

外卖的背后-饿了么基础架构从0到1的演进

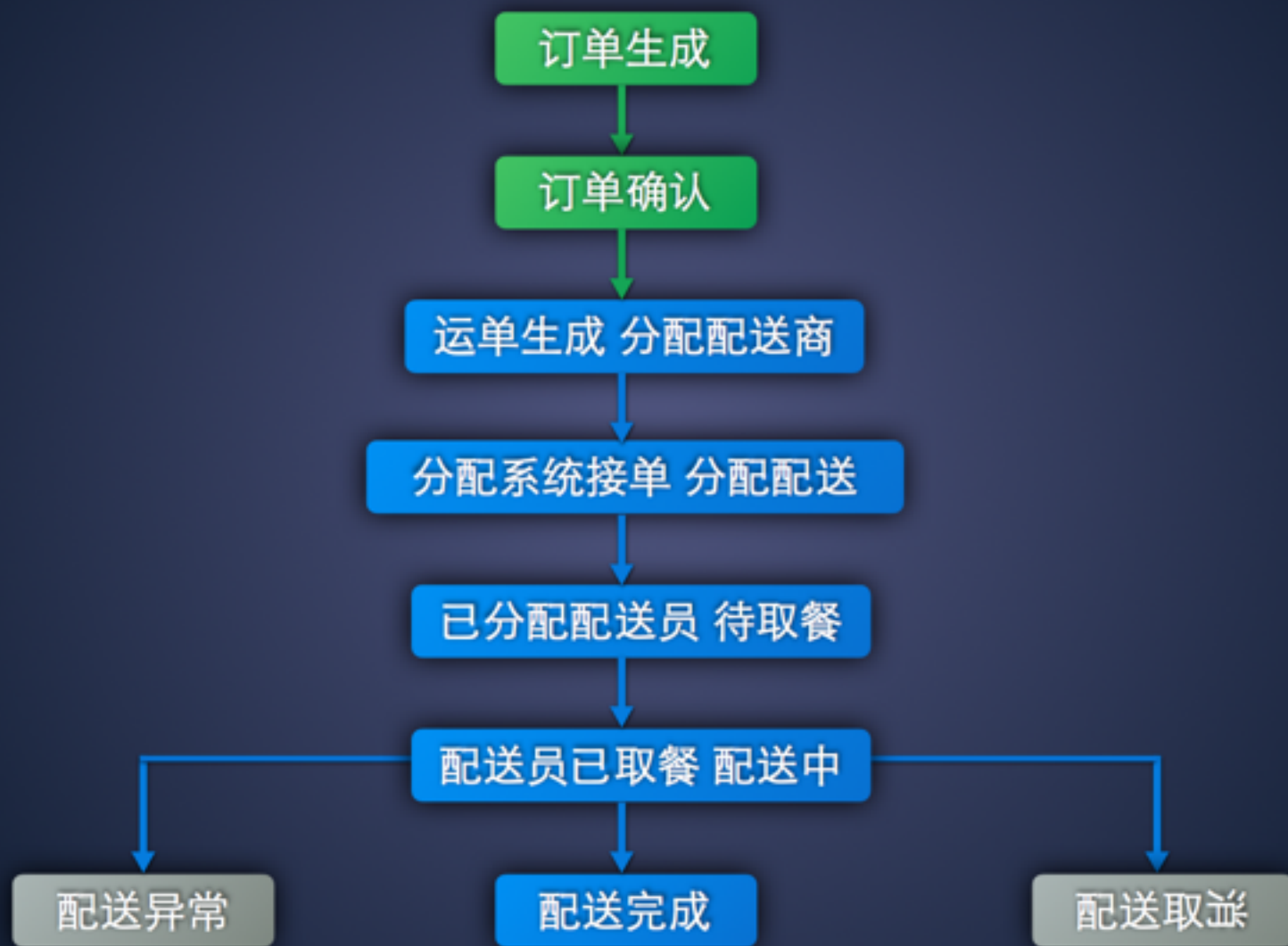
兰建刚

饿了么 框架&工具

美好生诞触手可得

一份互联网“ 外卖” 到底有多复杂

饿了么技术挑战最大领域：订单至运单



介绍

- 2015年初组建框架&工具团队
- 从0到1的演进 – 话题太大
- 方向决定成败，细节关乎痛苦
- 讲点细节的东西

APIGateway		CI/CD
ServiceAPI		Config
Services		Trace
SOA Framework		Log
DAL	Corvus	Metrics
DB	Cache	Alert
DRC		Job
Python, Java, Go		Message
Container		SD

饿了么



大纲

负载均衡

无损升级

基础组件设计

负载均衡

最初的方案

- F5
- HAProxy
 - 部署在客户端本地
 - 不需要考虑HAProxy的高可用
 - 流量不是问题

负载均衡

遇到的问题

- 部署量变大
 - 痛点：扩容需要更改所有客户端的HAProxy配置，维护客户列表超级复杂
 - 后果：运维拒绝再部署HAProxy
- 配置不统一
 - 目录不一致，文件格式不统一
 - 批量修改容易出问题
 - 案例：DAL采用N+1的容灾策略，由HAProxy控制切换，完成一次切换演练需要1小时 +

