SDCC 2017深圳站
互联网应用架构实战峰会

CSDN

乾坤

领域模型+内存计算+微服务的协奏曲

主讲人：深蓝. 陈秋余

个人简介

陈秋余，深蓝

原卷皮网架构研发总监，现支付宝高级专家。

资深架构师&技术管理专家，十多年传统企业应用与互联网从业经验。

曾主导**中国移动集团**、**南方电网**IT战略架构规划；

负责**卷皮网**从0到1的技术架构落地和团队搭建，支撑数亿PV，百万级日订单。

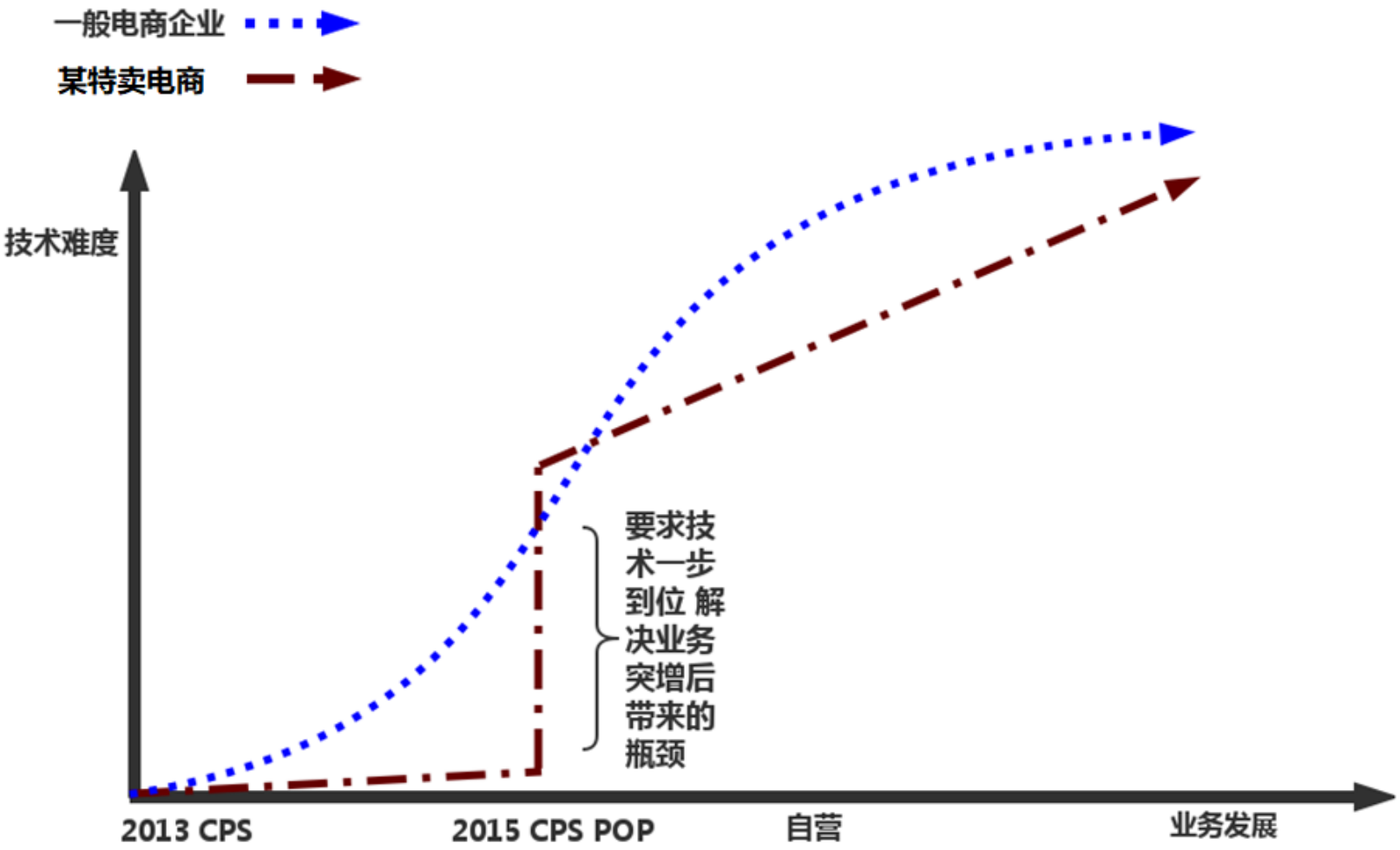
经历过多种类型，不同阶段的技术团队，注重团队组织与技术架构融合互促，对敏捷有深刻理解。

在**TOGAF**，**SOA**以及**微服务**领域有相当建树。

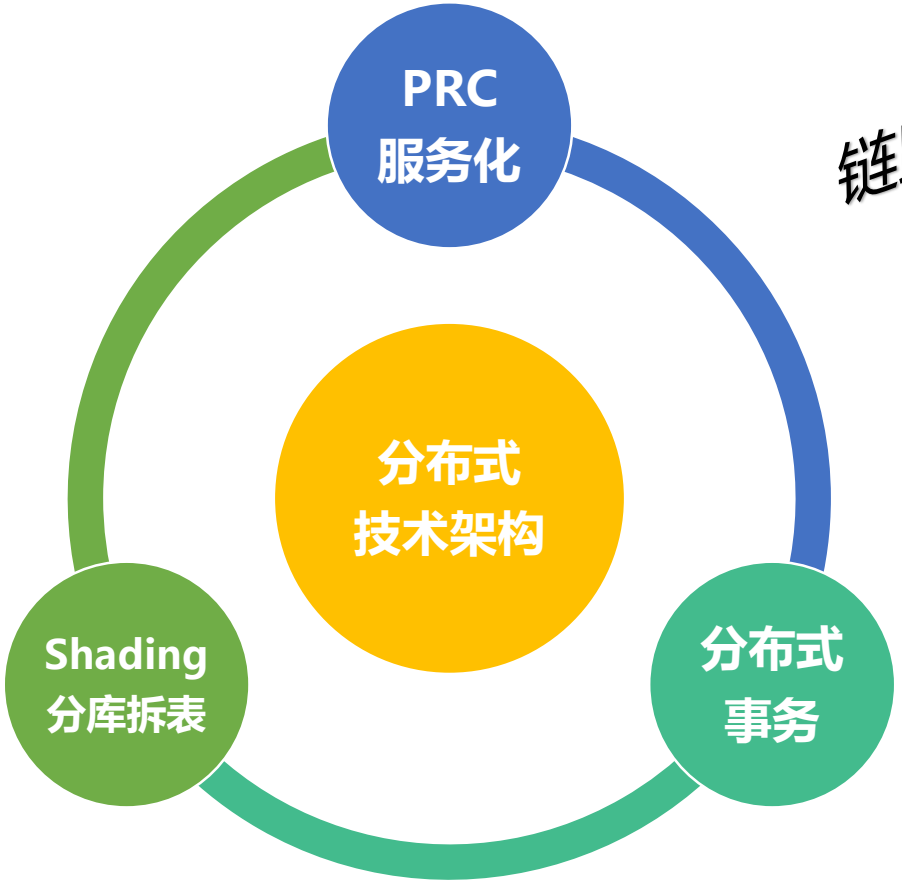
什么是乾坤？

- 特卖类电商的核心架构体系
- 特卖类电商场景和技术特点？
  - 早10点，晚8点特惠商品上架，类似全场秒杀场景
  - 加购物袋同时必须实时扣库存，超卖容忍度极低
- **技术特征&指标**
  - **非主流**分布式交易系统架构：非RPC体系，类事件驱动架构；
  - 超过**150万**日订单量，核心接口TPS峰值**6万+**；
  - 仅用**20台**应用服务器、**30多台**数据服务器达成；
  - *压测性能：单应用节点TPS轻松过10万+；

