



从金融架构的视角看数据库

张晓通

平安壹钱包

个人介绍

个人信息表:

Id	Name	Sex	Mobile	***	Remark
001	张晓通	M	135****	***	十多年来一直从事保险、银行、支付等相关传统与互联网金融系统的开发与设计

活期账户表:

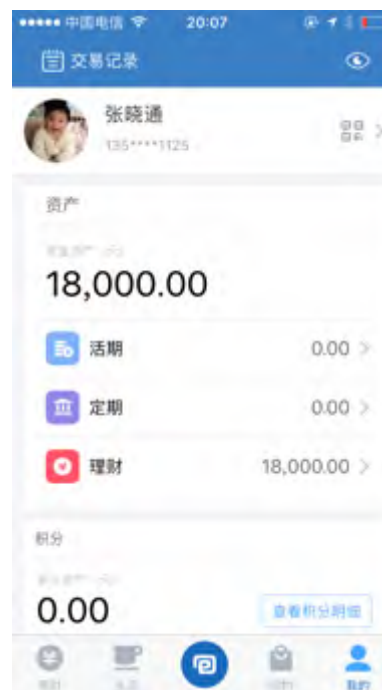
Id	Acct	Amount
001	01001	0.00

定期账户表:

Id	Acct	Amount
001	02001	0.00

理财账户表:

Id	Acct	Amount
001	03001	18000.00



从金融系统演进的观点来看数据库

架构演进

数据库默认指关
系型数据库

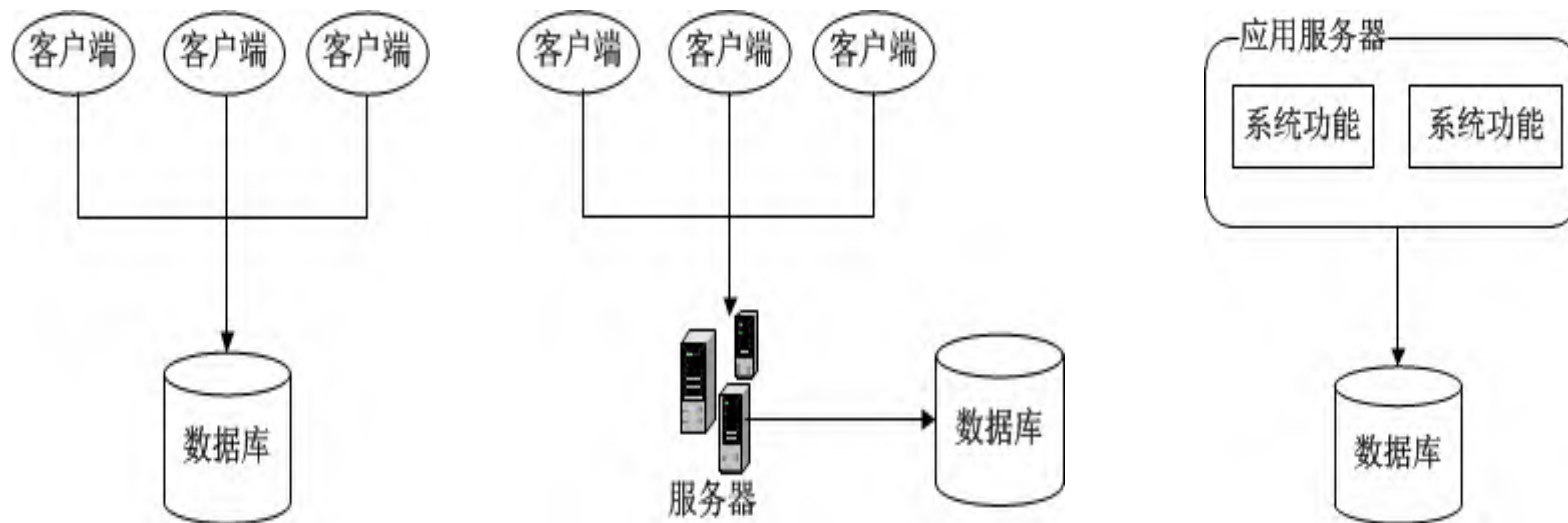
演进观点

对数据库的诉求



架构演进----数据库即系统

- C-S架构和早期的B-S架构



- 系统 ≈ 数据库：系统设计以数据库设计为核心。
- 轻应用重数据库：Database is everything。



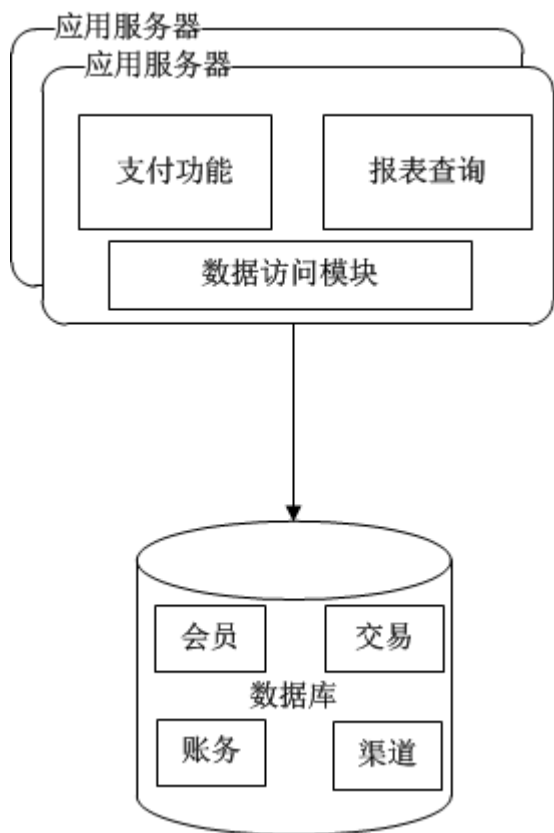
架构演进----数据库即系统

- 我们使用数据库：
 - 建表、建视图、建索引，对数据进行增删查改
 - 建作业、建消息队列
 - 遵守相关范式、使用各种复杂的join
 - 部分厂商甚至提供应用程序的开发工具集
 - 最主要的是：我们优先使用数据库（存储过程等）**完成业务逻辑**
 - 潜规则是：能在数据库上完成的，不在客户端和服务器上完成



