

# 目录

- Insight
- Strategy
- SubSystem

Insight

致力于的目标:

极高的自动化水平、极高的效率

现状:

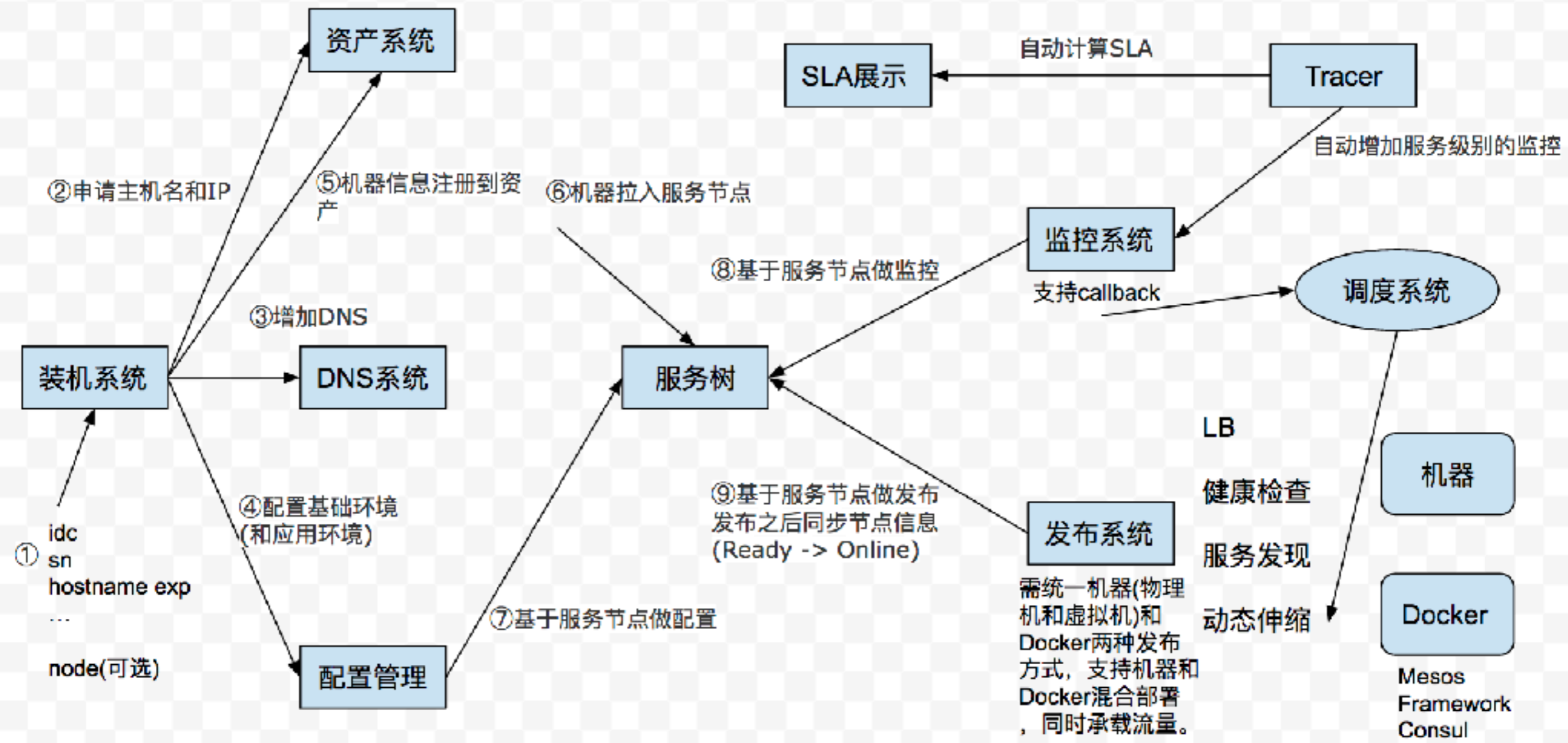
- 手工操作太多;
- 运维工具不成体系;
- ...

