典型部署 应用主动规避写备库

PostgreSQL的主备模式



关键特性：

- 备库物理级别禁止写操作（尝试写入将报错）
- 故障通过WAL(预写日志)实现块级同步
- 支持级联复制（Standby可再级联Standby）

关键差异对比：

特性	Oracle ADG	MySQL	PostgreSQL
备库写入能力	物理禁止写入	默认允许写入*	物理禁止写入
同步机制	重做日志 (块级)	binlog(SQL语句级)	WAL (块级)
只读控制	自动强制	需手动设置read only=1	自动强制
故障切换	需手动/FSFO自动切换	需手动/工具切换	需手动/pg auto failover
典型用途	高可用+报表分离	读写分离	高可用+报表分离

传统行业对于主备架构下的数据库高可用 的使用最佳实践

主备架构
一致性



备库被写入



延迟
(延迟是不是一定是
坏事)

主备架构下要不要切换?

什么时候切换?

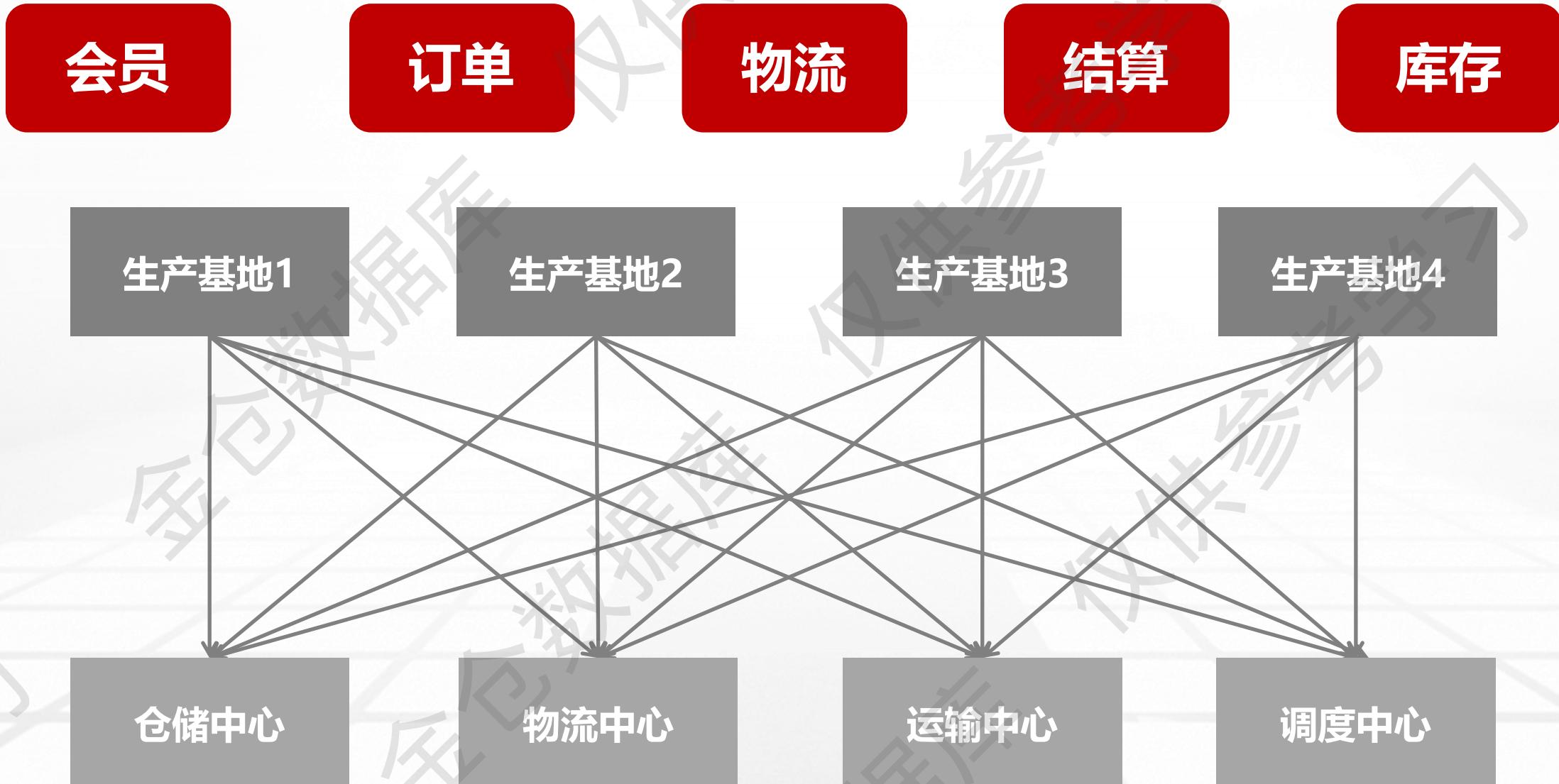
对一致性造成冲击的最大原因

微服务的陷阱

业务数据缺设计，系统拆分不合理

资源服务投入大，费用支出持续增

橘生淮北则为枳



缺点：

- 1.更多的服务也就意味着更多的运维。传统的架构开发者只需要保证一个应用正常运行，而现在却需要保证几十甚至上百道工序高效运转，这是一个艰巨的任务。
- 2.DevOps要求使用微服务架构后，开发团队需要保证一个Tomcat集群可用，保证一个数据库可用，这就意味着团队需要高品质的DevOps和自动化技术。
- 3.隐式接口服务和服务之间通过接口来“联系”，当某一个服务更改接口格式时，可能涉及到此接口的所有服务都需要做调整。
- 4.重复劳动在很多服务中可能都会使用到同一个功能，而这一功能点没有足够大到提供一个服务的程度，这个时候可能不同的服务团队都会单独开发这一功能，重复的业务逻辑，这违背了良好的软件工程中的很多原则。
- 5.分布式系统的复杂性微服务通过REST API或消息来将不同的服务联系起来，这在之前可能只是一个简单的远程过程调用。而实际上发现面对一个接口后面还有无数个接口，调用链长的超出了想想。
- 6.分布式系统也就意味着开发者需要考虑网络延迟、容错、消息序列化、不可靠的网络、异步、版本控制、负载等。一句话：原来是单机数据库实例来保证一致性，现在谁来保证？

