索引隔离与有赞的实践

hehua@youzan.com

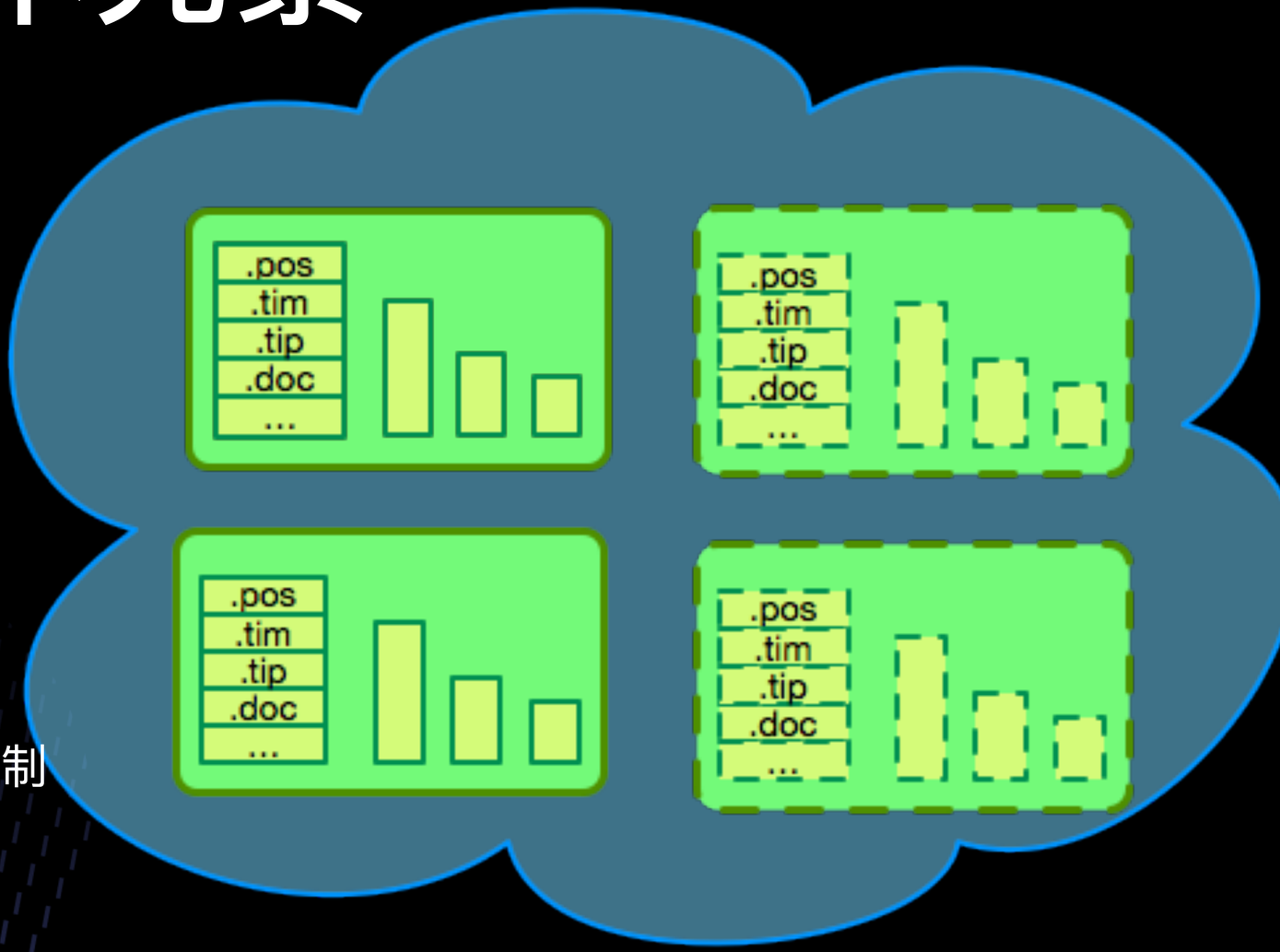
Outline

- 自底向上看索引
- 索引在不同级别的隔离
- 有赞在隔离上的实践
- 总结/展望

Youzan.com

基本元素

- 段：segment，索引的基本工作单位
- Lucene索引：段文件集
- 分片：等同于一个Lucene索引
- Elasticsearch索引：若干分片的数据集
- 主从副本：是独立的Lucene索引，ES来控制数据双写



