MTSC2019

中国移动互联网测试开发大会

Mobile Testing Summit China 2019

2019年6月28-29日 / 北京 国际会议中心

主办方：TesterHome 腾讯课堂

MTSC2019
中国移动互联网测试开发大会

混沌工程在创业公司的实践

酷家乐：小眠

主办方：TesterHome 腾讯课堂

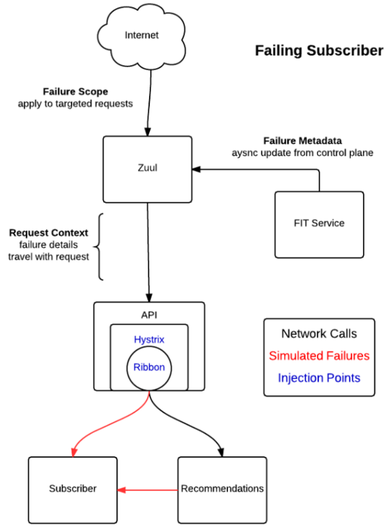
- 混沌工程简介
- 酷家乐技术架构现状
  - 微服务化
  - 基础设施
  - 混沌工程实施条件
- 故障模拟
  - 故障场景抽象
  - 故障实现方案
  - 系统防御能力判定
  - 故障演练与混沌工程融合
- 混沌工程产品化落地&收益
- 总结和展望