怎么做？不会。

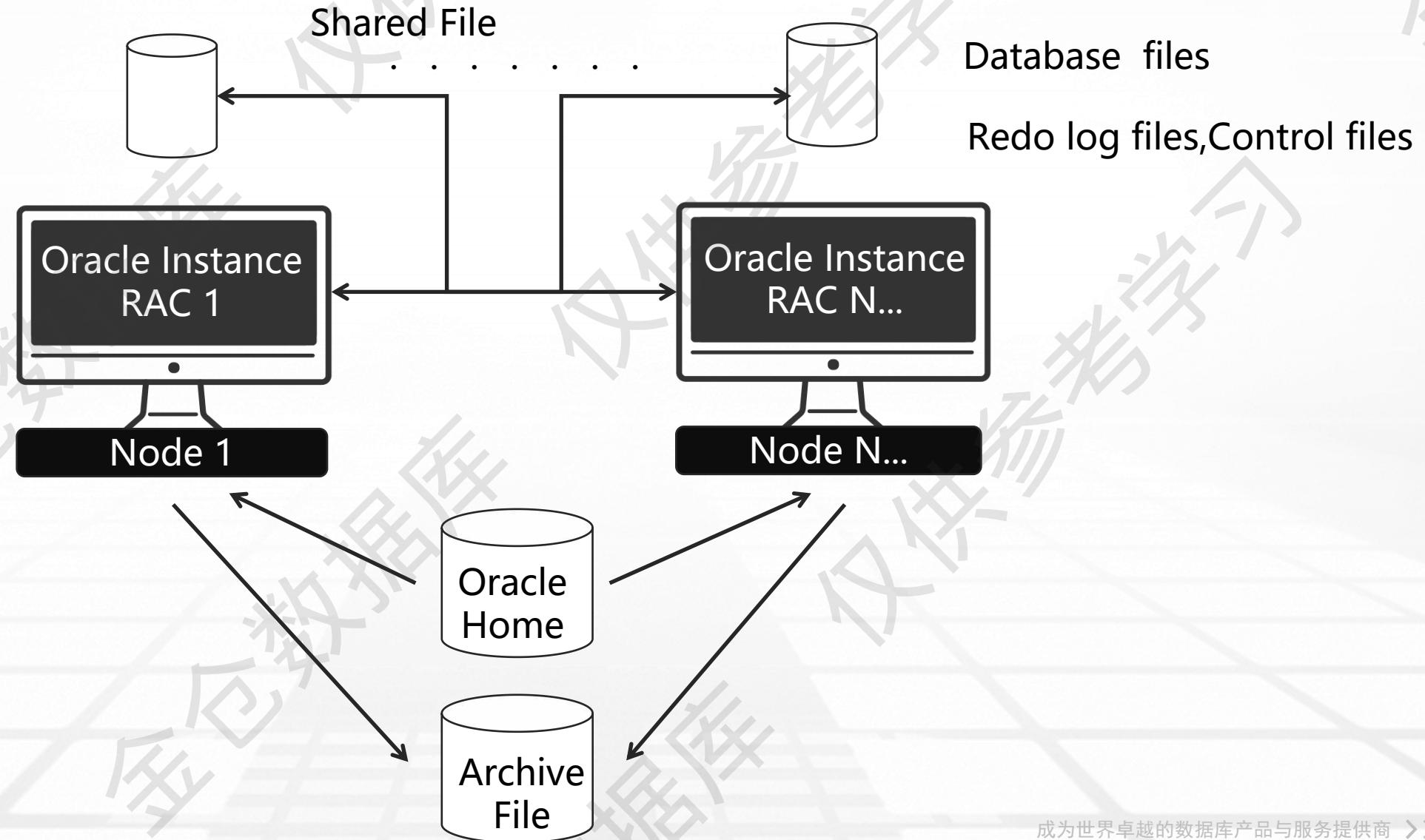
不会做还不会抄吗？阿里腾讯怎么做的？

传统企业往往只能学其表，引入大型互联网企业的技术的同时也引入了IT的复杂