GOPS2017
Beijing

GOPS

GOPS 全球运维大会 2017 北京站

指导单位：数据中心联盟 Data Center Alliance　开放运维联盟 OOPSA Open OPS Alliance

主办单位：高效运维社区 GreatOPS Community　DevOps 时代　CIO时代

大会时间：7月28-29日

大会地点：北京朝阳悠唐皇冠假日酒店

eleme分布式服务化演进

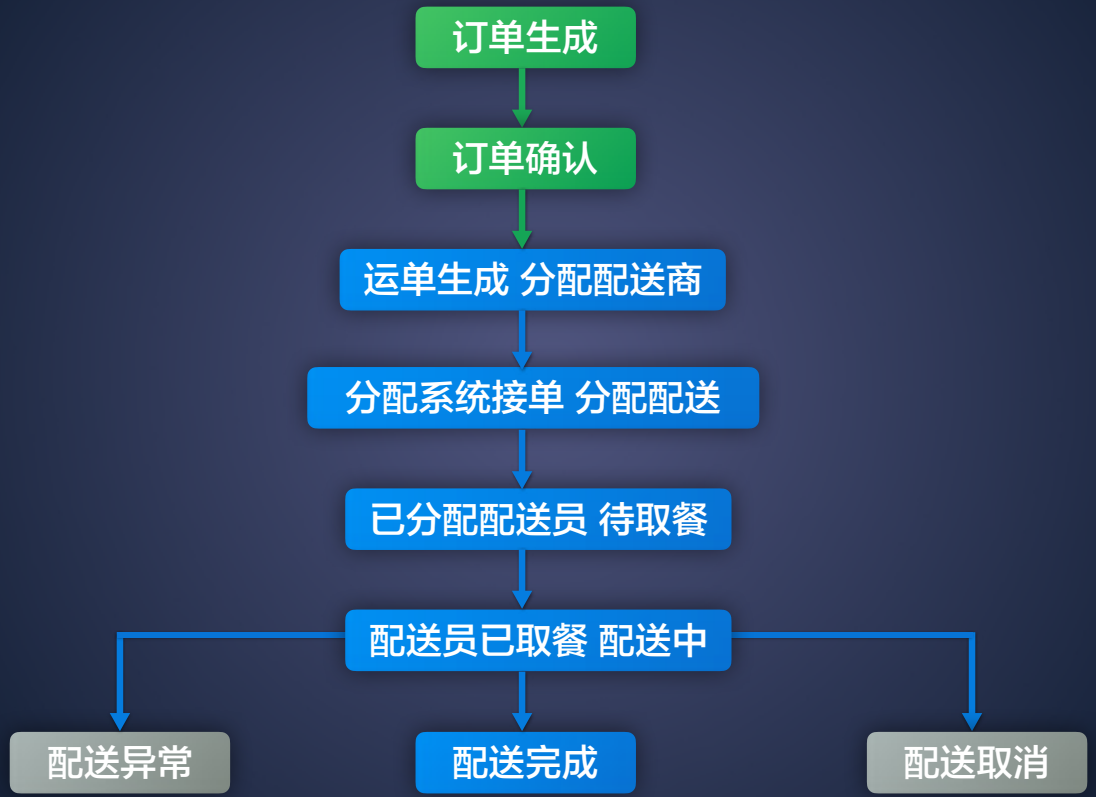
提纲

- 饿了么技术挑战最大领域：订单 ➔ 运单
  - 饿了么订单挑战
  - 饿了么运单挑战
- 外卖 vs.［衣住行］之技术挑战
- 服务化架构演进
  - 服务化架构内关系
  - 服务化架构演进阶段

