



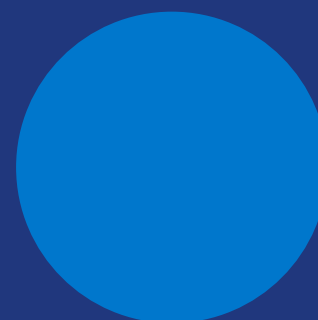
基于 Elasticsearch 打造时序引擎

- 阿里云与 Elastic 社区共建实践分享

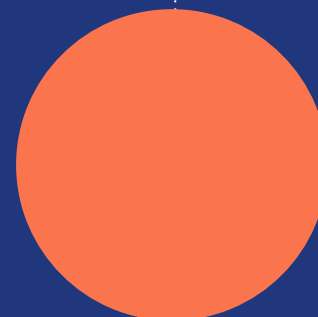
- 魏子珺，内核专家
- 阿里巴巴，2023/04/08



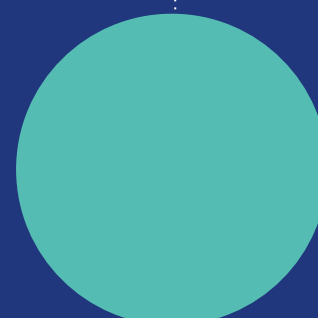
Elasticsearch社区国内Top 1 , 全球 Top 100 Contributor , Elastic Gold Contributor



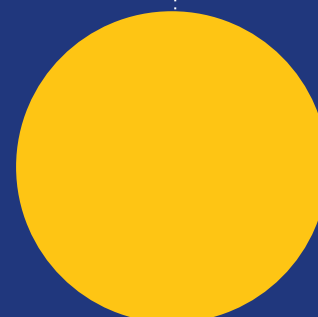
Elasticsearch为什么做TSDB



Elasticsearch做TSDB的挑战



Elasticsearch TSDB特性介绍



阿里云Elasticsearch TimeStream介绍



Elasticsearch为什么做TSDB

阿里与Elastic社区时序引擎共建 issues #74660



Add better support for metric data types (TSDB) #74660

Open

37 of 70 tasks

imotov opened this issue on 29 Jun · 1 comment



imotov commented on 29 Jun · edited by nik9000

Member



Phase 0 - Inception

- ✓ Obtain schemas annotated with dimensions and metrics from the Metrics team (small) @nik9000
- ✓ Prototyping Lucene Data Pull Mechanism(medium) @imotov
- ✓ Prototyping Data Pull Mechanism in elasticsearch @imotov

Phase 1 - Mappings

- ✓ Add `time_series_dimension` mapping parameter to fields
 - ✎ Add dimension mapping parameter #74450 @csoulios
 - ✎ Add constraints to dimension fields #74939 @csoulios
 - ✎ Rename `dimension` mapping parameter to `time_series_dimension` #78012 @csoulios
- ✎ Add time_series_metric parameter #76766 @csoulios
- ✎ TSDB: Add time series information to field caps #78790 @imotov
- ✎ TSDB: Automatically add timestamp mapper #79136 @weizijun

Phase 2 - Ingest

- ☐ Dimension-based tsid generator
 - ✎ Memory efficient xcontent filtering #77154 @nik9000
 - ✓ Add TSDB-specific tests
 - ✎ TSDB: More tests #78208 @nik9000
 - ✎ TSDB: CCS Tests #78042 @nik9000
 - ✎ TSDB: Tests for data_stream #78038 @nik9000
 - ✎ TSDB: CCR tests #78034 @nik9000
 - ✎ TSDB: Basic rolling upgrade test #78028 @nik9000
 - ✎ TSDB: snapshot/restore test #78022 @nik9000
 - ✎ TSDB: Add `_tsid` field to time series indices #80276 @csoulios
 - ✎ TSDB: Add security tests for `_tsid` #81382 @csoulios (prototype)
- ☐ Document best practices for using dimensions-based ID generator
- ☐ Routing
 - ✎ Pull index routing into strategy object #77211 @nik9000
 - ✎ Route documents to the correct shards in tsdb #77731 @nik9000
 - ✎ Validate tsdb's routing_path #79384 @nik9000
 - ✎ More validation for routing_path #79520 @nik9000
 - ✎ TSDB: Do not allow index splits for time series indices #81125 @csoulios
 - ✎ Improve FilterPathBasedFilter performance #79826 (Speed up xcontent filtering) @weizijun
 - ✓ Have a good hard look at the switch statement in `BulkOperation`. Maybe we can make this simpler.
 - ✎ Move routing calculation #79394 @nik9000
 - ✎ Move required routing validation to IndexRouting #79472 @nik9000
 - ✎ Remove tricky switch in bulk #80624
 - ✎ TSDB: Test `ids` query on time series index #81436 @csoulios
- ☐ Test and fix get-by-id
- ☐ Handling Time Boundaries
 - ✎ TSDB: add index timestamp range check #78291 (Added `start_time`, `end_time` index settings) @weizijun
 - ☐ Make time boundaries required in tsdb indices @weizijun ✎ TSDB: Make time boundaries settings required in tsdb indices #81146
 - ✓ Replace hard check for `index_mode=TIME_SERIES` with bounds checking on start and end time @nik9000 ✎ TSDB: Cleaner trigger for tsdb boundary check #81263
 - ✓ Tests for nanosecond timeprecision timestamp just beyond the limit
 - ☐ Automated update of index time boundaries on index rollover @martijnvg



weizijun



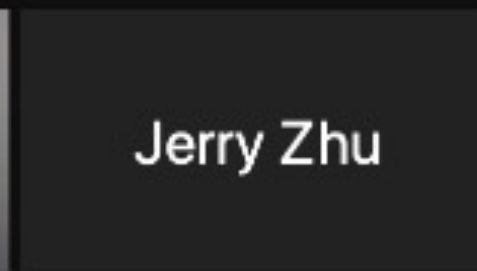
zhichen



William Chaparro



Igor Motov



Jerry Zhu



Jerry Zhu

视图



移除固定视频



think we can start that we waiting for anyone else will occur business. Right.

