GOPS2017
Beijing

GOPS

GOPS 全球运维大会 2017 北京站

指导单位：数据中心联盟 Data Center Alliance　开放运维联盟 OOPSA Open OPS Alliance

主办单位：高效运维社区 GreatOPS Community　DevOps 时代　CIO时代

大会时间：7月28-29日

大会地点：北京朝阳悠唐皇冠假日酒店

GOPS2017
Beijing

Pinterest 的监控系统

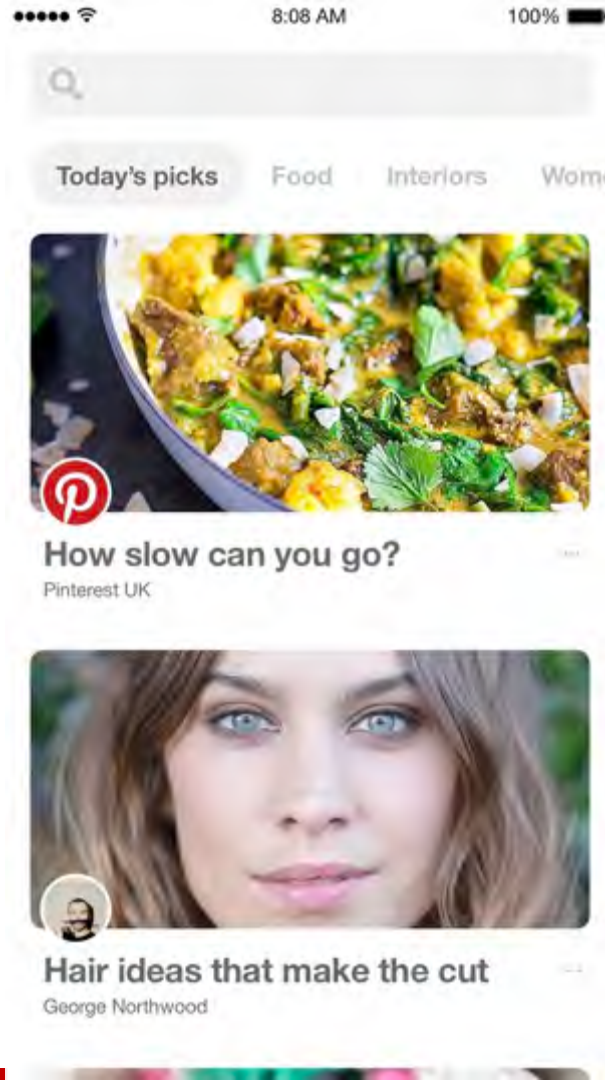
孟晓桥， Pinterest监控部经理

目录

Pinterest

世界上最大的图片探索引擎

- ✧ 1亿9千万月活跃用户
- ✧ 1000亿个图片收藏
- ✧ 20亿个收藏夹
- ✧ 每月20亿搜索



后台架构和运维

GOPS2017 Beijing

后台架构

✧搭建在三万多台亚马逊云的虚拟机上

✧少量亚马逊云服务 + 一百多个自开发的微服务 + 自开发的数据存储平台

✧搜索平台 + 内容平台+ 广告平台

运维

✧专职SRE团队负责第一线运维

✧少量SRE嵌入产品部门充当开发和运维间的桥梁

✧运维指标： 可靠性 > 99.9%

✧运维使用来自于监控组的各种实时监控系统

目录

成为一个10倍影响力工程师的诀窍，就是帮助10个工程师更好地完成他(她)们的工作

- 引用于互联网

组成

GOPS2017
Beijing

基于时序数据的监控和警报

- 实时了解所有系统和应用指标
- 实时报警

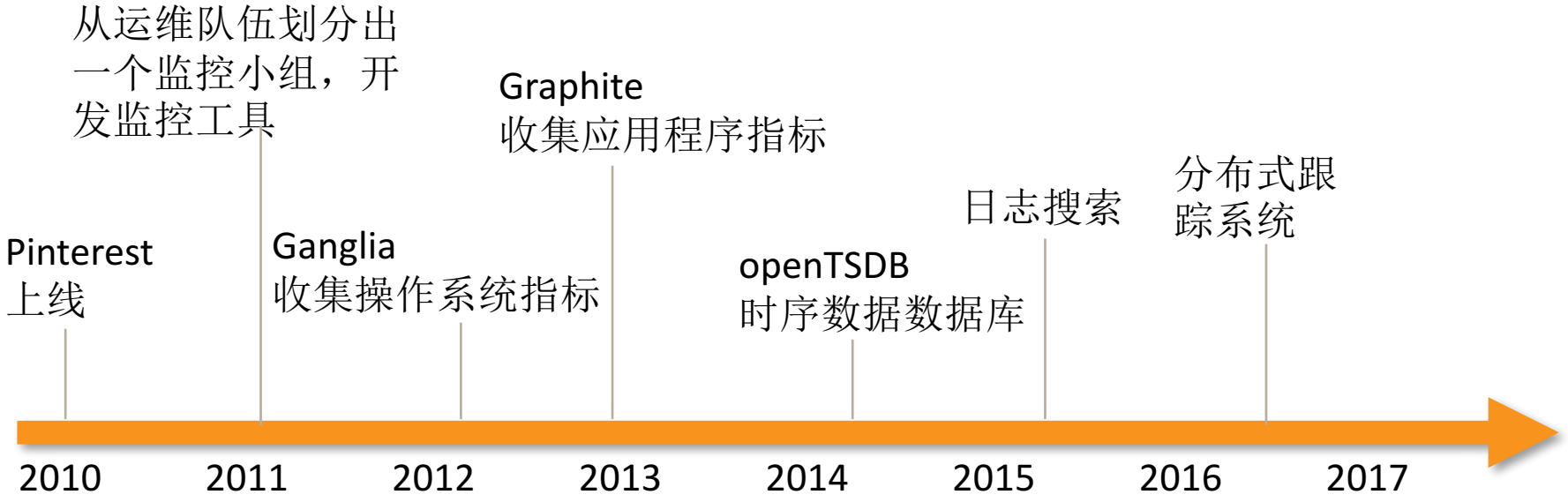
日志搜索

- 实时了解所有重要系统输出的日志
- 实时报警

分布式跟踪系统

- 理解用户请求对后台服务的调用
- 标示对用户延迟影响最大的瓶颈服务

衍变


从运维队伍划分出一个监控小组，开发监控工具
Graphite
收集应用程序指标
日志搜索
分布式跟踪系统
Pinterest
上线
Ganglia
收集操作系统指标
openTSDB
时序数据数据库
2010
2011
2012
2013
2014
2015
2016
2017

