

# PHP

2017·北京

全球开发者大会

高可用的 PHP

# 聊聊稳定性保障那些事

阿里技术专家 信海龙

# 个人简介



河北大学政法学院



毕业后，我就成了程序猿



目前就职于阿里巴巴



# 我的联系方式



信海龙

博学无忧

[www.bo56.com](http://www.bo56.com)





# 目录

常见的稳定性问题都有哪些

稳定性保障的几个基本原则

稳定性保障的一些案例解析



# 常见的稳定性问题都有哪些

# 环境不一致问题



# 开发测试的时候，风驰电掣



哦也！总算TMD的开发完了



上线后，举步维艰



这TMD什么破路！



# 服务未隔离问题



# 高速上，车辆有序的行驶





# 一个汽车出现问题，可能就会有连锁反应

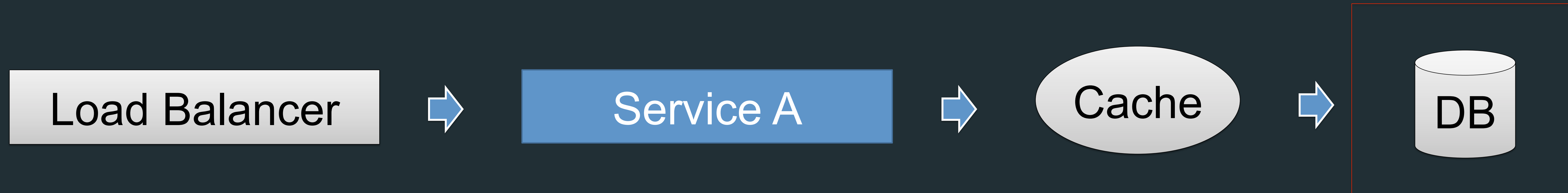




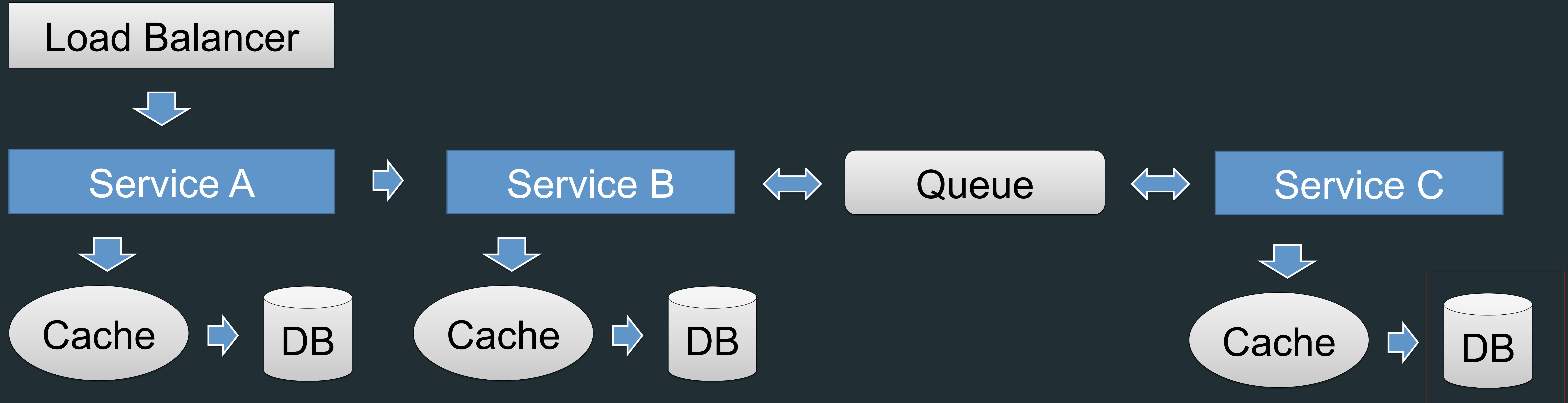
