

火龙果·整理
uml.org.cn

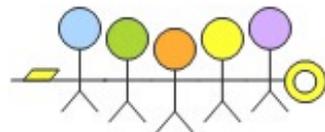
高可用架构设计与实践

代表58对外交流

- Qcon(全球软件开发大会)
- SDCC(中国开发者大会)
- Top100(全球案例研究峰会)
- DTCC(中国数据库技术大会)
- 《程序员》撰稿2次
- 58技术发展这10年[计划中]



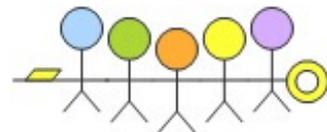
炼数成金课程



火龙果·整理
uml.org.cn

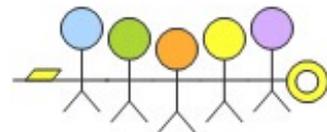
课程

- 《MongoDB 实战》
 - 已开课
 - 欢迎大家报名学习
- 《大规模高性能分布式存储系统设计与实现》
 - 已开课
 - 欢迎大家报名学习



上次课程回顾

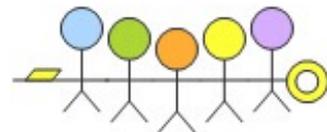
- 👤 高可用课程背景
- 👤 高可用课程安排
- 👤 什么是架构的高可用
- 👤 架构高可用重要性
- 👤 架构高可用的手段都有哪些
- 👤 架构高可用评价维度是什么
- 👤 架构高可用的如何分级&考核
- 👤 架构高可用的涉及环节
- 👤 典型问题你遇到过吗?



架构高可用常遇到问题

🐛 架构高可用遇到问题

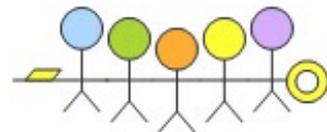
- 上线发生数据改动，格式和之前不兼容，回滚也不正常，如何处理？



架构高可用常遇到问题

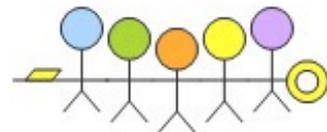
🐛 架构高可用遇到问题

- 上线更新删除了数据，回滚后数据也没有了，如何处理？



OutLine

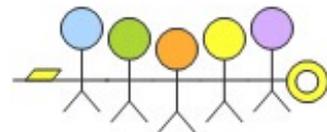
- 👤 高可用架构为什么需要分层?
- 👤 高可用架构分层设计原则是什么?
- 👤 高可用架构如何分层?
- 👤 高可用架构分层最佳实践
- 👤 我们的实践案例



高可用架构为什么需要分层

ALL IN ONE 架构

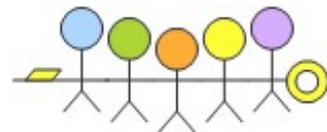
- 整个架构只有一个模块
 - 数据部分、逻辑部分、接入部分、展示部分等
- 架构存在问题
 - 耦合严重
 - 职责不分明
 - 模块庞大、臃肿
 - 开发成本高、效率低下
 - 运维成本高
 - 组件间相互影响，一旦一个组件有问题，整个服务都受影响
 - 扩展性差
 - 性能极限差
 - 牵一发而动全身!!!!



高可用架构为什么需要分层

🐼 高可用架构分层

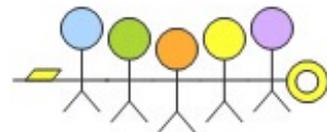
- ALL IN ONE架构问题多多
- 服务高可用需分层设计
- 模块耦合性低
- 模块职责分明
 - 数据层、逻辑层、接入层、展示层等等
- 模块间不再相互影响
- 模块独立扩展
- 系统整体性能高



高可用架构分层设计原则是什么

高可用架构分层设计原则

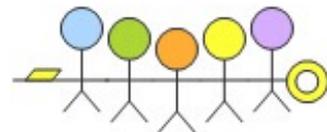
- 数据服务和逻辑服务分离
 - 数据存储
 - 业务逻辑
- 逻辑服务和接入服务分离
 - 业务逻辑
 - 接入层
- 接入服务和展示服务分离
 - 接入层
 - 数据展示