混沌工程演进史

2008 年 Netflix 开始从数据中心迁移到云上，之后就开始尝试在生产环境开展一些系统弹性的测试。

2010 随机杀节点

2011 随机杀AWS区域



2014 FIT（Failure Injection Testing）

2017 环境集成、自动化验证

混沌工程原则

EXAMPLE

服务A正常运行时，稳态指标：

- qps、响应时间
- cpu、磁盘、IO
- ……

真实可发生：

- 宕机
- 磁盘占满
- ……

宕机：3台服务器，选择1台

混沌工程实施

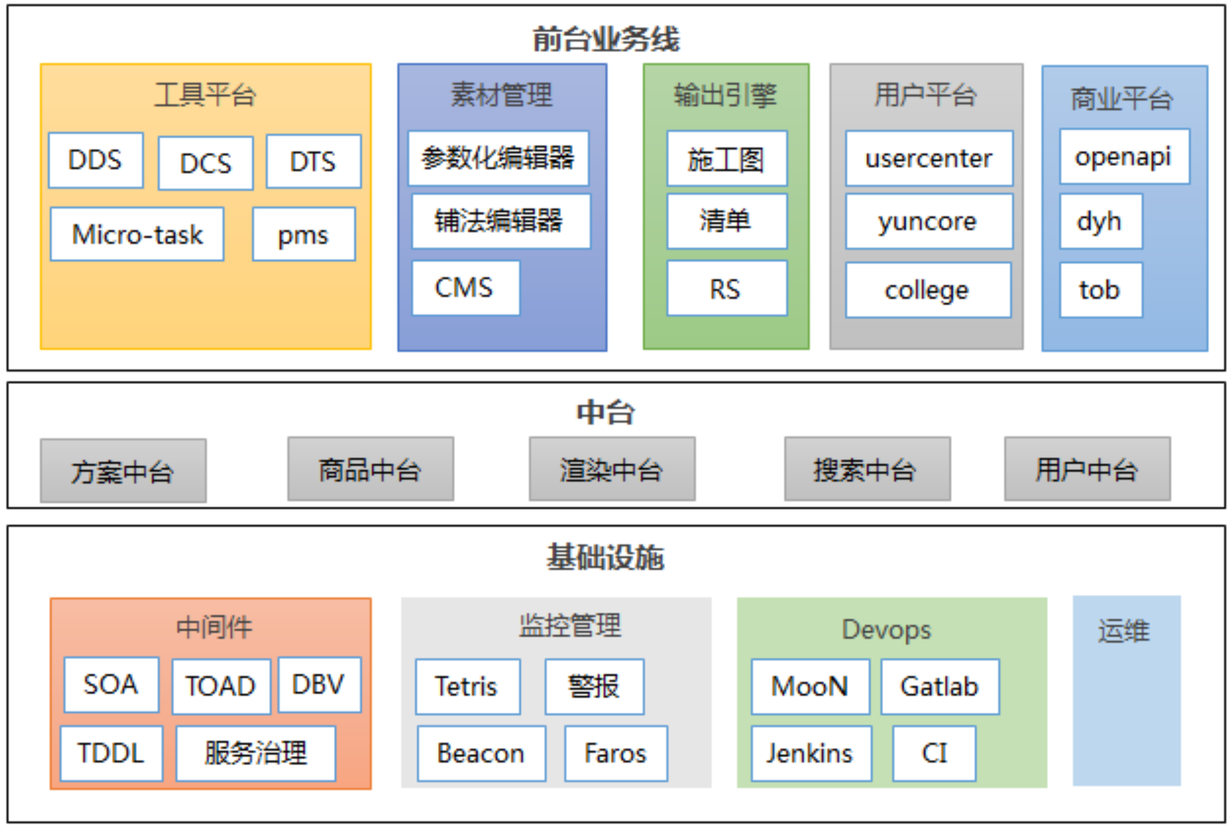

真实事件抽象
归纳当前架构、应用可能发生的故障
1
稳态指标设定
针对不同故障场景，设定合适的稳态监测指标
2
分析爆破半径
对故障后的影响面进行分析，注入确定实验范围
3
预案&回滚策略
明确故障发生后，相关处理预案及回滚方式
4
故障模拟
对抽象的故障场景模拟实现
5
系统防御能力判定
基于稳态指标，对某次攻击判定系统是否具备防御能力
6
持续集成
持续执行混沌工程试验，挖掘隐患
7

- 混沌工程简介
- **酷家乐技术架构现状**
  - 微服务化
  - 基础设施
  - 混沌工程实施条件
- 故障模拟
  - 故障场景抽象
  - 故障实现方案
  - 系统防御能力判定
  - 故障演练与混沌工程融合
- 混沌工程产品化落地&收益
- 总结和展望

酷家乐技术架构现状

微服务化

- 服务粒度细化
- 业务线内聚
- 前后端分离
- 中台化



酷家乐技术架构现状

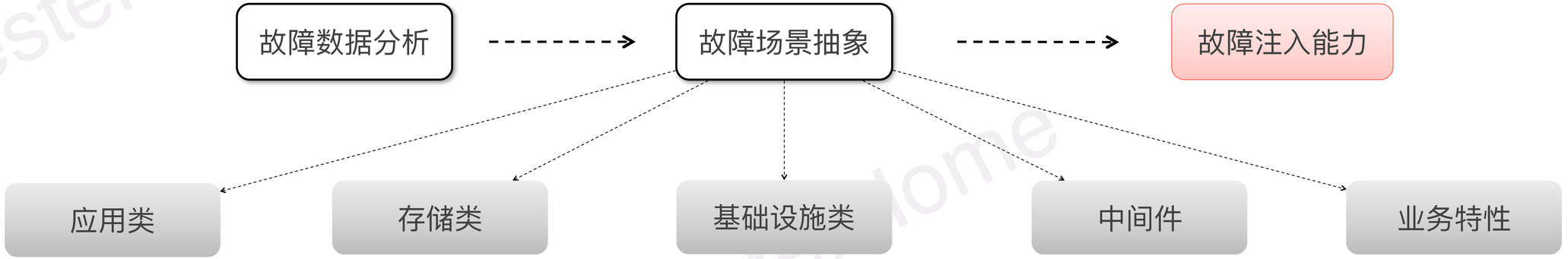
基础设施

- 微服务架构 & 服务治理
- 监控 & 警报
- 以应用为中心的Devops平台

酷家乐技术架构现状：混沌工程实施条件

- 混沌工程简介
- 酷家乐技术架构现状
  - 微服务化
  - 基础设施
  - 混沌工程实施条件
- **故障模拟**
  - 故障场景抽象
  - 故障实现方案
  - 系统防御能力判定
  - 故障演练与混沌工程融合
- 混沌工程产品化落地&收益
- 总结和展望

故障场景抽象



- 应用下线
- FullGC
- 接口限流
- 磁盘打满
- 内存占满
- ……

- 数据库宕机
- 数据库资源不足
- 连接池占满
- DB不可写
- 慢查询
- ……

- 高IO
- 高CPU
- 网络延迟
- 网络丢包
- DNS解析异常
- ……

真实可发生

故障模拟

虚拟机

- 固定ip
- 以单台虚拟机为最小攻击粒度
- 可通过shell脚本实现传统故障模拟

Dokcer

- VM中运行容器，host模式，直接使用宿主机的ip
- 以container作为最小攻击粒度
- 支持传统故障模拟

k8s

- Pod节点漂移
- 以Pod作为最小攻击粒度
- CPU打满、IO等传统故障影响面发生变化

- Shell脚本
- Docker API
- Kubectl

故障模拟

- Shell Example

CPU高负载

```
for i in `seq 1 $(cat /proc/cpuinfo | grep "physical id" | wc -l)`; do cat /dev/urandom | gzip -9>/dev/null & done
```