乾坤的诞生


一般电商企业
某特卖电商
技术难度
要求技术一步到位 解决业务突增后带来的瓶颈
2013 CPS
2015 CPS POP
自营
业务发展

当前主流技术架构三板斧及主要困境



链路依赖

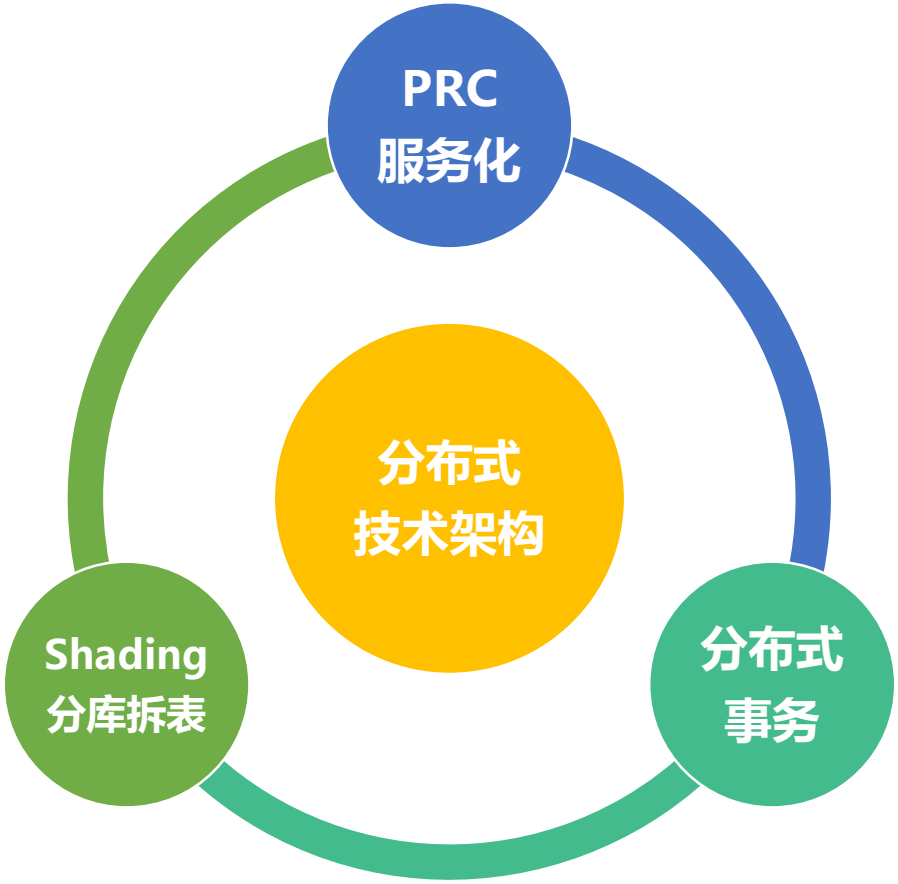
线程&连接阻塞

分布式事务性能堪忧

服务的边界和粒度

微服务 VS 跨域查询

当前主流技术架构三板斧及主要困境



我们有:

