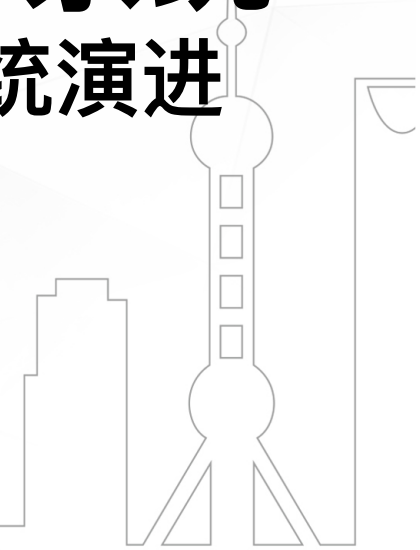
QCon[上海站]
全球软件开发大会2016

LinkedIn实时日志分析系统
基于Kafka&ElasticSearch的系统演进

SPEAKER

李 琥 (Li Xiao)



International Software
Development Conference

主办方 Geekbang. 极客邦科技 InfoQ

日程

1. 自我介绍 & LinkedIn简介
2. 日志分析系统基本需求
3. LinkedIn的日志系统演进过程
4. 我们的经验

自我介绍

1. 2012 至今：LinkedIn SRE
2. 负责 LinkedIn 在线支付系统，高级会员功能，公共 API 接口，视频上传和分享等系统的SRE工作。
3. 2015年：负责 ELK（日志系统）@LinkedIn 。
4. 2016年：负责 LinkedIn 纽约地区SRE团队建设

LinkedIn 简介

1. 450M（4.5亿）用户
2. 4000+ Eng
3. 2012年 SRE：20人
4. 2016年 SRE：350人
5. 5个数据中心，50000+ 服务器
6. 领英中国的发展

为什么要日志分析系统

1. 日志（log) = 数据 + 时间戳
2. cat/tail/grep/less/awk/cut + 正则表达式就足够了
3. 如果不够，加上mssh,cssh,multitail…
4. 如果还不够，还有很多“创意”

为什么要日志分析系统

Enough is enough...

为什么要日志分析系统

- 找一下某个服务两天前9:42-10:28之间的日志
- 只要看警告或者更严重的消息 [warn] [error] [fatal]
- 有十几个错误是已知的，要忽略
- 这个服务跑在5个数据中心600多台服务器上
- 有没有新的错误？是否只发生在某特定用户的请求造成？
- 是否和其他服务的错误消息相关？
- 现在是凌晨3点

