高可用系统在大众点评的实践与经验

陈一方

大众点评 dianping.com

目录

大众点评 dianping.com

- 可用性的理解
  - 理解目标
  - 分解目标
- 频率要低
  - 高可用设计
  - 易测试
  - 易运营
  - 重视发布
- 时间要快
  - 持续关注运营时
  - 快速的发现机制
  - 有效的恢复机制
- 几点经验
  - 珍惜真实流量
  - 复盘
  - 可用性不只是技术问题
  - 可用性最大的敌人

可用性的理解 - 理解目标

大众点评 dianping.com

Level	每年宕机时间	每天宕机时间
9	36.5 days	2.4hrs
99	3.65 days	14min
99.9	8.76 hrs	86sec
99.99	52.6 min	8.6sec
99.999	5.25 min	0.86sec
99.9999	31.5 sec	8.6 ms

可用性的理解 - 分解目标

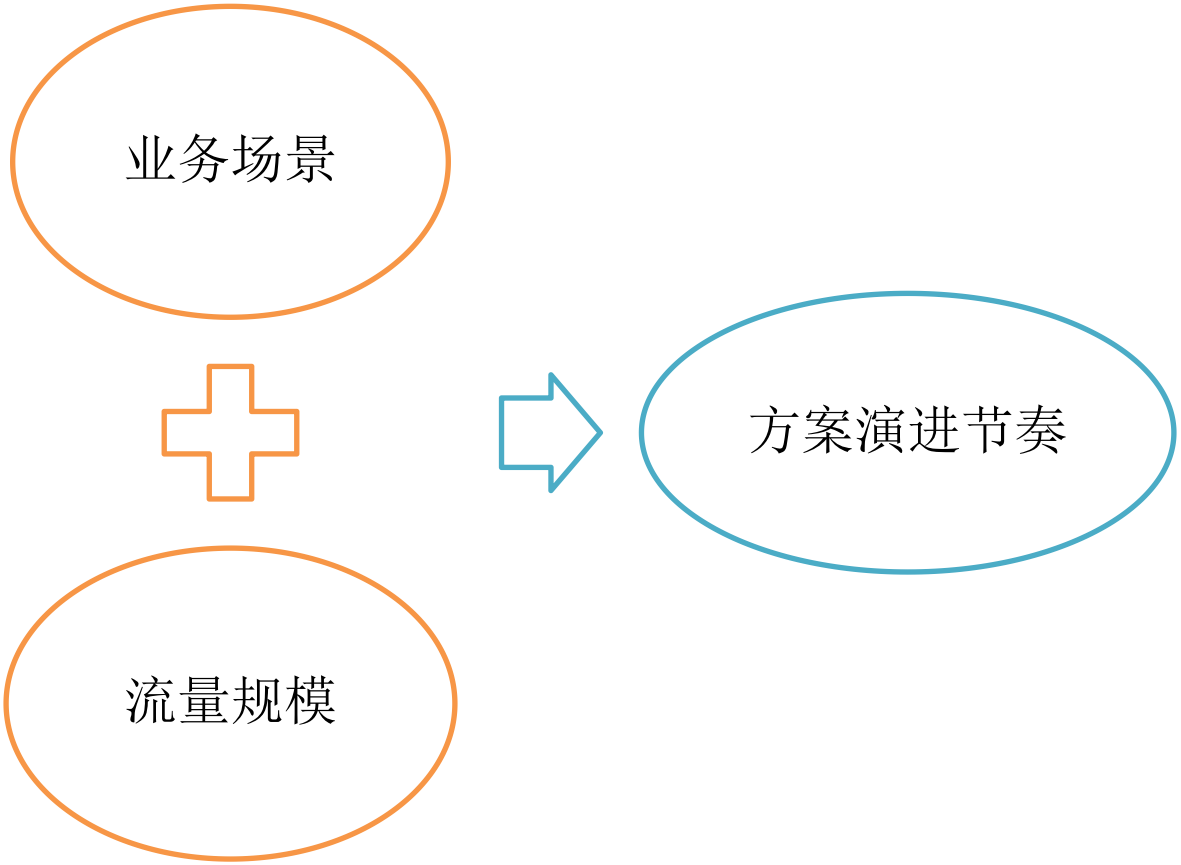
MTBF :Mean Time Between Failures

- 频率要低

MTTR:Mean Time To Recover

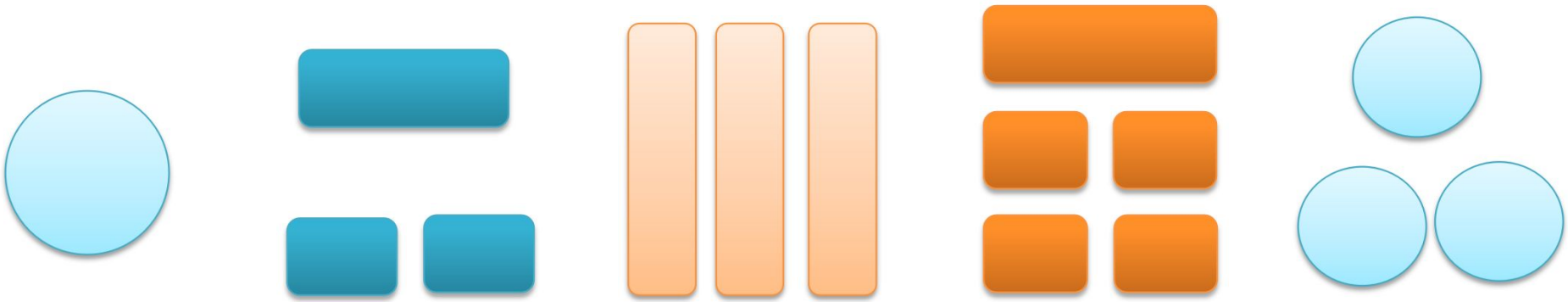
- 时间要快

频率要低 - 高可用性的设计



高可用性的设计 - 演进节奏

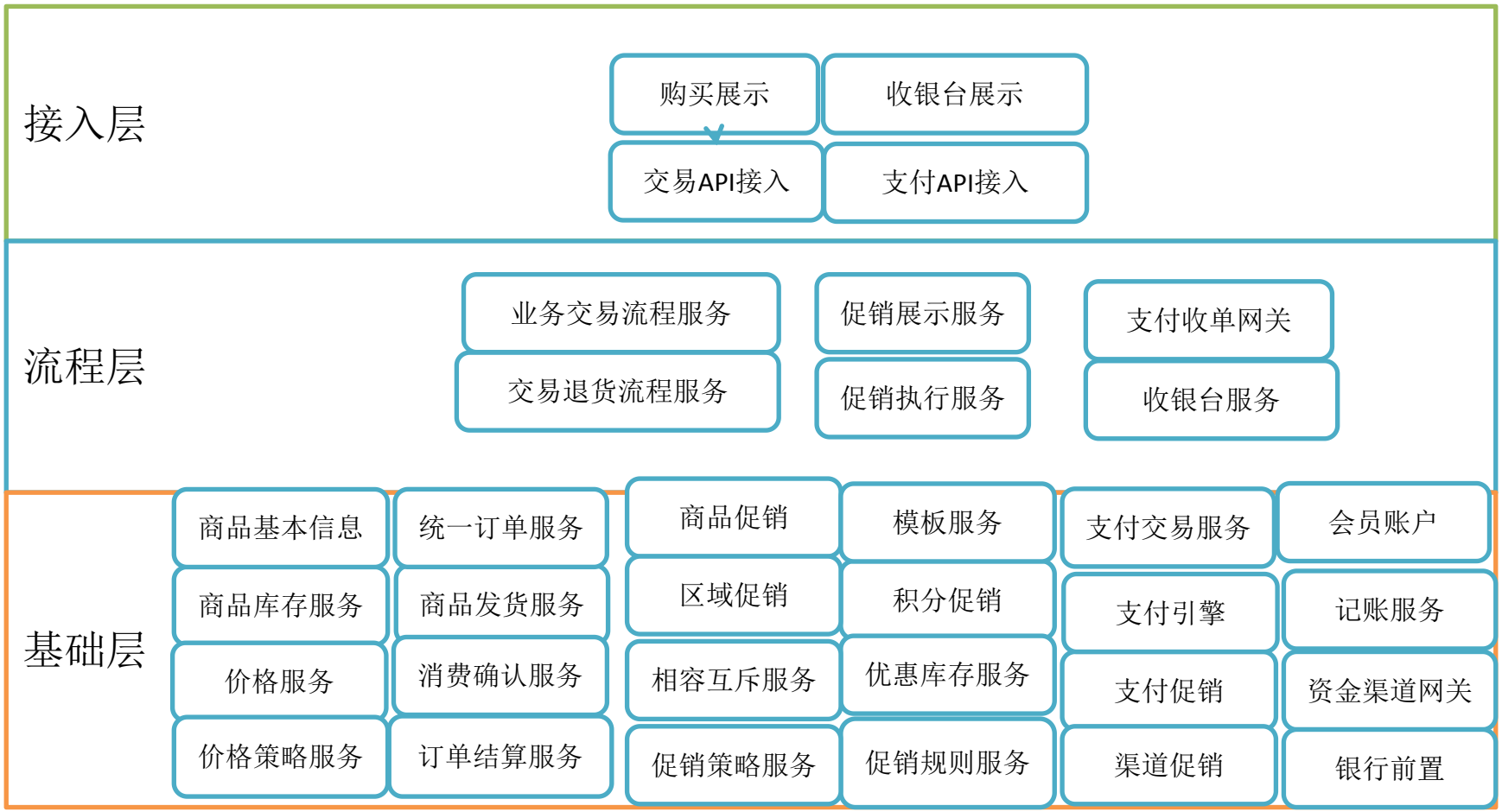
化简为繁→化繁为简



一个系统 → 模块化 → 垂直服务化 → 平台服务化 → 化整为零

高可用性的设计 - 方案节奏的重要性

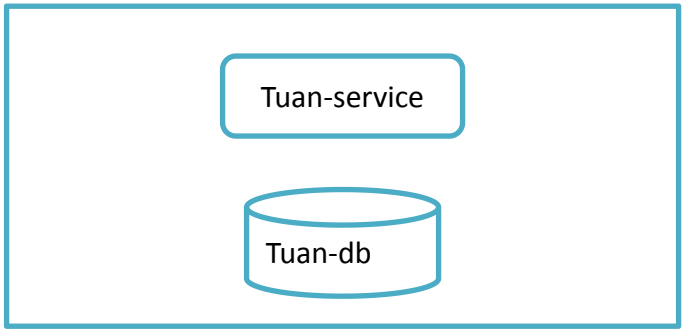
高可用的架构很容易找到方案，难点是演进的节奏把握
以点评交易的一些服务组件演进为例，我们如何一步步走到如图所示的。



