如何着手打造公司的APM系统?

主讲：鲁班

时间：2018/3/4 8:20

地址：腾讯课堂–图灵学院

1、整体认知APM系统

2、现有监控系统解决方案对比

3、APM系统架构实战

讲师介绍：

一、整体认知APM系统

提问：

同学们是否有遇到过这样的场景：业务运营人员反映、会员很多功能访问很慢。你做为会员系统的负责人，接到反馈后做了如下事情：

1. 检查线上11台机器各硬件负载指标均正常
2. 查询线上11台会员系统日志，出现大量请求Timeout
3. 检查3台mySql数据库，访问非常慢，并且连接已占满
4. 进一步查明，大量连接都在执行一个会员注册的Sql语句，找到真凶。

大家预估一下以上4个过程如果不借助任何工具，都以人肉方式进行，需要多长时间?

【学员发散式回答问题】

那以上步骤怎么去优化呢?

第1：采用Zabbix 对线上机器各性能指标进行集中监控。

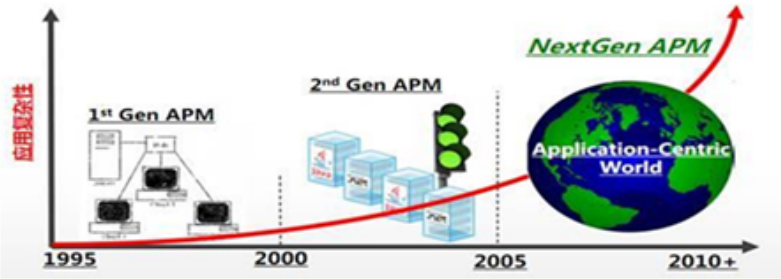
第2：使用ELK对线上日志集中存储查询。
第3：使用Anemometer 集中监控mysql 语句

以上3个功能合起来就是我们的APM系统，它对公司的意义，还需要我在啰嗦吗？当然一个完整的不此之些，APM还可以进行代码级定位。什么叫代码级定位？
我们继续上述例子，上述问题解决了么？并没有，因为这个Sql语句不是慢查询，且这时的会员注册流量并不大，原因又是什么呢？只能翻开代码找。这个时间可能比前面四个步骤花的时间还要长。而代码级定位可直接定位至问题所在源码。

APM 简介

APM全称*Application Performance Management(应用性能管理),APM致力于监控和管理应用软件性能和可用性。通过监测和诊断复杂应用程序的性能问题，来保证软件应用程序的良好运行*

APM发展史



APM 发展的三个阶段
第一阶段: 以网络为中心，网速即应用速度，APM 主要指基础组件的性能监控
第二阶段: 以 IT 部件/组件为中心，监控围绕着部件/组件健康状态，基础设施可用性，伴随 IT 基础架构组件发布
第三阶段: 以 IT 应用核心和业务交易为中心，IT 架构高度复杂性，专注 IT 应用，面向用户，提供应用生命周期管理

各互联网公司APM解决方案

1. **google** :dapper
2. **twitter:** zipkin （开源）
3. **淘宝：**鹰眼
4. **京东：**hydra （开源）
5. **大众点评：**CAT （开源）
6. **小米：**open-falcon （开源）

二、现有监控系统解决方案对比

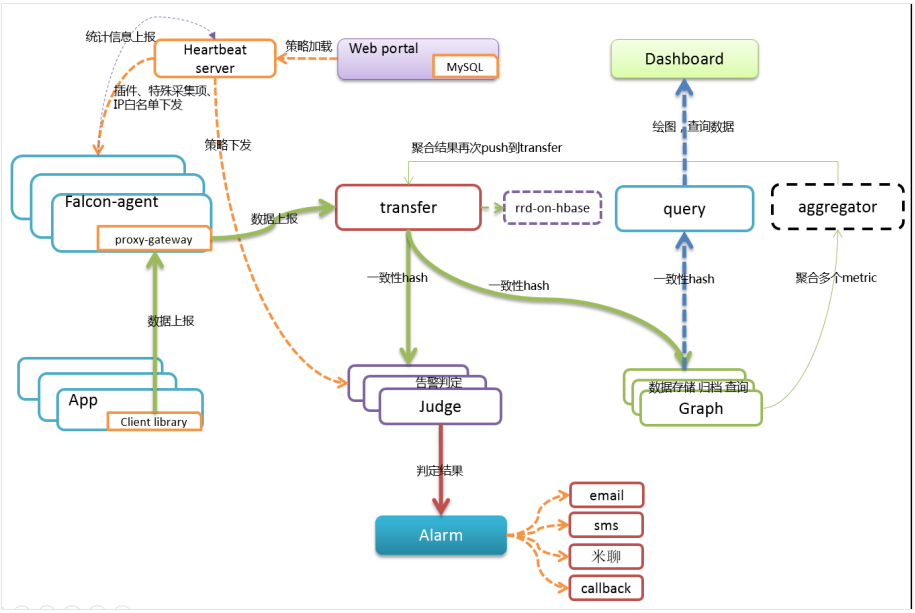
Zabbix 与open-falcon
主工功能与特点:
– CPU负荷
– 内存使用
– 磁盘使用
– 网络状况
– 端口监视