隔离级别

级别	原生方式	我们的方式
段隔离	不改代码就算了吧	
分片隔离	routing	
索引隔离	allocation/rollover	外部路由
集群隔离	cross cluster search	proxy请求转发/cross cluste search
机房隔离	allocation	异步复制

分片隔离

- 普通查询
  - 请求发送至全部分片
  - 聚合各分片数据后返回
- 指定routing查询
  - 根据routing路由到指定分片
  - 单分片结果返回

权衡

- 分布式的边界
  - 分布式系统中单机工作任务量一般小于单机系统
  - 考虑数据聚合时间开销是否大于单机遍历其他数据的开销
- 数据倾斜
  - 某一个分片上的执行时间明显高于其他分片，毛刺严重

索引隔离

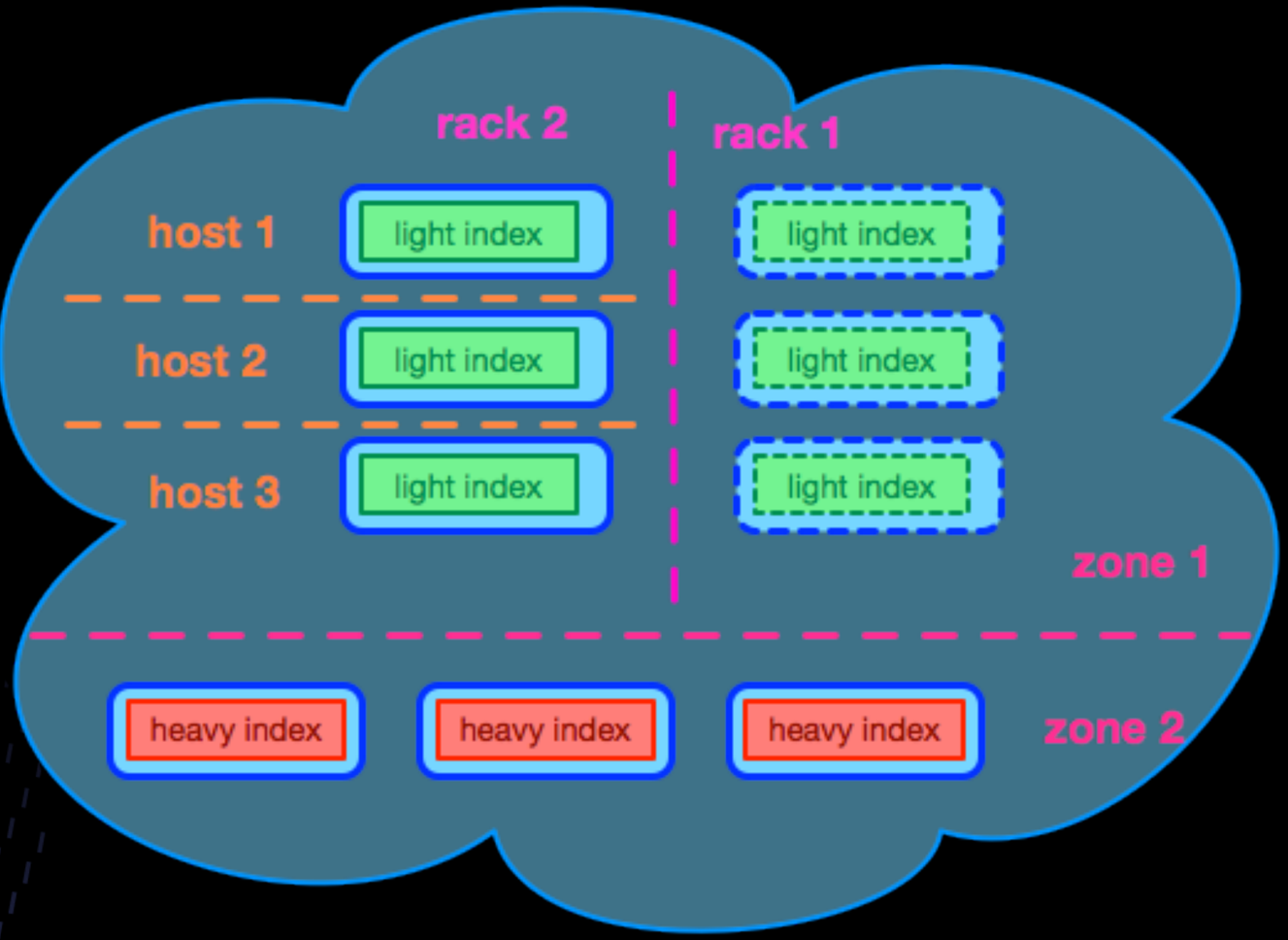
- 索引分布
  - 根据disk watermark等计算候选节点
  - 在候选节点中随机分布
  - 之后通过启发式算法rebalance
- 资源抢占
  - 重索引影响轻索引访问
  - 流量激增-io/慢查询-cpu/缓存-mem

