中生代技术 FRESHMAN TECHNOLOGY | IT大咖说 不止于技术

容器生态在百度金融的应用

pippo1980.du@gmail.com

我们做了什么

1.微服务开发框架
2.服务注册/发现
3.服务网关

1.镜像自动构建
2.交付环境自动搭建
3.多套环境配置管理

1.业务日志采集
2.热点服务分析
3.服务单元扩容管理

1.灰度发布
2.小流量实验&多版本共存

交付治理

需要解决的业务问题

一次交付涉及多个
模块的构建安装

沟通成本

环境复用导致
代码/版本不稳定

多套环境
维护成本极高

持续交付时配置
经常不匹配
(开发/QA/联调/沙盒)

业务线的期望

并行交付能力

- 服务元数据（接口API契约，接口协议等）依靠人工维护，经常有信息缺失、更新不及时的情况
- 项目并行时存在的服务依赖缺乏有效管理，导致时间消耗在沟通、联调

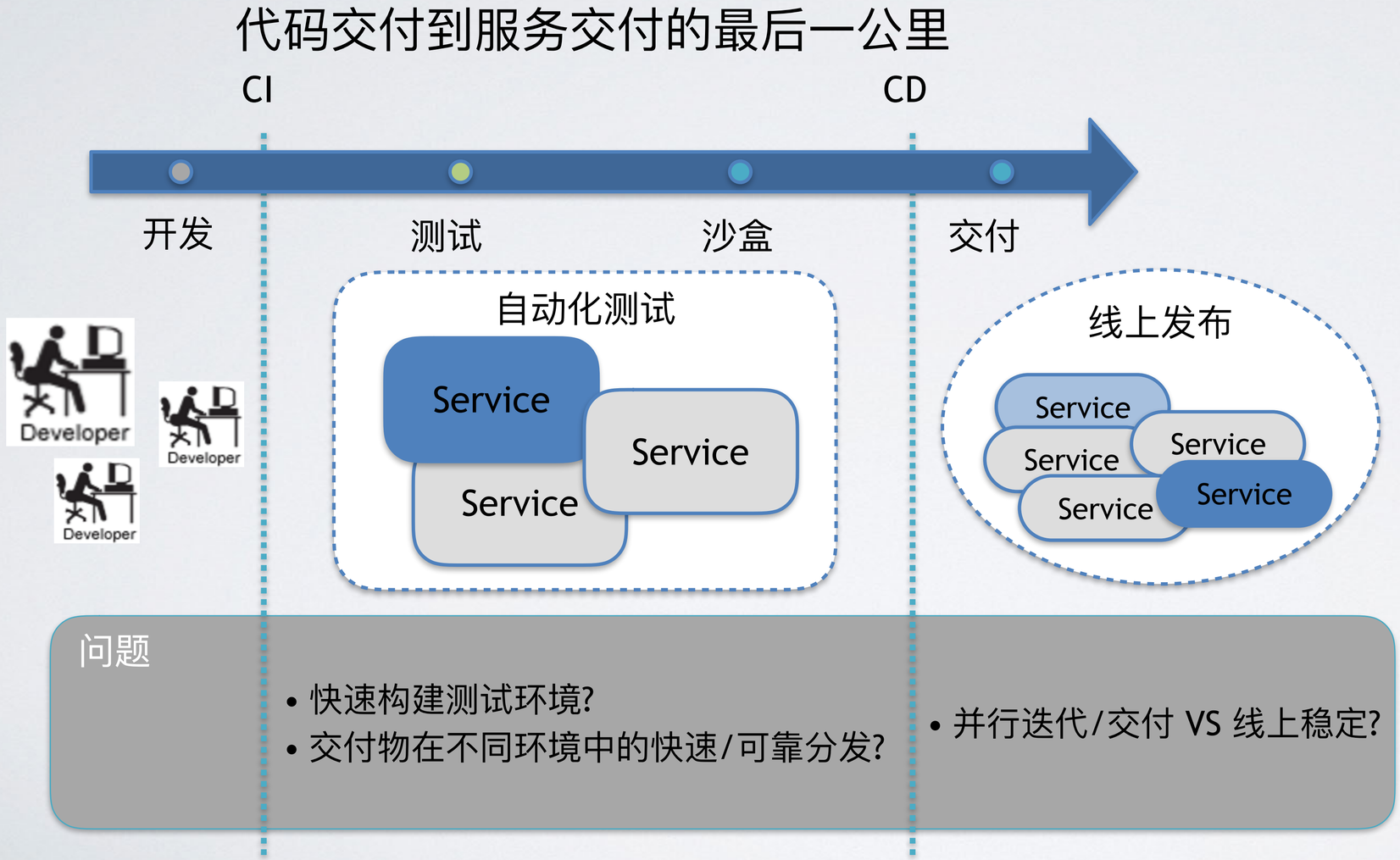
持续交付能力

- 线下环境管理混乱，集成环境构建时由于环境、版本等问题消耗大量的时间
- 同一产线的会共用集成环境，难以保证环境稳定，引入大量不确定因素

交付质量的保障

- 线下环境与线上环境存的运行环境不统一，存在无法通过QA验证的Case
- 交付物的业务配置与环境强耦合（构建期合并），配置管理不完善导致不必要的线上故障

如何做到



1. 镜像自动构建
2. 配置自动下发
3. 整套环境自动部署
4. 多套环境各自隔离

交付治理基础设施

Dev

- 服务构建
- 镜像构建

QA

- 测试环境构建
- 集成测试
- 沙盒测试

1
- 镜像自动构建
- 依赖关系自动扫描
- 服务API Doc自动生成

2
- 测试环境快速搭建
- 服务配置自动切换

DevOps交付平台(EMC平台)

