



🕒 2016年8月26日-27日

📍 北京珠三角JW万豪酒店

World Of Tech 2016

移动互联网技术峰会

THE BEST MOBILE TECH IN HERE

主办：51CTO

如何打造移动环境下满足业务场景的高可用架构?

孙玄@58

关于我

- 👤 58同城高级系统架构师
- 👤 58集团技术委员会主席
- 👤 公司技术委员架构组主任
- 👤 即时通讯、转转架构算法部负责人
- 👤 代表公司对外分享
 - 👤 QCon、SDCC、DTCC、Top100、Strata + Hadoop World、WOT等
- 👤 《程序员》杂志撰稿2次
- 👤 前百度高级工程师
- 👤 毕业于浙江大学

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)
- Strata + Hadoop World
- 《程序员》撰稿2次

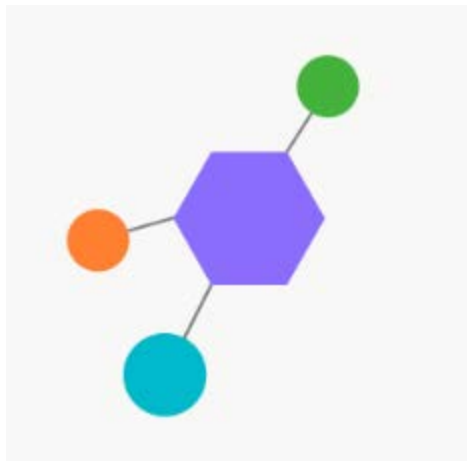


OutLine

- 👤 系统架构高可用概念
- 👤 系统架构高可用重要性
- 👤 系统架构高可用涉及环节
- 👤 系统架构高可用手段
- 👤 系统架构高可用如何评价
- 👤 移动环境如何打造满足业务场景高可用架构
- 👤 我们的实践案例

系统架构概念

- 系统架构
 - 类似于建筑物架构
 - 遵循一定设计原则
 - 系统各个部分组合，形成架构
 - 系统架构两大要素
 - 各个组件
 - 组件间的关联



系统架构高可用性

系统架构高可用性

— 系统运行依赖的基础

- 硬件
 - 机器
 - CPU
 - 内存
 - 硬盘
 - 网卡
- 软件
 - 操作系统
 - 文件系统
 - 软件本身

系统架构高可用性

系统架构高可用性

— 系统运行依赖的基础的不可用性

- 硬件

- 生命周期

- 硬件故障

- 软件

- 软件BUGS

- 软件间资源的争抢

-

系统架构高可用性

系统架构高可用性

- 任何时间都正常提供读写服务
 - 7*24
 - 机器硬件故障
 - CPU、内存、硬盘、网卡
 - 机器软件故障
 - OS、FileSystem……
 - 网络划分
 - 机器间网络隔离
 - 网络弱可用

系统架构高可用重要性

系统架构高可用重要性

- 业务快速发展的支撑
- 系统不可用[弱可用]
 - 用户体验差，用户留不住
 - 精力放在系统稳定性方面
 - 无精力响应业务功能需求
- 系统不可用，公司品牌、形象受影响
 - 2011年4月，亚马逊云计算服务EC2宕机半天
 - 贵公司有不可用的案例吗？！

系统架构高可用重要性

系统架构高可用重要性

- 系统不可用，公司利益
 - 2010年1月，百度DNS域名被劫持
 - 经济损失X万刀
 - 宕机时间==损失收入
- 没有系统高可用性，谈其他一切都是耍流氓!!
 - 架构高可用重要性

系统架构高可用涉及环节

- 涉及环节
 - 软件
 - DNS
 - CDN
 - LB
 - 接入层
 - 逻辑层
 - 数据层
 - 存储系统
 - 硬件
 - 机器
 - 内存、CPU、网卡、硬盘等

