

WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE



出品人及主持人：

于雪

51CTO WOT大会主编

共享经济下的技术变革

新美大外卖订单系统 实践

何轶@新美大外卖
2017.04



何轶

新美大

餐饮平台外卖订单系统技术负责人

分享主题：

新美大外卖订单系统架构实践

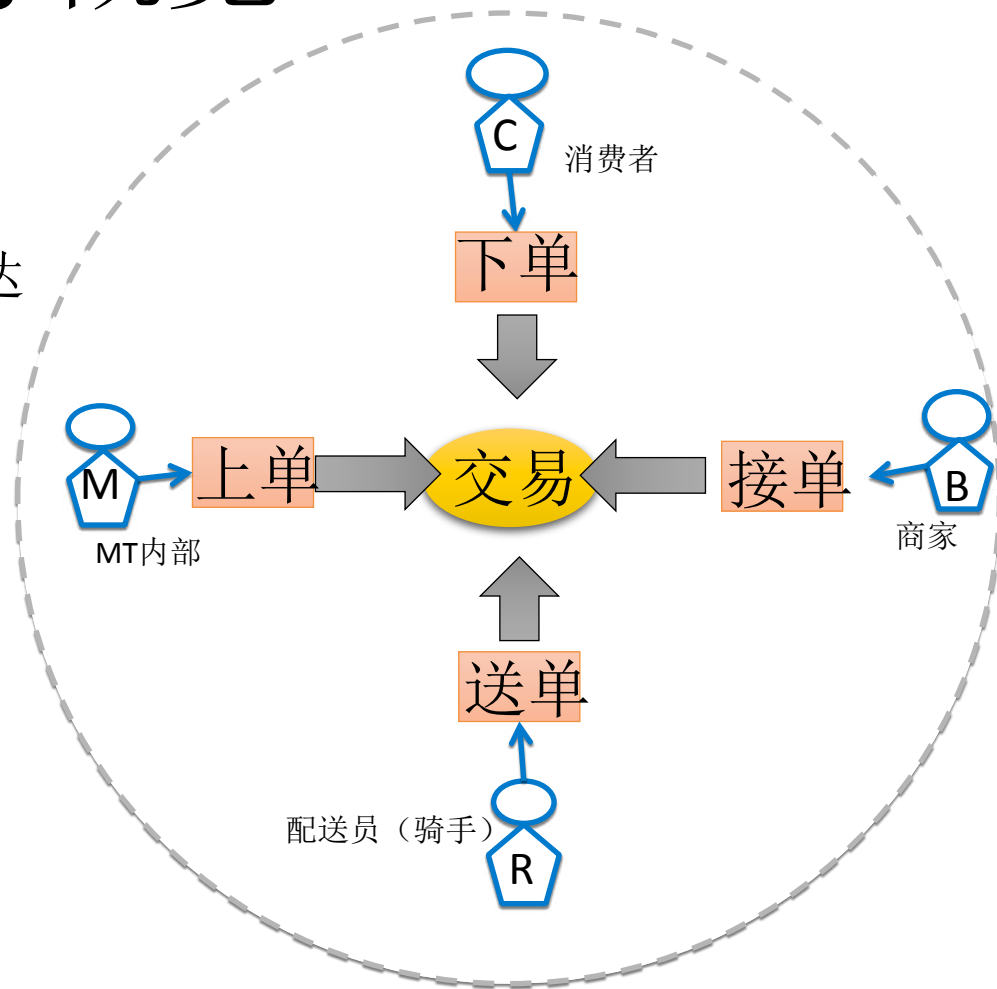
目录

- 新美大外卖业务概览
- 外卖订单系统架构演进
- 外卖订单系统存储、服务、性能及稳定性实践

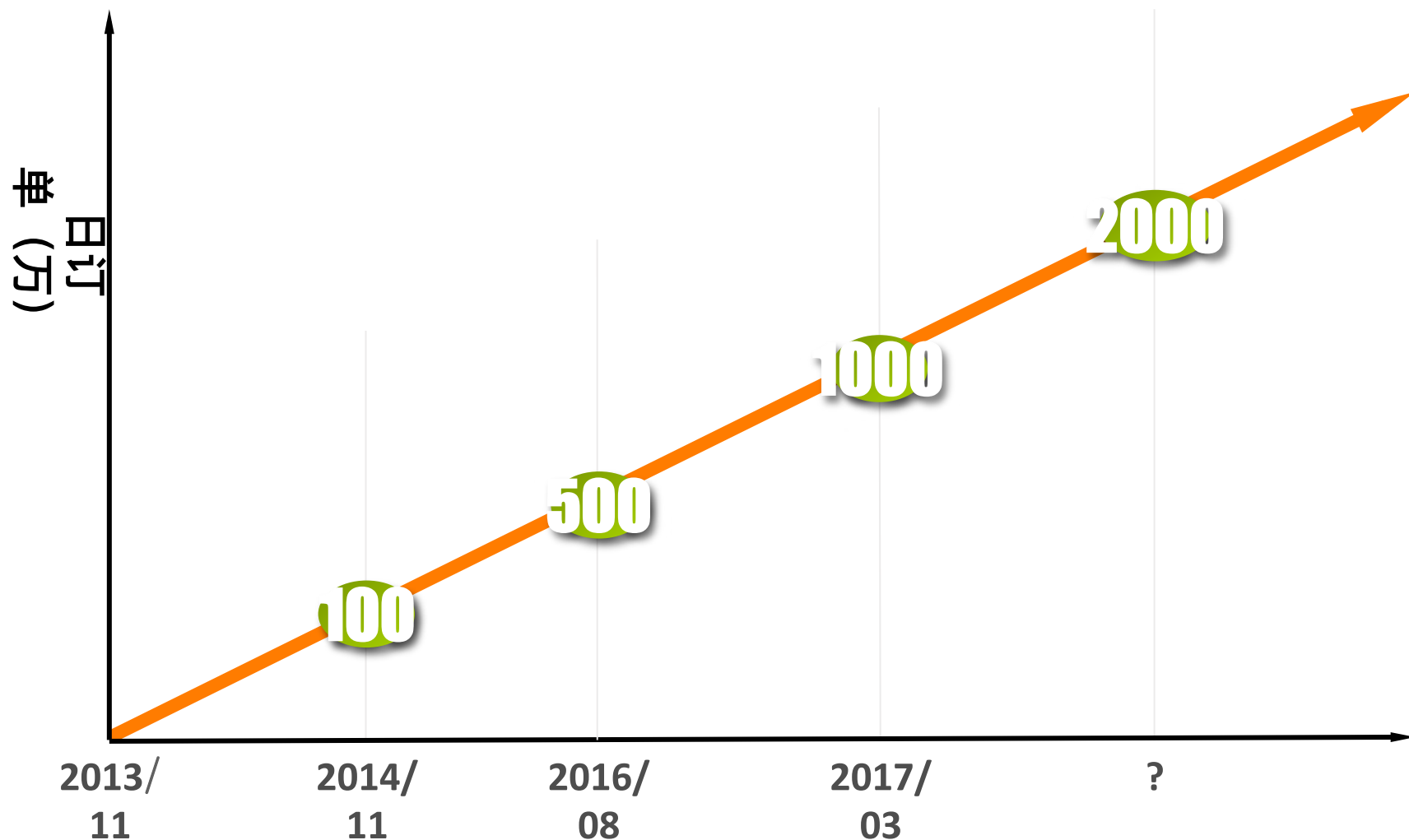
新美大外卖业务概览

新美大外卖业务概览

- 新美大外卖的定位
围绕在线商品交易与及时送达的O2O电商交易平台