架构演进----应用与数据库解耦

• 业务功能逐步从数据库剥离

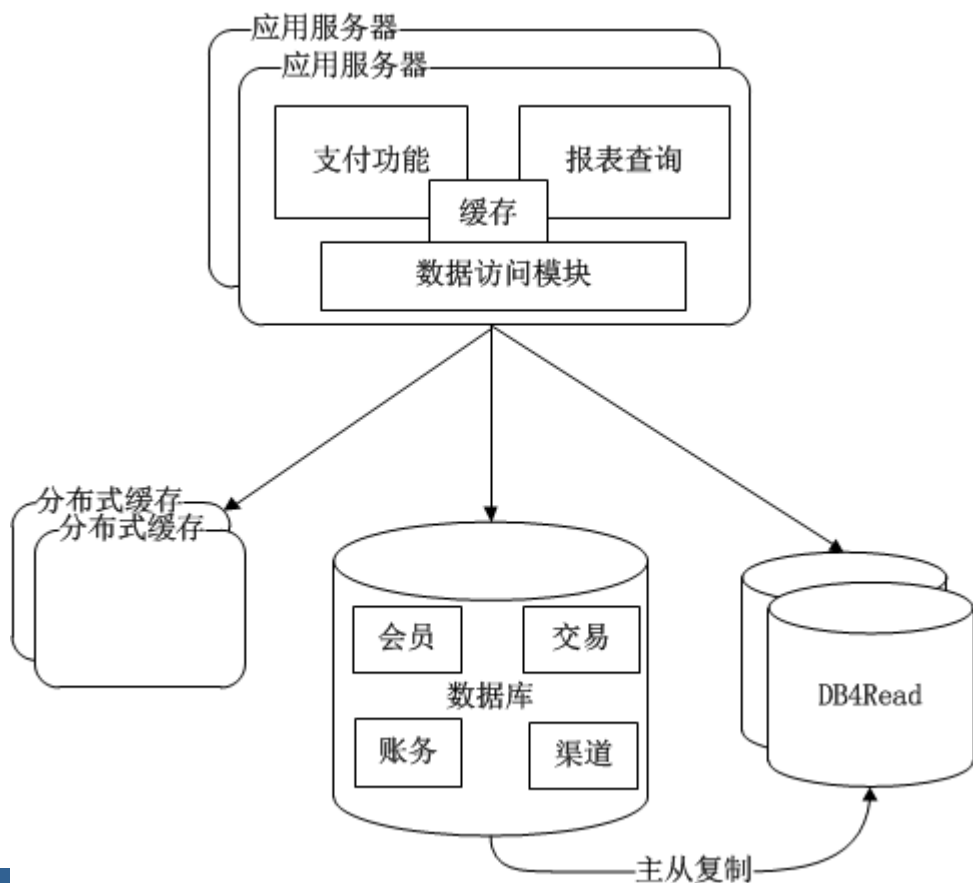


- 应用服务器走向集群，应用的各个功能仍然耦合在一起
- 应用内部由单独数据访问模块负责和数据库的交互
- 部分轻应用重数据库型系统仍然大量使用存储过程进行业务处理
- 其余系统业务功能逐步从数据库剥离到应用服务器上
- 应用不再建议使用数据库存储过程
- 应用在数据库上做各种复杂join操作



架构演进----数据库读写分离

• 缓存+数据库读写分离



- 数据库性能问题开始出现
 - 数据库性能不足
 - 连接池不足
- 写数据库只可以写到主库上去
- 使用各种缓存技术加快数据读取，减少数据库访问
- 使用读库分担主库读的压力