解决方法:

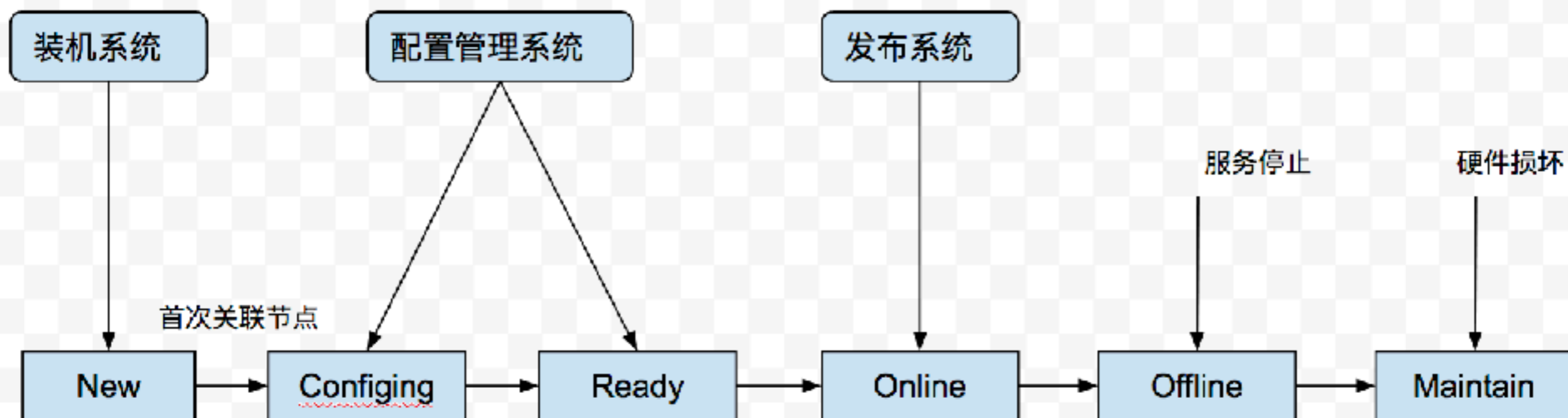
通过对运维的不断思考，从运维整体的角度出发，构建一套完善的运维自动化工具体系。

Strategy

运维系统架构图



## 机器状态变迁过程





SubSystem

# 服务树

服务树的重要性：

服务树是运维系统里最核心、最重要的组件，它的重要性在于它负责维护服务和服务器的关联关系这一运维中最主要的数据。

# 服务树

服务树的功能：

- 维护服务与服务器的关联关系；
- 按业务组织成树状结构；
- 与上下游运维系统联动

# 服务树

节点设计:

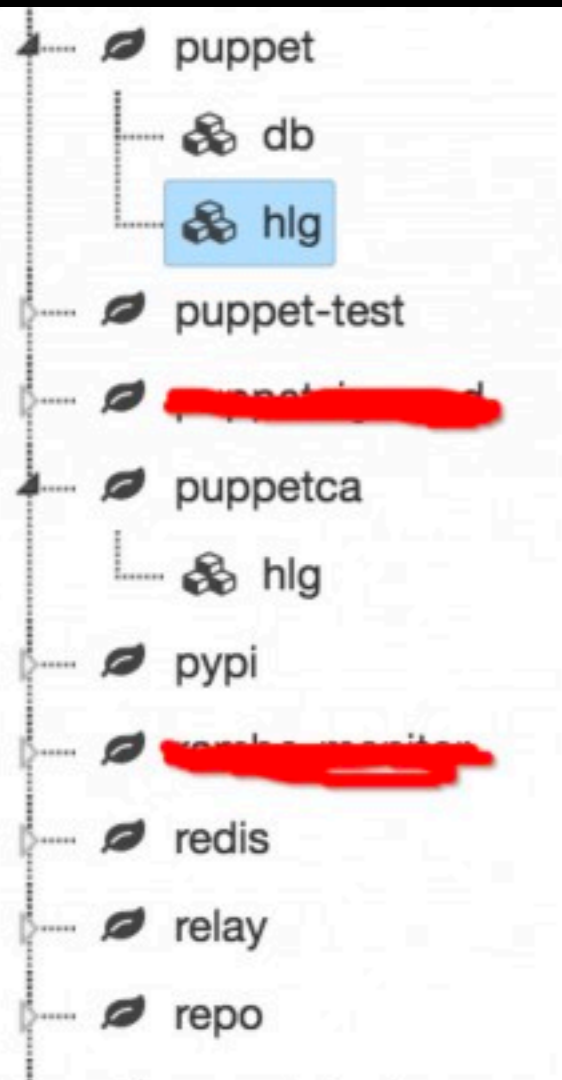
- 节点管理: ID、名称、路径、负责人;
- 服务器关联: 机器名、状态(online、configing...)、权重
- Docker 实例: 实例名称、所属宿主机、(权重)
- Docker 配置: 是否使用 Docker、流量占比
- Upstream(可选): 名称、端口