性，但是并没办法掌握解决复杂性问题的方法。

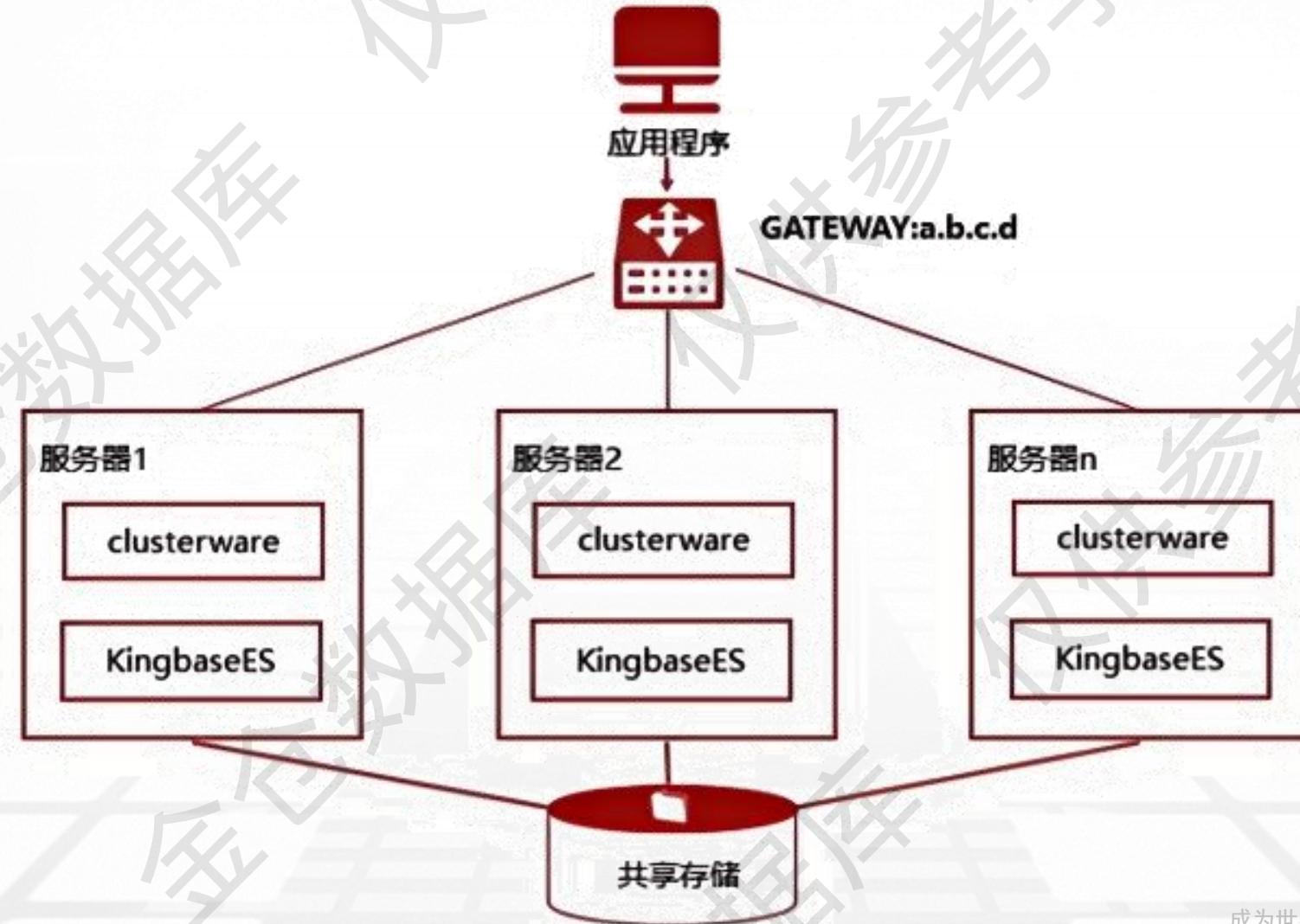
很多企业或者团队低估了复杂性所带来的成本，因此过于强调了敏捷和可扩展性带来的好处。