高可用性的设计 - 点评交易系统演进为例

幼儿时期：2010-2012。流量规模较小，万级订单

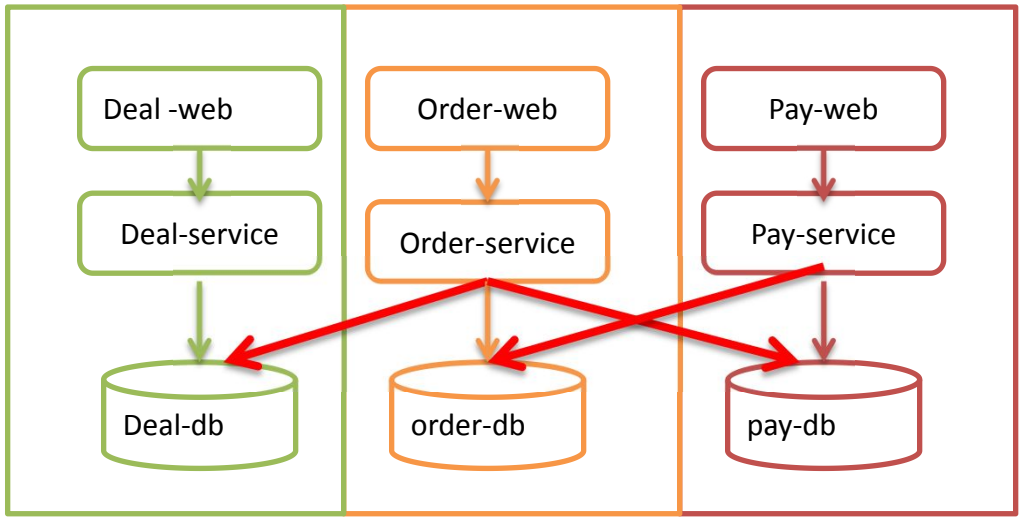
使命：满足业务要求，快速上线



高可用性的设计 - 什么时候该垂直拆分？

少年时期：垂直拆分 2012-2013。 流量规模中等，十万级订单

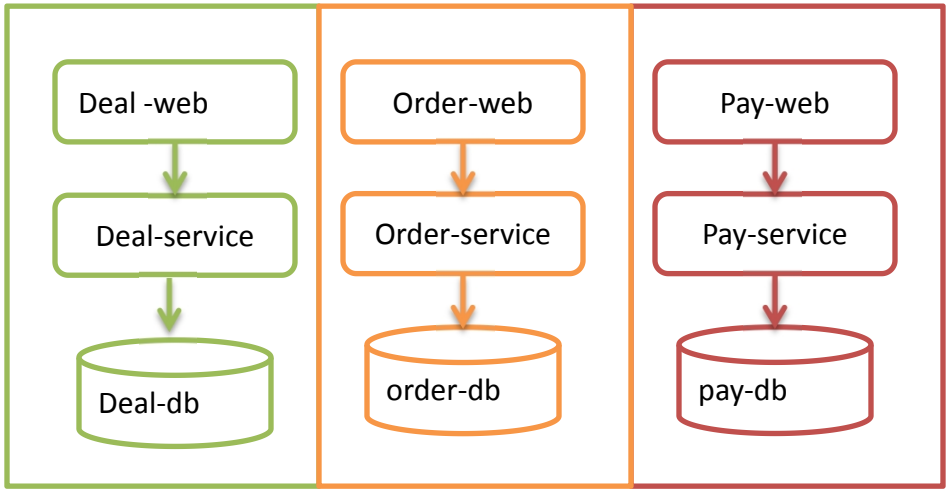
使命：研发效率&故障隔离



高可用性的设计 - 什么时候全面服务化？

青年时期：服务做小，不共享数据 2014-2015。 流量规模大，日订单量百万级

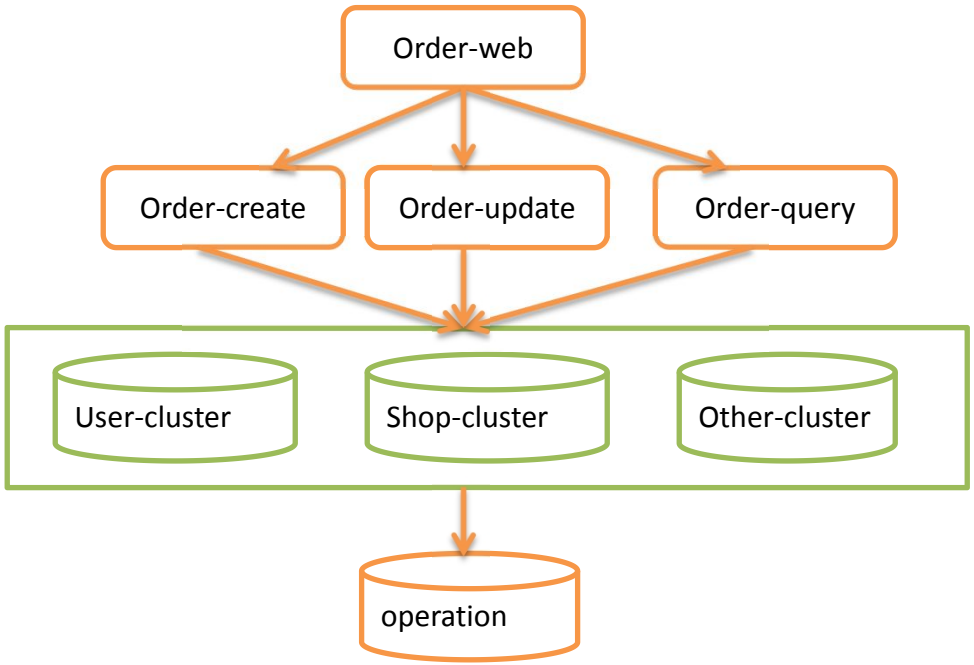
使命：支撑业务快速发展，提供高效、高可用的技术能力



高可用性的设计 - 什么时候水平拆分？

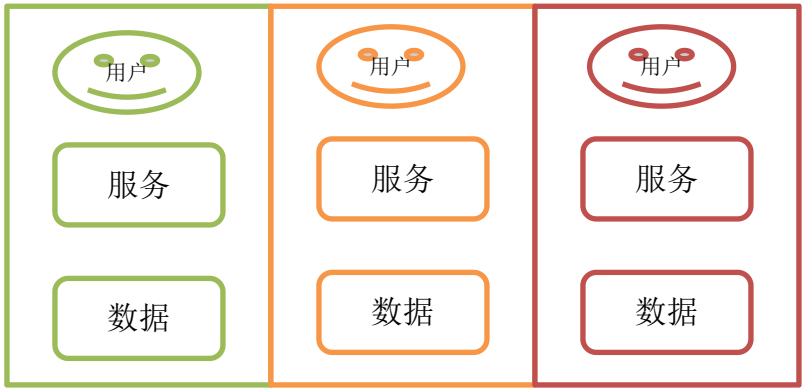
成年时期：水平拆分 2015-至今

使命：支撑大规模的促销活动，系统能支撑每秒几万的QPS，日千万订单量



高可用性的设计 - 流量再涨10倍怎么办？

