elastic 中国开发者大会 2023

ES 在字节跳动的实践

黄杨锋，ByteES 负责人
字节跳动， 2023/04/08

ByteES 之

应用 在字节的应用概括和场景

架构 可用性架构和分离架构

内核 性能、可用性、安全、可观测性、功能强化、成本优化

容器 ES on K8S上的一些演进和探索

生态 ES的周边生态建设

未来 未来我们准备重点投入到哪里

elastic

应用：ByteES 是什么？

团队
成长成就

- 业务BP
- 平台服务
- Core&Plugins
- 云原生

产品
追求极致

- 内核 (ByteES 23.1)
- 架构演进
- 插件
- 周边生态

服务
稳定好用

- 稳定是基石
- 性能快
- 响应快
- 体验好

应用：业务概况

~1500	~3000	~40,000
业务数	集群数	节点数

应用：应用场景

01	结构化搜索	• 等值查询/范围查询 • 区分度高低不均	• 电商/教育/房产 等业务
02	高安全性场景	• 存储敏感信息（如消息/薪资/简历等） • 安全分级/全链路加密/落地加密 • 审计/认证授权	• 飞书/People OKR 等业务
03	全文检索	• 模糊查询（match， query_string 等） • 注重排序/打分/分词 • 结合搜索中台做各种搜索策略	• 小说/音乐搜索 等业务
04	时序型场景	• 写多读少（日志/metric 等） • 按天/月 建立索引 • 冷热分离，节约成本	• 日志/风控/metric 等业务
05	向量检索	• 音频/视频/图片的检索 • 维度/召回率/动态更新/内存消耗	• 音视频/拍搜题/云图片 等业务
06	POI 检索	• GEO查询(球面距离查询，范围查询） • CPU消耗大，按区域分索引	• 地理位置中台/抖音POI 等业务
07	私有化/公有云	• ToB类 • 版权问题	• KA1/KA2/KA3 等客户

ByteES 之

应用 在字节的应用概括和场景

架构 可用性架构和分离架构

内核 性能、可用性、安全、可观测性、功能强化、成本优化

容器 ES on K8S上的一些演进和探索

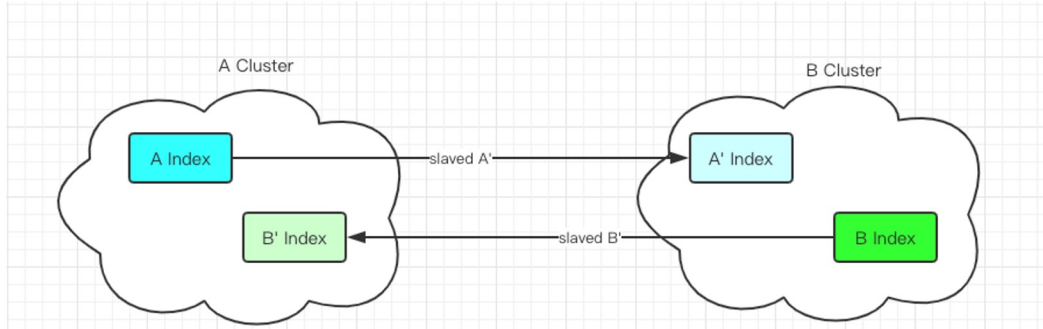
生态 ES的周边生态建设

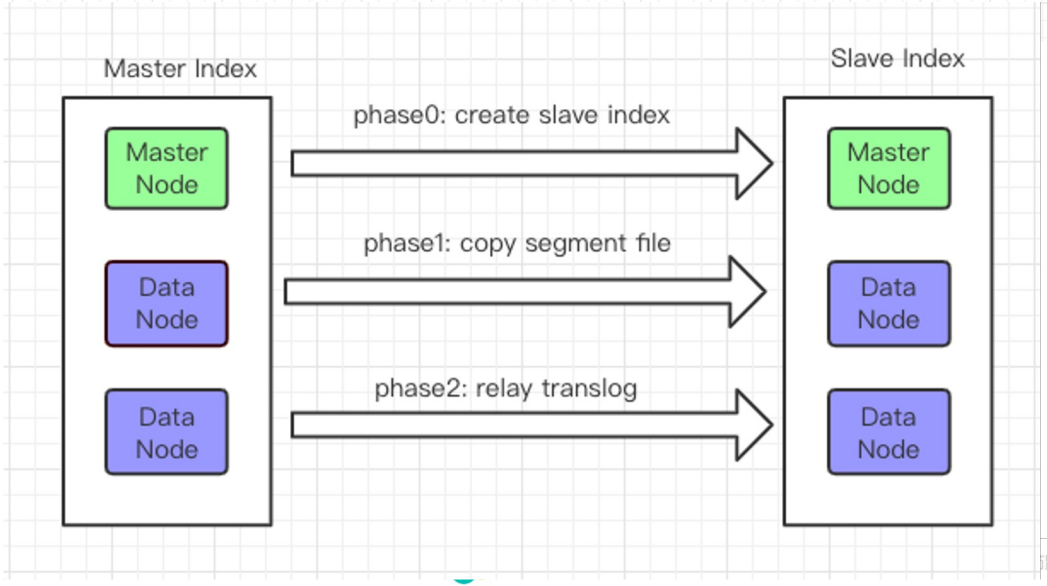
未来 未来我们准备重点投入到哪里

elastic

架构：可用性架构 - freeCCR 主从同步

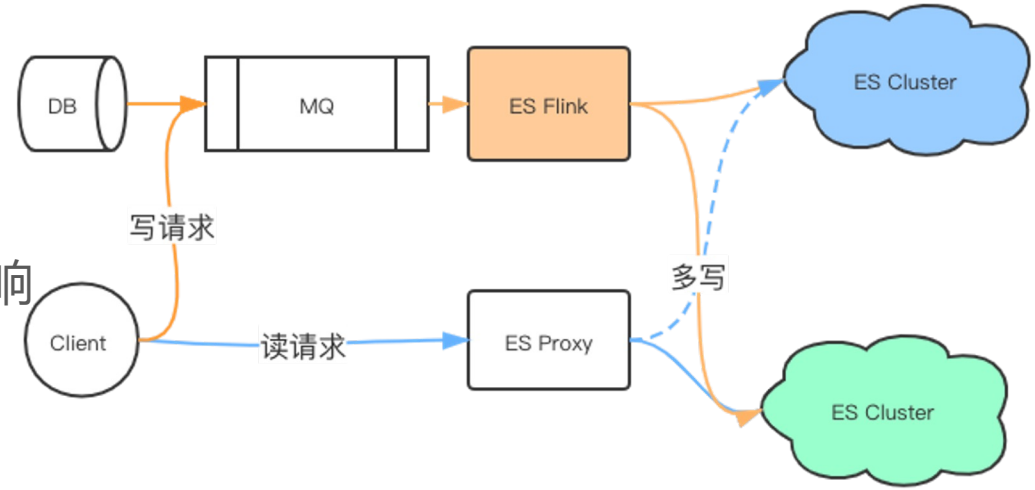
- 诉求：
  - 多备分摊读压力
  - 容灾需求
  - Licencse Free
  - 实时性要求较高
- 特点：
  - 索引维度的主备
  - 不支持主备切换
  - 采用推送方式
  - 秒级延迟