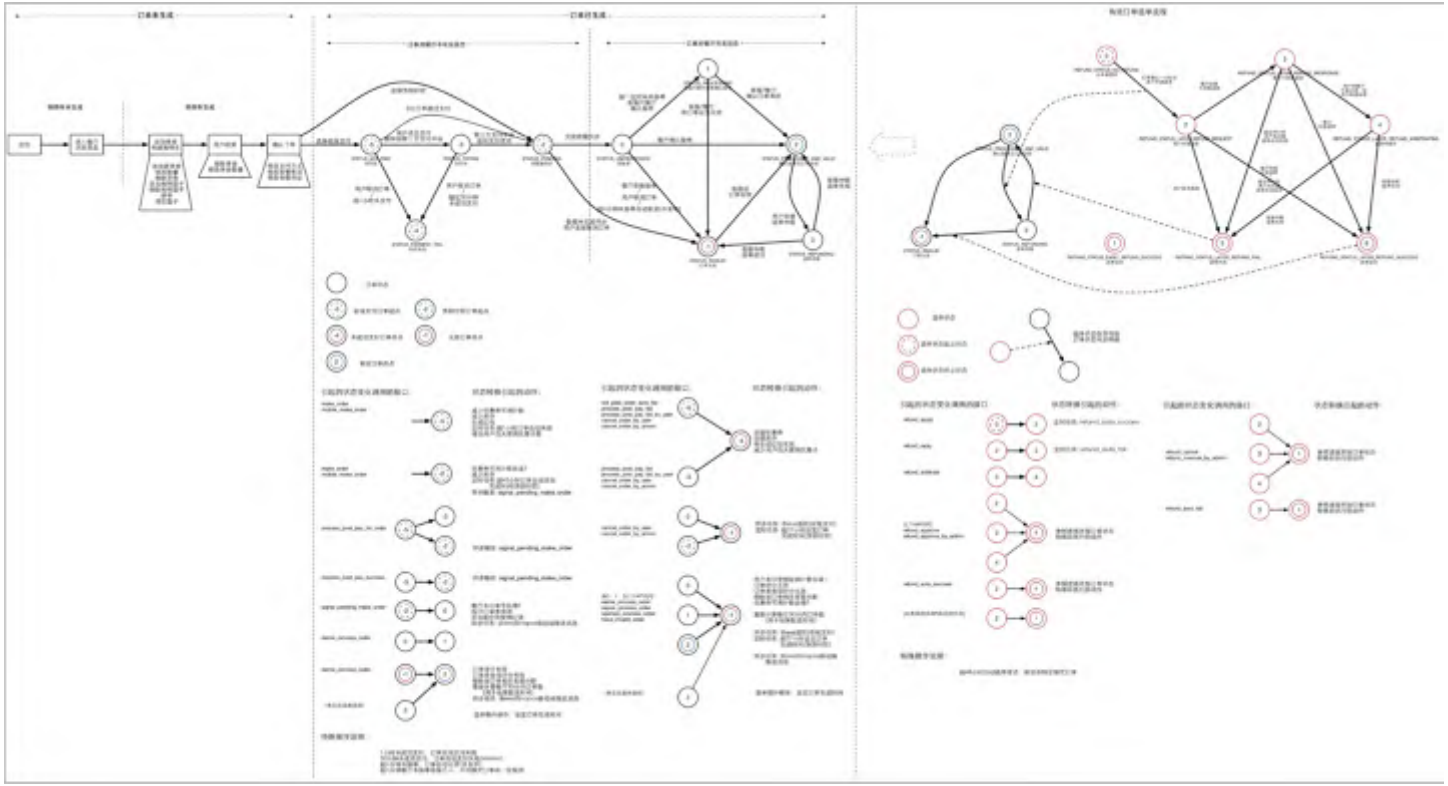
美好生活触手可得

一份互联网“外卖”到底有多复杂

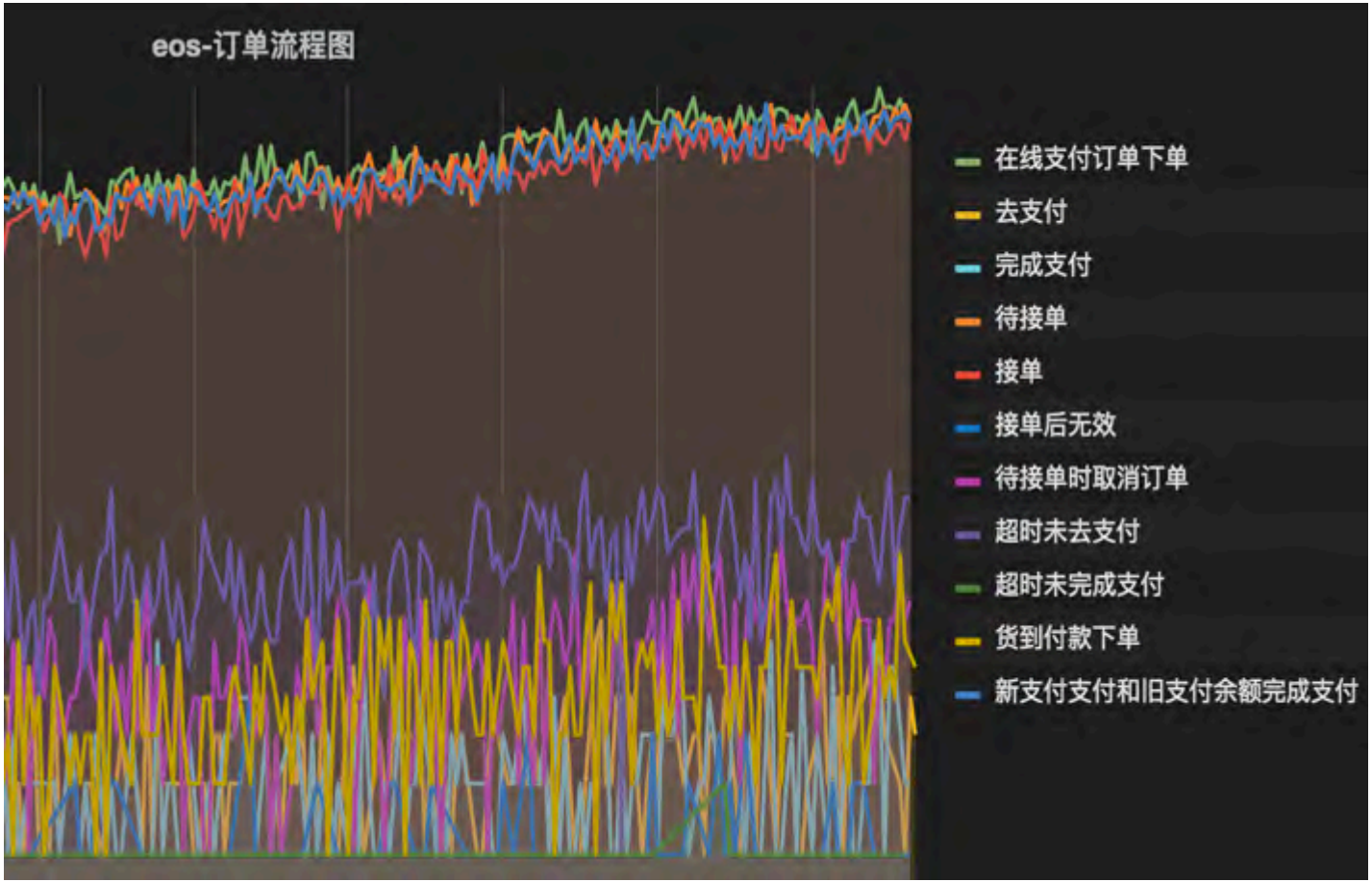
饿了么技术挑战最大领域：订单至运单



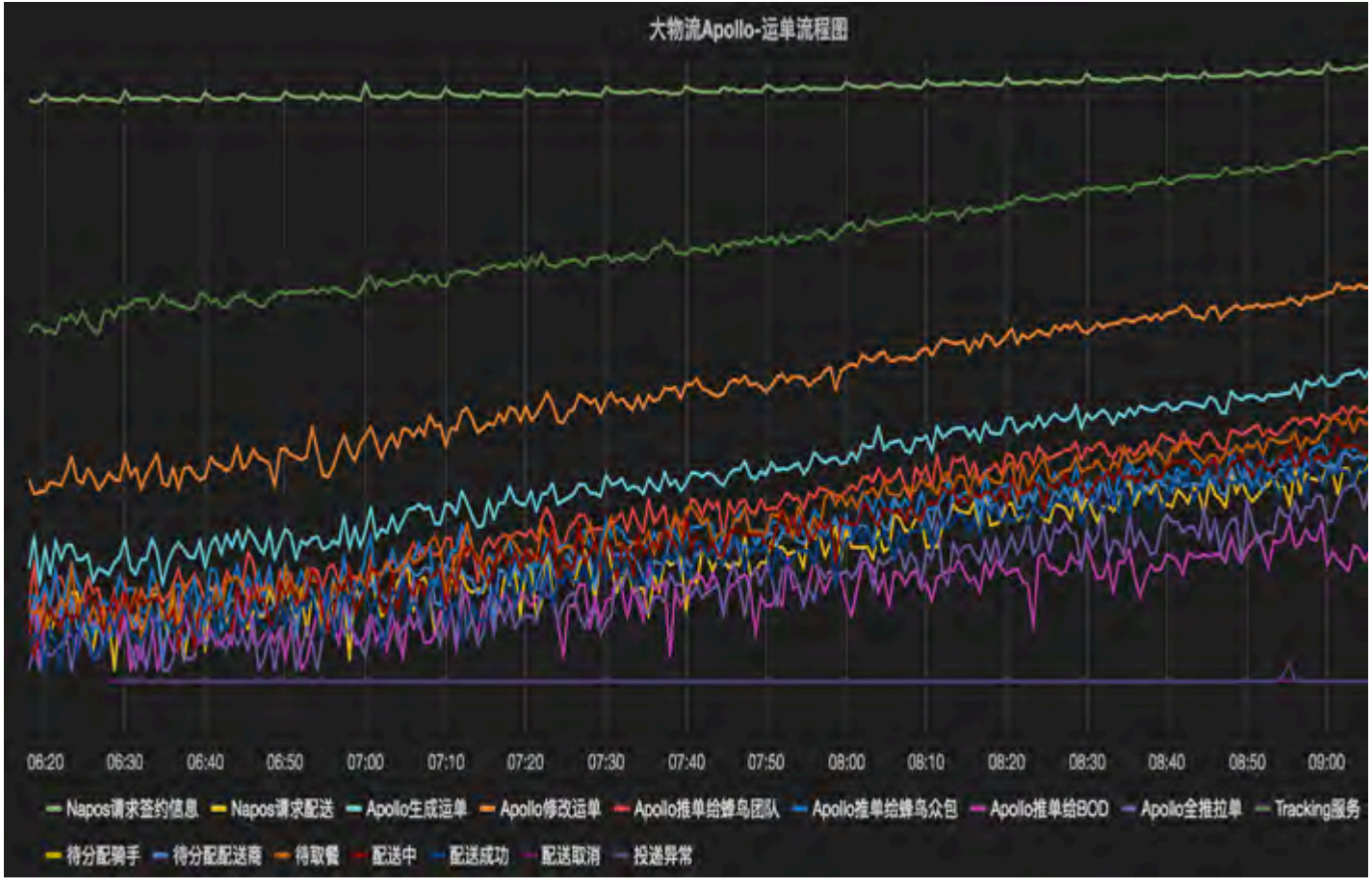
订单流程概览 – 正向 + 逆向



NOC 大屏 (2) – 订单处理


eos-订单流程图
在线支付订单下单
去支付
完成支付
待接单
接单
接单后无效
待接单时取消订单
超时未去支付
超时未完成支付
货到付款下单
新支付支付和旧支付余额完成支付

NOC 大屏 (3) – 运单处理


大物流Apollo-运单流程图
06:20
06:30
06:40
06:50
07:00
07:10
07:20
07:30
07:40
07:50
08:00
08:10
08:20
08:30
08:40
08:50
09:00
Napos请求签约信息
Napos请求配送
Apollo生成运单
Apollo修改运单
Apollo推单给蜂鸟团队
Apollo推单给蜂鸟众包
Apollo推单给BOD
Apollo全推拉单
Tracking服务
待分配骑手
待分配配送商
待取餐
配送中
配送成功
配送取消
投递异常

饿了么订单挑战

- 订单：高并发、瞬时冲击、517、秒杀

- 策略：全链路监控 *(含开放平台)*、降级、熔断、限流
- 难点：限流守门，不仅仅是技术上挑战
