elastic 中国开发者大会 2023

腾讯云大数据 Elasticsearch 服务自治探索实践

陈曦
腾讯，2023/04/08

分享嘉宾

陈曦

- ES 在腾讯的发展
- 自治索引
- 智能诊断
- NEXT

腾讯云 ES 发展历程


内部云建设
• 时序场景应用（腾讯云监控）
• 可用性：区间平滑限流
• 成本：Rollup (成本数量级下降)
公有云CES发布
• 内部云单集群 PB 级 500+节点
• 运营系统建设（公有云）
• 内核 Bug 持续修复
- 分布式锁
- 大量重复任务压垮 master
- 大规模集群长时间无法选主
- 节点反复加入、离开
.....
Elastic商业合作
• X-Pack 商业套件
• 专有云适配
• 可用性：
Breaker 增强，全链路限流
• 性能：
- 普通写入提升 20%+
- 带主键写入提升 50%+
- 聚合查询提升 7倍+
- 查询毛刺降低 10倍+
规模化运营
• 百 PB 规模
• 内部增速 300%+
• 外部增速 500%+
• 扩展性：千级节点、百万级分片
• 成本：高效压缩算法、编码优化
扩生态，强内核
• 完善 Elastic Stack 产品生态
• 闭环 PB 级数据处理方案
• 智能诊断
• 可用性：均衡算法、自治索引
• 成本：
- 存算分离
2017
2018
2019
2020
2021