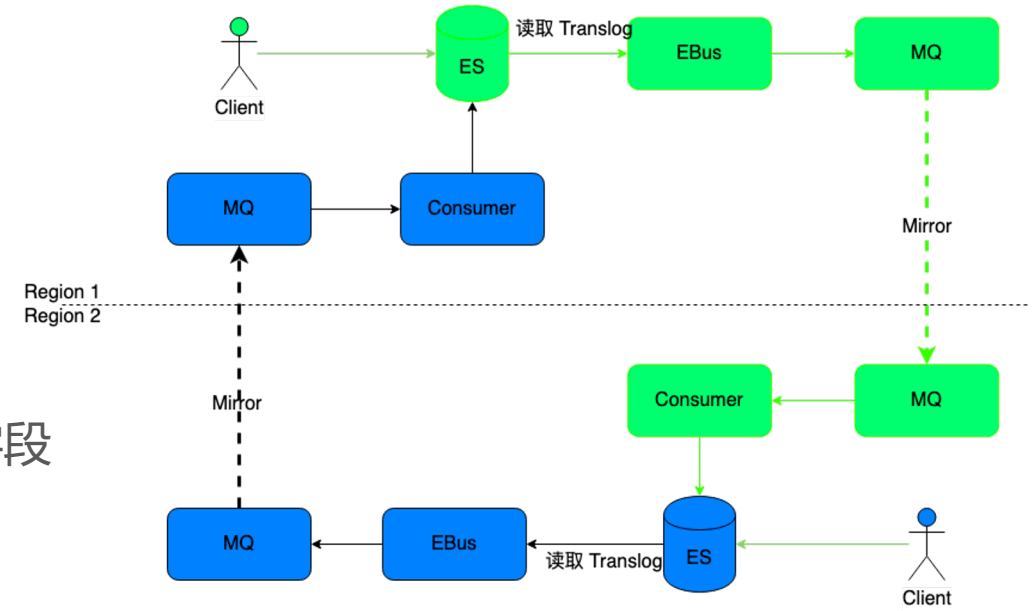
架构：可用性架构 - 多活

- 诉求：
  - 数据已经在MQ/Mysql
  - 不能降级读
- 特点：
  - 消费MQ，消峰稳定
  - 回拨offset，补写重放
  - 相互独立，集群间互不影响
  - 故障期间能读写最新数据
  - 链路较长



架构：可用性架构 - 跨区域数据同步(CRR)

- 诉求：
  - 跨洋跨区域容灾
  - 跨洋跨区域数据同步
- 特点：
  - 适用于网络延时高(200ms)
  - 数据回环
  - 写入冲突
- 局限：
  - 多区域对同一文档的不同字段同时部分更新
- 数据一致性校验及补偿



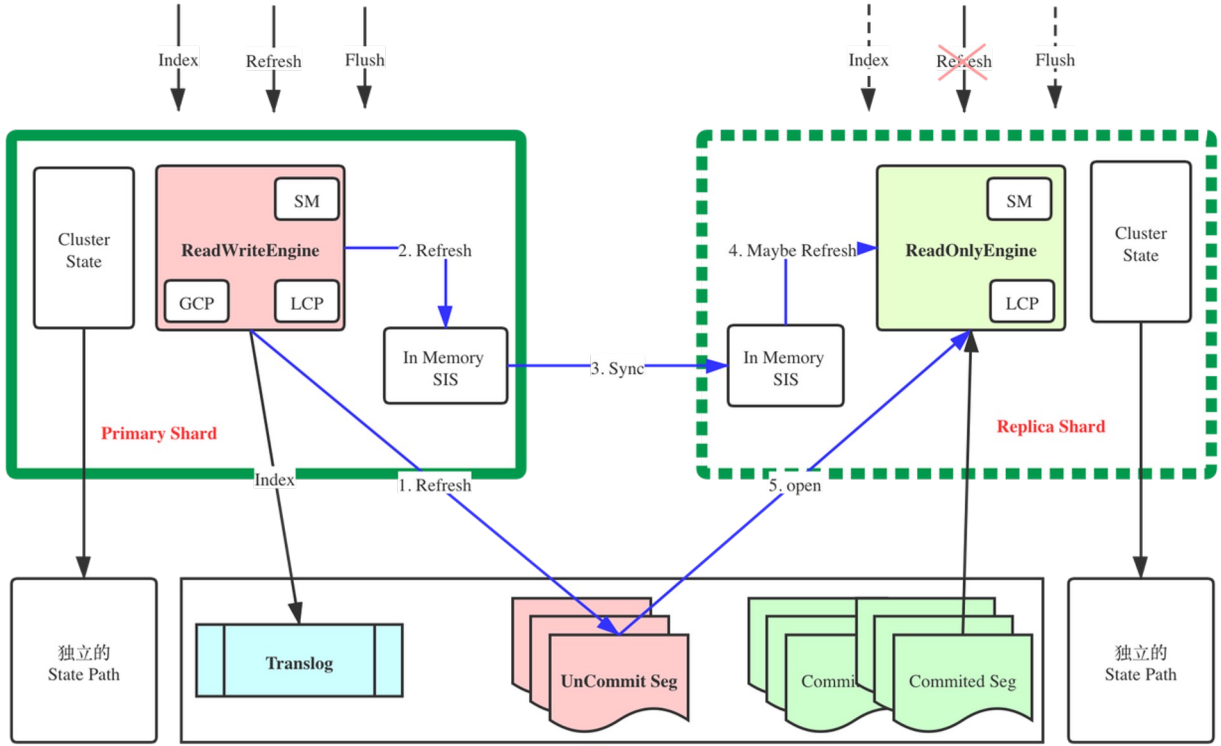
架构：分离架构 - 计算存储分离

- 业务诉求：
  - CPU 100%了，能否快速扩计算资源？
  - 写入不够快？
  - 扩/缩容再快一些？
  - 运维是否更高效？
  - 成本能否更低？

- 计算存储分离：
  - 不仅仅是使用了网络存储
  - 计算资源和存储资源彻底解耦，不会相互影响
- 收益：
  - 计算/存储资源秒级扩容
  - shard relocation/split/shrink更快速
  - 不用写副本，写性能更快
  - 存储成本只有以前的1/(n+1)，并且安使用量计费

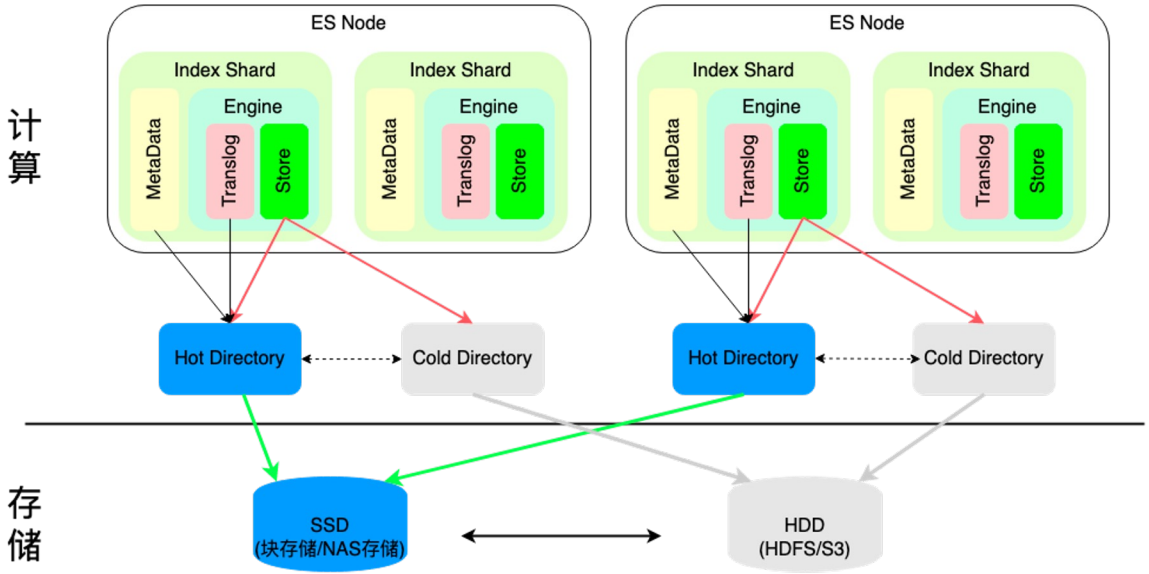
架构：分离架构 - 计算存储分离

- 特点：
  - Primary Shard负责写
  - 数据和translog 均存储在NAS上
  - 同步SegmentInfos
  - state 互相独立
  - 主备毫秒级可见
  - 可插拔Directory，支持HDFS等其它存储
- 前提条件：
  - 存储支持一写多读