索引隔离

参数/命令		用途	用法
index.routing.allocation	include exclude require	将索引分布锁定在带有指定属性的若干个es node中或者排除若干node	需要搭配node.attr或者内置属性使用
	total_shards_per_node	限制在单个es node中最多能分布的分片数，避免过多分片分布在同一台机器	
cluster.routing.allocation	same_shard.host	限制同一个分片的主从副本不在同一host中出现，主要出于可用性考虑	
	awareness.attributes	在有条件情况下隔离索引的主从副本	需要搭配node.attr使用
	awareness.force.*	强制隔离索引的主从副本在不同区域	需要搭配node.attr使用
_rollover		滚动索引，适用于时间序列相关的数据，可以冷热隔离	

利好

- 减少资源竞争
  - 缓存—query cache节点级别共享
  - io/cpu—避免异常流量索引对其他索引的影响
- 冷热隔离
  - 不同热度的索引分配在不同分组，使用不同的硬/软件配置，节约成本
- 稳定性提升



不足

- 运维成本
  - node.attr配置需要重启
  - 不同机器的attr可能不一致，不能一体更新
- 共享master/client
  - shard数膨胀可能引起元数据更新慢，小索引变更堵塞全局元数据更新
  - client抖动也会影响全局访问

集群隔离

- 按业务核心级别隔离集群
- 按不同的业务场景隔离集群

集群隔离

读		写
tribe node	感知并聚合多集群metadata，会对元数据更新带来压力	写入数据通过连接不同的cluster来做
cross cluster search	不作为node加入remote cluster，轻量级访问	