腾讯云 ES 产品形态

公有云

产品丰富

Elasticsearch Services
云监控
云日志

内部云

超大规模

公有云产品日志底座
内部开源协同
千万级写入，十万级查询

私有云

标准化、自动化

完全隔离的环境
标准化交付
自动化运维

腾讯云 ES 服务自治为什么？

服务接入

接入门槛
概念原理、参数设置丰富，接口复杂

实例管理
实例监控、组件管理、容量评估

数据管理
生命周期、数据排布

使用调优

成本
集群伸缩
数据压缩、归档

性能
索引调参
业务适配

稳定性
大规模
高压力

业务运营

故障频率
线上故障多

运营效率
故障分析、定位慢

变更效率
发布升级慢

专家系统：业务团队、运维、DBA、内核研发

腾讯云 ES 服务自治怎么做？

接入层

SaaS 化体验

接入、使用简单

Serverless

实例、数据管理
全托管

平台层

运营管理

基础托管
自动化运营

故障诊断

自发现、自定位、
自恢复

智能调优

成本、性能
智能 DBA

内核层

健壮性架构

服务限流、异常容忍
扩展性

容灾方案

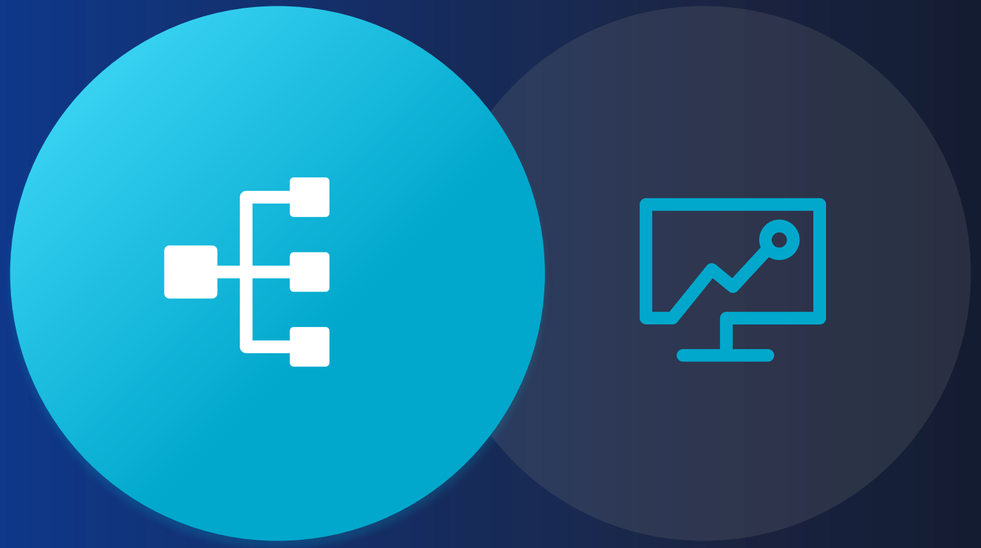
多可用区、备份回档
垃圾桶机制

持续交付

CI/CD
周级交付

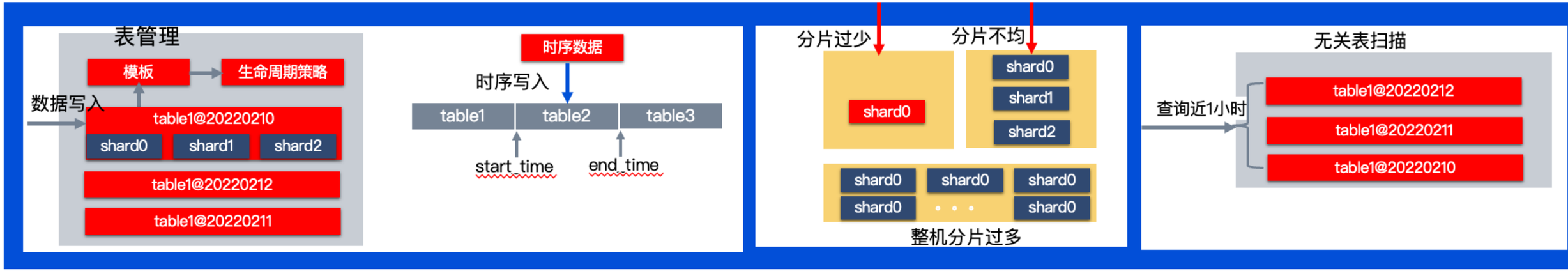
腾讯云 ES 服务自治

自治索引



智能诊断

自治索引　PB级日志数据管理的挑战



数据接入阶段

- **索引创建**
  - 模版、路由管理
- **生命周期策略**
  - 索引滚动
  - 降冷、归档
  - 过期删除

数据维护阶段

- **定期调整分片数**

  分片数过小、分片分布不均：bulk 写入拒绝

  集群分片总数过多：集群响应慢、可用性下降

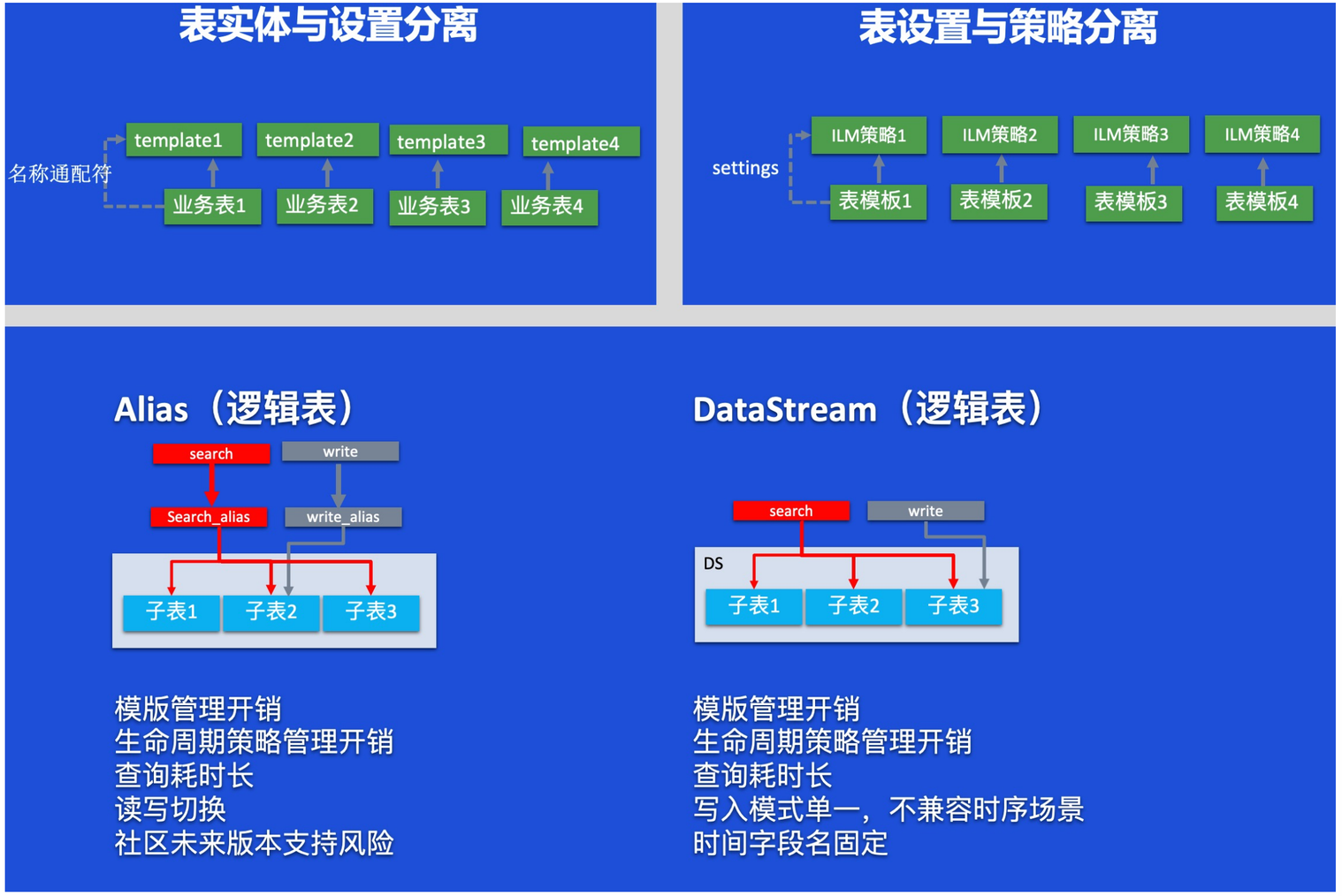
- **读写性能低**

  无关子表扫描，查询耗时明显

  写入性能低：分布式长尾问题

- **异常维护**
  - 机器故障，及时滚动写入索引

自治索引　社区方案



使用门槛

- **创建繁琐**：ILM>>模板>>DataStream
- **时间字段**：限定@timestamp字段
- **使用场景**：只能追加写，数据无法更新
- **无查询裁剪**：查询指定时间段的数据，依然扫描所有index
- **维护开销**：

  分片数调整

  index 预创建：

  写入触发创建阻塞写入

  零点批量 rollover 导致集群压力过大

Set up a data stream

To set up a data stream, follow these steps:

1. Create an index lifecycle policy
2. Create component templates
3. Create an index template
4. Create the data stream
5. Secure the data stream

