

Yunba.io云巴

ArchSummit 2014 实时系统架构与实践

@Tiger_张虎
zhanghu@yunba.io

关于我

- 张虎
 - 极光推送 (JPush) 创始人，前CTO
 - 云巴 (Yunba.io) 创始人，CEO
 - @Tiger_张虎

Agenda

- 为什么需要实时系统
- 实时系统的设计
- 自动化运维实践

为什么需要实时系统？

- 实时内容分发
- 实时聊天
- 实时统计

协议的选择 Why MQTT

- 移动互联网的特点
 - 网络慢、不稳定
 - 功耗低
- MQTT (Message Queuing Telemetry Transport)
 - 轻量级 pub/sub
 - 开放协议，支持广泛
 - 开源资源多
 - 扩展方便
- Why Not XMPP
 - 网络流量大，据统计 70% 消耗在 XML 标签传输
 - XML 编解码重



海量用户接入 - 负载均衡问题

- 负载均衡器 (LVS...) 的问题
 - 单点失效
 - 单点性能瓶颈
- 负载均衡从客户端开始做起
- 域名负载的问题
 - 域名系统不可靠
 - 更新延迟大

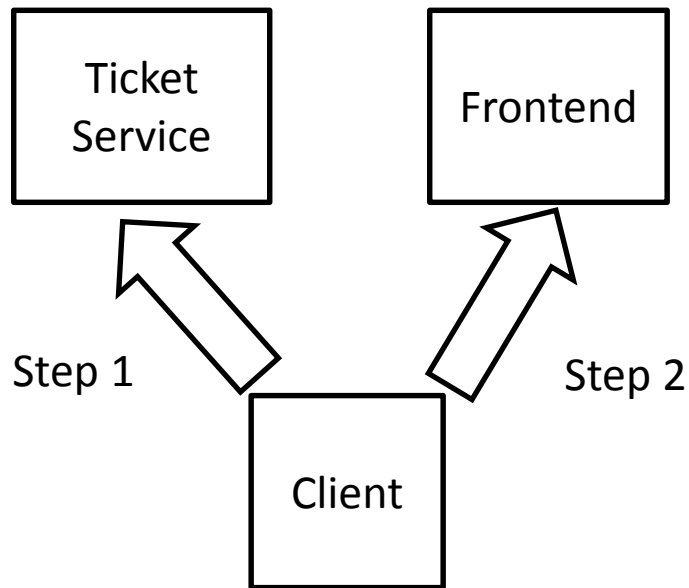


海量用户接入 - 负载均衡问题

- 静态分片 ($Uid \% N$)
 - 单点失效
 - 后期维护成本高

海量用户接入 - 负载均衡方案

- Ticket Service , 动态分配接入点



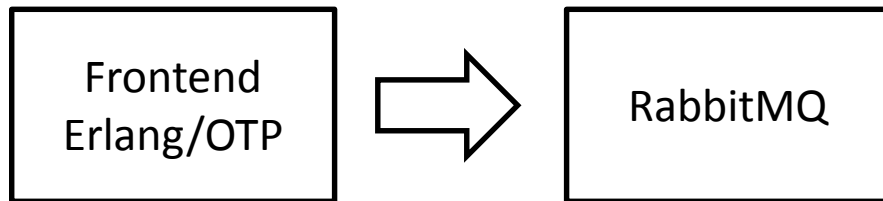
海量用户接入 - 负载均衡方案

- DNS 服务器响应慢？
- 全网 DNS 服务器失效 ！？
 - 我们的服务还可以用 ！



前端服务器设计

- 维持客户端长链接
- 同步缓存请求，异步处理请求

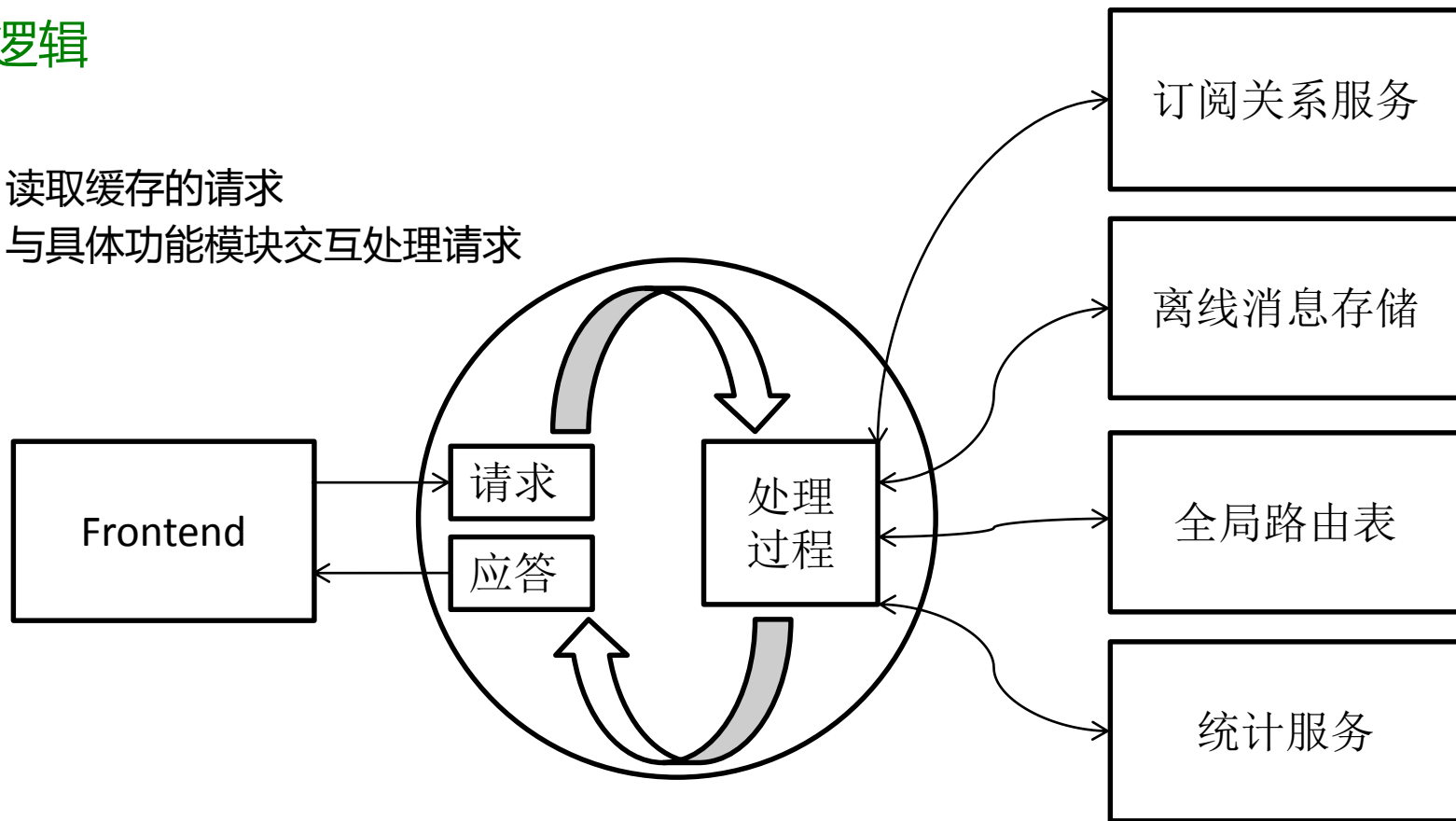


Why Erlang?

- 优点
 - Process 模型适合处理高并发
 - 管理方便: escript
 - 开发周期比用 C 短
- 缺点
 - CPU load
 - Memory footage
- 下一步
 - 回归 C ? 单机维持 350万 长链接

主逻辑

- 读取缓存的请求
