

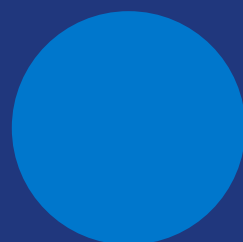


# 阿里云Elasticsearch内核持续优化

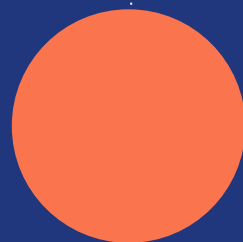
---

杨孔仕，阿里巴巴技术专家  
2023/4/8

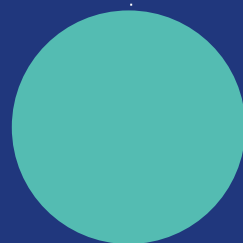
# 分享嘉宾



业务发展



泛日志引擎架构演进



自研引擎内核通用增强  
(Apack)



1w+

集群数量



7w+

节点数



20PB+

数据量



## 阿里云Elasticsearch产品架构

多模态搜索			云原生全观察与分析				
AliWS分词	向量检索	DB检索加速	日志增强版	Indexing Service	Open Store	高级监控报警服务	云原生数据采集

Elastic Stack产品组件	X-Pack商业特性	内核增强 高性能、低成本、稳定性增强	
<ul style="list-style-type: none"> <li> <b>阿里云Elasticsearch</b> 100%兼容开源，一键部署，开箱即用</li> <li> <b>阿里云Logstash Service</b> 在数据管道中进行处理、过滤、传输</li> <li> <b>Kibana可视化</b> 灵活可视化，搭建数据报表分析大盘</li> <li> <b>多Beats数据采集中心</b> 从多个来源轻量化的采集各类数据</li> </ul>	<ul style="list-style-type: none"> <li> <b>安全与权限</b> 支持各类授权认证和角色权限管控</li> <li> <b>机器学习</b> 对数据实时建模，实现根源分析和预判</li> <li> <b>监控告警</b> 集中管理监控指标，便捷实现异常告警</li> <li> <b>APM/SIEM</b> 应用性能监控和安全场景解决方案</li> </ul>	<ul style="list-style-type: none"> <li> <b>通用物理复制</b> 有副本索引的写入性能提升</li> <li> <b>自动热点均衡</b> 解决负载不均，提升CPU利用率</li> <li> <b>主键查询去重</b> 带有主键文档的写入性能提升</li> <li> <b>写入请求聚合</b> 提升写入吞吐，降低写入拒绝</li> </ul>	<ul style="list-style-type: none"> <li> <b>索引压缩</b> 更高索引压缩比，降低存储成本</li> <li> <b>慢查询隔离池</b> 单条查询的开销追踪和逻辑隔离</li> <li> <b>协调节点流控</b> 流量调度和异常机器秒级切流</li> <li> <b>集群Qos限流</b> 支持节点索引的读写限流</li> </ul>

高级管控和智能运维平台			
<ul style="list-style-type: none"> <li> <b>容灾部署</b> 一键完成多可用区部署，通过高可用架构保证服务稳定性</li> </ul>	<ul style="list-style-type: none"> <li> <b>冷热分离+计算存储分离架构</b> 支持集群hot-warm分离和计算存储分离架构，实现成本和性能的双重保障</li> </ul>	<ul style="list-style-type: none"> <li> <b>弹性伸缩</b> 集群灵活扩缩，热重启平滑变更，轻松应对业务峰谷波动</li> </ul>	<ul style="list-style-type: none"> <li> <b>智能诊断和监控</b> 全面监控、分析集群多维度潜在风险，提供异常分析和优化建议</li> </ul>