生命周期管理（热->冷->过期）
模板管理：index mappings、index settings

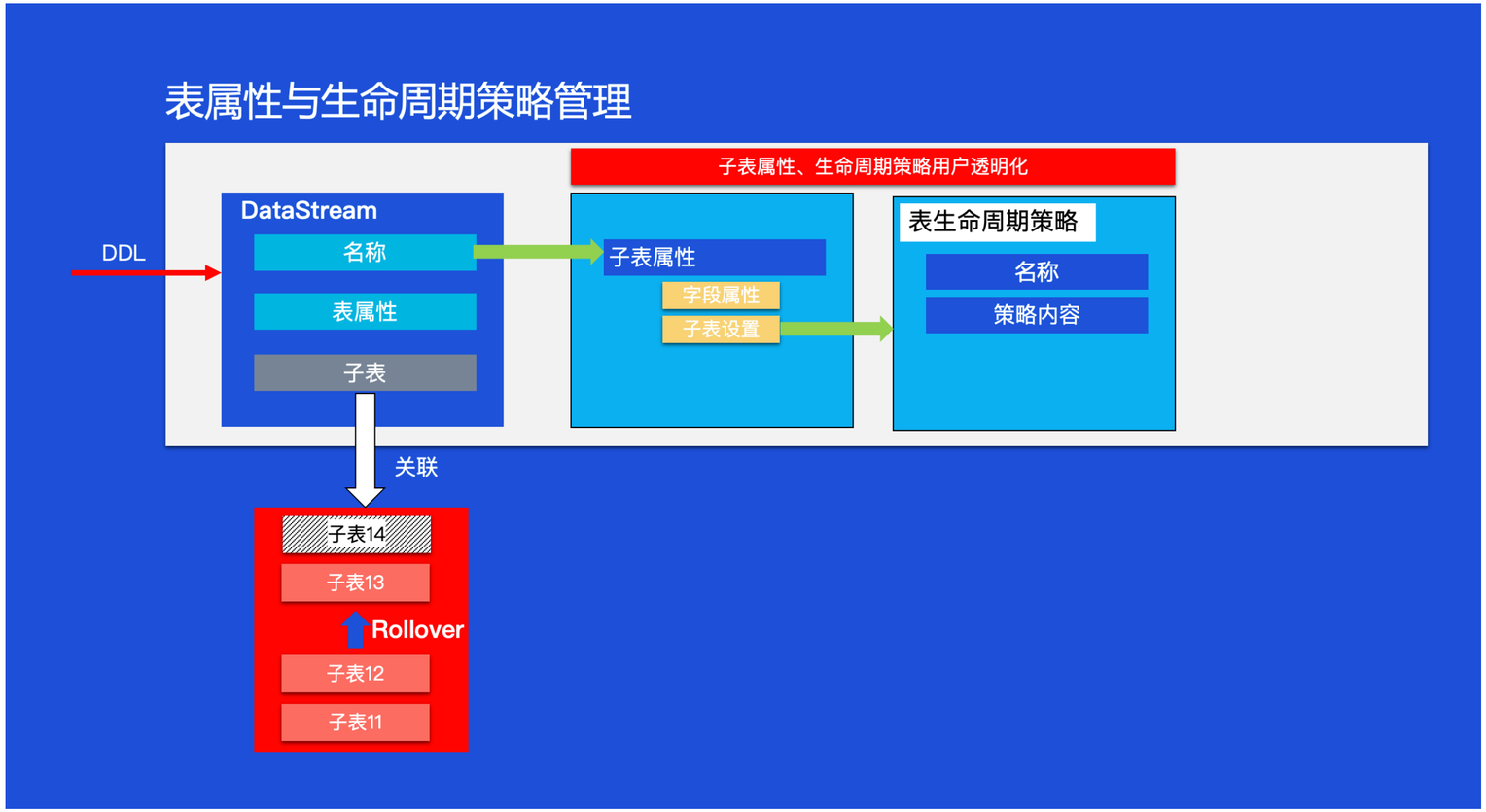
创建 data stream

自治索引　腾讯云 ES 解决方案

DataStream 接口扩展，100%兼容社区开源版本

普通日志用户：建表后直接写入即可

```
PUT _data_stream/mylog
{
  "options": {
    "timestamp_field": "my_time",
    "expire.age": "360d"
  }
}
```



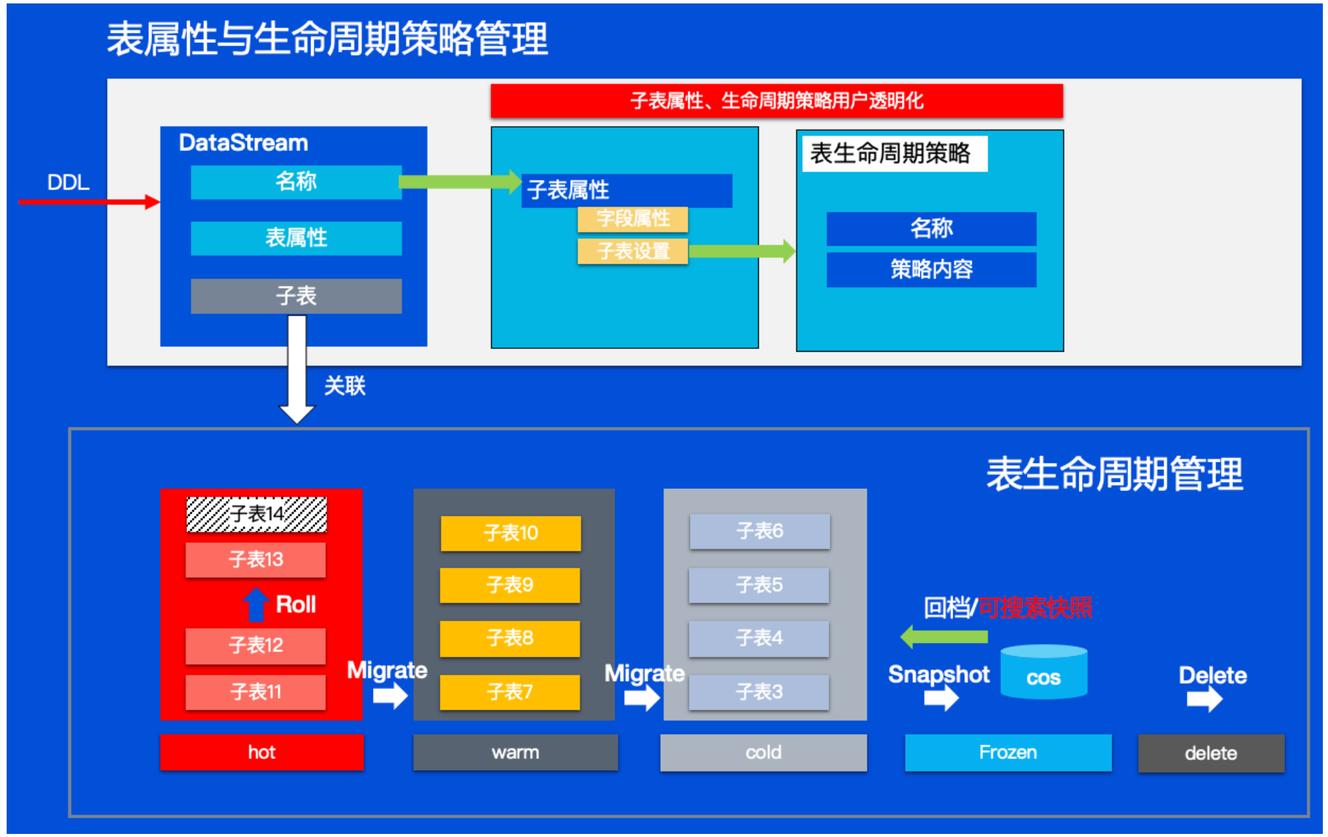
我们帮助用户

- 创建 data stream、关联子表
- 创建 生命周期策略[policy]
- **自动调整 rollover 周期（3h？1d？5d？）**
- **自动预创建 index**
- **机器故障自动滚动索引，防止写入阻塞**
- **自动调整分片数**

自治索引　腾讯云 ES 解决方案

高阶用户：像使用一张表一样简单

```
PUT _data_stream/mylog
{
  "mappings": {
    "properties": {
      "field1": {
        "type": "text"
      }
    }
  },
  "settings": {
    "index.refresh_interval": "10s"
  },
  "policy": {
    "warm.min_age": "7d",
    "warm.actions.shrink.number_of_shards": 1,
    "cold.min_age": "30d"
  },
  "options": {
    "expire.max_size": "1TB",
    "write_mode": "time_partition"
  }
}
```