实际上很多企业的业务场景就不适合微服务。因为都不需要敏捷也做不到。

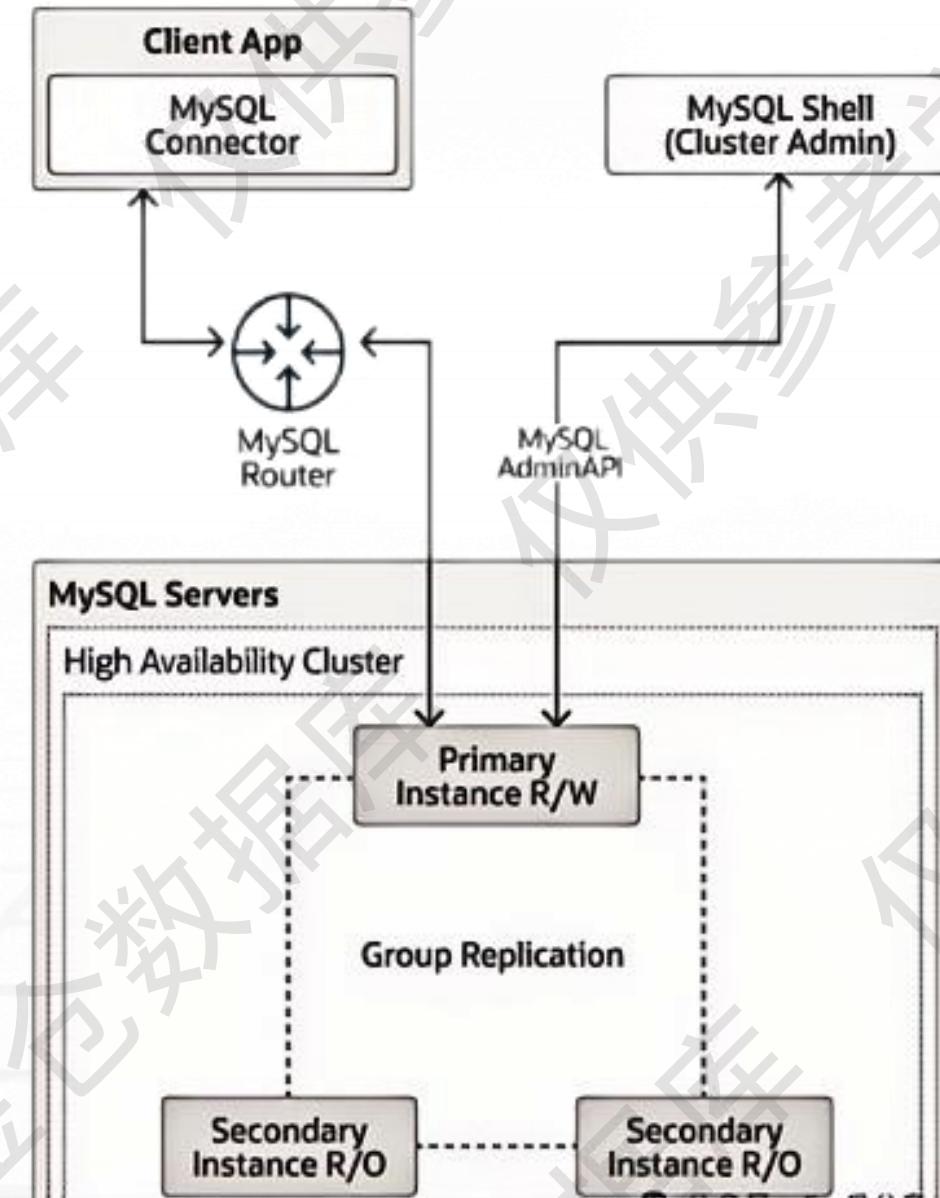
抄袭了互联网公司架构，设计中引入的复杂性，早晚还是会以运营成本的方式给予回报的。