# 服务化后，问题排查难的问题



# 单体应用，架构比较简单



# 服务化后，调用关系复杂



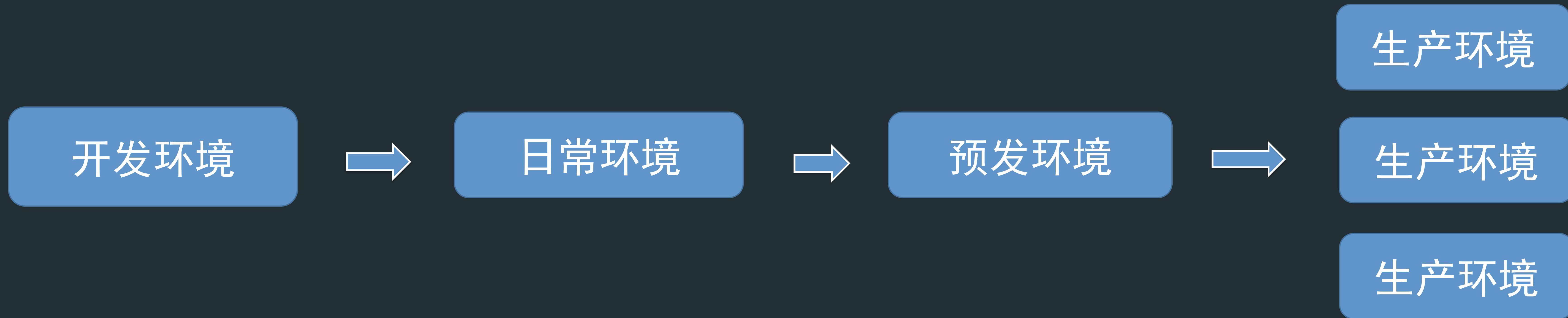


# 稳定性保障的一些原则



# 环境一致性原则

# 从开发到上线经历的环境



# 环境一致都需要注意哪些因素

版本

配置

路径

# 环境隔离原则

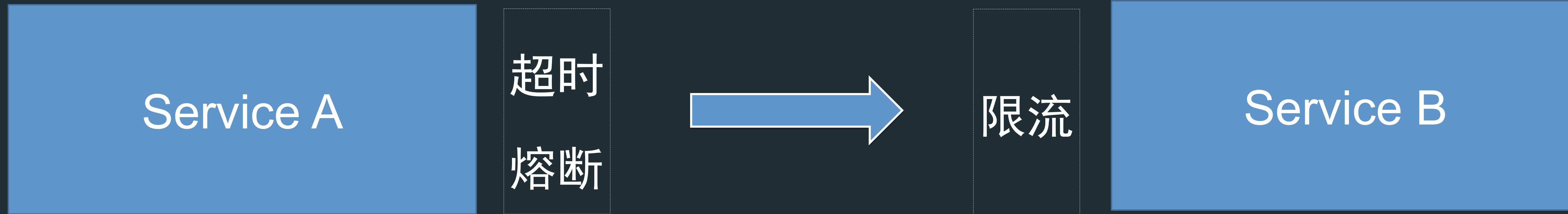


# 物理隔离

Service A

Service B

# 服务隔离





# 有效监控原则



# 监控项目

## 系统

CPU    Disk    Memory

Network    Process    .....

## 业务

Error Log    Data Status

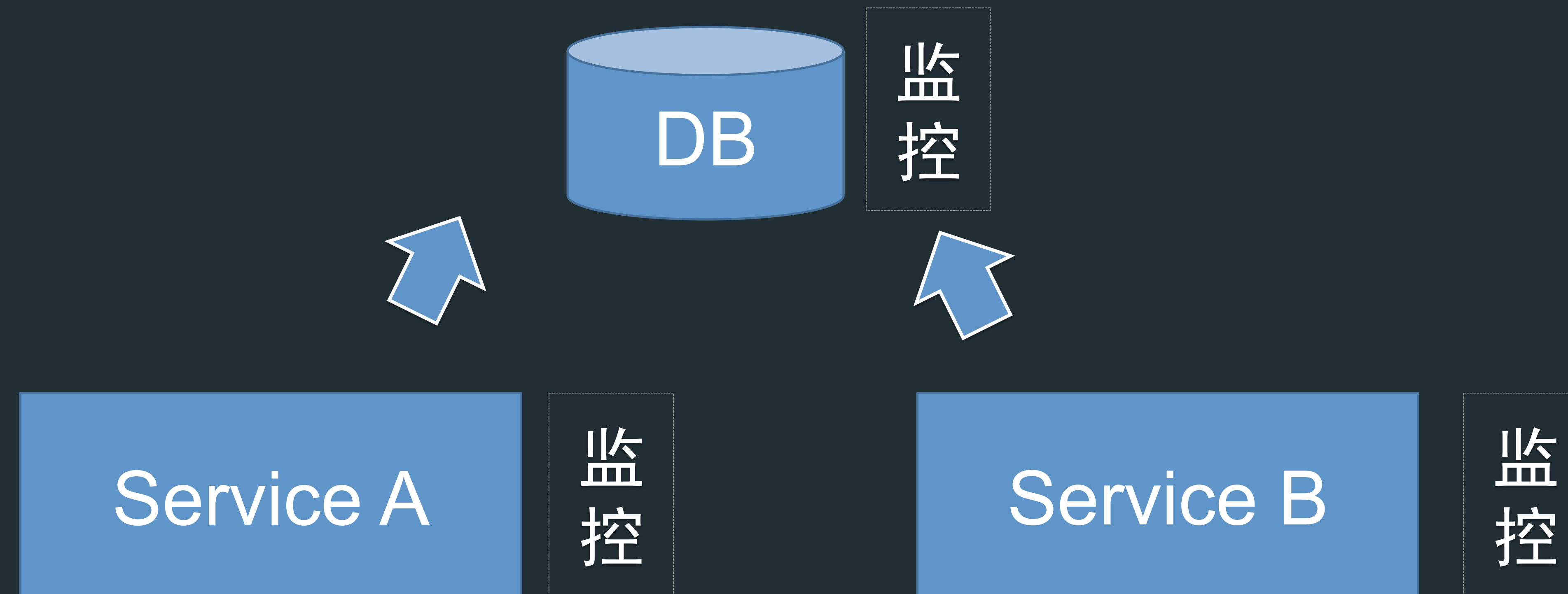
.....