自定义 mappings、settings、policy

支持按数据量过期数据

支持update写入（time_partition），自动路由至相关 index（子表）

支持更新各项设置

收益：

- **1 条　命令完成创建**
- **100% 兼容开源**
- **托管　索引维护**

自治索引　索引维护托管-分片数调整

分片数评估

- **存量评估**：分片存储量、文档数
- **增量评估**：写入速度(写入毛刺、快速增长、周期性波动)
- **即时反馈**：写入拒绝触发

分片数调整

依据目标跟踪算法和数字信号处理思想，采用时序特征拟合、预测、傅里叶级数等多种时域、频域分析算法，分析用户读写特性，确认最终分片数

- **最大边界**：数据节点倍数(热、温、冷、多盘)
- **容忍比例**：2倍扩容比例，1/4缩容比例
- **确认机制**：快速扩容，谨慎缩容(长时间窗口确认)

分片调整效果

使单个 index 的 shard 数保持在合理范围内，防止分片数数过少导致 bulk 拒绝。

使单个 shard 的容量保持在合理范围内，防止因单个 shard 数据量过大导致读写性能下降。

使整个集群的分片数保持在合理范围内，防止因集群shard总数过多导致集群响应变慢、可用性降低。

- **写入拒绝5分钟内自动恢复**
- **典型场景集群分片数下降60%+**

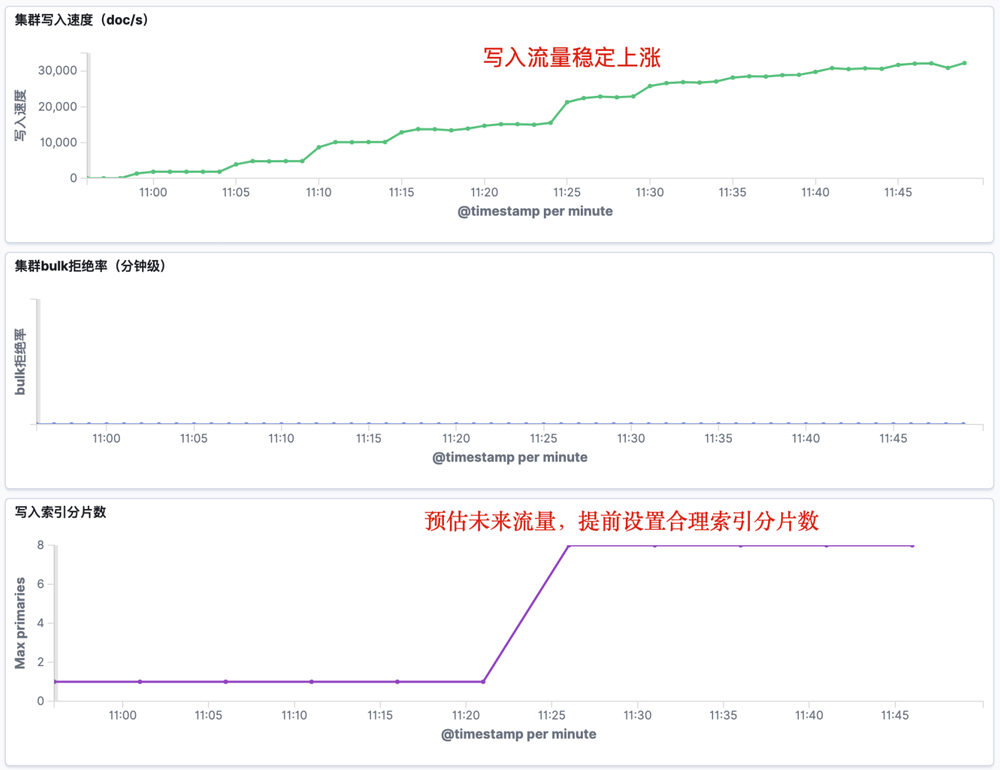
自治索引　索引维护托管-分片数调整

分片调整效果

基于业务负载动态调整分片，兼容写入快速增长、周期性波动、写入毛刺等场景，并且和节点数量关联，使得业务在写入快速放量、扩缩容等场景下，不会出现写入拒绝 或 需要手动调整