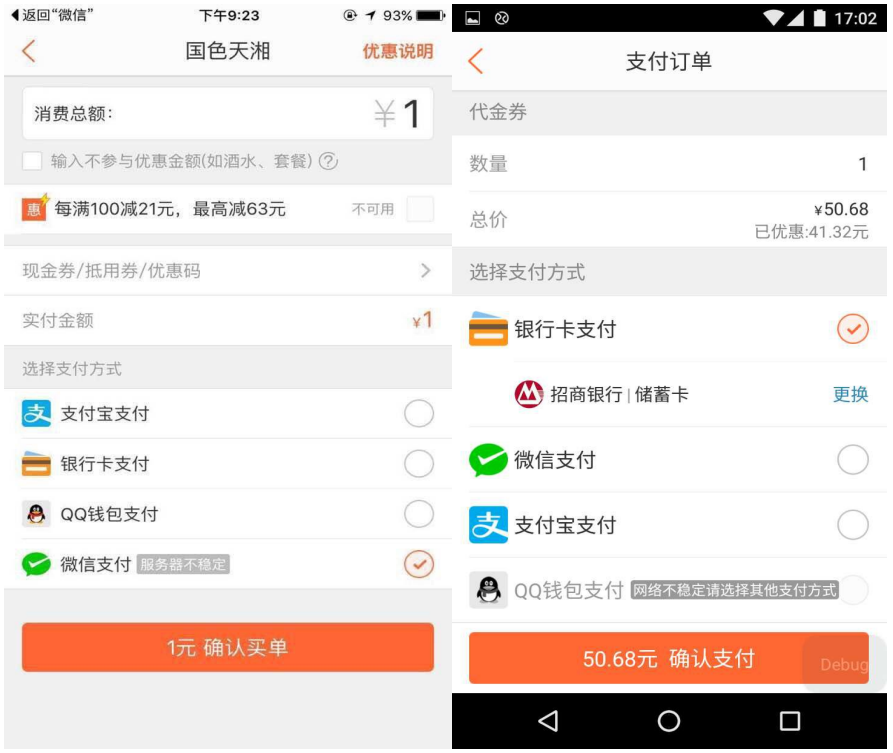
未来：思路仍然是大系统做小，基础通道做大，流量分块



频率要低 - 易运营

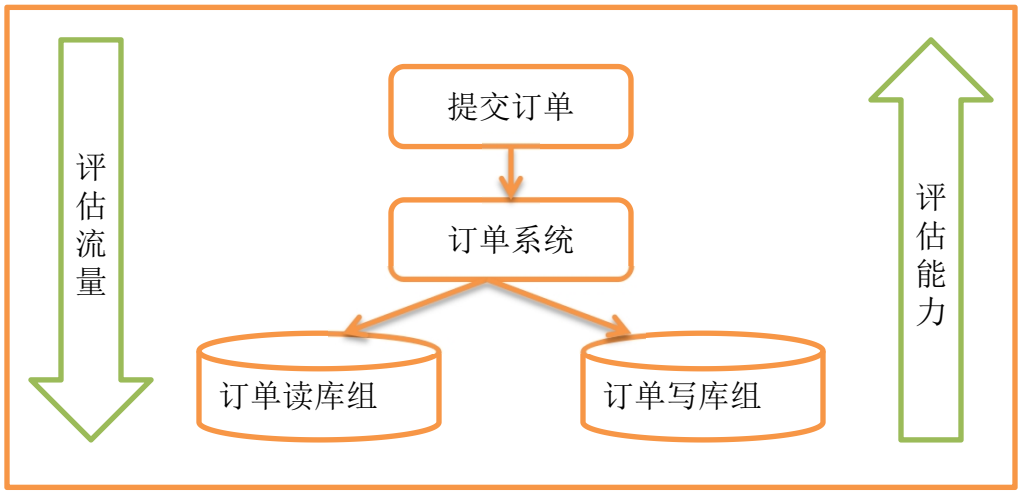
高可用的系统一定是可运营的

- 可限制流量
- 无状态
- 降级能力



频率要低 - 易测试

测试经验：从上到下评估流量，从下到上评估能力



频率要低 - 重视发布

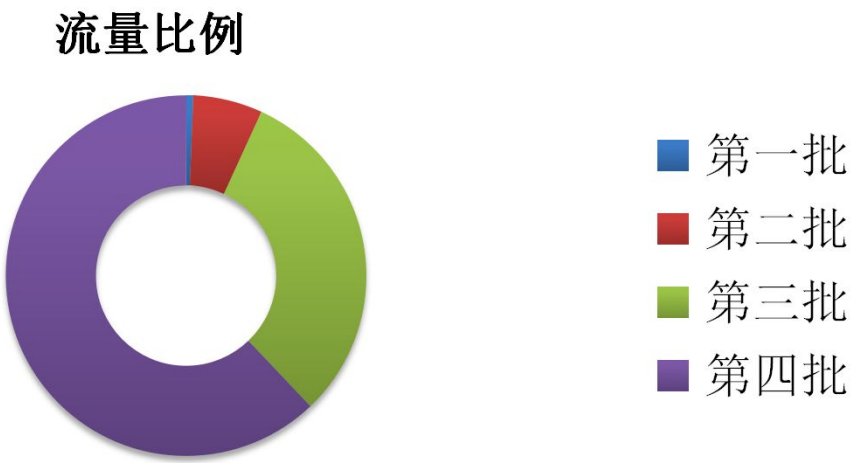
大众点评 dianping.com

严格的发布流程，可回滚是标配

计划上线结束时间：	2015-07-30 16:07:27
本次上线RD负责人：	chaofan.wu
升级内容：	alipay退款通知， BUYER_ERROR错误码转强提
项目链接：	-
是否高峰期上线：	否
是否要做预发检查：	是
是否灰度发布：	是
是否计划内上线：	是
上线过程是否需要OP配合：	否

频率要低 - 重视发布

灰度机制：灰度发布&灰度运营



时间要快 - 持续关注运行时

大众点评 dianping.com

熟悉系统及下游的容量、响应时间、QPS 不同时间流量变化

熟悉系统的业务指标变化，尽可能打点到方法级

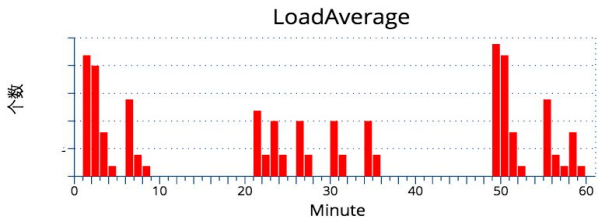
Type	Total	Failure	Failure%	Sample Link	Min(ms)	Max(ms)	Avg(ms)	95Line(ms)	99.9Line(ms)	Std(ms)	QPS
[:: show ::] System	2,757	0	0.0000%	Log View	19	161.3	31.6	36.2	49.8	8.9	0.8
[:: show ::] PigeonService	22,718,310	0	0.0000%	Log View	0	587	22.7	59.3	136.6	17.4	6,310.6
[:: show ::] PigeonCall	28,951,218	14,492	0.0501%	Log View	0	406	11.6	30.0	103.8	12.9	8,042.0