- 压缩随机数据并将结果发送到 /dev/null
- 回滚动作：ps aux | grep /dev/urandom | awk '{print $2}' | xargs sudo kill -9

Remote Service Unavailable

```
iptables -t filter -A OUTPUT -d [ip] -p tcp --dport [port] -j REJECT
```

- 影响面可控：仅对本服务调用的特定下游依赖有影响，不影响下游依赖对其他业务提供服务
- 回滚动作： iptables -D OUTPUT -d [ip] -j DROP

FullGC

```
jmap -histo:live <pid>
```

- 弊端：能触发FullGC，但内存表现和实际不同

网络情况模拟

```
tc qdisc add dev eth0 root netem delay <time>
```

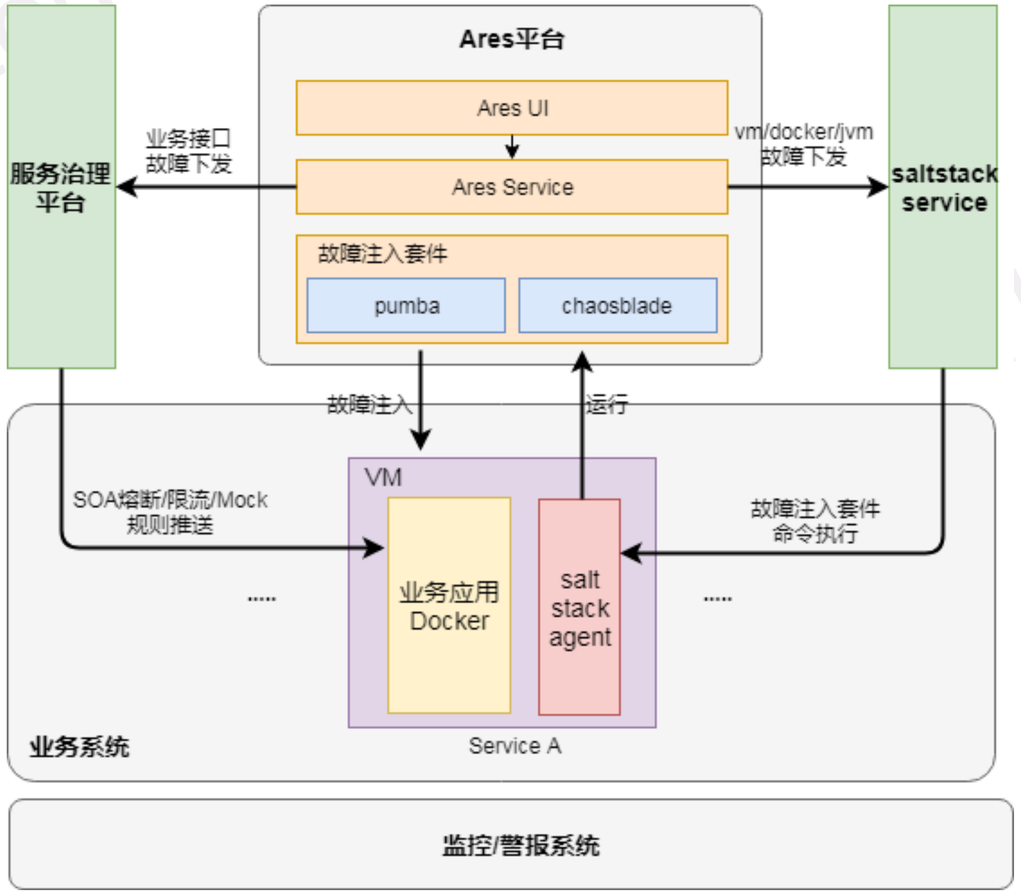
- 弊端：模拟整个网络情况，无法精确控制出口、入口、单一应用

故障注入方案

- 开源方案调研

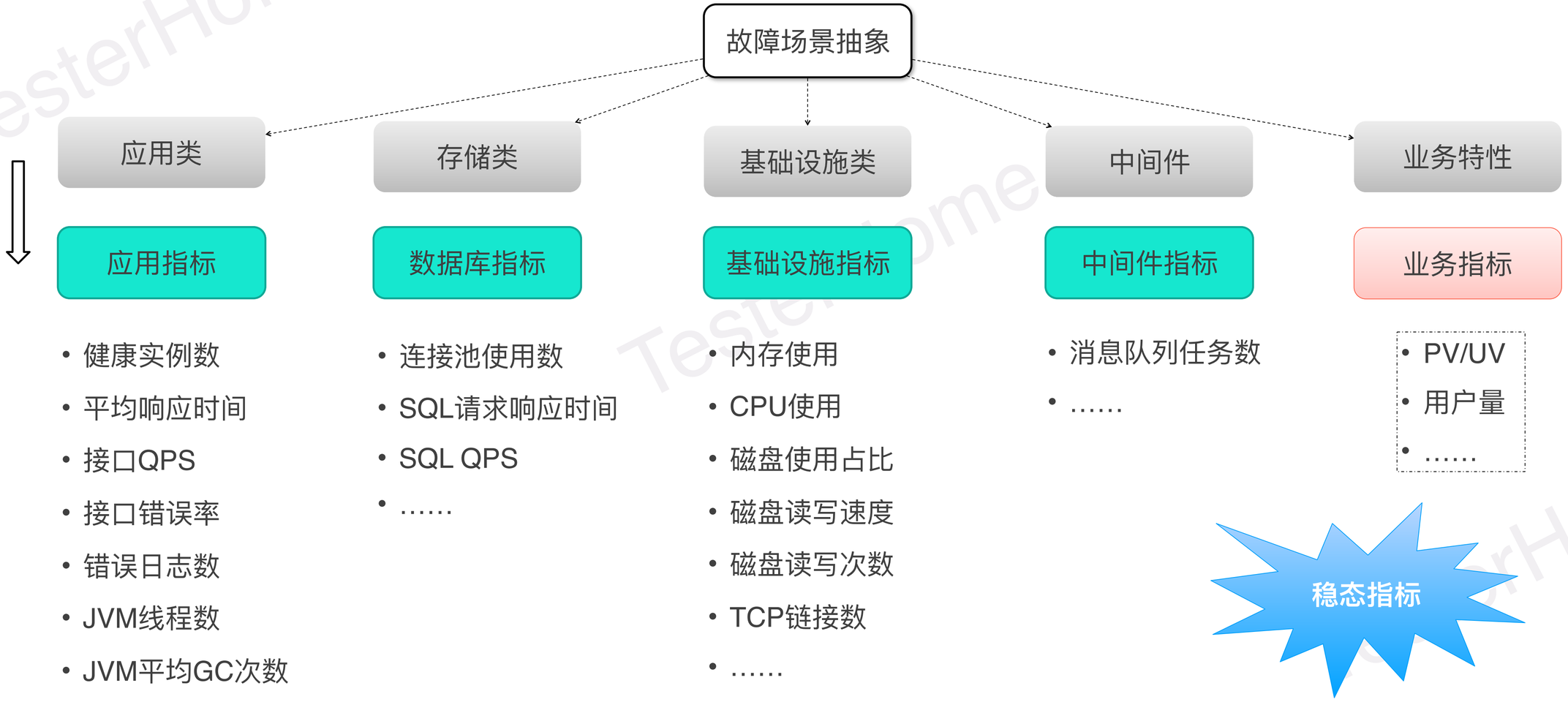
	开源方案	说明	优势	弊端
基于go	Chaos Monkey	随机中断虚拟机实例或容器		最新版本ChaosMonkey集成在Netflix的开源持续云交付平台Spinnaker中，因此使用过程依赖Spinnaker。
	kube-monkey	K8s集群，随机删除pod来验证服务部署的容错性		攻击方式单一
	Pumba	container级别攻击	• 随机kill、stop或者remove正在运行的container • 暂停运行中容器所有进程一个特定的时间 • 网络状况模拟，如：延迟、丢包、乱序、带宽受限等	
	Chaosblade	k8s	• 除了基础的CPU、disk、I/O、network外，还支持docker、dubbo、jvm的攻击 • 支持回滚	
基于python	chaostoolkit	开发工具包，提供Open API	支持不同系统的开发插件，包含k8s和spring boot	提供了某种实验方法的组织形式，具体的攻击方式需要自己通过脚本实现。
代码层面故障注入	Chaos Monkey for Spring Boot	通过配置定义攻击类型和watcher，不同的watcher会扫描app特定的annotation，基于AOP，实现不同的攻击。	• 配置简单、使用方便 • 实现了APP component级别的故障注入	需要app添加chaos-monkey-spring-boot相关依赖