1）写入流量突增导致bulk拒绝，立即调整索引分片数

2）写入流量稳定上涨，按预期流量调整合理索引分片数

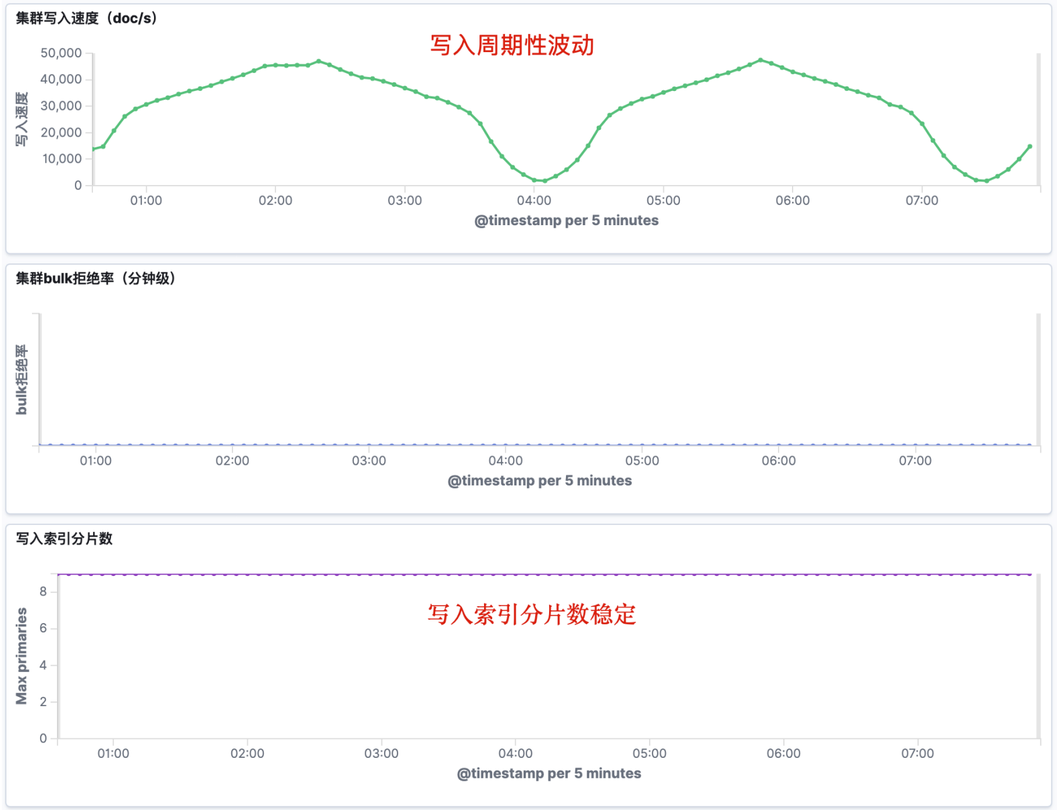


自治索引　索引维护托管-分片数调整

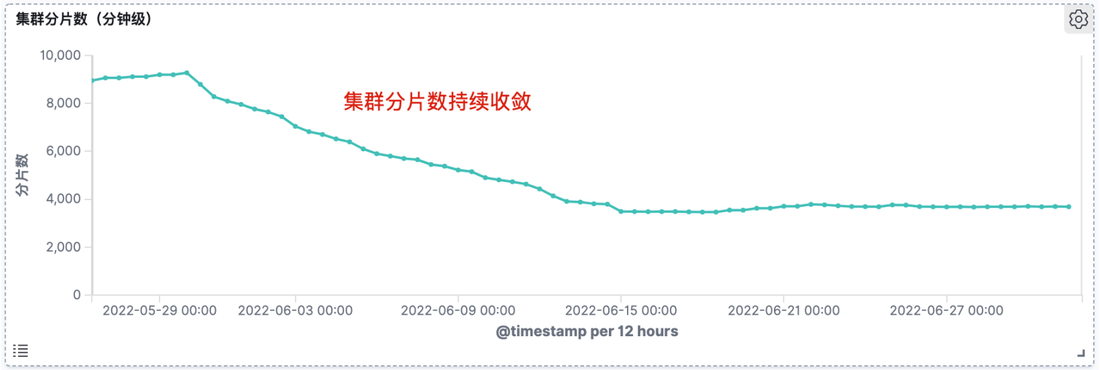
分片调整效果

自治索引会保障分片数合理设置，收敛整个集群的分片数，提高集群稳定性

3）写入周期性波动不频繁调整



4）持续收敛集群分片数：分片总数降低60%~

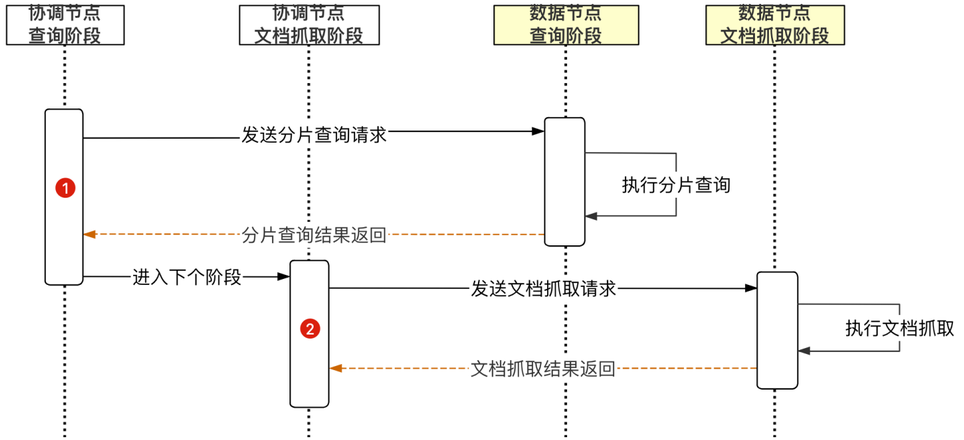


自治索引　查询裁剪

问题背景

PB级日志业务按时间分区

- 查询冷热特点明显：近多远少
- 索引前缀查询比指定索引慢3倍+



原因分析：

- 耗时集中协调节点QueryPhase阶段
- 单次网络请求开销 30+微秒

查询阶段	耗时(ms)	耗时占比
1、prepare target shards	77	8.26%
2、query phase	848	90.99%
3、fetch phase	1	0.11%
4、expand phase	6	0.64%
汇总	932	100.00%

耗时统计	串行发送请求	
总次数	22500	占比
stage1:前置检查(ns)	289	0.77%
stage2:获取连接(ns)	304	0.81%
stage3:构造request对象(ns)	317	0.84%
stage4:发送请求(ns)	36703	97.58%
合计	37613	100.00%
阶段总耗时		846

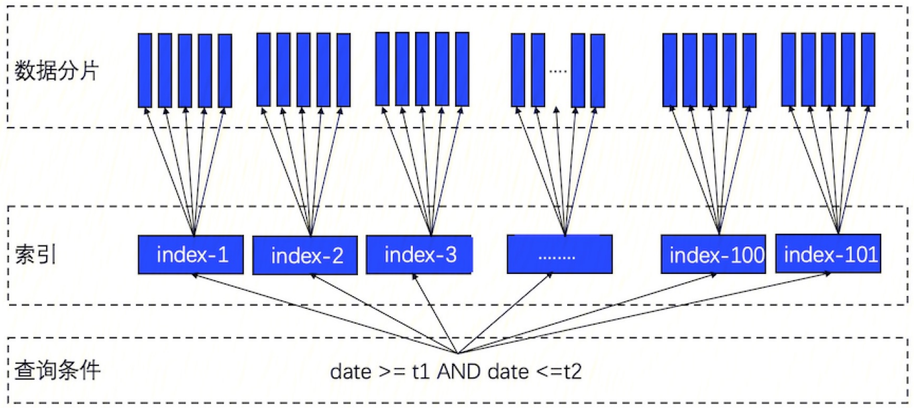
结论：

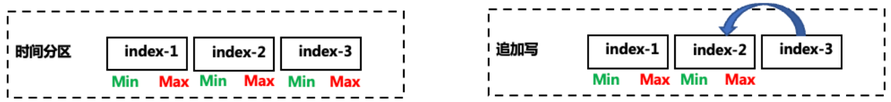
- 查询耗时集中在大量分片遍历操作

自治索引　查询裁剪

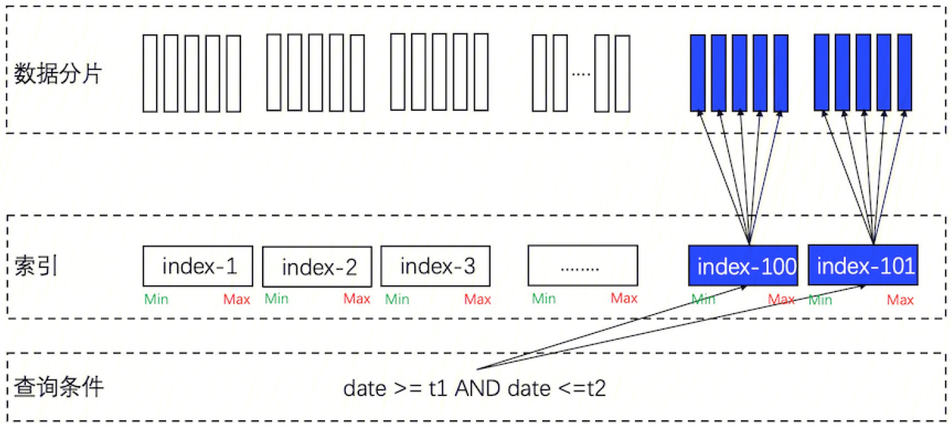
优化方案：协调节点查询裁剪 支持索引级/分片级 Min/Max 索引，实现索引级别的快速裁剪