Gilad Gal



静音



停止视频

12

参会者



聊天



共享屏幕



录制



实时转录文字



回应

离开

用户们的声音

怎么能用ES做TSDB呢？肯定要用专业的TSDB？



ES本来就可以做TSDB，为什么还要开发？



TSDB介绍（特点鲜明）：

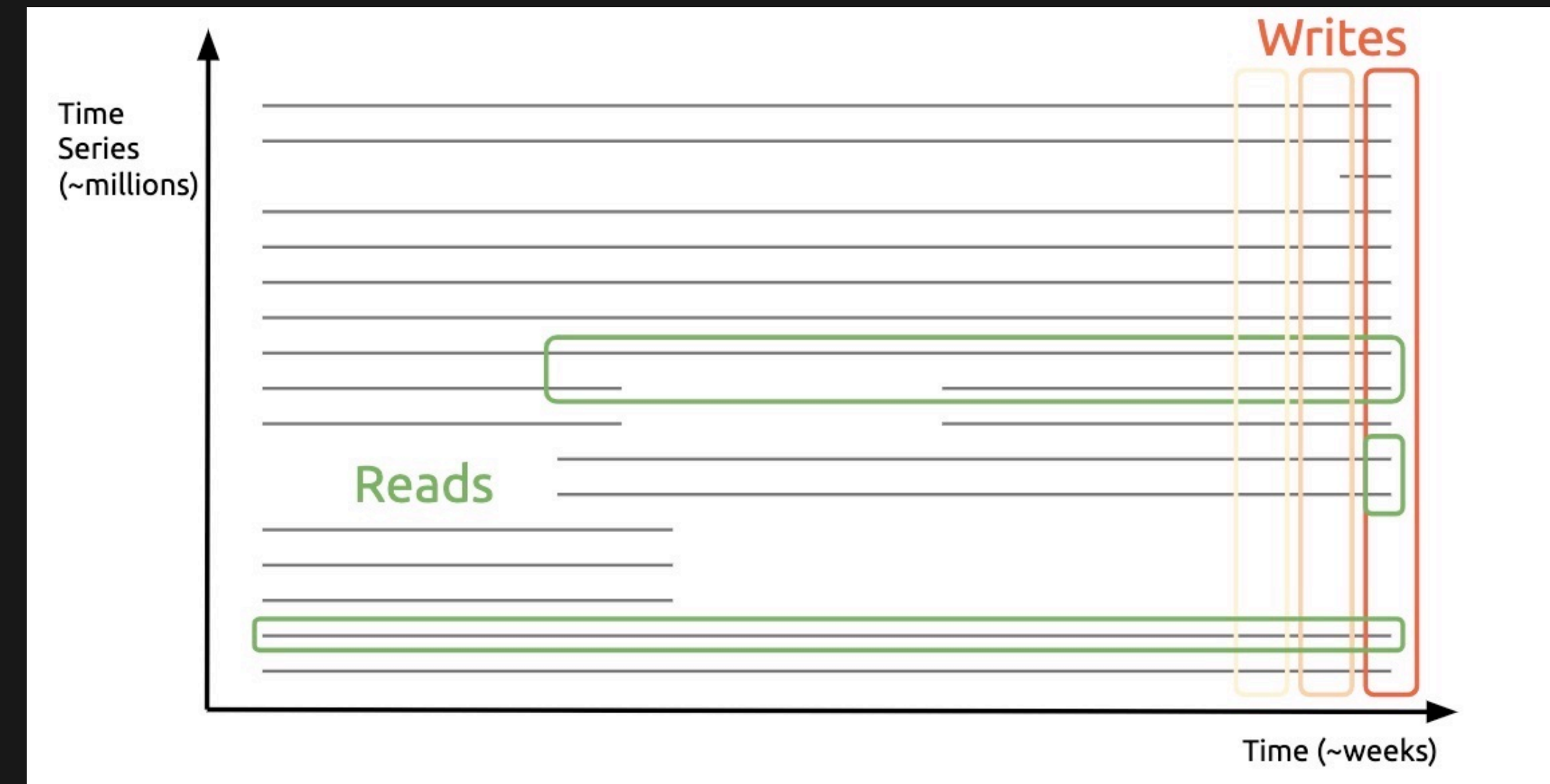
采样数据：基于稳定频率持续产生的一系列数据。

数据模型固定：timestamp（时间）、dimension（维度）、metric（指标）。

时间线：由维度组成，保持不变，时间和指标在不断变化。

持续产生海量数据，没有波峰波谷。

垂直写（最新数据），水平查。



TSDDB需求

01

写入

支持高TPS写入

02

查询

支持多维度查询、聚合

03

成本

支持高压缩率



TSDB关键能力 (ES都具备)



Elasticsearch做TSDB的挑战



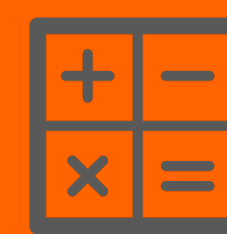
使用门槛过高

需要对ES有深入掌握，才能在时序场景最佳实践



查询慢&复杂

需使用复杂的query DSL，而且性能不佳



成本过大

存储容量是专业TSDB容量的几十倍

ES做TSDB痛点



使用门槛过高

ES做TSDB最佳实践方式：

- metrics字段，只存储doc values
- 根据dimension字段生成时间线id
- 使用时间线id做routing
- 使用时间线id和timestamp做index sorting
- 不存储_source
-



/api/v1/query_range?query=up&start=2015-07-01T20:00:00.000Z&end=2015-07-01T21:00:00.000Z&step=15s'

up ← PromQL

ES DSL →

查看指标up监控曲线

```
POST test/_search
{
  "size": 0,
  "query": {
    "range": {
      "@timestamp": {
        "gte": "2015-07-01T20:00:00.000Z",
        "lt": "2015-07-01T21:00:00.000Z"
      }
    }
  },
  "aggs": {
    "by_time_series": {
      "terms": {
        "field": "_tsid",
        "size": 10
      },
      "aggs": {
        "by_date_histogram": {
          "date_histogram": {
            "field": "@timestamp",
            "fixed_interval": "15s"
          },
          "aggs": {
            "last_value": {
              "top_metrics": {
                "metrics": {
                  "field": "up"
                },
                "sort": {
                  "@timestamp": "desc"
                }
              }
            }
          }
        }
      }
    }
  }
}
```