产品化方案-Ares

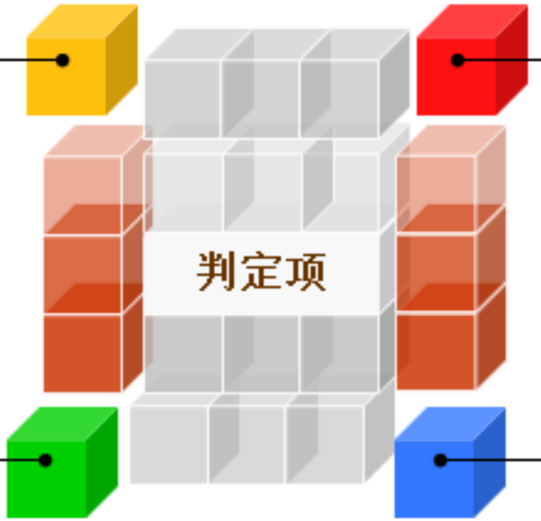


- 平台化调度
- 集成开源镜像（pumba、chaosblade…）
- 基于SaltStack集中化命令下发
- 集成服务治理能力

系统防御能力判定——判定指标



系统防御能力判定——判定项



▪ 故障生效判断

攻击发生后，对于本次攻击是否有效进行判定

▪故障修复能力判断

系统无自愈能力的情况下，从攻击发生到人员修复完成、业务恢复正常所需的时间。

▪ 应用自愈能力判断

对于某类攻击，系统是否在无人员干预的情况下，具备自愈能力。

▪ 故障影响面判断

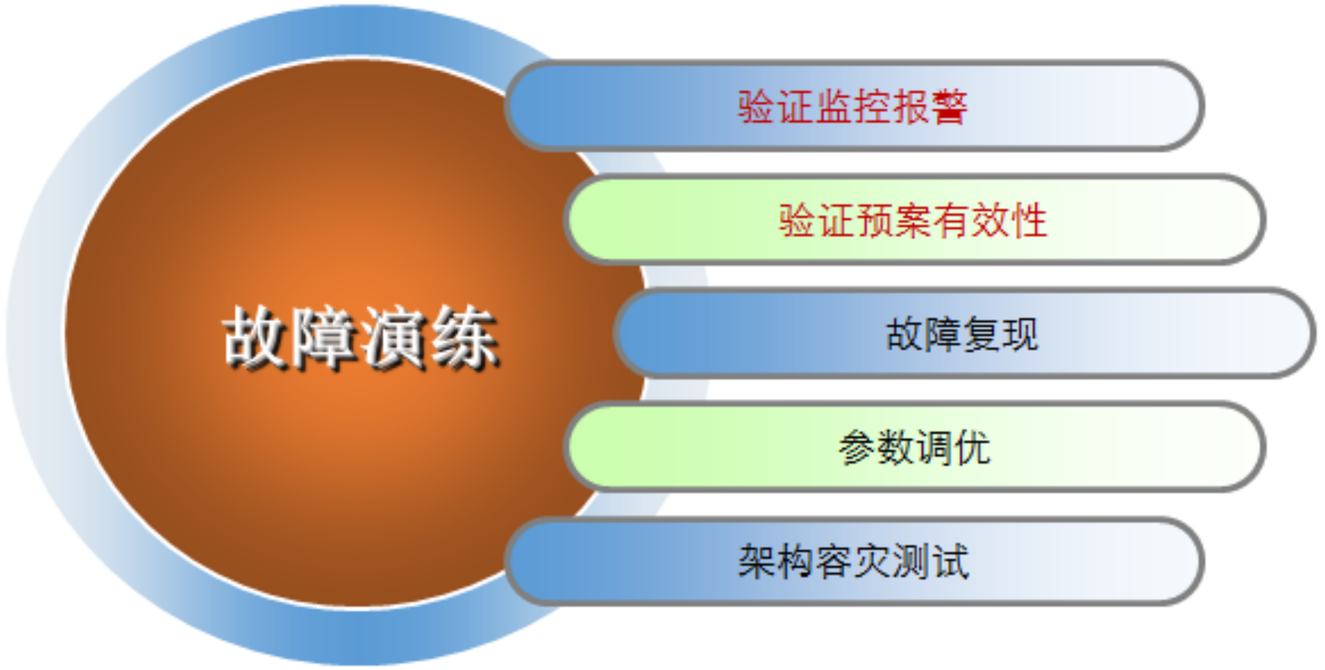
某应用受到某种攻击后，受影响面判断，包含上游依赖应用。

系统防御能力判定

- 指标获取：监控
- 判定结果获取：警报
- 补充判断：应用提供，如health check接口等

故障演练与混沌工程融合

把故障以场景化的方式沉淀，以可控成本在线上模拟故障，让系统和工程师平时有更多实战机会，加速系统、工具、流程、人员的进步。



故障演练与混沌工程融合

故障场景准备

eg:下游服务一台实例异常，共三台

假设期望

eg:上游业务调用正常，请求转发到其他实例

故障模拟

eg: docker stop [container name]