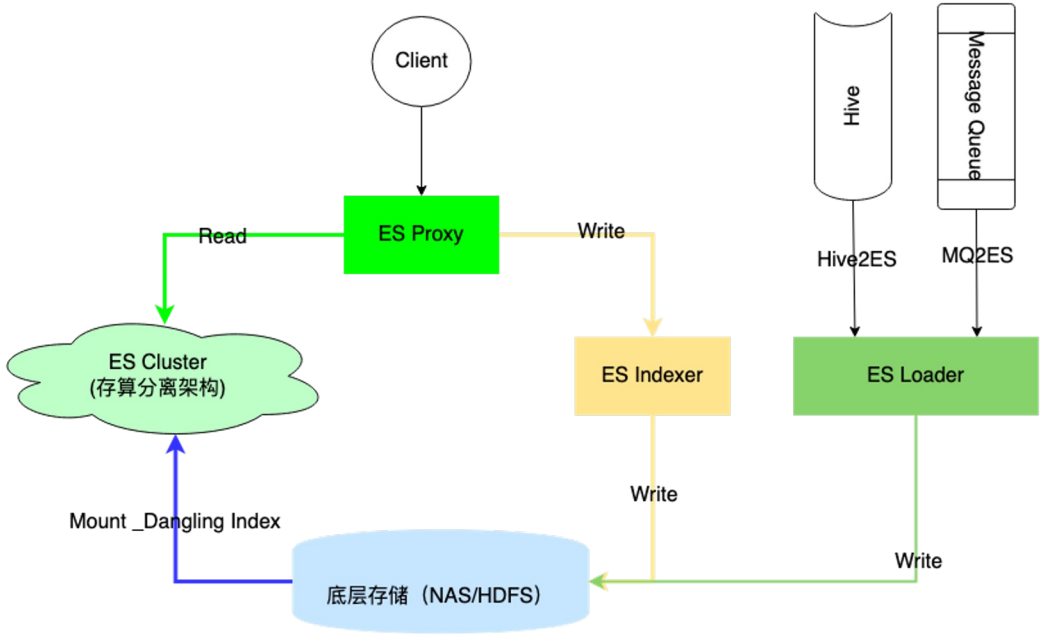
架构：分离架构 - 冷热分离

- 传统冷热分离：
  - 区分热节点/冷节点
  - 计算和冷热存储介质强绑定
- 特点：
  - 计算不分冷热，存储分冷热
  - Master节点无感知、Shard不用迁移
  - 迁移流程更简单/性能更快
  - Translog/metadata 存热介质，写冷数据性能更快
  - 同时适配存算分离架构和传统架构
  - Segment 级别的细粒度调节，灵活性更高

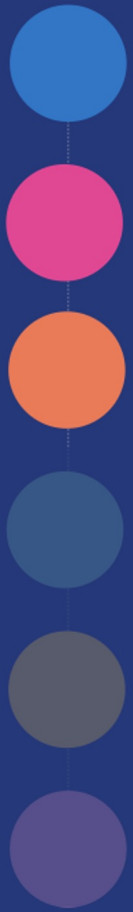


架构：分离架构 - 读写分离

- 非读写分离：
  - 读写耦合一起，扩缩容不方便，性能较低，成本较高
- 常规读写分离：
  - 额外拷贝至少一次数据
- 特点：
  - 写服务无状态、自适应扩缩容，提升性能
  - 潮汐特性，降低成本
  - 基于存算分离架构，零拷贝索引挂载
- 难点：
  - 优雅处理更新/删除



ByteES 之



应用 在字节的应用概括和场景

架构 可用性架构和分离架构

内核 性能、可用性、安全、可观测性、功能强化、成本优化

容器 ES on K8S上的一些演进和探索

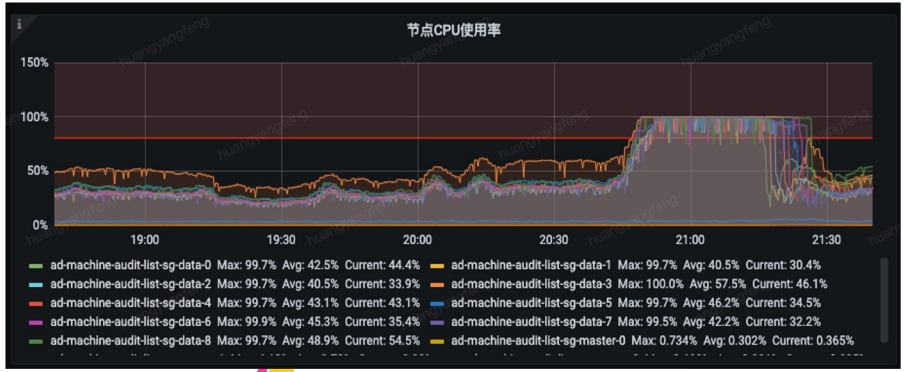
生态 ES的周边生态建设

未来 未来我们准备重点投入到哪里

elastic

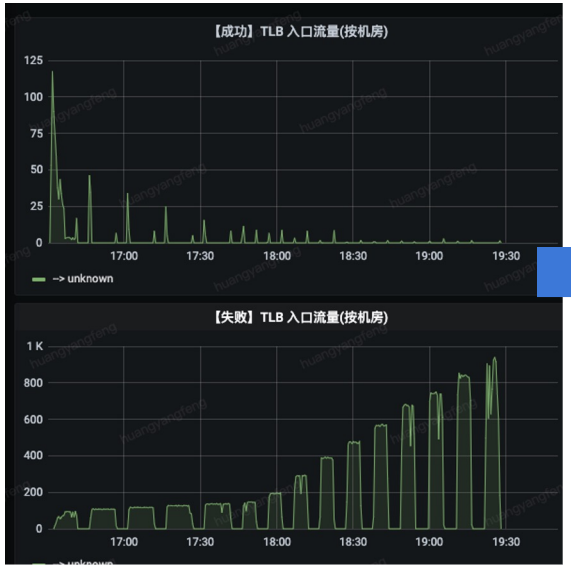
可用：可控延时队列

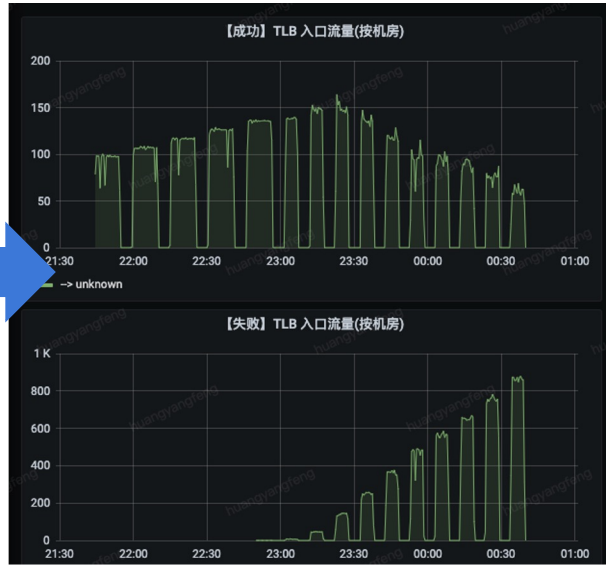
- 现象：请求波动导致服务不可用
  - 延时很长，基本都超时
  - CPU 暴增
  - 内部堆积和Reject 请求数激增
- 业务预期：
  - 过载的部分失败，不过载部分应该要正常，**而实际都失败。**
  - CPU 的变化应该和QPS的变化应该差不多的，**而实际不是。**