为什么要日志分析系统

为什么要日志分析系统

基本需求

- 满足索引、检索、排序、分类、可视化、分析日志的功能
- 可根据数据规模横向扩展
- 跨数据中心支持
- 支持功能扩展，可以接入已有其他系统

扩展需求

- 提高系统可维护性
- 提高安全性、保护用户信息
- 取样vs记录全部

LinkedIn 日志系统演进

?? ~ 2012 : Splunk

优点：好用

缺点：很贵

LinkedIn 日志系统演进

2012~2014：混沌期

优点：N/A

缺点：乱、功能不足

LinkedIn 日志系统演进

2014-2015 : ELK

优点：开源，发布周期短

缺点：比较新

LinkedIn 日志系统演进

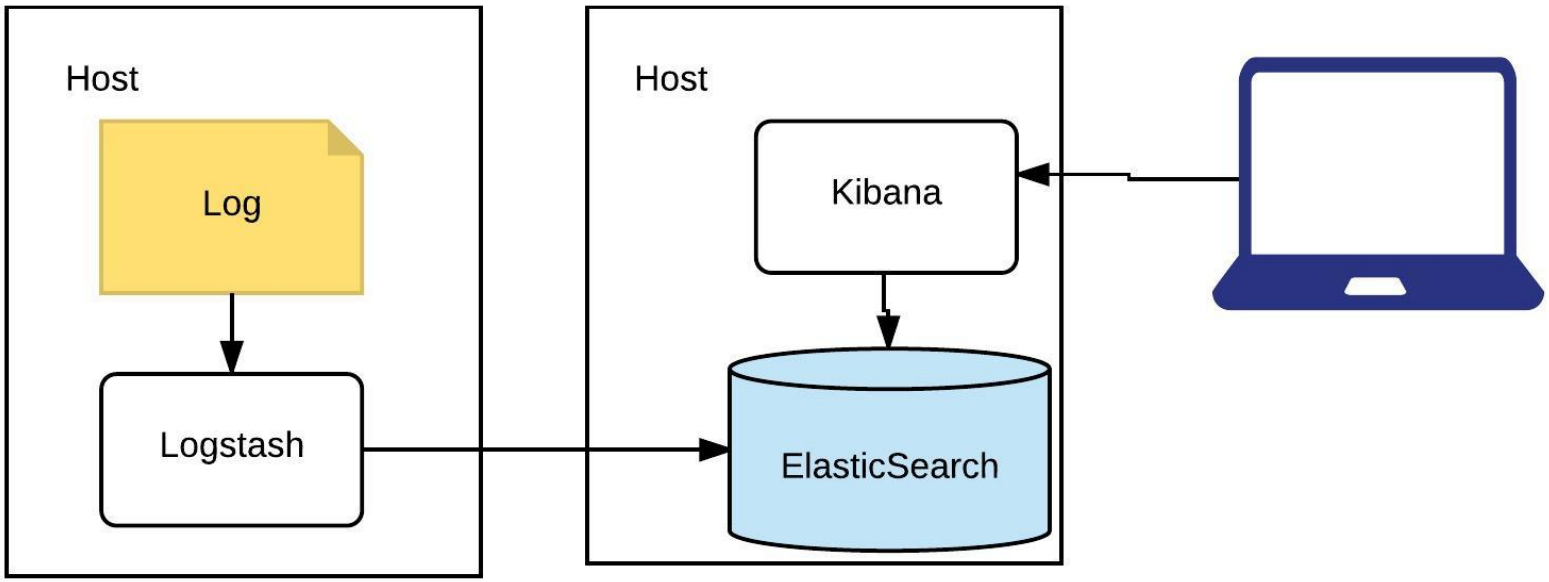
ELK 是什么

ElasticSearch: 基于lucene的存储，索引，搜索引擎

Logstash: 提供输入输出以及转换处理插件的日志标准化管道

Kibana: 提供方便可视化和查询ES的用户界面

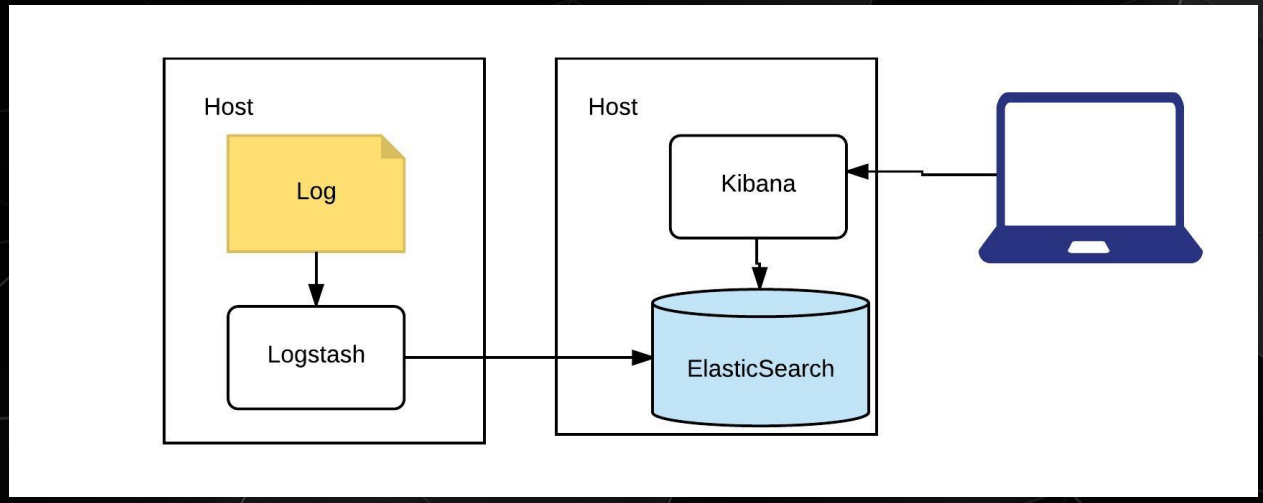