- 新美大外卖的口号
美团外卖，送啥都快



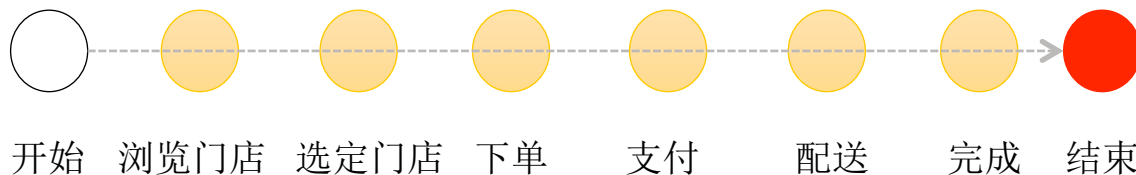
新美大外卖重要里程碑



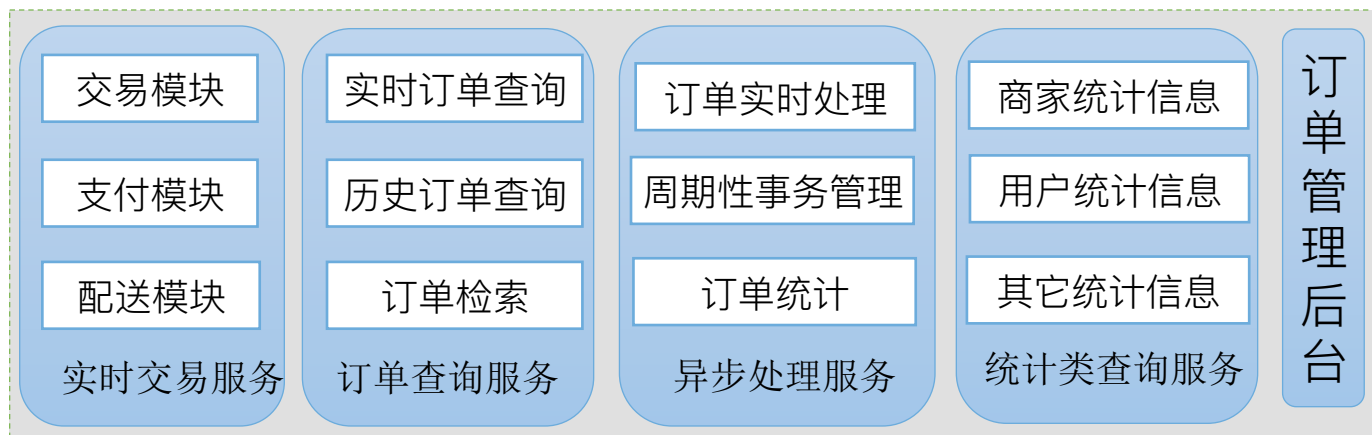
外卖订单系统架构演进

新美大订单系统概览

交易流程



订单中心统一访问层



订单系统

- 围绕外卖交易，提供高效、稳定的全流程服务；
- 分布式微服务；
- 并发高、数据量大；

基础服务

atlas

mtthrift

octo

日志收集

falcon

数据 存储层

DB

DB

DB

tair

redis

es

HIVE

tair

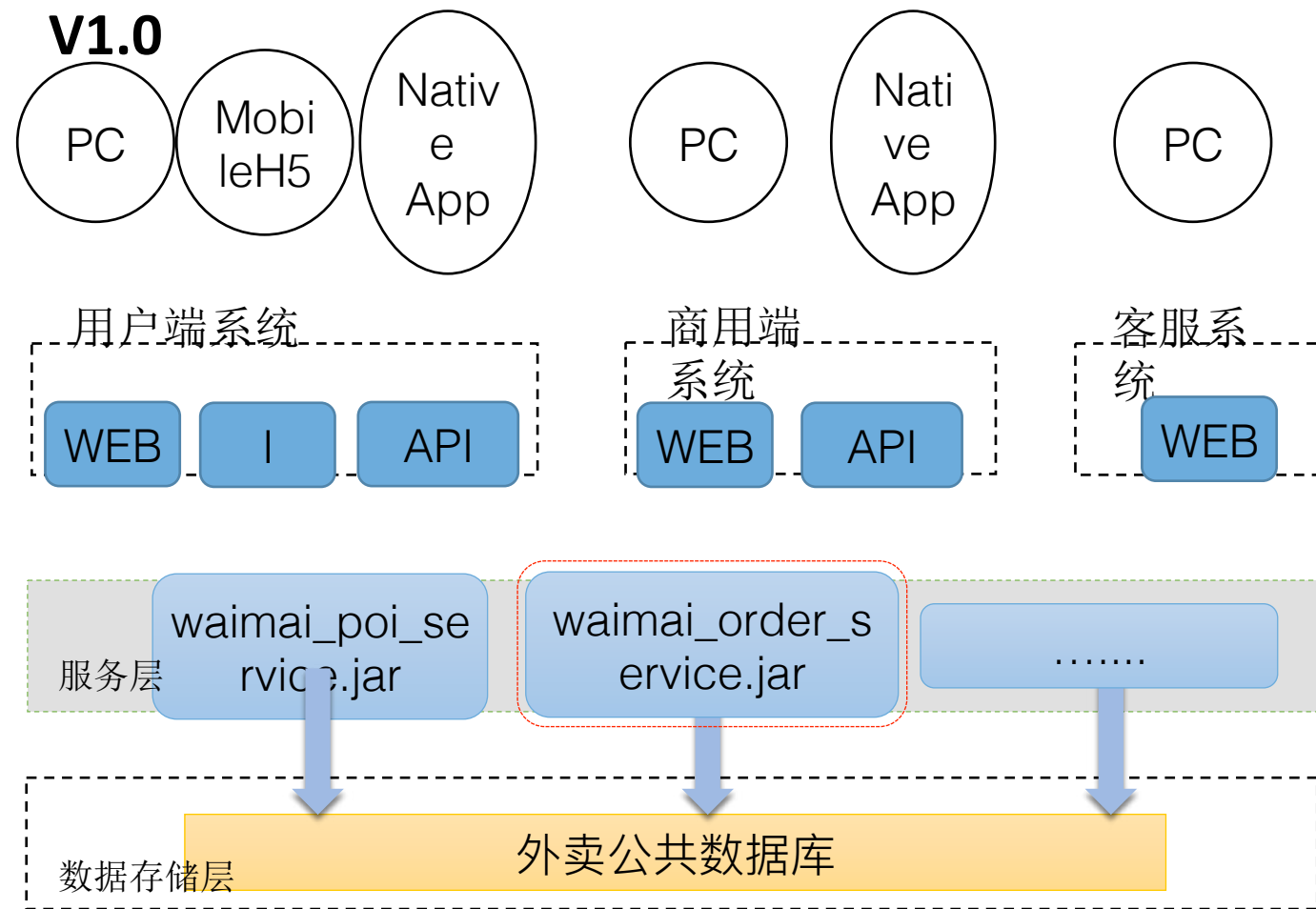
redis

es