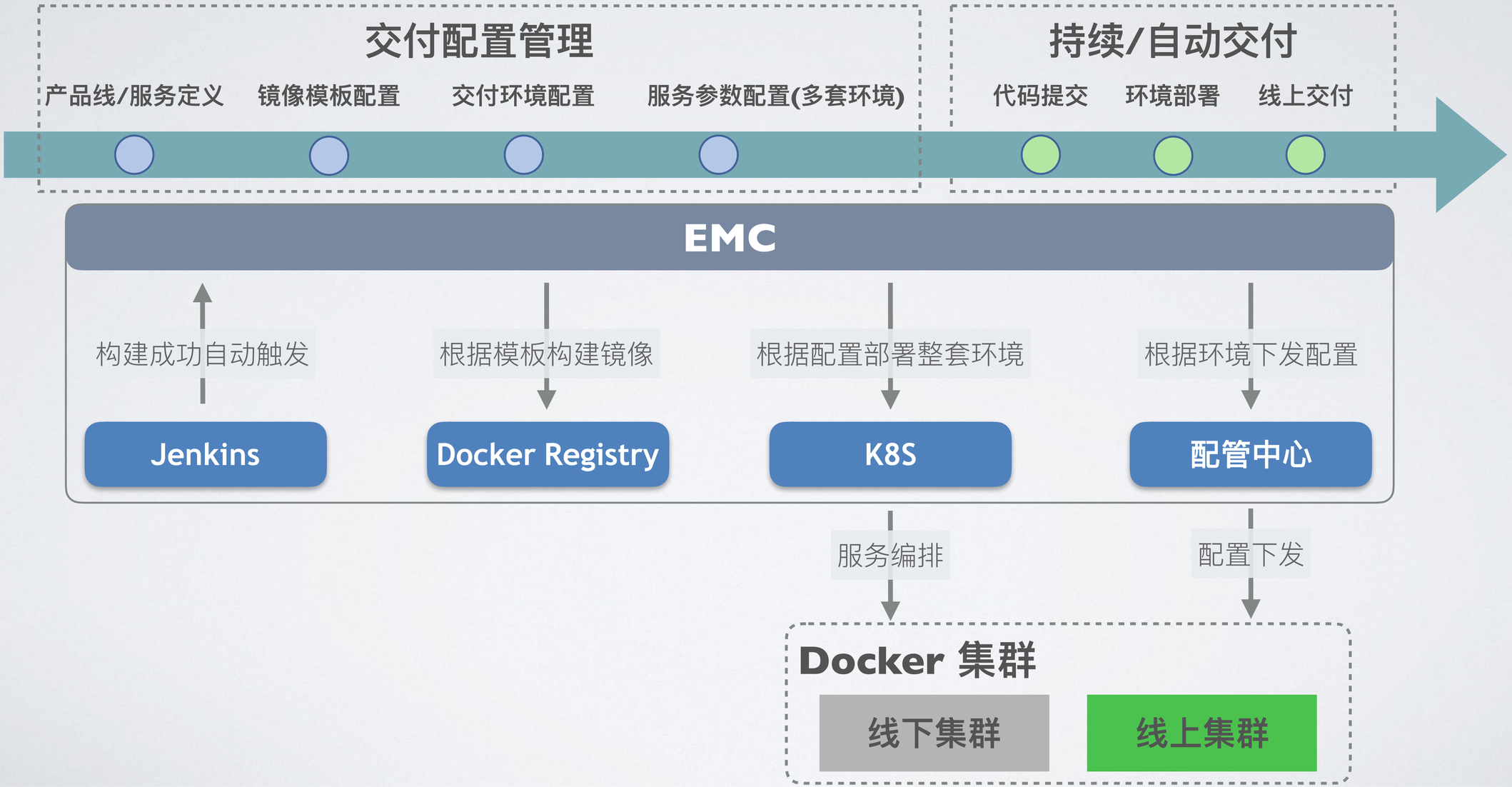
镜像元数据管理	服务依赖管理	服务API Doc
编排管理	镜像仓库	配置中心
线下 Docker 集群		K8S