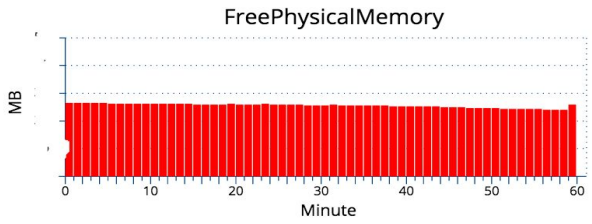
时间要快 - 持续关注运行时

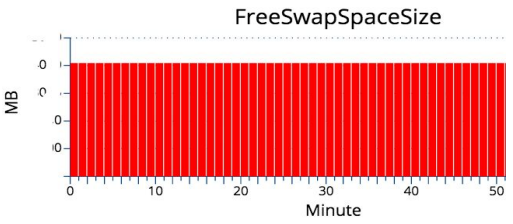
大众点评 dianping.com

熟悉系统所在服务器、网络、数据库等的系统结构和运行状态

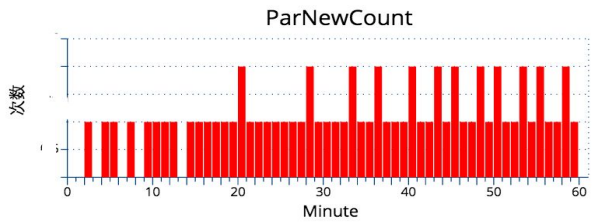
System Info

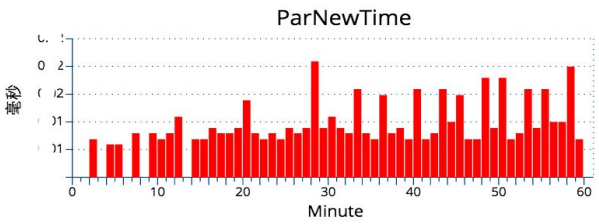


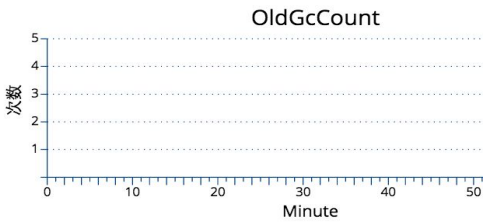




GC Info