新美大订单系统V1.0

订单系统

V1.0



业务背景

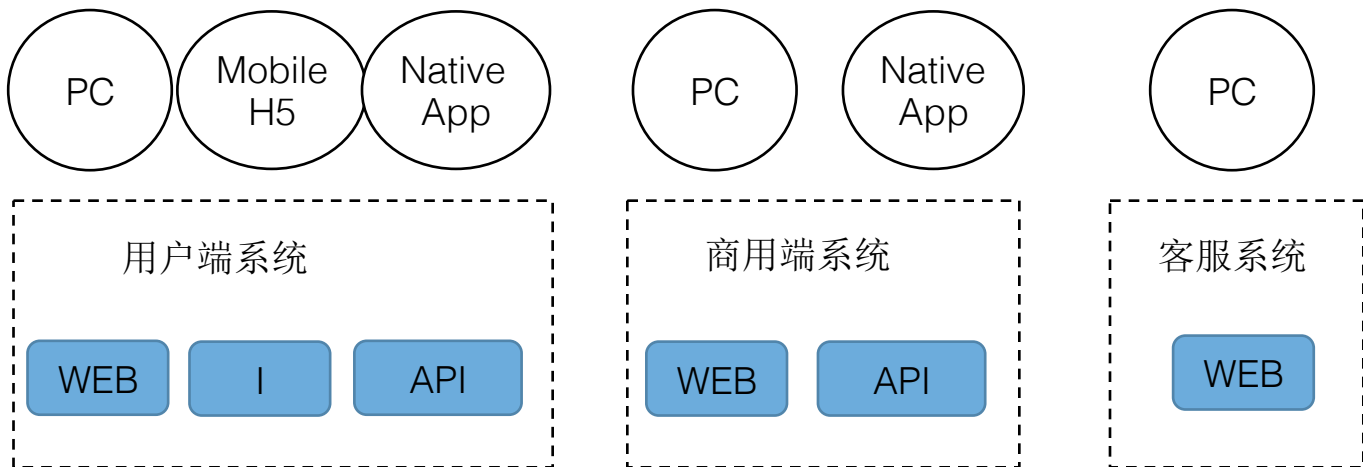
- 业务起步
- 快速迭代
- 快速试错

技术架构

- 架构简单
- 易部署&维护
- 逻辑分散&不易扩展

新美大订单系统V2.0

订单系统 V2.0

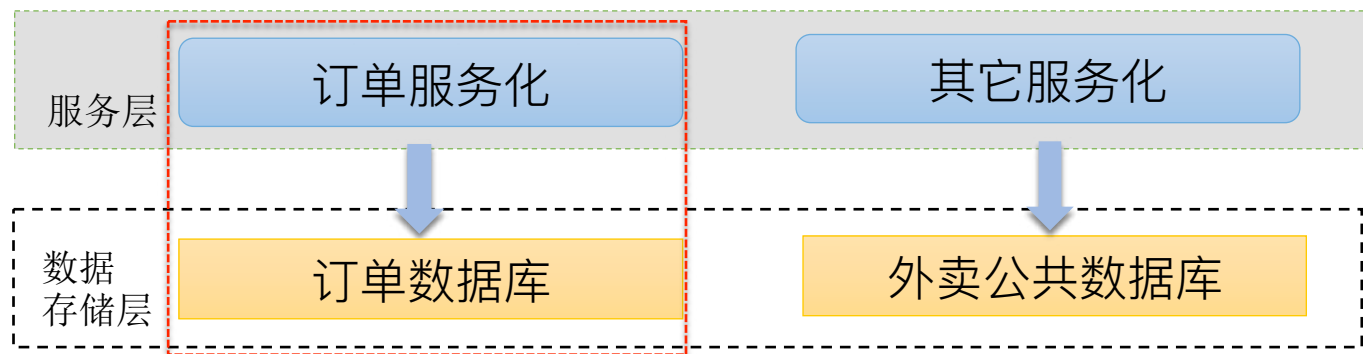


业务背景

- 业务发展快，订单量剧增
- 业务复杂度增长
- 订单逻辑分散

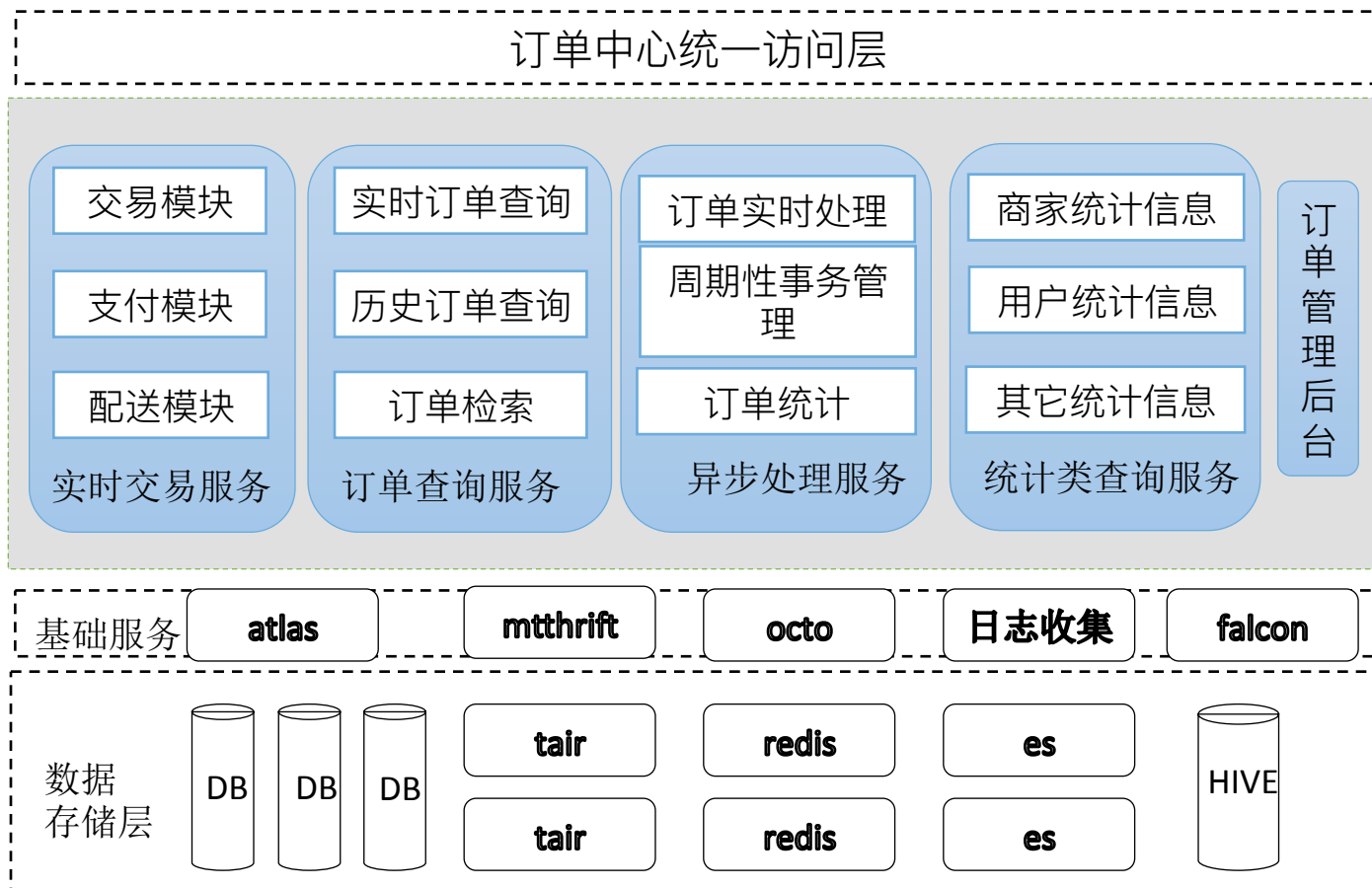
技术架构

- 业务逻辑内聚
- 独立服务
- 独立DB



新美大订单系统V3.0

订单系统V3.0



业务背景

- 业务增长迅猛
- 业务品类增长、订单逻辑日趋复杂
- 保证稳定性