- 实战：分区错峰、多步延迟、攻击应对

- 延伸：实时订餐领域每天都面临两次大的瞬时冲击，再加上秒杀活动，不仅技术压力大，商户、配送、客服压力都很大。。。那么，有没有可能在产品形态或用户引导上做些创新？

饿了么运单挑战

- 运单：快、准、全

- 快：客人第一反应，肯定是快
- 准：客人长期期望，基本是准
- 全：覆盖挑战最大，模型为王

- 运单四大模型：网格、定价、调度、路径
- 运单技术挑战：机器学习、各种算法策略

一点数据

- 每日订单：8M/天，峰值订单：9M+/天
- 每秒订单：常规 500/秒，活动 15K+/秒

- 覆盖城市 1400+

- 每日配送：3.5M/天
- 峰值配送：4.5M+/天

- 平均人效：35/天
- 配送人数：自营 6K，团队 100K，注册众包 2.9M

- 逻辑数据中心：6 *(4 IDC + 阿里云 + 其他云平台)*
- 服务器节点：18K+
- SLOC：50M+ *(master/default-only)*
- App ID：1400+

衣、食、住、行四大互联网行业技术特点对比

衣

食

住

行

外卖 vs.［衣住行］之技术挑战

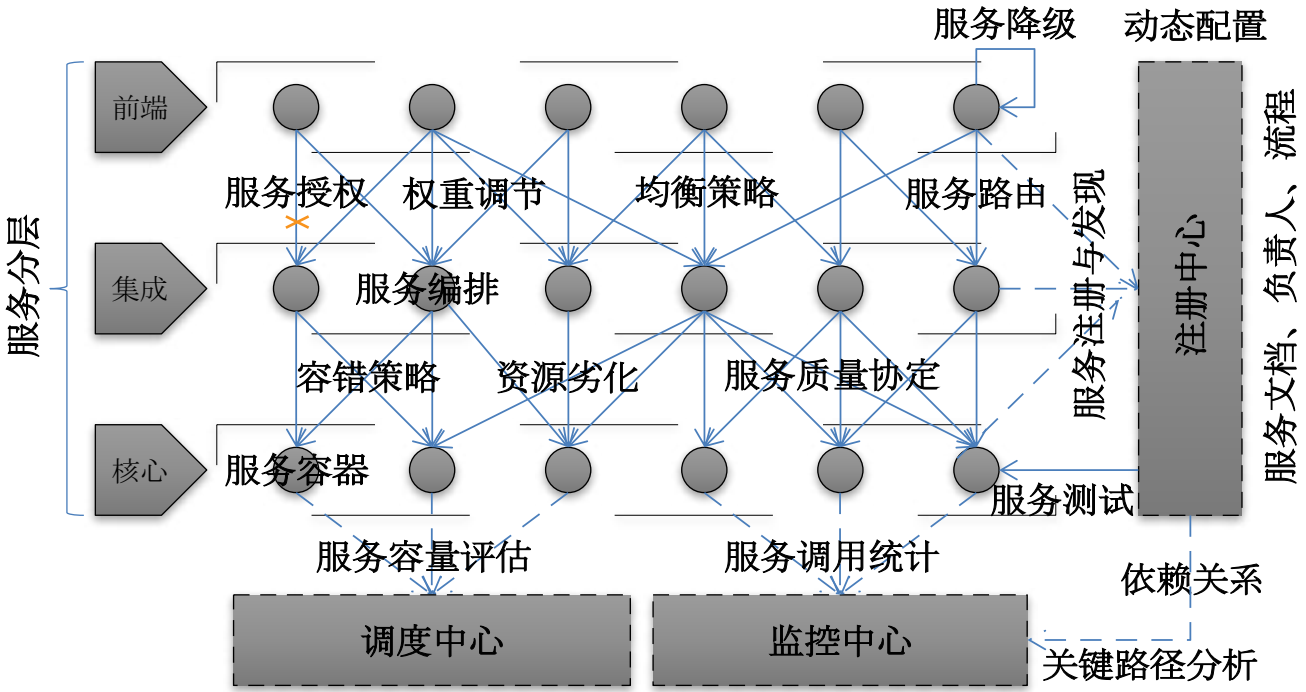
- 核心角色挑战，尤其开放平台技术挑战
  - 用户（*含企业订单/团餐/拼单/开放平台*）
  - 商户（*含直营连锁/加盟连锁/开放平台*）
  - 骑手（*含团队/众包/自营/开放平台*）