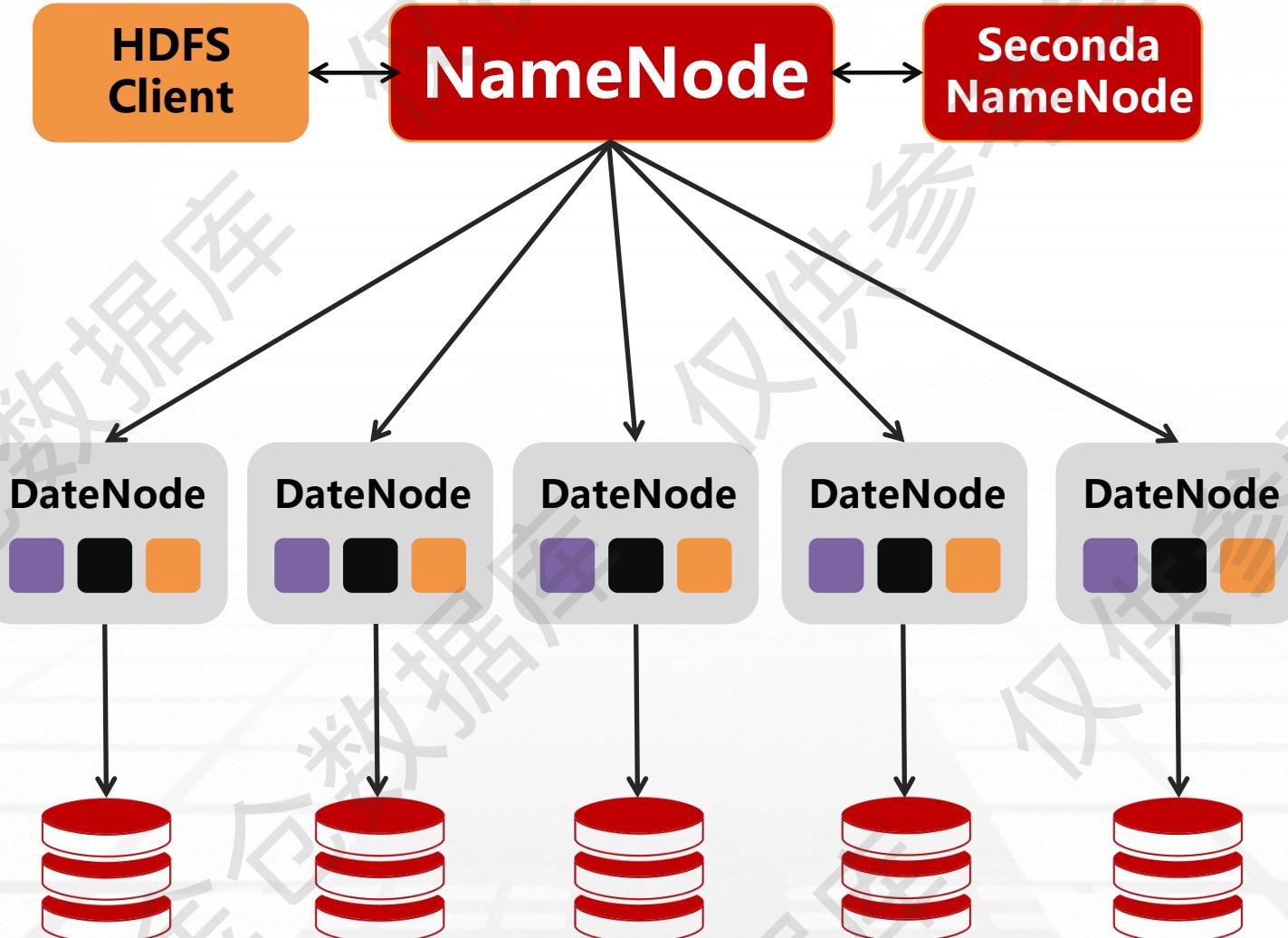


KingbaseES Clusterware 共享存储集群



- 互联网场景的确不太需要RAC（即使这样阿里去IOE之前是拥有全国最多节点的RAC）。
- 互联网公司多还是非互联网公司多？
- 有人说替换Oracle是因为有存储过程。那么没有存储过程就容易替换吗？
- 这些场景从理论模型上就不适合分布式数据库。无论是跨节点访问数据还是跨节点关联数据，对数据库的冲击都比RAC架构下大。