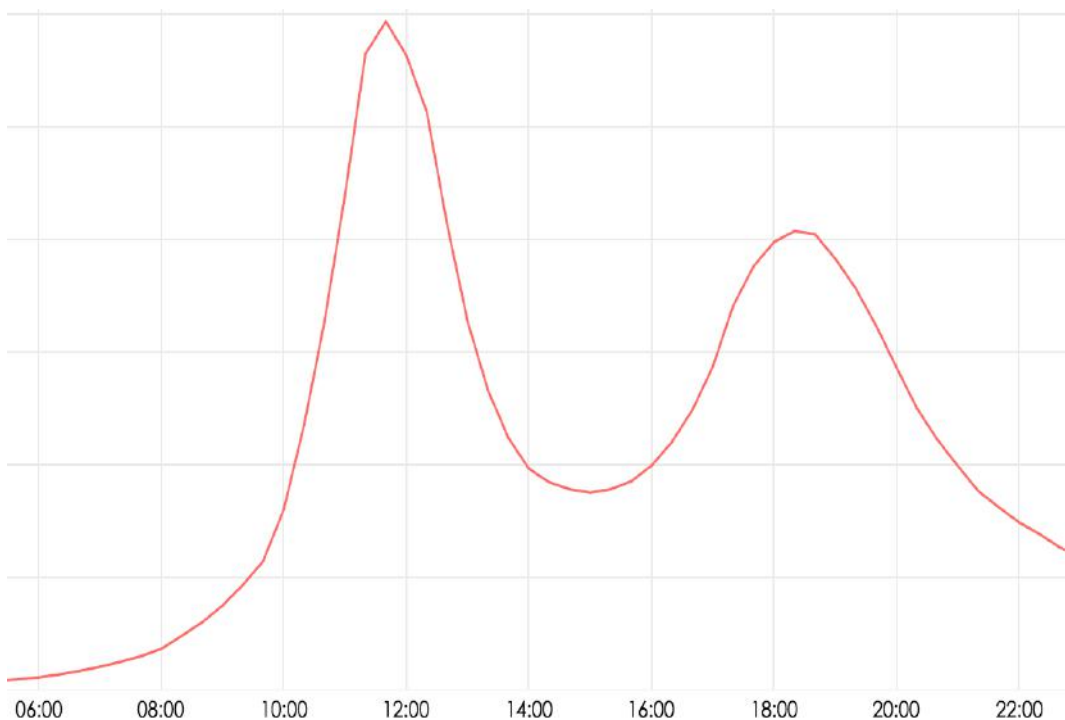
技术架构

- 微服务化
- 综合运用各类存储, 满足存储层的功能、容量及可扩展性要求
- 运用各类中间件保障服务稳定性

外卖订单系统存储、服务、性能 及稳定性实践

新美大订单业务特点及挑战

订单量QPS时段分布图



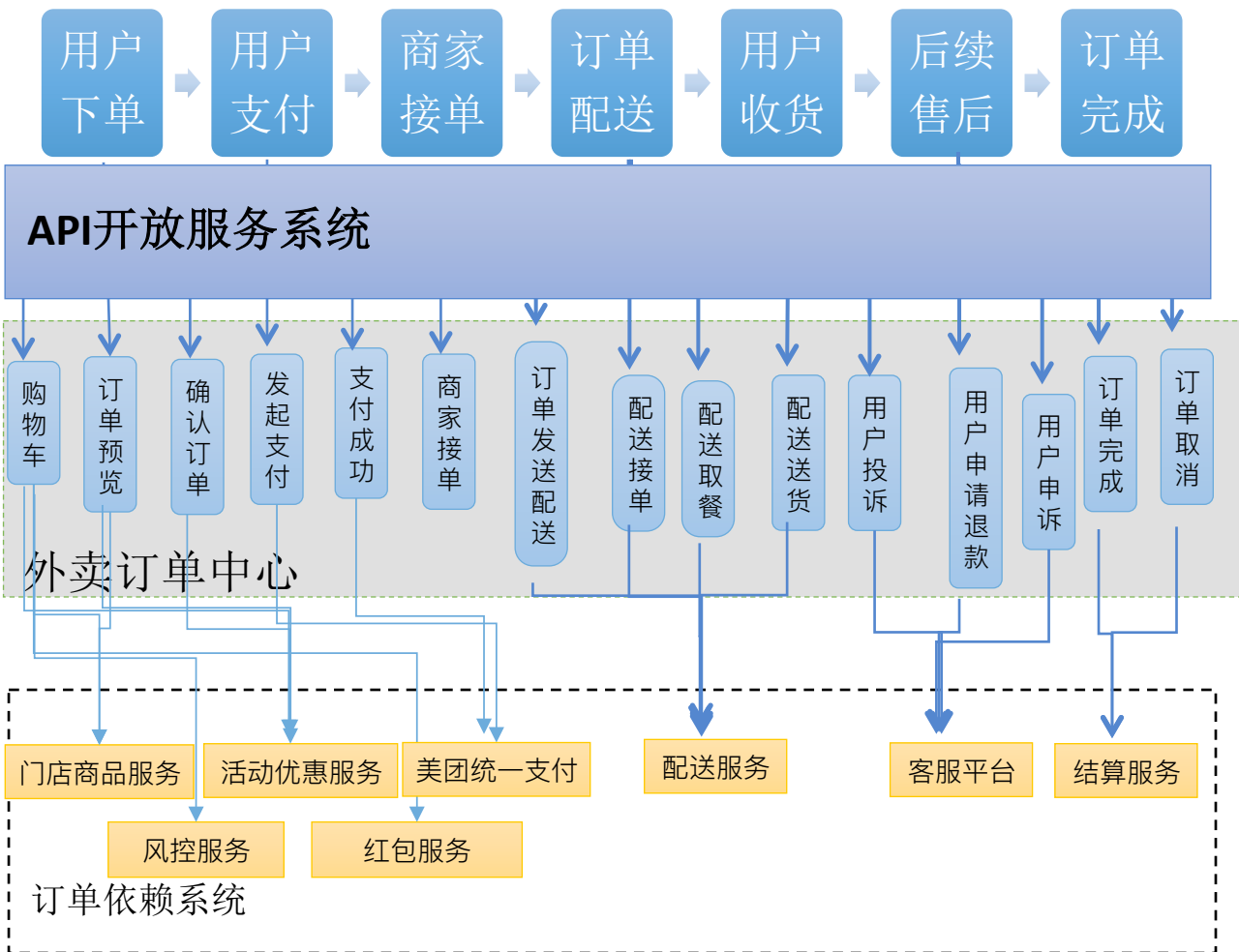
• 业务特点

- 业务集中在中午、晚上两个饭点；
- 业务不可降级、稳定性要求高；

• 系统挑战

- 高峰期并发高、数据量大；
- 一旦故障，损失较大；

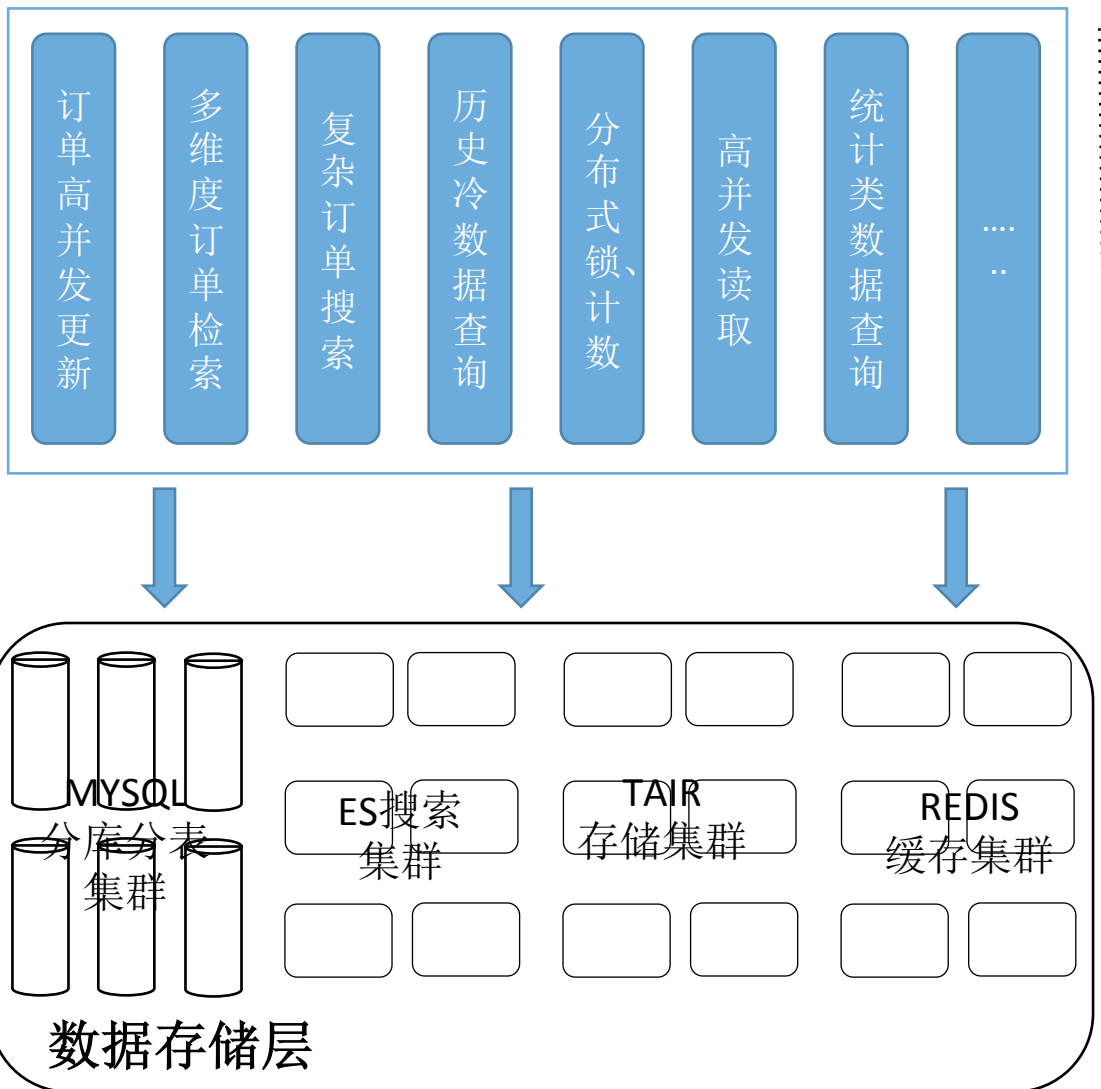
新美大订单交易流程及挑战



• 系统挑战

- 业务流程复杂，服务多；
- 涉及多个系统，一致性要求高；