# 监控策略

## 监控可用性

- ◆ 不只是监控服务是否存在
- ◆ 更重要的是监控服务是否可用





# 报警策略

不要让报警成为骚扰

- ◆ 不要把报警项设置的太敏感
- ◆ 不同级别不同的报警方式
- ◆ 报警内容要有意义

```
【GOC】 15:02:17 [C]  
SEARCH_etao_asp[ET2] etao-  
asp[10.178.0.24].et2 php-  
fpm (process_num:91),
```



# 充分预案原则

# 充分预案原则

## 面向问题预案

- ◆ 想一想都会遇到哪些问题
- ◆ 所使用的任何服务都不可靠
- ◆ 服务不可用时，如何处理

## 预案策略

- ◆ 设定超时，避免过长等待
- ◆ 默认数据，服务不可用时使用
- ◆ 预案尽量平台化，自动化执行



# 充分预案原则

## 预案验证

- ◆ 周期性验证
- ◆ 模拟服务不可用，封端口
- ◆ 查看服务表现是否符合预期

```
sudo iptables -A OUTPUT -p tcp --dport 80 -j DROP
```

# 追根溯源原则



# 追根溯源原则

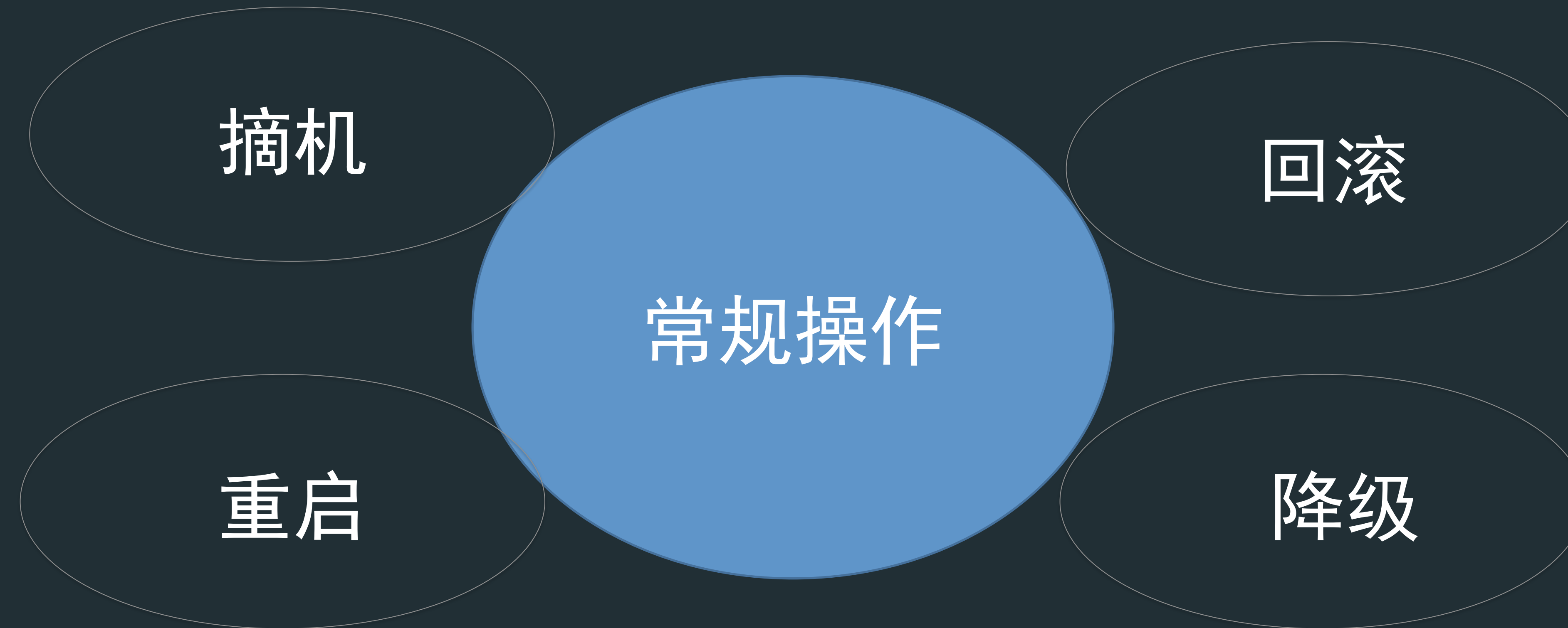
验证

问题排查

恢复服务

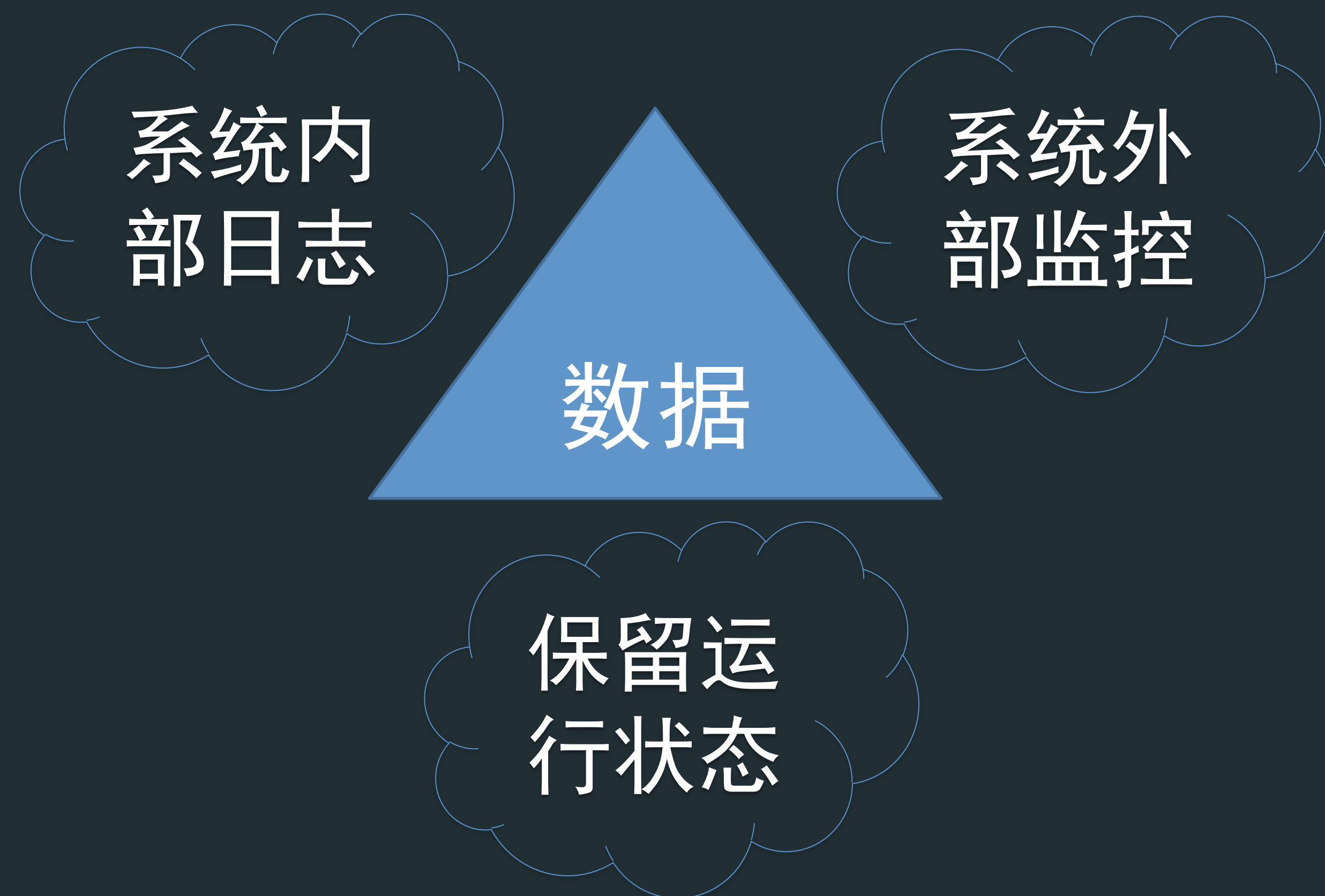
保留现场

# 恢复服务



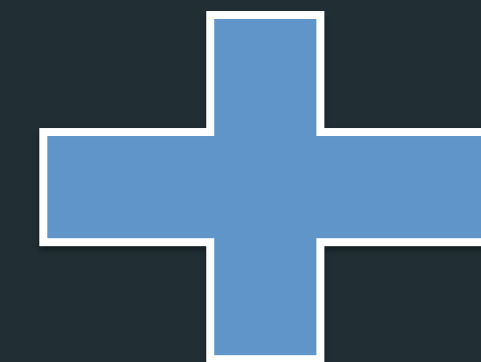
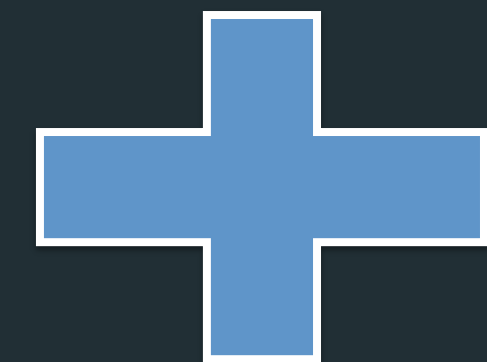