- 淘宝：用户/商户为主，[骑手]走开放平台

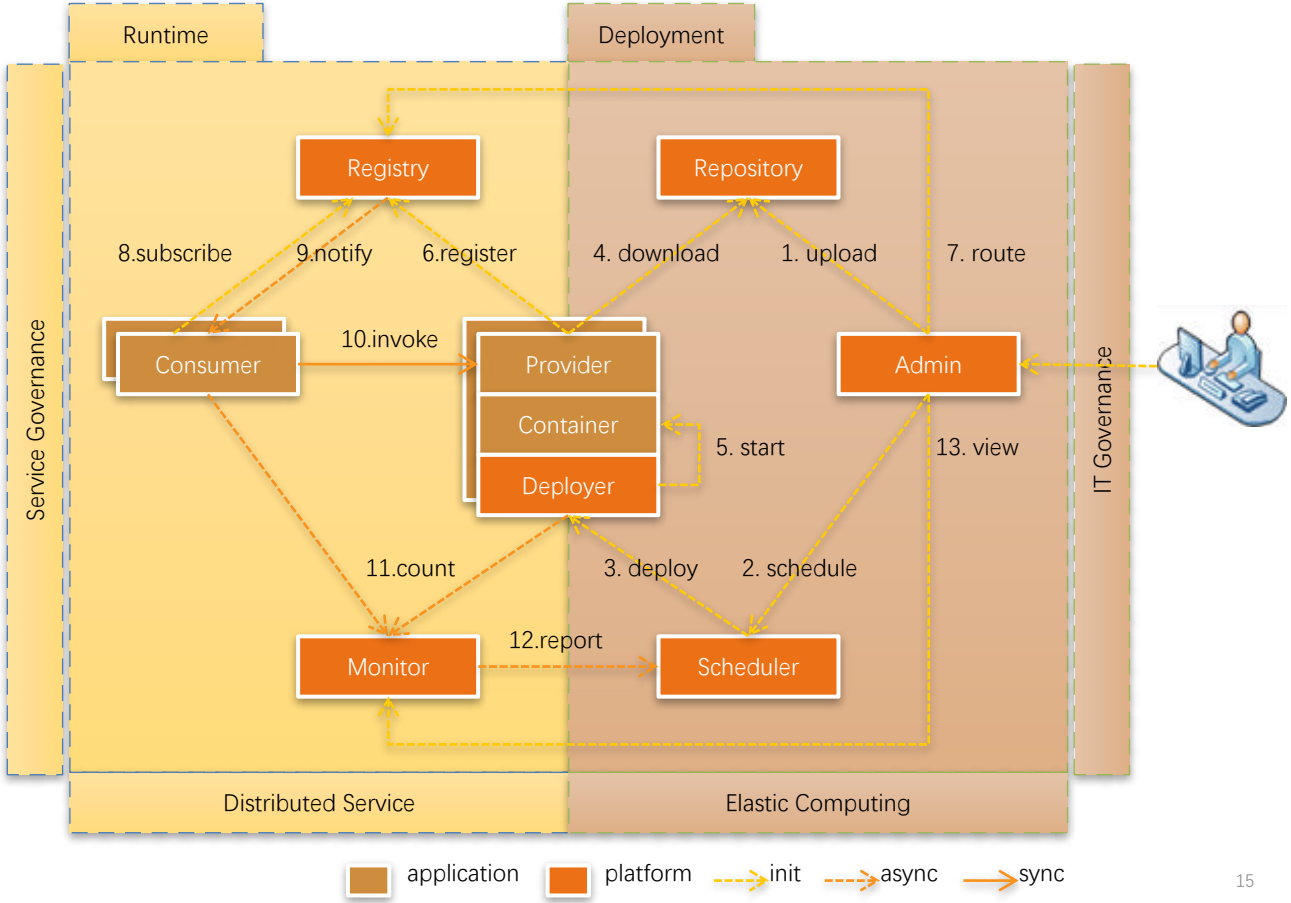
- 携程：用户/商户为主，基本没有[骑手]

- 滴滴：用户/[骑手]为主，基本没有商户

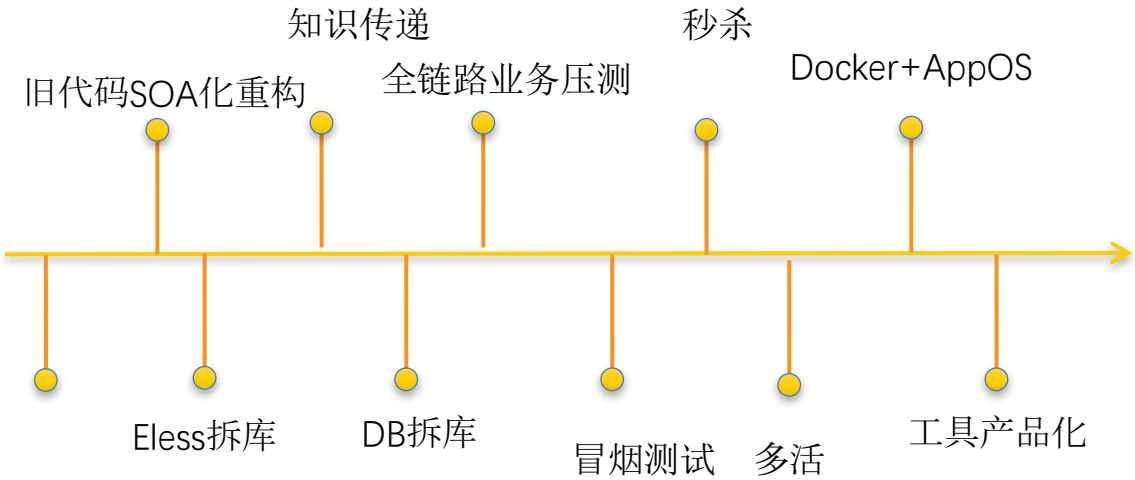
服务化架构


服务降级
动态配置
前端
集成
核心
服务分层
服务授权
权重调节
均衡策略
服务路由
服务编排
容错策略
资源劣化
服务质量协定
服务容器
服务注册与发现
注册中心
服务文档、负责人、流程
服务测试
服务容量评估
服务调用统计
依赖关系
调度中心
监控中心
关键路径分析