- ✓ 2PC->3PC->TCC
- ✓ 数据库中间件->Ocean base
- ✓ Dubbo->熔断器
- ✓ ……

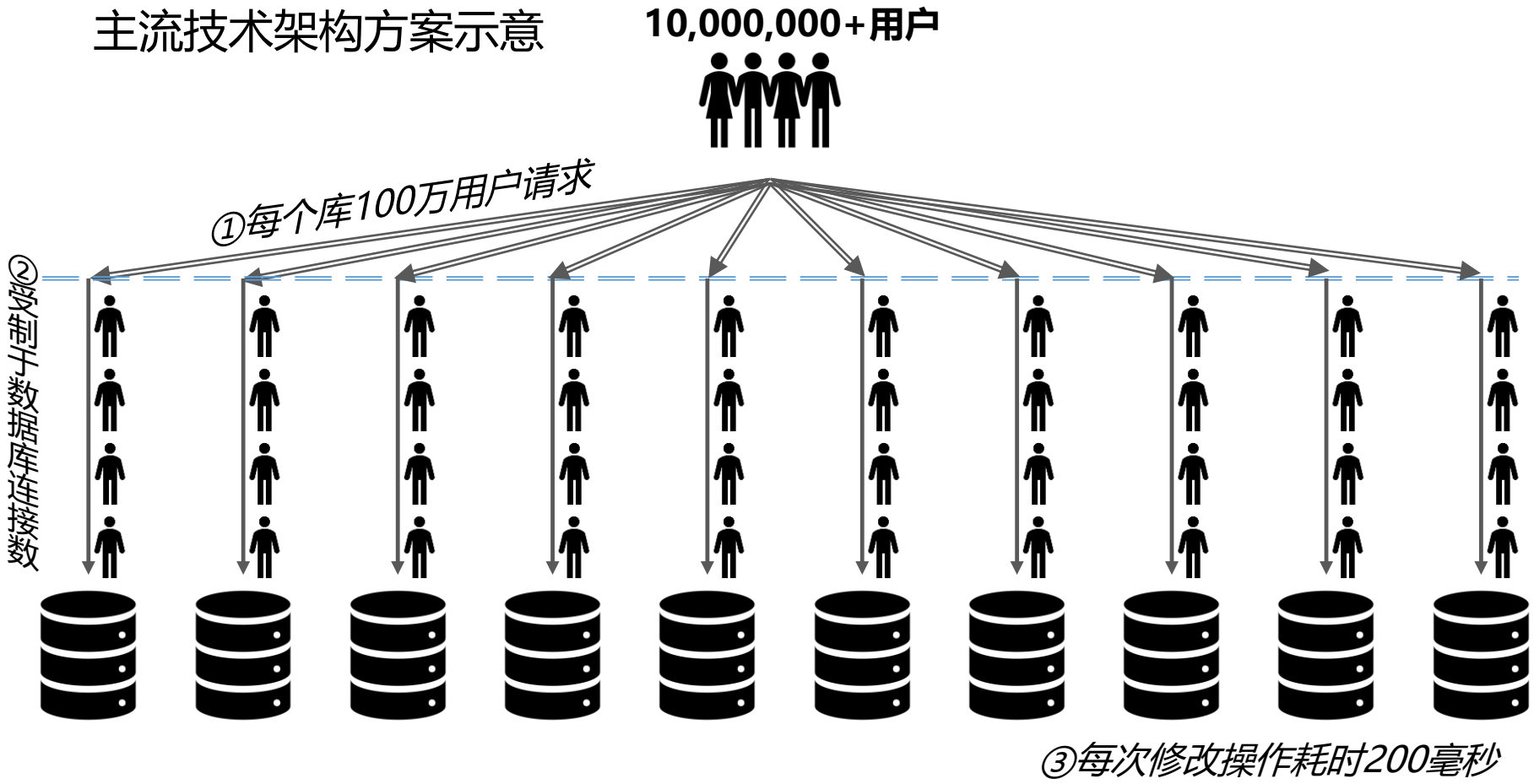
与其一直苦逼补天，不如换个思路吧

乾坤的架构哲学

1. 解决高并发争抢问题的最好办法就是完全避免争抢的发生；
2. 解决分布式事务问题的最佳选择就是根本不用分布式事务；

从1000万用户并发修改用户资料的假设场景开始

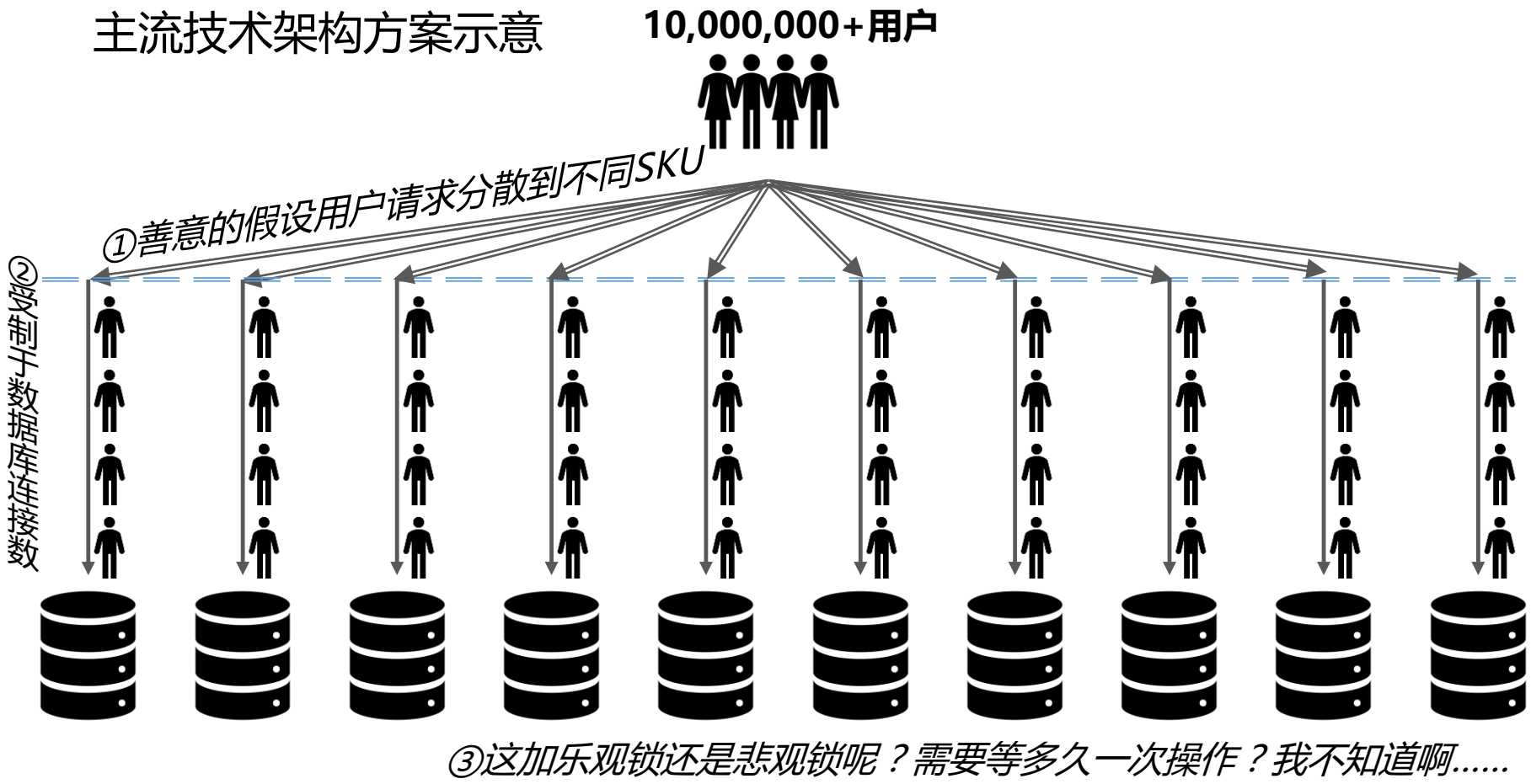
主流技术架构方案示意



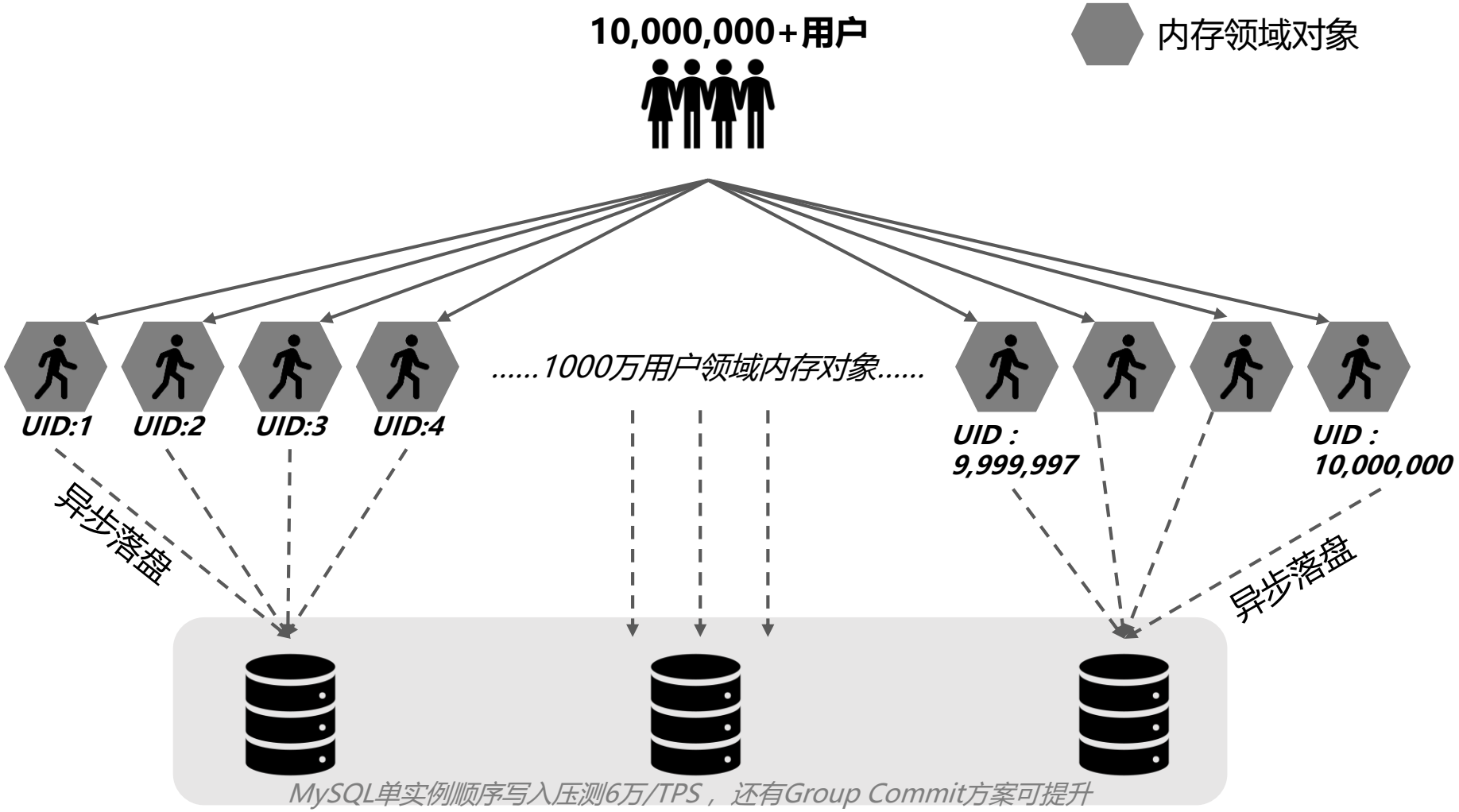