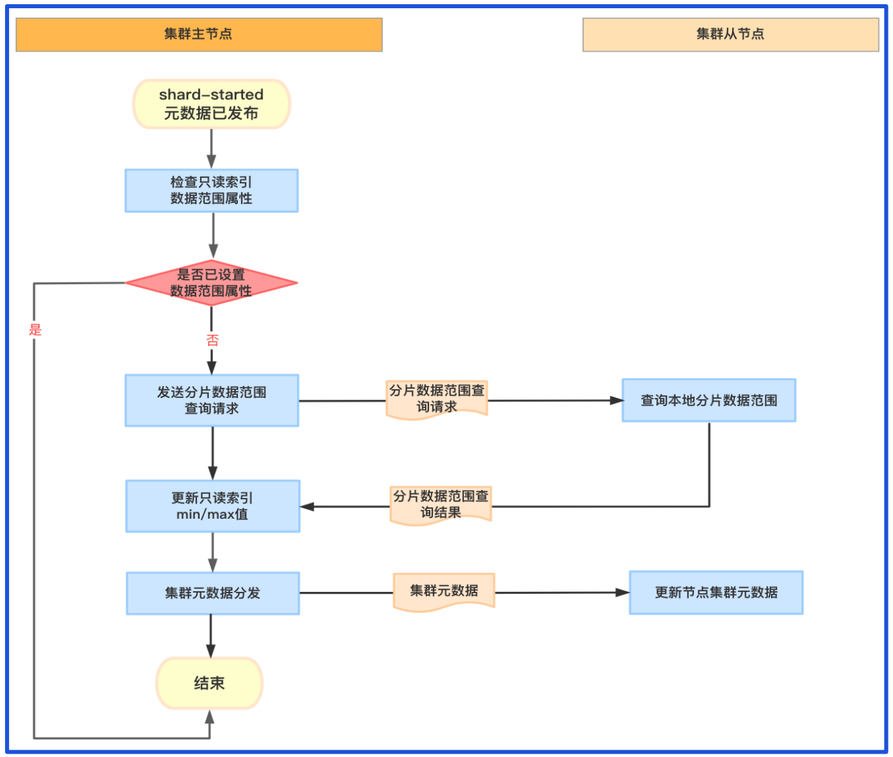
优化前





优化后





优化效果：

- **PB级日志查询性能提升 8倍+**

腾讯云 ES 服务自治

智能诊断

智能诊断　为什么要做诊断系统

值班问题梳理	处理方式
机器故障 硬盘、内存、网卡...	机器替换、数据搬迁
用户使用读写接口不合理 读写不平滑、读写倾斜...	指导用户改造
集群配置不合理 分片数、cache大小...	部分参数可以热更新 其他需要用户配合更新
ES内核缺陷	ES内核代码改动
系统参数未优化 numa、min_free_kbytes...	调整系统参数
其他	...

- **值班问题分析：重复度高，95+%**问题是已知问题

- **为什么还是慢？**
- **排查时间长：**整理了checklist，**但仍需要人工逐一排查**
- **重复定位：**人工排查容易疏漏，**导致问题可能需要多次定位**
- **经验难以共享：**同样问题，没接触过的人会比曾经定位处理过的人会比**花费时间要多得多**