可用：可控延时队列

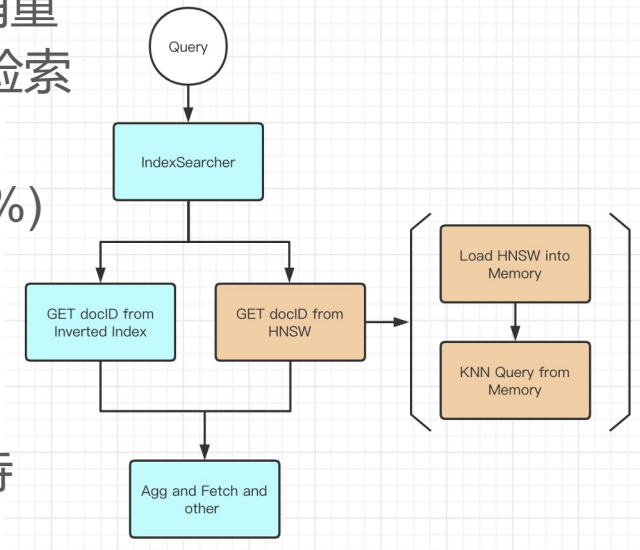
- 思路：
  - Search Queue 里还应考虑等待时间，对于等待过长的直接失败返回
- 效果：
  - 请求成功率有数倍提升
  - 堆积也能快速恢复

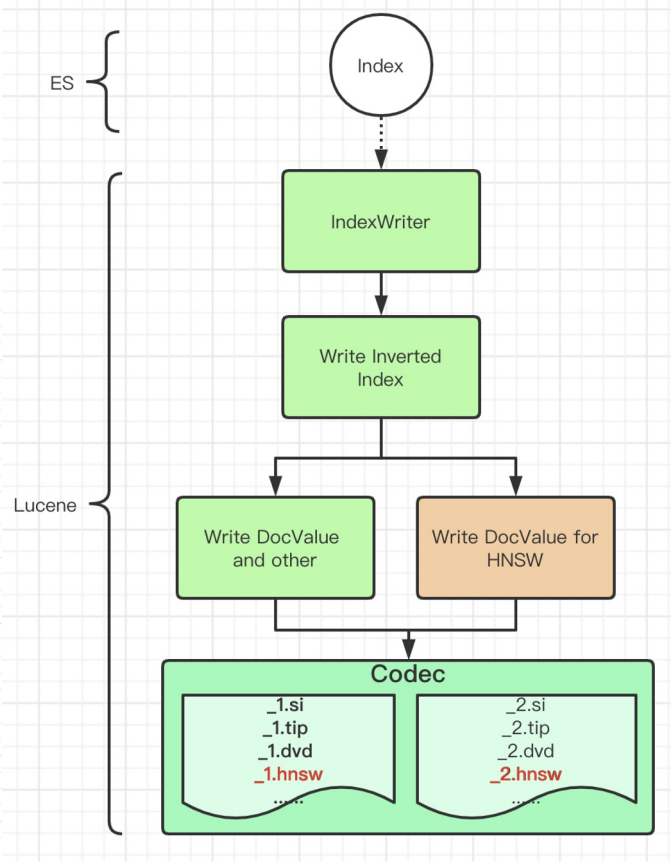




功能：向量检索

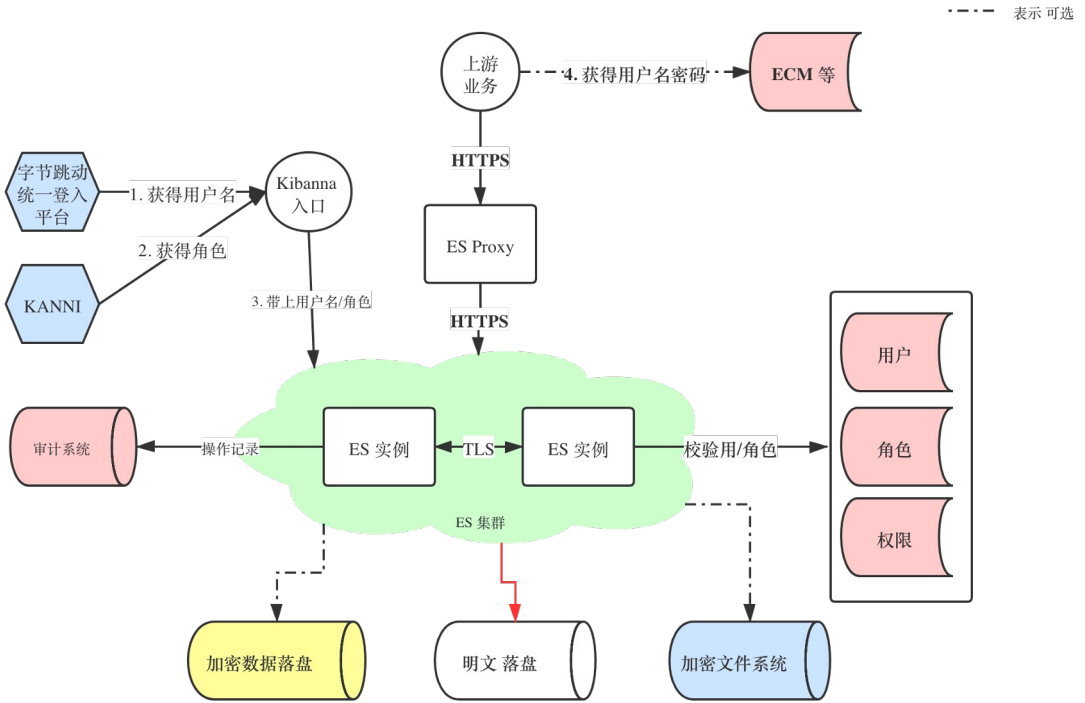
- 诉求
  - 抖音/西瓜 音视频搜索
  - 高危音视频消重
  - 教育类 图片检索
- 功能特点
  - 召回率高(98%)
  - 查询速速快
  - 全内存
  - 耗CPU
  - HNSW已支持
  - PQ支持中