- 并发高且流程较长，性能要求高；

订单系统数据存储层实践



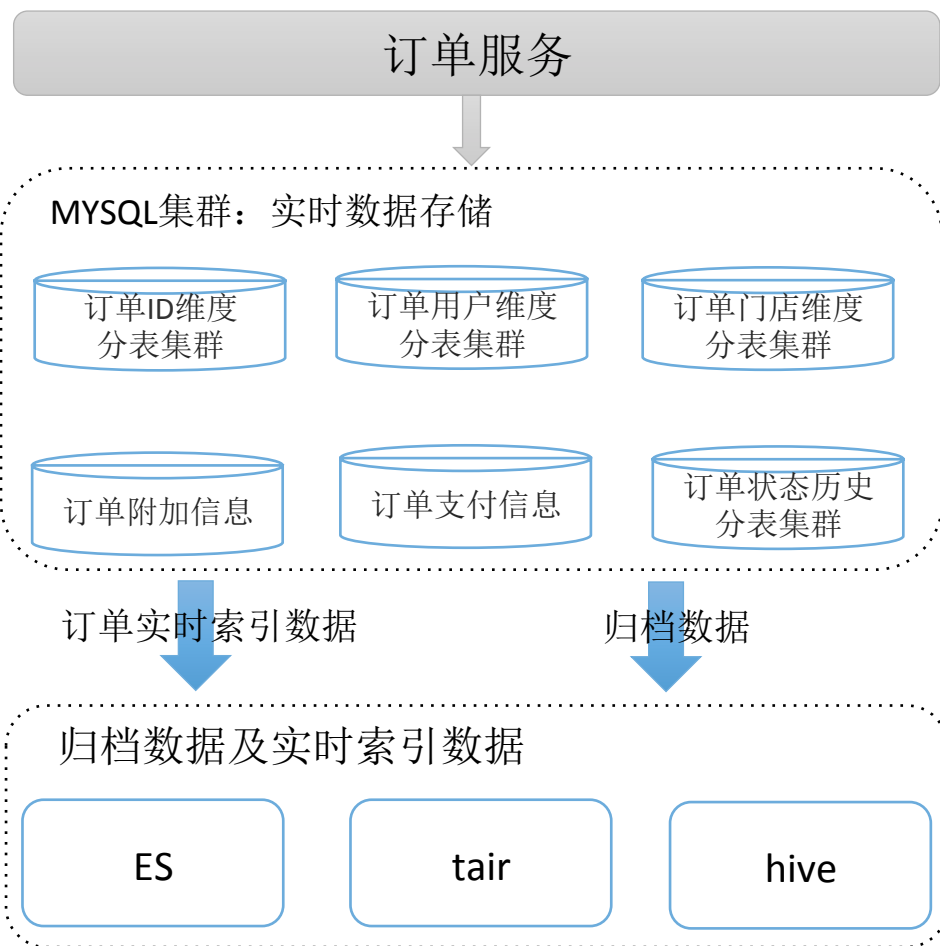
• 问题

- 满足多类查询需求
- 存储海量数据
- 高并发、高性能、高可扩展性

• 解决方案

- 针对不同场景，采用不同存储方案
- 分库/分表
- 冷热数据分离

存储层实践场景 - 订单分库分表+归档



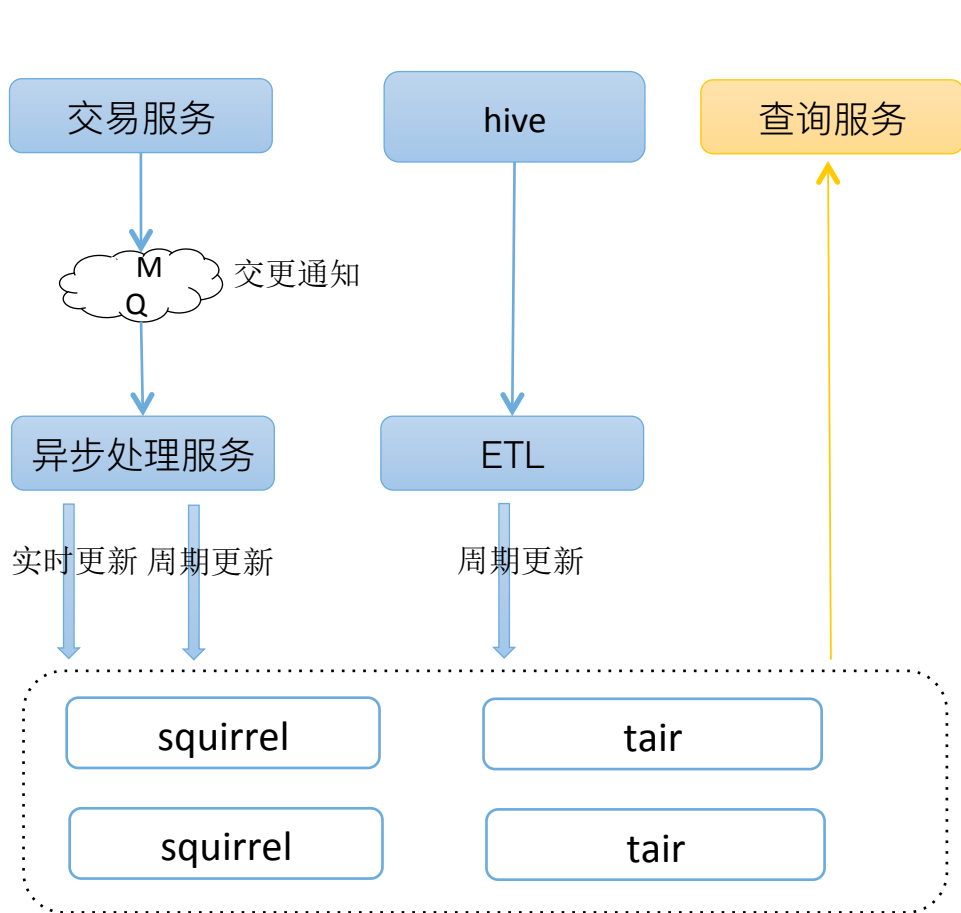
• 问题

- 数据库读、写高
- 数据库单库单表压力大
- 数据访问存在明显的冷热

• 解决方案

- 订单基本库&订单详情库&流程状态库分离
- 按照门店、用户、ID三维度分基本信息库
- 各维度按照ID散100个表、每维度25个表一个库
- 所有表进行归档，归档数据按时间分片