高可用架构分层设计原则是什么

👤 高可用架构分层设计原则

- 分层服务功能单一
 - 数据
 - 逻辑
 - 接入
 - 展示
 -
- 分层间低耦合
 - 接口交互
- 分层内高内聚
 - 功能聚焦单一

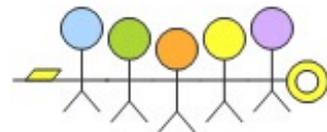


高可用架构分层设计原则是什么

高可用架构分层设计原则

— 分层适中

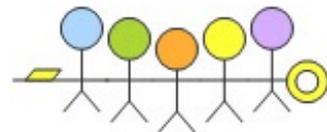
- 层次过多
 - 请求交互路径长
 - 请求响应延迟高
 - 层次多，运维成本高
 - 定位问题涉及层次多，定位复杂多增加，定位时间长
- 层次过少
 - 每个层次功能不单一，耦合性高
 - 模块内组件间相互影响高
 - 高可用无法保证



高可用架构如何分层

高可用架构分层

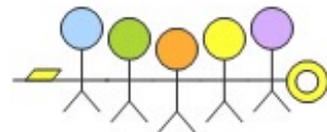
- 前端架构
 - MVC架构分层
- 后端架构
 - 按照功能水平划分
 - 四层
 - » 接入层、逻辑层、数据层、数据存储
 - » 接入层、逻辑层、原子服务层、数据存储
 - 五层
 - » 接入层、序列化层（异步消息队列）、逻辑层、数据层、数据存储
 - 按照业务垂直拆分
 - 房产、招聘、二手、二手车、黄页
 - IM、交友等



高可用架构分层最佳实践

高可用架构最佳实践

- 脱离业务场景谈架构分层绝对是要流氓
 - 架构的分层取决于业务场景
 - MVC
 - 三层
 - 四层
 - 五层



高可用架构分层最佳实践

高可用架构最佳实践

— 脱离业务场景谈架构分层绝对是耍流氓

- 架构的分层取决于业务发展阶段

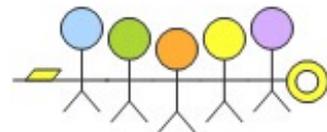
- 创业初期

- » 满足业务快速发展

- » 可用性低

- » 分层少

- » ALL IN ONE



高可用架构分层最佳实践

🐼 高可用架构最佳实践

- 脱离业务场景谈架构分层绝对是要流氓
 - 架构的分层取决于业务发展阶段
 - 业务数据量、请求量快速增长期
 - » 引入分层
 - » 接入层、逻辑层、数据存储等
 - » 满足业务增长需求
 - 业务请求高并发，海量存储期
 - » 每层进一步细化
 - » 分布式存储、NoSQL、RDBMS分库分表
 - 业务多、请求多、关系复杂
 - » 58列表页、详情页问题
 - » 服务化
 - » 解耦、稳定

我们的实践案例

👤 架构高可用分层案例一

— 58帮帮

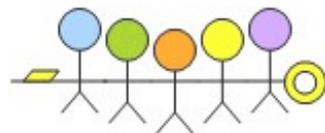
- 定位
 - 传统IM
 - 58用户与商户沟通
- 性能
 - 满足千万在线