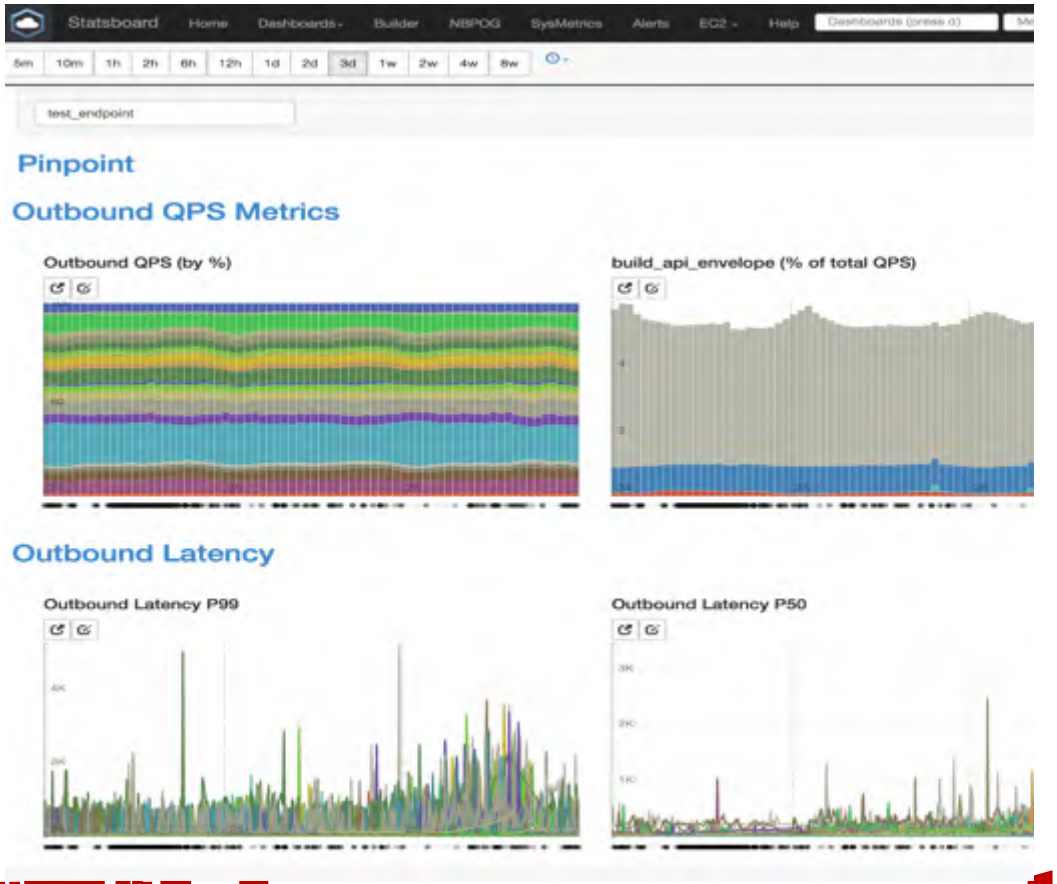
目录

基于时序数据的监控系统

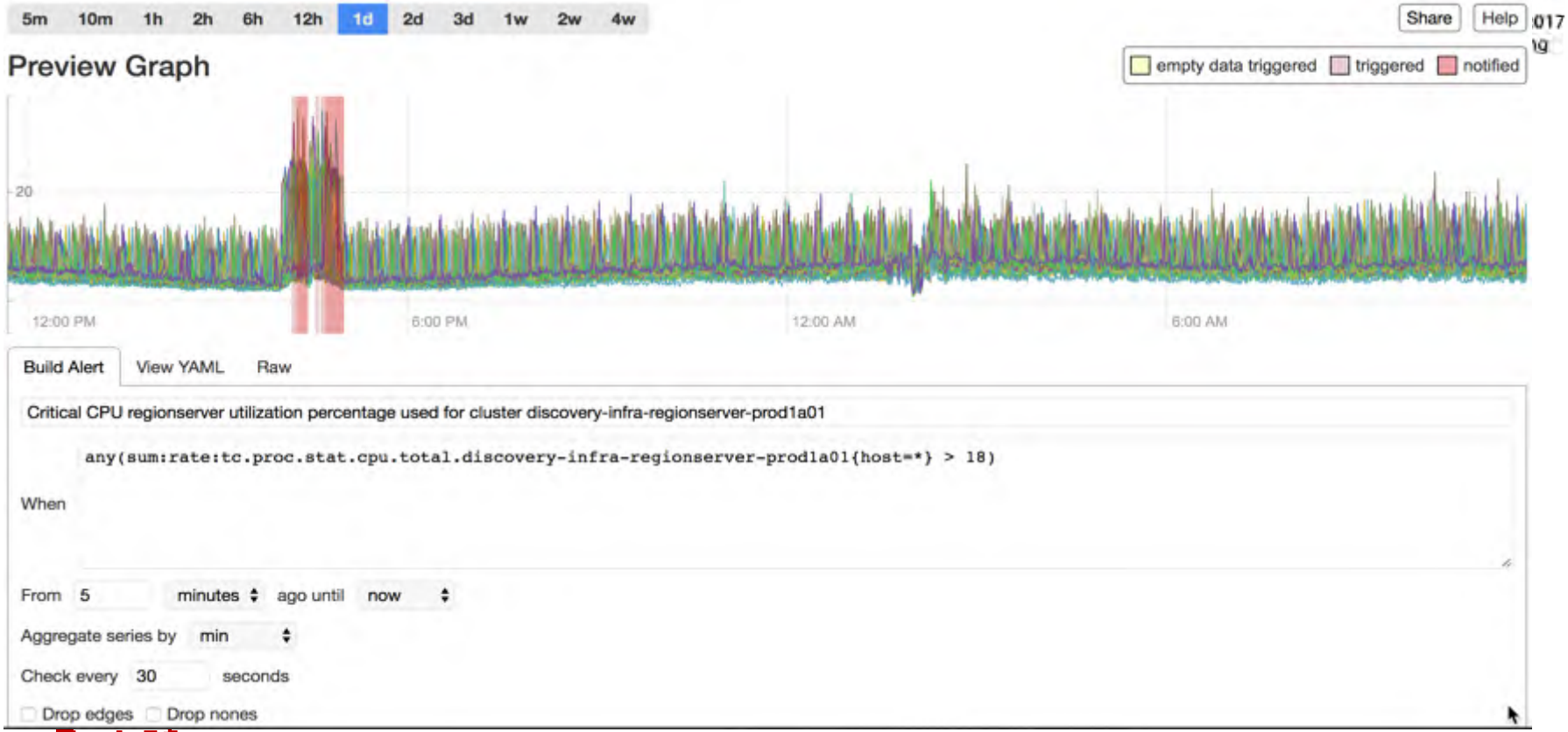
- 架构
  - 基于Kafka和openTSDB/Hbase。其他部分均自开发
- 规模
  - 每秒250万个数据点
  - 每秒3万5千个查询请求（90% 是用于报警系统）

应用场景一：数据搜索和可视化

应用场景二：仪表板


Statsboard
Home
Dashboards
Builder
NBPOG
SysMetrics
Alerts
EC2
Help
5m
10m
1h
2h
6h
12h
1d
2d
3d
1w
2w
4w
8w
test_endpoint
Pinpoint
Outbound QPS Metrics
Outbound QPS (by %)
build_api_envelope (% of total QPS)
Outbound Latency
Outbound Latency P99
Outbound Latency P50

应用场景三：警报


5m 10m 1h 2h 6h 12h 1d 2d 3d 1w 2w 4w
Share
Help
Preview Graph
empty data triggered
triggered
notified
20
12:00 PM
6:00 PM
12:00 AM
6:00 AM
Build Alert
