CODE A BETTER LIFE
一 行 代 码 亿 万 生 活

美团旅行酒店住宿
订单系统架构与高可用实战

李磊

美团点评

美团旅行

李磊
美团旅行酒店住宿订单组

2014~2016年 美团到店餐饮交易组；
2017~2018年 美团旅行酒店住宿订单组；

主要内容

- 系统可用性评估
- 美团旅行酒店订单系统介绍
- 美团旅行酒店订单系统高可用实战

主要内容

- 系统可用性评估
- 美团旅行酒店订单系统介绍
- 美团旅行酒店订单系统高可用实战

系统可用性评估

可用性指什么

- 理性认识

$$availability = MTBF \div (MTBF + MTTR) \times 100\%$$

- MTBF : mean time between failure
- MTTR : mean time to recovery

系统可用性评估

可用性指什么

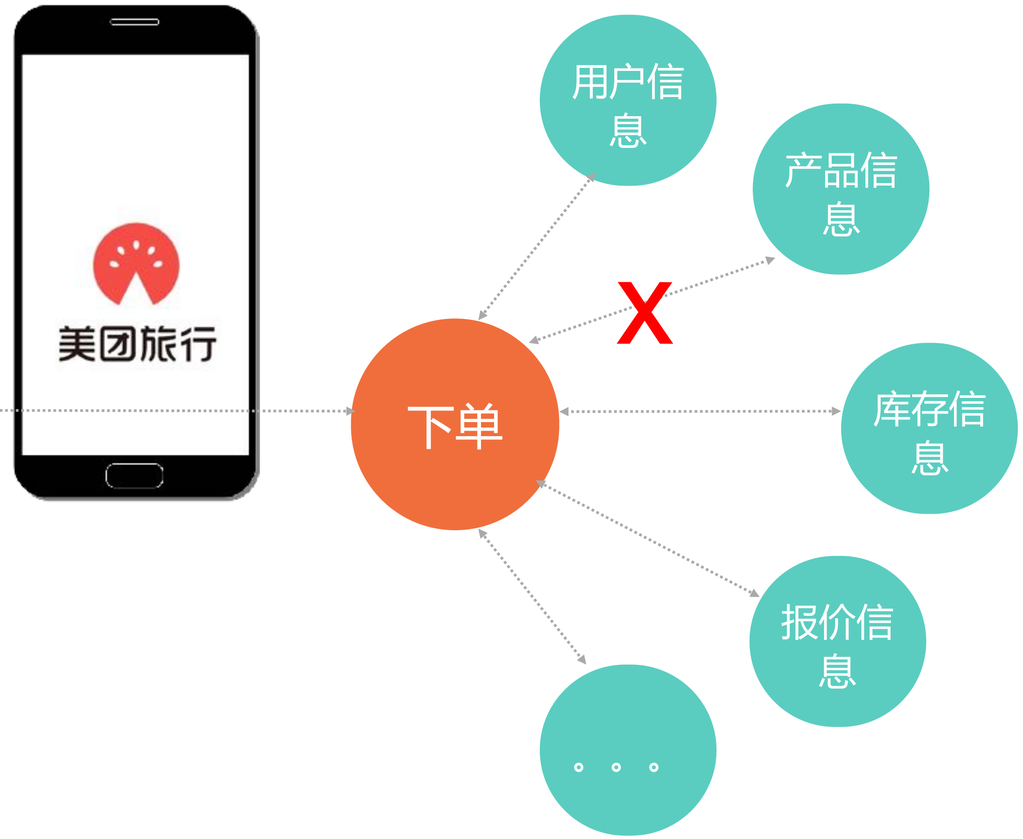
- 感性认识

8h	52min	5min	31s
99.9%	99.99%	99.999%	99.9999%
Normal	Good	Excellent	Terrific

系统可用性评估

可用性指什么

- 就这么简单？？？
- 正确指定业务系统可用性指标
- 服务可用 ≠ 业务可用



主要内容

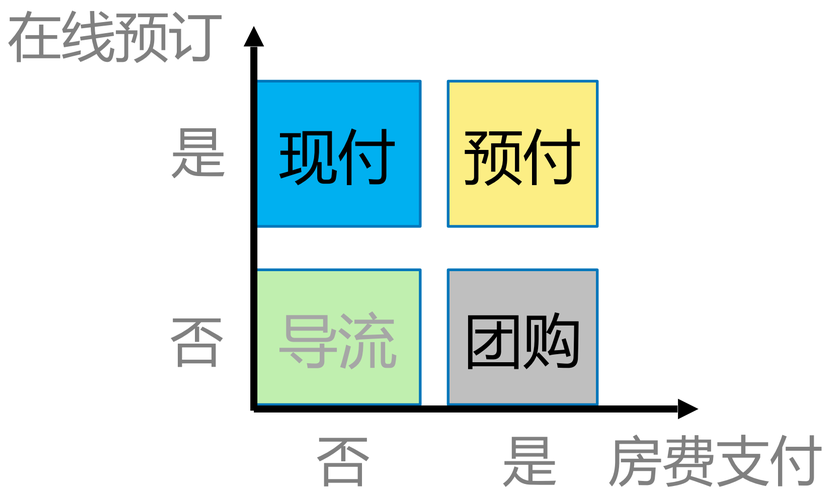
- 系统可用性评估
- 美团旅行酒店订单系统介绍
- 美团旅行酒店订单系统高可用实战