– 日志监视

不同点：

	Zabbix	open-falcon
用户群	85%泛化互联网企业	包括不限于：美团、金山云、快网、宜信、七牛、赶集、滴滴、金山办公、爱奇艺、一点资讯、开心网、借贷宝、百度、迅雷等等
优点	1、安装部署简单2、多数据集采集灵活集成3、用户群体大，使用时间长	1、数据采集免配置2、吞吐量高，支持单周期亿次数据采集、告警判定3、中文文档齐全4、水平扩展多IDC支持5、持策略模板、模板继承和覆盖、多种告警方式、支持callback调用
缺点	1、易用性不高、而且很多定制化监控实现起来很麻烦。2、应对超大流量有点力不从心	1、部署复杂，各个组件太零散

Open-falcon 架构图



单纯服务节点监控与报警，以上两个开源软件都在一定程度上可以支持，但做为APM还缺失了日志集中搜索与查询的功能。这就需要另外一套技术来补充。

Elastic Stack

解决方案对比：

解决方案	优势	劣势	适应场景
ELK	1） 文档资料齐全，社区活跃2） 关键字搜索强大 3） 不需要二次开发 4） Kibana统计功能强大	1）不支持监控报警2)服务节点状态无法获取3)无法统计应用内部组件性能4)Kibana没有汉化，不符合国人使用习惯	日志集中归档异常查询
open-falcon	1） 文档资料齐全2） 很多大型公司使用（美团、金山云、京东金融、赶集）3） 基础组件监控功能完善	1） 功能局限性较大，监控应用内部性能比较麻烦2） 无法查询异常日志3） 不支持搜索	运维人员的宝贝，针对性监控Mysql、Redis、MQ、Solr、及硬件监控
CAT （开源）	1） 支持监控程序内部 2） 支持分布示调用链监控	1）文档较少2）通用性不强3）接入成本较高，需要在程序内部引入Agent4）搜索功能不强	分布式服务调用链追踪微服务治理，异常排错

注：

open-falcon 文档中心:http://book.open-falcon.org/zh/intro/index.html

Elastic 官网：https://www.elastic.co/cn/products

以上方案的缺陷：无法做到灵活的监控应用内部

三、APM系统架构实战

架构目标：

1. 功能性需求
   a. 系统和服务指标收集
   b. 日志集中存储与搜索
   c. 应用内部监控
   d. 图表展示
   e. 监控指标报警
2. 非功能性需求
   a. 高性能：支持TB级别监控日志收集
   b. 高可用：整个系统无核心单点
   c. 强伸缩：能根据公司规模灵活调整