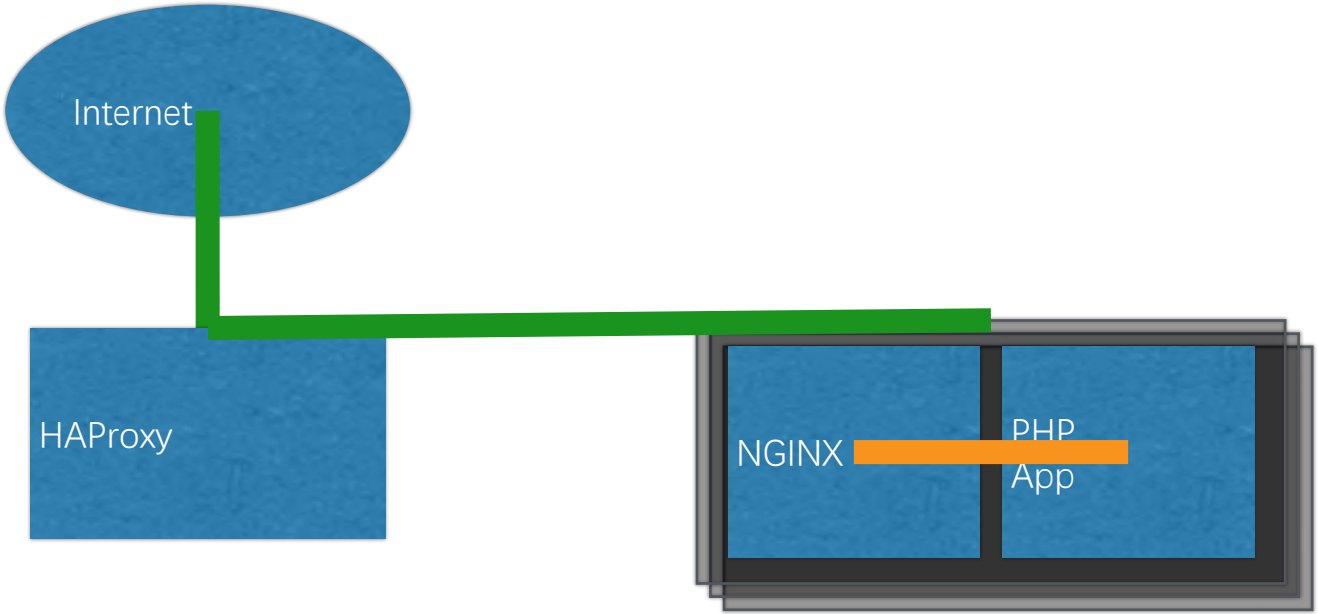
分布式服务架构内关系


Runtime
Deployment
Registry
Repository
8.subscribe
9.notify
6.register
4. download
1. upload
7. route
Service Governance
Consumer
10.invoke
Provider
Container
Deployer
Admin
IT Governance
5. start
13. view
11.count
3. deploy
2. schedule
12.report
Monitor
Scheduler
Distributed Service
Elastic Computing
application
platform
init
async
sync