背景知识

酒店订单系统介绍

- 供给
  - 地区：国内、海外
  - 接入：直连、自上单
  - 类别：高星、非高星

- 售卖
  - 售卖模式



- 售卖渠道：自销、分销

业务发展

酒店订单系统介绍

2014

刚刚起步，以团购的交易模式为主；

2015

大搞预订，逐渐以预订的售卖方式替代团购；

2016

业务迅速发展，订单量屡创新高；

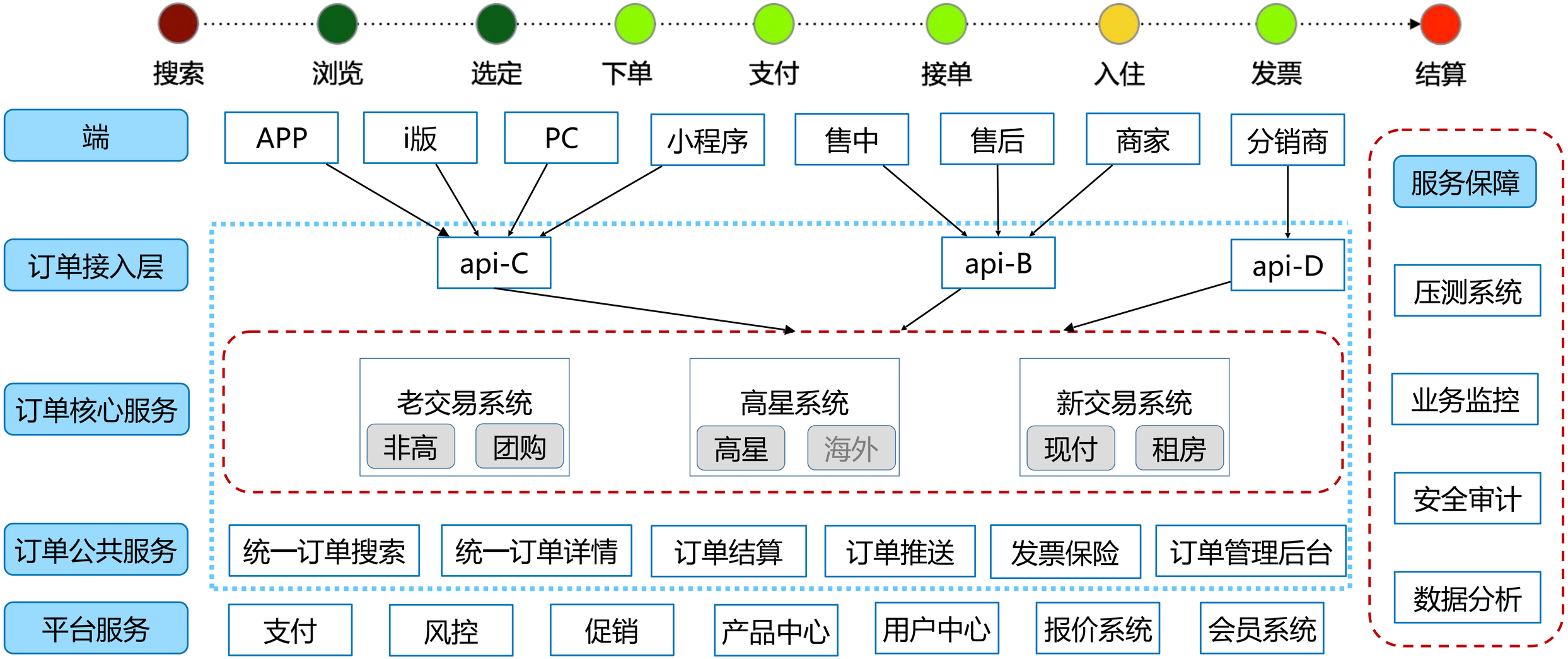
2017

高星酒店、海外酒店、现付交易、长租公寓，齐头并进；

150w / day

系统现状

酒店订单系统介绍



主要内容

- 系统可用性评估
- 美团旅行酒店订单系统介绍
- 美团旅行酒店订单系统高可用实战

高可用实战

酒店订单系统高可用

- remind

$$availability = MTBF \div (MTBF + MTTR) \times 100\%$$

- MTBF : mean time between failure

- MTTR : mean time to recovery

高可用实战

提高系统故障间隔

A 理依赖

依赖分级

熔断保护

B 做隔离

快慢分离

读写分离

冷热分离

C 定规矩

上线SOP

D 估容量

服务容量

存储容量

E 做自检

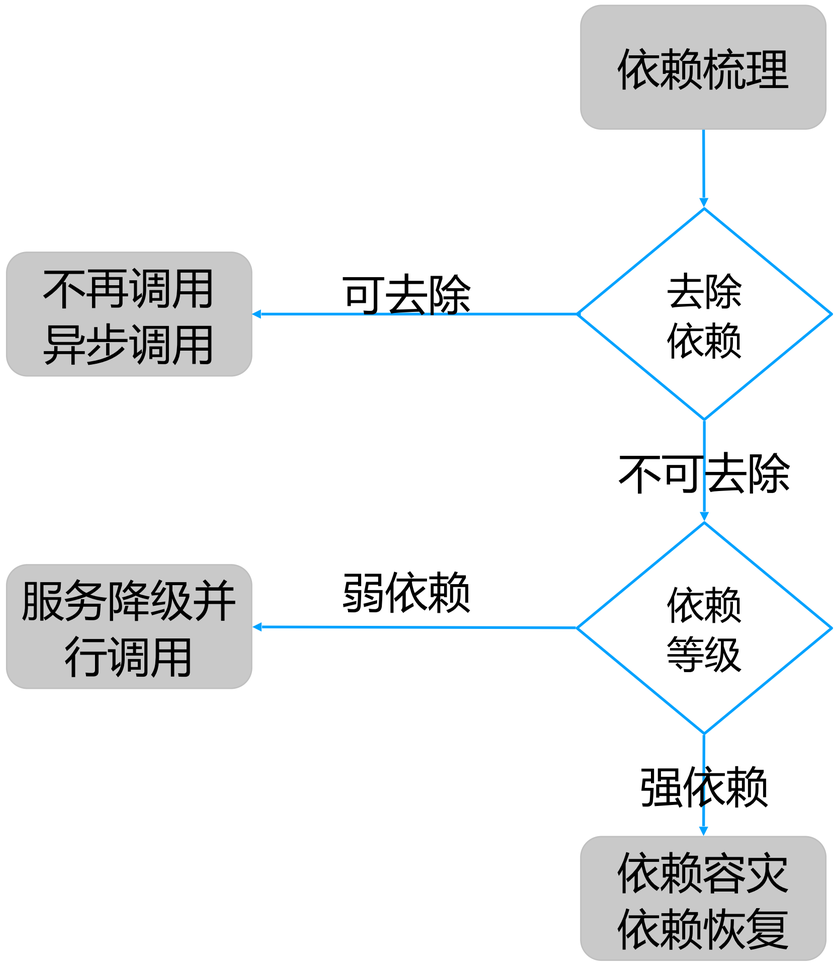
压测

故障演练

高可用实战

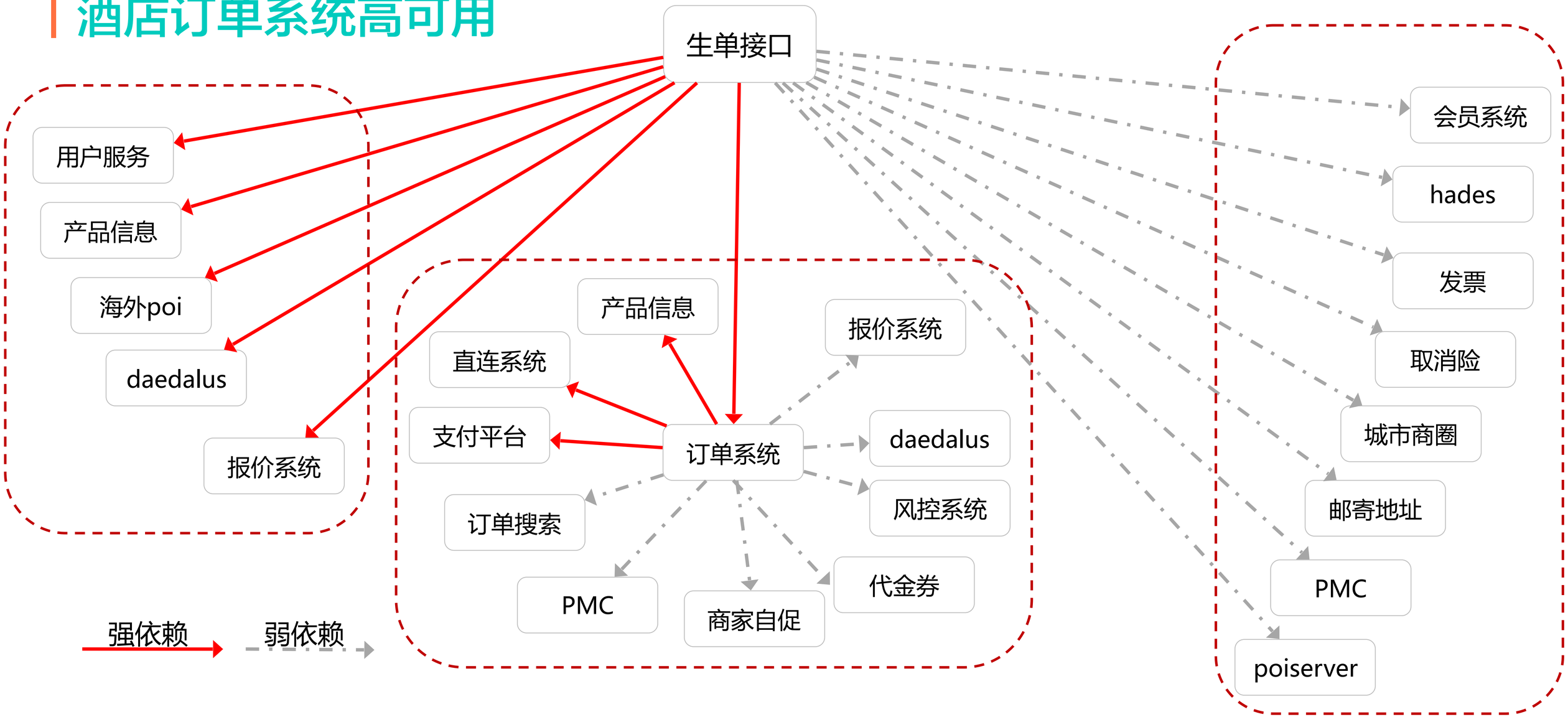
酒店订单系统高可用

- 依赖
  - 依赖管理
  - 能去则去
  - 能弱则弱



高可用实战

酒店订单系统高可用



高可用实战

酒店订单系统高可用

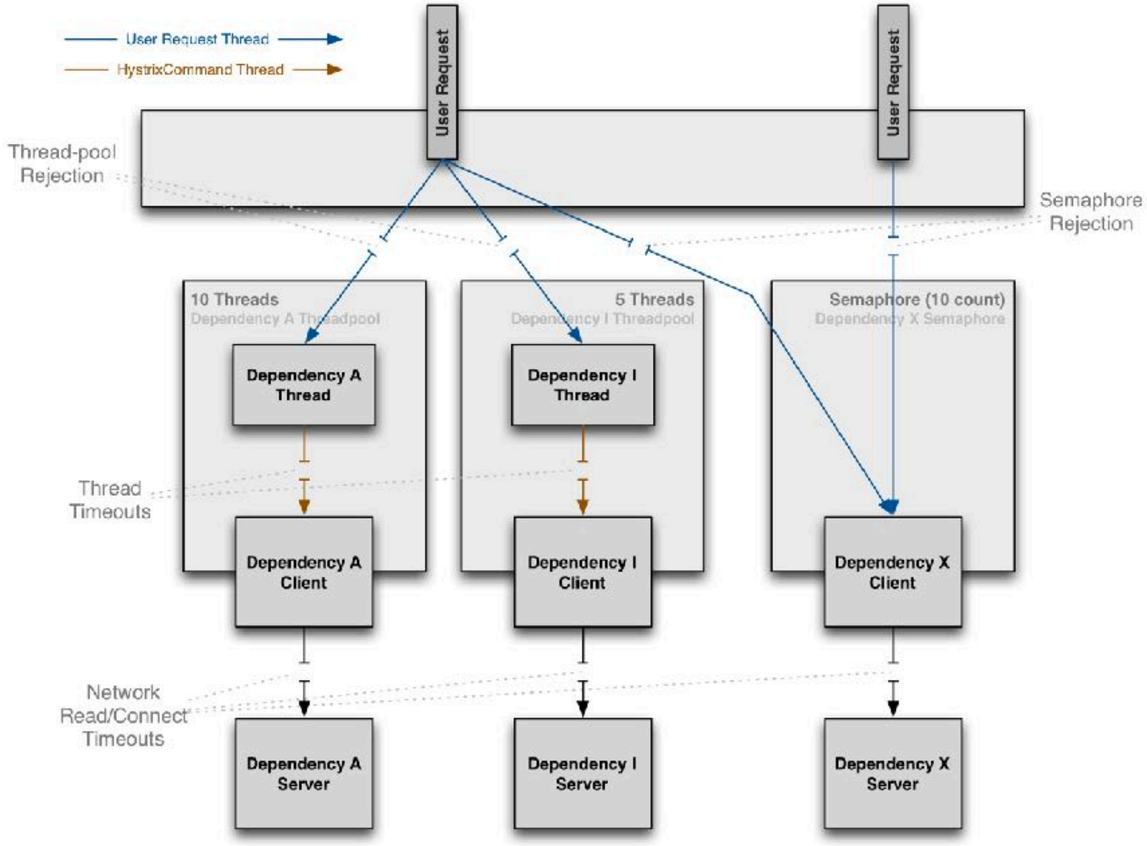
- 分布式系统中经常会出现某个服务不可以的情况，容易造成整体系统 **雪崩**，影响系统稳定性；应对这种情况，hystrix提供的 **熔断机制** 可以很好的保护我们的系统；

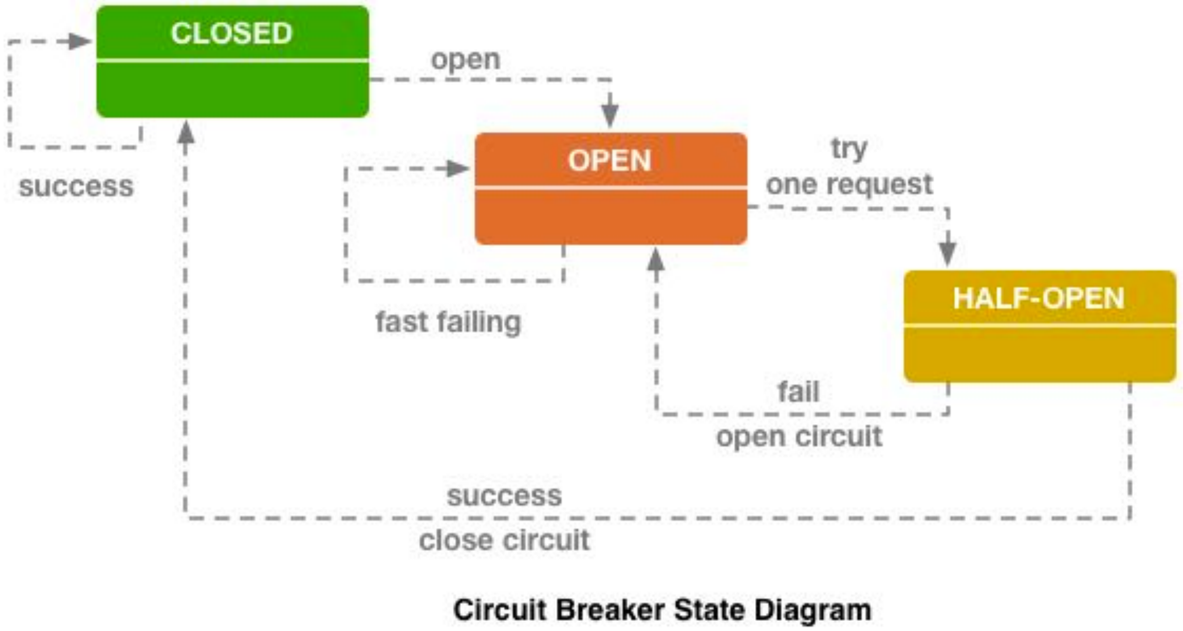
HYSTRIX
DEFEND YOUR APP

高可用实战

酒店订单系统高可用

- Hystrix原理





高可用实战

酒店订单系统高可用

- 超时
  - 超时管理：500ms 红线