架构方案描述：

整套架构在ELK基础上进行扩展升级，基主要分为日志信息**采集**、**传输**、**存储**与**展现**四部分。

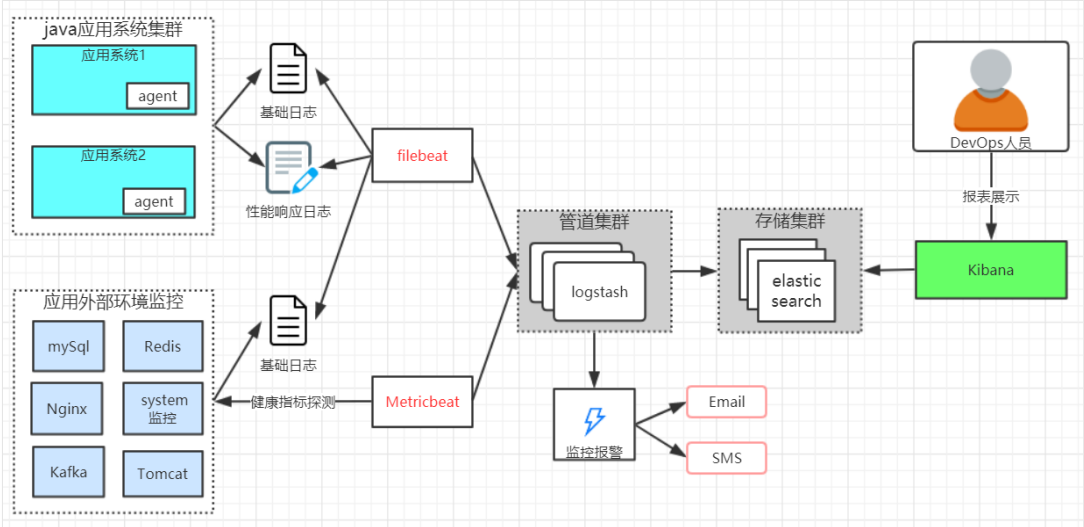
采集：javaagent 采集应用内部性能指标并输出至日志、FileBeat采集文本日志、Metricbeat采集系统性能指标。

传输：beats、logstash、kafka

存储：Elasticsearch

展现：Kibana

架构图：



应用内部性能采集说明：

基于字节码插桩技术监控应用内部性能指标如：Http响应时间、服务响应、异常、SQL语句、外部接口响应。该方式特点是：即可以精准监控应用内部，又不会对应用造成侵入从而简化上线推广成本。

注：字节码插桩技术指使用javaagent与 javassist动态重构Class字节已插入监控指令。

应用外部环境采集说明：

基于metricbeat 主动采集系统信息如： CPU 使用率、内存、文件系统、磁盘 IO 和网络 IO 统计数据，以及获得如同系统上 top 命令类似的各个进程的统计数据。此外基于 metricbeat 中提供的插件去主动采集 mySql、Redis、Nginx等状态信息。

监控报警说明：

如果报警条件比较简单，可以直接基于logstash当中logstash-output-http 插件输送数据流至自实现报警中心实现报警。如果报警条件复杂则可基于logstash-output-zabbix输送数据流至 zabbix系统触发报警。

问题1：日志太多，磁盘存储不够用怎么办?

1、设定日志过期时间，写脚本定期清理Elasticsearch 当中过期数据。

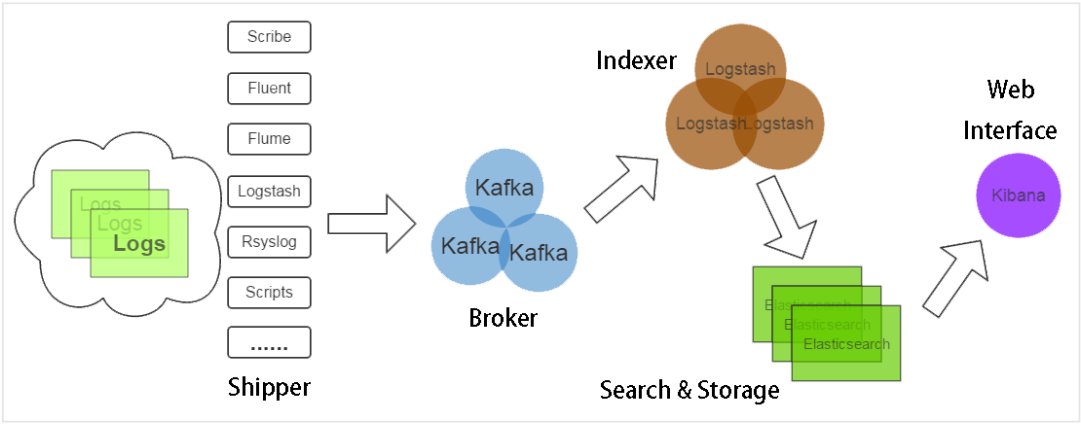
2、在logstash中设置过滤器，过滤debug 和info 日志。

问题2：logstash 和Elasticsearch 是分开部署的么，不想要这么复杂的架构怎么办?

可以直接去掉logstash ，数据直接从beats 发送至logstash

问题3：日志信息吞吐量太大怎么办?

这种情况只能在整个传输管道中间加入一层 Broker作缓冲Redis或Kafak。当然这样做将会使整个系统变得更复杂，众多节点自身的健康状况维护也变得困难。具体看下图。



（管道横向扩充增大吞吐能力）