打造系统架构高可用手段

架构高可用手段

— 思路

- 分层解耦
- 每层高可用
 - » 分而治之

— 方案

- 冗余
- 系统无状态化
- 超时机制
- 幂等设计
- 分级管理
- 异步调用
- 服务降级

打造系统架构高可用手段

👤 架构高可用手段

— 服务治理

- 监控
 - 进程
 - 语义
 - 错误
 - Coredump
 - 数据波动
 -
- 服务可管理、可视化
 - 日志监控系统
 -

—



系统架构高可用评价维度

架构高可用评价维度

- 系统可用性以系统停服务时间和正常服务时间比例来衡量
- 系统不可用时间(故障时间=故障修复时间点-故障发现时间点)
- 4个9的可用性 (99.99%)
 - 一年停机的时间不能超过53分钟
 - $365 * 24 * 60 / 10000 = 53$ 分钟

架构高可用评价维度

架构高可用评价维度

- 2个9的可用性 (99%)
 - 基本可用
 - 一年停机的时间不能超过88个小时
 - $365 * 24 * 60 / 100 = 88$ 个小时
- 3个99的可用性 (99.9%)
 - 较高可用性
 - 一年停机的时间不能超过9个小时
 - $365 * 24 * 60 / 1000 = 9$ 个小时

架构高可用评价维度

架构高可用评价维度

- 4个9的可用性 (99.99%)
 - 具备自动恢复能力的高可用
 - 一年停机的时间不能超过53分钟
 - $365 * 24 * 60 / 10000 = 53$ 分钟
- 5个99的可用性 (99.999%)
 - 极高可用性
 - 一年停机的时间不能超过5分钟
 - $365 * 24 * 60 / 100000 = 5$ 分钟

架构高可用评价维度

架构高可用评价维度

— 网站可用性

- 较多的大网站可用性不足2个9
 - 88小时
- 国外、国内

— 目标

- 做到4个9
- 具备自动恢复能力高可用

移动环境如何打造满足业务场景高可用架构

移动环境特点

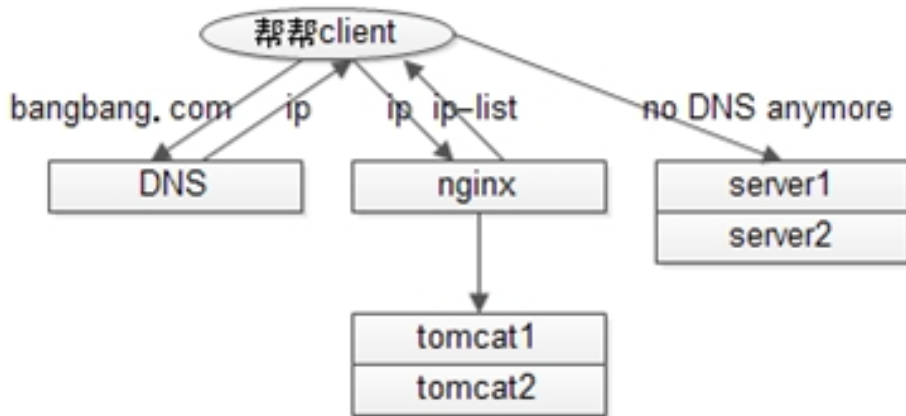
- 网络不稳定
- 流量消耗
- 电量资源保证
- 安全
-

移动环境如何打造满足业务场景高可用架构

👤 网络不稳定

- DNS优化

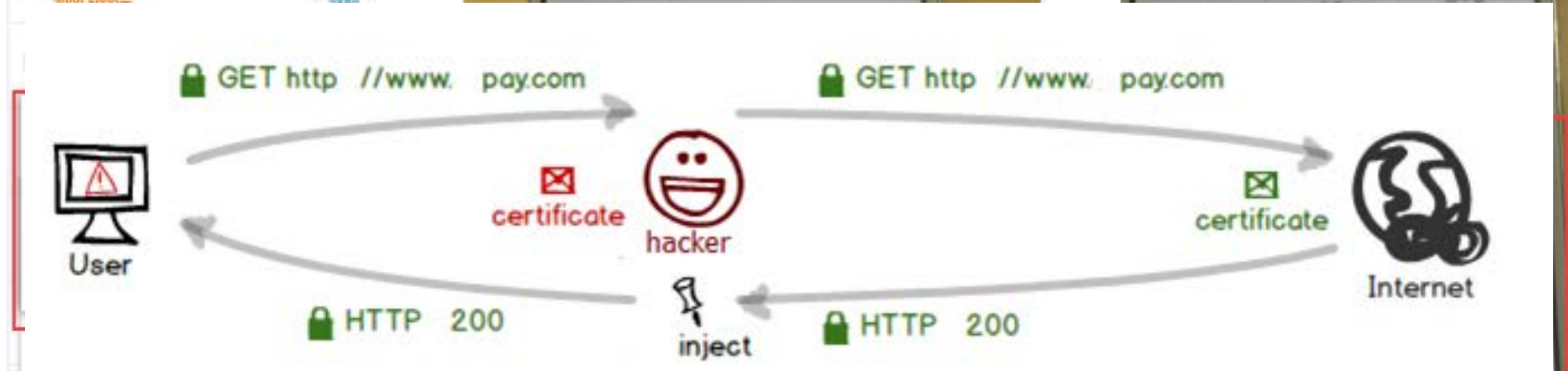
- App第一次访问，先拉取server ip-list保存到App本地；
- 未来访问时，App直接使用ip-list中的IP来访问server，不再需要DNS；
- 如果DNS劫持或者可用，直接使用ip-list中的IP访问，达到柔性可用；