仓库	项目	超时配置值	负责人	review意见	其它说明wiki等
hotel-order-search	订单搜索	ievent.properties thriftTimeout=4000	fengxuan03	OK	此配置为ievent设置的超时时间，等同于job的超时设置因此值设置的大小不会影响主流程。
	IEsproxy4SearchService	timeout=2000	fengxuan03	TODO	调用esproxy设置超时为2000ms，依据：1、年前测试结果显示99.9线在2000ms；2、调用方apib和预订中心的超时设置分别为1s和2s。
hotel-order-platform-apic	IOrder4ThirdPartService	timeout=5000	heqianchi	OK	包括创建点评订单：CommandCreateDianPingOrder hystrix的超时是5000 微信或第三方订单：CommandCreateThirdPartyOrder的hystrix的超时是1000
	IOrder4ConsumerServiceOnly4ZL	timeout=10000	heqianchi	OK	创建直连订单：CommandCreateZhiLianOrder
	IOrder4ConsumerService	timeout=1200	heqianchi	TODO	CommandCreateYuFuOrder99.9线是1,023.8ms
	OHGoodsServiceForAPI	timeout=8000	heqianchi	TODO	CommandGetOverseaGoods99.9线是5200ms
	IOrderService	timeout=1000	heqianchi	TODO 后面2个getOrderDetail，再次确认一下	IOrderService.cancelOrder的99.9线是963.8msIOrderService.hideOrder的99.9线是322.8msIOrderService.getOrderFreeTipModel的99.9线是51.2msIOrderService.getOrderDetail的99.9线是87.6ms，IOrderService.isNeedRegistered的99.9线是65.5ms,所以OCTO服务统一设置为1000ms