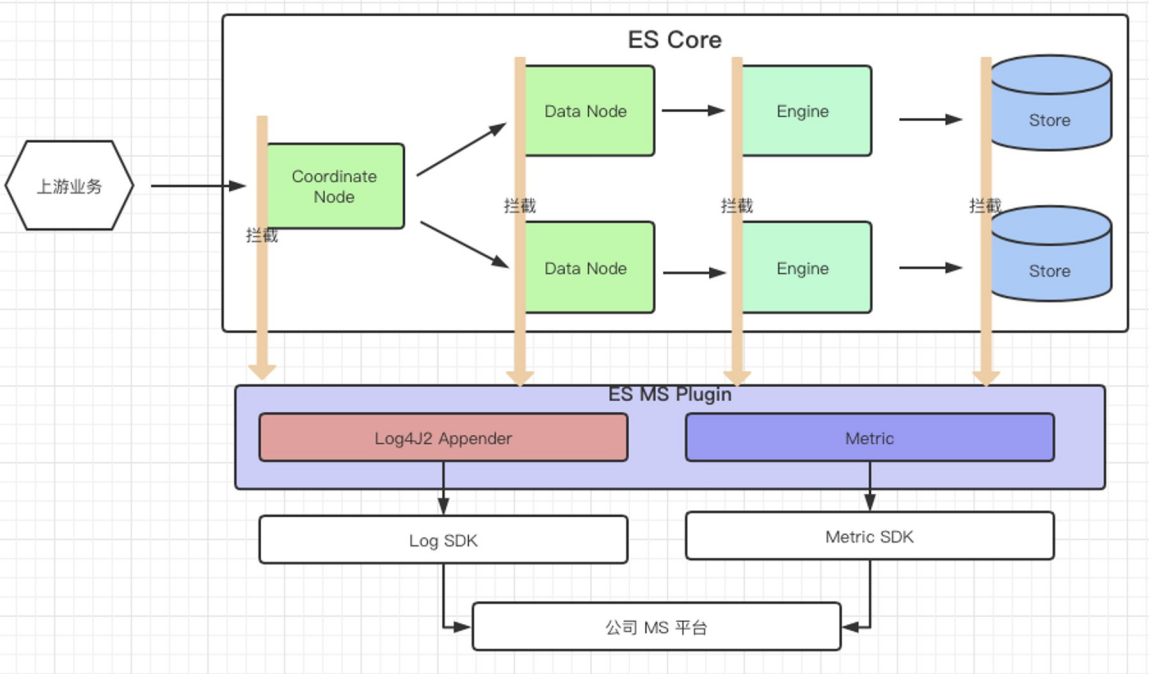
安全：全链路

- 认证授权
  - 基于角色的访问权限控制
  - 支持KANI /SSO
  - 支持GDPR
  - 支持API 细粒度授权
- 安全加密
  - 支持 HTTPS/TLS 访问
  - 支持ECM/KMS 数据加密
- 监控审计
  - 支持审计操作
  - 支持审计监控告警
- 数据安全
  - 支持数据加密落地
  - 支持加密文件系统
  - 堡垒机管控
  - 日志脱敏



问题定位： 内核服务治理

- 诉求
  - 内核侧全链路追踪
  - 内核多层次Metric
  - 分析各个环节的性能
- 功能
  - 可追踪所有读/写请求
  - 可追踪读写慢日志
  - 无缝接入MS的日志搜索方便使用
  - 可观测到接入层/数据处理层/引擎层/存储层的metric和trace

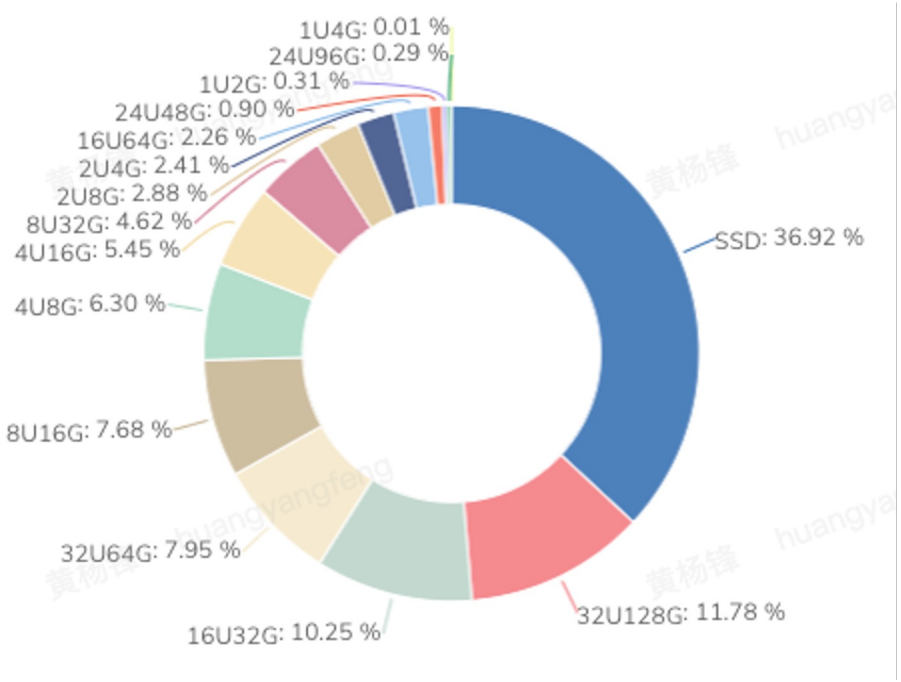


问题定位： 内核服务治理

- 诉求
  - 内核侧全链路追踪
  - 内核多层次Metric
  - 分析各个环节的性能
- 功能
  - 可追踪所有读/写请求
  - 可追踪读写慢日志
  - 无缝接入MS的日志搜索，方便使用
  - 可观测到接入层/数据处理层/引擎层/存储层的metric和trace

成本：降本增效

- 资源方面：
  - 容器化迁移提升利用率
- 运营方面：
  - 计费，按需使用、按量付费
- 技术方面：
  - ILM(Rollover/Freeze/Merge/冷热分离)
  - setting/mapping 优化
  - 压缩算法/压缩块大小调整
  - 架构升级/ES 版本升级



贡献社区

- [lucene] CoveringQuery 相关： 10367
- [lucene] Memory 相关： 10627,10657,10661
- [elasticsearch] Translog 相关：82721,
- [elasticsearch] Offset 相关：86110
- [elasticsearch] CachingDirectory 相关：92440
- [elasticsearch] AsyncFetch 相关：93632, 94139
- [elasticsearch] RoutingTable 相关：94379, 94417
- [opensearch] KNN支持FAISS：285