高并发≠高并发争抢，两张图你就懂了

改成1000万用户疯抢200万SKU的商品会咋样？

主流技术架构方案示意



乾坤又是如何解决高并发的性能问题



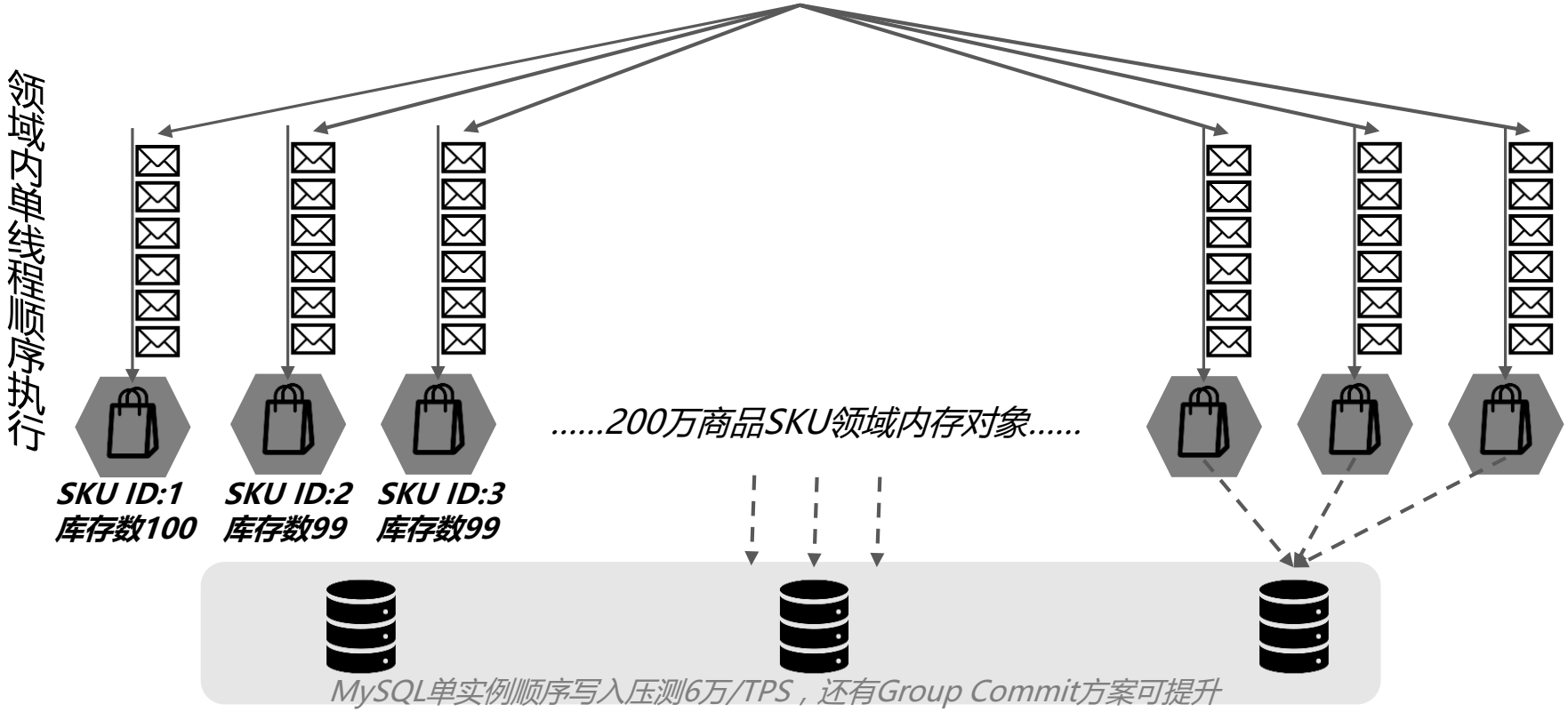
乾坤又是如何解决高并发争抢的性能问题

领域对象上，
单次请求处理时间：**微秒级**

10,000,000+用户

内存领域对象

无竞争争抢



内存领域对象 VS. 分布式缓存

从技术实现角度看：

1. 额外的性能开支；
2. Redis单线程模式的优势与劣势都很明显；
3. 数据结构的灵活性；

从架构和设计角度看：