高可用实战

提高系统故障间隔

A 理依赖

依赖分级

熔断保护

B 做隔离

快慢分离

读写分离

冷热分离

C 定规矩

上线SOP

D 估容量

服务容量

存储容量

E 做自检

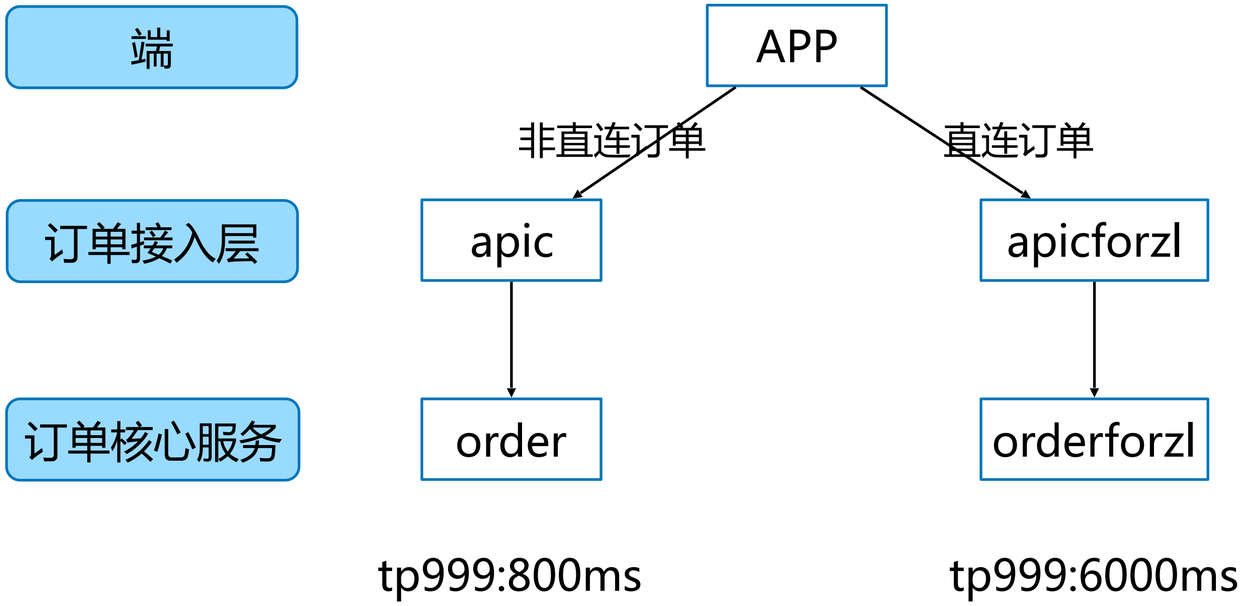
压测

故障演练

高可用实战

酒店订单系统高可用

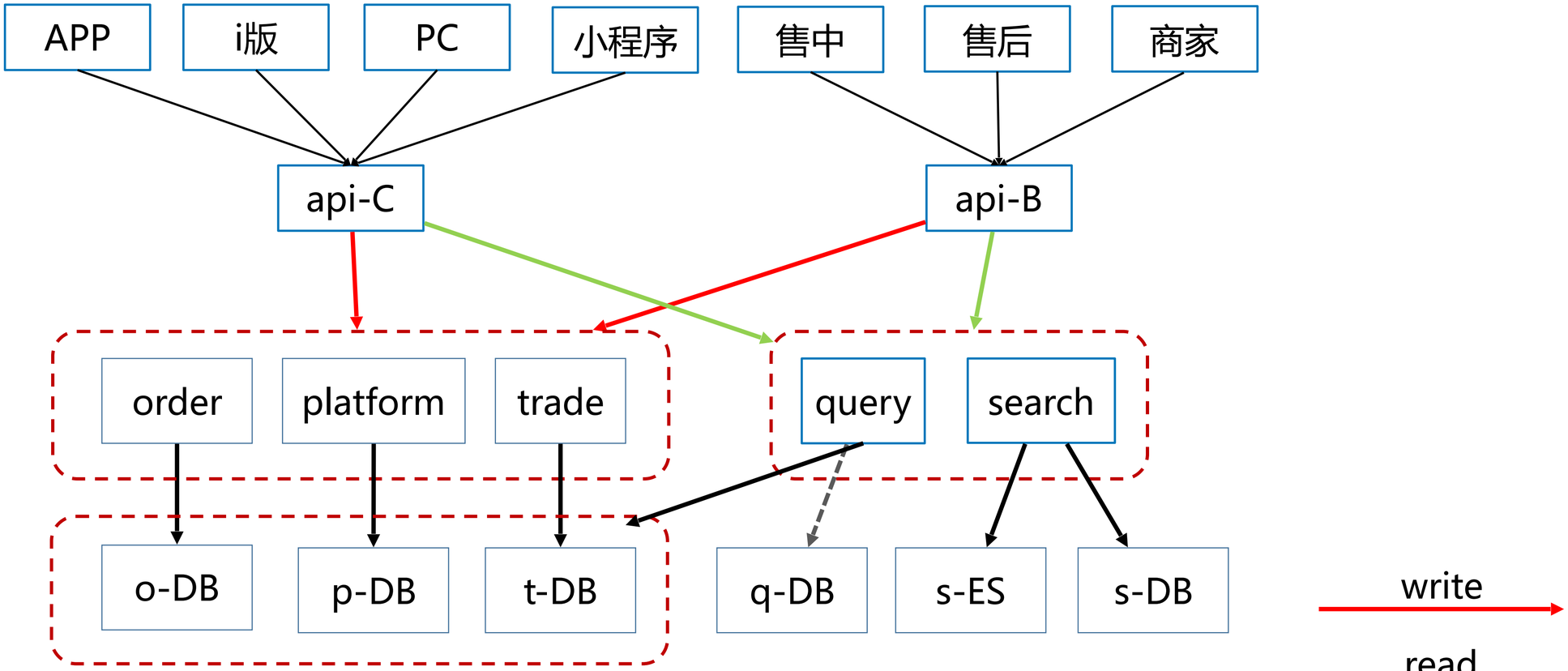
- 隔离
  - 快慢隔离
  - 超时设置更合理
  - 告警更准确
  - 服务异常不会相互影响



高可用实战

酒店订单系统高可用

- 隔离
  - 读写隔离

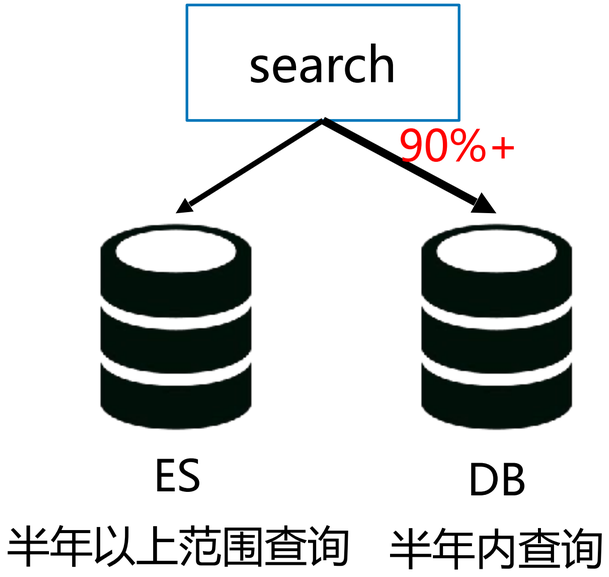


- search：根据条件查询订单id
- query：根据id查询订单详情

高可用实战

酒店订单系统高可用

- 隔离
  - 冷热隔离
  - 分析热点数据
  - 推动调用方修改
  - 删除历史数据



高可用实战

提高系统故障间隔

- A 理依赖
  - 依赖分级
  - 熔断保护
- B 做隔离
  - 快慢分离
  - 读写分离
  - 冷热分离
- C 定规矩
  - 上线SOP
- D 估容量
  - 服务容量
  - 存储容量
- E 做自检
  - 压测
  - 故障演练

高可用实战

酒店订单系统高可用

- 规矩
  - 上线SOP
  - 大型项目测试报告发出后才可上线
  - 2人以上review且approve
  - release分支发布
  - 必须 灰度发布 且观察监控与日志
  - 上线前准备好checklist和 回滚 方案
  - 上线前做好相关方周知
  - 限制 发布次数，每周二、周四上线