# 服务树

支持机器和 Docker 同时承载流量

	Docker	机器
总流量	$x\%$	$(1-x)\%$
单个实例流量	$x\% / \text{replicas}$	可以自定义(支持小流量)

# 服务树



节点管理

服务器关联

~~节点管理~~

## 节点详细信息

ID: 3570

名称: hlg

路径: ~~/...~~/puppet/hlg

负责人:

添加子节点

重命名节点

移动节点

删除节点

添加负责人

# 资产系统

- 记录机器分区、控制卡 IP、网卡 bond 等可变信息;
- 记录机器状态、机器所属服务节点;
- 为装机系统提供申请主机名和 IP 的 API

# DNS 系统

为什么需要自建 DNS? 使用公网 DNS 的坏处:

- 需要每台机器可以访问外网;
- 有可用性问题;
- 机器名没法解析成 IP(也不能反解);
- 不能自定义内网域名



# DNS 系统

关注点:

- 每个机房需要两台 DNS;
- 数据从中心机房向其他机房同步(MySQL 主从)
- 每个机房的机器使用本机房的 DNS(配置管理);
- 每台机器使用 Dnsmasq 做缓存

# 配置管理系统

功能:

- 配置基础环境(kernel、nofiles、user/group、agent...);
- 配置应用环境(应用的依赖或特殊配置)

# 配置管理系统

设计要点:

- 拉入节点时触发配置(New -> Configing -> Ready);
- 支持自定义 cron;
- 支持查看历史配置状态

# 配置管理系统

实施难度:

- 历史包袱重，机器环境复杂；
- 混布严重；
- ...

# 监控系统

- 统一多套监控系统；
- 可扩展；
- 和服务树整合

# 监控系统

和服务树整合：

- 机器拉入节点监控自动增加；
- 机器拉出节点监控自动删除；
- 新增节点监控，节点下机器自动生效；
- 节点迁移，监控自动迁移；
- 自动向节点负责人发送报警；
- 报警策略复用；
- 系统监控自动增加

# 监控系统

使用文本继承

# 发布系统

要点:

统一机器(物理机和虚拟机)和Docker两种发布方式, 支持机器和 Docker 混合部署, 同时承载流量。



# 发布系统

Docker 部分

「此页略」

# 发布系统

## 机器 部分

重构基于机器的发布，向 Docker 看齐：

- LB （负载均衡器 + 服务发现 SDK）
- 健康检查 (监控系统)
- 服务发现 (守护进程)
- 动态伸缩 (需要编写调度系统，依赖监控系统、装机系统、发布系统等组件，较重)

# 发布系统

机器 部分

机器的部署原则：

- 不混布；
- 实在拆不开，可以尝试虚拟机(讨论)

# 权限系统

作用:

- 为服务树、DNS 管理系统、配置管理系统、监控系统、发布系统等运维系统提供权限控制;
- 机器登陆权限控制 (何时做?)