ByteES 之

应用 在字节的应用概括和场景

架构 可用性架构和分离架构

内核 性能、可用性、安全、可观测性、功能强化、成本优化

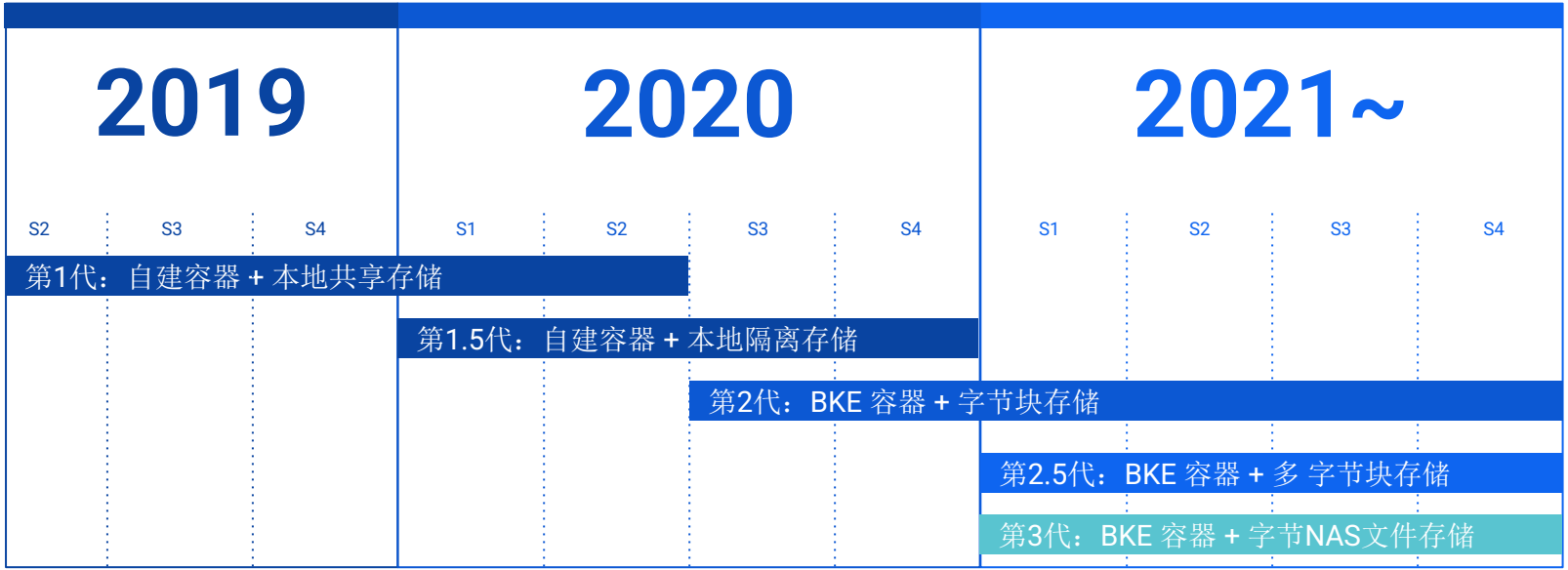
容器 ES on K8S上的一些演进和探索

生态 ES的周边生态建设

未来 未来我们准备重点投入到哪里

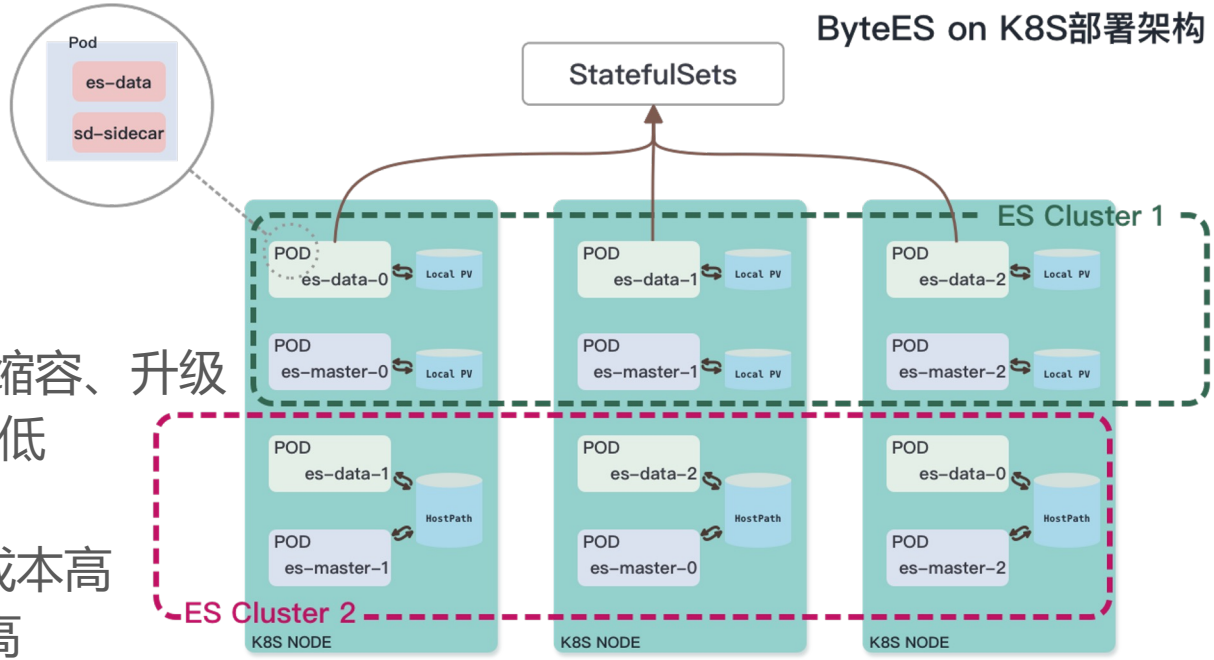
elastic

容器化架构演进


2019
2020
2021~
S2
S3
S4
S1
S2
S3
S4
S1
S2
S3
S4
第1代：自建容器 + 本地共享存储
第1.5代：自建容器 + 本地隔离存储
第2代：BKE 容器 + 字节块存储
第2.5代：BKE 容器 + 多 字节块存储
第3代：BKE 容器 + 字节NAS文件存储

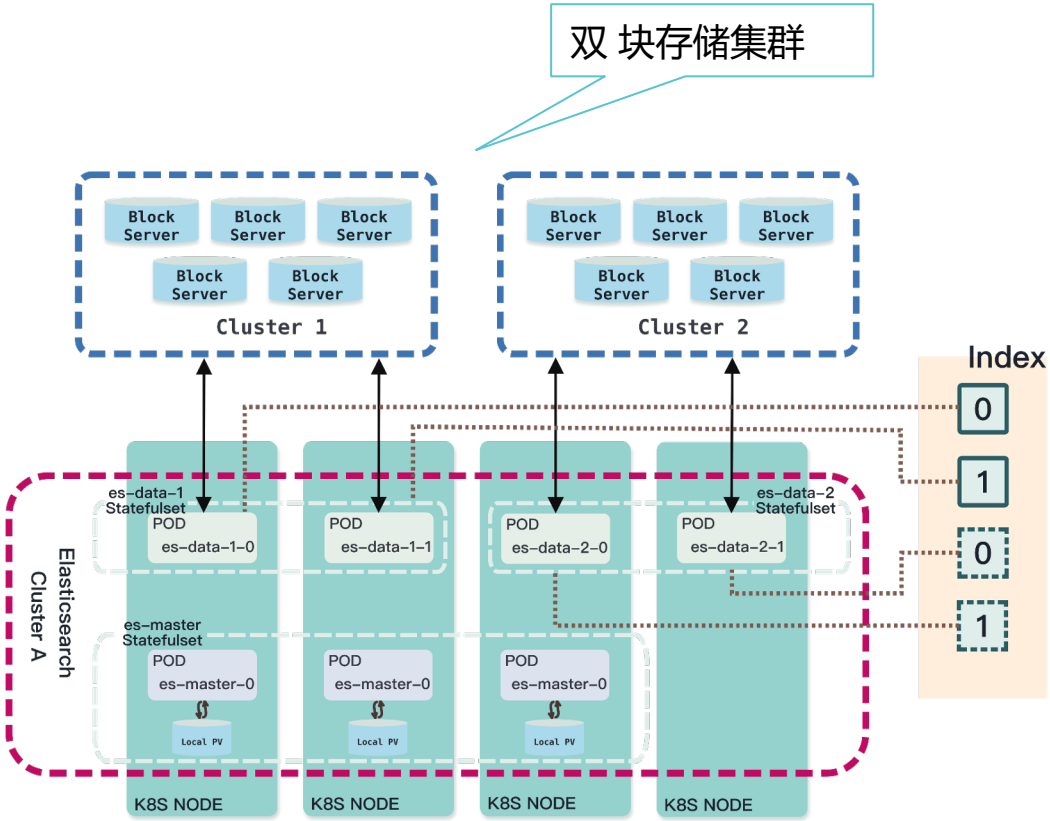
第一代：自建容器 + 本地存储

- 优点：
  - 隔离
    - 业务隔离
    - 资源隔离
    - 影响隔离
  - 快速部署、扩/缩容、升级
  - 资源消耗较VM低
- 缺点：
  - 自建K8S运维成本高
  - 机器维护成本高
  - 存储上限较低