3
- 沙盒自动交付及验证
- 上线辅助分析评估

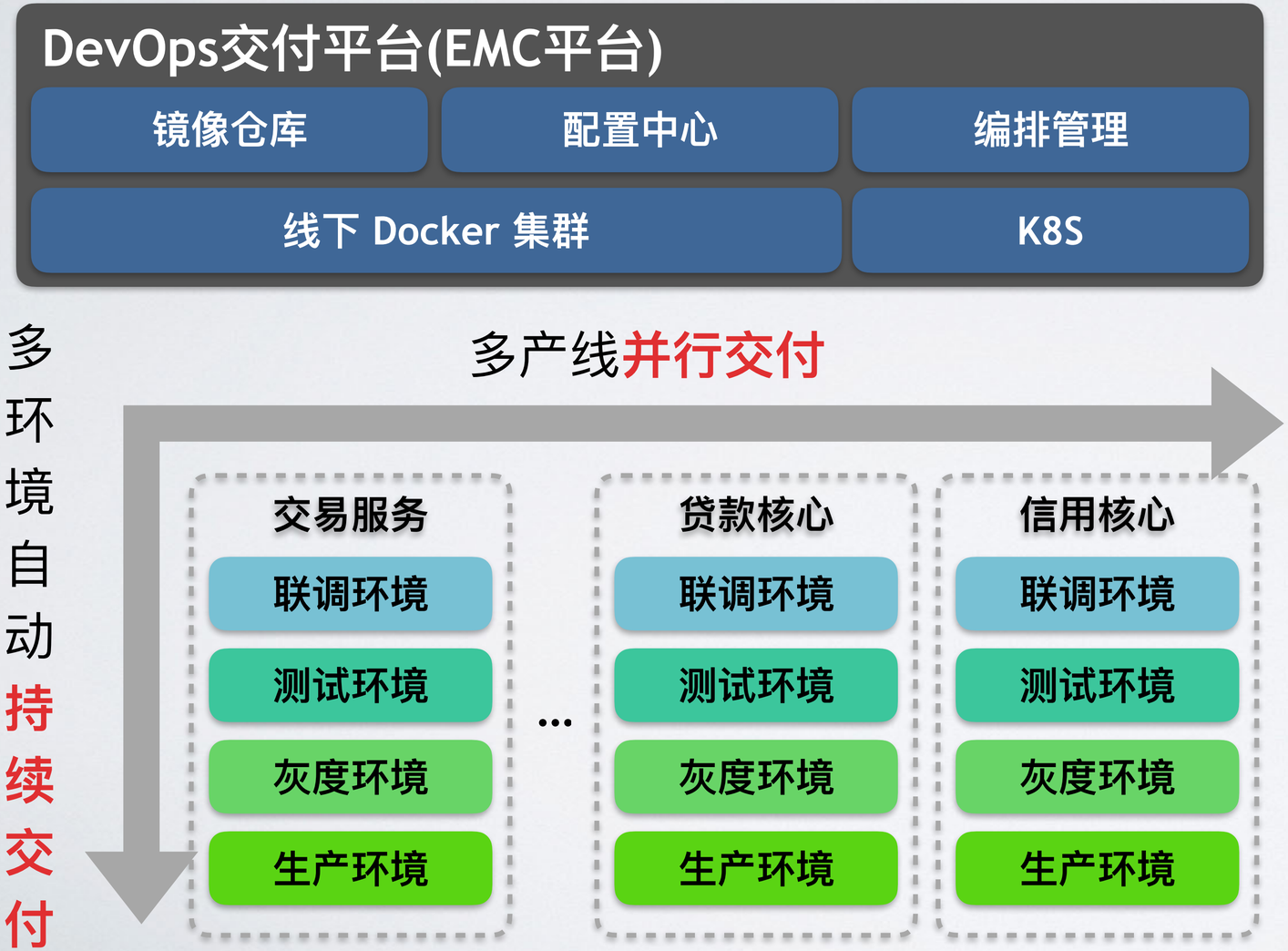
线上Docker集群

- 沙盒环境
- 生产环境

交付流程规范化/自动化


交付配置管理
产品线/服务定义
镜像模板配置
交付环境配置
服务参数配置(多套环境)
持续/自动交付
代码提交
环境部署
线上交付
EMC
构建成功自动触发
根据模板构建镜像
根据配置部署整套环境
根据环境下发配置
Jenkins
Docker Registry
K8S
配管中心
服务编排
配置下发
Docker 集群
线下集群
线上集群

环境隔离&并行交付



镜像仓库

镜像标准化　通用服务无缝集成/升级

- 标准镜像在注册后,可在不同环境无缝流转
- 通过基础镜像,透明的集成公司监控/安全通用服务组件

配置中心

配置/代码分离　配置运行时获取

- 镜像在通过容器启动时,根据实际的产线/app抓取配置
- 提供与Spring框架集成的配置组件,实现配置的动态更新

编排管理

联调依赖环境构建　环境并行隔离

- 根据服务依赖数据,自动的调度/编排集成环境
- 不同环境互相隔离,避免服务共用依赖导致的环境/版本差异

功能展示-镜像模板配置


基础镜像
服务镜像
当前用户可管理的产品线
EMC
仓库管理
基础镜像
应用镜像
线下环境
任务列表
3 FSG
DEMO
FSG
通用基础镜像
业务基础镜像
自定义组件
通用组件
supervisor-tomcat-wrapper 1.0
node 7.2.1
node 6.9.2
nginx 1.11.6-http2
nginx 1.8.0
supervisor 3.3.0
jdk 1.8.0_92
jdk 1.7.0_79
tomcat 8.0.35
tomcat 7.0.67
镜像模板配置
基本信息
组件名称
tomcat
文件列表
文件名
apache-tomcat-7.0.67.tar.gz
执行命令
镜像制作
cd /home/work
mkdir /home/work/tomcat
cp -rf apache-tomcat-7.0.67/* /home/work/tomcat
rm -rf apache-tomcat-7.0.67
rm -rf /home/work/tomcat/webapps/*
容器运行
Runtime
SuperVisor
Supervisor Conf
运行信息
对外端口
环境变量
http
8080
CATALINA_HOME
/home/work/tomcat
勾选的端口表示主端口
关闭
取消
服务镜像
基本信息
镜像名称
qa-loan-version
镜像版本
1-0-0
默认仓库
fcore-loan
线上镜像
基础镜像
fcore-base/jdk8-tomcat8:1.0
构建信息
FTP
文件上传
FTP HOST
FTP PATH
Username
Password
包名
镜像内存放路径
/home/work
自动解压
保存FTP配置
文件名
镜像内路径
操作
app.war
/home/work/tomcat/webapps/ROOT
编辑
删除
执行命令
镜像制作
Dockerfile RUN command