按集群维度查看 index_qps监控曲线



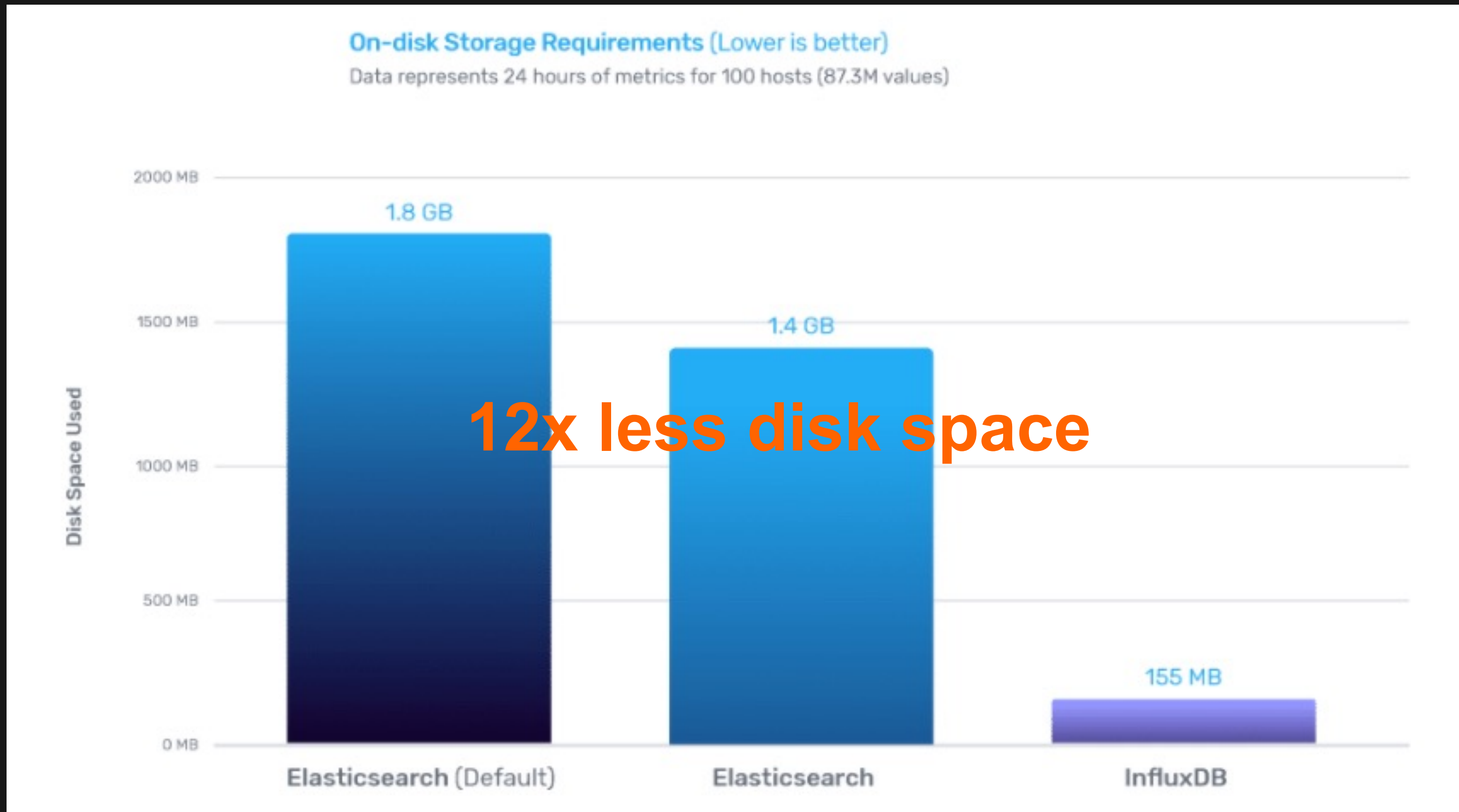
sum(rate(index_qps[1m])) by (cluster) ← PromQL

/api/v1/query_range?query=rate(index_qps[1m])
&start=2015-07-01T20:00:00.000Z&end=2015-07-01T21:00:00.000Z&step=1m'

ES DSL →

```
POST test/_search
{
  "size": 0,
  "query": {
    "range": {
      "@timestamp": {
        "gte": "2015-07-01T20:00:00.000Z",
        "lt": "2015-07-01T21:00:00.000Z"
      }
    }
  },
  "aggs": {
    "group": {
      "multi_terms": {
        "size": 50,
        "order": [{"order_standard": "desc"}, {"_key": "asc"}],
        "terms": [{"field": "cluster"}]
      },
      "agg": {
        "order_standard": {
          "sum": {
            "field": "index_qps"
          }
        }
      },
      "by_date_histogram": {
        "date_histogram": {
          "field": "@timestamp",
          "fixed_interval": "1m"
        }
      },
      "aggs": {
        "tsid": {
          "terms": {
            "field": "tsid",
            "size": 5000
          },
          "aggregations": {
            "downsample": {
              "rate": {
                "field": "index_qps",
                "unix": "1m",
                "detect_resets": true
              }
            }
          }
        }
      }
    },
    "order_standard": {
      "sum_bucket": {
        "buckets_path": ["tsid>downsample"],
        "gap_policy": "skip"
      }
    }
  }
}
```