智能诊断　目标

- **我们的目标**
  - 排障：实时触发，精准定位故障的 Root Cause
  - 调优：定期触发，降低用户使用门槛、故障预警

智能诊断

观察

信息采集

- 准确描述故障，尽可能还原现场，指标完备，应采尽采

分析

根因定位

- 量化数据，算法分析，找到准确的故障原因【Root Cause】

决策

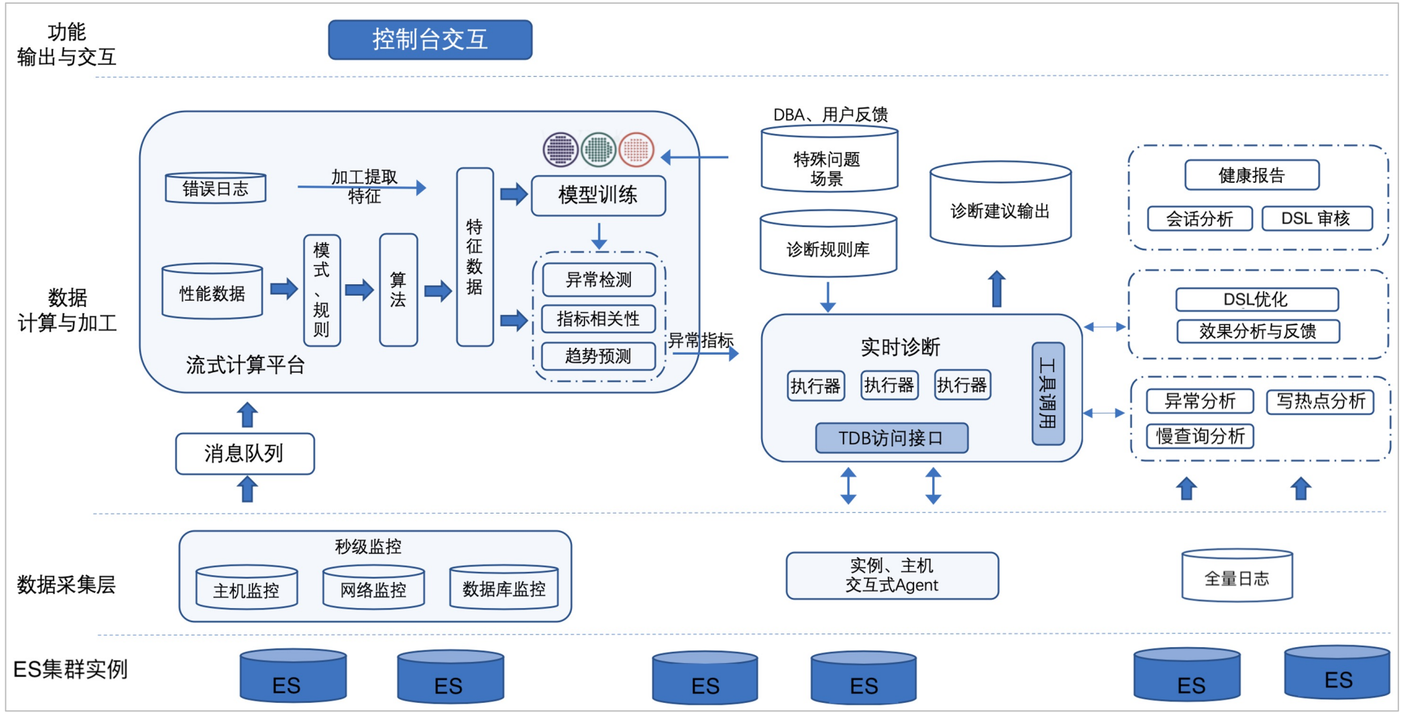
故障处理

- 针对原因给出高效解决方案，尽可能降低运营成本

数据

算法

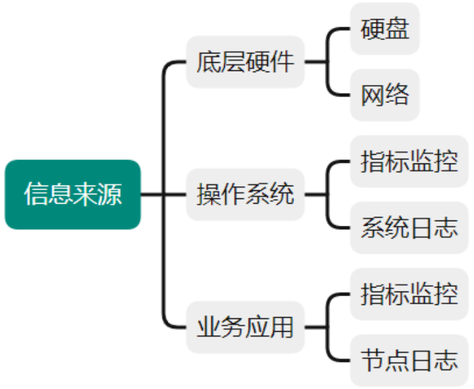
智能DBA



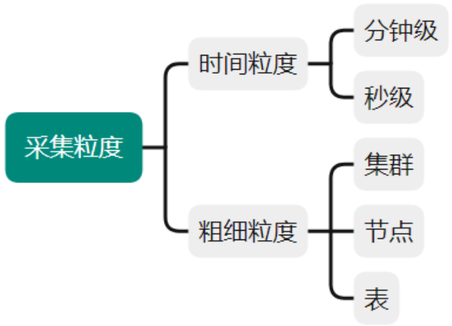
智能诊断 信息采集

多维度、多粒度全方位信息采集

监控覆盖全链路



动态时间粒度、粗细粒度



轻量化采集

采集逻辑优化：

http开启压缩

去除冗余序列化、反序列化步骤

去除冗余拷贝

减少单次上报数据量

成本优化：

差异化保存时长、监控粒度

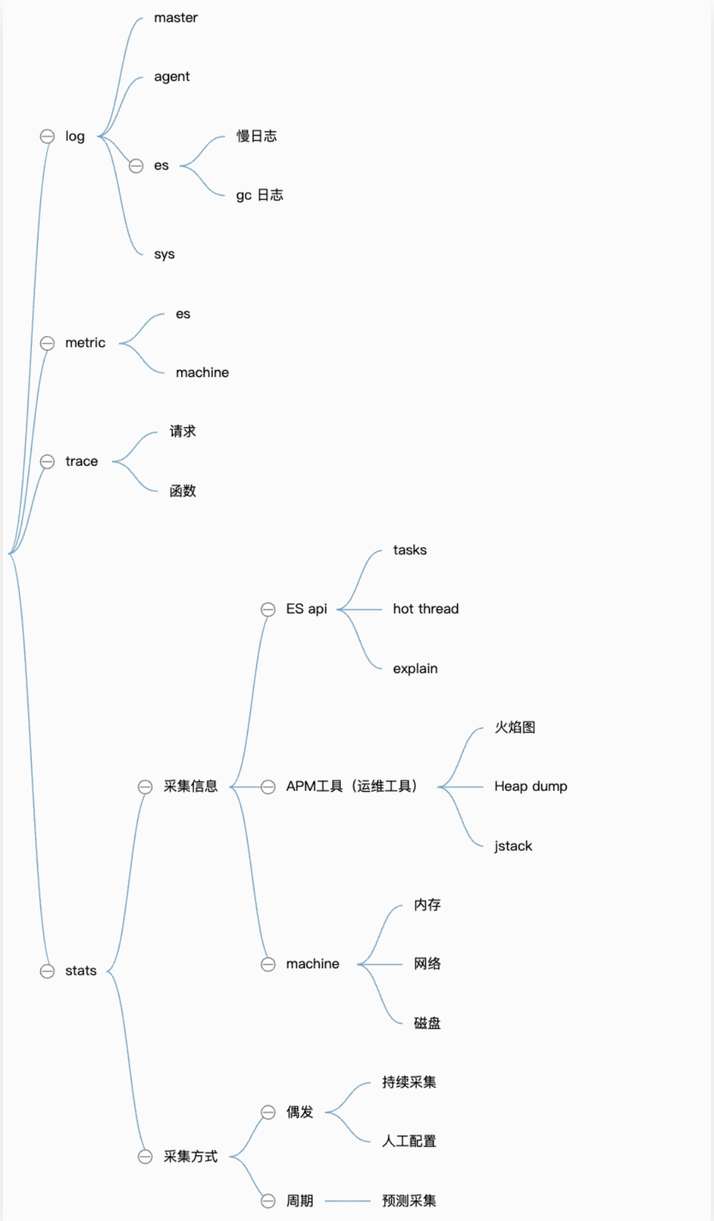
按需开启采集

细粒度采集项（堆栈、秒级表写入速度）

索引、分片指标取TOP N

异常考虑：故障时集群API不可用

基于历史数据和日志补齐部分指标



智能诊断　根因定位

特征提取

- 如何通过采集到的信息确定故障原因？

特征提取 + 异常检测 + 规则匹配

异常检测

- 如何判断监控指标出现异常？

无法统一标准：

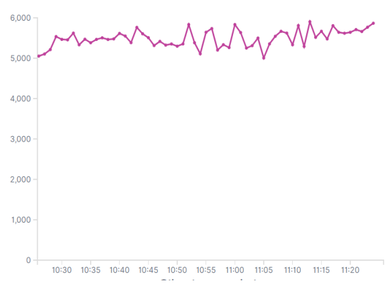
实例规格和使用场景不一，指标表现差异大

维护难度高：业务变化

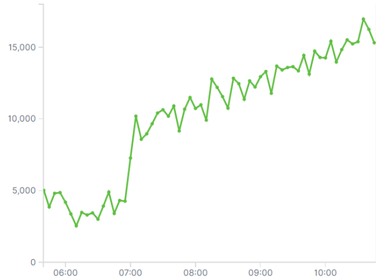
节点规格	单次批请求大小	最大写入速度(doc/s)
2C4G	100	8846
2C4G	10	2477
1C2G	100	4203

影响因素还包括：写入数据大小、是否有路由、刷盘间隔...