## 架构演进 - 云上用户痛点



云原生很重要的一个特征是弹性

### 负载不均

- 迁移热点分片带来大量数据拷贝

### 数据副本

- 多副本带来重复计算和存储

### 数据搬迁

- 节点扩缩容带来的数据搬迁

# 架构演进 - 计算存储分离



## 写入性能提升100%

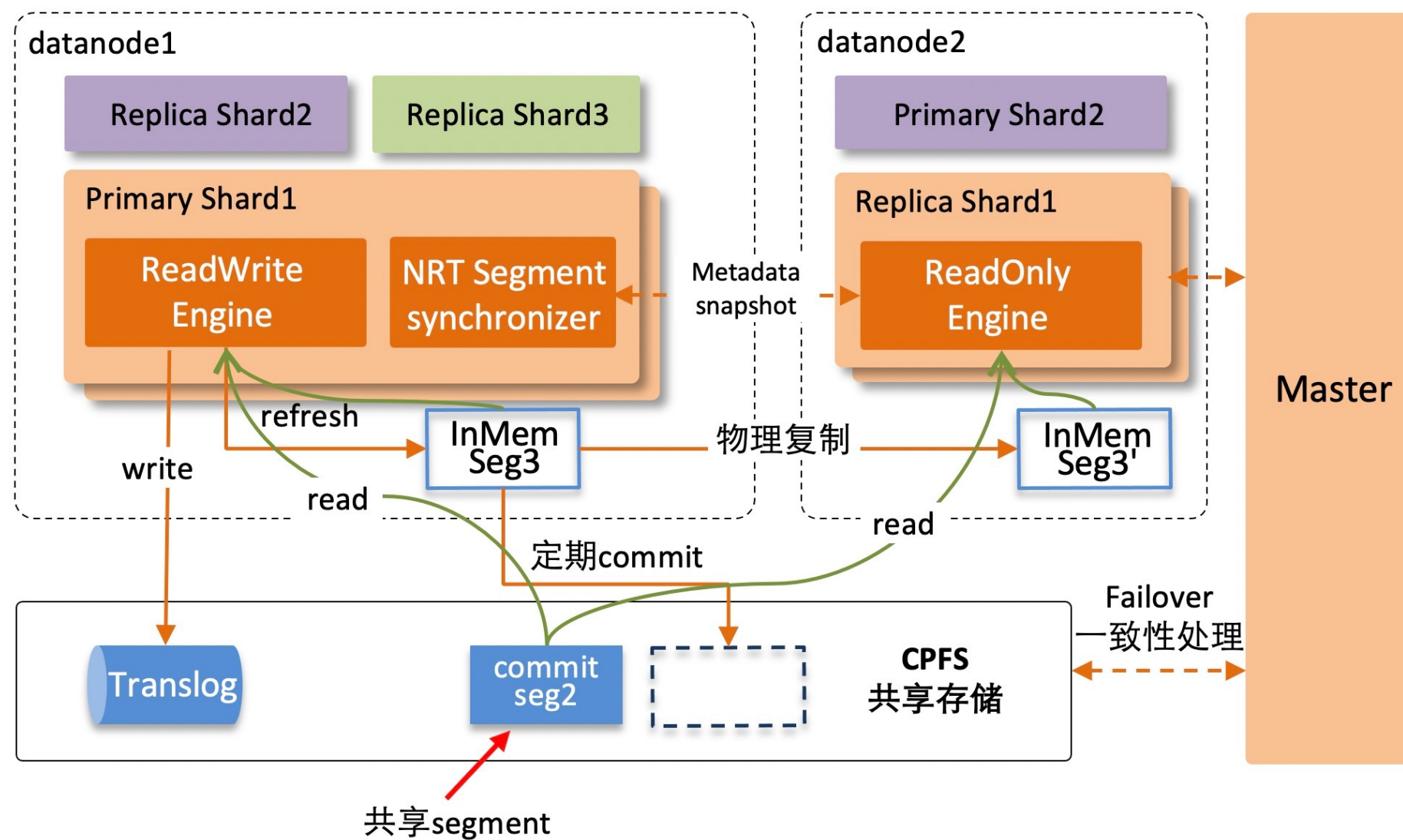
计算上避免了副本写入的CPU开销

## 存储成本倍数级降低

业务数据只存储一份

## 秒级弹性扩缩容

扩缩容无需搬迁数据，轻松应对高峰流量



# 架构演进 - 云上用户痛点



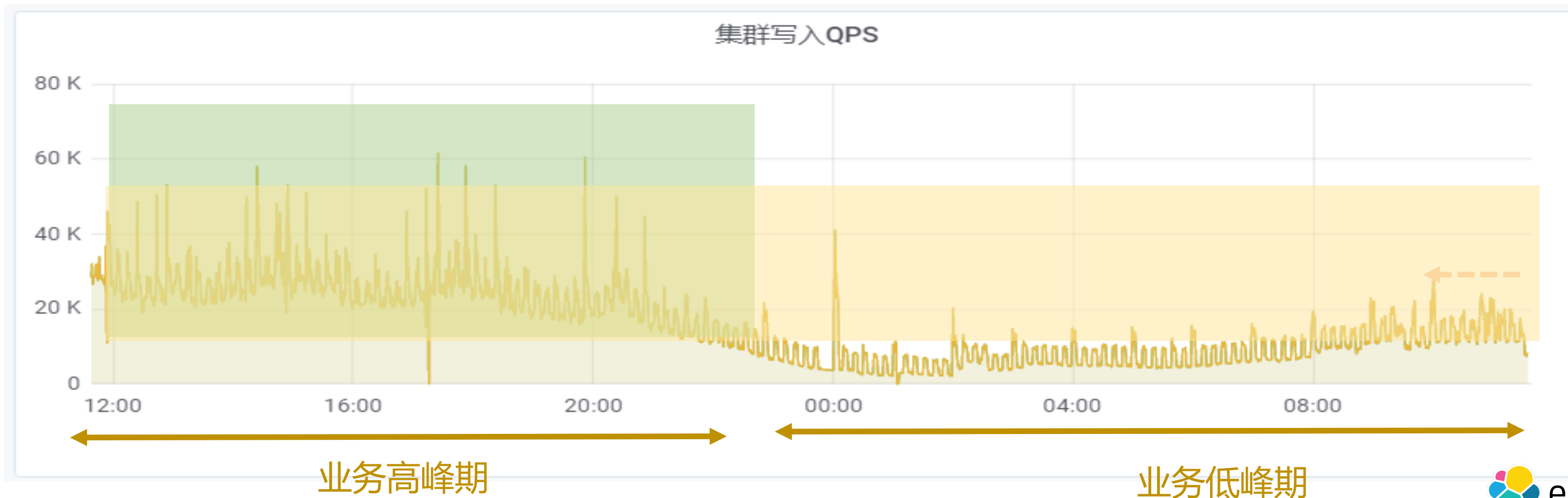