《InfluxDB vs. Elasticsearch for Time Series Data & Metrics Benchmark》

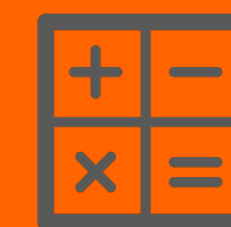
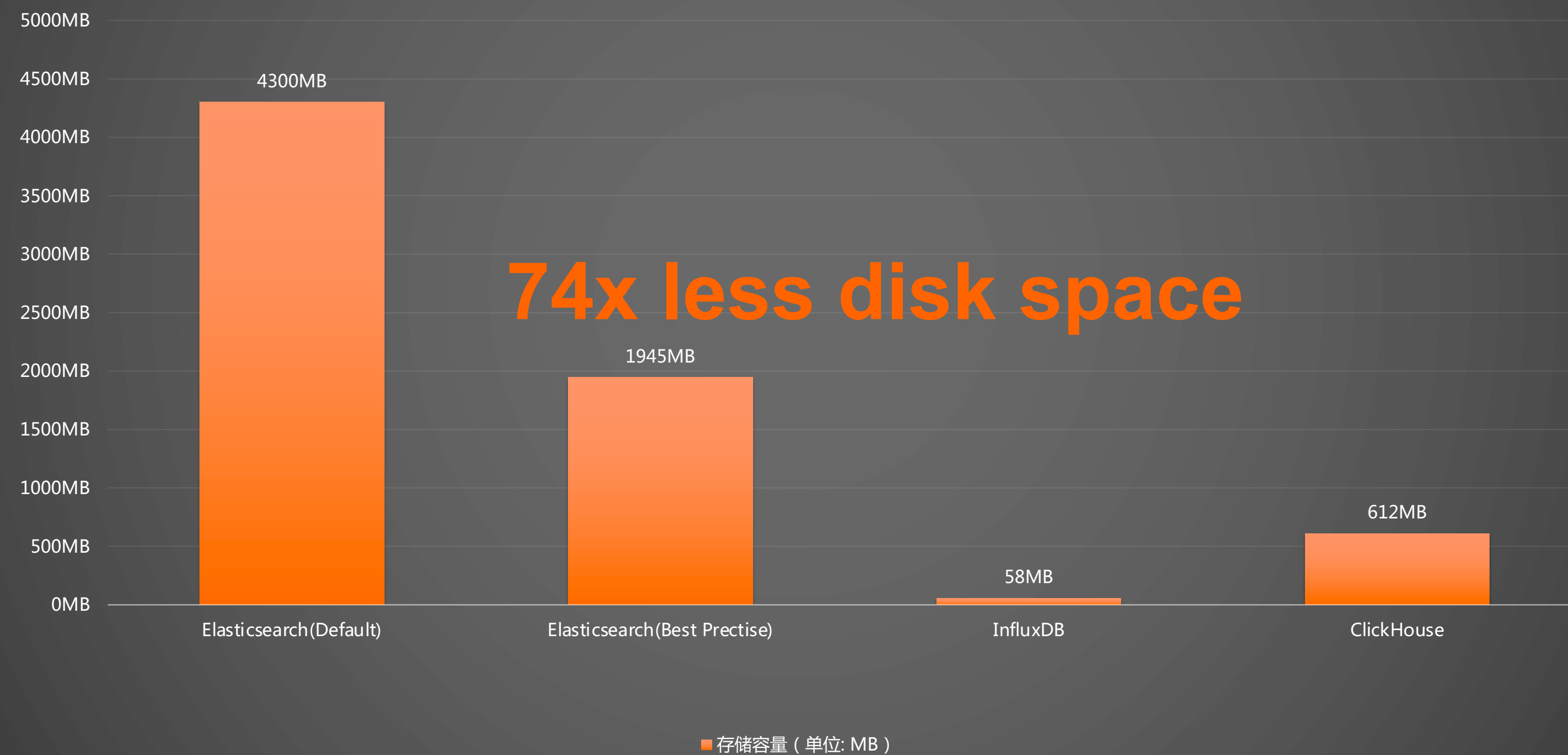


来源: <https://www.influxdata.com/blog/influxdb-markedly-elasticsearch-in-time-series-data-metrics-benchmark/>

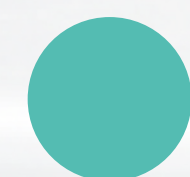


阿里云ES监控数据

TSDb存储容量



成本过大



Elasticsearch TSDB特性介绍



Index Level

PUT _index_template/tsds

```
{
  "index_patterns": [ "tsds" ],
  "data_stream": {},
  "template": {
    "settings": {
      "index.mode": "time_series",
      "index.routing_path": [ "dimentions.*" ]
    },
    "mappings": {
      "_source": {
        "mode": "synthetic"
      },
      "properties": {
        "hostname": {
          "type": "keyword",
          "time_series_dimension": true
        },
        "cpu.load": {
          "type": "double",
          "time_series_metric": "counter"
        },
        "@timestamp": {
          "type": "date"
        }
      }
    }
  }
}
```