LinkedIn 日志系统演进 V1



LinkedIn 日志系统演进 V1

V1 存在的问题

- Logstash Agent 维护
- Log标准化



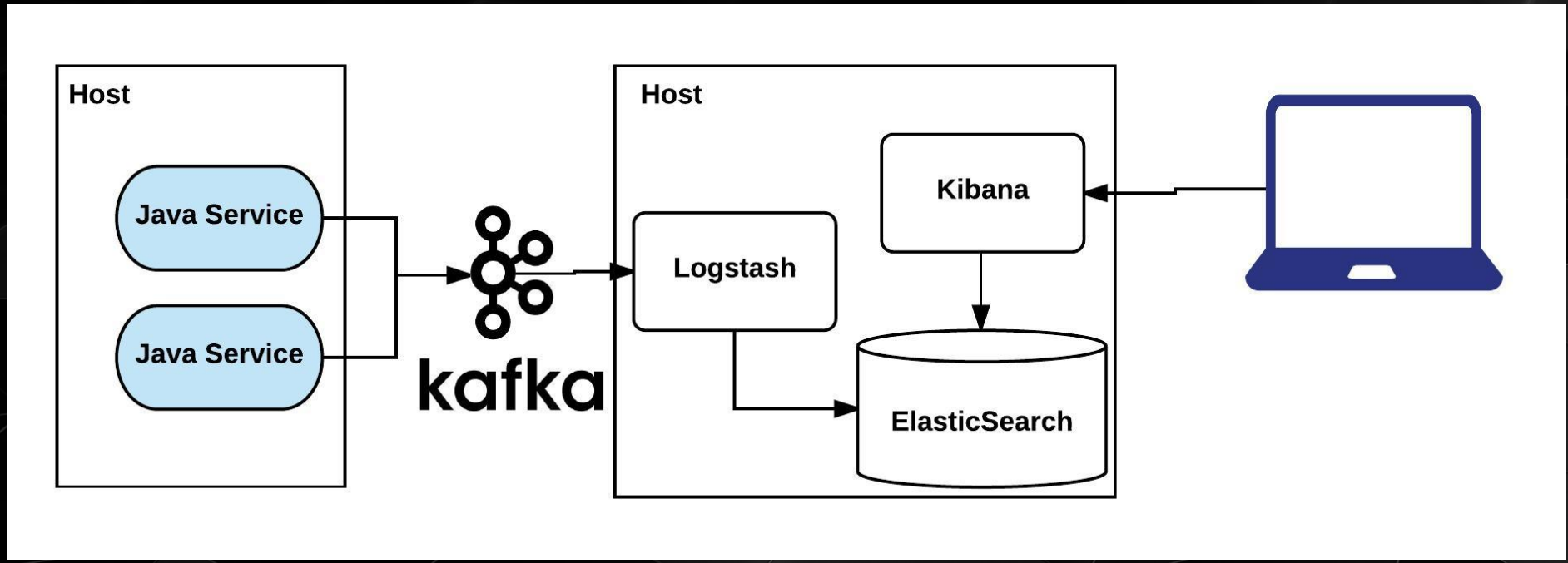
LinkedIn 日志系统演进 – 引入Kafka

- 高性能分布式消息系统
- LinkedIn Kafka 系统 1.1 兆 消息每天 (Trillion)
- Kafka SRE 维护
- 最高效的接入方式是 KCC + pipe
- 单独Logging cluster (Security)， 不做replication不进Hadoop

LinkedIn 日志系统演进 – Log 标准化

- LinedIn主要是Java service. 有15种+ log
- 最常用的： access log, application log
- 通过Java Container logger 标准化直接写入Kafka
- 程序日志：默认警告以上消息级别进入kafka，可通过JMX控制
- 访问日志：10%取样，可动态控制（通过ATS入口控制）