云上用户的用云成本不仅仅只是账单费用

**性能调优**  
写入配置优化

**预留机器成本**  
应对流量洪峰

**集群运维**  
业务增长扩容

**稳定性保障**  
异常流量保护



# 架构演进 – Indexing Service



## 低成本

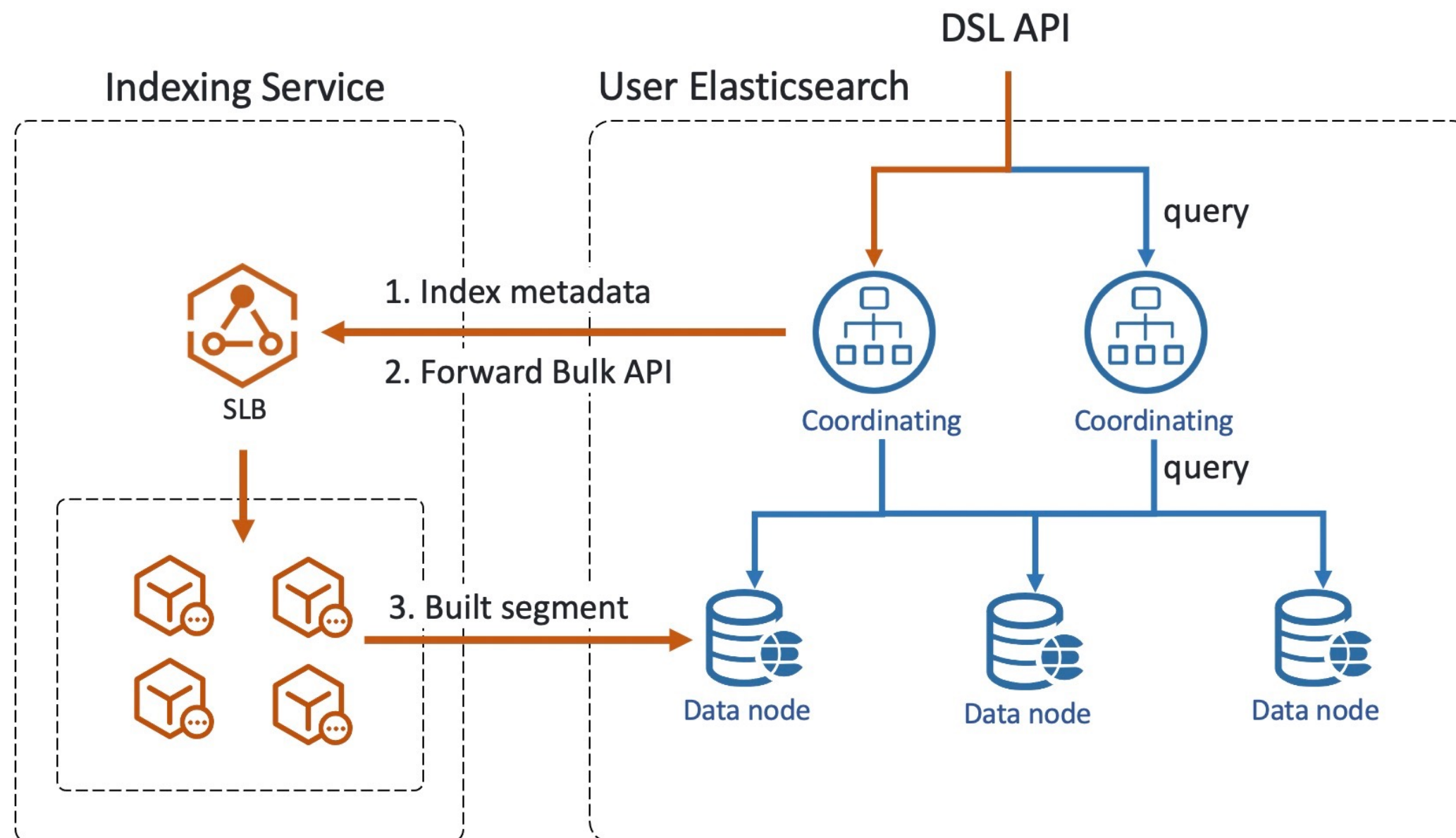
云端写入serverless架构，按实际写入流量计费，的极致的索引构建性能，通过自研索引结构、物理复制、优化机型性价比等手段，索引构建性能相比原生提升200%  
降低 70%索引构建成本