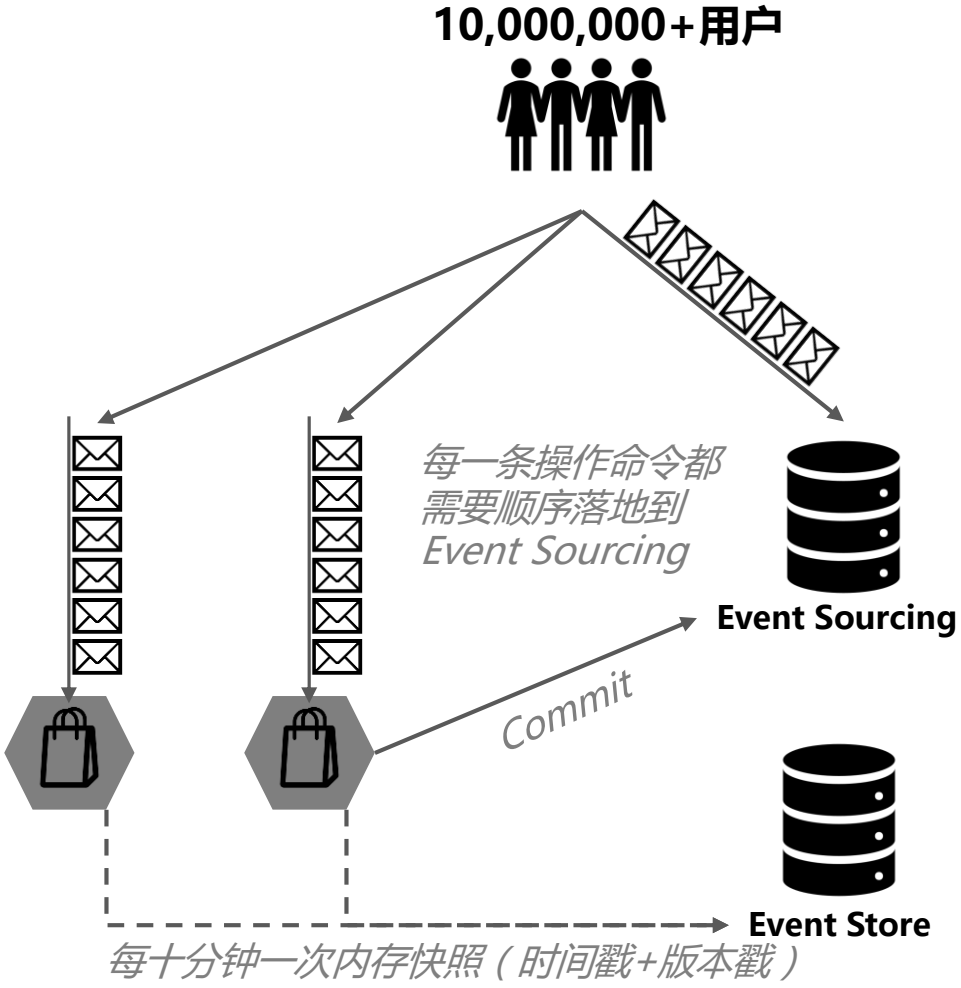
领域模型和表模式的区别

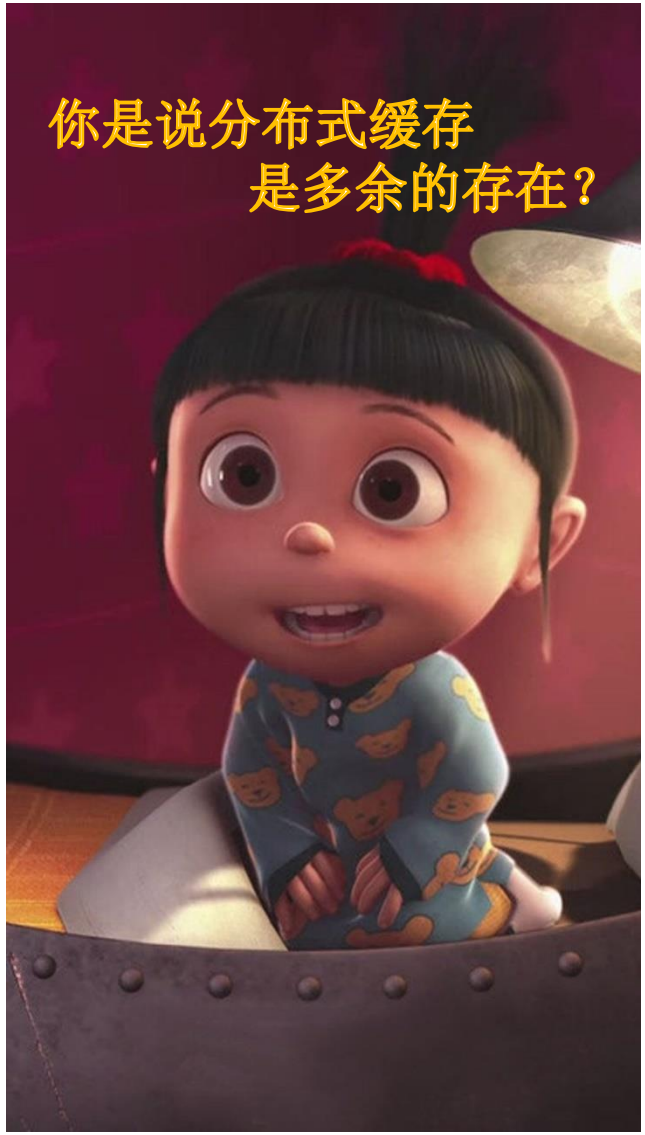
- 这个……
- 那个……还有……

你单独请我喝咖啡，我们再秉烛夜谈吧……

业务逻辑应尽可能少的被技术落地姿势约束，
业务逻辑不应依赖或受制于技术实现的基础设施！！

内存领域对象的高可用保障



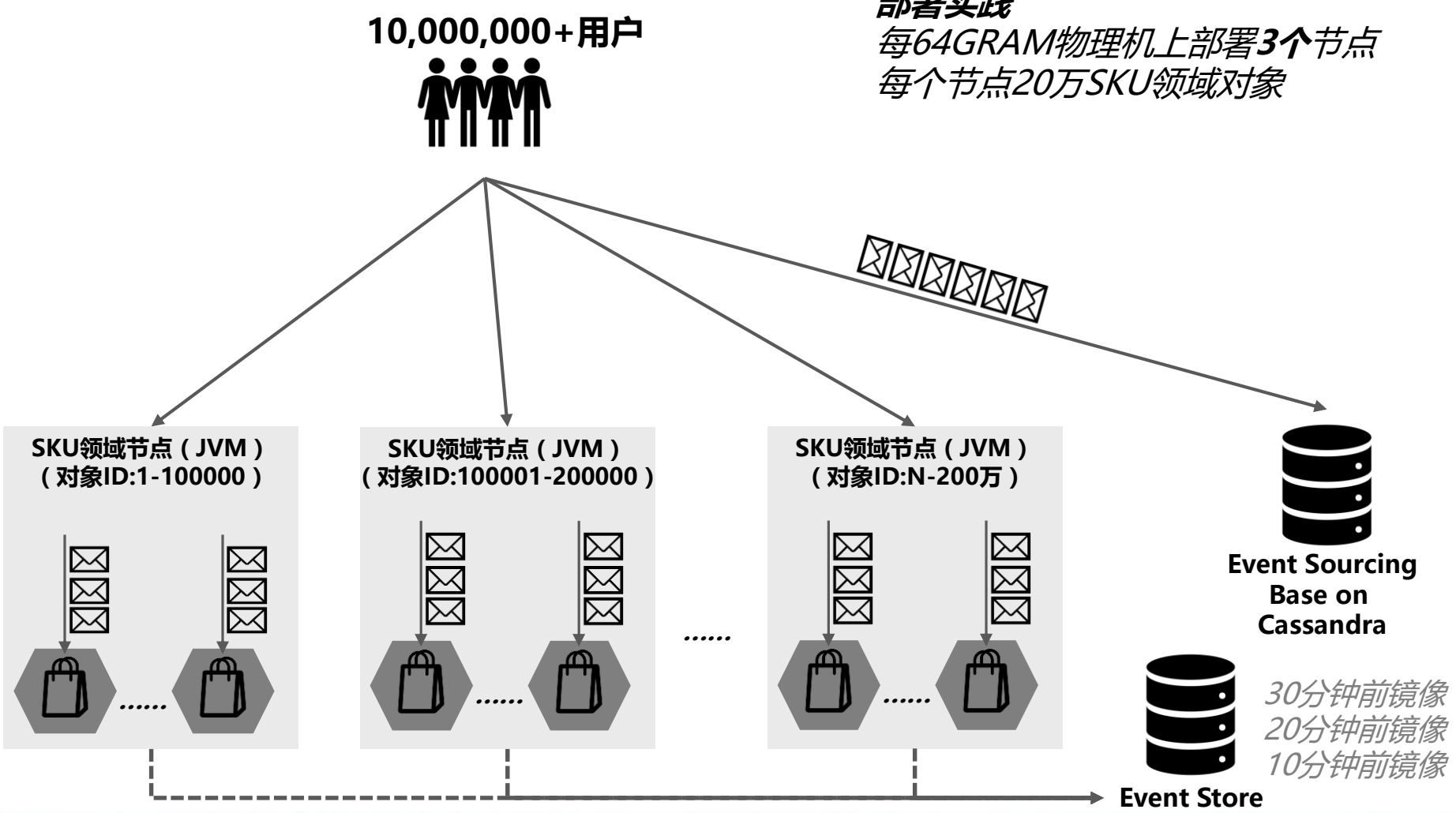


内存领域对象的高可用保障

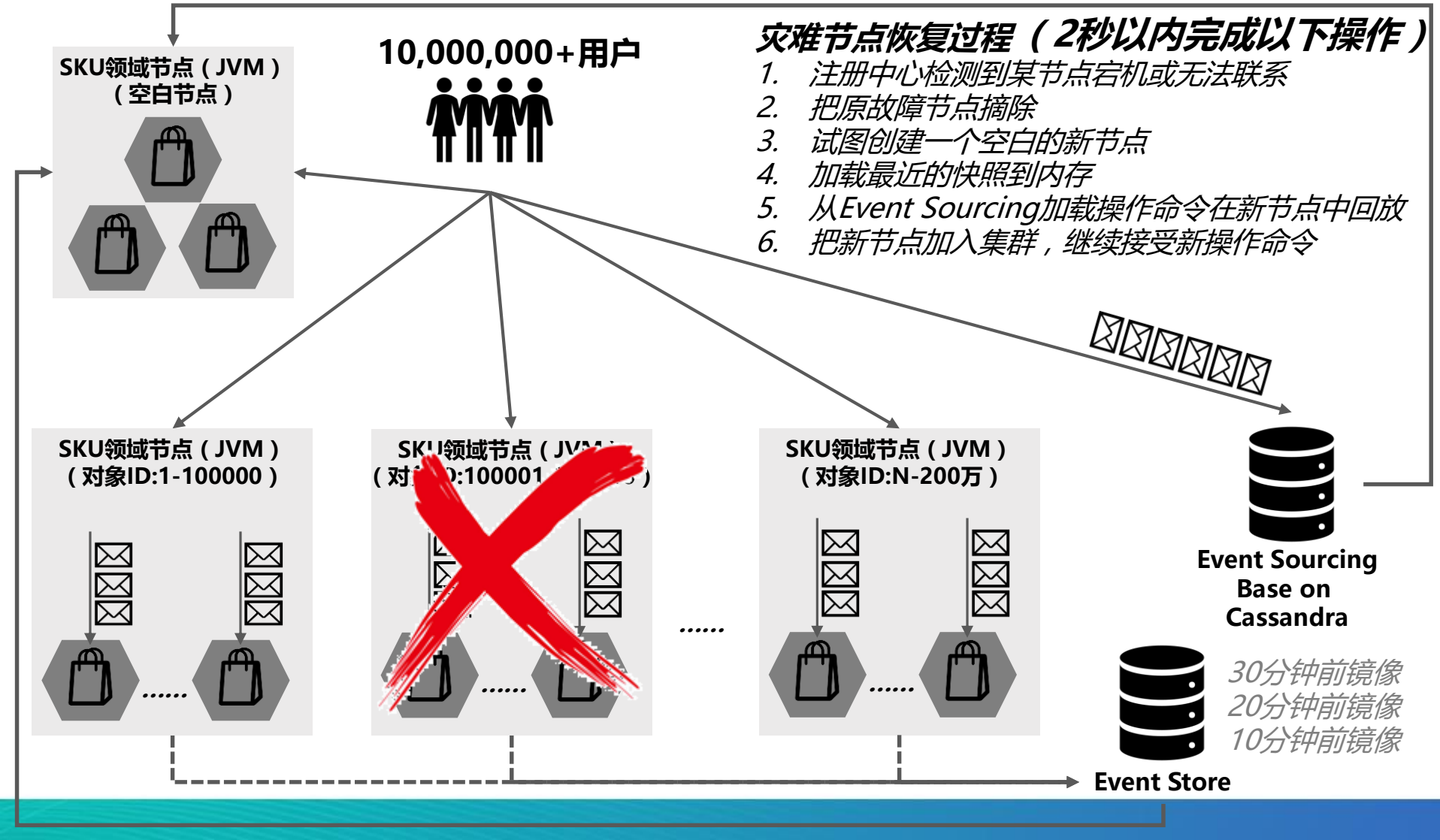
部署实践

*每64GRAM物理机上部署**3个**节点*