LinkedIn 日志系统演进 V2



LinkedIn 日志系统演进 V2

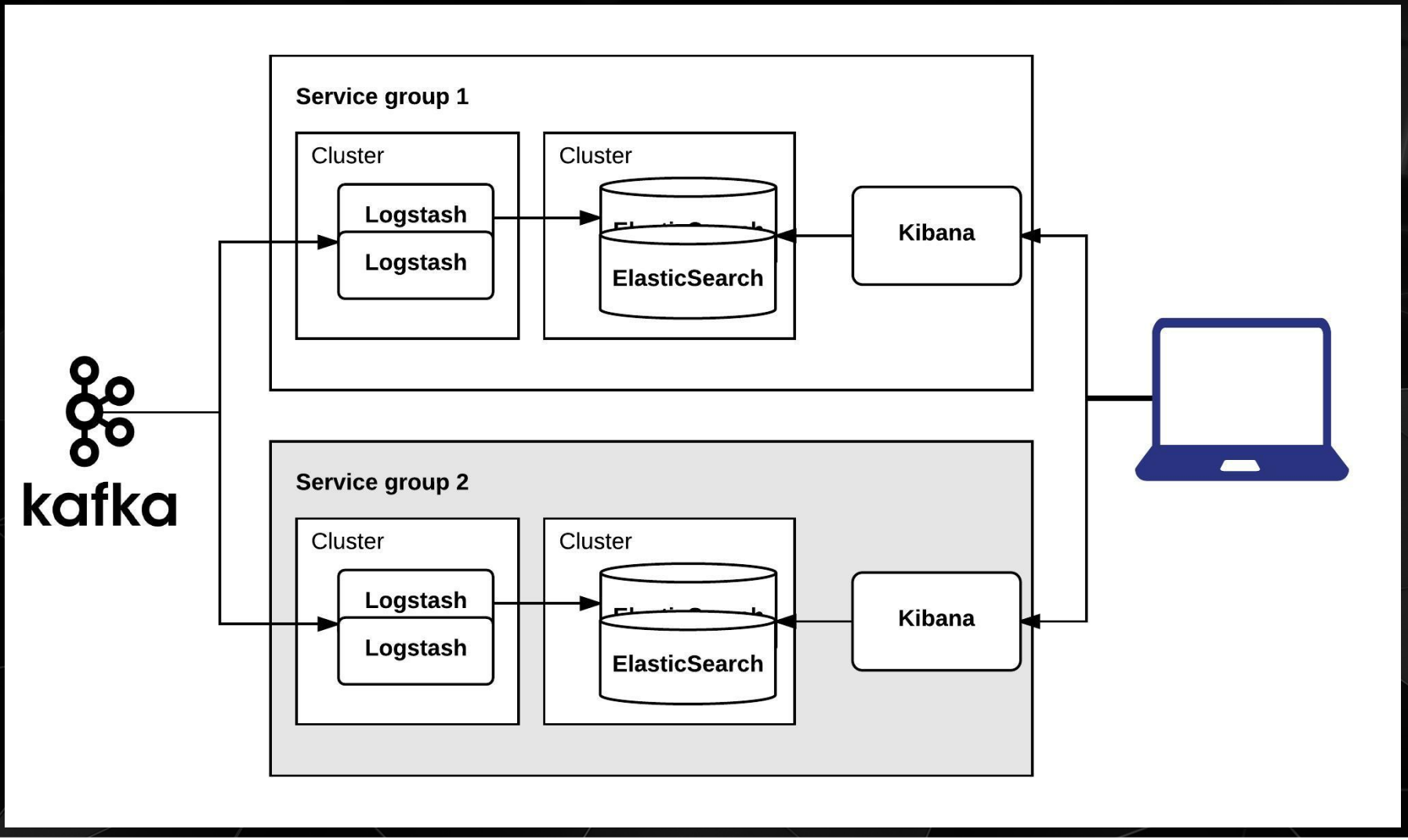
V2 存在的问题

- 一个服务出现问题会影响整个ELK cluster
- 网络饱和

LinkedIn 日志系统演进 – 拆分优化

- 按照业务功能拆分ELK Cluster, 做到互不干扰
- 将Logstash和ElasticSearch 分开运行
- 对于每个kafka话题，LS数量不少于话题partition数量