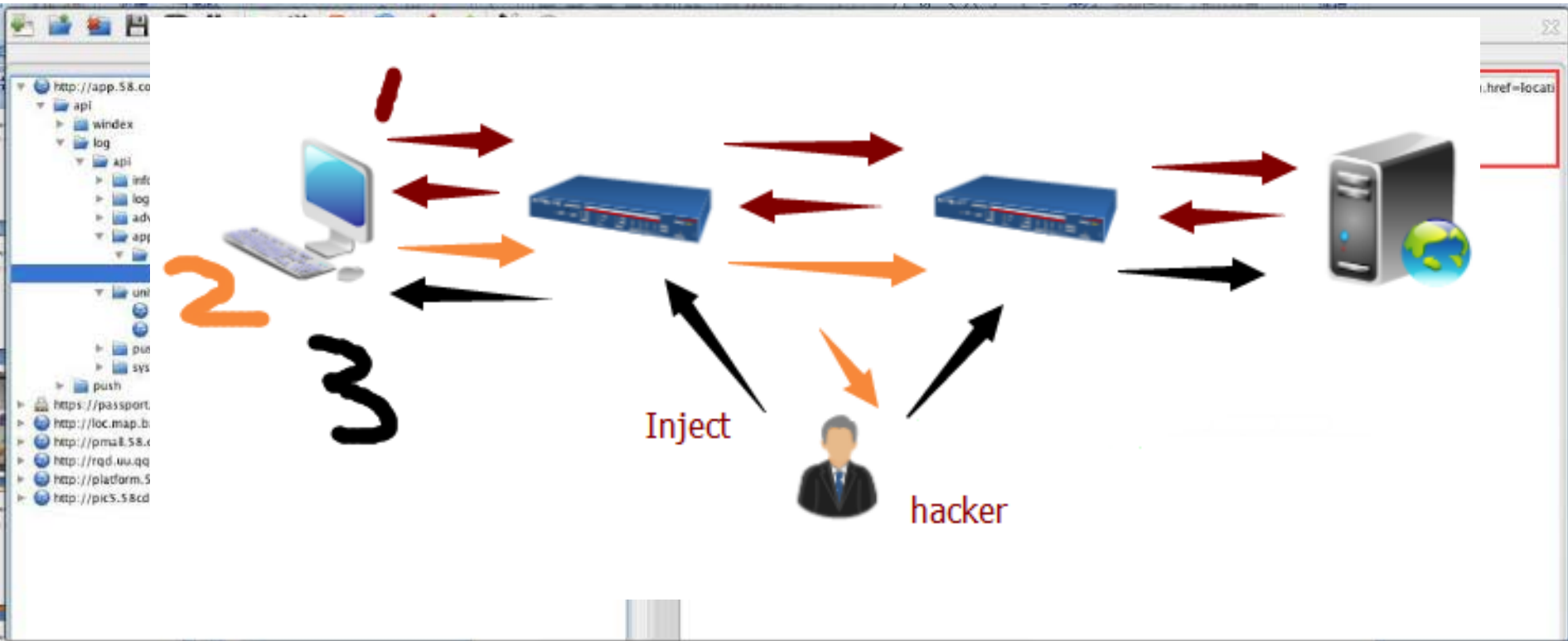
移动环境如何打造满足业务场景高可用架构

流量劫持

— 实例一



移动环境如何打造满足业务场景高可用架构



移动环境如何打造满足业务场景高可用架构

流量劫持

— 解决方案

- 传输数据明文
- 传输数据密文
 - HTTPS
 - » 全站HTTPS化
 - SSL
 - TLS

移动环境如何打造满足业务场景高可用架构

网络不稳定

- APP登录优化

• APP端快速重连

- 第一次走全部流程

» 加密

- 置后台

- 快速重连

• 解决

- 减少连接步骤

- 提高连接速度

- 较少流量

- 提升连接成功率

移动环境如何打造满足业务场景高可用架构

网络不稳定

– push推送优化

• 移动特点

- TCP长连接+push推送
- 优先TCP长连接
- 离线push推送

• 解决

- 弱网环境下消息到达的问题
- 通道优先级解决消息实时性
- 提升消息到达率
- 增加日活、转化等

移动环境如何打造满足业务场景高可用架构

流量、电量优化柔性可用

– 数据拉取优化

• 典型业务场景

- 移动IM好友信息
- 变动频率不高
- 每次从移动后台拉取，流量、电量浪费
- 有必要做到针对性拉取
- 怎么办？

移动环境如何打造满足业务场景高可用架构

流量、电量优化柔性可用

— 数据拉取优化

- 如何解决

- 时间戳机制【版本号】

- » 每次有数据更新移动后台更新时间戳
 - » Apps拉取时间戳并存储本地
 - » 每次上线从移动后台拉取时间戳，并比对
 - » 如果没变化，不需要拉取实际数据
 - » 有变化，针对性的拉取一次数据

移动环境如何打造满足业务场景高可用架构

流量、电量优化

— 实时&延迟拉取优化

- 数据优先级不同

- 优先级高的数据
- 一般优先级的数据
- 优先级低的数据

- 解决

- 实时要求高的数据
 - » 实时拉取
- 实时要求不高数据延时拉取
 - » APP判断网络环境
 - » WIFI环境拉取

移动环境如何打造满足业务场景高可用架构

流量、电量优化

— 协议优化

- 典型业务场景

- 埋点、日志上报
- 请求数据

- 如何解决

- 协议字段尽可能短小
- 批量合并请求
 - » HTTP头占用比重大
 - » 埋点、日志上报等
- 数据压缩
 - » 请求数据->tomcat->gzip