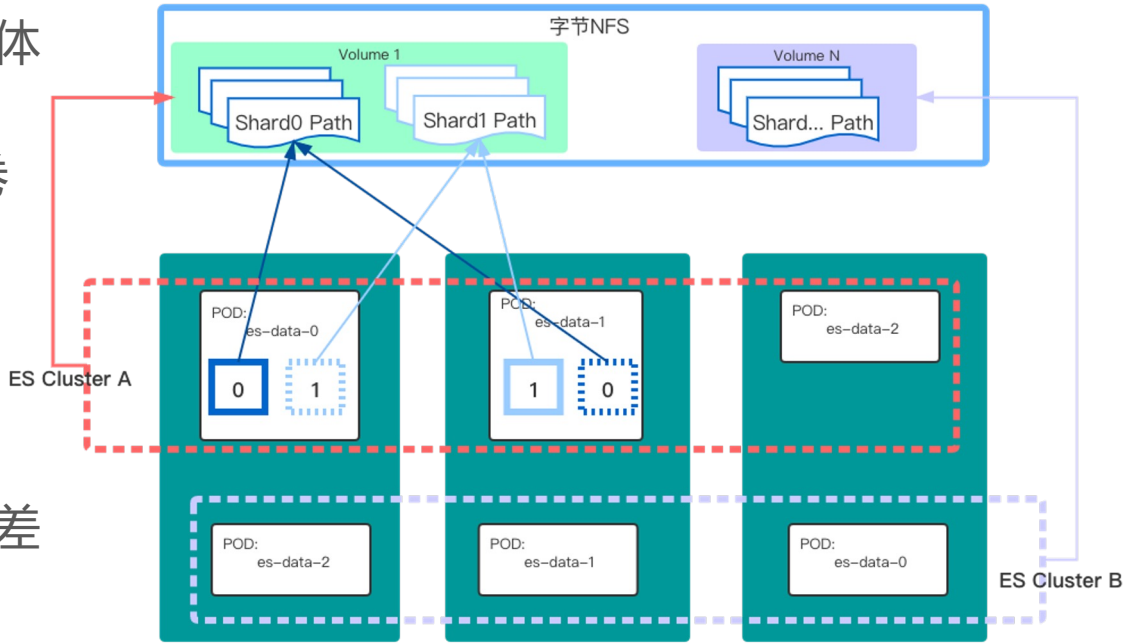
第二代：BKE容器 + 字节块存储

- 优点：
  - 运维成本更低
  - IO吞吐高
  - 资源利用率更高
  - 成本更低
  - 避免单存储集群故障
- 缺点：
  - IO 链路更长
  - 网络带宽占用更多



第三代：BKE容器 + 字节NAS文件存储

- 特点：
  - 计算存储分离架构的载体
  - NAS 支持多写多读
  - 一个ES集群对应一个卷
  - Pod 漂移不用迁移数据



- 风险：
  - 单副本性能较块存储稍差
  - 单存储集群风险

容器化后的一些思维转变

- 专属集群已是标配
  - 每个业务独占集群
  - 稳定性优先
- 索引治理 转向 集群治理
  - 集群个数将快速增长，如何高效治理
  - 快速部署
  - 不用集群的识别及下线
  - 智能扩缩容
- 滚动升级是新常态
  - 扩缩 CPU/Mem
  - 安装升级插件
  - KA业务定制化
  - POD 无感漂移

ByteES 之

应用 在字节的应用概括和场景

架构 可用性架构和分离架构

内核 性能、可用性、安全、可观测性、功能强化、成本优化

容器 ES on K8S上的一些演进和探索

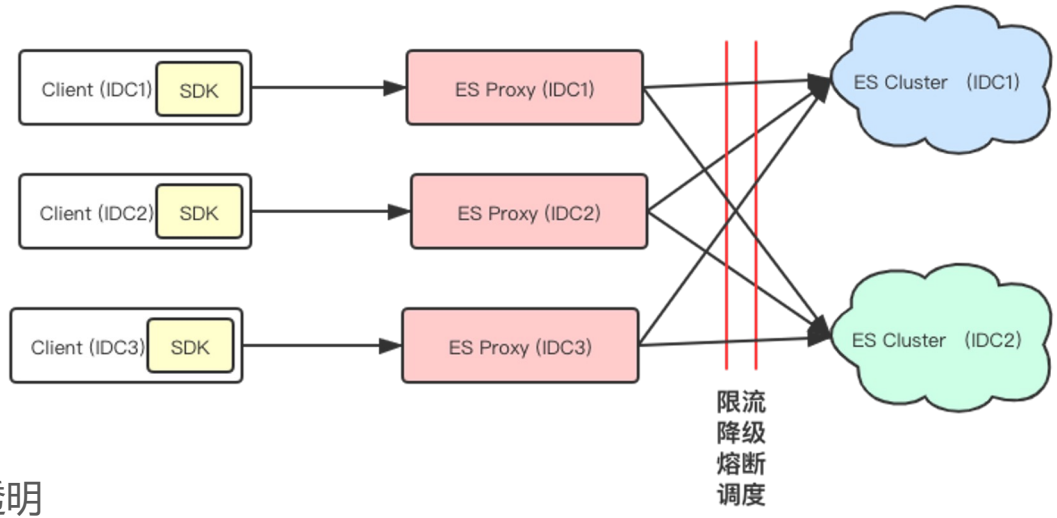
生态 ES的周边生态建设

未来 未来我们准备重点投入到哪里

elastic

流量接入&治理：ESProxy&Client SDK

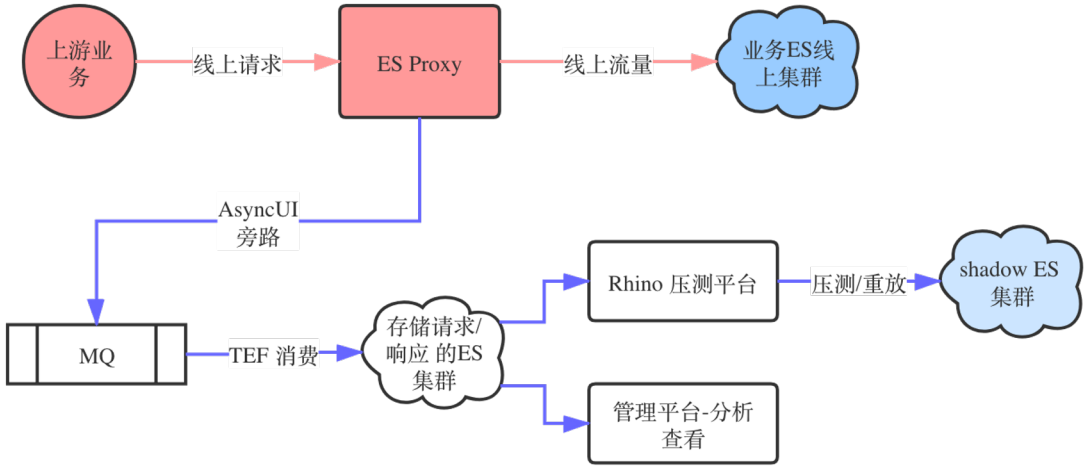
- 功能
  - 具备限流、 降级、熔断
  - 支持流量录制/压测/回放
  - 轻量级的鉴权能力
  - 完善的监控和告警
  - 全白屏化操作
- 切流/调度
  - 机房调度：自动均衡/指定比例
  - 路由：按机房/索引
  - 可自定义调度策略
- SDK 使用
  - 与ES原生API保持一致，对业务透明
  - 支持Golang/Python/Java Client SDK
  - API识别/信息注入(索引名/API/读写)
  - 全链路追踪(logID)