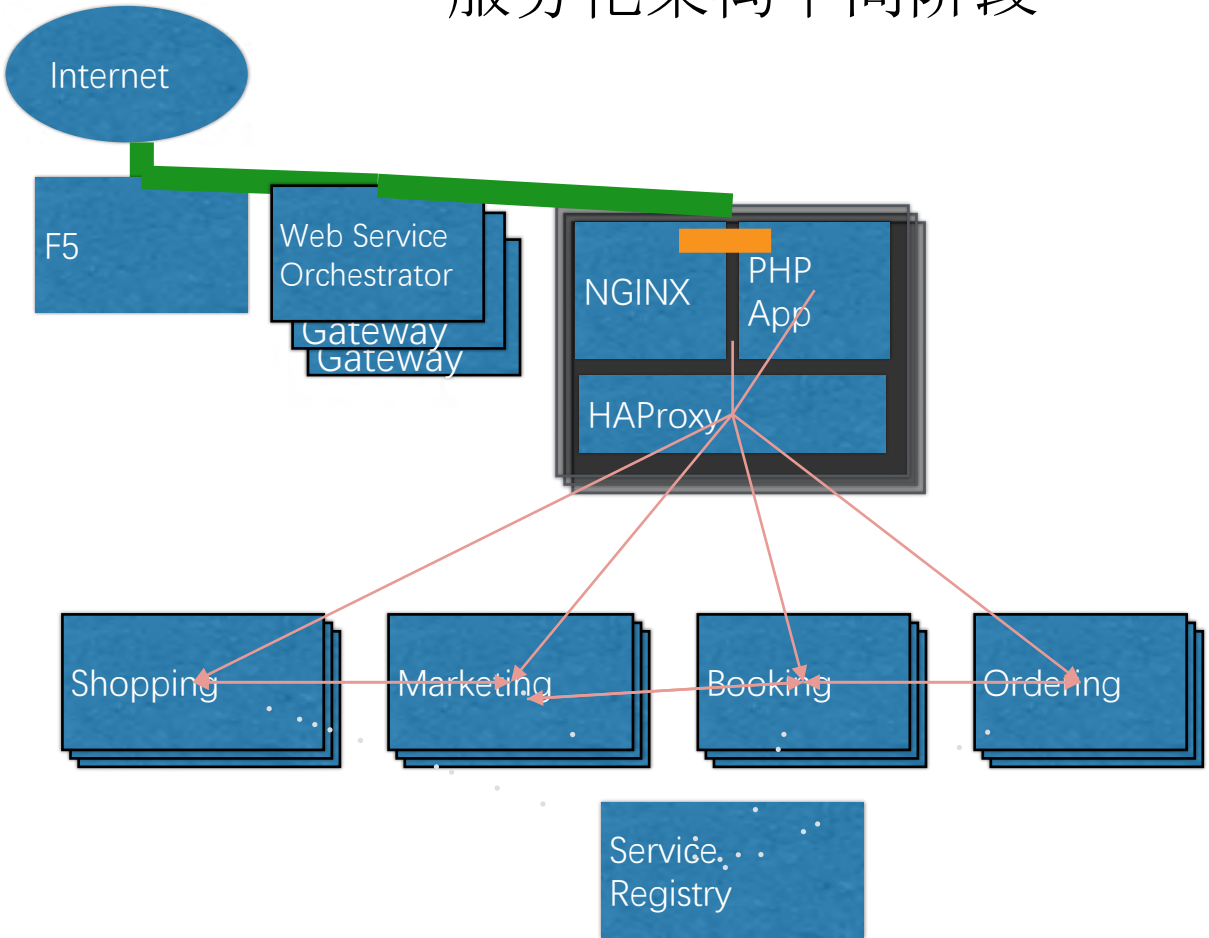
Eleme服务化架构演进阶段



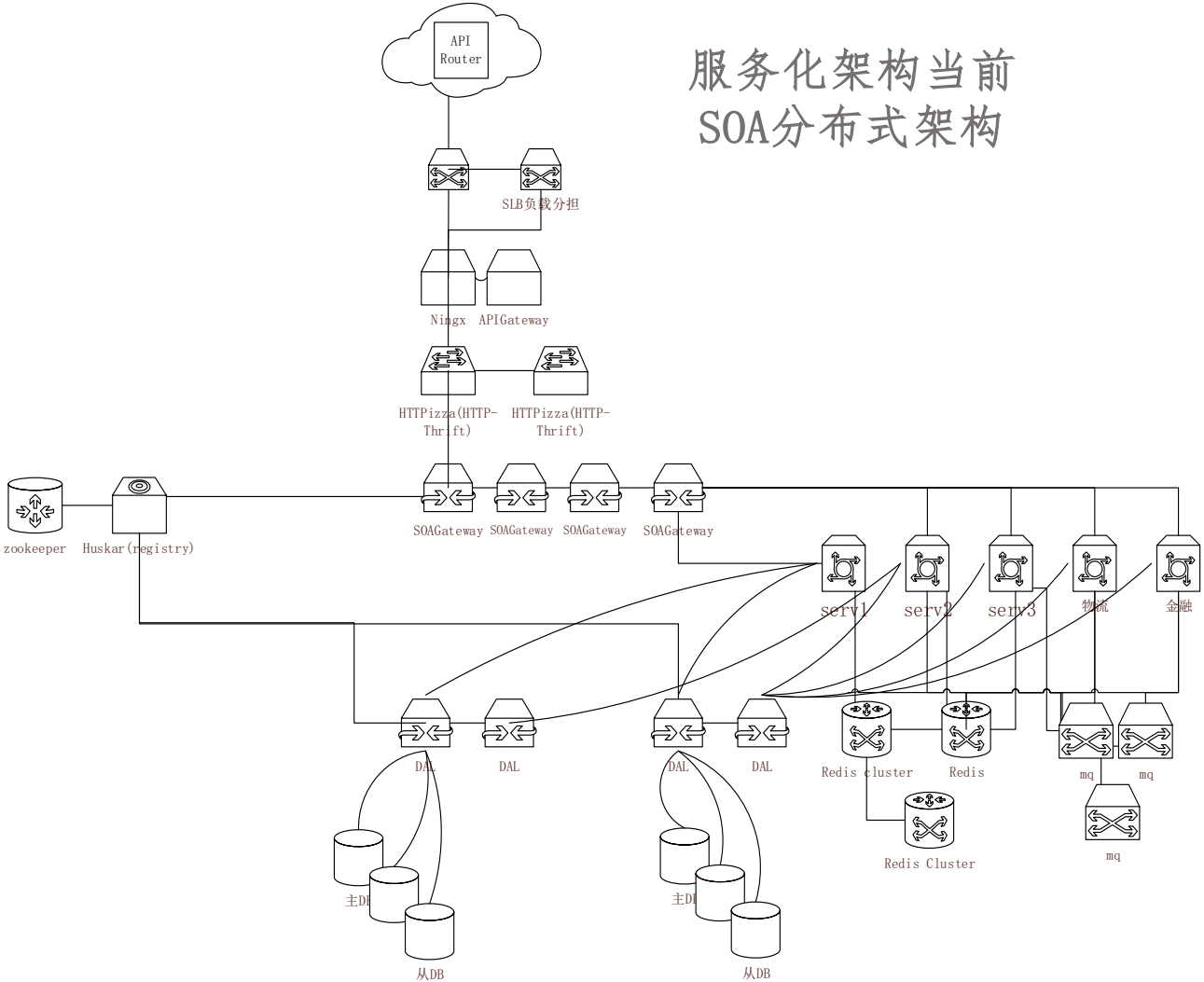
服务化架构前



服务化架构中间阶段


Internet
F5
Web Service
Orchestrator
Gateway
Gateway
NGINX
PHP
App
HAProxy
Shopping
Marketing
Booking
Ordering
Service
Registry

服务化架构当前
SOA分布式架构



GoProxy(SOA Gateway)

- 引擎功能：分布式系统的配置管理、服务发现、断路器、智能路由、微代理、控制总线等
- 分布式服务：
  - 模块间访问采用Thrift RPC的方式，开发者不用关注网络与报文格式，像写单机程序一样开发分布式软件服务。
- **服务高可用性：**
  - **负载自动均衡、熔断与容错，对于单机故障、服务过载、局部网络波动等状况自动应对（限流），服务有高可用性。**
- 支持多种语言：
  - 支持Python/Go/C++/Java/PHP语言，编程语言中立，兼具开发和运行效率。
- Web管理界面：
  - Web化的管理界面，在web界面完成配置、发布、监控、日志、降级等所有操作。
- Docker镜像安装（开发中）：
  - 需要复杂部署的服务器都采用docker镜像的方式安装，使得部署非常容易。