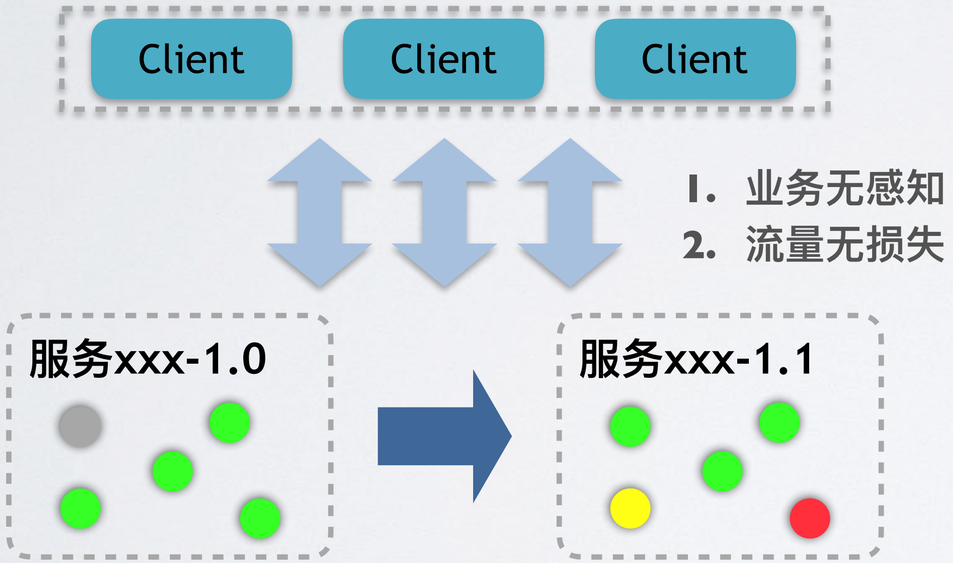
功能展示-交付环境治理


EMC
仓库管理
基础镜像
应用镜像
线下环境
任务列表
联系我们
帮助文档
3 FSG
Demo
同一产品线/多套隔离环境
创建环境
环境名称： sirius-sample 环境类型： 开发环境 ID： 305
添加模块
名称: portal
状态: RUNNING
配置 更新 实例 历史
名称: service
状态: RUNNING
配置 更新 实例 历史
名称: schedule-task
状态: RUNNING
配置 更新 实例 历史
交付环境模块定制
环境名称： cfg-hello 环境类型： 联调环境 ID： 187
添加模块
名称: portal
状态: RUNNING
配置 更新 实例 历史
名称: service
状态: RUNNING
配置 更新 实例 历史
环境名称： dubbo-client 环境类型： 沙盒环境 ID： 58
名称: schedule-task
状态: STOPPED
配置 更新 实例 历史
选择可用模块镜像
添加模块
mtp
openresty
mysql
ci
Edit
保存
取消