- 与具体功能模块交互处理请求



主逻辑

- 与多个模块交互
 - Frontend
 - 订阅关系服务
 - 离线消息存储
 - 路由表
 - 统计服务
- 事件驱动
 - 上一个应答驱动下一个操作
- 平行扩展
 - 添加一个实例，处理能力增强



Why node.js

- 优点
 - 异步模型
 - 广泛模块支持
- 缺点
 - 动态语言，代码规模不宜过大
 - 运算密集场景效率低
- 下一步
 - Scala?
 - Go?



基础服务

- 订阅关系服务、离线消息存储、全局路由表、统计服务
- 独立 Service、集群，上下文无关
- 低延迟、高并发

高性能 Key/Value

- 基础服务大部分依赖于 key/value 存储
- 基于 Couchbase 改造
 - Auto failover
 - 动态加节点
 - 读写负载均衡

高性能 Key/Value 设计

- 负载均衡从客户端开始
 - 客户端实时监控所有服务器状态
- Auto Sharding
- 无单点失效
- 动态扩容：增加机器，扩大容量、增加吞吐
- 异步持久化



云服务、自动化运维

- 运维就是要自动化
 - 持续集成开发实践
 - 人工部署运维大规模集群不可行
- 云主机降低运维成本
 - 启动成本低
 - 增加服务器快速
- 集群扩容过程
 - 自动申请VM
 - 自动配置DNS
 - 自动部署模块、监控
 - 自动测试

Ansible

- 不需要安装 agent
 - 只依赖 SSH
- 安装脚本模块化
 - 方便累积、分享

GitHub

We've found **3,885** repository results

Ansible

Inventory

[webservers] ←
www1.example.com
www2.example.com

Playbook

→ - hosts: webservers
roles:
- common
- webapp

Roles

roles/common/tasks/main.yml
roles/webapp/tasks/main.yml

Ansible 实践

- 百台服务器规模的集群，几十分钟部署完成
- 新增节点零配置
- 集群配置修改，避免遗漏

Thanks!