# 保留现场



保留现场的目的是为了获取数据

# 问题排查





# 问题排查工具

## 系统

- ◆ top 查看进程资源排行
- ◆ vmstat 服务器的当前运行状态
- ◆ tsar 收集系统和应用的信息

## 进程

- ◆ gdb 进程调试工具
- ◆ pstack 查看进程的栈信息
- ◆ strace 查看系统调用情况

## 网络

- ◆ tcpdump 网络抓包
- ◆ wireshark 图形界面抓包工具
- ◆ netstat 显示网络连接等信息

# 稳定性原则回顾

环境一致

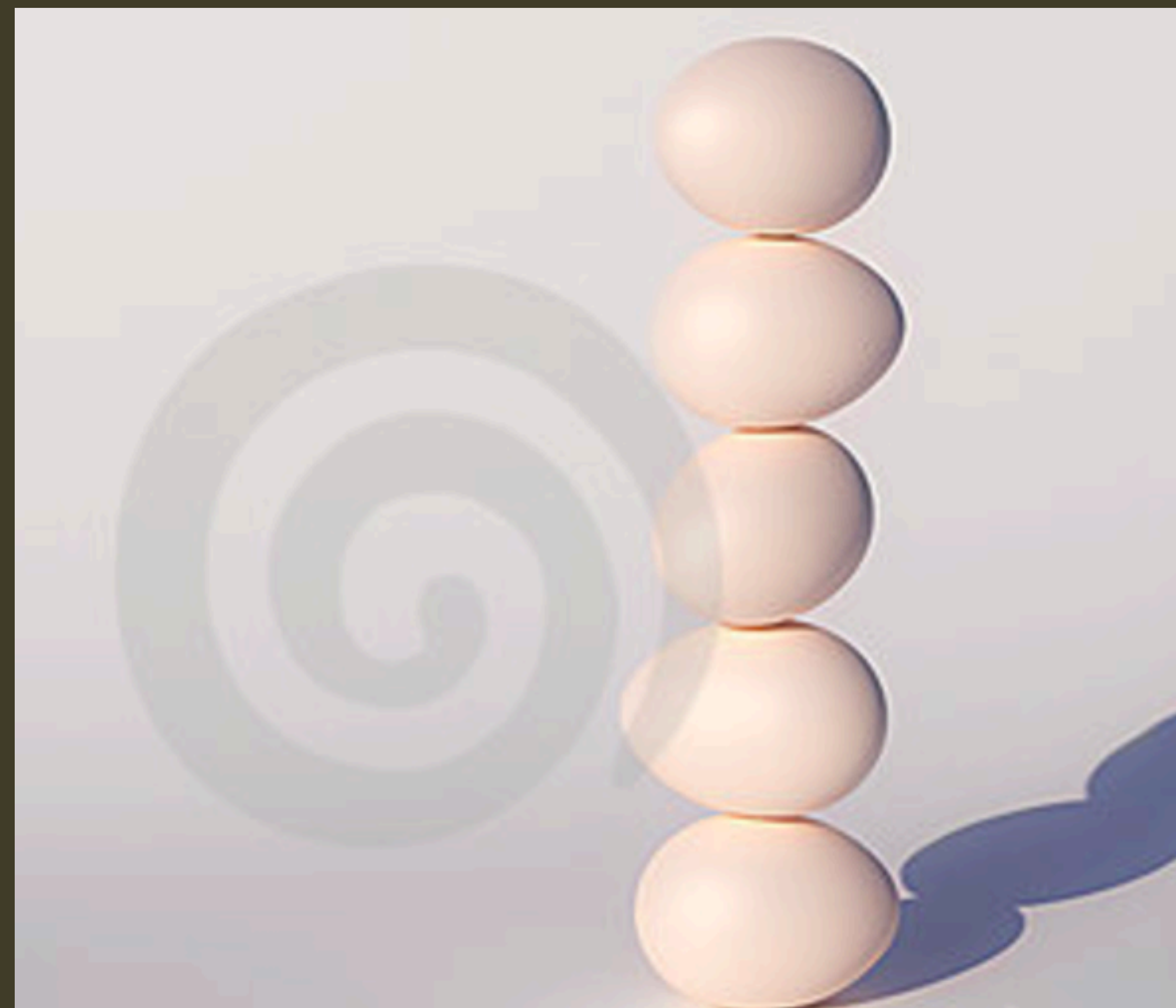