View YAML
Raw
Critical CPU regionserver utilization percentage used for cluster discovery-infra-regionserver-prod1a01
any(sum:rate:tc.proc.stat.cpu.total.discovery-infra-regionserver-prod1a01{host=*} > 18)
When
From 5 minutes ago until now
Aggregate series by min
Check every 30 seconds
Drop edges
Drop nones

痛点和对策

GOPS2017
Beijing

数据量大
每天～100Terabyte数据

- 对长期数据降维，调低分辨率
- 冷数据挪到成本低的存储上

可靠性要高
> 99.9%

- 多个监控系统，交叉监控
- 主动和被动式监控相结合

查询速度要快
打开图表的延迟在1秒内

- 多维度数据分片
  - 按数据类型
  - 按时间
- 热数据置于内存（开发中）
  - 参见Facebook的Gorilla

日志搜索

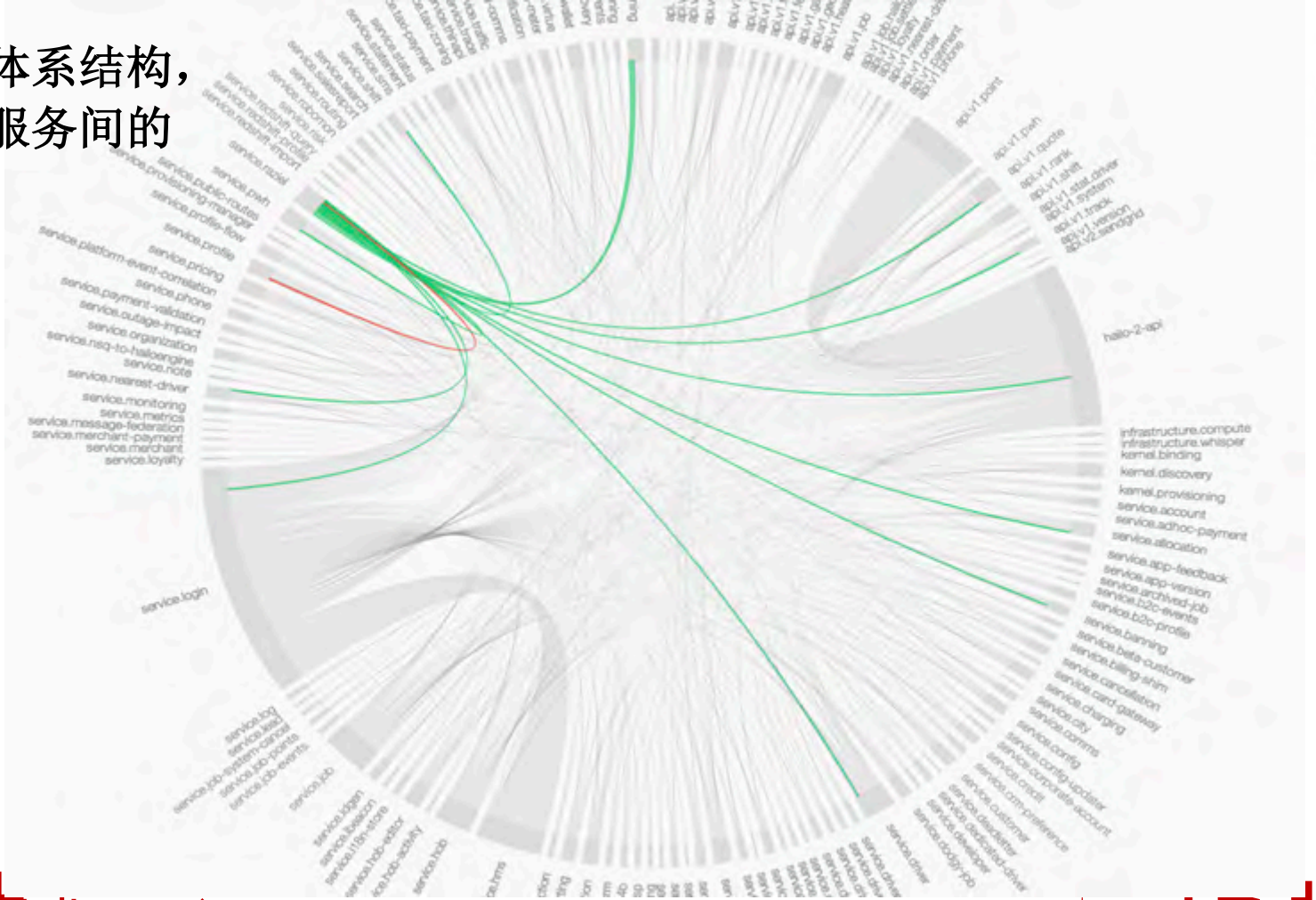
- 架构
  - 日志的索引，存储。其他部分均自开发
- 规模
  - 每秒采集250万个数据点
  - 每秒3万5千个查询请求（90% 是用于报警系统）