## 弹性扩展

写入资源由云端后台调配和管理，无需提前预留资源，极致的弹性扩展能力

## 免运维

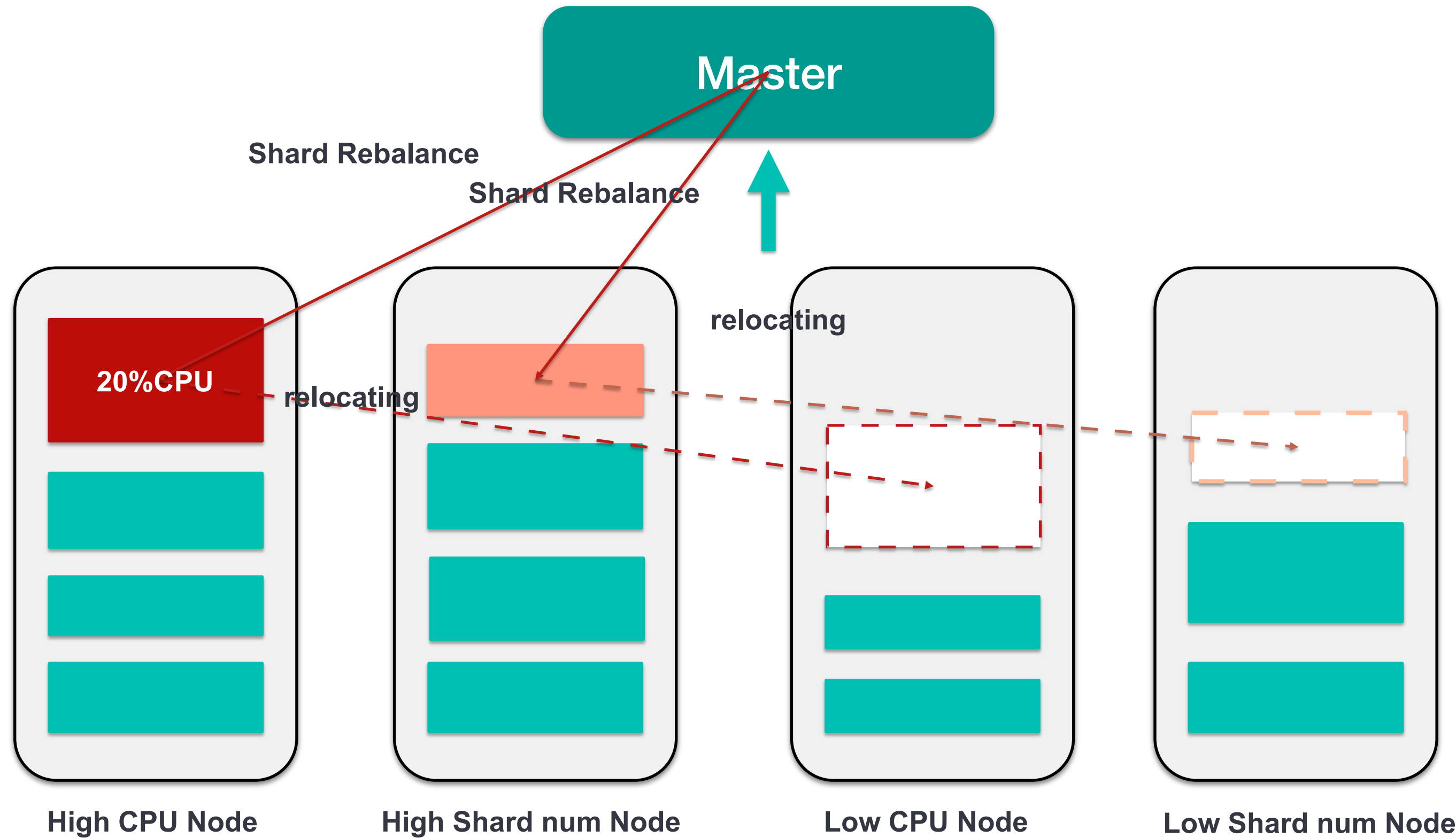
零学习成本，完全适配ES原生使用方式，无需关注写入资源和写入压力







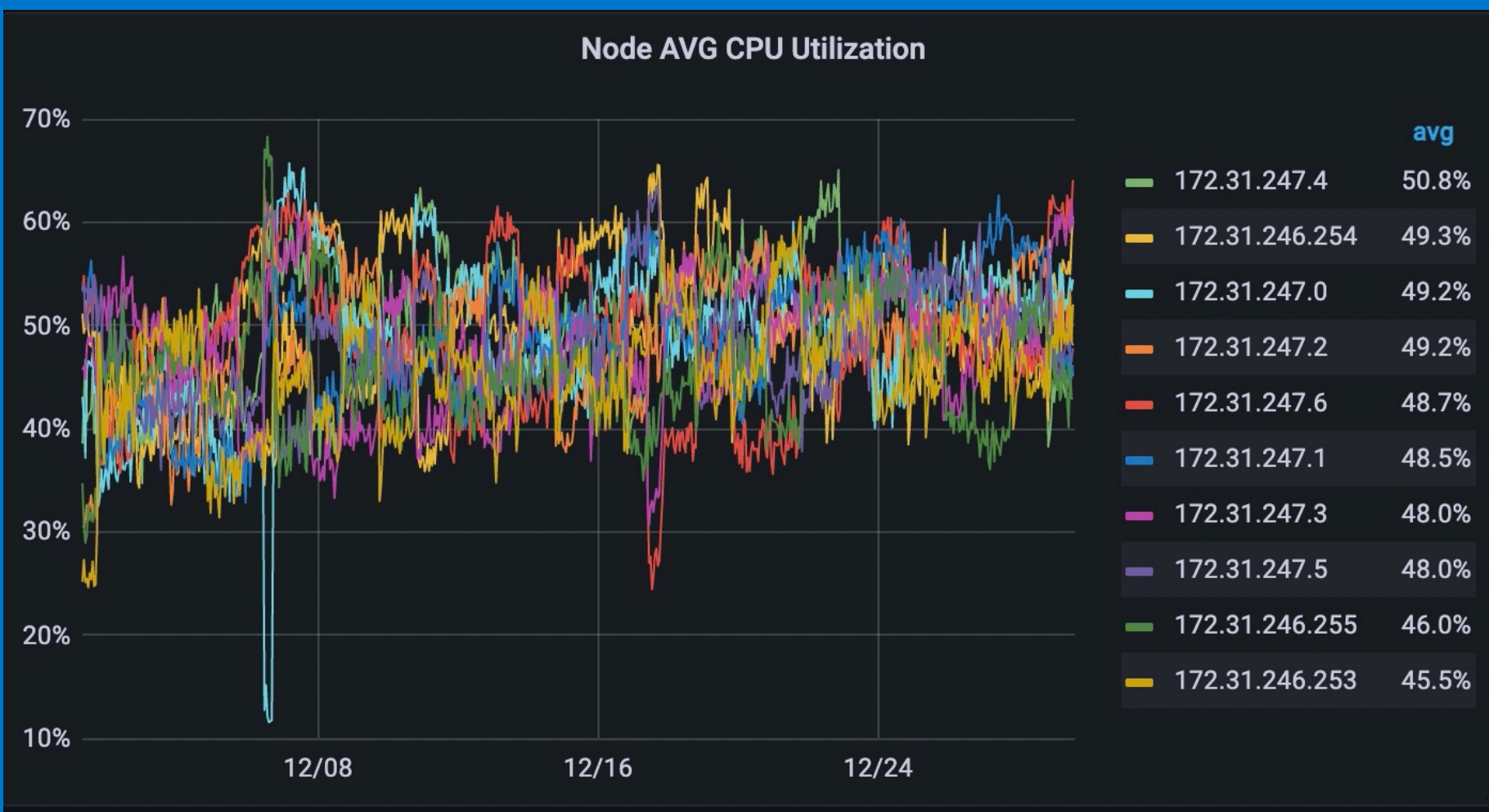
# 构架演进 – Indexing Service自动热点均衡



# 架构演进 - 自动热点均衡