高可用实战

提高系统故障间隔

A 理依赖

依赖分级

熔断保护

B 做隔离

快慢分离

读写分离

冷热分离

C 定规矩

上线SOP

D 估容量

服务容量

存储容量

E 做自检

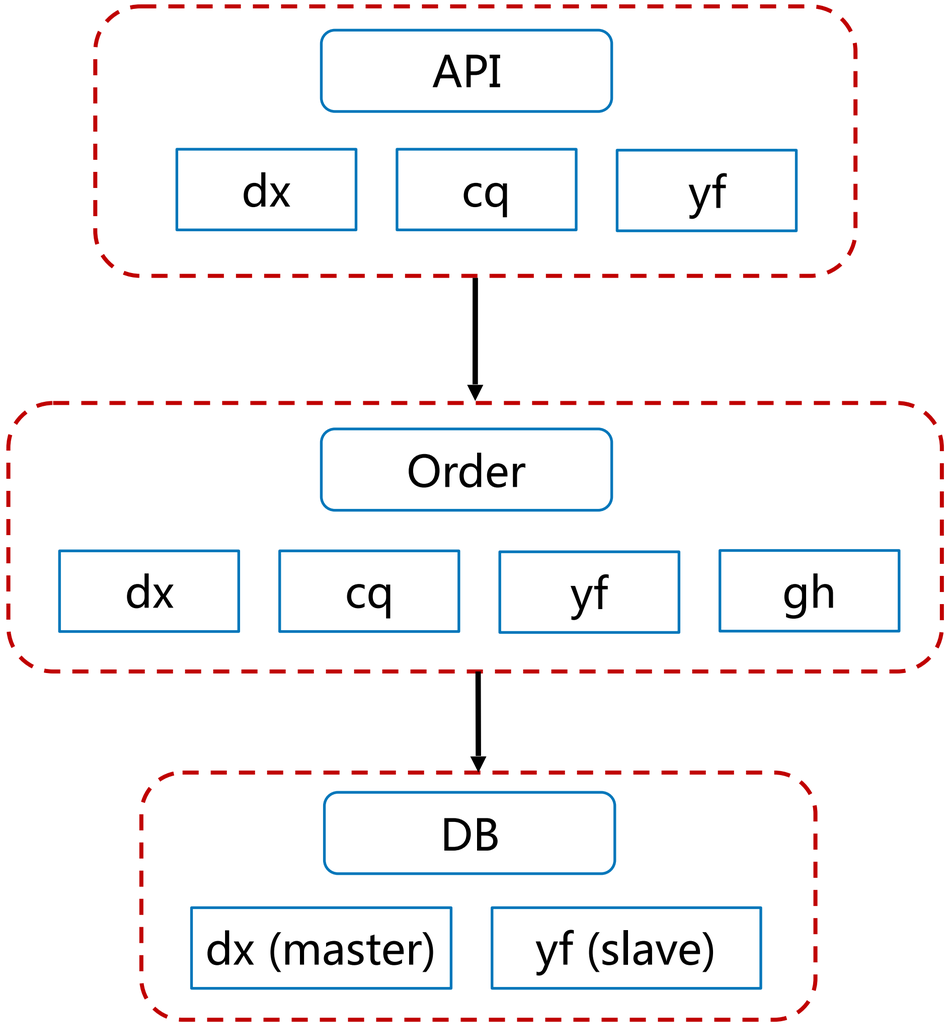
压测

故障演练

高可用实战

酒店订单系统高可用

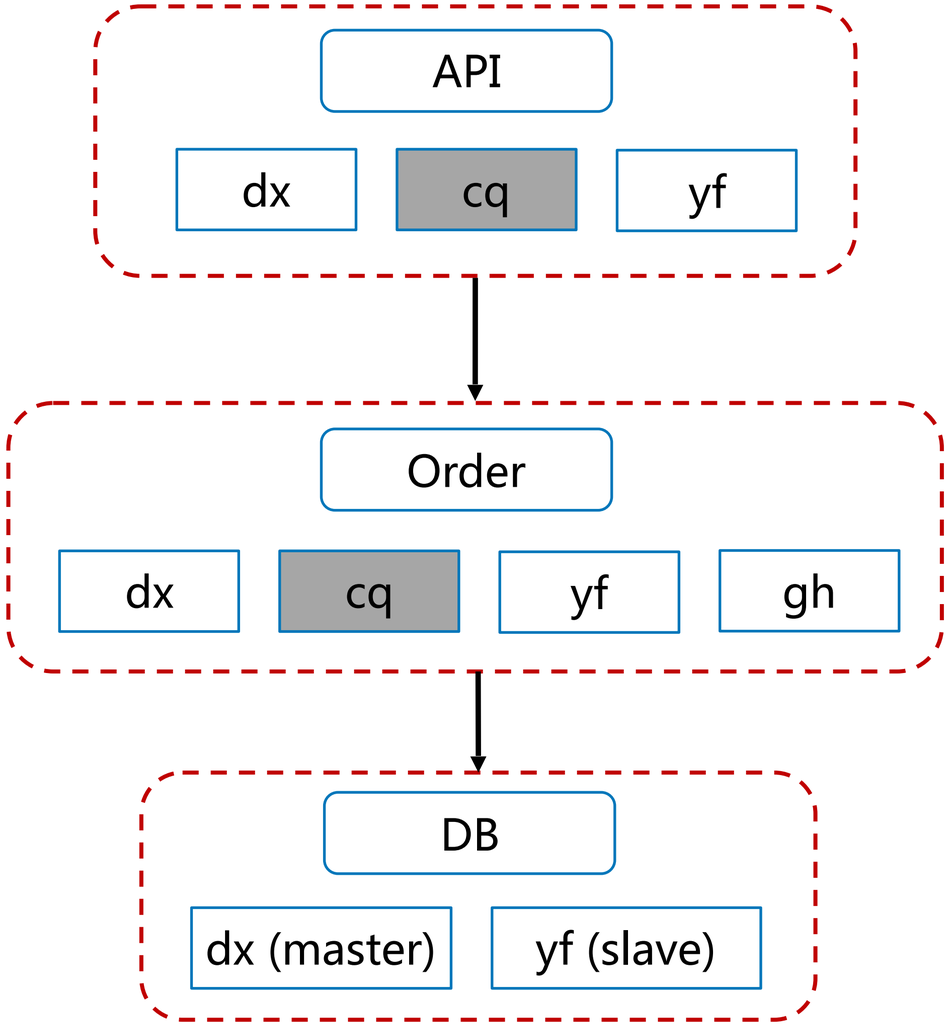
- 服务容量
  - 服务冗余



高可用实战

酒店订单系统高可用

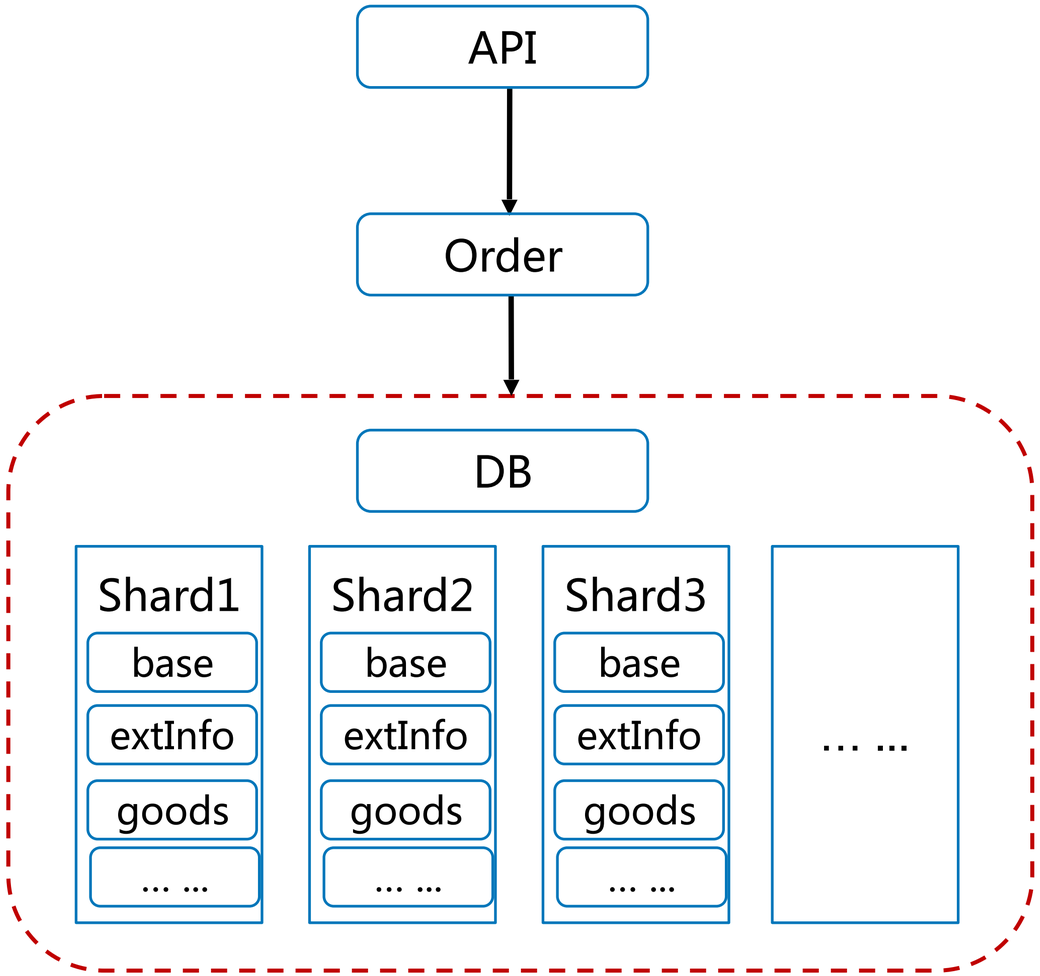
- 服务容量
  - 服务冗余



高可用实战

酒店订单系统高可用

- 容量
  - 逻辑独立
  - 方便扩容
  - 故障影响



高可用实战

提高系统故障间隔

A 理依赖

依赖分级

熔断保护

B 做隔离

快慢分离

读写分离

冷热分离

C 定规矩

上线SOP

D 估容量

服务容量

存储容量

E 做自检

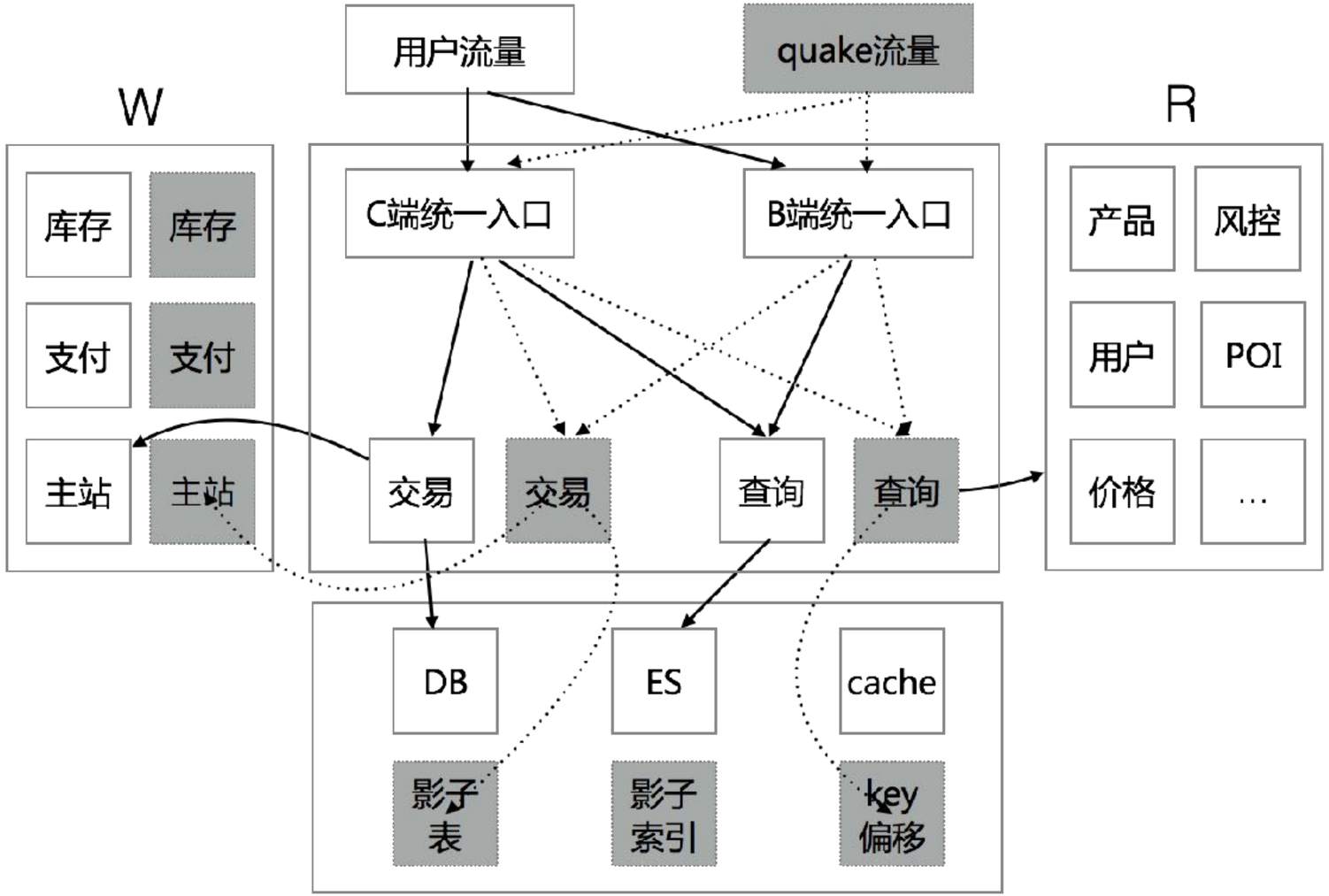
压测

故障演练

高可用实战

酒店订单系统高可用