修复方案

eg: docker restart [container name]

混沌工程实验方案

故障场景准备

eg:下游服务一台实例异常，共三台

故障影响面分析

eg:上游业务调用无影响

故障模拟方案准备

eg: docker stop [container name]

故障预案准备

应急响应、回滚策略

故障演练实施流程

混沌工程！=故障演练

- 共性：
  - 故障场景、影响面分析、方案准备
- 差异：
  - 最小化爆破半径 vs 预案
  - 实验 vs 实践

- 混沌工程简介
- 酷家乐技术架构现状
  - 微服务化
  - 基础设施
  - 混沌工程实施条件
- 故障模拟
  - 故障场景抽象
  - 故障实现方案
  - 系统防御能力判定
  - 故障演练与混沌工程融合
- **混沌工程产品化落地&收益**
- 总结和展望

混沌工程产品落地&收益

Ares产品设计

- 混沌工程实施流程化
- 故障演练与混沌工程相结合
- 演练结果数据沉淀

混沌工程产品落地&收益

Ares混沌工程平台

攻击目标

- **攻击应用**
  - 业务应用
  - 中间件
  - 数据库
- **攻击范围**
  - 集群
  - 单机
  - pod

攻击方式

- **应用类**
- **基础设施类**
- **DB存储类**
- **容器类**
- **K8s类**

攻击视角

- **单应用**：最小攻击粒度
- **链路**：请求所经过的应用构成链路
- **业务**：多链路构成业务场景

攻击策略

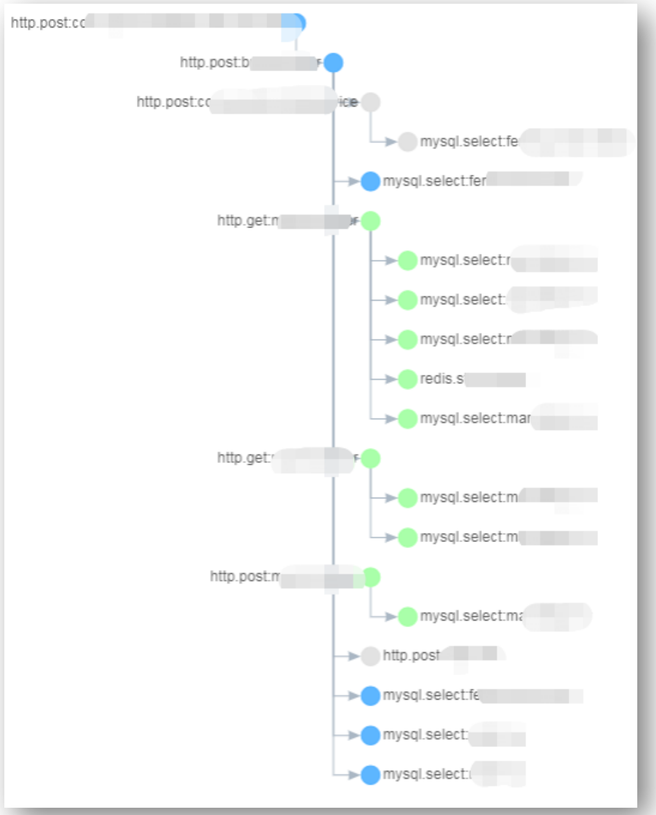
- **调度策略**
  - 手动
  - 自动随机
- **中断策略**
- **回滚策略**
- **应急预案**

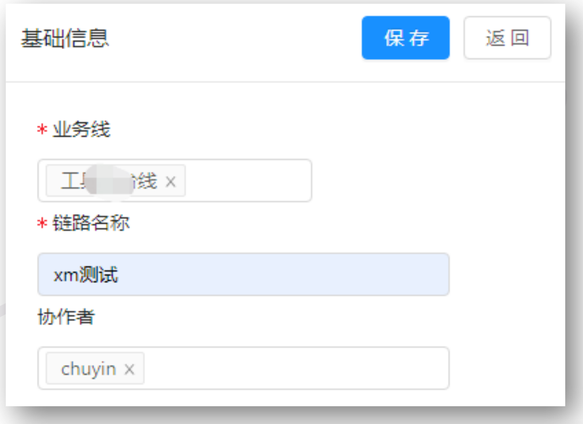
权限控制

- **应用owner**:对当前应用具有发布上线权限，直接具备应用各项攻击能力
- **协作者**：完成本次演练的其他应用owner
- **流程审批**：非应用owner攻击权限申请