应用隔离

有效监控

充分预案

追根溯源





# 稳定性保障的一些案例

# 由一个PHP漏洞引发的环境一致性改造



# PHP漏洞

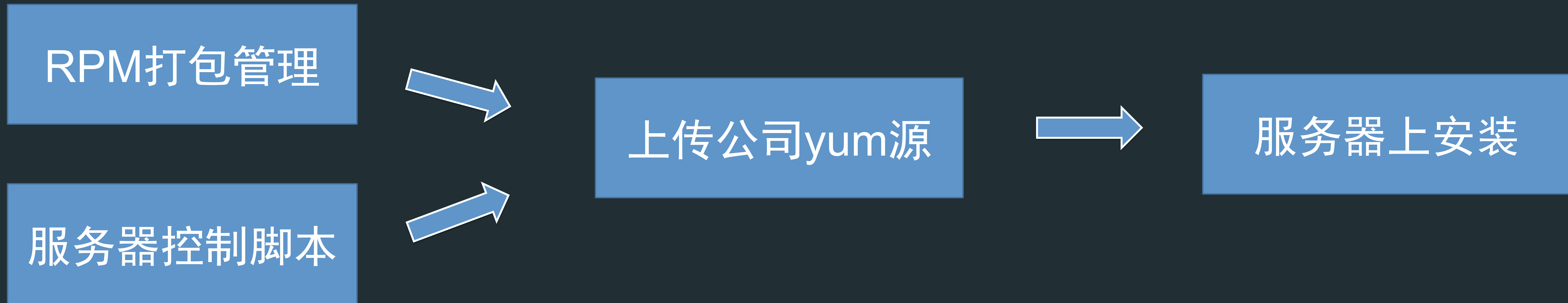
- ◆ PHP 5.0.0 - 5.0.5
- ◆ PHP 5.1.0 - 5.1.6
- ◆ PHP 5.2.0 - 5.2.17
- ◆ PHP 5.3.0 - 5.3.29
- ◆ PHP 5.4.0 - 5.4.40
- ◆ PHP 5.5.0 - 5.5.24
- ◆ PHP 5.6.0 - 5.6.8