- 自检
  - 压测
    - 读压测
    - 写压测
    - 混合压测
    - 全链路压测



高可用实战

酒店订单系统高可用

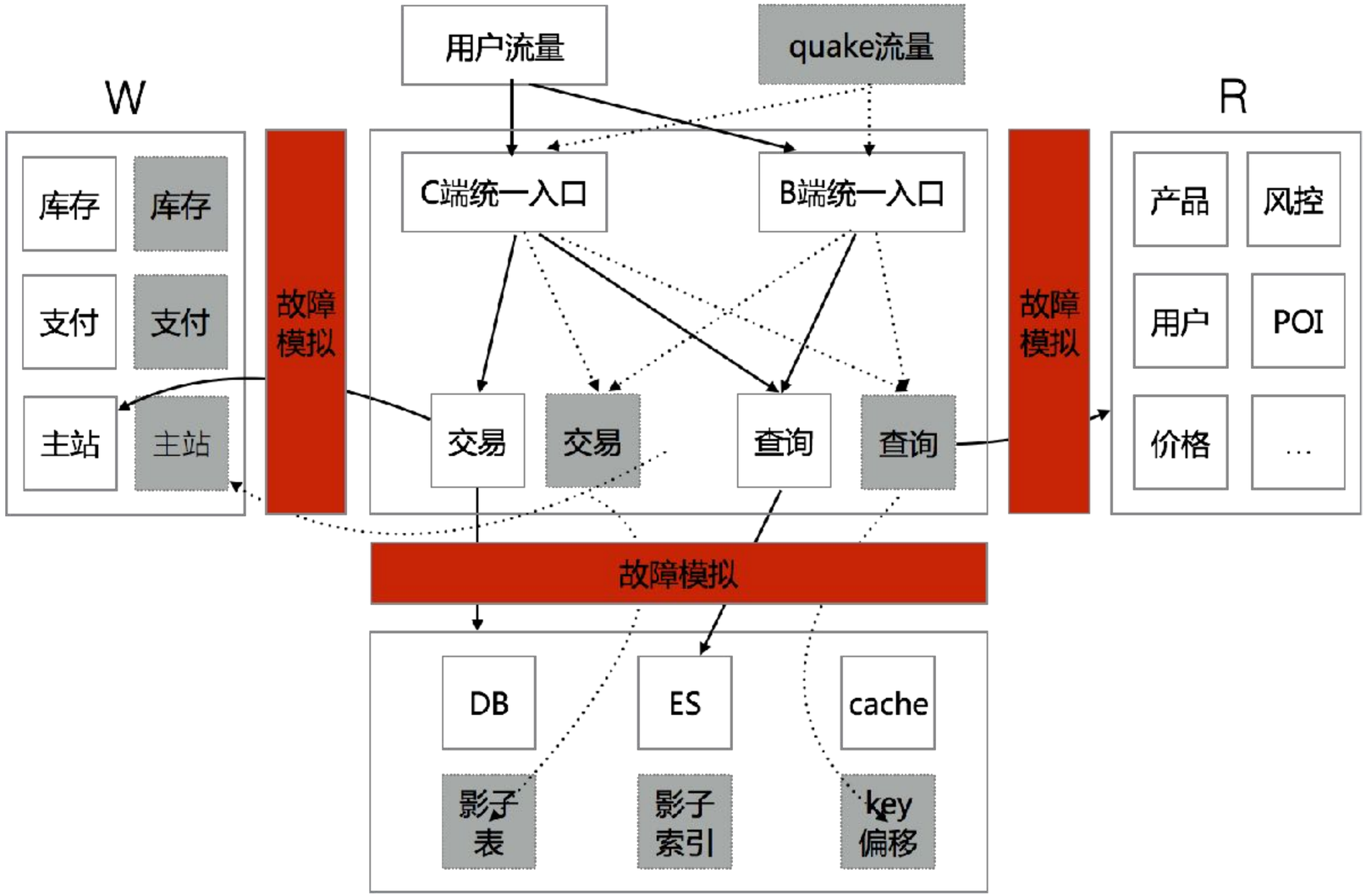
- 自检
  - 压测
  - 压测前：定目标、有预案、做周知
  - 压测中：看监控、看报警
  - 压测后：做总结、洗数据

- 压测阶段性复盘
- 阶段一-读压测-订单中心订单详情接口
- 阶段三-12-29apic下单前置页压测统计
- 阶段三-12-29日apic下单前置页压测计划与压测结果
- 阶段三-12月21日读写混合压测数据统计
- 阶段三-apib读压测报告
- 阶段三-apib读压测-线上压测执行计划
- 阶段三-apib读压测补充计划
- 阶段三-apic生单接口压测执行计划
- 阶段三-支付回调压测执行计划
- 阶段三-读写混合压测执行计划
- 阶段三－支付回调压测报告
- 阶段二-12月11日写压测-下单接口压测统计
- 阶段二-12月14日写压测-下单接口压测统计
- 阶段二-写压测-生单接口
- 阶段二-读压测-线上压测执行计划
- 阶段二-读压测-订单详情结果统计-(读cache)