日志的标准化

- 对每个编程语言，定制输出日志的函数库
  - 统一格式为JSON
  - 强制输出丰富的上下文信息

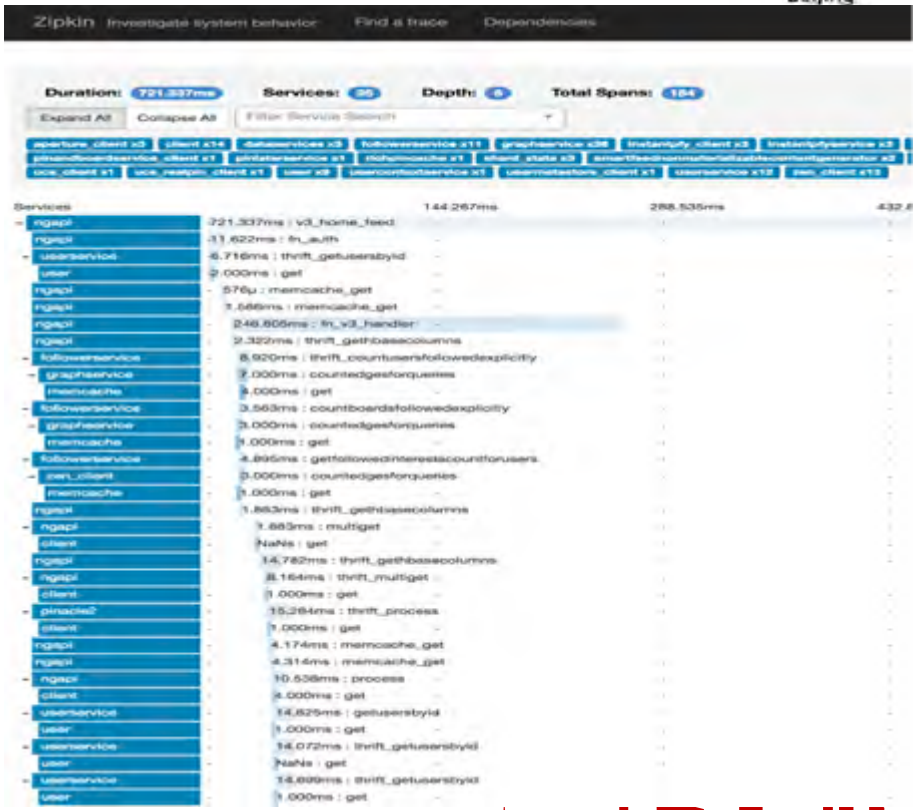
id: string // mandatory; globally unique to identify a log record. It can be a structure with timestamp.
timestamp: date // mandatory; when the log records are created.
service: string // uniquely identify a service which generates the log record.
project: string // or organization so that we can teams can be associated with
host: string // name of the host creating the log
env: string // deployment env
stage: string // deployment stage
version: string // git commit number
tags: { } // labels consisting of key value pair assuming both key and values are strings for expansion by clients. This can include locale or organization and so on.
payload_encoding: string // can be either string, JSON, base64 encoded binary format and so on. string by default
payload : // mandatory