每个节点20万SKU领域对象

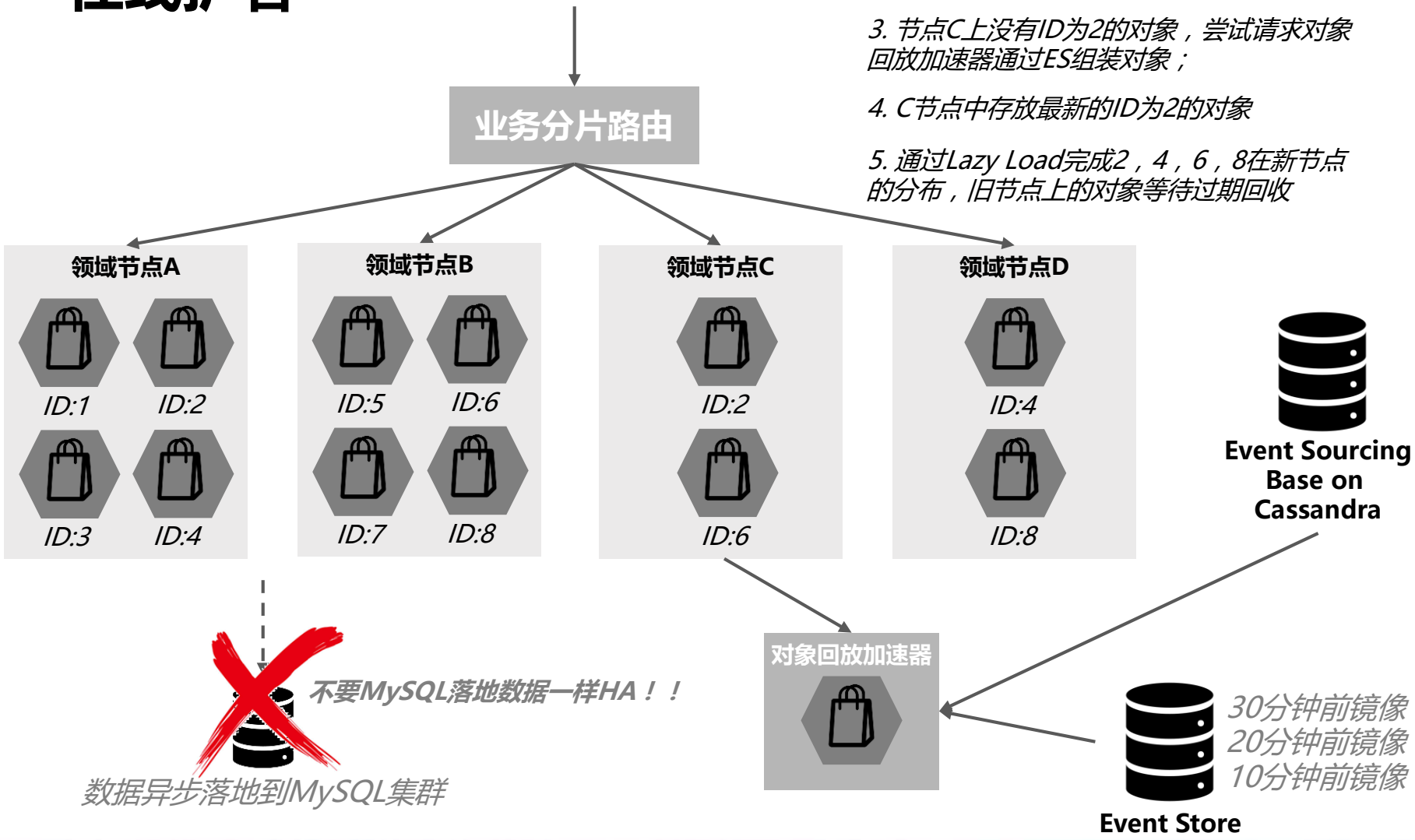


内存领域对象的高可用保障

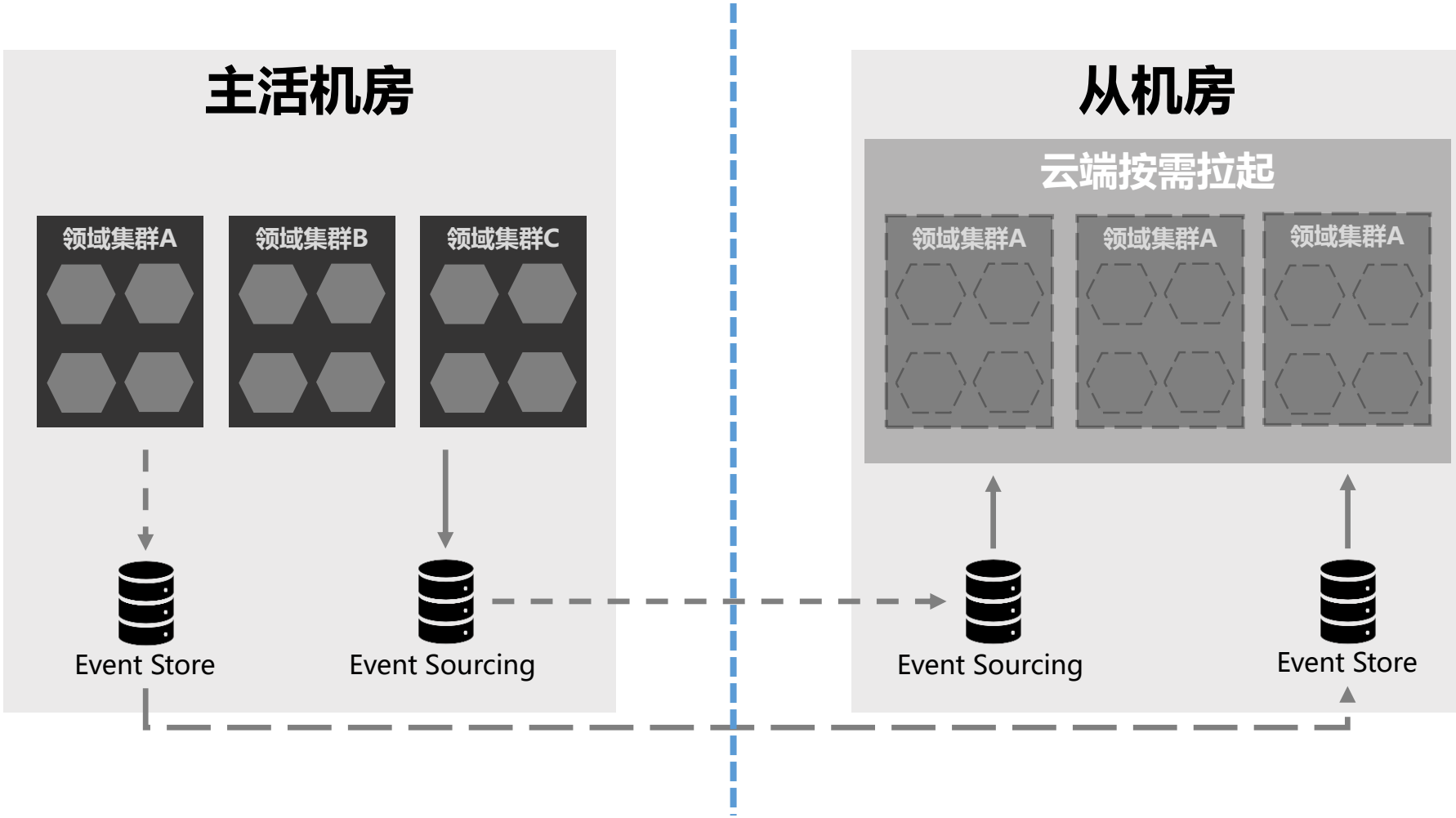


在线扩容

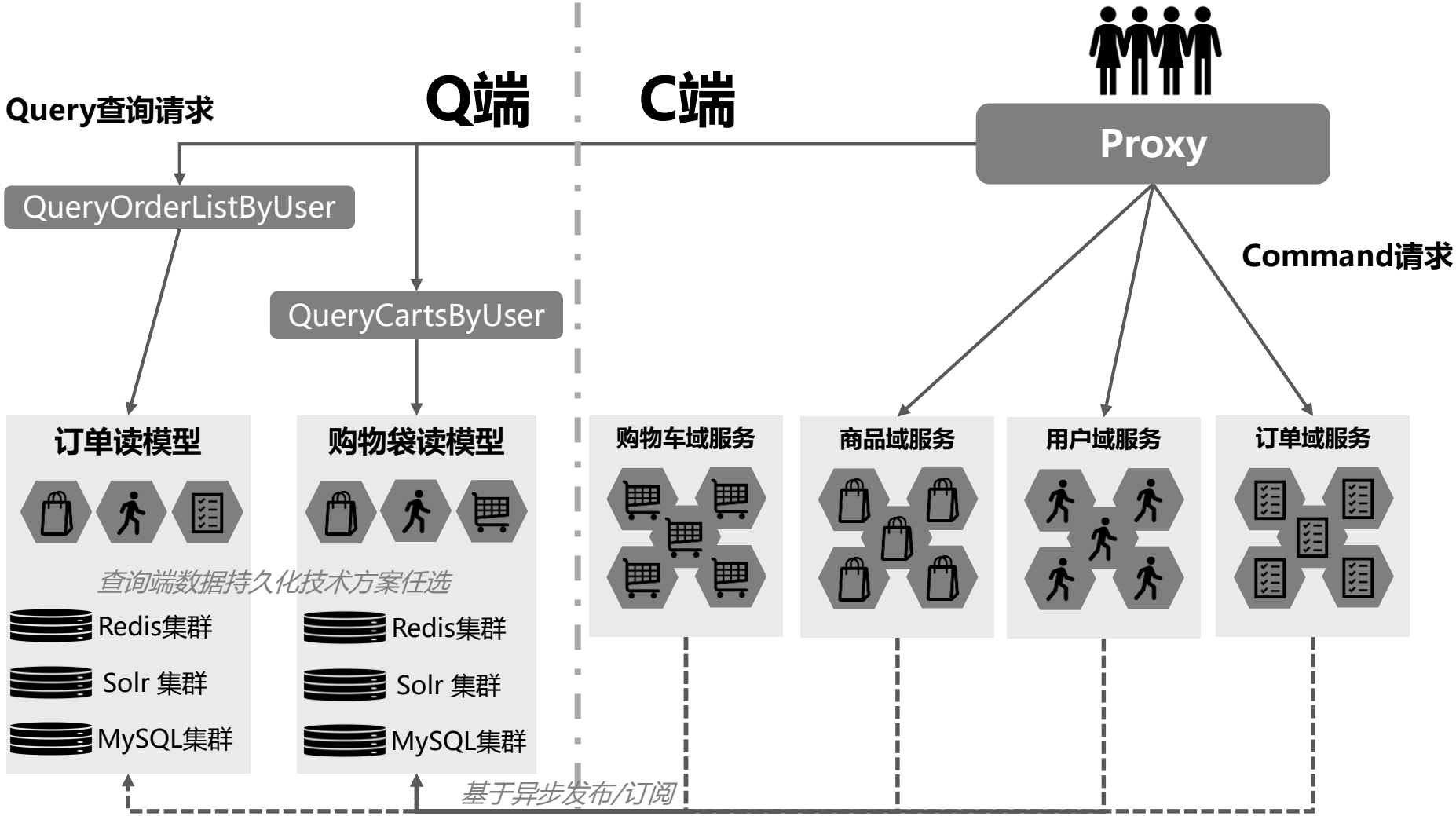
1. 访问ID为2的对象；
2. 路由到新的节点C，而不再访问节点A；
3. 节点C上没有ID为2的对象，尝试请求对象回放加速器通过ES组装对象；
4. C节点中存放最新的ID为2的对象
5. 通过Lazy Load完成2，4，6，8在新节点的分布，旧节点上的对象等待过期回收