Copyright © 2015 Bangbang.58.com

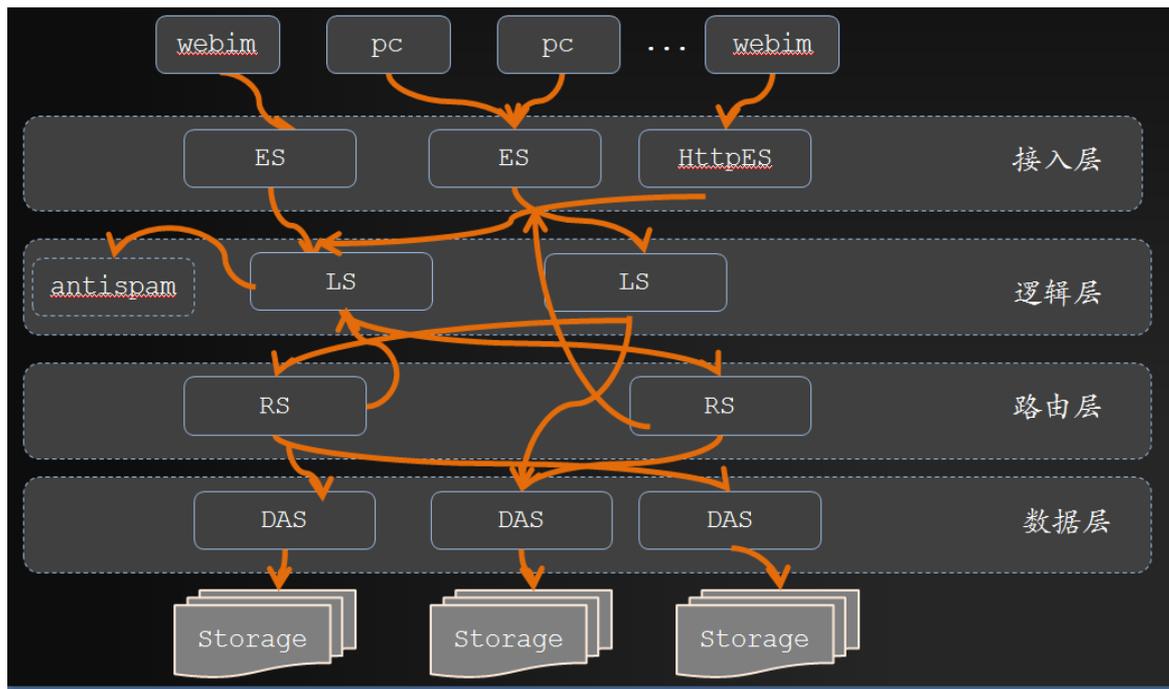


我们的实践案例



👤 架构高可用分层案例一

— 58帮帮架构





我们的实践案例

架构高可用分层案例二

项目背景

- 移动环境下消息到达机制
 - 文本、图片、音频、视频等
- 移动互联网最基础的需求
- 移动环境网络不稳定
- Apps消息无法触达
 - Msg(转转Apps交易消息等)
 - Operation(转转Apps运营活动)
 - Alert(转转红包未消费提醒)

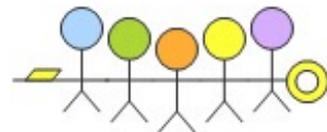




我们的实践案例

架构高可用分层案例二

- 定位
 - 58同城统一的移动PUSH推送平台
- 吞吐量
 - 10W+QPS
- 推送量
 - 高峰每天10亿+推送量
- 众多Apps接入
 - 58同城
 - 转转
 -

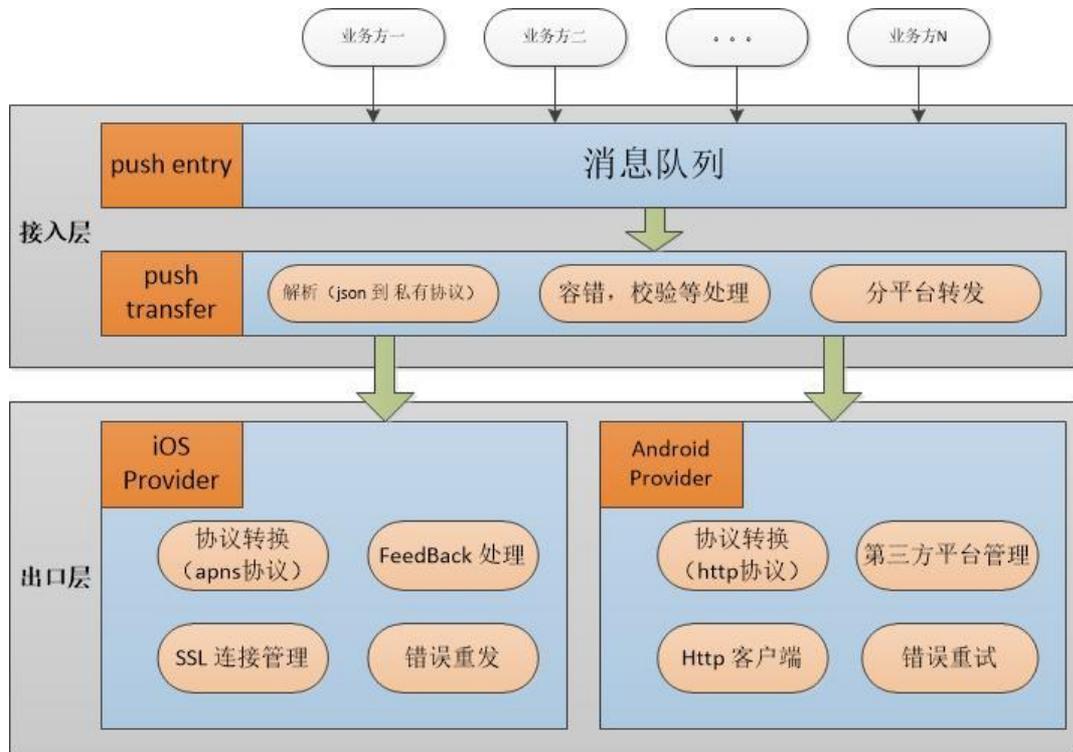


我们的实践案例

架构高可用分层案例二

58移动PUSH系统架构

- 分层架构体系
- push entry
 - 入口
 - 消息队列
- push transfer
 - 解析/处理/转发
- Android Provider
 - 出口
 - 第三方
- iOS Provider
 - 出口
 - APNS