高可用实战

酒店订单系统高可用

- 自检
  - 故障演练
  - 依赖处理验证
  - 监控报警验证
  - SOP验证



高可用实战

降低系统故障恢复时间

A 监控

监控分层

在线与离线

B 报警

快慢分离

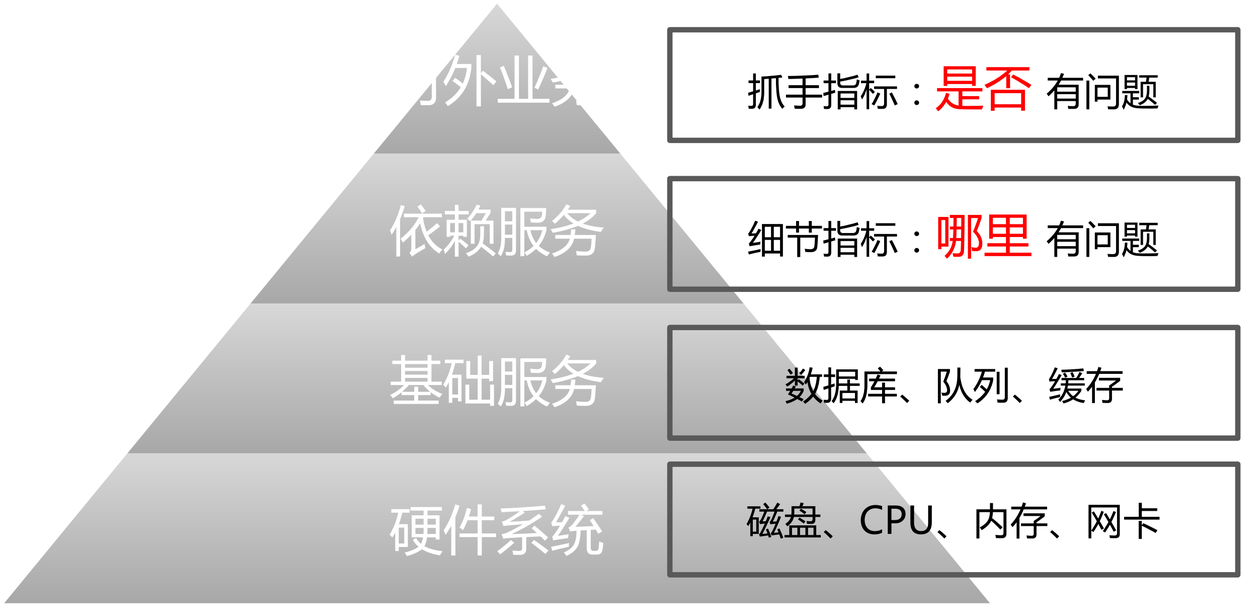
读写分离

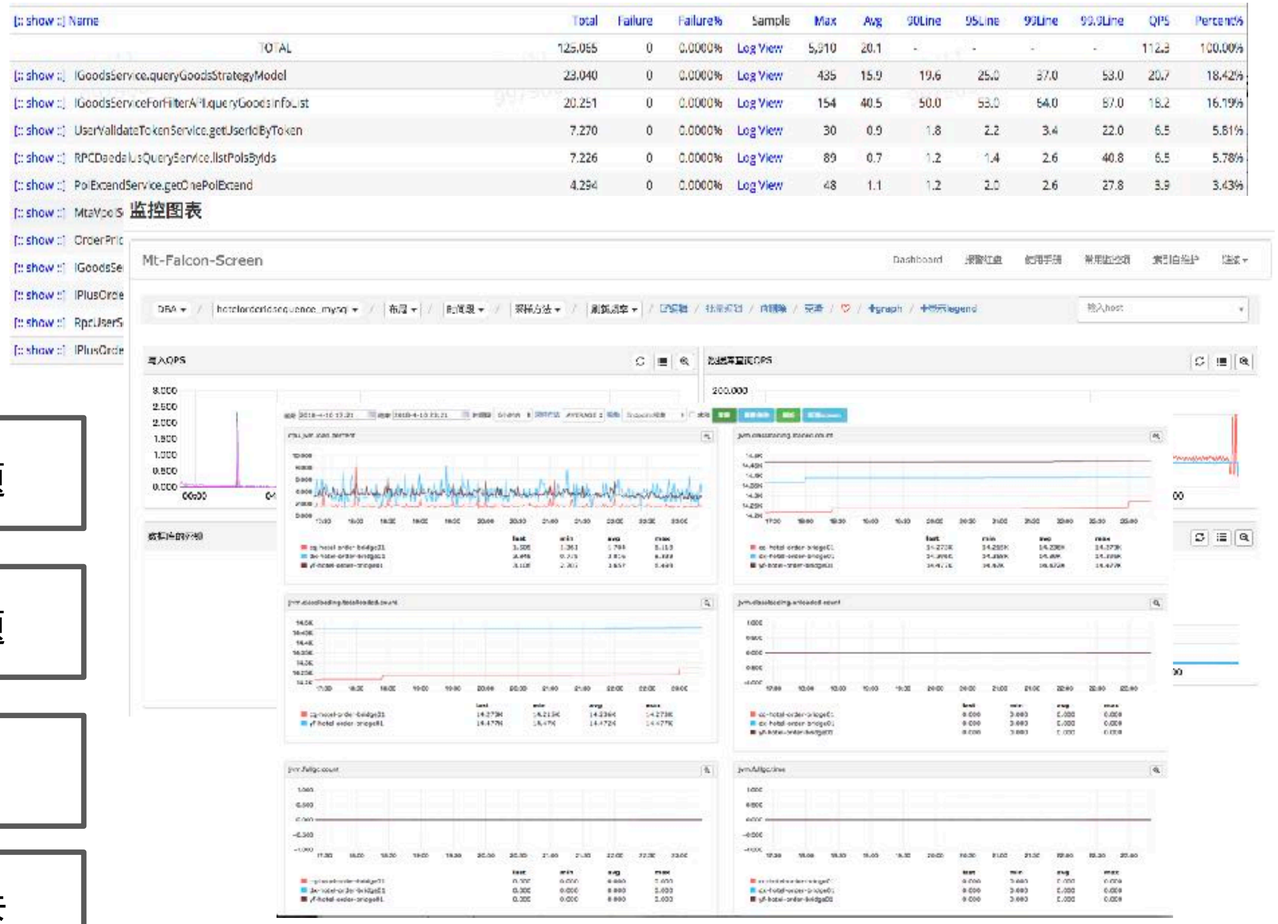
冷热分离

高可用实战

酒店订单系统高可用

- 监控
  - 监控是分层的
  - 在线 & 离线





高可用实战

降低系统故障恢复时间

A 监控

依赖分级

熔断保护

B 报警

报警分级

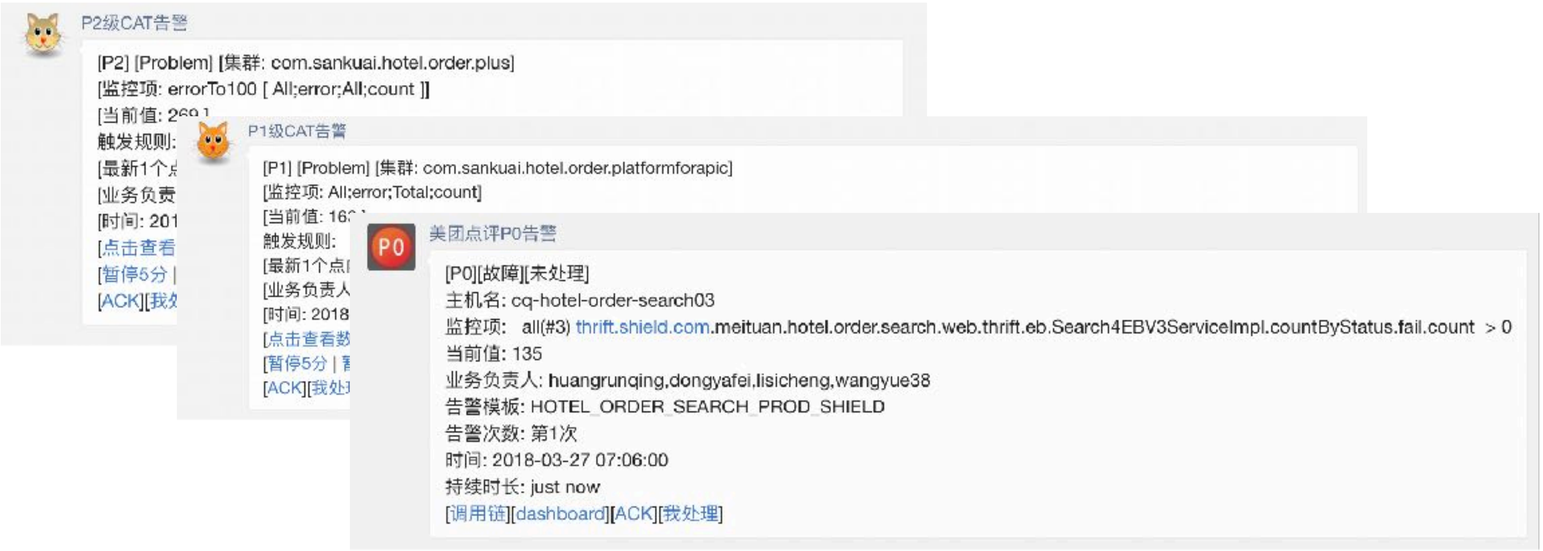
SOP

运维工具

高可用实战

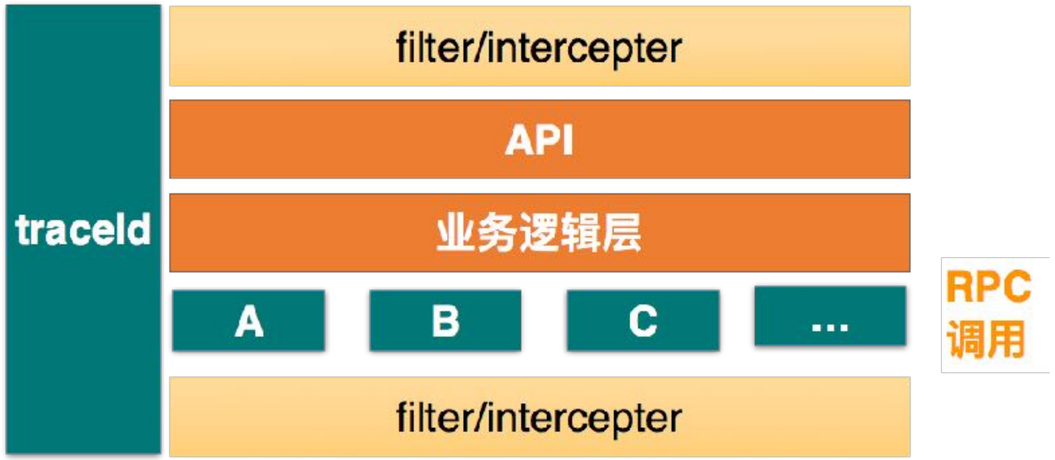
酒店订单系统高可用

- 报警
  - 报警分级
  - SOP
  - 运维工具



订单粒度：timeline

请求粒度：traceId



高可用实战

其它措施

A 人员单点

核心服务排查

核心链路排查

B 系统复杂度

运维成本

理解成本

开发成本

C 谨慎行事

系统升级

组件升级

高可用实战

酒店订单系统高可用

- 人员单点
  - 核心链路排查
  - 核心服务排查
  - 人员单点 ≈ 系统单点

CaseStudy-20170912-[订单中心组]-订单生单一定概率失败

创建： 许关飞，最新修改： 李磊 于 九月 21, 2017

三、为什么从开始排查到确认问题耗时8min

[redacted]

[redacted]

2. id生成器部分代码除开发同学外其他人熟悉程度不够，恰巧跟上开发同学问题发生时出去办事，导致排查耗时较长

高可用实战

其它措施

A 人员单点

核心服务排查

核心链路排查

B 系统复杂度

运维成本

理解成本

开发成本

C 谨慎行事

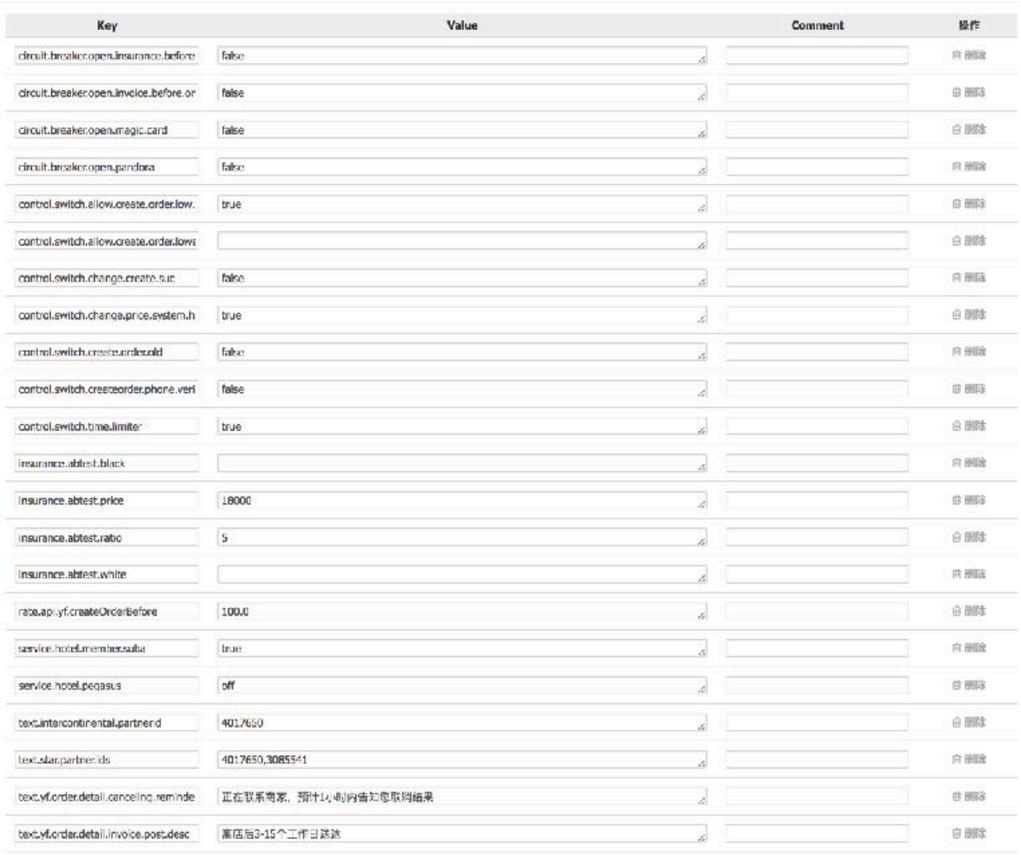
系统升级

组件升级

高可用实战

酒店订单系统高可用

- 系统复杂度
  - 降低运维成本
  - 降低理解成本
  - 降低开发成本



Key	Value	Comment	操作
circuit.breaker.open.insurance.before	false		删除
circuit.breaker.open.invoice.before.or	false		删除
circuit.breaker.open.magic.card	false		删除
circuit.breaker.open.pandora	false		删除
control.switch.allow.create.order.low.	true		删除
control.switch.allow.create.order.lowe			删除