存储层实践场景 – 统计类数据缓存



• 问题

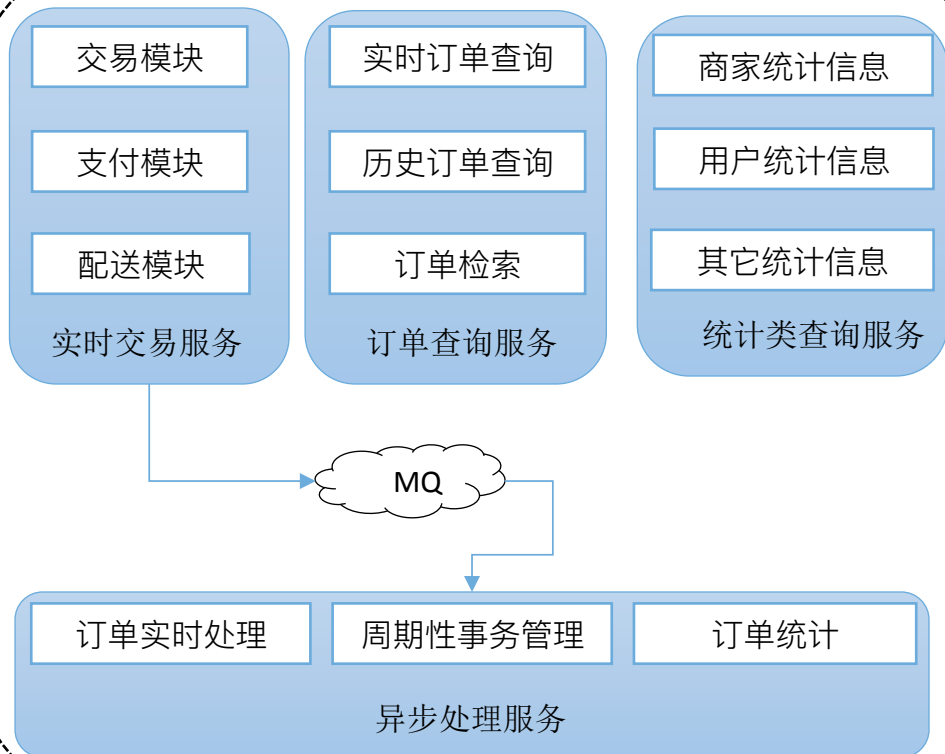
- 查询QPS高
- 更新频繁
- 实时同步

• 解决方案

- 缓存
- 更新后实时同步
- 周期性更新

订单系统服务拆分实践

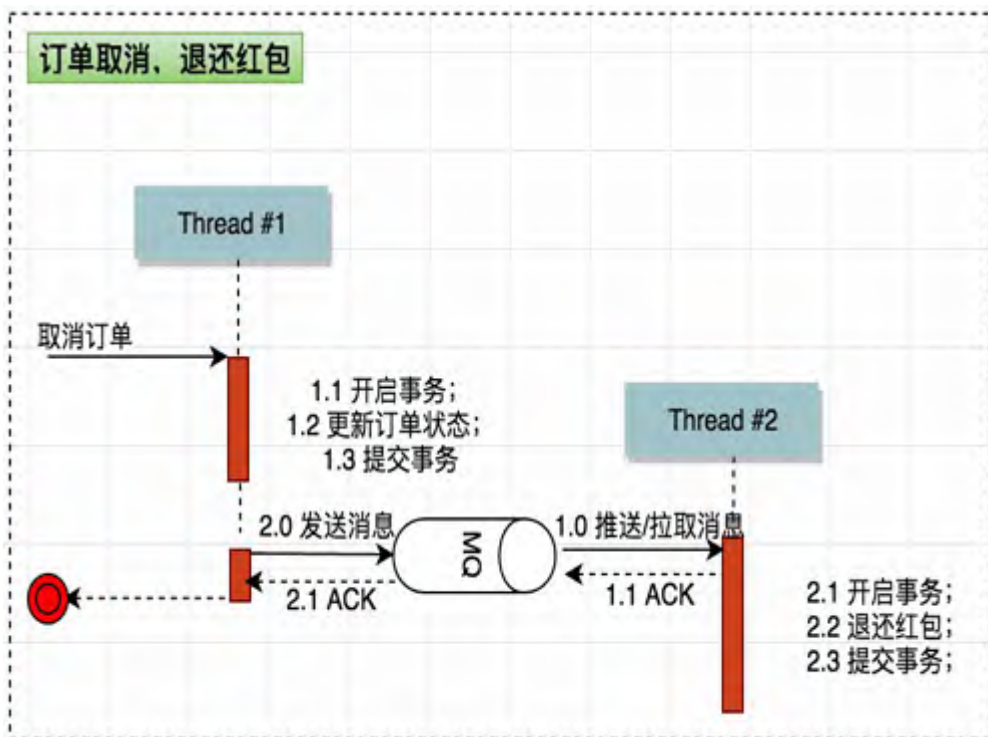
订单中心统一访问层



• 拆分实践

- 按业务划分为职责单一、清晰的模块
- 系统间避免相互依赖，通过MQ来解耦
- 核心路径服务与非核心路径服务隔离

订单系统分布式事务一致性



• 问题

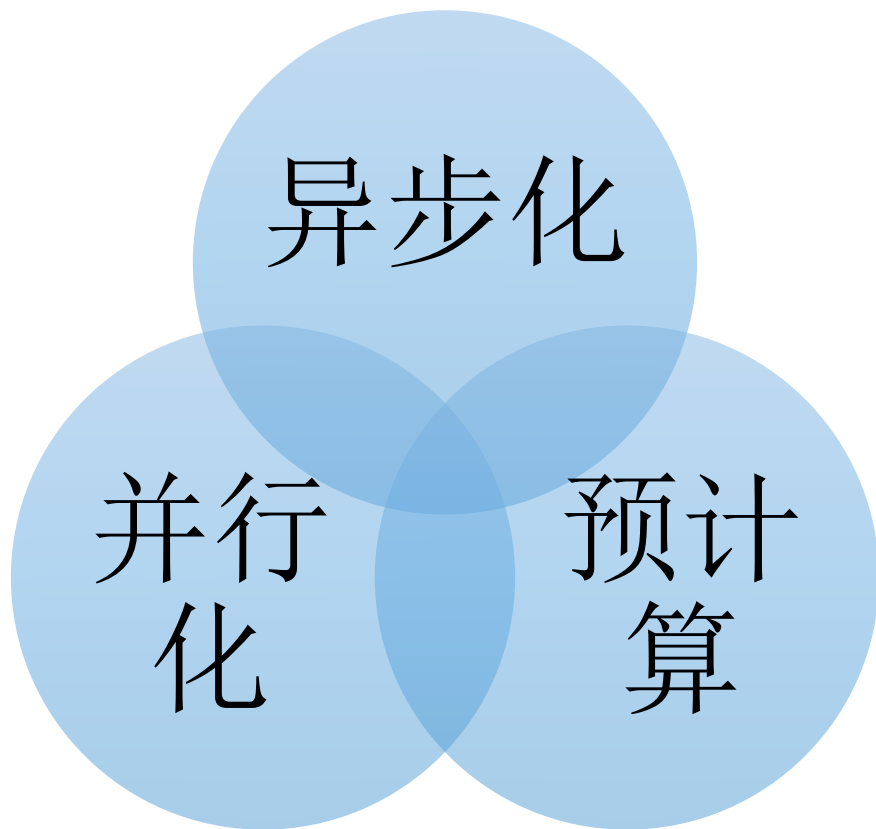
- 跨系统业务流程处理
- 分库分表事务一致性

• 解决方案

- 回滚接口，失败时回滚
- 主从同步，MQ异步通知，保证最终一致
- 兜底JOB，保证最终一致

	Backups	M/S	MM	2PC	Paxos
Consistency	Weak	Eventual		Strong	
Transactions	No	Full	Local	Full	
Latency	Low			High	
Throughput	High			Low	Medium
Data loss	Lots	Some		None	
Failover	Down	Read only	Read/write		