我们的实践案例

58帮帮

— 定位

- 传统IM
- 满足58用户与商户沟通

— 核心功能

- 用户关系
- 添加好友
- 发送消息



我们的实践案例

58帮帮

— 模块

- 30+ / JAVA / CPP

— 请求

- 10亿(IM) + 30亿(!IM)
- 同时在线用户数突破100w+

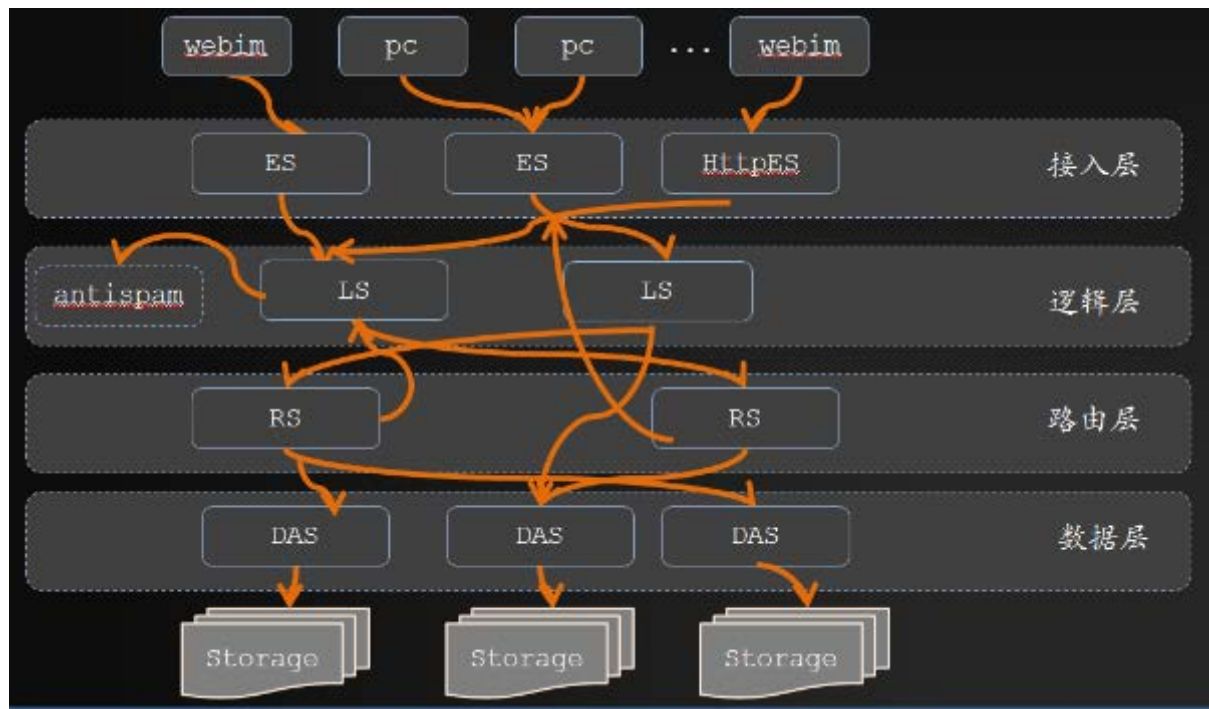
— 机器

- 60+



我们的实践案例

58帮帮



我们的实践案例

58帮帮如何满足千万同时在线?