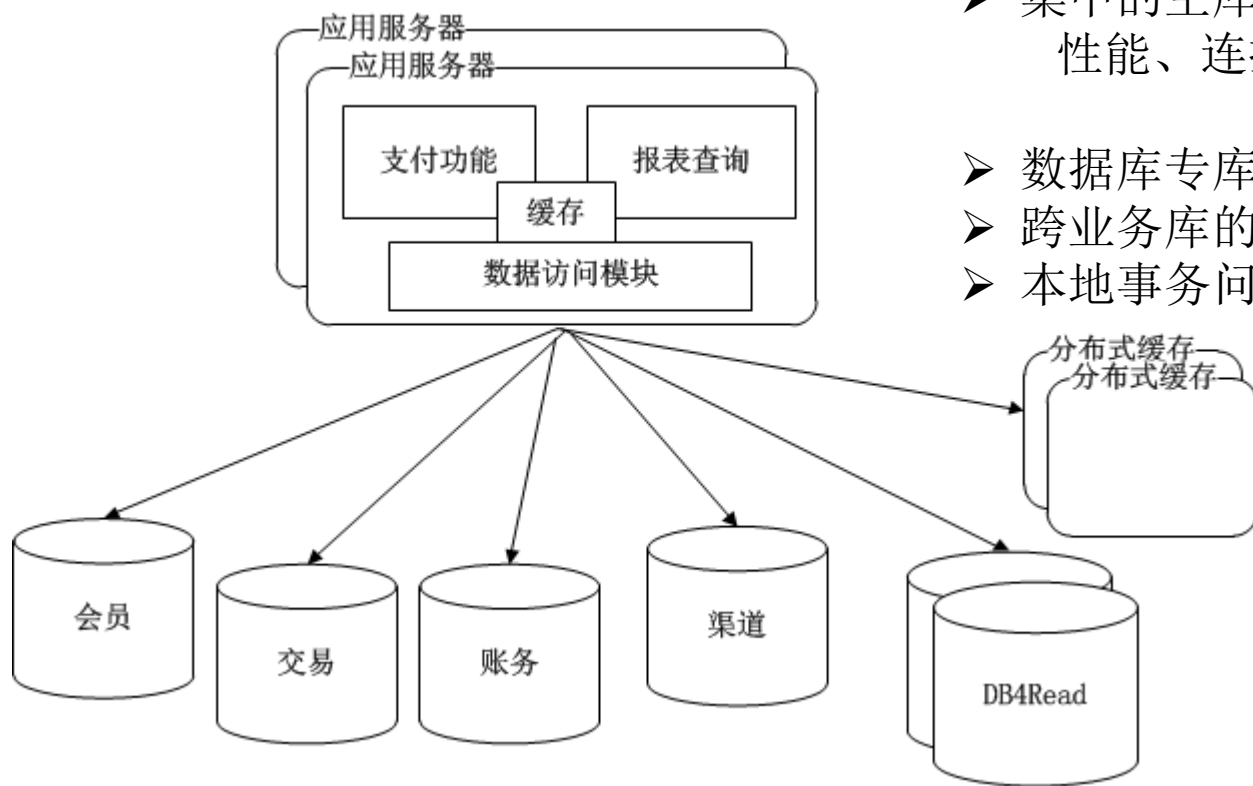


架构演进----数据库垂直拆分

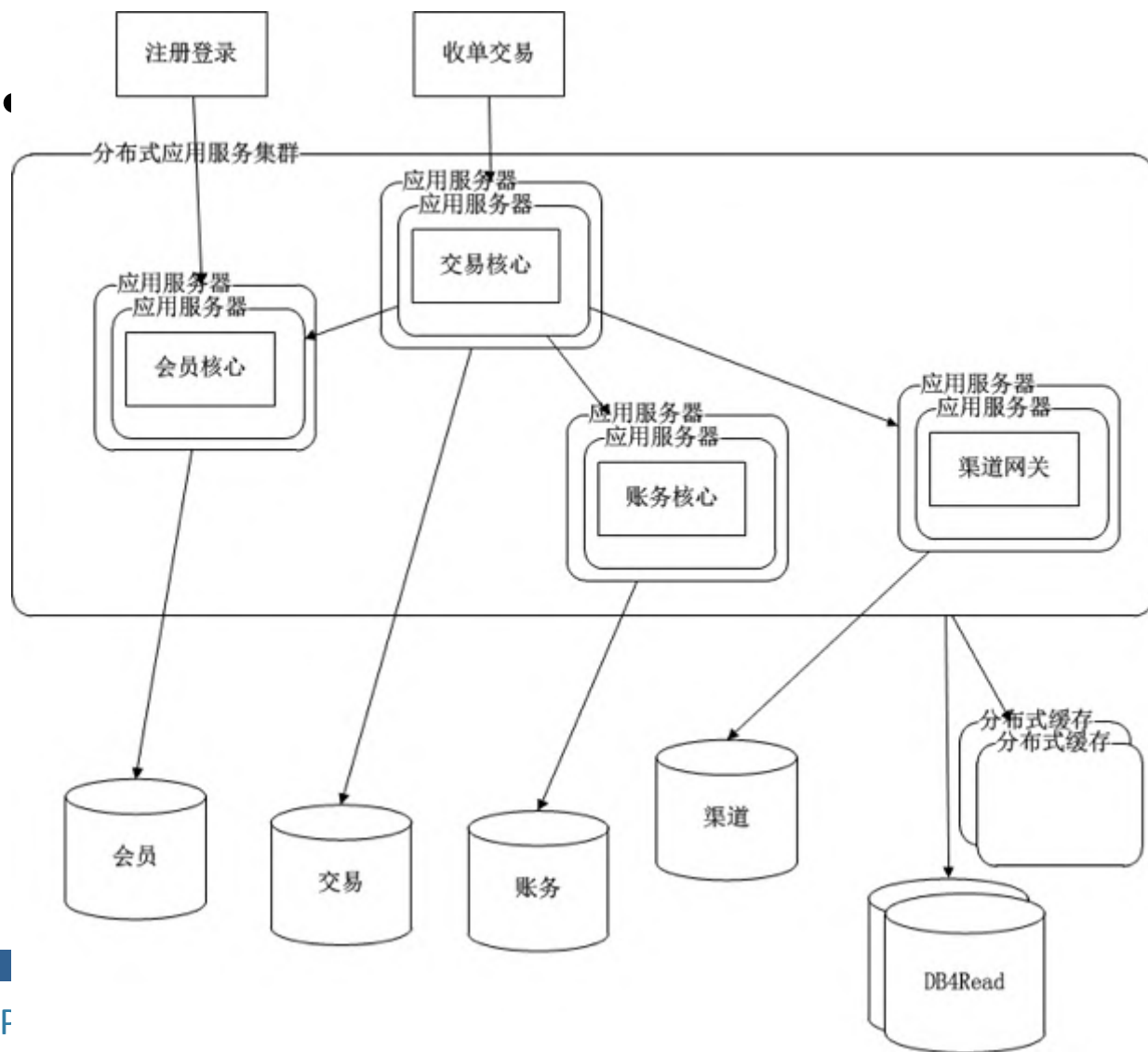
• 数据库垂直拆分

- 集中的主库容易碰到的问题
性能、连接池、业务相互影响

- 数据库专库专用，按业务垂直拆分
- 跨业务库的join不再可以使用
- 本地事务问题



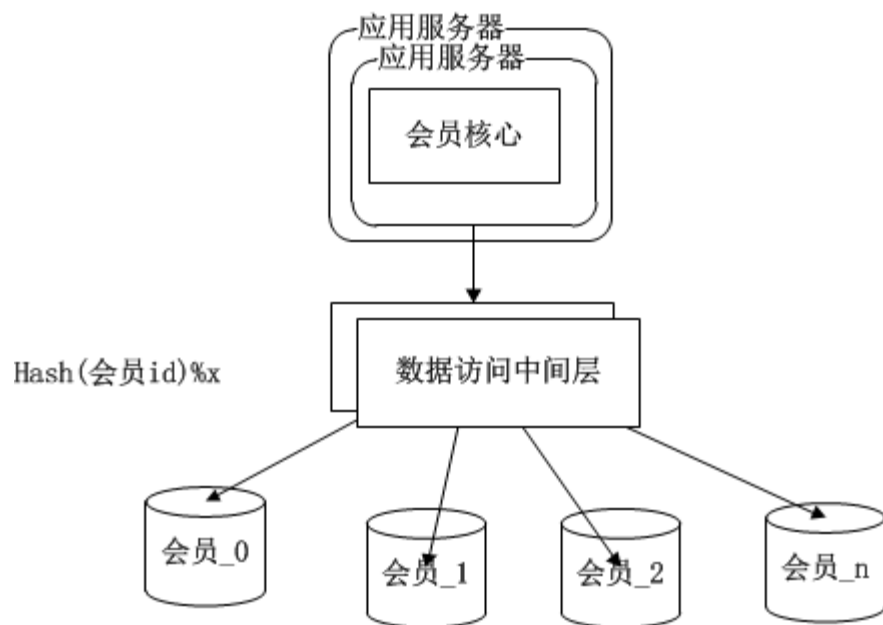
架构演进----应用中心化、服务化



应用系统开始业务中心化、服务化，走向分布式

架构演进----数据库水平拆分

- 数据库走向分布式



- 水平拆分，最好要保证单库的数据是内聚的（主表和辅表在一个库）
- 单库上易于实现的自增序列、外键控制、跨库join等问题要依赖应用系统或数据访问中间层解决
- 一般都引入数据访问中间层，对应用系统有很大侵入性
- 对大部分金融系统来讲，是非常重要的数据库访问方案



架构演进----NoSql数据库使用

- NoSQL数据库：Not only Sql
 - Redis、Hbase等在金融系统中大量使用，改变着相关系统的构建和数据库的使用
 - 关键的夜晚场景，NoSql并不能取代关系型数据库，而是作为其有益补充存在



架构演进

应用架构和数据库相辅相成，相得益彰；相互促进，共同演进。