内存领域对象的高可用保障-跨机房

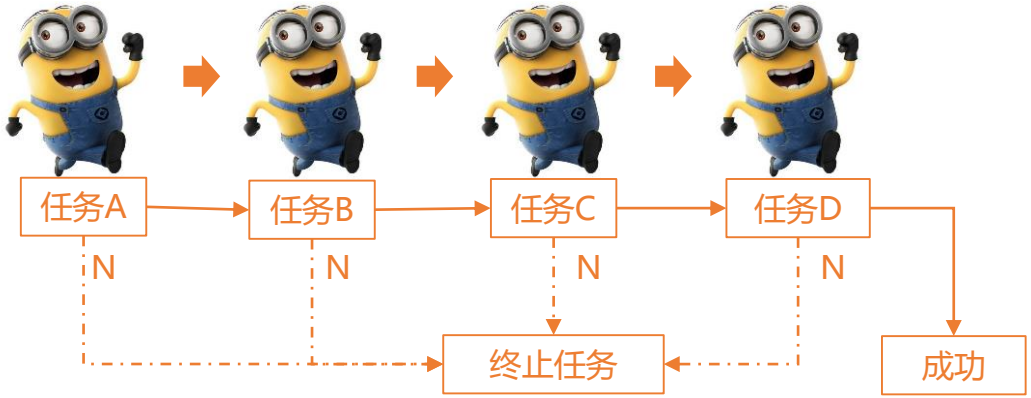


CQRS是破解微服务跨域查询问题的良药

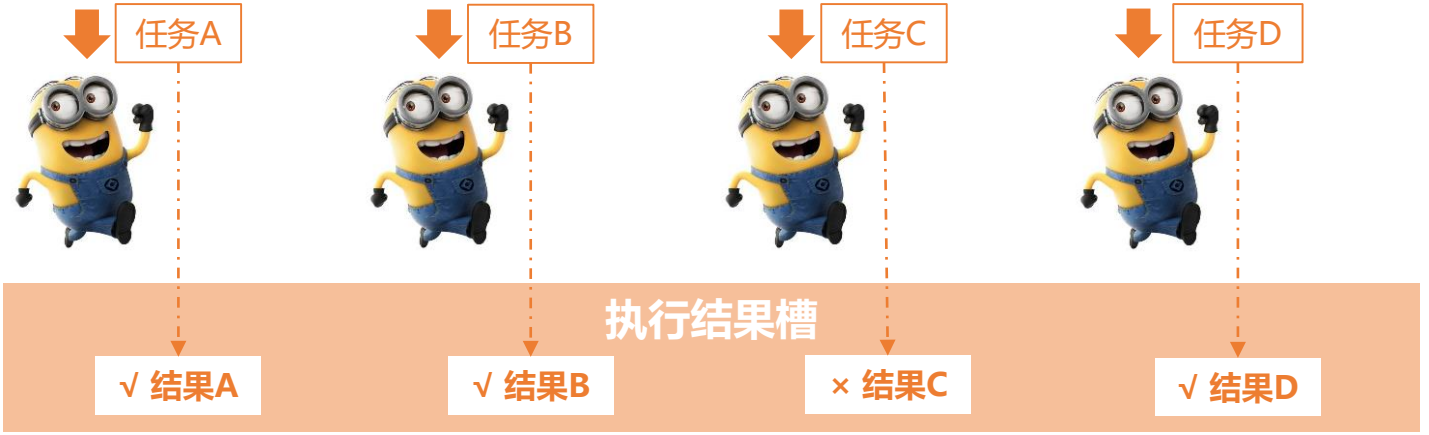


乾坤的最终一致性事务模型

常规事务处理模式



乾坤采用纯异步并发事务处理模式



乾坤的最终一致性事务模型

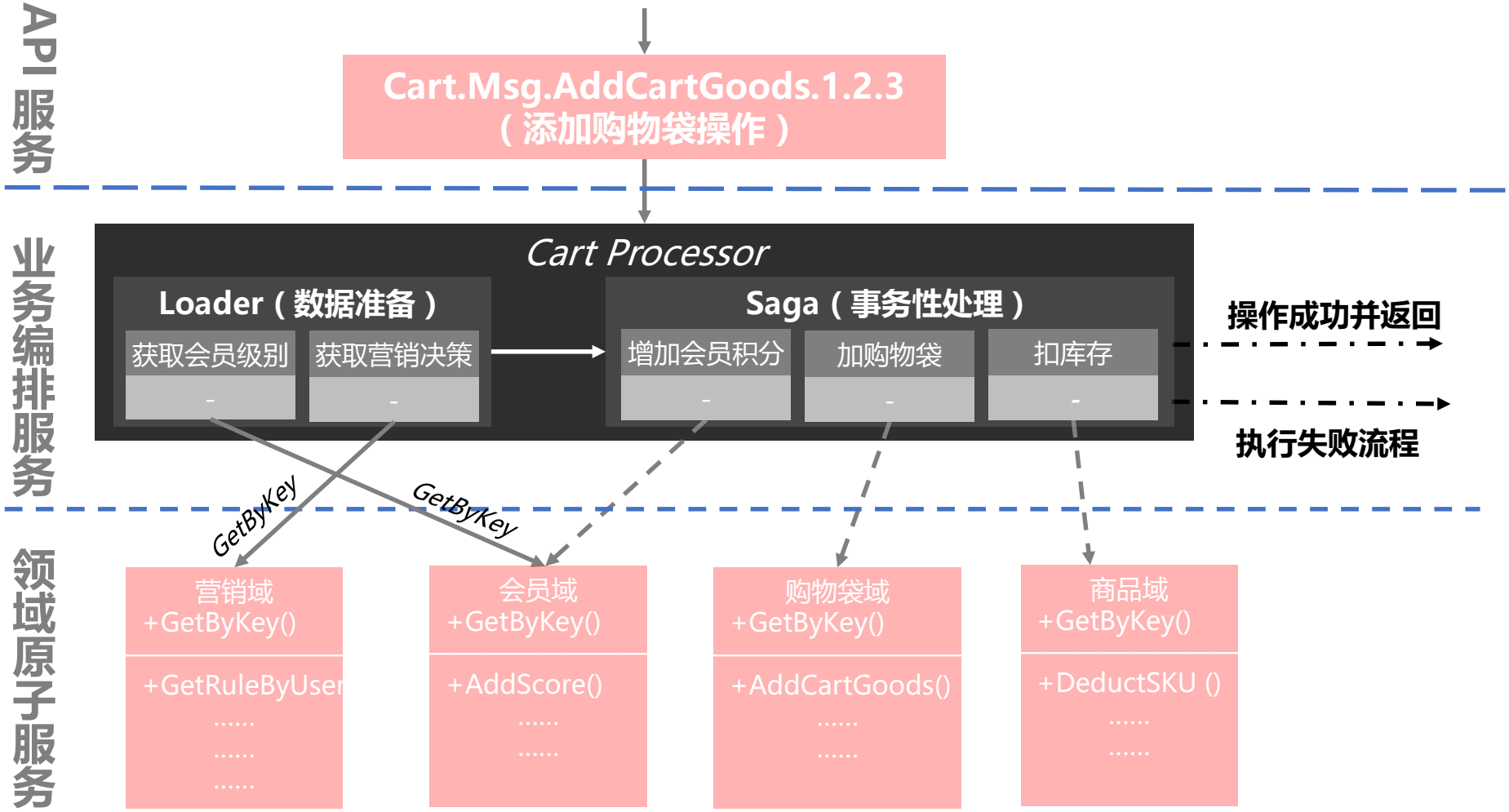
老板的紧急任务

- 强调响应速度，最大并行化；
- 假设一定成功，不依赖其前置任务结果；
- 设定固定执行时间限制（超时设定）；
- 汇总执行结果判定老板的紧急任务是否达成。

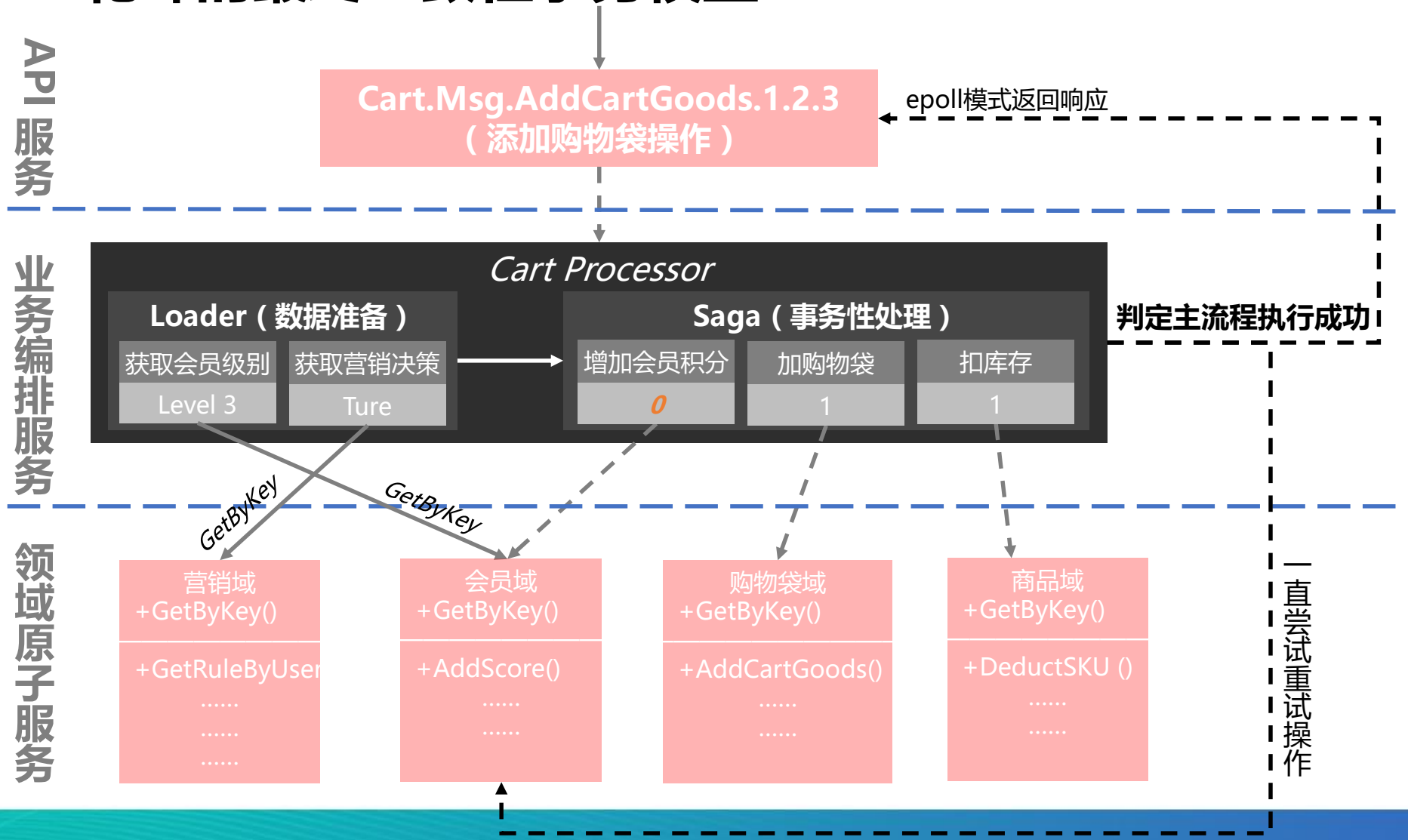
优势与缺陷

- 并行度高，无阻塞，处理响应快；
- 增加额外的无效处理，以及回退操作开支；
- 实现中无法完全做到前后任务不依赖，但数字时间几乎不受这个限制；

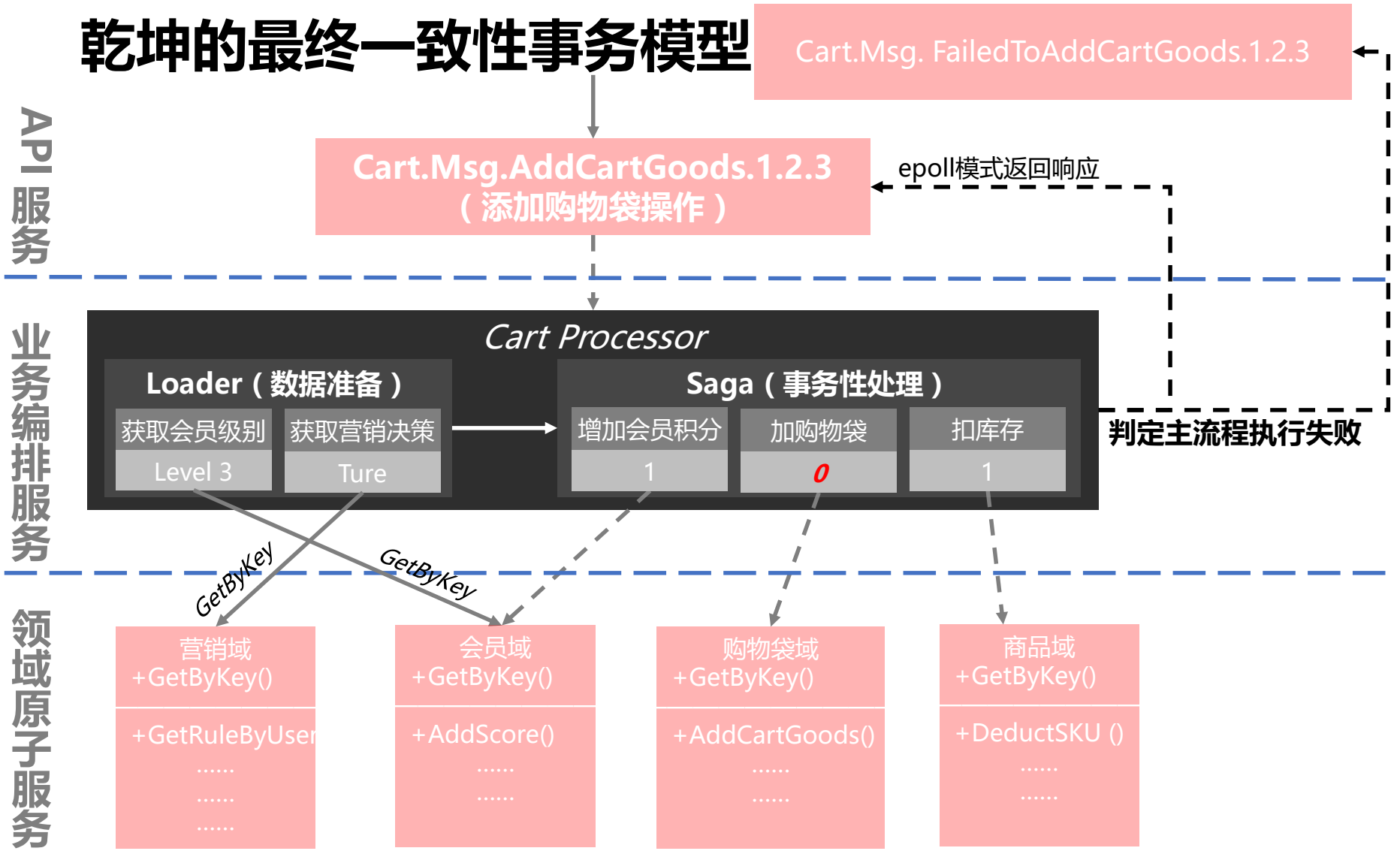
乾坤的最终一致性事务模型



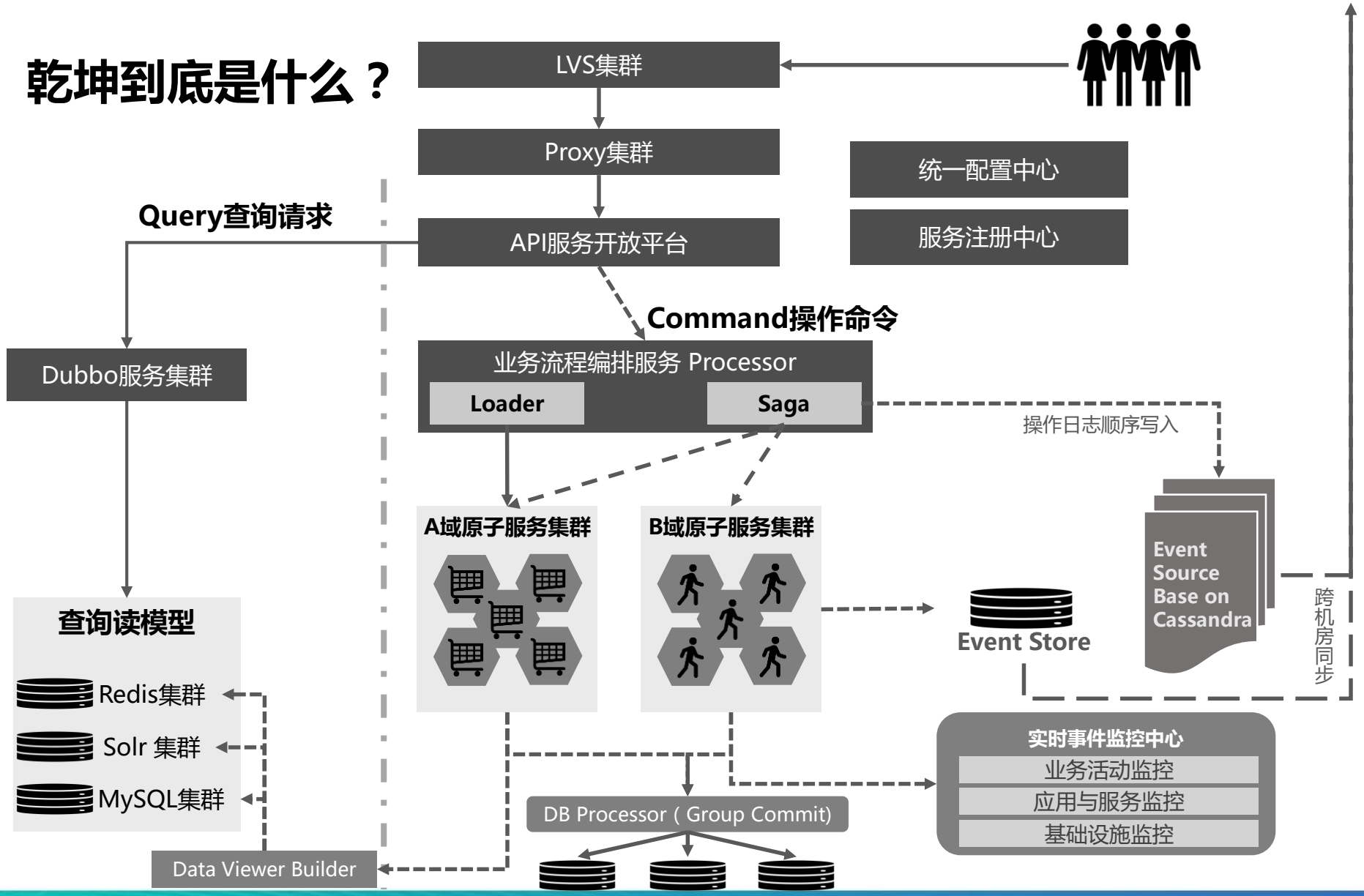
乾坤的最终一致性事务模型



乾坤的最终一致性事务模型



乾坤到底是什么？


LVS集群
Proxy集群
统一配置中心
服务注册中心
Query查询请求
API服务开放平台
Command操作命令
Dubbo服务集群