LinkedIn 日志系统演进 V3



LinkedIn 日志系统演进 V3

V3 存在的问题

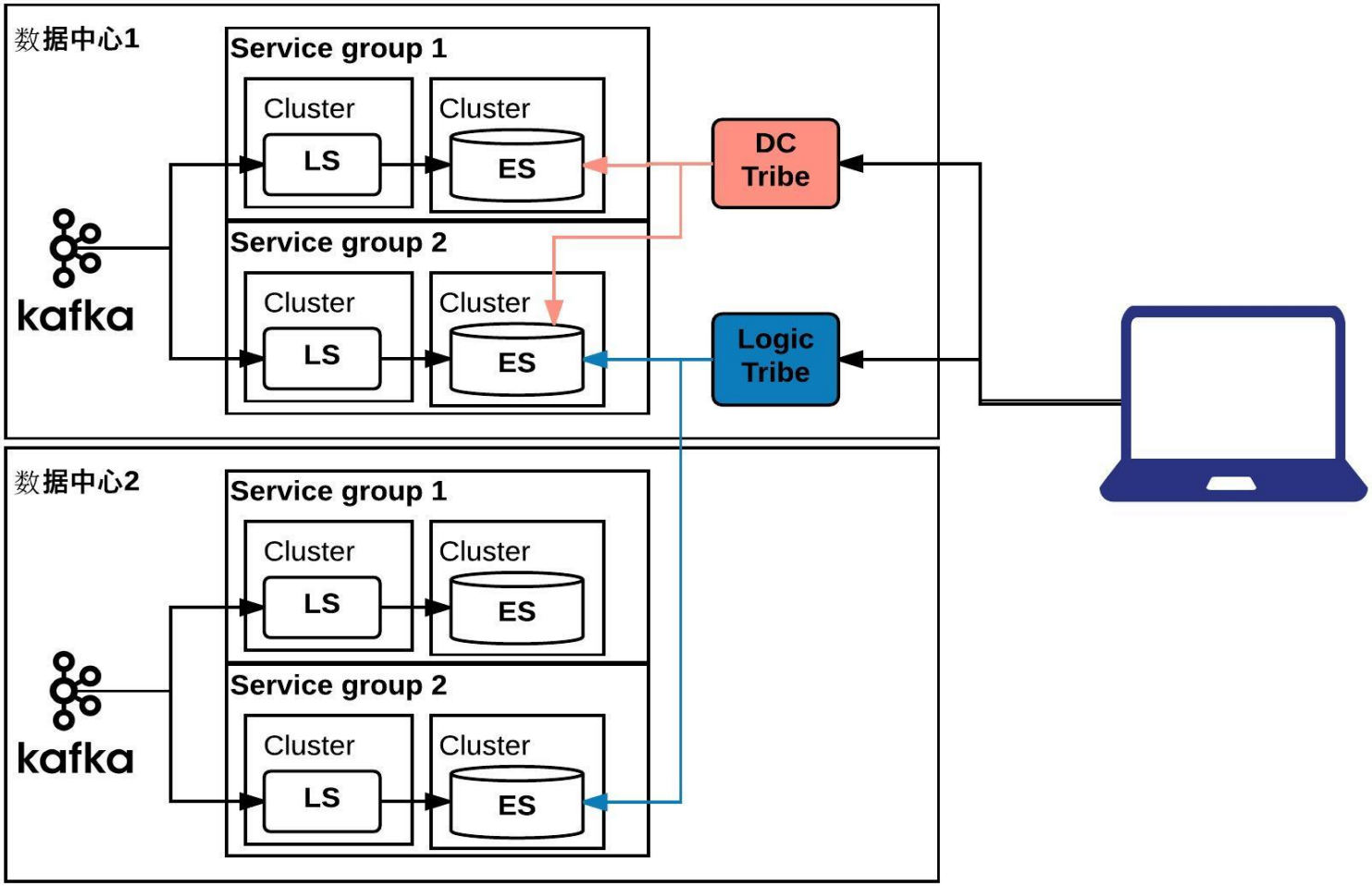
- 跨业务分组cluster查询
- 跨数据中心查询

LinkedIn 日志系统演进 – 跨界查询

引入Tribe

- Tribe不支持分层结构 ☹
- 跨数据中心Tribe
- 跨业务分组Tribe

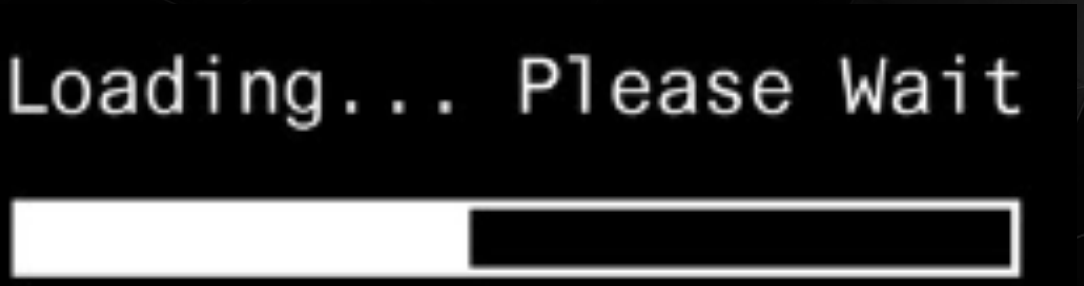