微服务的体系结构，怎样监测服务间的调用？

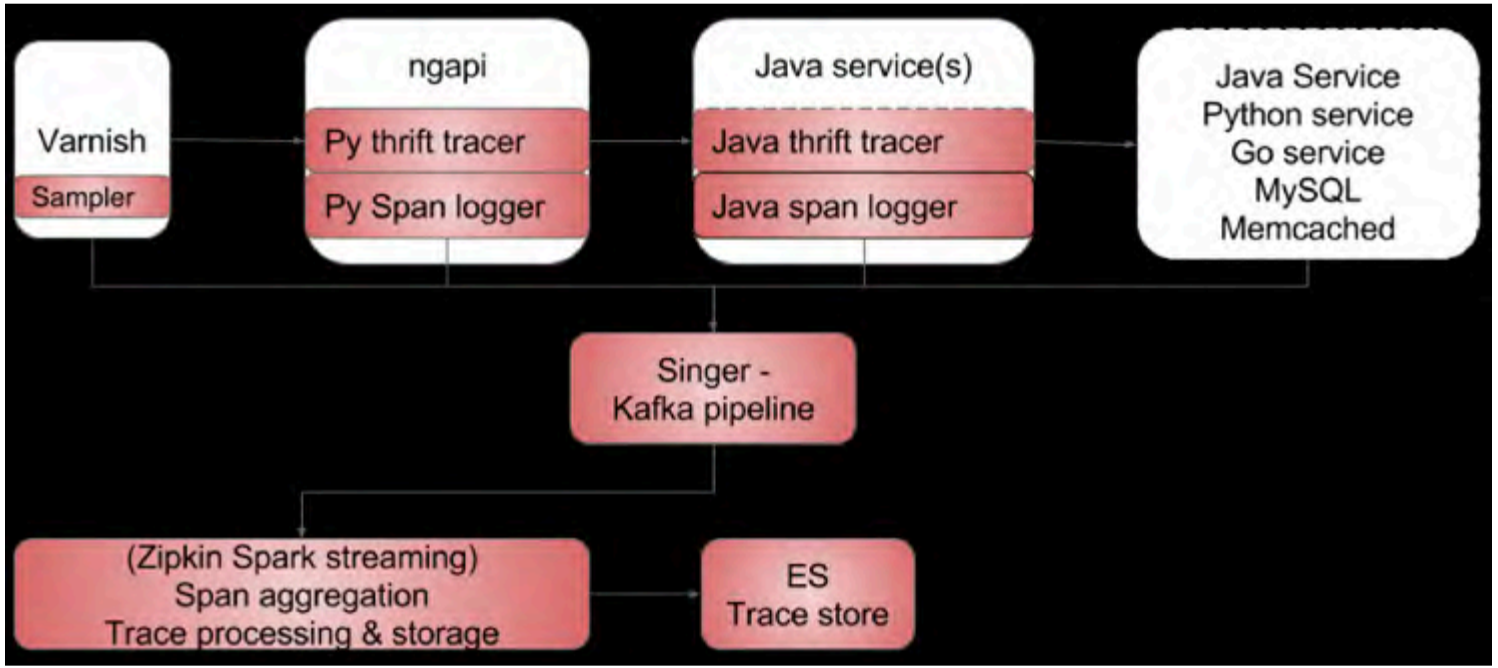


分布式跟踪系统

- 标准化的数据结构，可以描述用户请求在分布式系统中的所有事件



架构 TRACE PROCESSING AND STORAGE



我们已经开源了后台的Spark处理逻辑:

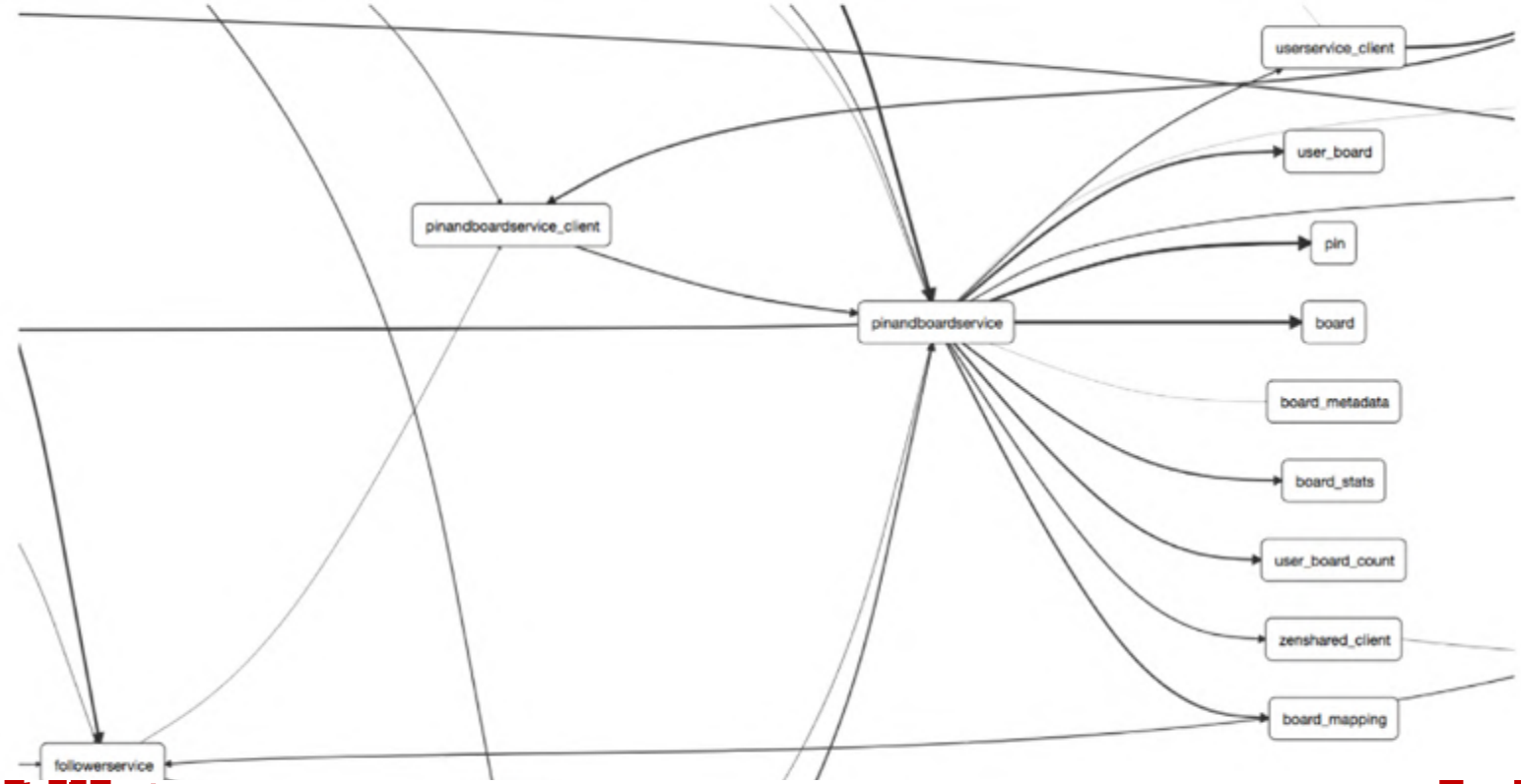
github.com/openzipkin/zipkin-sparkstreaming