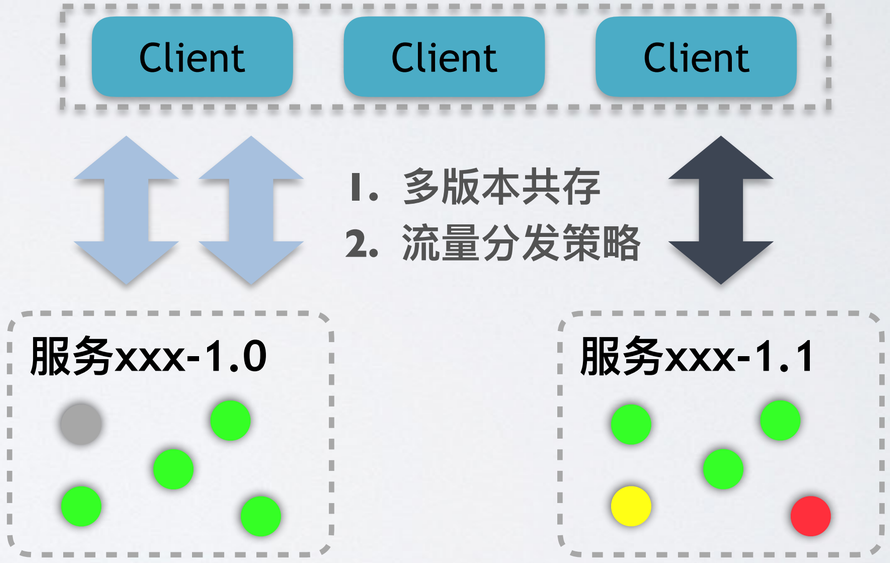
发布管理

业务场景

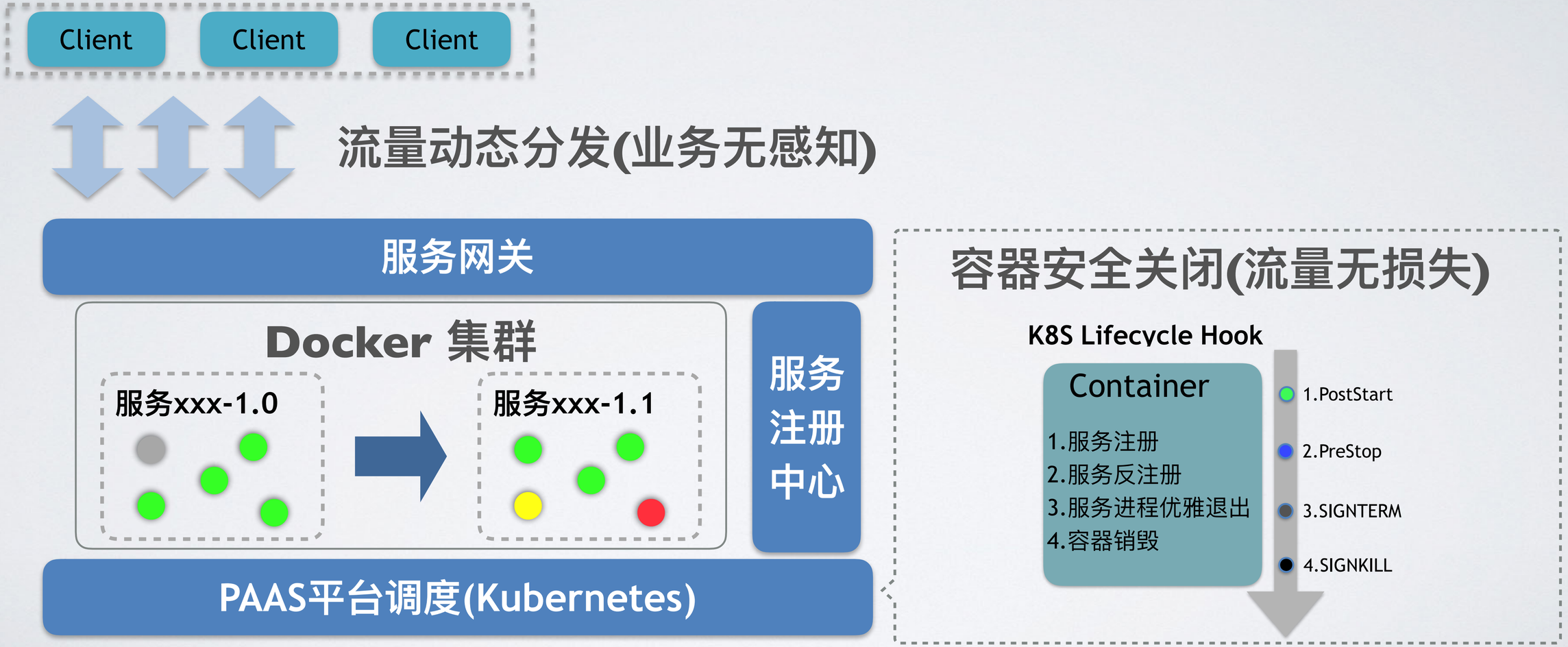
场景一:灰度发布



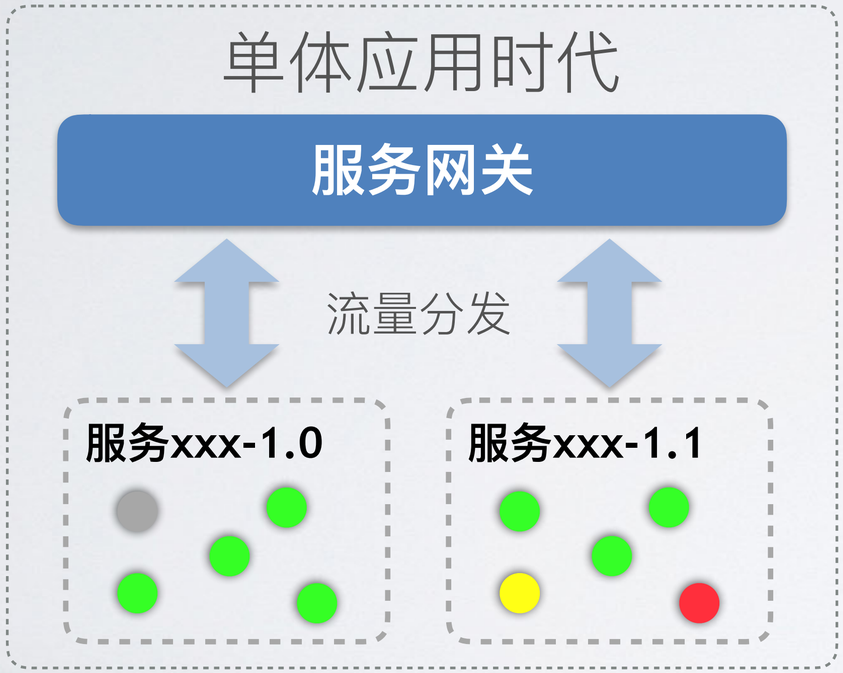
场景二:小流量及多版本共存



灰度发布

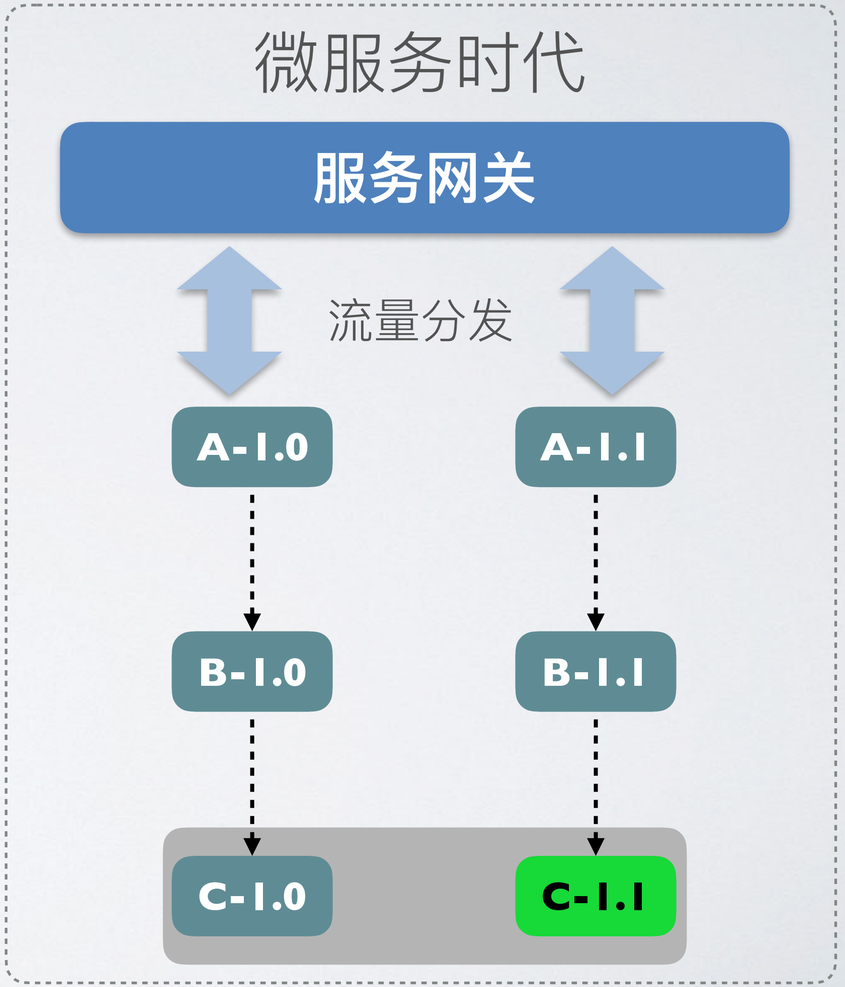

Client
Client
Client
流量动态分发(业务无感知)
服务网关
Docker 集群
服务xxx-1.0
服务xxx-1.1
服务注册中心
PAAS平台调度(Kubernetes)
容器安全关闭(流量无损失)
K8S Lifecycle Hook
Container
1.服务注册
2.服务反注册
3.服务进程优雅退出
4.容器销毁
1.PostStart
2.PreStop
3.SIGNTERM
4.SIGNKILL