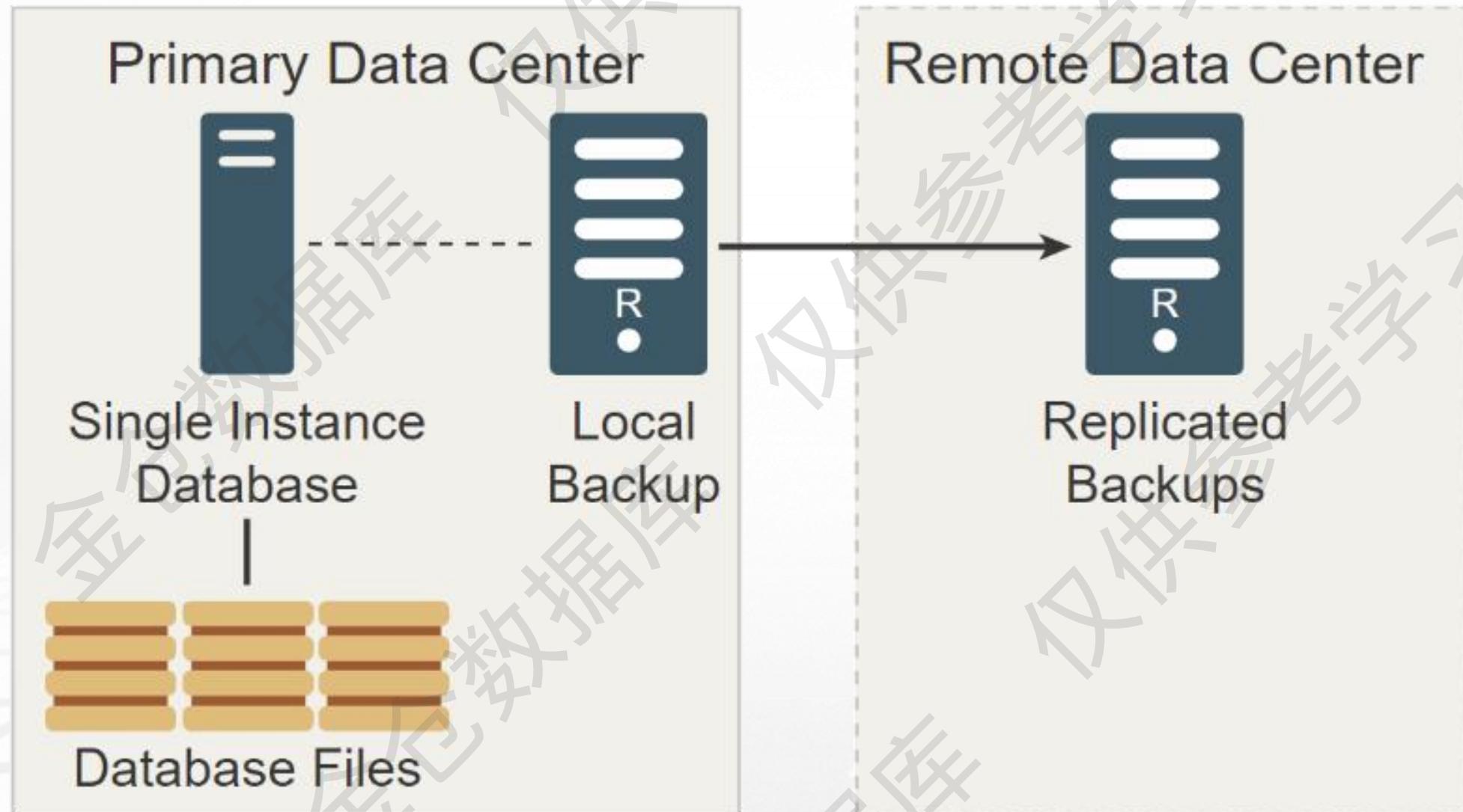


◆ 对MapReduce的批评

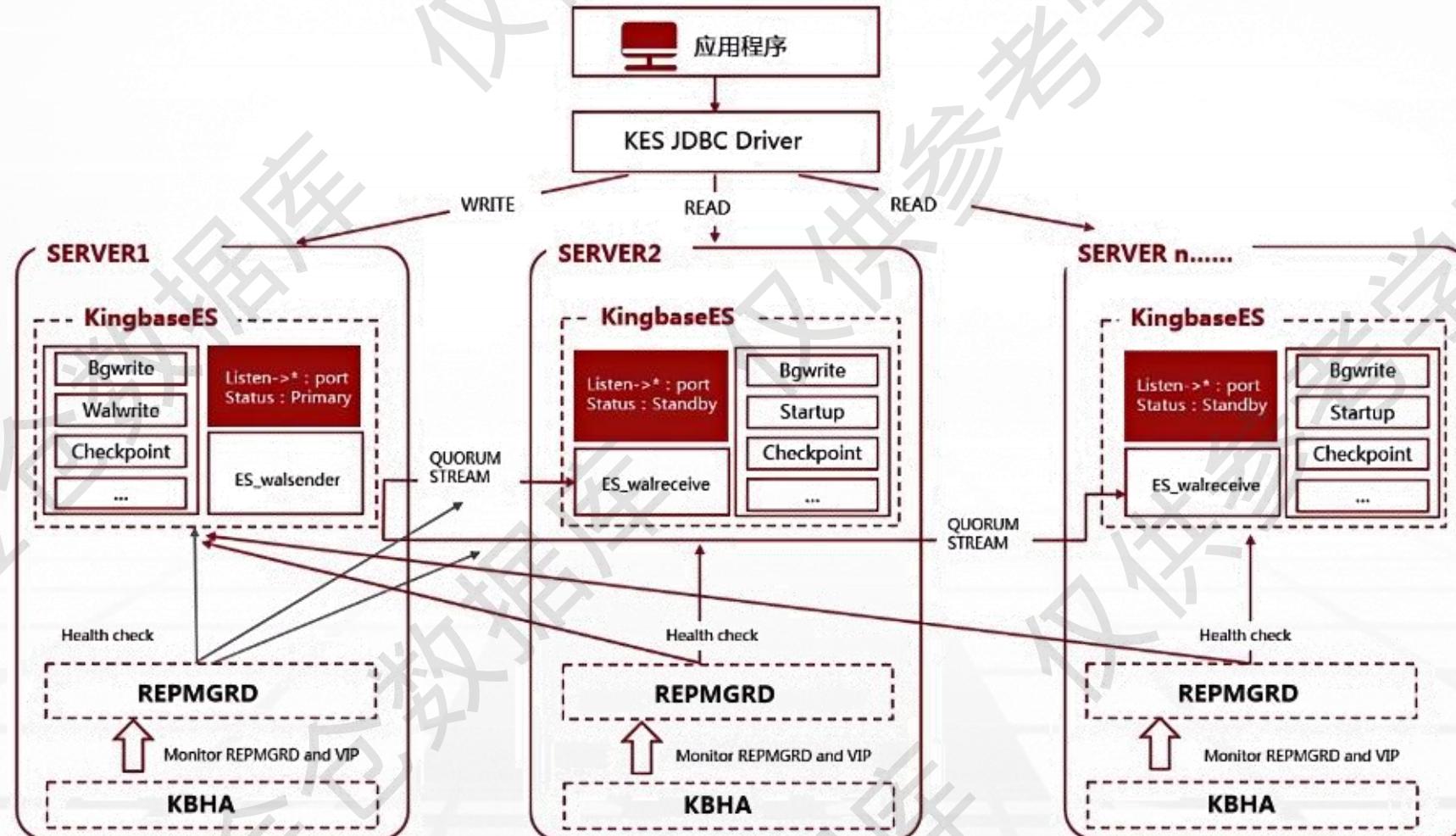
The Great MapReduce Debate: “MapReduce: A Major Step Back” ,
Database Column Blog, Jan.17,2008

- 大规模数据密集型应用编程模型的一个巨大的倒退
- 采用蛮力搜索而非索引的次优的实现
- 没创意：25年前已熟知技术的一个实例而已
- 缺少当前的DBMS的很多功能
- 与现有DBMS工具不兼容