订单系统性能实践



• 问题

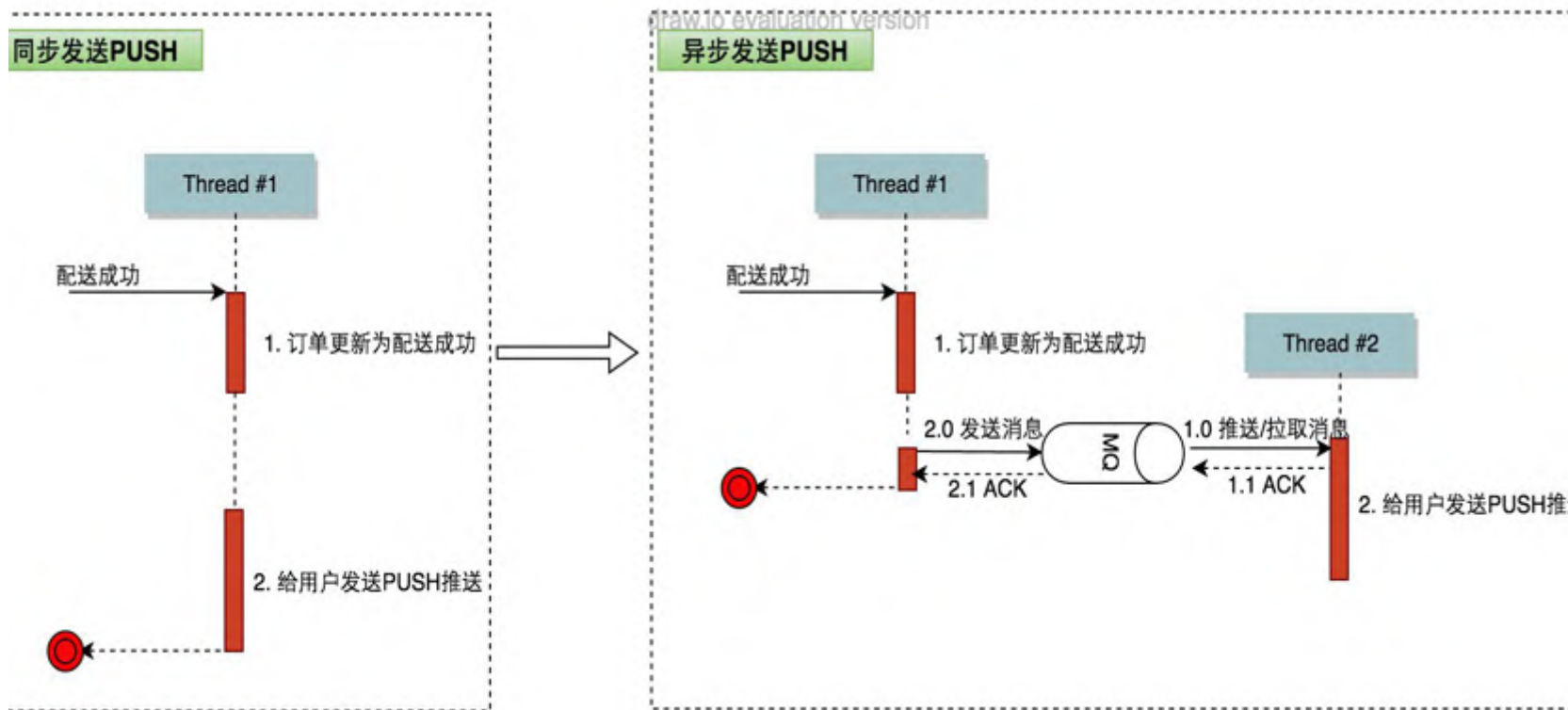
- 非必要操作影响关键路径
- 整体处理流程长
- 数据实时计算时间长

• 解决方案

- 非必要操作异步化
- 同步操作并行化
- 预计算

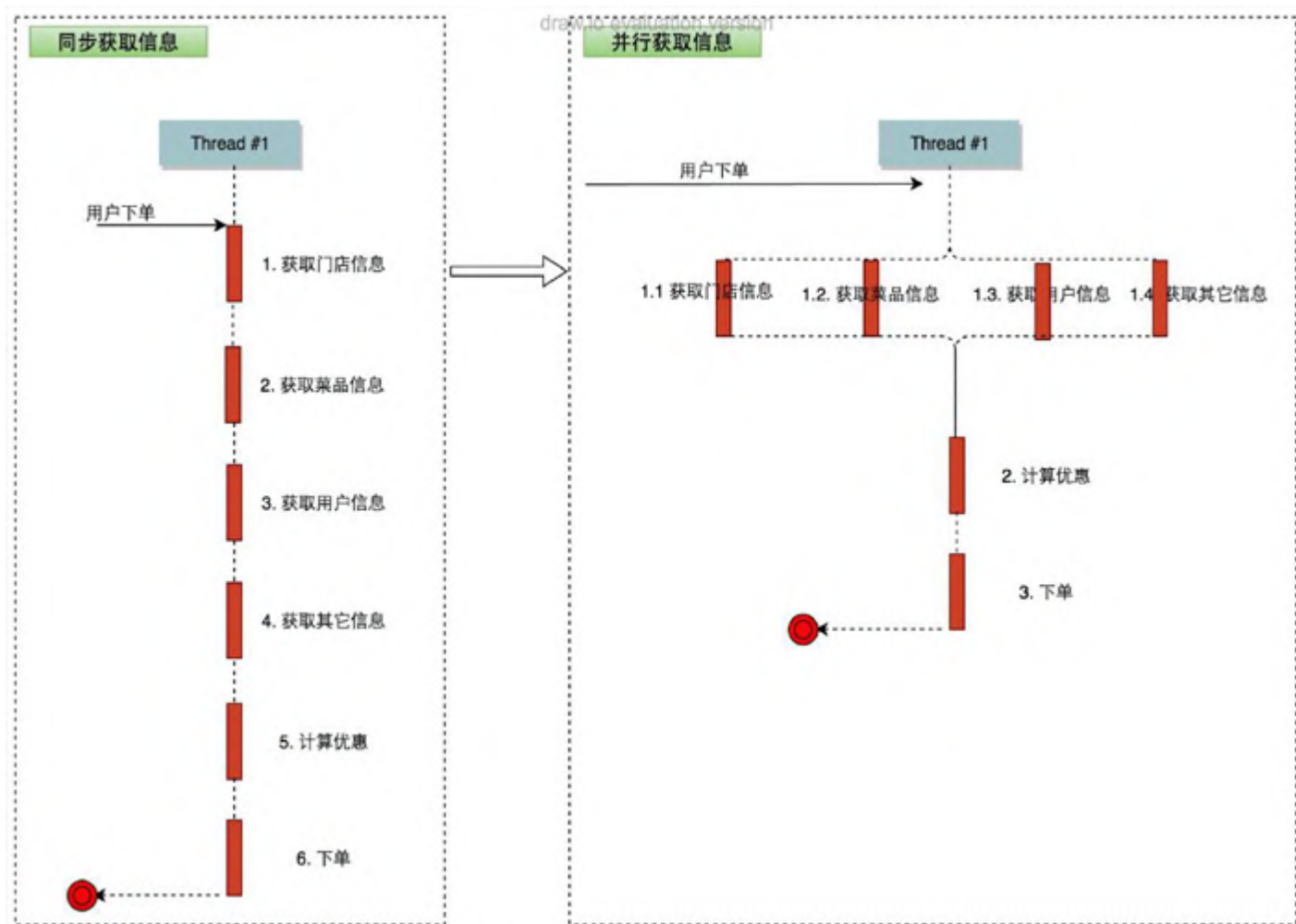
性能实践场景 - 异步化

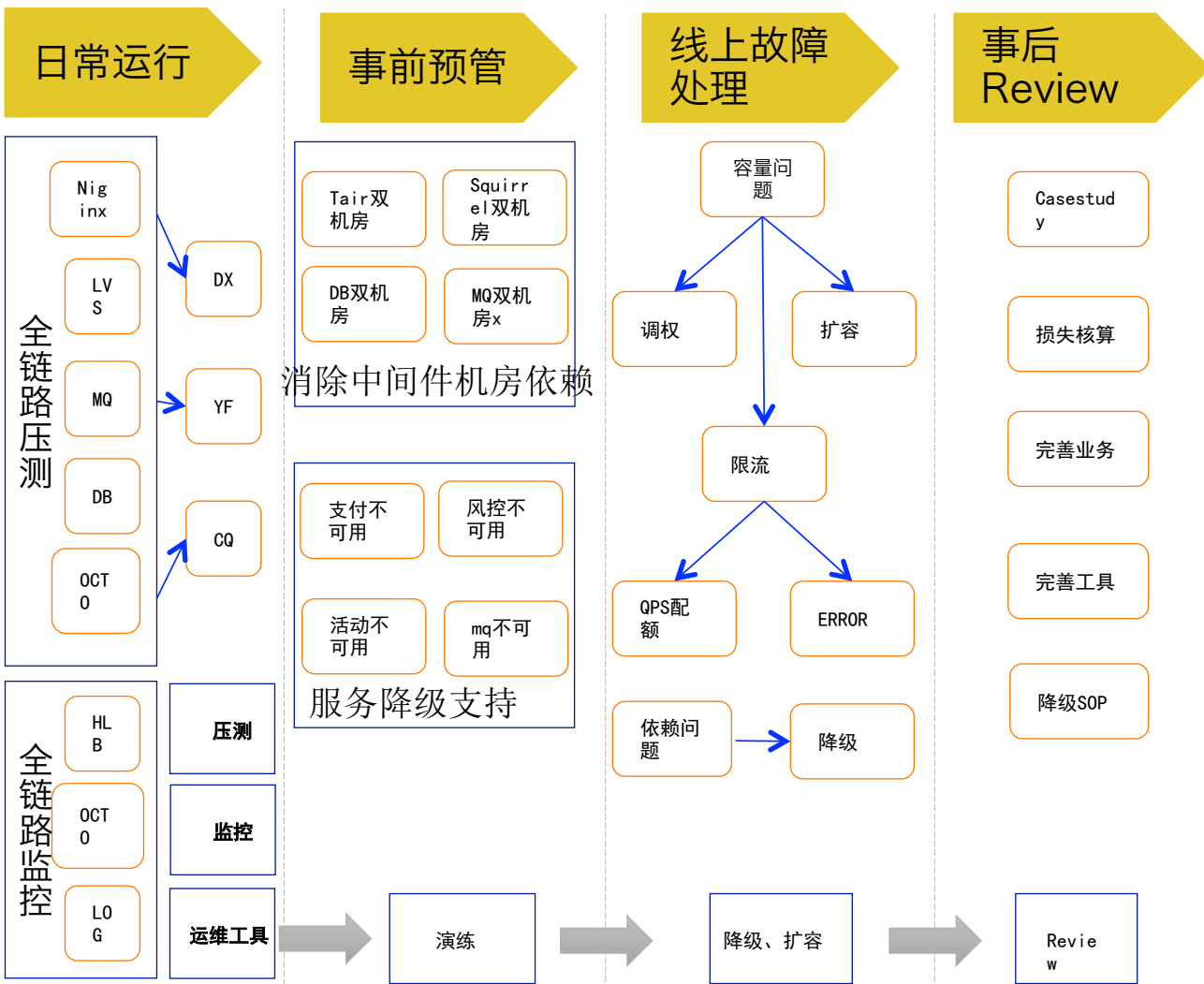
异步化：push、发红包等非必要操作通过MQ异步解耦。



性能实践场景 – 并行化

并行化：同步获取信息操作并行化，降低整体时间





核心要点

- 具备容灾能力的架构;
- 系统定期压测, 保障系统容量;
- 核心路径的降级策略及降级SOP;
- 降级方案定期演练;
- 服务支持水平扩展;
- 服务支持流量控制;
- 完善的事故总结机制;

Thank you !