定义索引类型为time_series，
引擎内部完成时序场景最佳使用

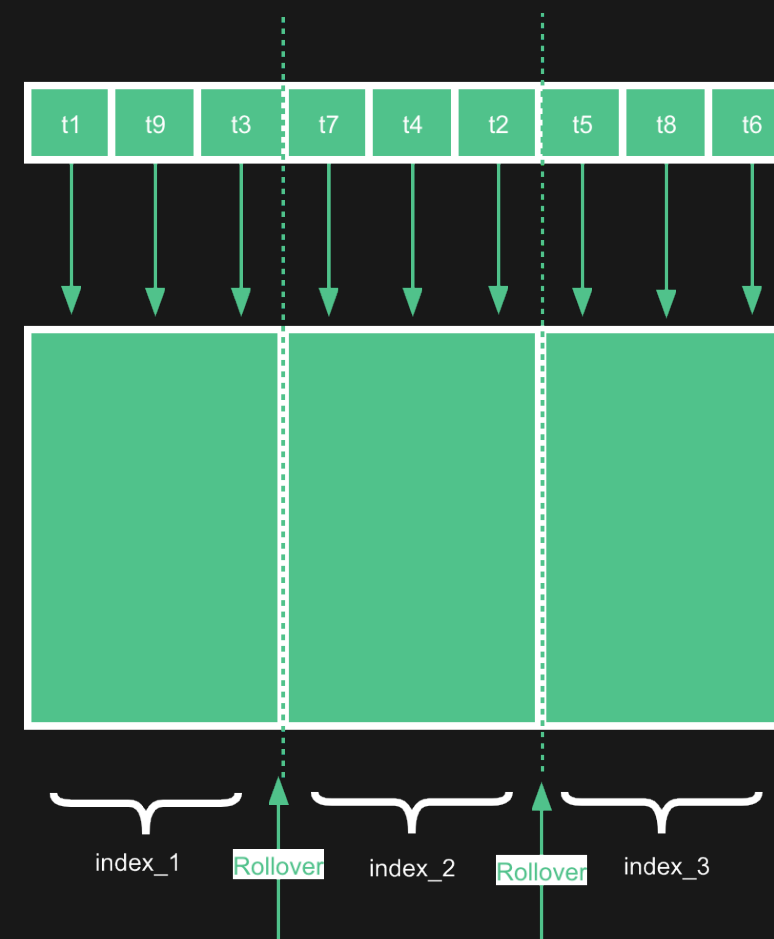
不存储source，使用
docvalues生成

设置维度和指标字段

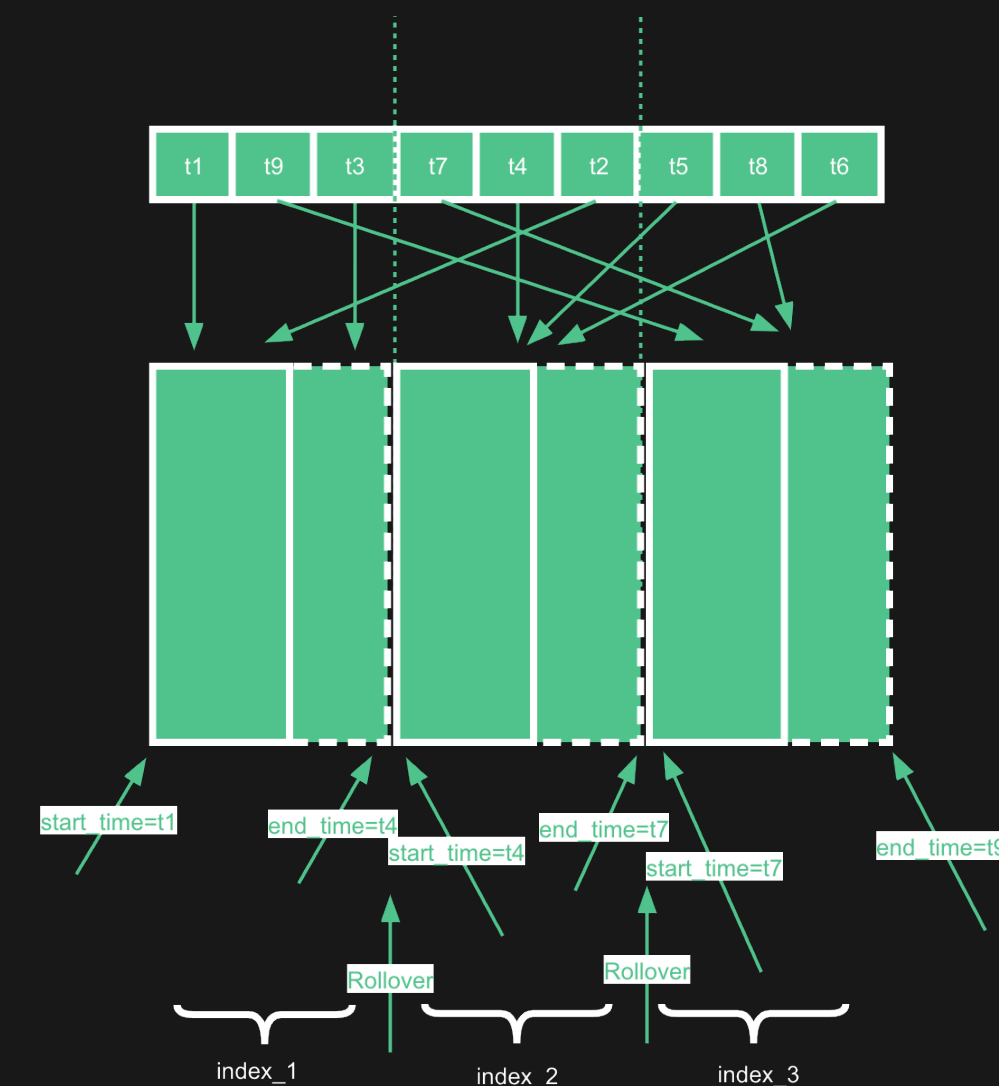
无需设置，time_series
索引会自动创建



TSDS Time Series DataStream



时间分区优化





阿里云Elasticsearch TimeStream介绍

核心目标

阿里云Elasticsearch TimeStream

阿里云基于Elasticsearch TSDB的服务化产品

1 进一步降低用ES做TSDB**使用门槛**

2 进一步**降本增效**

3 与更多系统**直接集成**

使用
介绍

```
创建TimeStream索引
PUT _time_stream/{name}

PUT _time_stream/test_stream
{
  "template": {
    "settings": {
      "index.number_of_shards": "10"
    }
  }
}
```

```
读写数据（ES原生API）
POST test_stream/_doc
{ ... }

GET test_stream/_search
```

```
InfluxDB write API

POST /_time_stream/influx/write?db=test_stream&precision=ms
measurement,namespace=cn-hanzhou cpu.idle=10.0,mem.free=10.1 1624873606000
```

数据
模型

类型	描述
labels	指标相关的属性，Time series ID可由labels生成
metrics	指标数据集合
@timestamp	指标记录对应的时间

```
POST test_stream/_doc
{
  "labels": {
    "namespace": "cn-hanzhou",
    "clusterId": "cn-xxx-xx",
    "nodeId": "node-xxx",
    "label": "test-cluster",
    "disk_type": "cloud_ssd",
    "cluster_type": "normal"
  },
  "metrics": {
    "cpu.idle": 10.0,
    "mem.free": 10.1,
    "disk_ioutil": 5.2
  },
  "@timestamp": 1624873606000
}
```


降本增效:节约存储空间

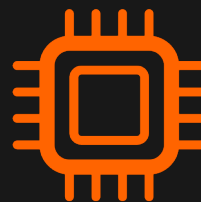
Elasticsearch数据存储容量分析---阿里云ES监控数据:

索引类型	索引大小	元数据大小	元数据占比	时序数据大小	时序数据占比
Default	4300MB	3013MB	70.07%	1287MB	29.93%
Best Practice	1945MB	1155MB	59.36%	790MB	40.64%

索引类型	索引大小	_id和_source占比	_seq_no占比	labels占比	metrics占比	timestamp占比	_tsid占比
Default	4300MB	61.95%	3.90%	16.49%	11.67%	1.77%	0%
Best Practice	1945MB	48.32%	11.04%	8.09%	27.18%	4.62%	0.75%



