

监控优化方案

- 背景
- 优化方案
- 规划

1 背景

系统服务日益增多，业务量上涨，现有监控系统已无法及时解决现有问题。

1.1 外患

前端在群里轰炸

- xx系统又没反应了
- xx系统又巨慢了
- 征信又没法查了

1.2 内忧

- 帮忙查一下日志
- xx定时任务执行了吗
- 日志大量异常
- 前天怎么没有备份?
- xx服务部署在哪个机器上? (位置复杂, 阿里云 * 10, 腾讯云 * 70, IDC * 18, 上海机房)
- 我老收到一大堆告警信息

1.3 架构

- 阿里云监控 + 腾讯云监控 + Zabbix监控
- 监控指标：CPU、内存、磁盘、网络、端口

1.4 痛点

- 快速发现并恢复问题
- 快速定位问题

2 优化方案

- 规范优化
- 管理优化
- 功能优化
- 架构优化

2.1 规范优化

- 《零售系统SLA说明》
- 《CACMP平台SLA说明》

规则定义（如多少秒算慢），响应时间，处理时效

2.2 管理优化

- 资源统一管理
- 服务平台

2.2.1 资源统一管理

- 服务器 (√)
- 服务
- 端口
- 黑白名单

2.2.2 服务平台 (Service Desk)

- 统一问题入口
- SLA统计

2.3 功能优化

- 基础设施层监控
- 业务应用层监控
- 中间件层监控
- 日志监控
- 调用链监控
- 告警规则
- 安全监控与审计
- 前端监控

2.3.1 基础设施层监控

- 物理机 (√)
- 虚拟机
- 容器

2.3.3 业务应用层监控

- 应用系统
- 微服务

可用性、异常数量、吞吐量、响应时间、当前执行任务数、资源占用率、请求量、日志大小、性能、线程数、服务调用次数、访问量、服务可用性

2.3.3 中间件层监控

- 数据库
- 缓存
- 消息队列
- 配置中心

2.3.4 日志监控

- 统一格式规范
- 接口日志
- 定时任务日志

2.3.5 告警规则

- 阈值报警策略
- 缺失报警策略
- 趋势型报警策略

2.3.6 前端监控

- 页面打开速度（测速）
- 页面稳定性（JS Error）
- 外部服务调用成功率（API）

2.4 架构优化

- 指标监控架构
- 日志监控架构

2.4.1 指标监控架构

拉取 Kafka -> Prometheus -> Grafana

- Tomcat: JMX, Spring metrics
- Spring Boot: Spring actuator

2.4.2 日志监控架构

Filebeat -> Kafka -> Elasticsearch -> Kibana

- 聚合
- 过滤
- 分析

3 规划

- 阶段性计划
- 资源需求

3.1 阶段性计划

- 一阶段
- 二阶段
- 三阶段
- 未来远景

3.1.1 一阶段——工具型平台

- 日志监控（高）
- 业务应用层监控（中）
- 中间件层监控（低）

3.1.2 二阶段——一体化平台

- 调用链监控（高）
- 架构优化（中）

3.1.3 三阶段

- 埋点/探针（高）
- 告警智能化（中）

3.1.4 未来远景

- 统一监控平台（高）
- 服务平台（中）
- 报表&报告（中）
- 前端监控（低）
- ChatOps（低）

3.2 资源需求

- 运维: Zabbix/Prometheus/日志解析 * 1
- 开发: 埋点/探针/日志分析 * 1
- DBA: Prometheus/HBase/Elasticsearch * 1