线上用户集群经过自动负载均衡优化后，CPU平均利用率从 **20%** 提升至 **50%**



## 架构演进 - 云上用户痛点



日志场景更看重成本，存储成本占比在50%以上



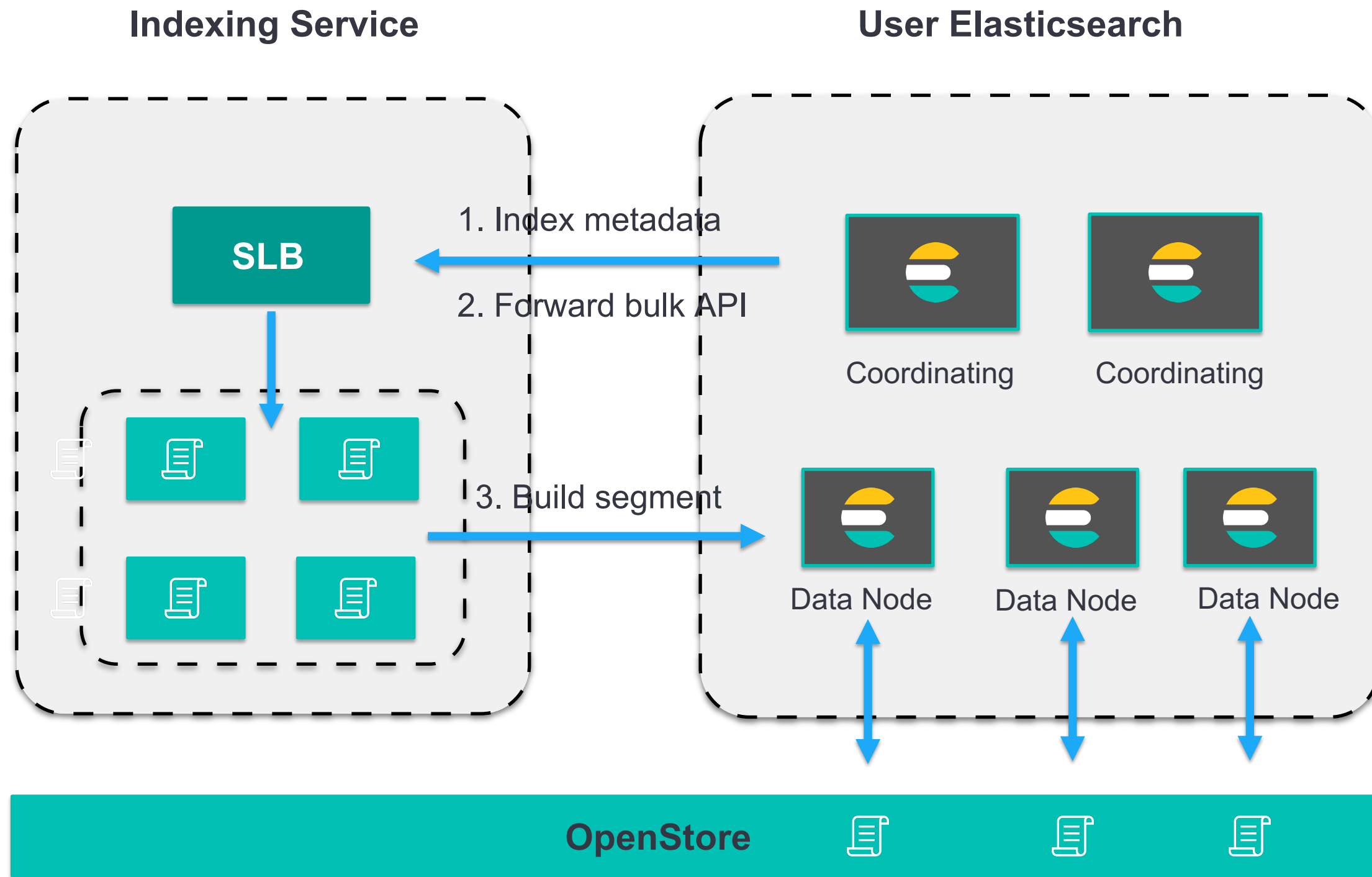
# 架构演进 – Open Store

## 高性价比的日志存储服务OpenStore

- 存算分离架构：基于OSS降低冷数据存储成本
- 极致的存储压缩优化：索引大小优化70%