业务流程编排服务 Processor
Loader
Saga
操作日志顺序写入
A域原子服务集群
B域原子服务集群
Event Source Base on Cassandra
跨机房同步
Event Store
查询读模型
Redis集群
Solr 集群
MySQL集群
实时事件监控中心
业务活动监控
应用与服务监控
基础设施监控
DB Processor (Group Commit)
Data Viewer Builder

乾坤到底是什么？

- **一种领域驱动设计的微服务架构最佳实践落地方案**
  - 高度解耦：读写分离，领域隔离，发布/订阅；
  - 无需关心高并发争抢；
  - 无需关心数据持久化方式；
  - 无需过多考虑分布式事务；
  - 无需过多担心性能、扩展性问题；
  - Actor & Service模式，即领域对象Command & 读模型Query;
- **轩辕在团队的实践经验**
  - 需求分析人员：领域模型、业务流程（正向流程、失败处理流程）；
  - 设计人员：服务接口定义，接口验收；
  - 开发人员：领域对象功能实现、读模型查询实现；

一些扩展话题

1. AWS和Azure Fabric Service提到的Actor & Service模式是云计算的未来；
2. 基于事件机制的应用与业务监控体系；
3. ES体系与区块链技术的亲缘性；
4. 架构参考：Axon Framework, Akka, Disruptor，ENode；

常见问题Q&A