时间要快 - 快速的发现机制

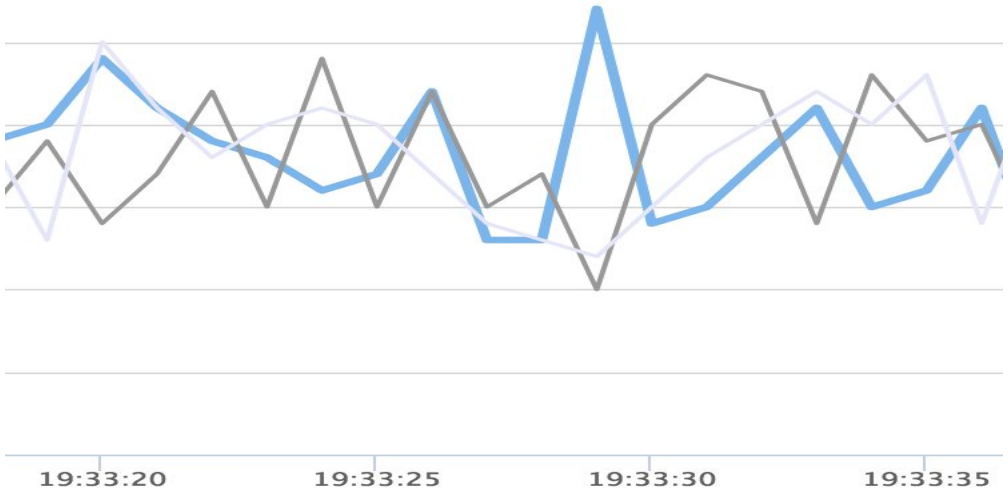
大众点评 dianping.com

故障出现后，时间分类

告警的移动化

监控的实时化

监控的可视化



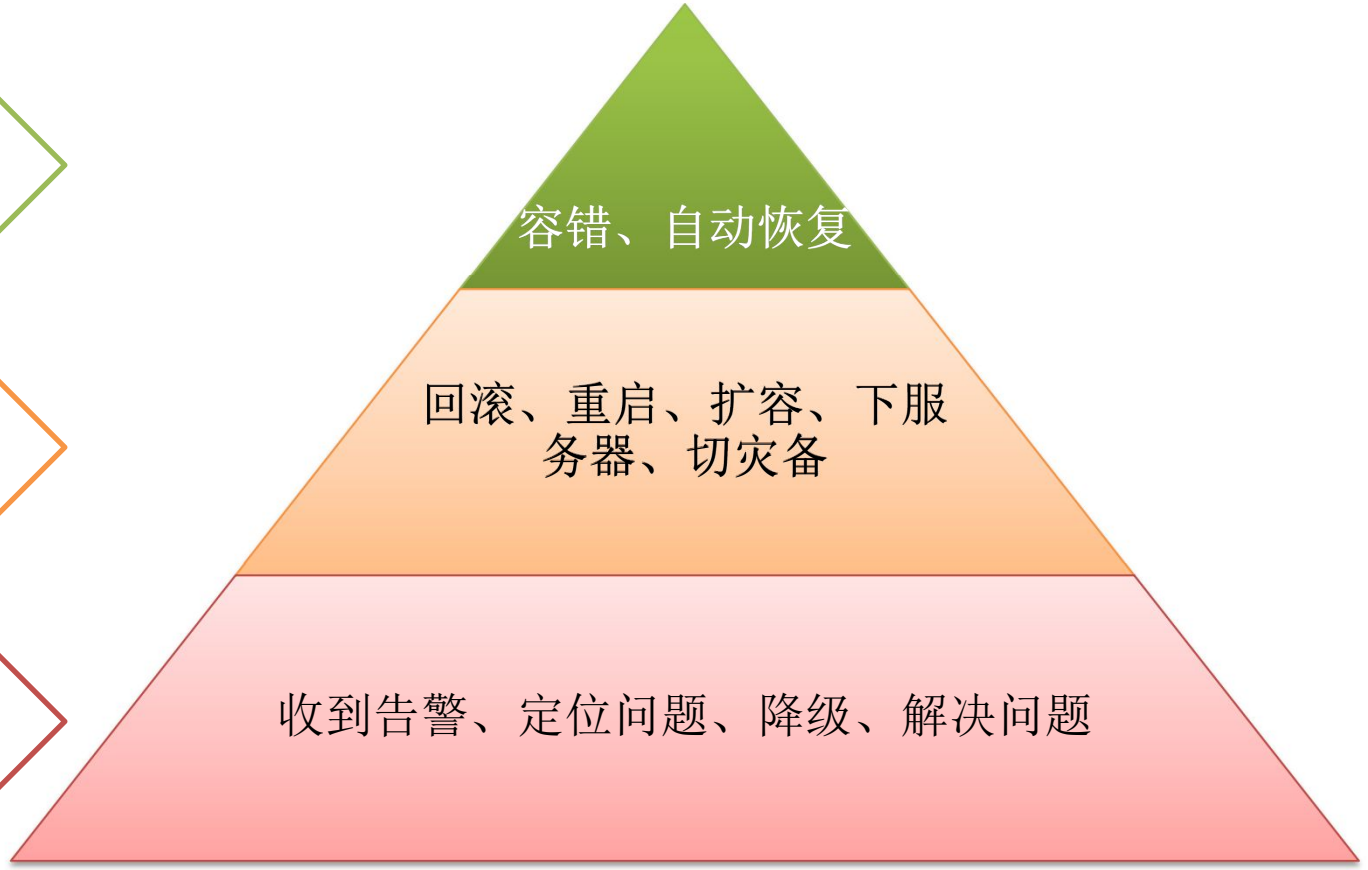
时间要快 - 有效的恢复机制

大众点评 dianping.com

上策：系统

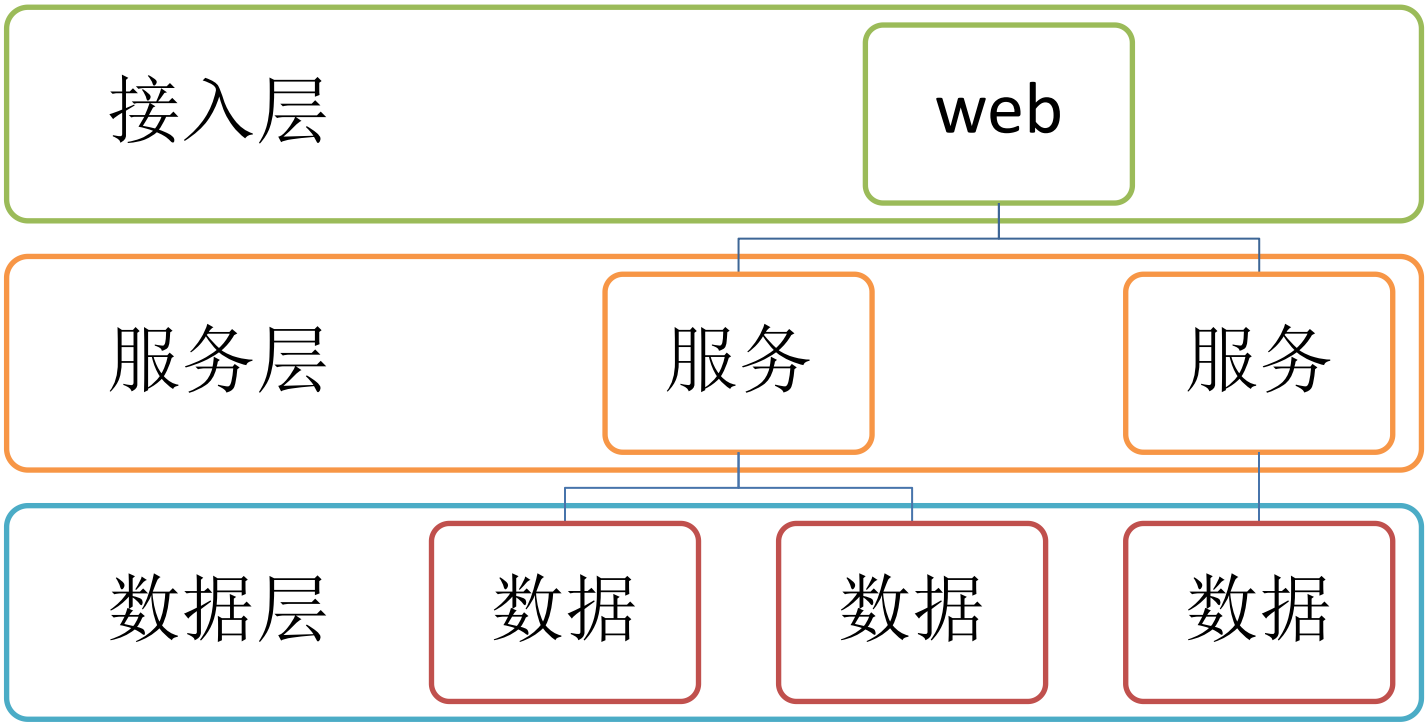
中策：运维

下策：研发



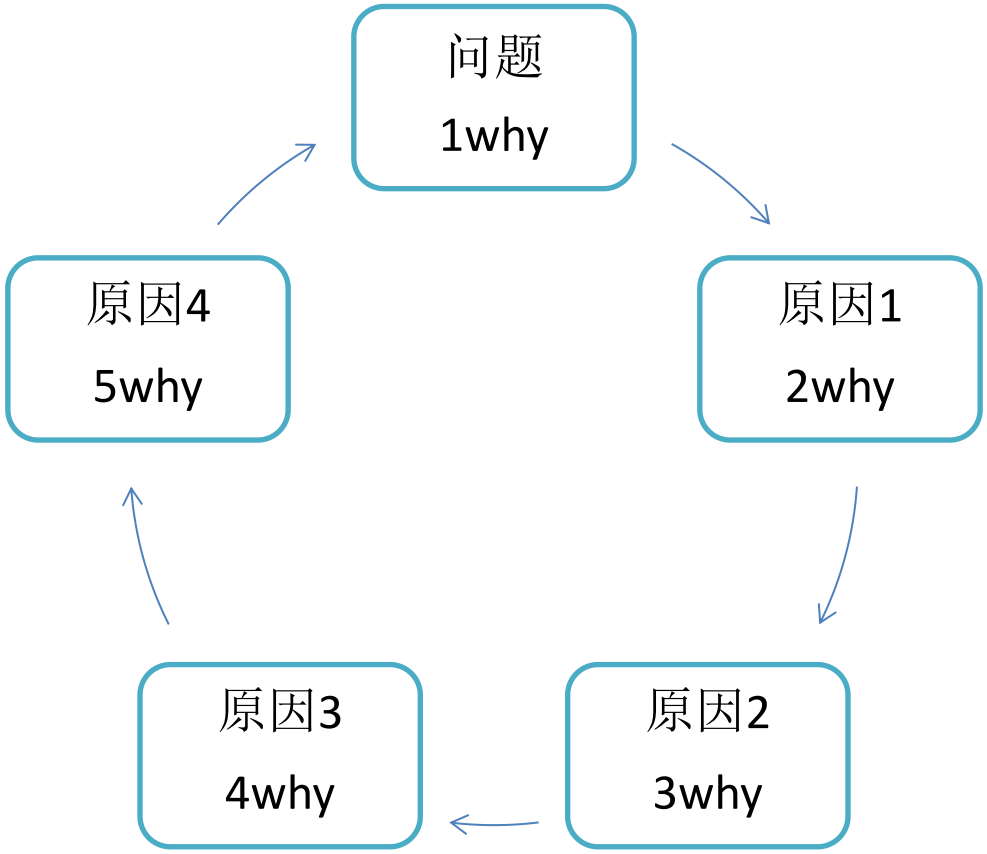
几点经验 - 珍惜每次大流量机会