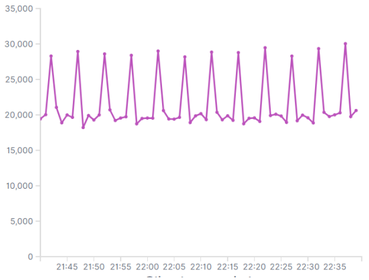
根据时序特征进行分类，选择相应算法进行预测分析



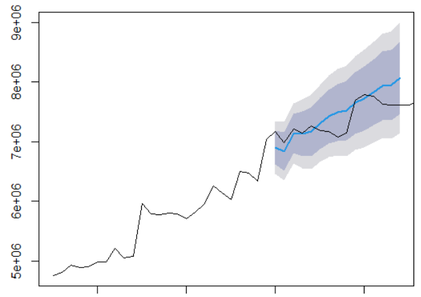
平稳性



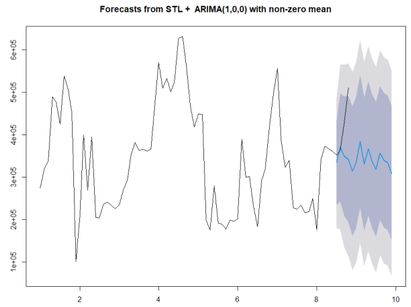
趋势性



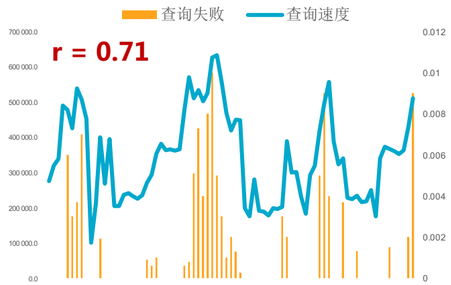
周期性



预测范围内的持续变化被认为是正常的



分析相关性



智能诊断　效果 - 写入拒绝（分片数过少导致）

现场描述

事件详情

诊断项	BulkReject
风险等级	严重
概要	[redacted]
起止时间	2022-03-26 03:34:21 ~ 2022-03-26 03:34:31
持续时长	10 秒

现场描述　智能分析　优化建议

问题描述

写入拒绝: 在往集群写入数据时, 由于节点写入缓冲队列打满而出现请求被拒绝返回错误的情况, 导致数据写入失败。
如果生产端没有失败重试, 会出现数据丢失, 如果有重试机制, 生产端可能出现积压。
拒绝率大于0.1%节点数量 1, 集群节点数量 80
拒绝率较高的节点列表

节点名	节点ip	拒绝率(%)
[redacted]	[redacted]	24.23

智能分析

现场描述　**智能分析**　优化建议

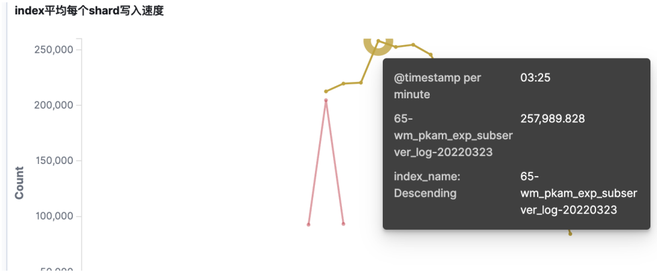
原因描述

索引分片不均

索引的写入压力一般会均匀分配到各个分片上, 如果写入速度较高, 而分片数量不足或者无法均匀分配给各个节点, 就会导致分片数量较多的节点有更高的写入压力, 容易出现打满写入队列, 写入请求拒绝甚至打垮节点的情况

以下索引有分片数过少或不均的情况

索引名	分片数/总节点数	单个节点最大分片数	平均每分片写入速度	是否分片数过少
65-wm_pkam_exp_subserver_log-20220323	21 / 80	1	252538.03	是



优化建议

现场描述　智能分析　**优化建议**

处理方法和建议

长期方案

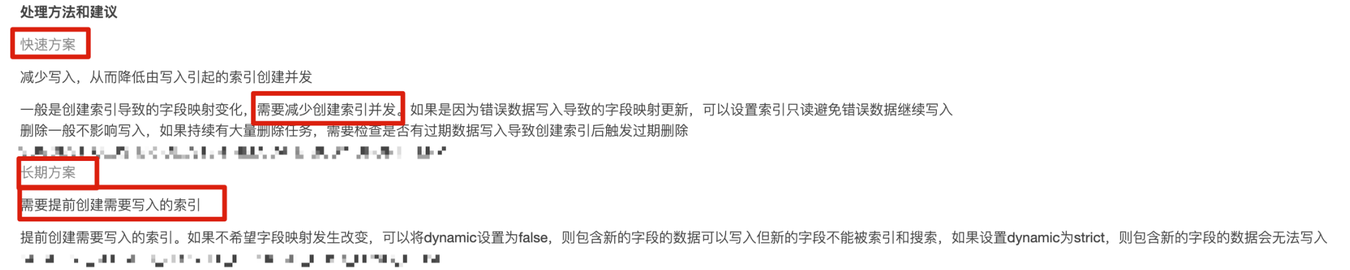
调整索引分片数，写入速度大的索引[65-wm_pkam_exp_subserver_log-20220323]分片数建议为设置为节点数量的倍数(80/160)

智能诊断　效果 – 写入拒绝（写入时触发 Index 创建导致）

现场描述

智能分析

优化建议



智能诊断　效果 - 集群 Red（低版本Bug导致）

现场描述

智能分析

优化建议

问题定位流程简化

常见问题MTTR（Mean Time To Recovery）

从30分钟级减少至3分钟

NEXT 服务自治

Andy Pavlo

What is a Self-Driving Database Management System?

Posted on Monday April 09, 2018 at 12:46 PM

TL;DR

Some organizations and people are inaccurately labeling their systems as "self-driving". A true self-driving DBMS automatically (1) decides what actions to use to optimize itself, (2) decides when to deploy those actions, and (3) learns from those actions — all without any human intervention.

1. 可以决定该如何优化自身。
2.可以决定何时执行优化措施。
3.可以从行为中学习而无需人工干预。

自治工具（外围建设）

负载分析、索引推荐、自动伸缩、参数调优、异常检测
智能诊断、智能告警

自治组件（内核增强）

学习索引、代价估计、执行计划、缓存策略

https://www.cs.cmu.edu/~pavlo/blog/2018/04/what-is-a-self-driving-database-management-system.html

elastic 中国开发者大会 2023

感谢观看

elastic
中文社区

专业、垂直、纯粹的 Elastic 开源技术交流社区

https://elasticsearch.cn/