优劣

优势

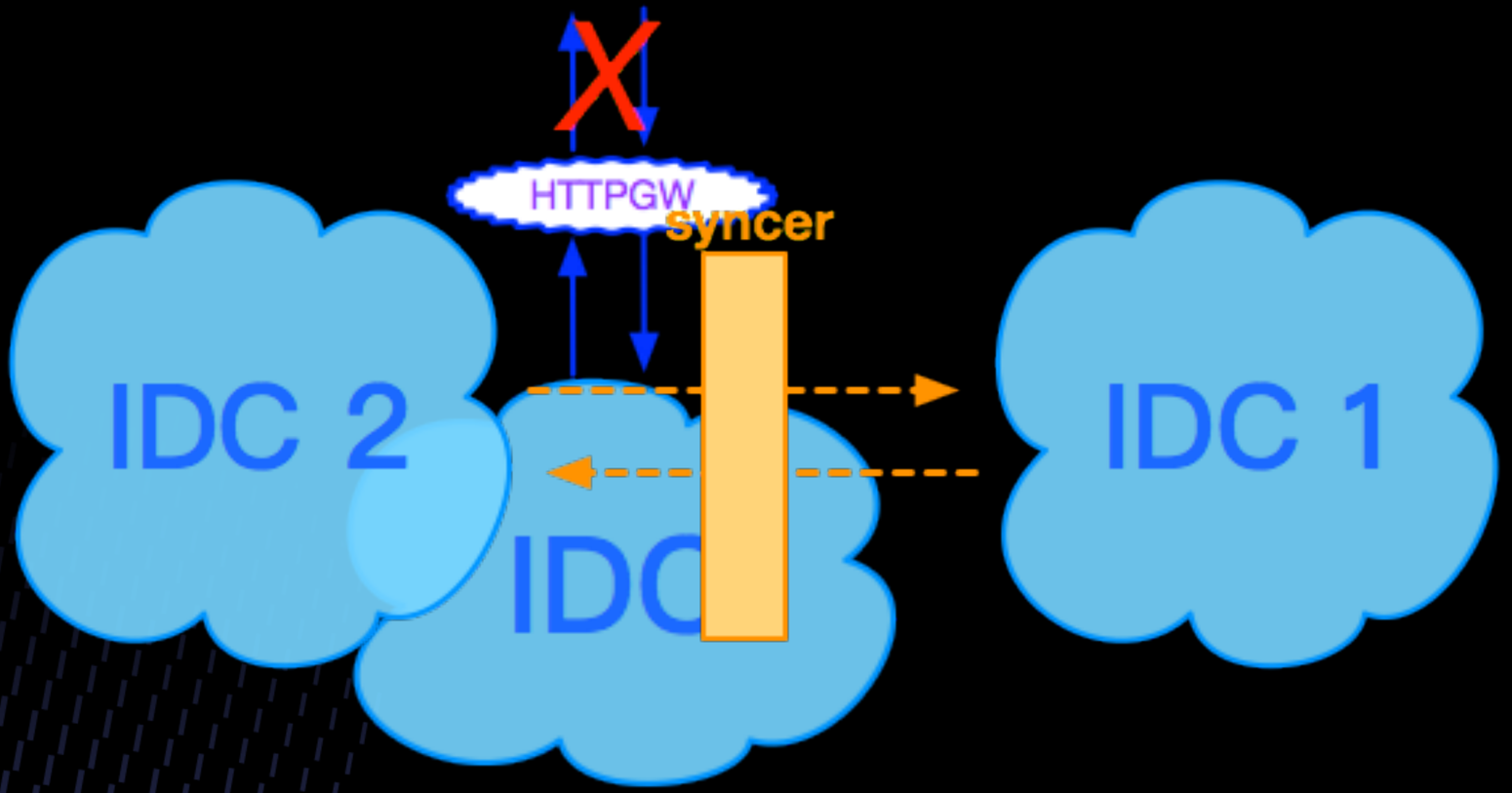
- 稳定性提升
- 用户体验提升
  - 可以针对不同业务场景优化集群配置

劣势

- 运维成本
  - 集群增多，配置增多

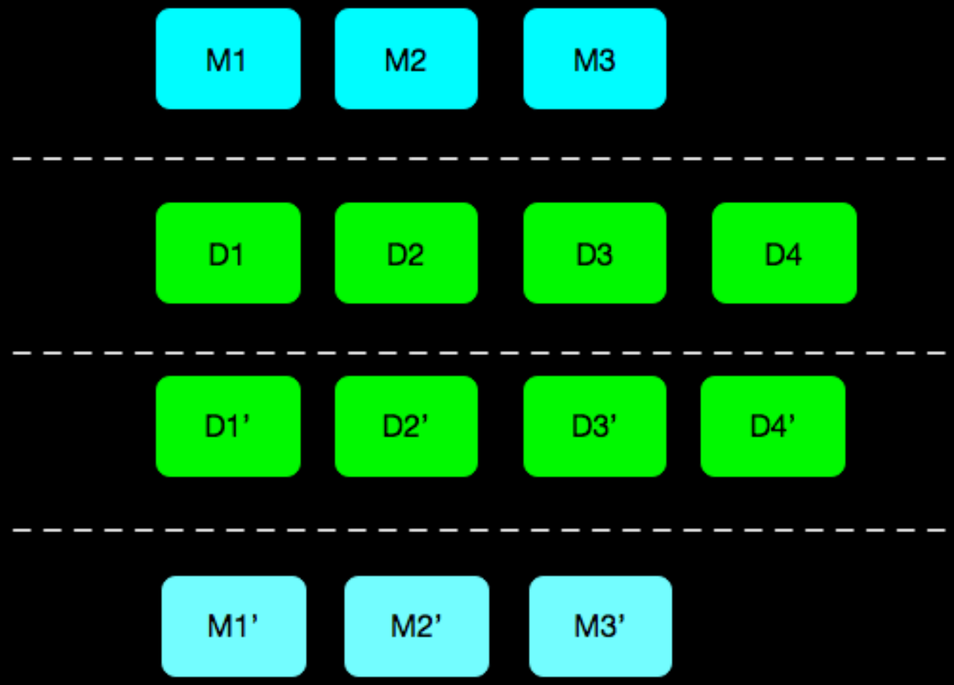
机房隔离

- 机房整体出入口受阻/掉电
- 机房内部分区域掉电
- 内部硬件故障
- …



机房隔离

跨机房集群	通过 cluster.routing.allocation.awareness 配置主从副本在不同机房
xpack’s cross-cluster replication	beta版本，不做深入