数据压缩



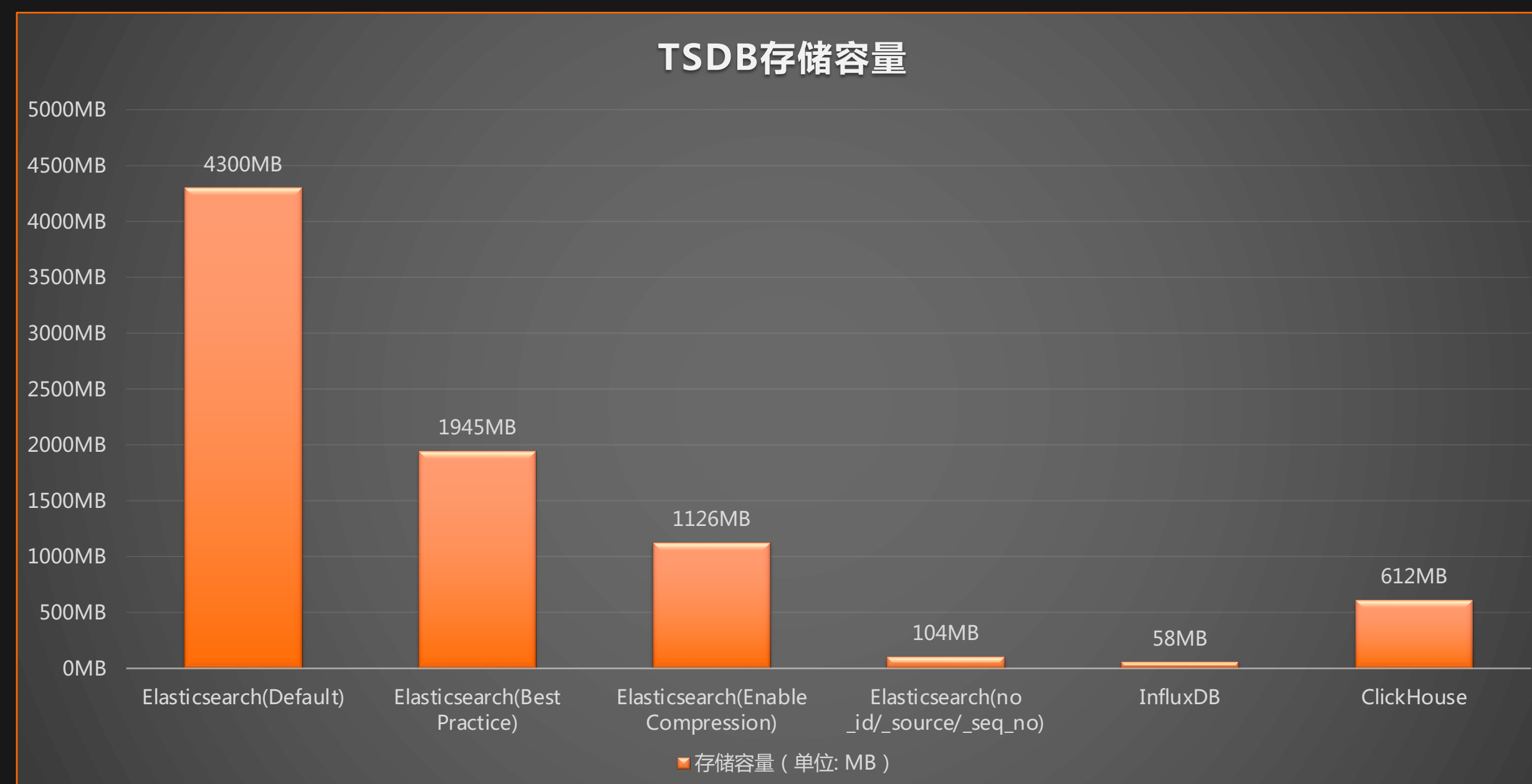
不存_id_source_seq_no

降本增效:节约存储空间

降本增效： 节约储存空间

TimeStream存储优化结果

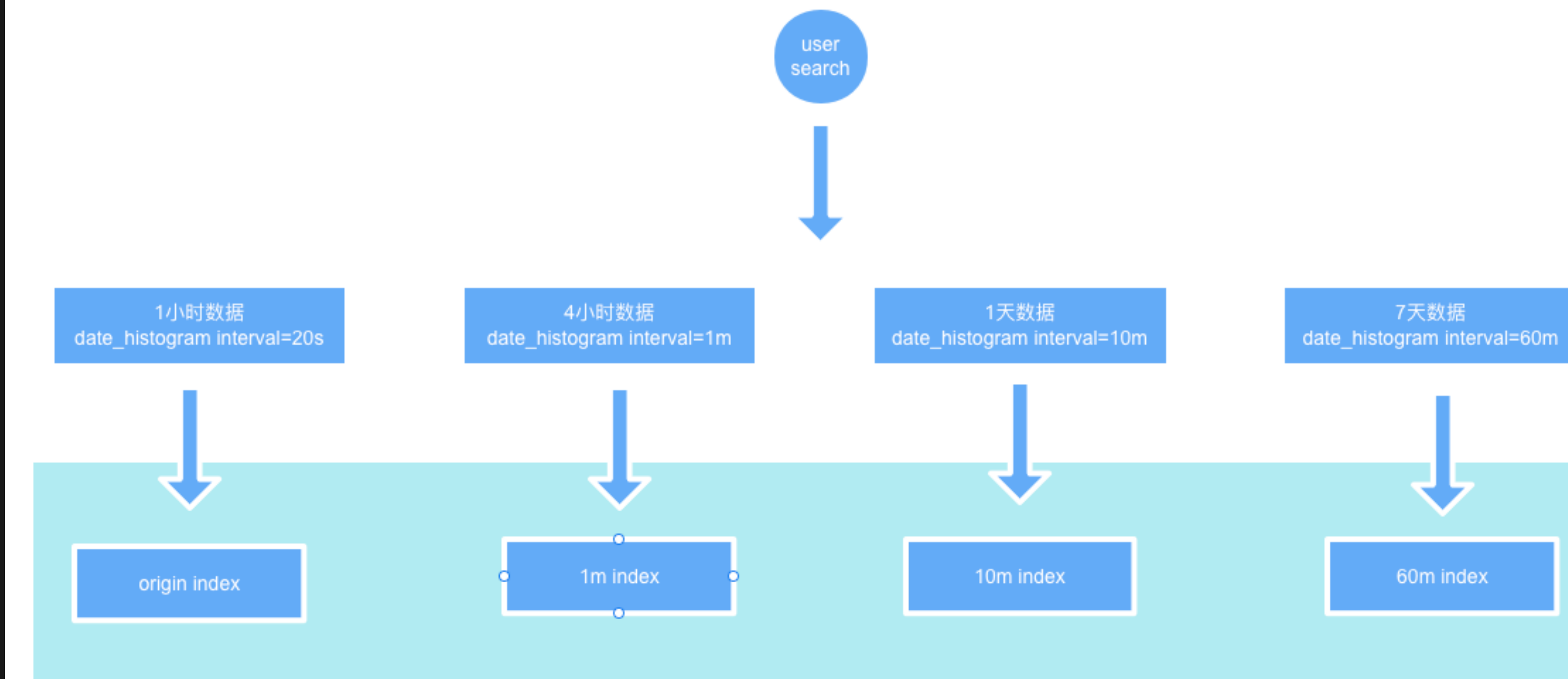
阿里云ES监控数据:



降本增效: 性能提升

支持DownSample

```
PUT _time_stream/test_time_stream
{
  "time_stream": {
    "downsample": [
      {
        "interval": "1m"
      },
      {
        "interval": "10m",
        "ilm_policy": "long_expire_policy"
      },
      {
        "interval": "60m",
        "ilm_policy": "long_expire_policy"
      }
    ]
  }
}
```

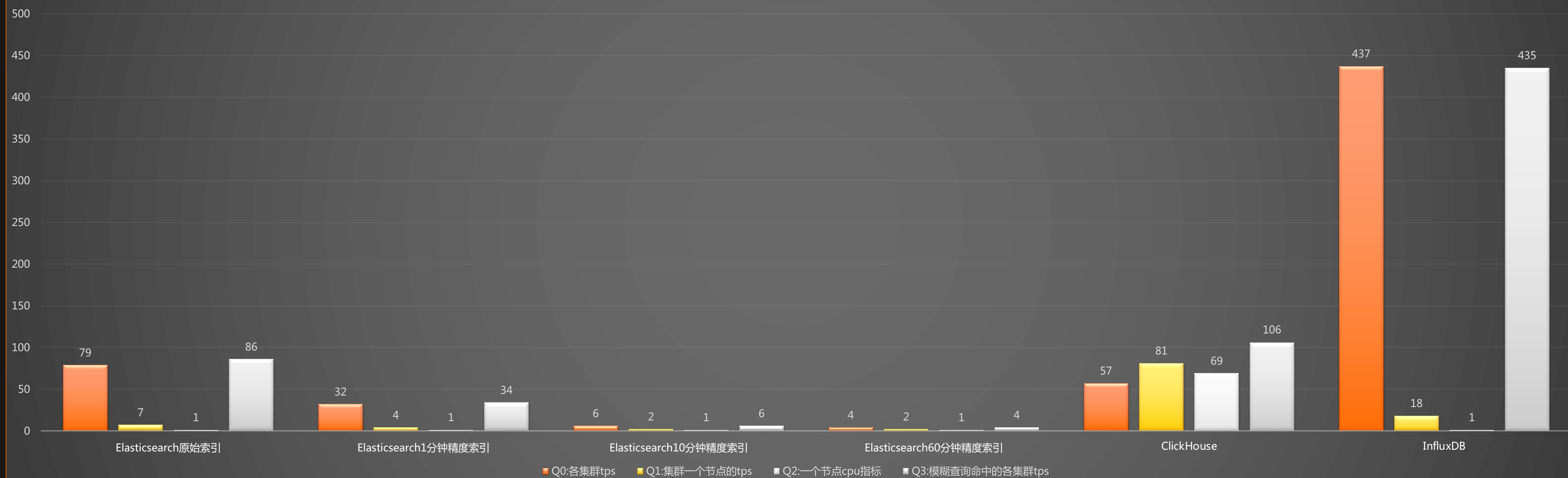


降本增效:

性能提升

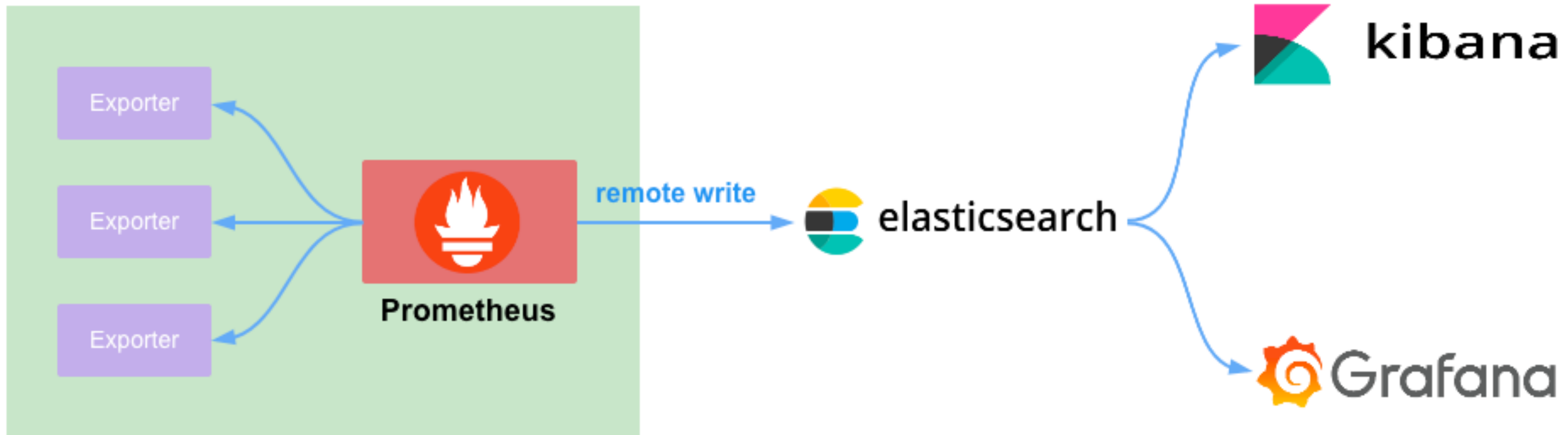
支持DownSample

阿里云ES监控数据查询性能对比



系统集成

支持同步prometheus数据，并使用PromQL查询ES数据



系统集成

与Prometheus + Grafana整合



Data Sources / Prometheus-TimeStream

Type: Prometheus

Settings

Dashboards

Name

Prometheus-TimeStream

Default

☒

HTTP

URL

xxxx:9200/_time_stream/prom_read/prom_index

Access

Server (default)

Help >

Allowed cookies

New tag (enter key to a

Timeout

Timeout in seconds

Auth

Basic auth

☒

With Credentials

☐

TLS Client Auth

☐

With CA Cert

☐

Skip TLS Verify

☐

Forward OAuth Identity

☐

Basic Auth Details

User

elastic

Password

General / Node Exporter

Quick CPU / Mem / Disk

CPU Busy

1.67%

Sys Load (5m avg)

0.667%

Sys Load (15m avg)