应用演进



数据库演进



演进观点（一）

- 数据库是金融系统的最关键组成
 - 数据是金融的核心。
 - 数据库是保护数据的最坚实城堡，数据架构是系统架构的基石。
 - 数据架构相对应用架构更稳定，也更难改变。
 - 将数据库成为系统的一部分来设计，而不仅仅只是数据的存储。



演进观点（二）

- 好的系统架构是演化出来的，而不是设计出来的。
 - 演化过程中，数据库的架构起着关键作用。
 - 好的系统架构一定是数据库和系统相辅相成，相得益彰的架构。
- 没有最好的系统架构，只有最合适的系统架构。
 - 金融行业现在仍然存在大量的轻应用重数据库的系统，在数据库上完成关键的业务处理。



演进观点（三）

- 合理使用数据库，充分发挥数据库的优势
 - 数据库成为应用系统设计的一部分，充分利用数据库的特点完成应用功能的开发。
 - 让数据库做最适合其完成的事情，避免不合理的数据库使用。
 - 不要过度设计。



对数据库的诉求

- 可用性与安全性：可用性与安全性是金融系统的重中之中，数据库是最关键一环。
- 可扩展：纵向扩展+横向扩展的支持。
- 性能与稳定性：尤其是事务场景的性能。
- 低成本：软件成本+人员成本+学习成本。
- 易用型：数据库本身功能的完善与语法的兼容。



结束

- 金融行业使用的数据库仍被被商用闭源产品牢牢统治。



Thanks!

Q & A