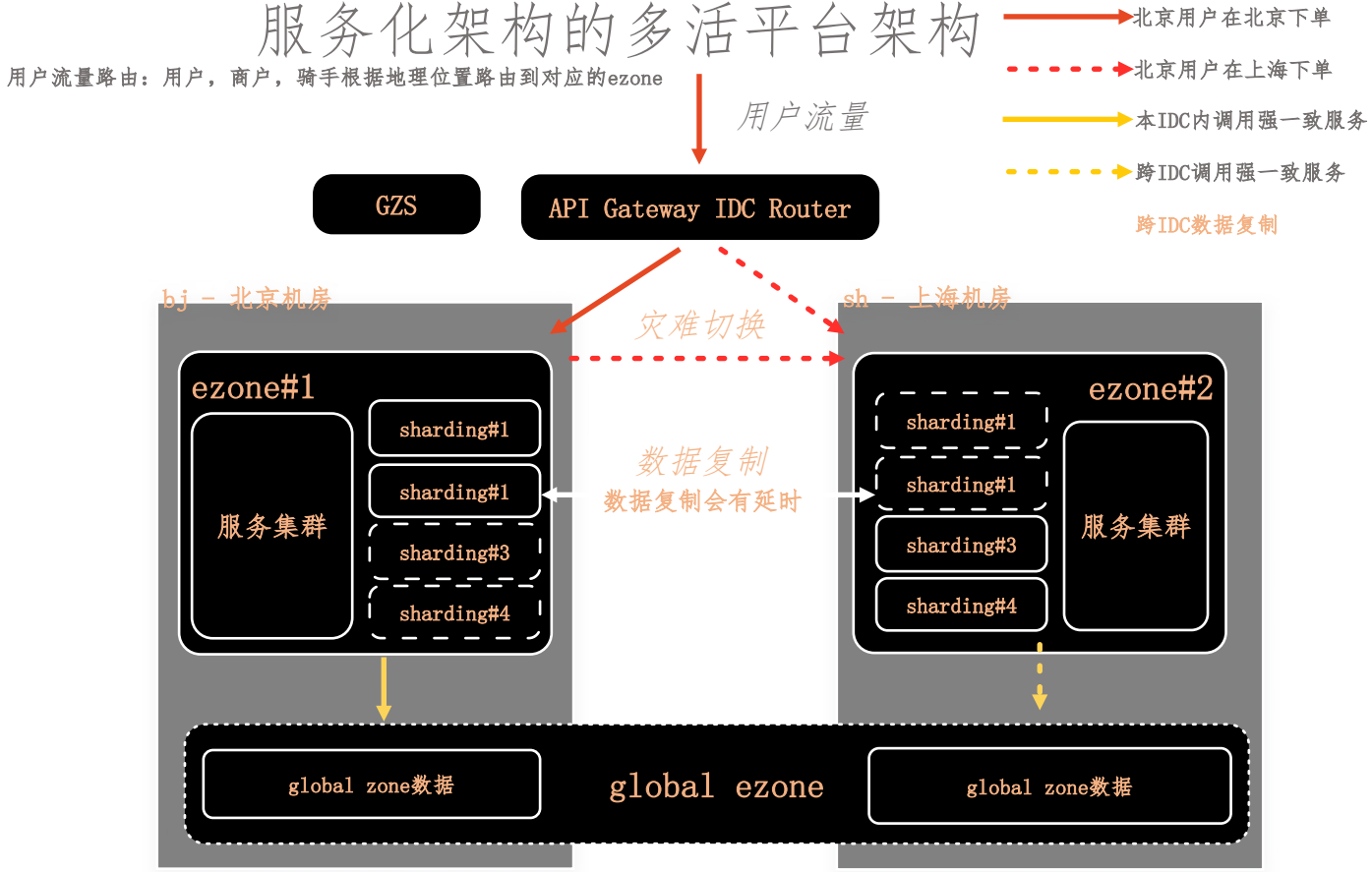
SOA架构/微服务治理实践

- 目的：SOA架构/Microservice Architecture，就是基于微服务的架构。简单来说， 微服务的目的是有效的拆分应用，实现敏捷开发和部署。

- 业务BU领域划分
- 按照业务而不是技术来划分组织
- 一系列的独立的服务共同组成系统
- 单独部署，跑在自己的进程里
- 做有生命的服务而不是项目
- Smart endpoints and dumb pipes（强内聚弱耦合）
- 自动化运维（DevOps）
- 容错