- 接入层、逻辑层、路由层、数据层
- 无状态设计
- 每层模块动态高扩展
- 模块冗余，高可用性保证
- 动态负载均衡，动态切换可用服务节点
- 异步化
- 优化效果
 - 单机线上支持50W+同时在线
 - 单机线上3w+qps

我们的实践案例

转转

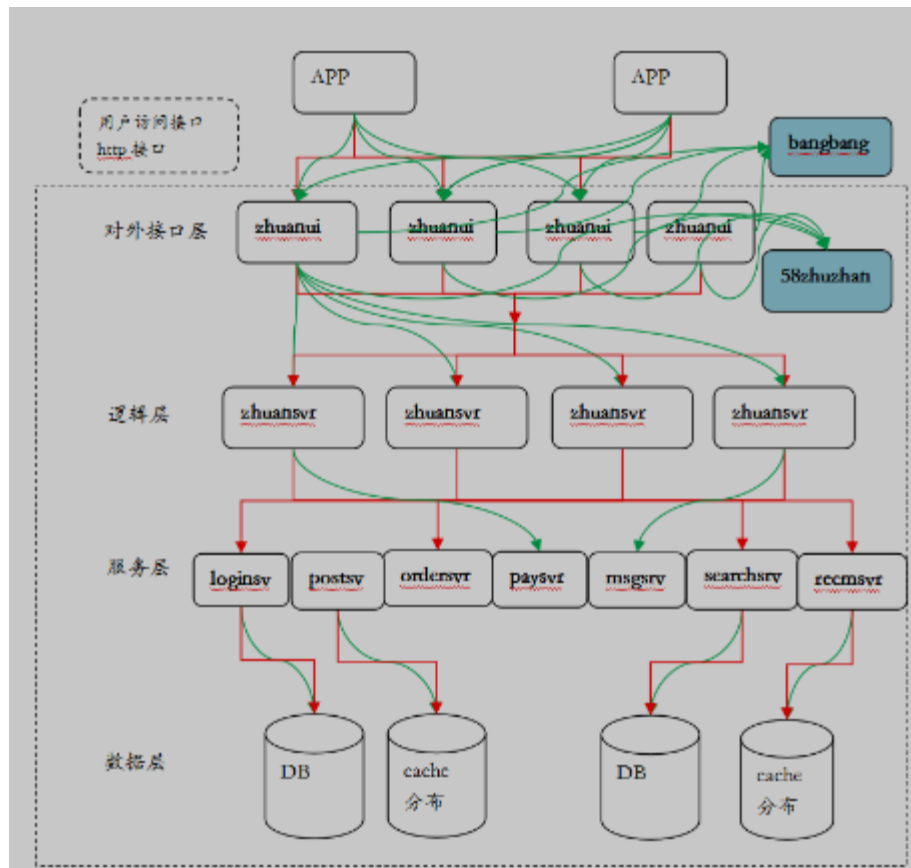
- 转转定位
 - 最终专业的二手交易平台
- 功能
 - 用户体系
 - 商品/交易
 - 社交
 - 留言、评论、私信、好友关系
 - 推荐
 - 搜索
 - 运营