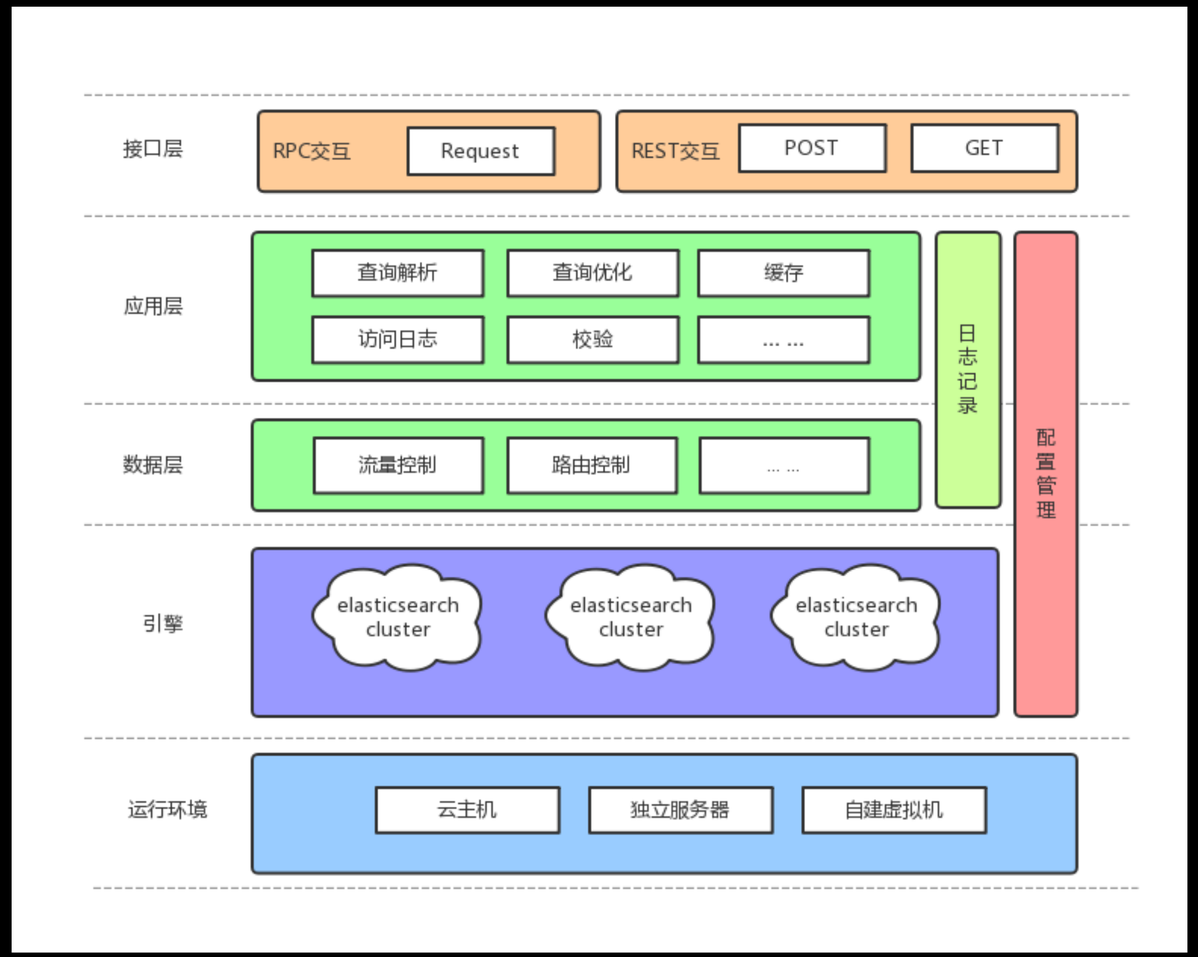


有赞的实践

Youzan.com

隔离方式

- 通过proxy转发全部请求
- proxy内部路由，索引拆分/隔离不需改变使用方式
- 业务不需感知隔离级别

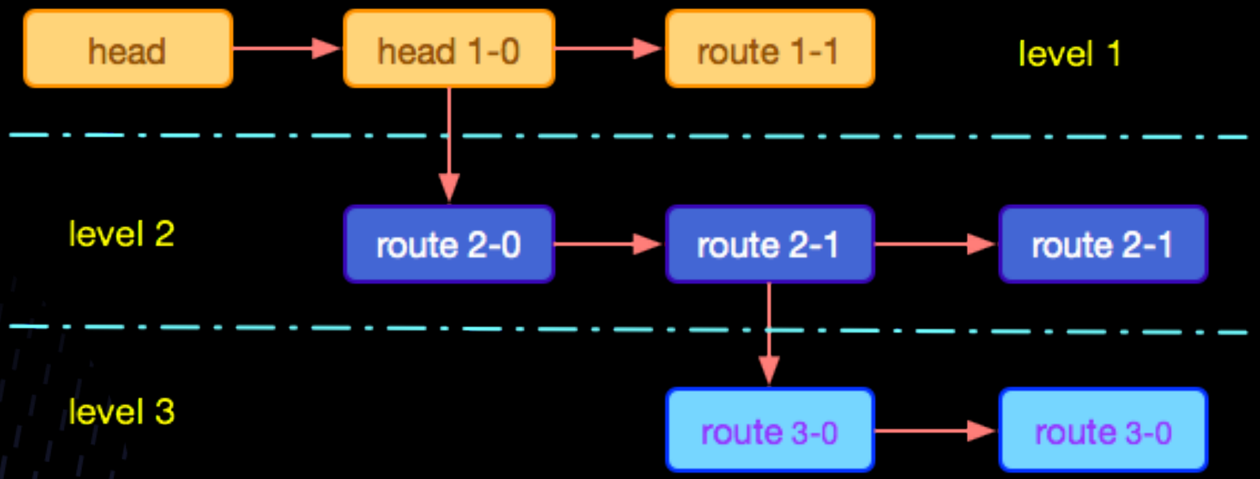


隔离策略

- 索引隔离
  - 业务间多租户，业务内索引拆分
- 集群隔离
  - 业务核心级别L1/L2/L3，业务使用场景
- 机房隔离
  - 独立集群，自研同步工具