负载均衡

探索解决之道

- 方案一：减小部署量
 - 集中式的HAProxy
 - F5
 - OSPF + LVS
- 方案二：减少运维成本
 - 服务发现 – 只需要配置服务名和集群名
 - 自动配置下发 – 服务上下线自动同步至订阅方
 - HAProxy不支持

负载均衡

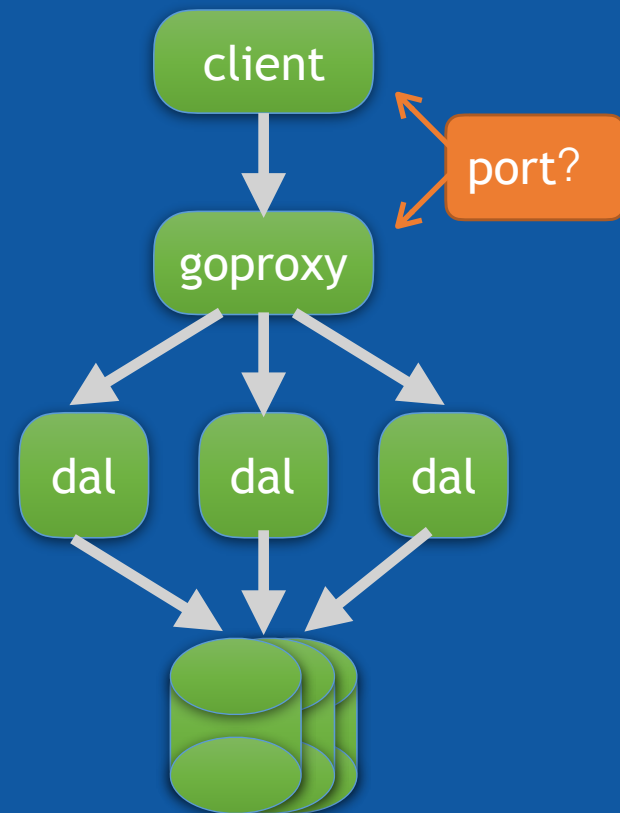
RPC解决方案 – 内置LB SDK

- 服务自注册/自发现
- 配置少 – 只需要服务名 + 集群名
- 部署简单 – 随应用一起部署
- RoundRobin – 简单可用

负载均衡

Redis/DB解决方案 – GoProxy

- DAL – DB代理, sharding, 读写分离
- Corvus – Redis代理, redis cluster, 协议转换
- DAL/Corvus作为服务自注册
- GoProxy
 - 基于Go语言
 - 订阅DAL/Corvus注册事件
 - 带服务发现的HAProxy: 配置少
 - 业务方改造成本低



负载均衡

Tips

- 长连接 vs 短连接
 - 把长连接当短连接用
 - 以节点而不是连接做负载均衡
- 健康检测
 - 客户端与服务端心跳检测
 - 可扩展健康语义：进程在、端口活着不代表服务“可用”
 - “半死不活”比“死透了”伤害更大

大纲

负载均衡

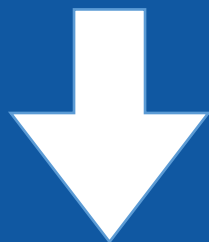
无损升级

基础组件设计

无损发布

最初的状态

- 开发都有生产权限
- 随时随地发布
- 没有考虑无损发布这回事



诉求：“一单都不能丢”



无损发布

经典解决方案

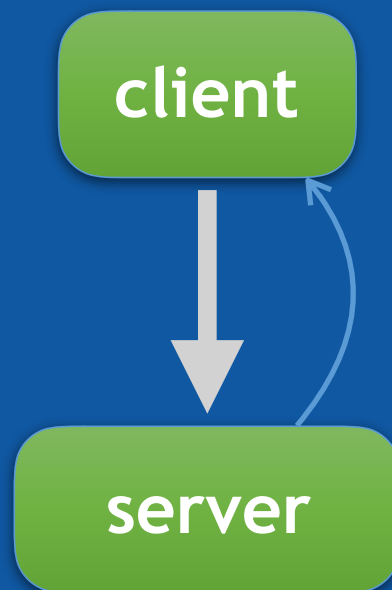
1. 服务端准备下线
2. 通知客户端
3. 客户端停止向服务端发送新请求
4. 服务端等待正在处理的请求结束
5. 服务端下线



无损发布

经典解决方案

1. 服务端准备下线
2. (通知) (所有) 客户端
3. (客户端停止) 向服务端发送新请求
4. 服务端等待正在处理的 (请求结束)
5. 服务端下线



无损发布

RPC调用

- 服务发现机制
 - 注销时所有客户端都会得到通知
- 客户端与服务端直连
 - 客户端从可用列表里剔除准备下线的服务
- 绝大多数的RPC的响应时间在秒级以下
 - sleep 2s搞定