- 性能优化：针对日志场景写入和查询分别定制优化

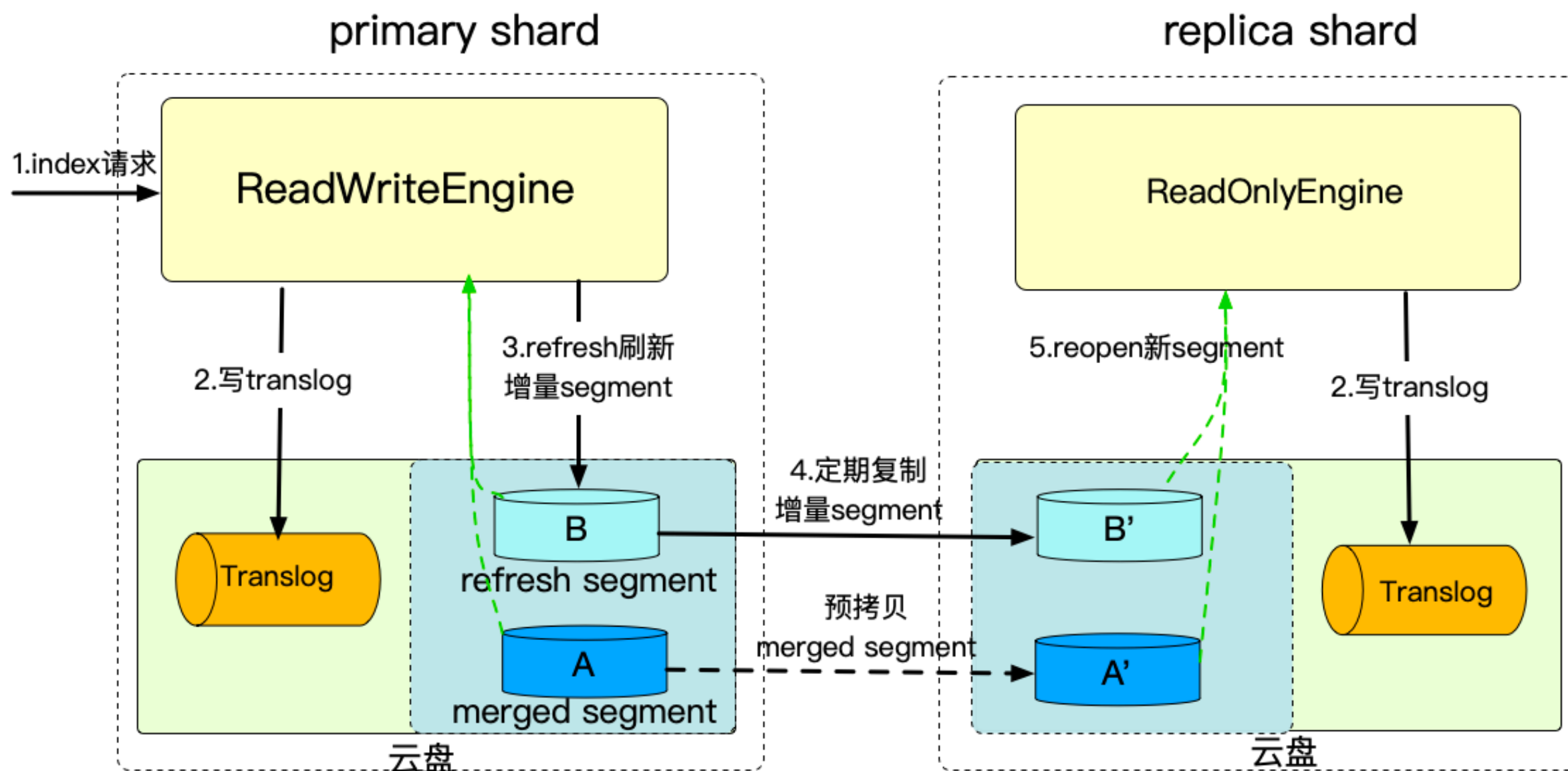
## 开箱即用的Serverless服务

- 定制索引存储服务：
  - 用户可按需购买：降低冗余的存储成本
  - 智能冷热分离：降低用户区分冷热的运维成本



# 内核优化 – 物理复制

写入性能提升50%，CPU开销降低35%



## 内核优化-索引压缩

日志索引编码压缩算法优化前后对比

Filter for system indices

compress\_optimization

Name ↓	Status	Document Count	Data	Index Rate
without_compress_optimization	● Green	1.7b	857.5 GB	14,919.78 /s
with_compress_optimization	● Green	1.7b	203.2 GB	14,917.66 /s