multipart/form-data 远程DOS漏洞

# 环境不一致



# 改造流程





# 为什么是RPM

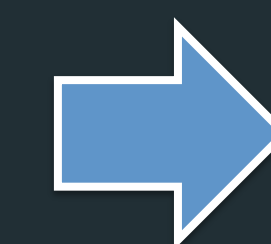
## RPM

- ◆ 打包方便
- ◆ 安装简便
- ◆ 解决依赖关系

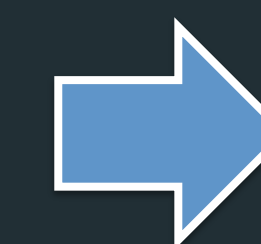
%pre



%post



%preun



%postun

# 最终Docker化



Docker化，保障机器无状态

# 我们在服务隔离方面做的一些事情



## 未隔离前，遇到的问题



## 隔离后的状态



## 其他隔离方面的一些事情

读写分离

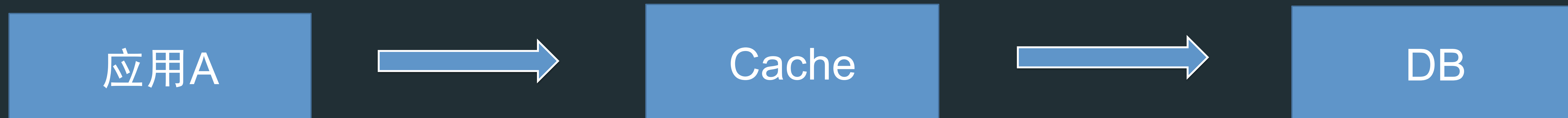
SQL分级

慢SQL隔离

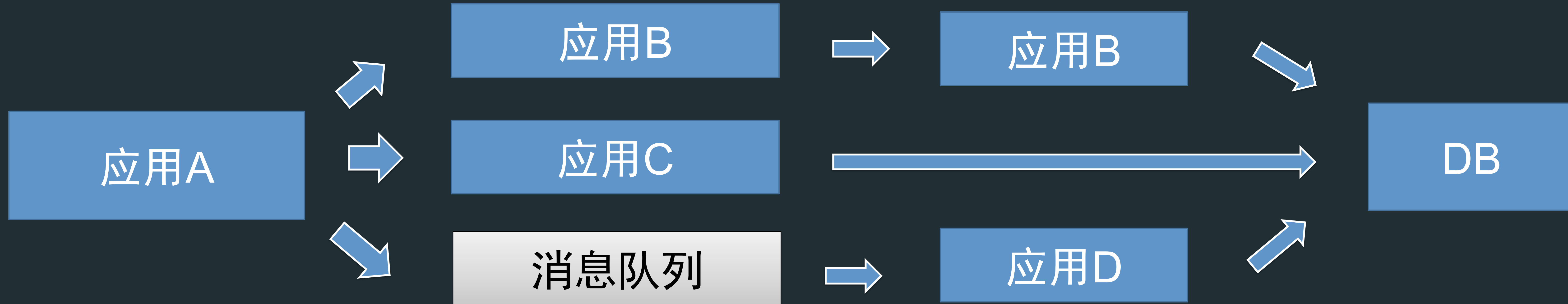


# 如何设计服务调用链路分析系统

# 简单单体应用



# 复杂应用问题排查





# 调用链

ac18adb113661968589621810

调用链入口 IP: 172.24.173.177, 开始时间: 2013-04-17 19:07:38.962, 调用链总时长: 424ms。 [日志原文](#)

☐ IP ☐ rpcID

应用名	类型	状态	大小	服务/方法	时间轴	
tf_buy	TRACE	OK	-	http://buy.taobao.com/auction/buy_now.htm		293ms
tradeplatform	HSF	OK	670B	tc.TcTradeService@getOutOrderSeqIdByBuyerId~l	0ms	
tradeplatform	HSF	OK	8.0KB	trade.ICreatingOrderService@createOrdersForTaobao~R	46ms	
itemcenter	HSF	OK	5.0KB	item.ItemQueryService@queryItemAndSkuWithPVToText~L	8ms	
itemcenter	HSF	OK	3.0KB	item.SpuService@getSpu~l	3ms	
(notify)	NOTIFY	OK	-	Notfy@send	6ms	
timeoutcenter	NOTIFY	OK	-	Notfy@recv	10ms	
tradelogs	NOTIFY	OK	-	Notfy@recv	10ms	
tmallcommontep	NOTIFY	OK	-	Notfy@recv	12ms	
tradelogs	NOTIFY	OK	-	Notfy@recv	12ms	
tradelogs	NOTIFY	OK	-	Notfy@recv	13ms	
traderecord	NOTIFY	OK	-	Notfy@recv	13ms	