服务化架构的多活平台架构

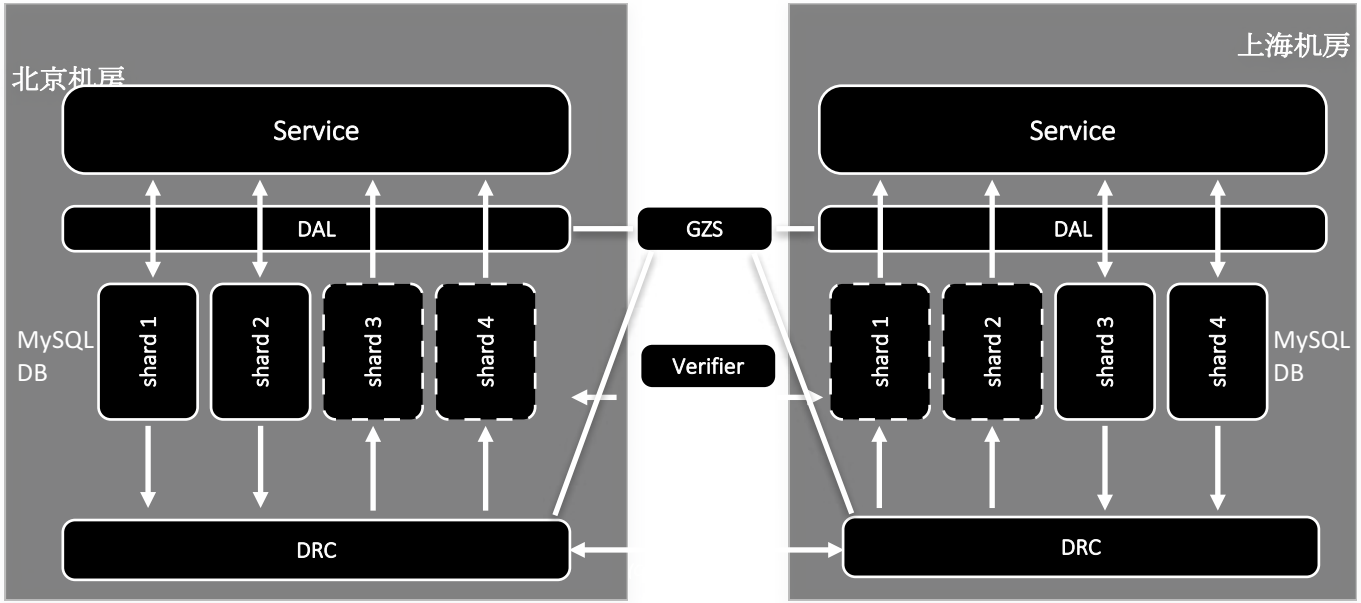
用户流量路由：用户，商户，骑手根据地理位置路由到对应的ezone



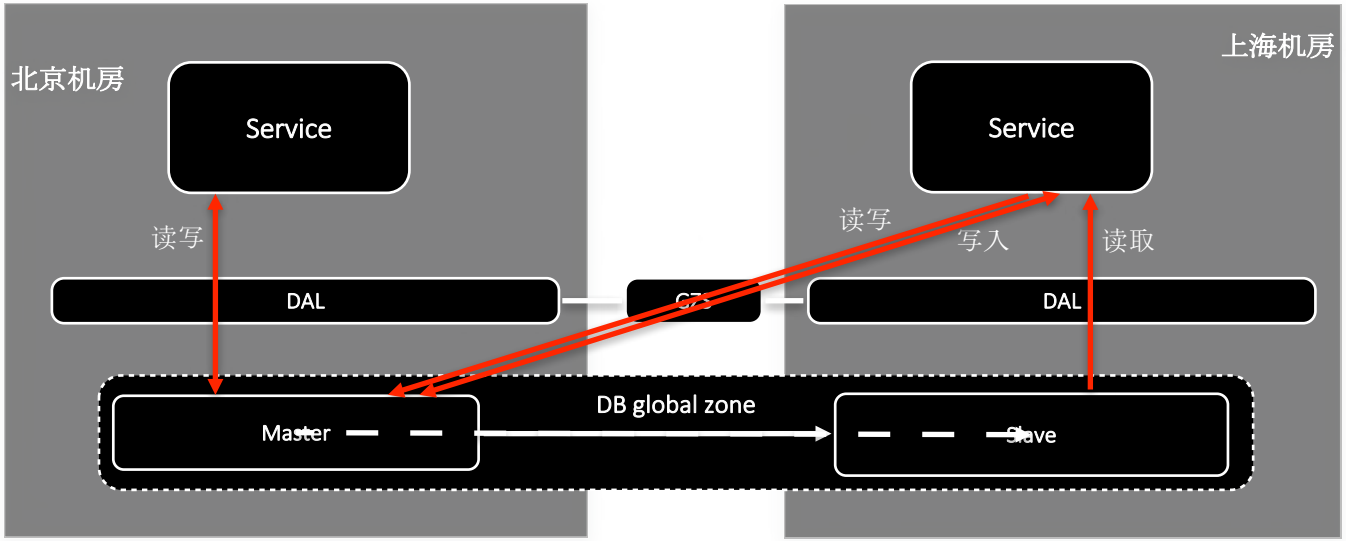
每个ezone对应一个数据中心，交互单在一个ezone内完成，ezone内的数据相互复制，复制到其他数据中心。

global ezone：对强一致性有要求的数据放在global zone，global zone在单中数据单元复制

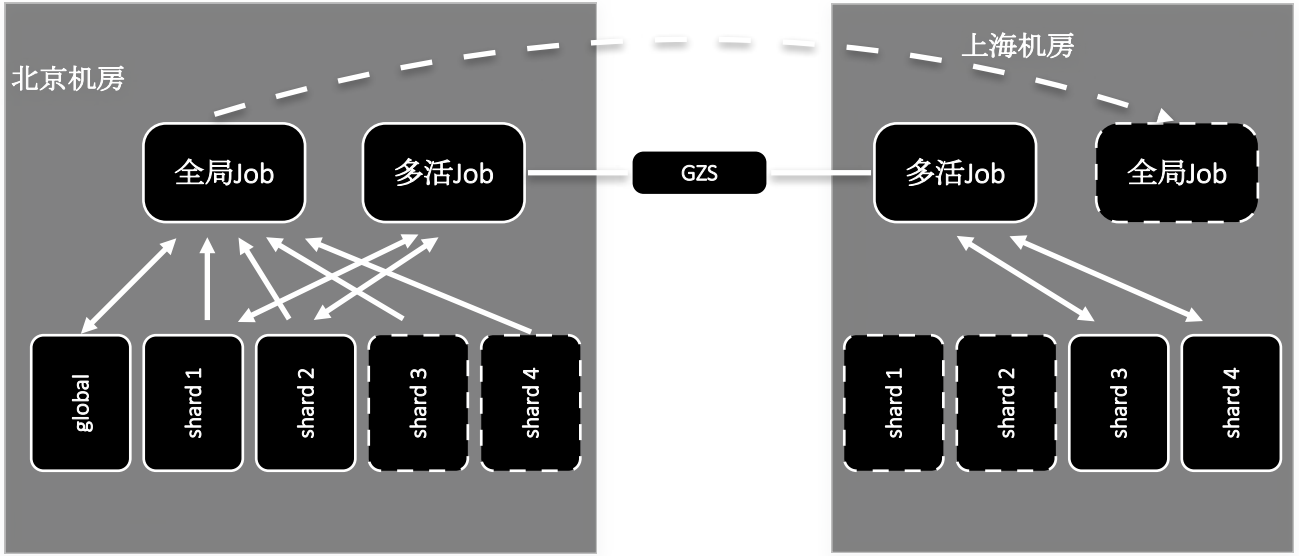
多活设计：数据库同步


北京机房
Service
DAL
MySQL
DB
shard 1
shard 2
shard 3
shard 4
DRC
GZS
Verifier
上海机房
Service
DAL
shard 1
shard 2
shard 3
shard 4
MySQL
DB
DRC

多活设计：全局强一致性业务



多活设计：后台任务


北京机房
全局Job
多活Job
GZS
global
shard 1
shard 2
shard 3
shard 4
上海机房
多活Job
全局Job
shard 1
shard 2
shard 3
shard 4

多活设计：基础组件、工具

- 改造已有组件和工具
  - DAL
  - PIZZA
  - Python框架
  - Java框架
  - Trace
  - Metrics
  - 发布系统
- 新增组件和工具
  - IDC Router
  - GZS
  - DRC
  - SOA Proxy
  - DB比对工具
  - 一键部署工具Eone
  - 多活控制台gzs console

Q&A

高效运维社区
GreatOPS Community

GOPS2017
Beijing

会议

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

培训

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训

咨询

- 企业DevOps 实践咨询
- 企业运维咨询

商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com

Thanks

高效运维社区

开放运维联盟

荣誉出品

想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好