用途实例

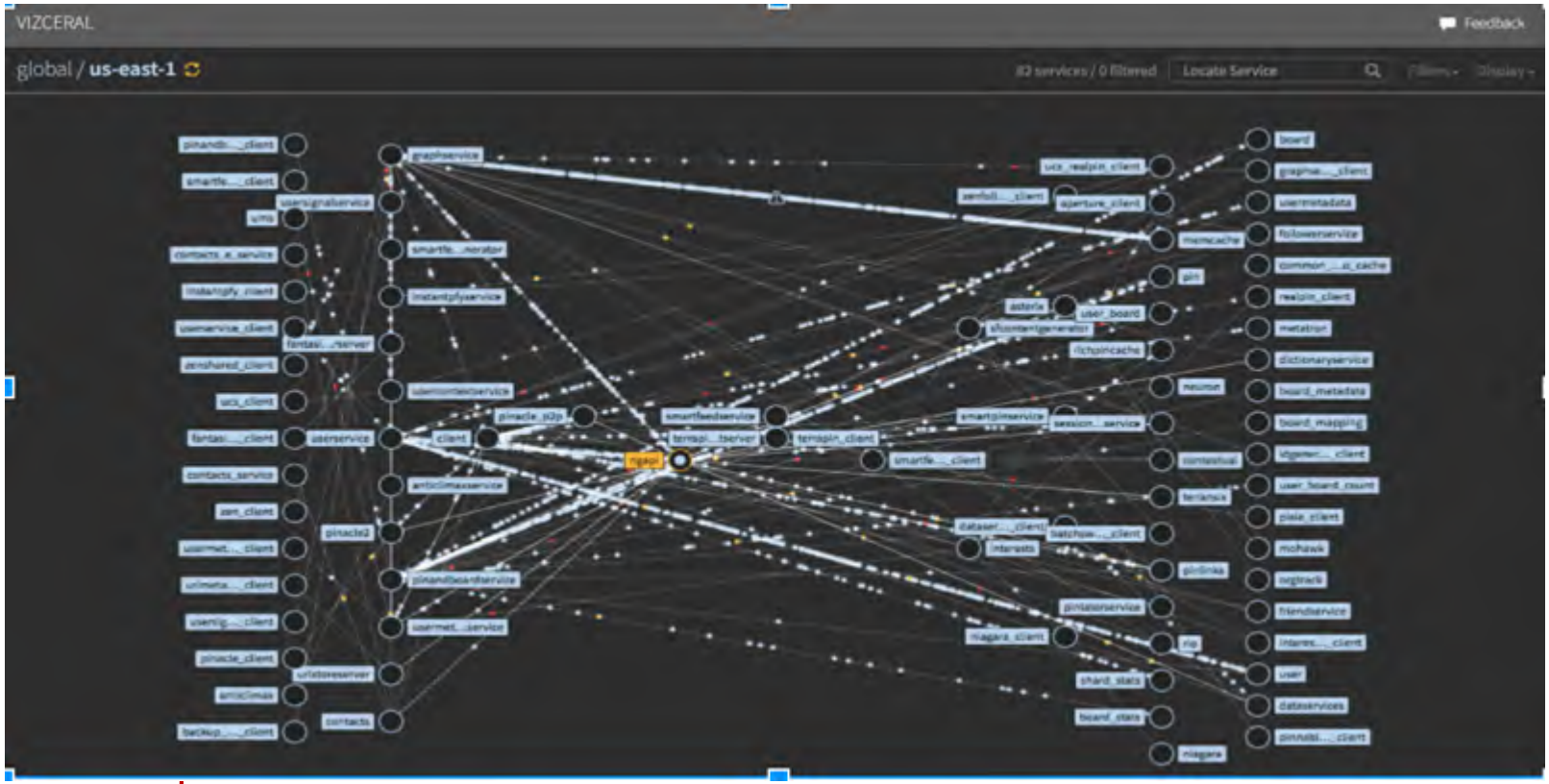
GOPS2017
Beijing

- 理解服务之间的调用关系
- 已成为整个公司所有与性能有关的数据的格式标准
- 发现最影响用户请求的瓶颈服务
- 帮助发现代码里的错误。 例如： 同一个API被调用多次
- 建立模型来计算服务对资源消耗

服务间的调用关系


userservice_client
user_board
pinandboardservice_client
pin
pinandboardservice
board
board_metadata
board_stats
user_board_count
zenshared_client
board_mapping
followerservice

基于服务调用关系的一站式仪表板（开发中）



目录

最大的挑战来自工具的使用者

- 监控系统本身的产能规划困难
  - 工程师对数据的读写规律难以预测
  - 对策：自动阔缩，冗余产能以保障高可用性
- 90%的数据从未没读取
  - 根源：冗余代码，采样率过大，非关键点的没必要的测量
  - 对策：教育
- 工具新性能的推广困难
  - 对策： 重视用户界面设计，借用专业图表提供商的解决方案


99.98
OVERALL SUCCESS RATE
274.5K
TOTAL RPS
730.36
P90 LATENCY (ms)
100
99.86
99.89
100


varnish app 5xx count
varnish api 5xx count
varnish widgets 5xx count
@timestamp per minute
ngapp_dhistogram_count_by_payloadSignature [TOP 6]
ngapi_dhistogram_count_by_payloadMessageSignature [TOP 6]
Legend
Count