小流量及多版本共存

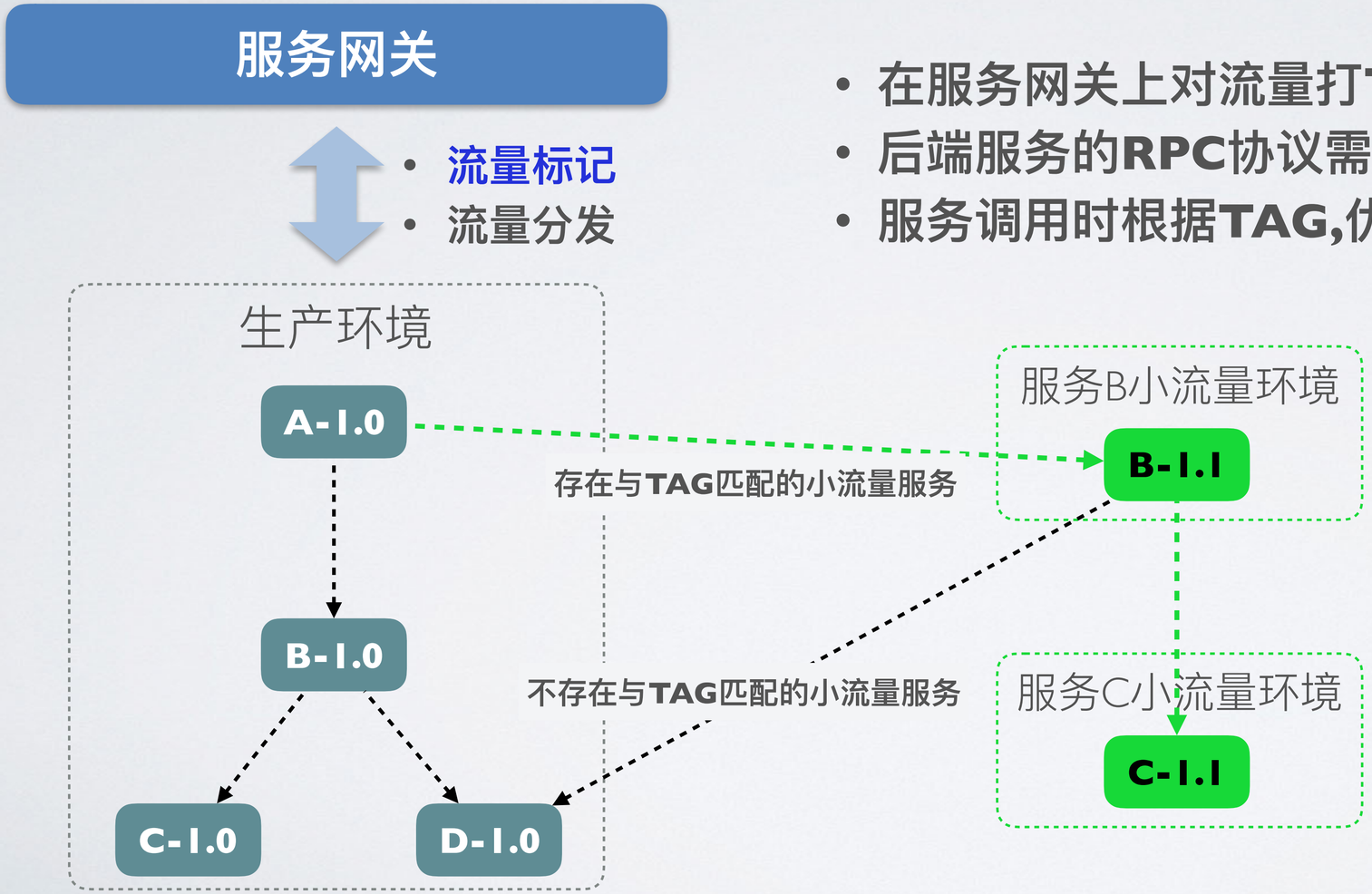


可用但不高效

- 一个服务的小流量版本往往需要整套链路的单独部署
- 资源利用率低
- 管理成本高



解决方案优化

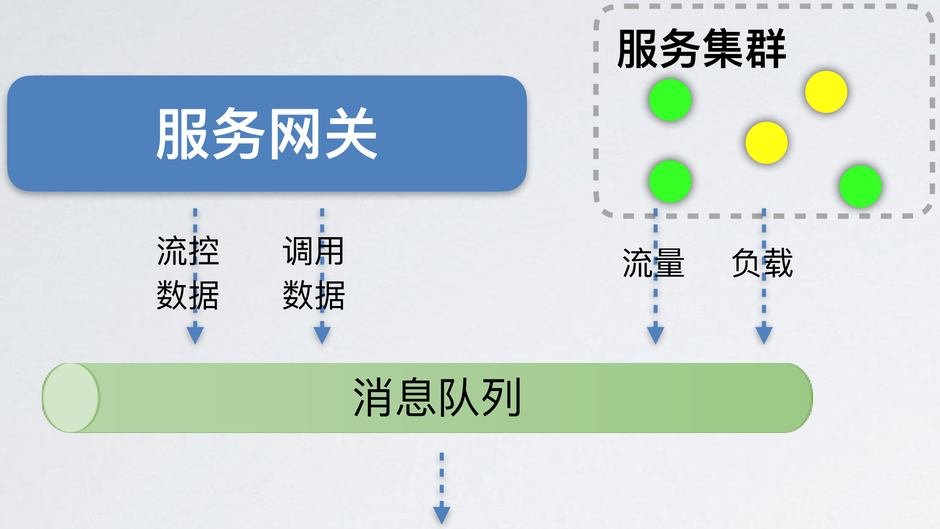


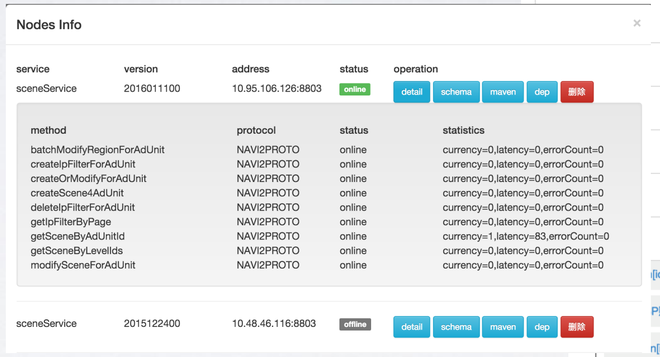
- 在服务网关上对流量打TAG(需要适配后端RPC协议)
- 后端服务的RPC协议需要在调用链路中透传TAG
- 服务调用时根据TAG,优先寻址小流量环境