LinkedIn 日志系统演进 V4



LinkedIn 日志系统演进 V4

V4 存在的问题

- 慢



LinkedIn 日志系统演进 – 性能提高

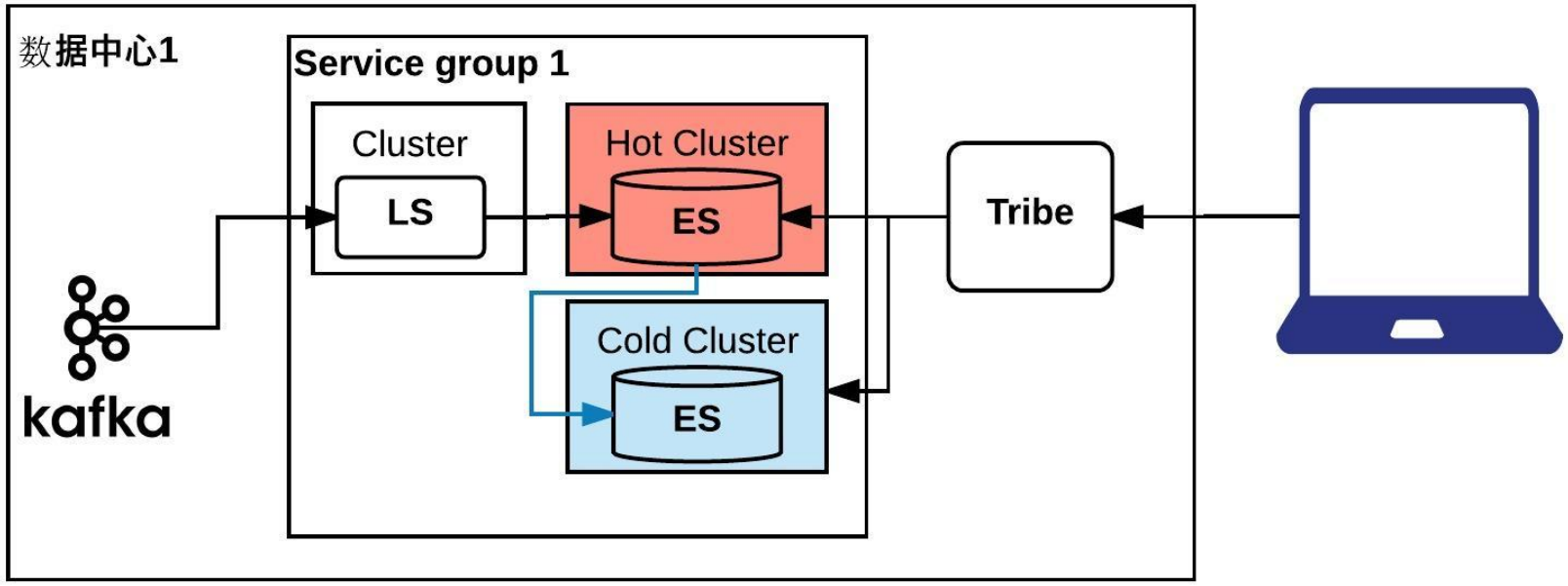
理解具体用例

- 最近24小时的日志查询最多
- 14天以前数据很少用到
- 查询速度：5秒内最佳，15秒可接受，30秒超时
- 索引速度：30秒以内可接受，5分钟以上触发警报