珍惜每次真实高峰流量，建立高峰期流量模型



几点经验 - 复盘

珍惜每次线上故障复盘，上一层楼看问题，下一层楼解决问题



几点经验 - 可用性不只是技术问题

大众点评 dianping.com

系统初期

- 研发

系统中期

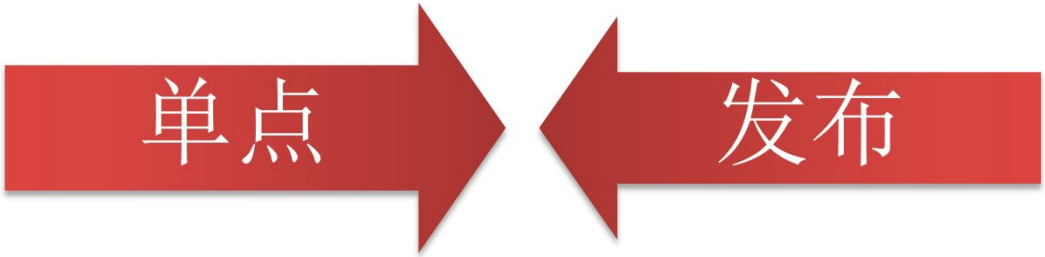
- 研发、运维、DBA

系统后期

- 研发、运维、DBA、产品

几点经验 - 可用性最大的敌人

大众点评 dianping.com



几点经验 - 发展的眼光看

大众点评 dianping.com

一切现有架构都是错的

谢谢大家

大众点评 dianping.com

Q&A