control.switch.change.create.suc	false		删除
control.switch.change.price.system.h	true		删除
control.switch.create.order.old	false		删除
control.switch.createorder.phone.veri	false		删除
control.switch.time.limiter	true		删除
insurance.abtest.black			删除
insurance.abtest.price	18000		删除
insurance.abtest.ratio	5		删除
insurance.abtest.white			删除
raze.ap..yf.createOrderBefore	100.0		删除
service.hotel.member.suba	true		删除
service.hotel.pegasus	off		删除
text.intercontinental.partnerid	4017650		删除
text.star.partner.ids	4017650,3085541		删除
text.yf.order.detail.canceling.reminde	正在联系商家，预计1小时内告知您取消结果		删除
text.yf.order.detail.invoice.post.desc	离店后3-15个工作日送达		删除

Key	Value	Comment	操作
dirtyOrderId	281475608328756	* 不需要处理的脏订单号，逗号分	保存 删除

高可用实战

其它措施

A 人员单点

核心服务排查

核心链路排查

B 系统复杂度

运维成本

理解成本

开发成本

C 谨慎行事

系统升级

组件升级

高可用实战

酒店订单系统高可用

- 谨慎行事
  - 系统升级
  - 组件替换

CaseStudy-20180122-熔断导致微信钱包外卖无法登录
创建： 岳国栋，最新修改： 袁俊子 大约1小时以前

Casestudy-20160724 mq内存异常导致下单支付异常
创建： 徐方华，最新修改于： 十月 20, 2016

组件升级小坑记
创建： 章鑫，最新修改于： 十一月 09, 2017

zebra迁移总结
创建： 章鑫，最新修改： 柯宇 于 三月 14, 2018

memcache-squirrel迁移总结
创建： 王斐，最新修改： 昨天9:39 下午

高可用实战

酒店订单系统高可用

别人死 我们不死	不作死 就不会死	我觉得 我还可以抢救一下
	上线SOP	
依赖分级	服务容量	监控分层
熔断保护	存储容量	在线 & 离线
快慢分离	压测	报警分级
读写分离	故障演练	SOP
冷热分离	人员单点	运维工具
	系统复杂度	
	谨慎行事	

Q&A