LinkedIn 日志系统演进 – 性能提高

- 使用不同硬件区分Hot/Cold index
- SSD Host 做 索引和24小时内查询
- 24小时以后把index移到普通host

LinkedIn 日志系统演进 – V5



LinkedIn 日志系统演进 – V5

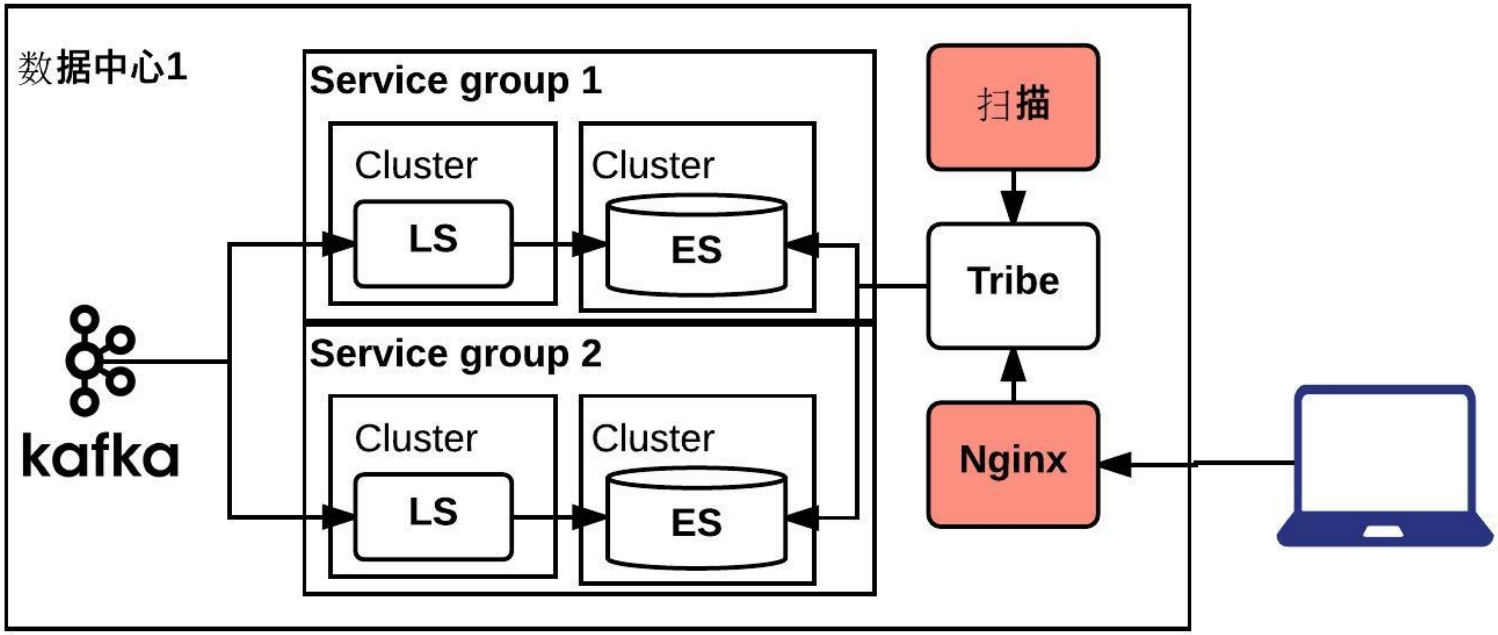
数据安全

- 定期扫描所有ES，防止敏感信息进入日志

用户隐私

- 所有ELK查询记录同样送入Kafka
- 敏感业务部门访问隔离
- 定期离线审计ELK查询记录

LinkedIn 日志系统演进 – 现状



LinkedIn 日志系统演进 – 现状

- 20多个针对不同业务模块的ELK集群 1000+服务器
- 准实时，保留7-14天
- 500亿索引文档，500-800T
- 每个业务模块SRE 维护自己的ELK集群
- Virtual Team 模式确保 ELK 及时更新
- ES 节点间通过SSL连接 避免未授权访问 (SearchGuard/Sheild)