三种监测工具，三种数据

GOPS2017
Beijing

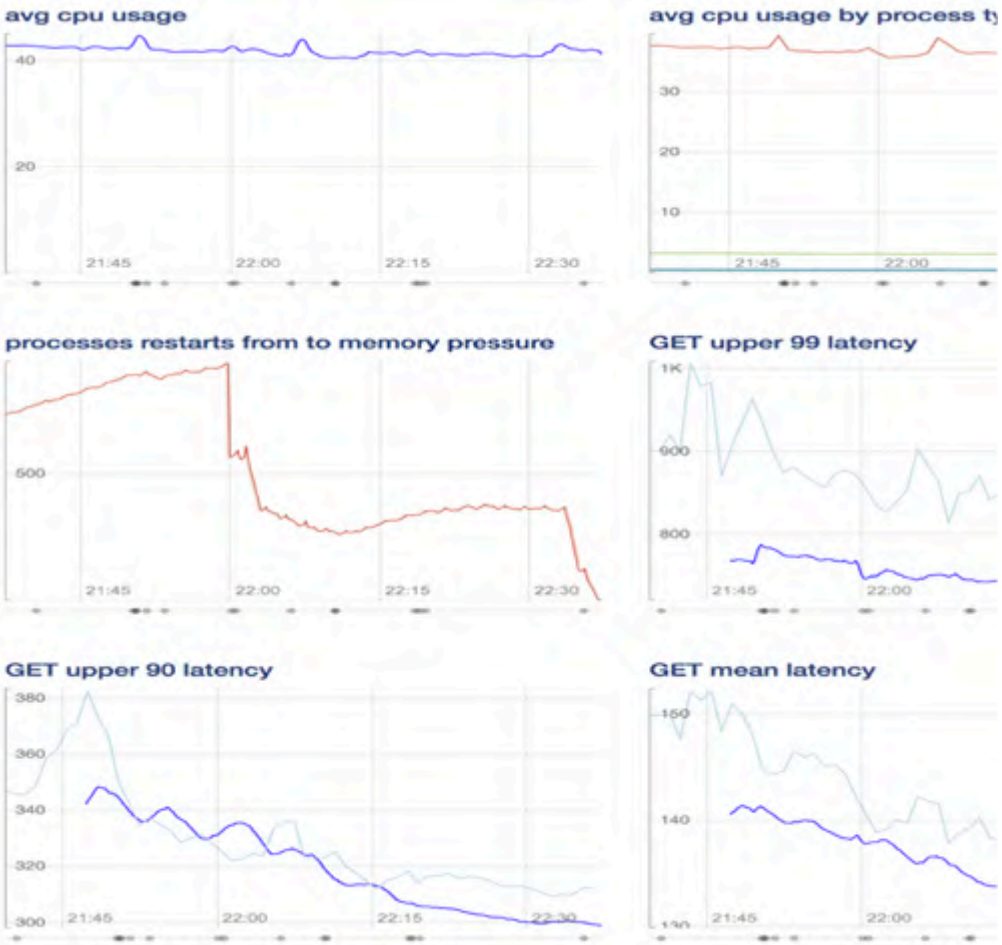
时序指标数据

日志

分布式跟踪数据

时序指标数据

- 对系统总的数值化描述，一般是计数器，延迟，测量值等
- 成本低
- 缺乏服务间关系的描述
- 缺乏对单一用户请求的描述

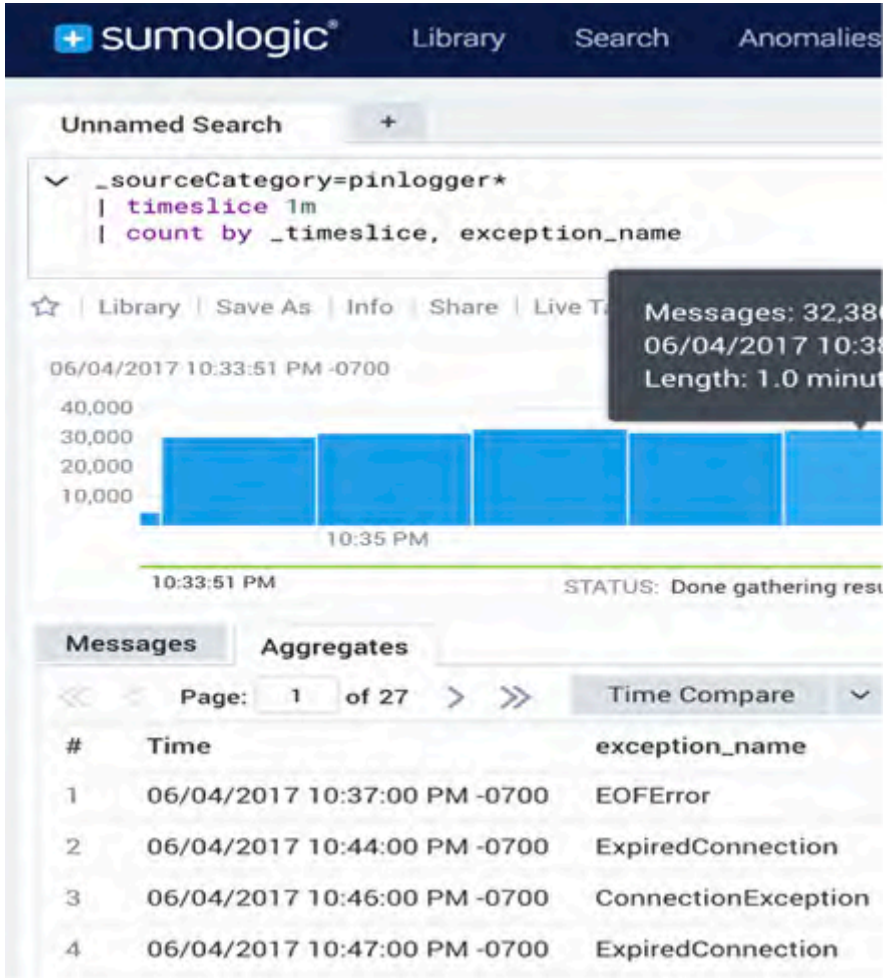


GOPS2017
Beijing

日志

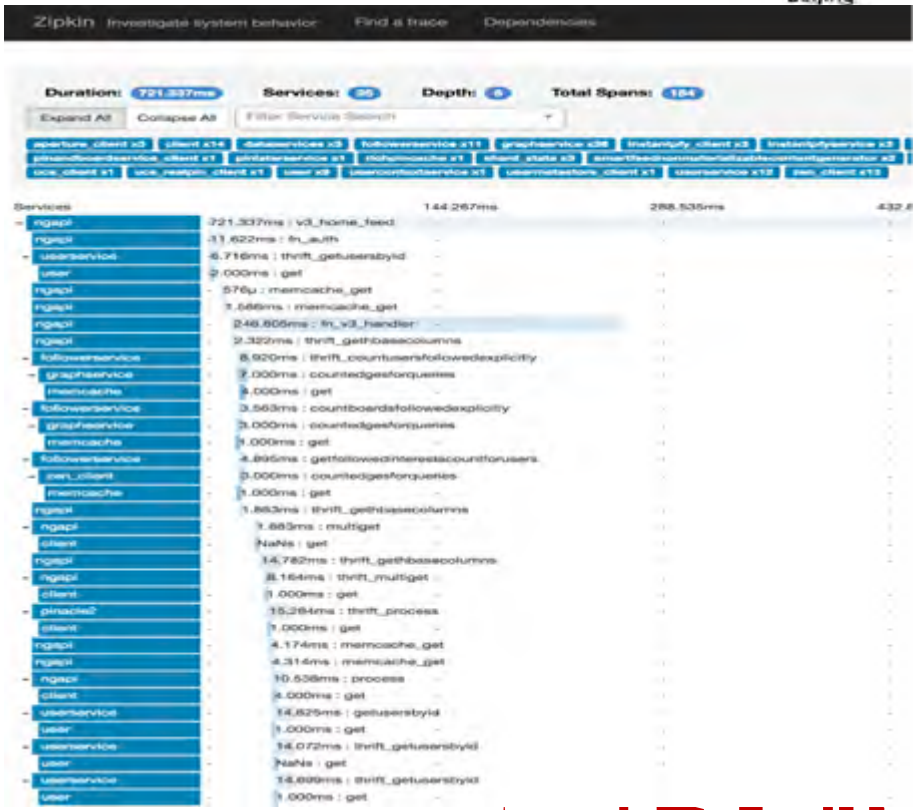
- 对单一事件，尤其是错误事件的记录
- 事件记录里有丰富的上下文描述

- 缺乏对服务总体和服务间关系的描述
- 日志索引的成本较高



用户请求的跟踪数据

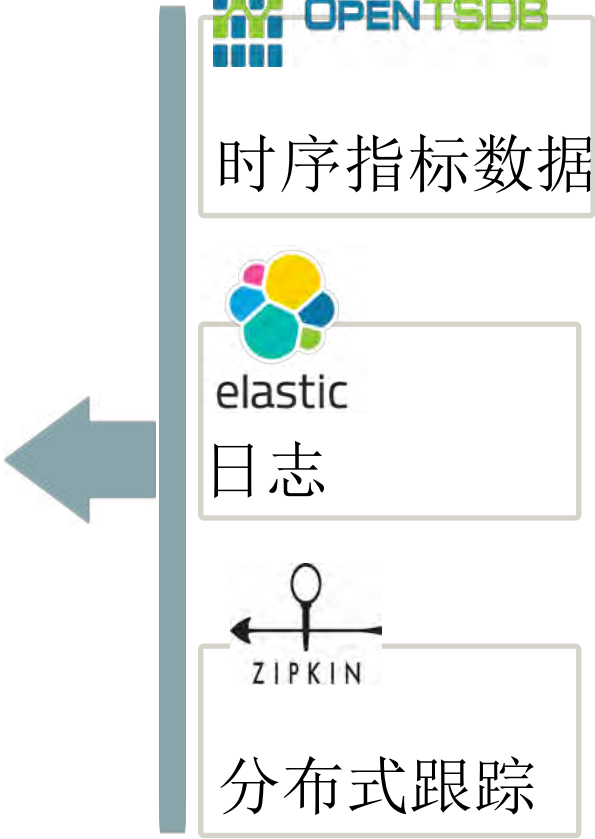
- 记录每个服务API的调用
- 把跨服务，跨设备的记录串联起来
- 事件记录里有准确的时间信息
  和丰富的上下文描述



- 对用户请求性能有影响，只能
  少量抽样 (例如0.1%)

监控系统下一步： 集成化

- 串联起三种数据类型，营造功能更强大的集成化的监控工具
- 实例：
  - 前端工具上，把来自于同一服务的时序指标和日志放在一起，提高查错速度
  - 利用用户请求ID的独一性，串联起日志和跟踪数据，可以快速确定响应慢的用户请求的来源
- 大量监控系统提供商的共识



监控系统下一步： 智能化

- 运维数据的深度挖掘和智能分析
  - 自动搜索关联数据以快速发现问题根源
- 智能报警系统
  - 冗余警报的自动过滤
  - 自我调整的阈值设定
  - 警报的智能推送

signifai

高效运维社区
GreatOPS Community

GOPS2017
Beijing

会议

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

培训

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训

咨询

- 企业DevOps 实践咨询
- 企业运维咨询

商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com

Thanks

高效运维社区

开放运维联盟

荣誉出品

想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好