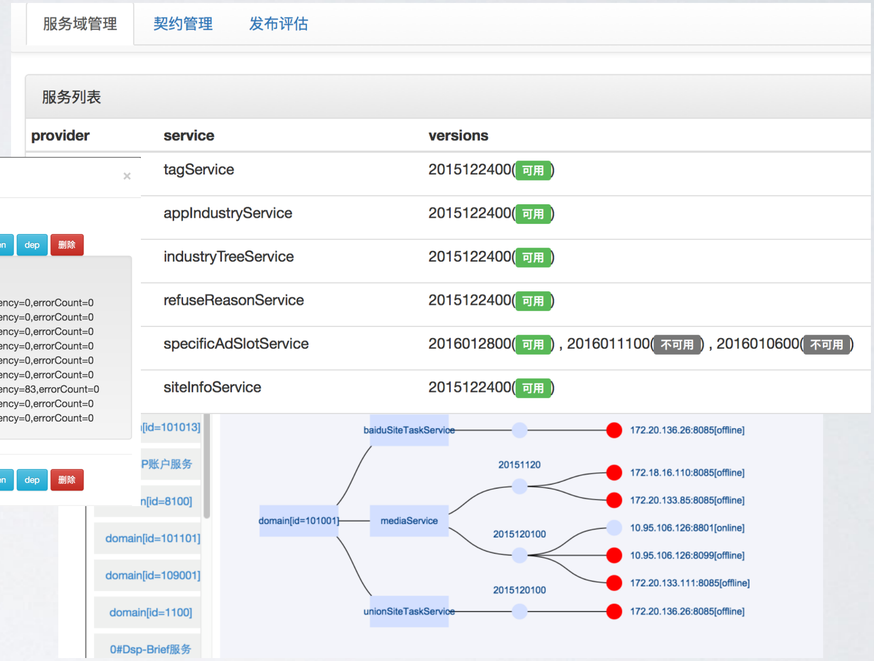
1. 只需要部署待验证服务
2. 支持多套小流量环境并存
3. 资源利用率高/配置灵活

服务监控与弹性扩容

服务监控



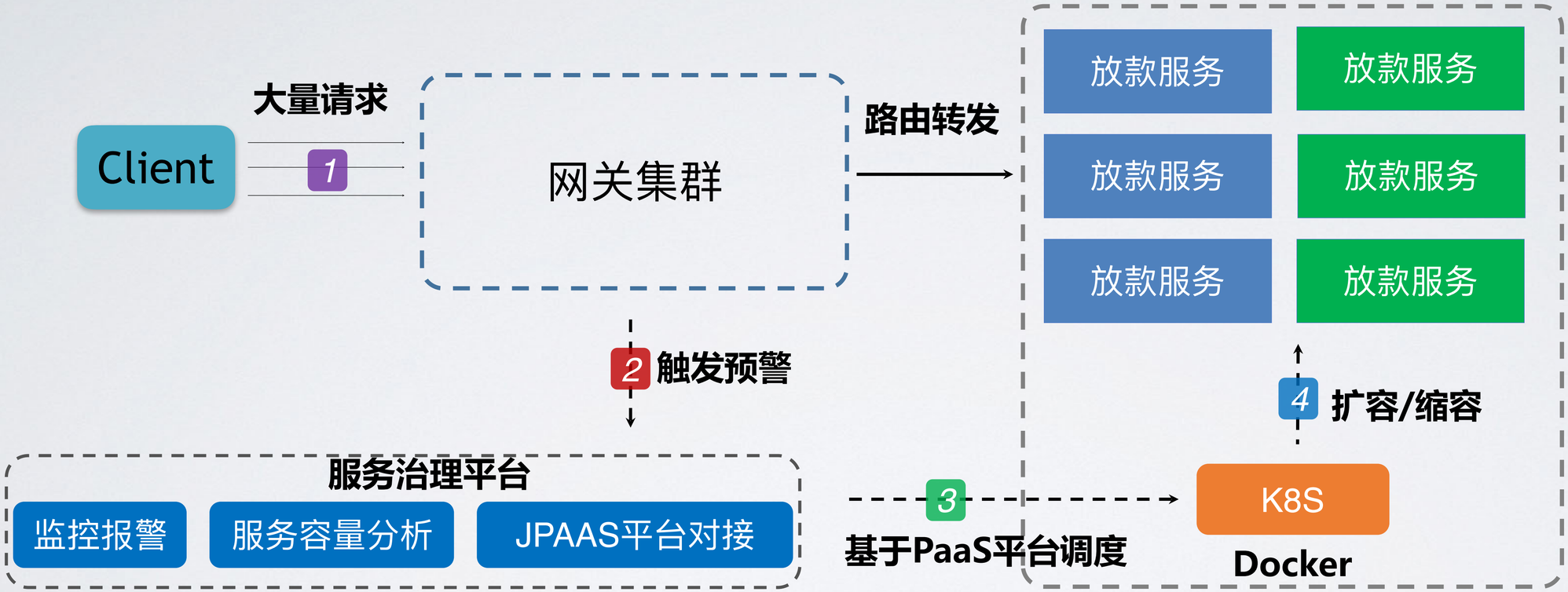




快速/准确的分析/计算是治理成功的关键

- 通过调用数据绘制实时的物理调用链路拓扑
- 根据流量/负载等数据分析服务热点及服务容量
- 与日志平台&JPass对接,打通线上排查/日志分析/弹性扩容多个环节

弹性扩容



1.涌入大量服务请求，如秒杀等营销活动

2.服务治理平台通过容量分析，触发服务预警

3.通过PaaS平台，进行弹性扩容

✓**计算资源并不是不够，而是没有充分被利用**

✓**服务中瓶颈的触发，往往先于计算资源的瓶颈**

✓**业务驱动，弹性伸缩；PaaS平台，计算资源弹性伸缩**

Thanks

–Pippo