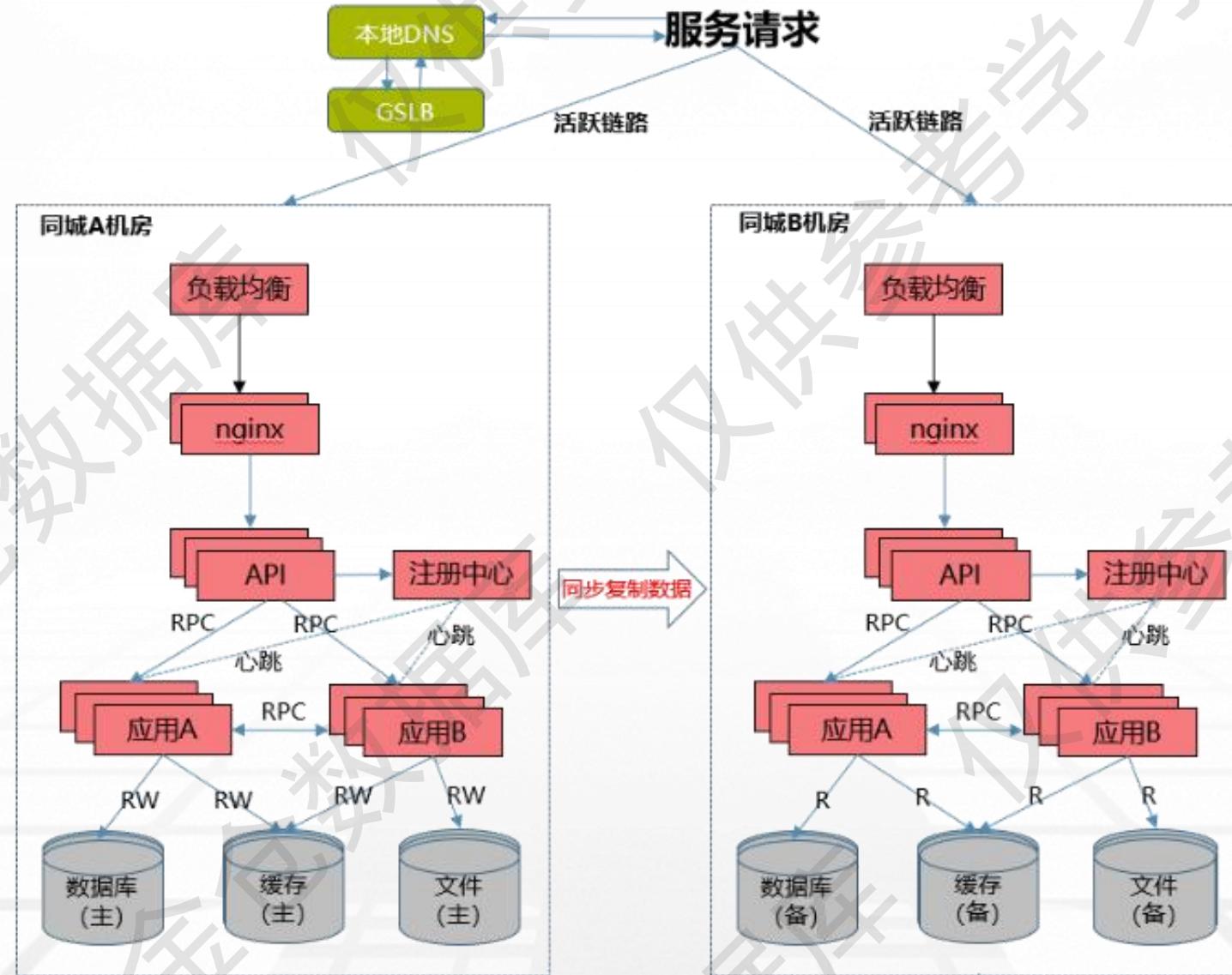


KingbaseES 读写分离集群架构



高可用--同城异地灾备基本架构和成本

KING BASE | 金仓社区



➤ 可靠性99.99

可靠性是随着机器的增多而?

>>>



Oracle的多写架构下只有RAC



MySQL的MGR官方不推荐多主



多写场景下网络是一个重要影响因素



CAP理论

- 高可用能解决锁吗?
- 高可用能解决性能问题吗?

吐槽与期待

- SQL兼容性100%，那么同等硬件性能是不是也是100%？
- 一个SQL几十万行？小意思
- 一个SQL50M？不稀奇
- 一个SQL20个表关联？平均水平
- 没有存储过程好替换？SQL超过了存储过程

吐槽与期待

- 可观测性足以支持DBA诊断？
- 是DBA水平问题还是给我的数据不足以支持我定位



THANKS

成为世界卓越的数据库产品与服务提供商