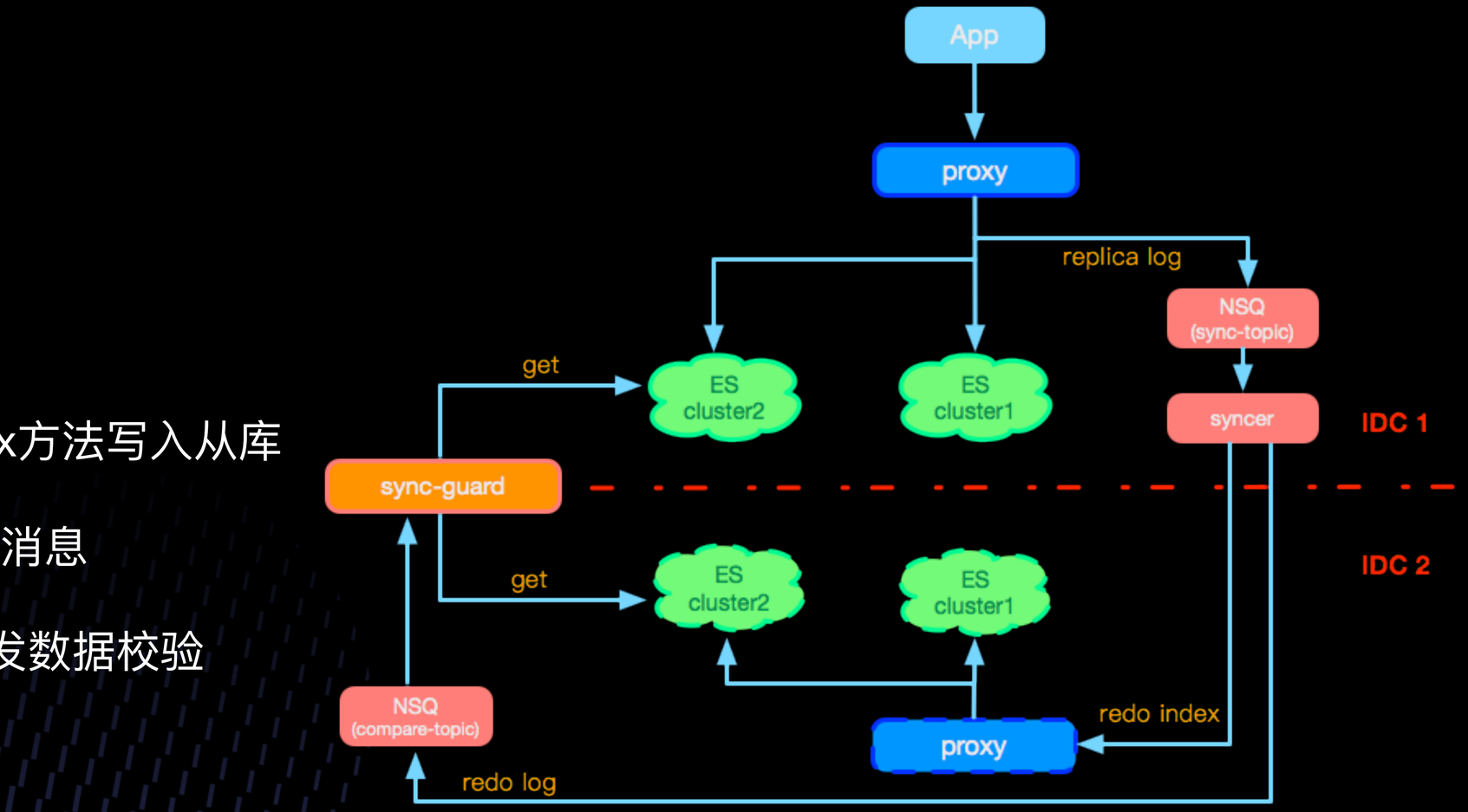
索引隔离

- 多租户
  - 通过index allocation filter实现
- 索引拆分
  - user_based/time_based/whitelist/compound



机房隔离

- proxy发送复制消息
- syncer消费消息并通过index方法写入从库
- syncer写入完成后发送redo消息
- guard负责通过redo消息触发数据校验



总结

- 隔离的作用
  - 降低资源竞争
  - 避免被动影响
- 更好的隔离
  - 容器化/PaaS化

YOUZAN power

专业、垂直、纯粹的 Elastic 开源技术交流社区

https://elasticsearch.cn/