我们的实践案例

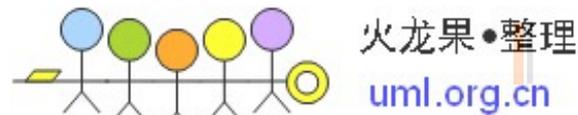
架构高可用分层案例三

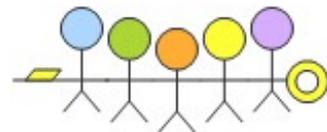
定位

- 全国最大的个人闲置交易平台

功能

- 用户
- 商品
- 社交
- 交易
- 圈子
- 推荐
- 搜索
- 运营
-



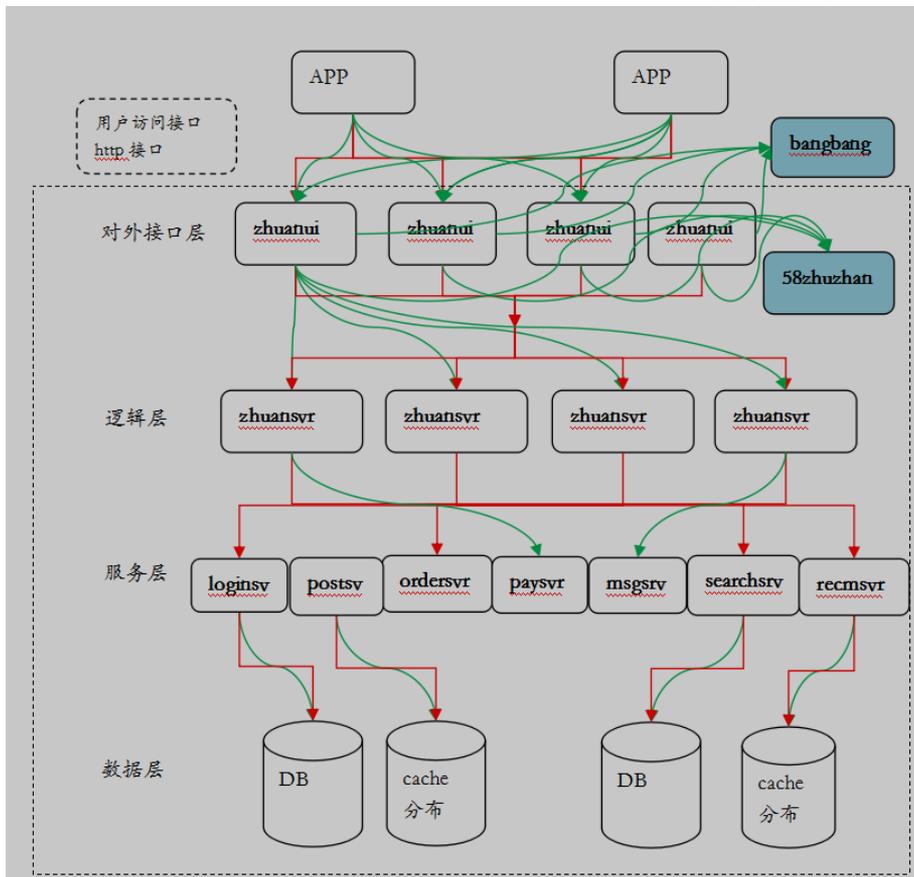


我们的实践案例

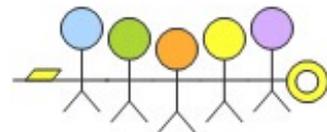
👤 架构高可用分层案例三

👤 架构如何设计

- 功能多
- 业务复杂
- 未来扩展
- 高可用性
-

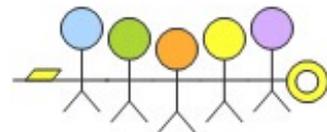


问题讨论



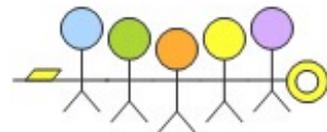
火龙果·整理
uml.org.cn

👤 架构高可用分层带来的潜在问题是什么？



本课总结

- 👤 高可用架构为什么需要分层?
- 👤 高可用架构分层设计原则是什么?
- 👤 高可用架构如何分层?
- 👤 高可用架构分层最佳实践
- 👤 我们的实践案例



火龙果·整理
uml.org.cn



THANK YOU