问题复现： 消息录制/回放/压测

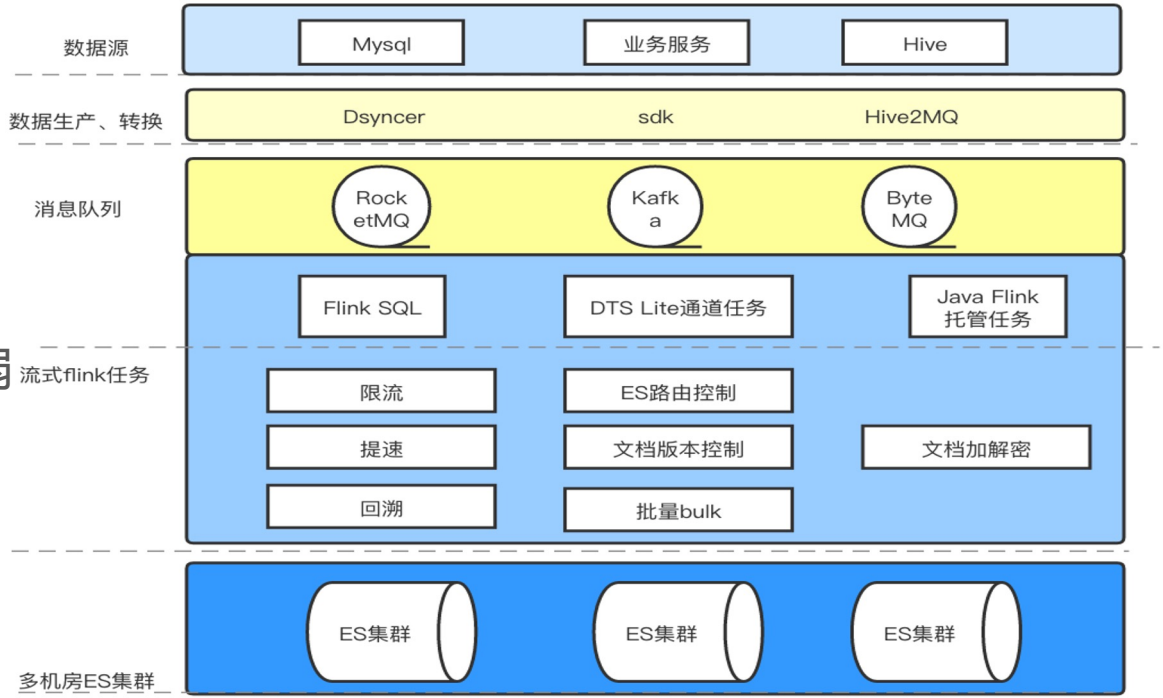
- 目的：
  - 通过查看和重放线上请求来复现线上问题，加快问题的定位解决
  - 用线上请求来压测，提供的准确的性能信息

- 功能：
  - 录制线上实际请求
  - 查询线上请求
  - 重放线上请求
  - 用线上请求进行压测



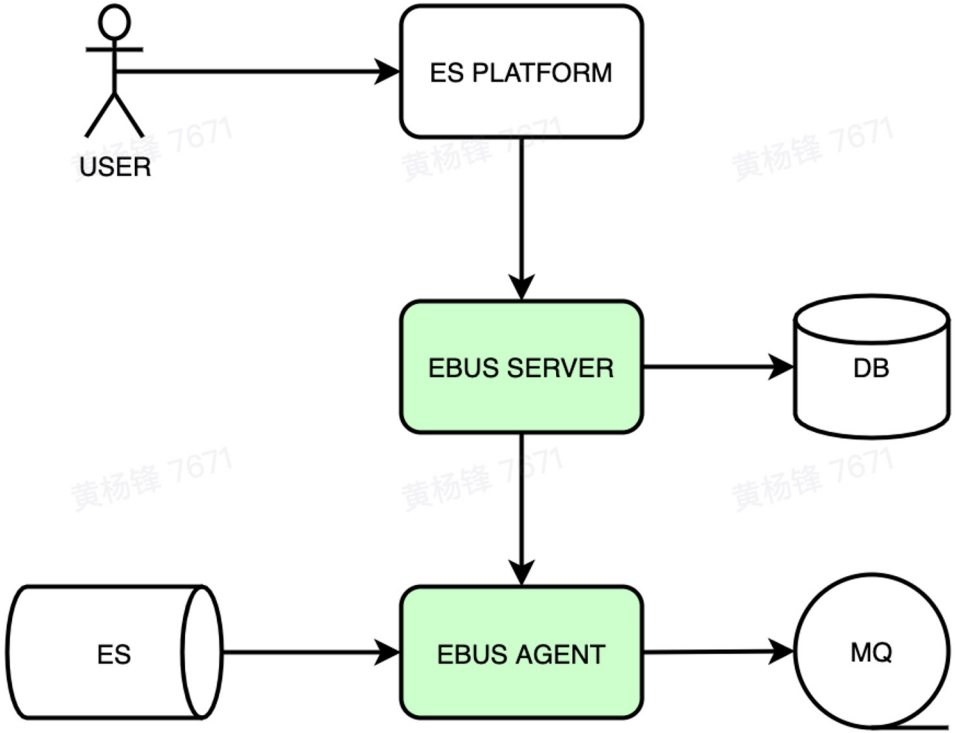
流批一体：ES Flink

- 典型场景：
  - MQ2ES
  - MySql2ES
  - Hive2ES
- 性能：
  - 提前bulk分组，降低扇出。
  - 去掉translog
  - 去掉压缩
  - 可配压缩块
- 功能：
  - 限流/提速/回溯



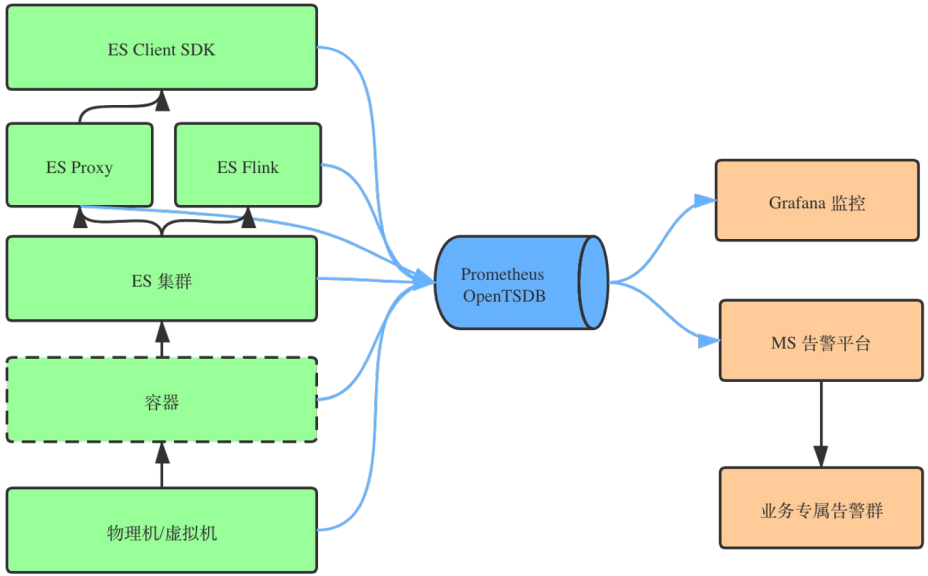
非侵入式的CDC：EBus

- 介绍
  - ES的数据导出组件，可持续获取数据的变更，并将其发送至消息中间件 (MQ)
  - 基于日志，非侵入式
- 适用场景：
  - 对账
  - 跨区域的数据同步
  - 离线分析
  - 实时备份
- 功能：
  - 支持导出存量
  - 持续获取增量
  - 支持索引粒度



监控告警： 全栈式/千人千面

- 监控：
  - 所用组件均在监控之中
  - 涵盖metric生成、采集、存储、转存、分析、展示 等全链路功能
  - 区分业务视图/平台视图
  - 各类 监控大盘
- 告警：
  - 零门槛配置
  - 实现告警自动注入
  - 支持自定义告警
  - 支持自定义告警接收人/告警值班模版
  - 专属告警群，让告警直达业务
- 健康检查： 定期巡检/异常告警
- 运营周报



ByteES 之

应用 在字节的应用概括和场景

架构 可用性架构和分离架构

内核 性能、可用性、安全、可观测性、功能强化、成本优化

容器 ES on K8S上的一些演进和探索

生态 ES的周边生态建设

未来 未来我们准备重点投入到哪里

elastic

未来

架构

持续演进

- 读写分离架构
- 冷热分离架构
- 离在线混合
- Serverless

功能

更快更强

- 滚动升级：无感/快速
- 成本：超卖
- Join: 性能提升
- Range Shard
- 向量检索：加强
- POI：性能优化

探索

前瞻性

- CPU 和 Mem 分离
- ES on FPGA 软硬结合
- 基于机器学习的语义搜索

elastic 中国开发者大会 2023

感谢观看

专业、垂直、纯粹的 Elastic 开源技术交流社区

https://elasticsearch.cn/