无损发布

数据库访问

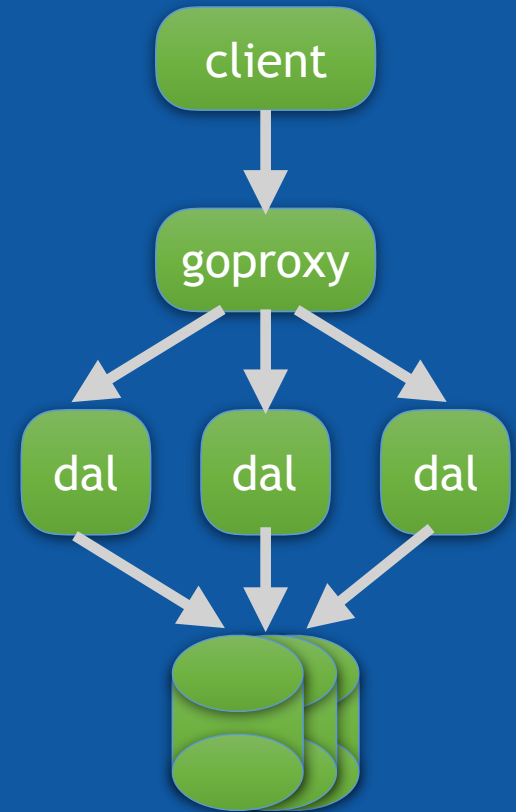
- 客户端使用jdbc/sqlalchemy通过GoProxy访问DAL
- 没有通知机制



- goproxy支持mysql协议
- 支持dal侧连接重连



- 客户端改造：限制连接存活时间



无损发布

Tips

- 服务心态：能用技术解决的尽量用技术解决
- 业务方的改造能节省中间件的很大精力

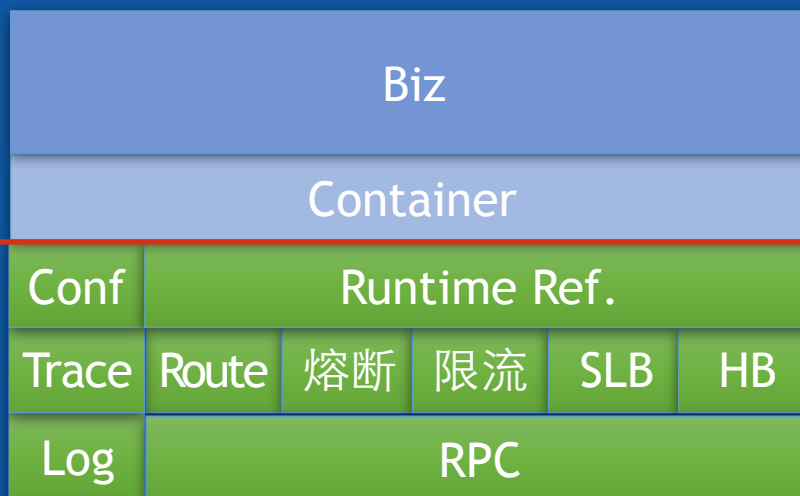
大纲

负载均衡

无损升级

基础框架设计

开放式架构 – 基于组件



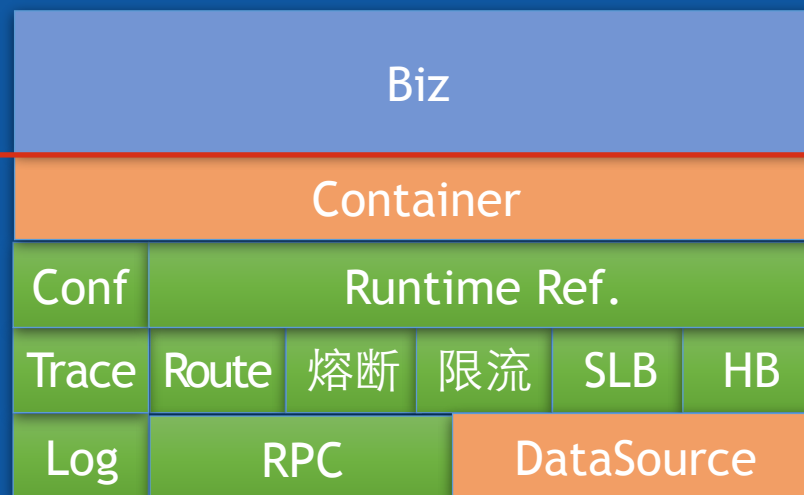
给业务开发团队
最大限度的自由

咸吃萝卜

淡操心

开放式架构 – 基于运行时

封闭

给业务开发团队
最大限度的自由

开放封闭原则

《敏捷软件开发-原则、模式与实践》

开放—封闭原则（OCP）

软件实体（类、模块、函数等等）应该是可以扩展的，但是不可修改的。

开放封闭原则

《饿了么基础框架实践》

基础框架应该是可以扩展的，但是不可“选择”的

THANKS