1%

RAM Used

61%

SWAP Used

N/A

Root FS Used

68.8%

CPU Cores

12

Uptime

7.1 week

RootFS Total

492 GiB

RAM Total

46 GiB

SWAP Total

0 B

Basic CPU / Mem / Net / Disk

CPU Basic

Memory Basic

Network Traffic Basic

Disk Space Used Basic

CPU / Memory / Net / Disk

CPU

Memory Stack

Network Traffic

Disk Space Used

Query 1 Transform 0

Data source

Prometheus-TimeStream

Query options

MD = auto = 2128 Interval = 15s

Query inspector

Metrics browser >

(((count(count(node_cpu_seconds_total{instance="\$node",job="\$job"}) by (cpu))) - avg(sum by (mode)(rate(node_cpu_seconds_total{mode='idle',instance="\$node",job="\$job"}[\$__rate_interval]))) * 100) / count(count(node_cpu_seconds_total{instance="\$node",job="\$job"}) by (cpu)))

Legend

legend format

Min step

Resolution

1/1

Format

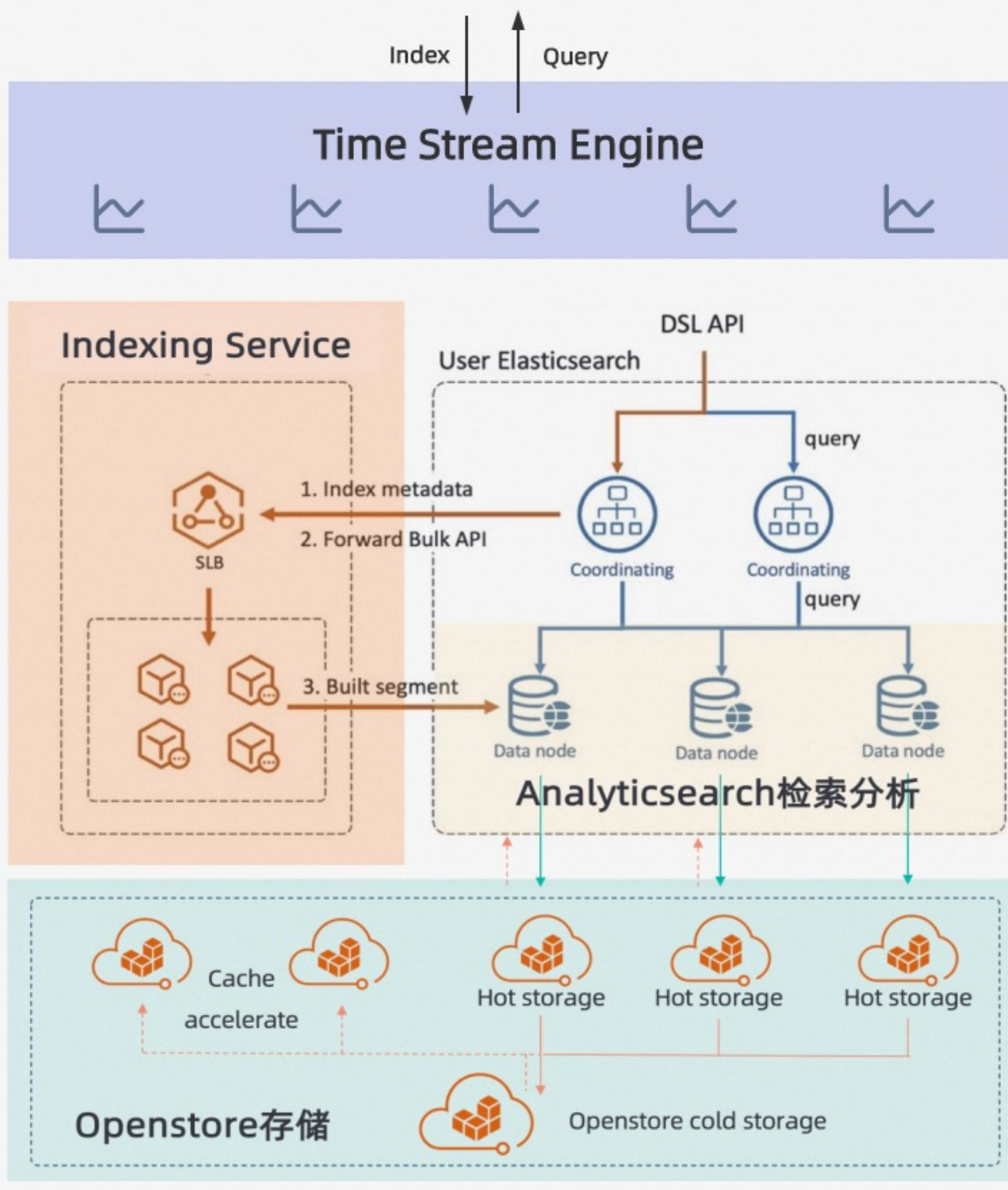
Time series

Instant

Prometheus

Exemplars

基于云原生存算分离引擎，提供Serverless时序服务



Elasticsearch Serverless服务

应用管理

Elasticsearch Serverless服务 / 概览

概览

欢迎体验Elasticsearch Serverless服务

Elasticsearch Serverless是基于云原生Serverless技术架构打造的Elasticsearch服务化产品，围绕Elasticsearch开源技术生态，提供端到端的数据接入、数据管理、数据应用及可视化的全链路产品能力。用户无须管理集群资源及配置，按实际业务读写流量、存储量等按需计费。为用户提供简单易用、弹性灵活、开箱即用的产品体验，做到快速响应业务变化的同时，合理优化使用成本，助力企业降本增效。[产品详情](#)

使用引导

1. 创建应用

2. 数据接入

3. 数据应用

快速入门

应用是Elasticsearch Serverless的基础服务单元，您无须规划集群资源配置，一键创建Serverless应用，即可在应用内获取读写访问地址、管理用量信息，按业务实际产生的读写流量及数据量按需使用

前往创建

前往Demo应用

最新推荐

使用教程

技术博客

更多

快速入门

创建一个应用，完成日志和指标数据流创建和数据接入，使用全观测应用中心进行可视化查询分析

查看详情

日志运维分析实践

当您需要查看并分析ECS的指标数据时，可通过Prometheus采集ECS日志并发送到阿里云Elasticsearch Serverless中，并在控制台完成可视化查询、分析和展示。

查看详情

Elasticsearch TSDB总结：

- 补齐了ES存储和查询时序数据的短板
- 支持云原生存算分离架构和Serverless TSDB服务
- 不止于TSDB

Elasticsearch优势：

- 系统方面：分布式、高可用、高可靠、弹性、分层存储、数据备份等
- 产品方面：Elastic Stack (ELK、ML、Security)
- 领域方面：搜索、可观察性、安全



感谢观看



专业、垂直、纯粹的 Elastic 开源技术交流社区

<https://elasticsearch.cn/>