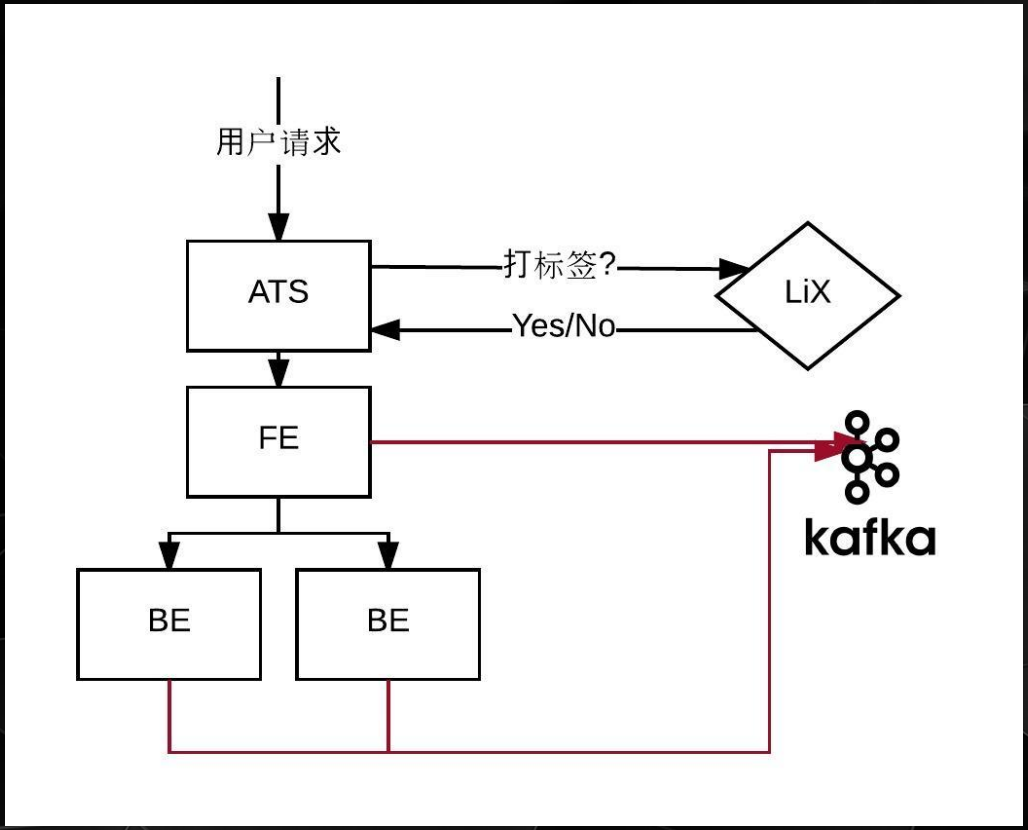
LinkedIn 日志系统演进 – 现状（2）

访问日志取样方式：

- 10% + 特定用户

查询聚合：

- memberID
- requestID



LinkedIn 日志系统演进 – 日常运维 之 坑

- 集群默认名字是大坑
- Master, Data, Client 使用单另节点
- JVM 30G heap 限制
- 小心JVM 版本兼容性

LinkedIn 日志系统演进 – 日常运维 之 硬件

- 根据数据量，索引和查询速度要求，数据类型决定
- 我们的主要硬件: 12核/64G/2x1TB SATA
- 数据量大优先考虑JBODs
- 索引速度快优先考虑 SSD

LinkedIn 日志系统演进 – 日常运维 之 集群

- 集群大小和影响因素 系统性的压力测试
- 横向扩展
- Shard 不要超过50G
- 关闭冗余* (replication)
- 每天建新的索引*
- 仔细测试不同映射（mapping）带来的影响
- 只analyze必要字段

LinkedIn 日志系统演进 – 日常运维 之 工具

- 主动扫描敏感信息：内部自建
- 结合警报系统：内部自建，ElasticAlert
- 循环删除index: Curator
- 系统健康状况监控：自建/Marvel

LinkedIn 日志系统 – 提高代码质量

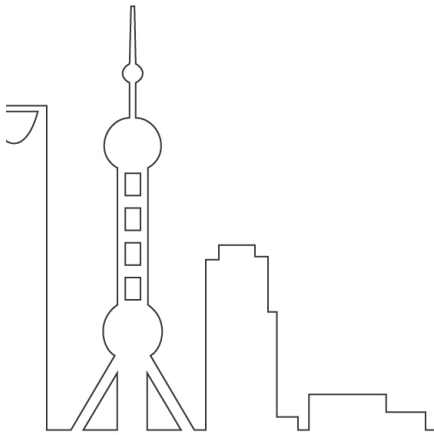
- 请求数量/日志总行数
- 请求数量/错误（异常）行数
- 标准化异常处理

相关分享

《如何打造一个百万亿级的日志搜索引擎：Poseidon》
360 高级工程师 & 资深顾问 魏自立

《使用 Apache Kafka 进行关键业务消息传输》
LinkedIn 数据基础架构部门 Kafka 组高级软件工程师 秦江杰

如何决定集群大小 Quantitative Cluster Sizing



Thanks!

International Software Development Conference

主办方 Geekbang. 极客邦科技 InfoQ