混沌工程产品落地&收益

Ares产品demo：调用链攻击





- 链路树节点根据攻击权限点亮
- 多名协作者共同完成方案
- 攻击某一节点，观测上下游影响

攻击节点	攻击范围	攻击方式	状态	更新时间	操作
pdl. |service. ce	10.1. |de e-alpha:owner	基础设施类|CPU满负载	未开始	2019-05-06 18:00:52	攻击 | 修改 | 删除
pdl. |service	10.1. |sit- owner	基础设施类|指定服务网络阻断（iptables）	未开始	2019-05-06 18:00:52	攻击 | 修改 | 删除

混沌工程产品落地&收益：攻击试验1

目标：验证基础设施的完备性

攻击下，监控展示是否正确；相关警报是否发出。

攻击方案：暂停某应用

攻击结果

暂停服务期间：

- 监控：暂停服务期间，监控看板能清晰展示数据截断
- 警报：无警报发出

服务恢复后：

- 监控：监控看板无法识别此次异常波动

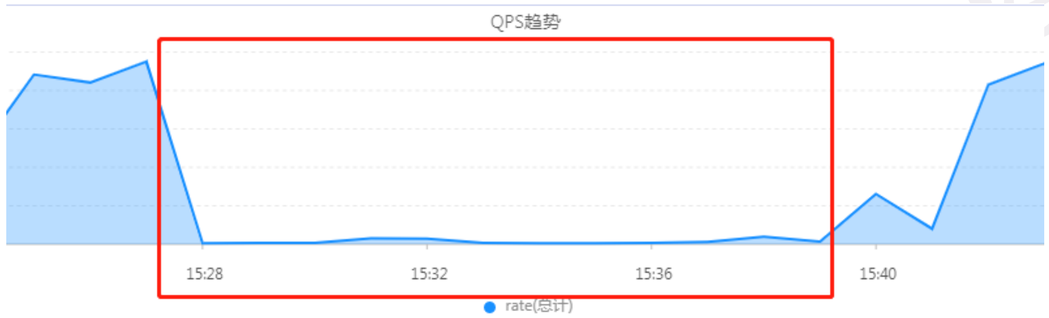
混沌工程产品落地&收益：攻击试验2

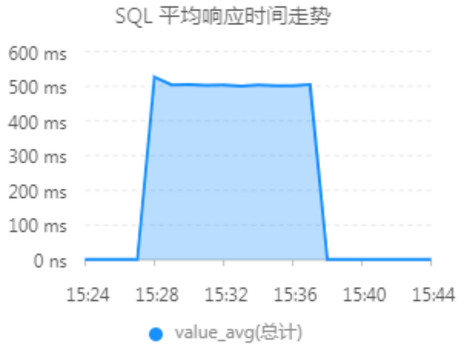
目标：验证应用在攻击下的工作情况

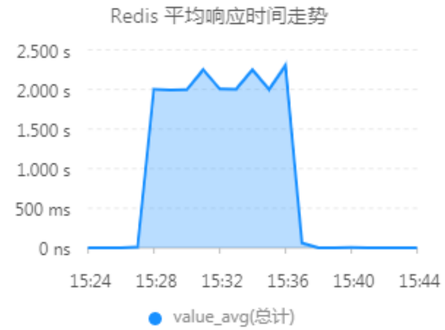
攻击方案：所有请求增加网络出口延迟500ms，攻击时长10min。选择某一接口请求进行观测。

观测项：应用QPS、接口&数据库平均响应时间、应用告警、上层调用业务告警

攻击结果







- 攻击期间，QPS显著降低
- 某接口SQL平均请求响应时间增加500ms，说明该接口有一次数据库查询请求；
- 某接口Redis平均请求响应时间增加2s。经review代码，redis client连接到释放执行了4次 操作，为正常表现。