优化索引和原文的压缩编码算法，索引大小相比原生 ES 优化 **70%**

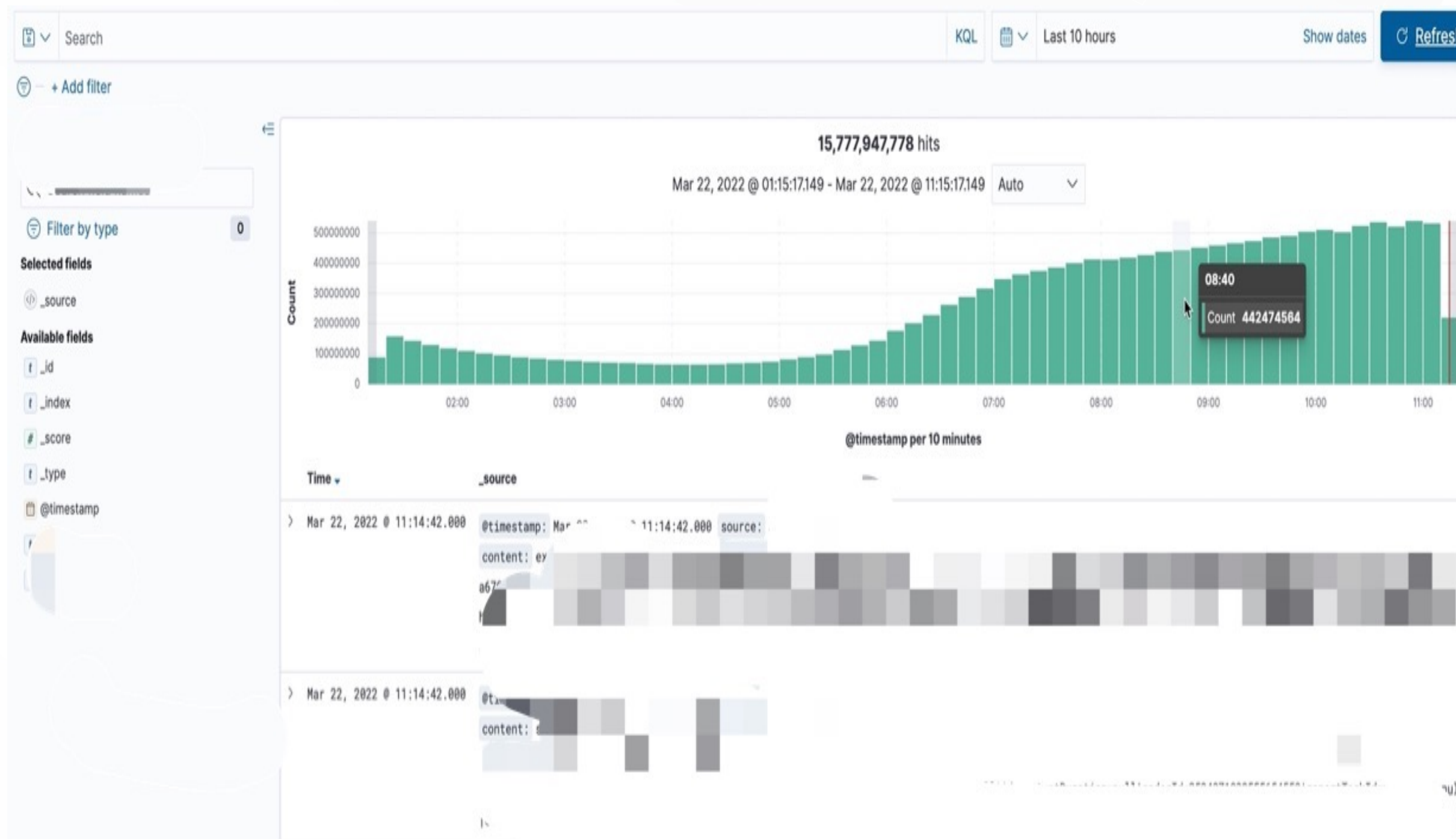


# 内核优化 – AnalyticSearch



Discover查询瓶颈：百亿数据分钟级返回优化至秒级

- 优化索引合并策略
- 查询剪枝，优化 date\_histogram 执行计划
- 并发查询，成倍提升计算资源使用效率



## Discover查询瓶颈：百亿数据分钟级返回优化至秒级

### Kibana Discover查询加速功能

- 适用场景：日志检索场景。例如Kibana Discover中的无条件或单条件查询。
- 功能优势：通过优化索引合并策略及Date\_histogram执行计划策略，在没有查询条件或者仅有一个查询条件时，大幅度降低查询耗时（日增量>1 TB数据量的场景下，分钟级查询可优化至5s内）。
- 性能测试信息：
  - 测试环境
    - 节点：10 \* 16核64 GB。
    - 数据集：某业务日志数据，一天600亿个文档，分成12个索引，每个索引60个分片。
  - 开启Kibana Discover查询加速后，不同存储介质的性能表现如下：

查询条件	SSD云盘	高效云盘	OpenStore存储
没有查询条件	耗时降低96%	耗时降低95%	耗时降低94%
一个查询条件	耗时降低88%	耗时降低77%	耗时降低85%
多个查询条件	耗时降低8%	耗时降低11%	耗时降低14%

### 开启Kibana Discover查询加速

创建索引时，在settings和mappings中增加如下配置即可。

🔍 说明 以下示例仅供参考。实际使用时，您需要按照业务实际字段名称调整index.sort的时间戳字段名和排序。

```
{
  "settings": {
    "index.points.same_sort_order_as_index_sort": true,
    "index.sort.field": [
      "@timestamp"
    ],
    "index.sort.order": [
      "desc"
    ]
  },
  "mappings": {
    "properties": {
      "@timestamp": {
        "type": "date"
      }
    }
  }
}
```





感谢观看

---



专业、垂直、纯粹的 Elastic 开源技术交流社区

<https://elasticsearch.cn/>