# Tracer

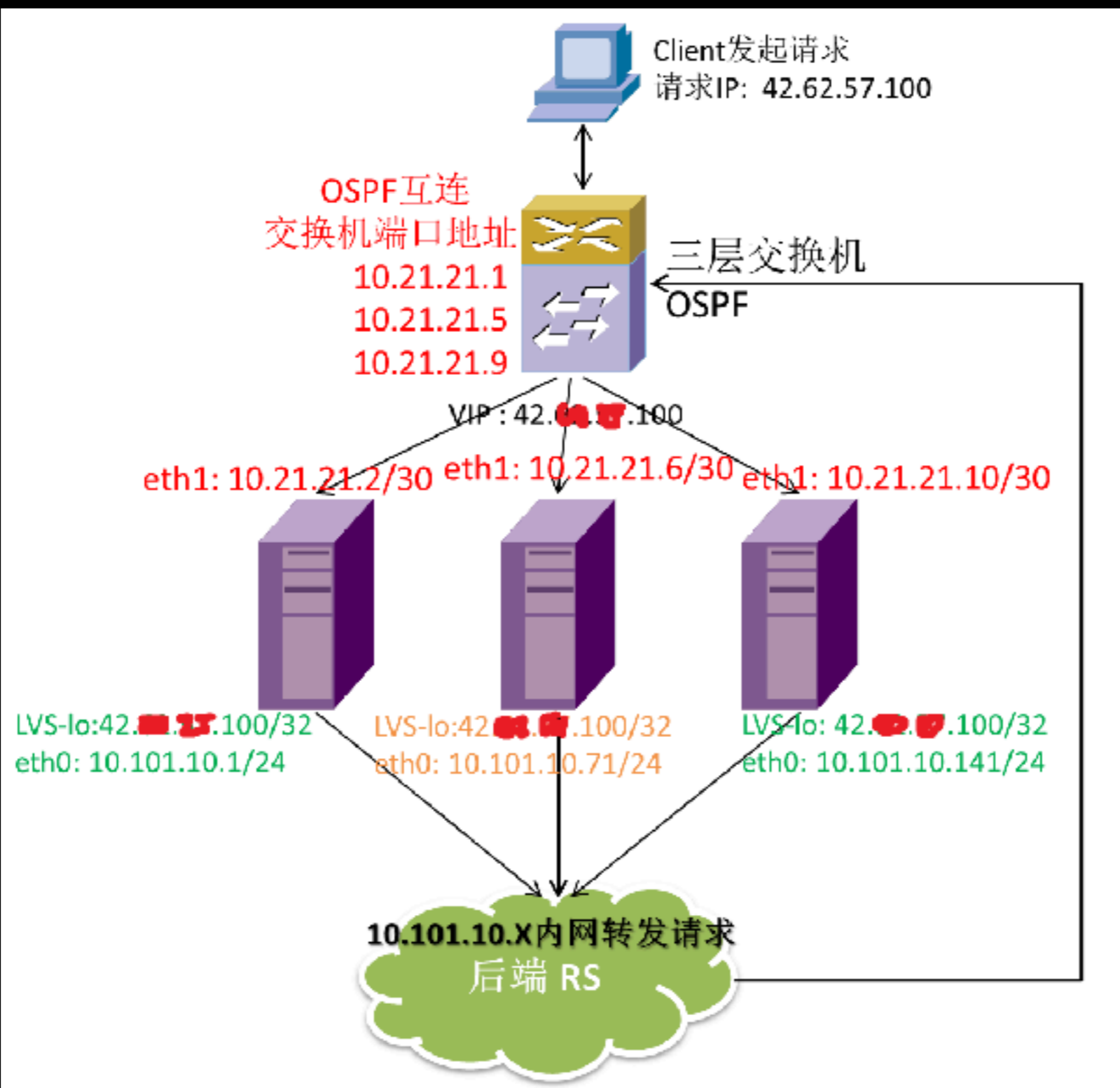
- SLA 展示(DONE);
- 实现服务级别的监控;
- 基于 Trace id 做性能分析;
- 自动计算服务依赖树

# 统一负载均衡器

方案：

OSPF + LVS + FULLNAT(DR)

# 统一负载均衡器



# 统一负载均衡器

优点：

- 可水平扩展；
- 多台 LVS 同时工作；
- 高可用(有点问题)



# 统一负载均衡器

问题：

- LVS 宕机会造成 Hash 重分布，影响其他 LVS 机器的连接；
- CPU 瓶颈

# 统一负载均衡器

解决方法：

- Consistent Hashing、Session 同步;
- 中断绑定、DPDK(需要开发)

OVER