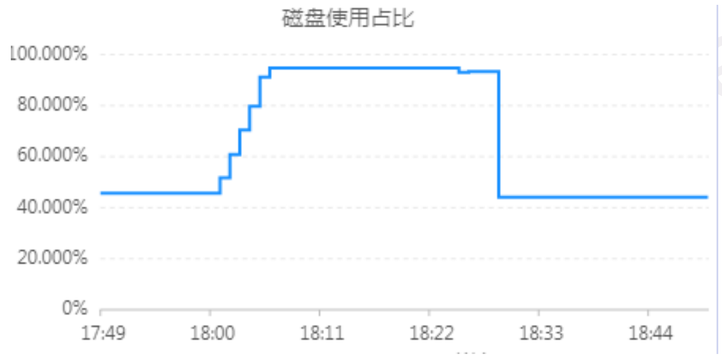
混沌工程产品落地&收益：攻击试验3

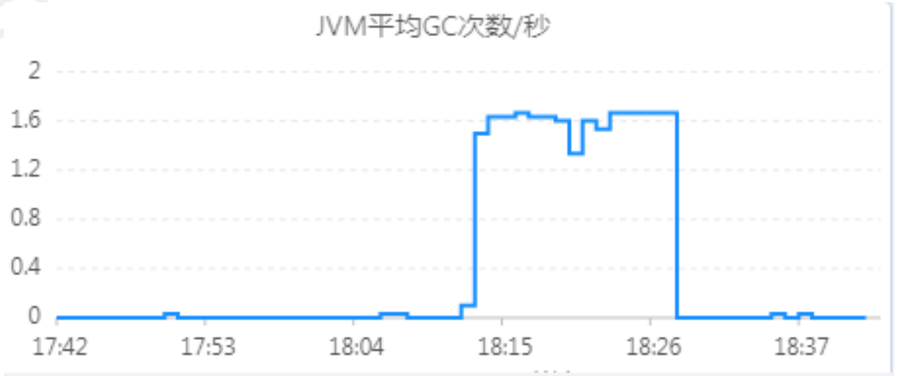
目标：验证应用在攻击下的工作情况

攻击方案：磁盘占满

观测项：应用QPS、CPU、磁盘使用占比等

攻击结果：引起非相关指标异常





- 攻击期间，磁盘使用占比达到100%
- JVM指标异常。

混沌工程产品落地&收益

可见收益

- 完善基础设施：监控&警报
- 挖掘漏洞：优化应用，提前排除故障风险
- 完善预案：提前掌握应用在不同故障下的指标状态，在故障发生时迅速判断，迅速处理
- 验证服务间强弱依赖
- 深入了解应用本身

⇩

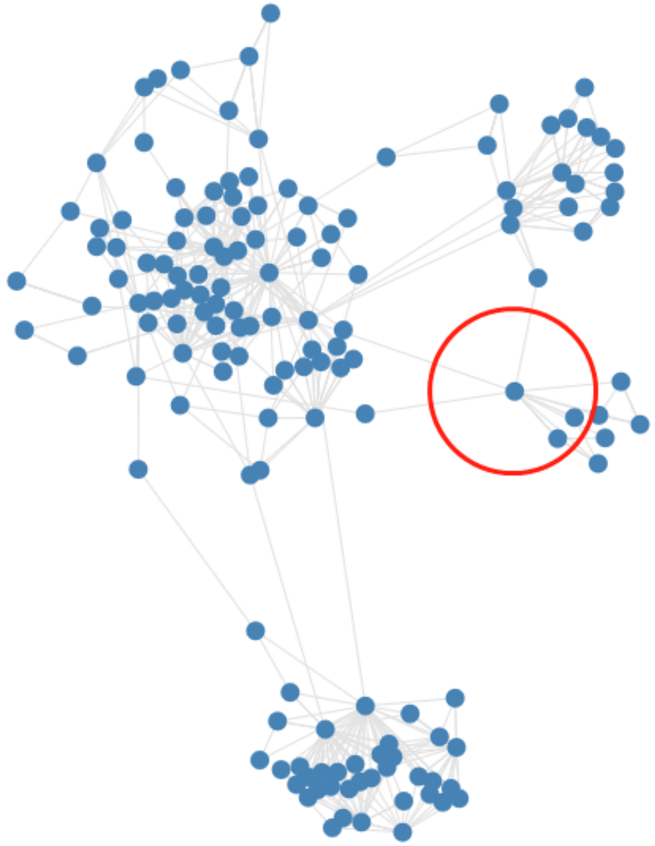
稳定性、业务保障

- 混沌工程简介
- 酷家乐技术架构现状
  - 微服务化
  - 基础设施
  - 混沌工程实施条件
- 故障模拟
  - 故障场景抽象
  - 故障实现方案
  - 系统防御能力判定
  - 故障演练与混沌工程融合
- 混沌工程产品化落地&收益
- 总结和展望

总结和展望

爆破半径分析

- 依赖分析
- 流量分析



总结和展望

产品完善

- 脆弱点视图
- 历史故障改进措施的覆盖率
- 场景预案
- 自愈能力判断
- ……

谢谢

THANKS

加入我们

加入我们

联系我：xiaomian@qunhemail.com