我们的实践案例

🐼 转转架构设计

- 功能多/业务复杂
- 交易高安全
- 未来扩展性
- 高可用性
- 低耦合分层架构
 - 对外接口层
 - 逻辑层
 - 原子服务层
 - 数据层
-



我们的实践案例

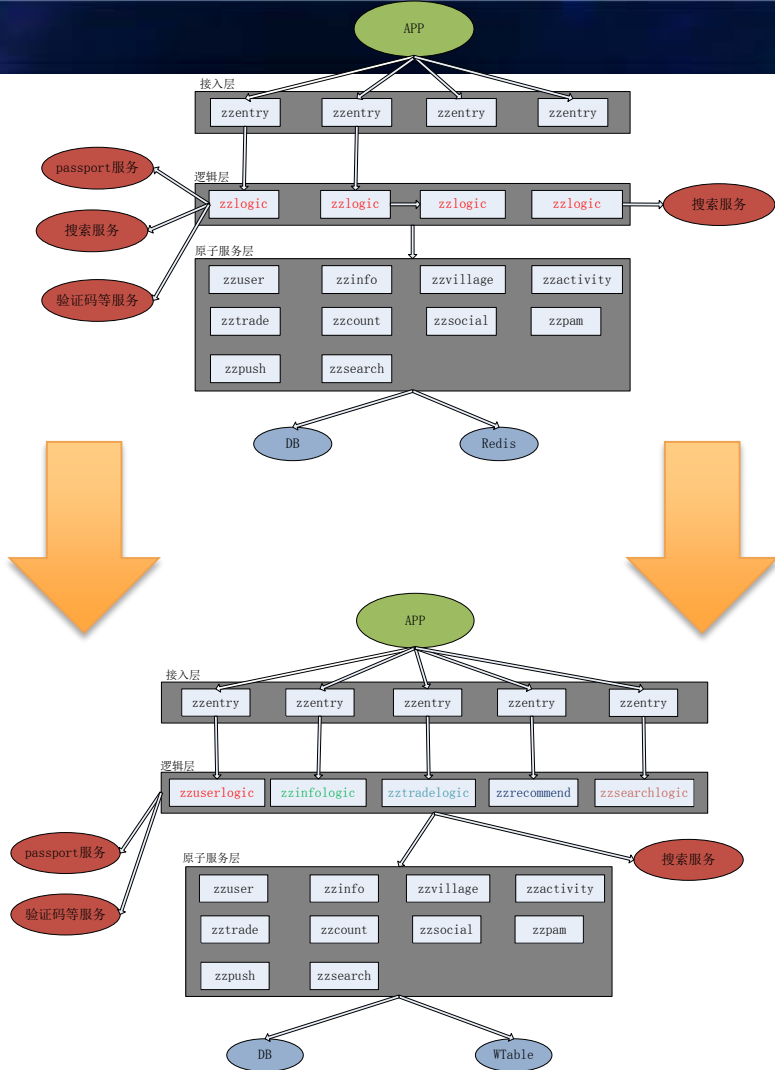
转转架构演进

— 物理垂直划分

- 每个业务一个独立的进程

— 优点

- 业务间完全解耦
- 业务间互不影响
- 业务模块独立
- 单独开发、上线、运维
- 效率高



演讲总结

- 👤 系统架构高可用概念
- 👤 系统架构高可用重要性
- 👤 系统架构高可用涉及环节
- 👤 系统架构高可用手段
- 👤 系统架构高可用如何评价
- 👤 移动环境如何打造满足业务场景高可用架构
- 👤 我们的实践案例

转转-更专业的二手交易平台
搜索、推荐等方向急需精英
欢迎自荐&推荐
sunxuan@58.com



Q&A

THANK YOU

欢迎关注“架构之美”
感谢！