# TraceId

## TraceId

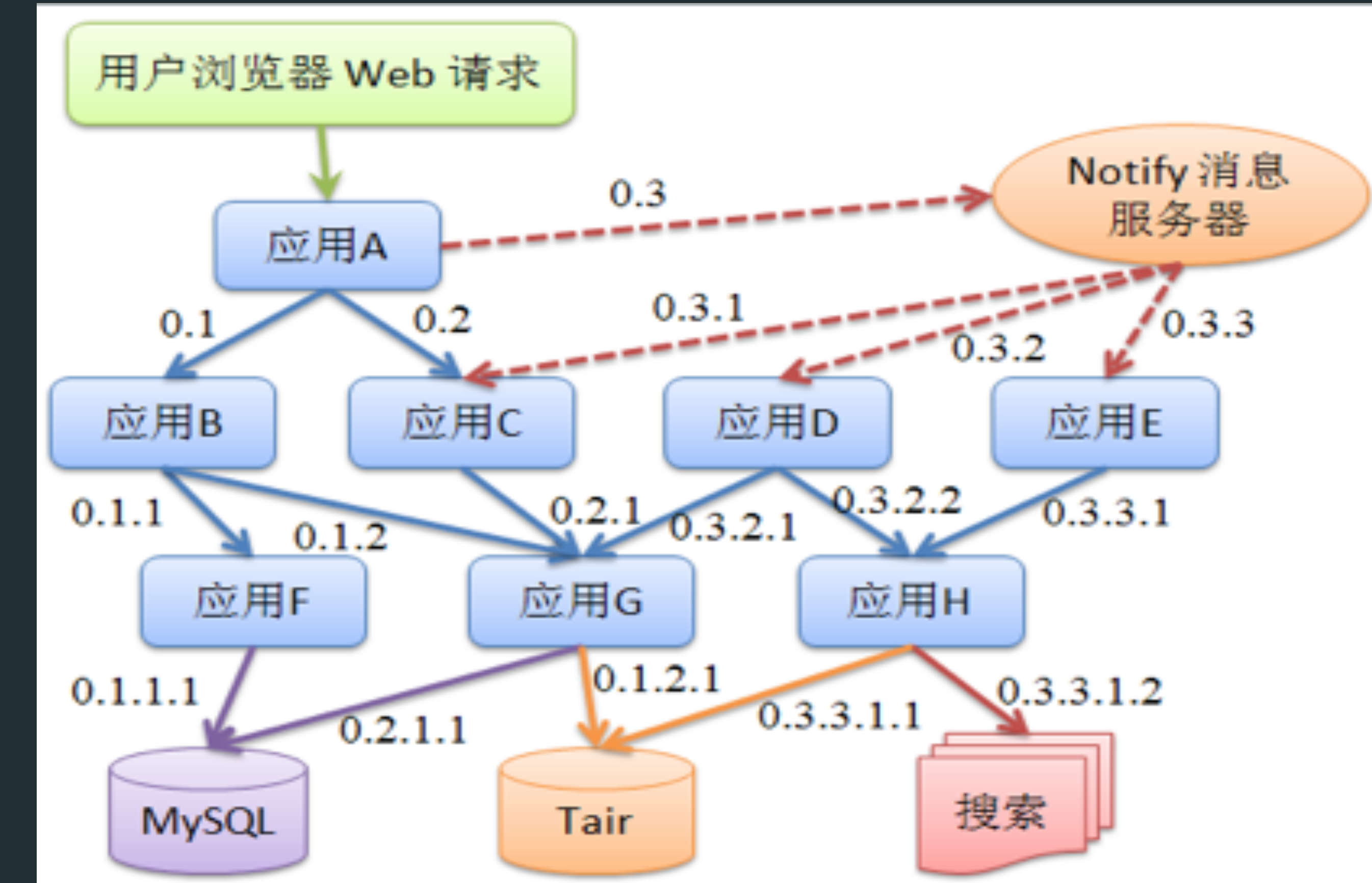
- ◆ 关联相关日志
- ◆ 全局唯一
- ◆ 系统中传递

- IP地址：用于识别前端应用和来源机器
- 创建时间：用于存储时的分区
- 顺序数：用于链路采样
- 标志位：用于调试和标记
- 进程号：记录进程号

# RpcId

## RpcId

- ◆ 标识埋点顺序和嵌套关系
- ◆ 系统间传递
- ◆ 同步、异步、一对多调用



# 埋点数据

开始时间

调用类型

对端IP

响应耗时

传输量

处理结果

其他数据



# PHP 2017·北京

## 全球开发者大会

