

亿级规模的ES查询优化实战

主讲人：何金城



- 何金城 塔布数据高级项目经理
- 参与重大项目：
 - ◆ 某舆情监控公司爬虫
 - ◆ 某家电企业用户画像
- Github：
<https://github.com/sezina>
- Blog:
<http://sezina.github.io/>
- Wechat: sezina00

目录

1

◆ 背景

2

◆ 发现问题

3

◆ 分析问题

4

◆ 优化实战

5

◆ What next?



01

背景



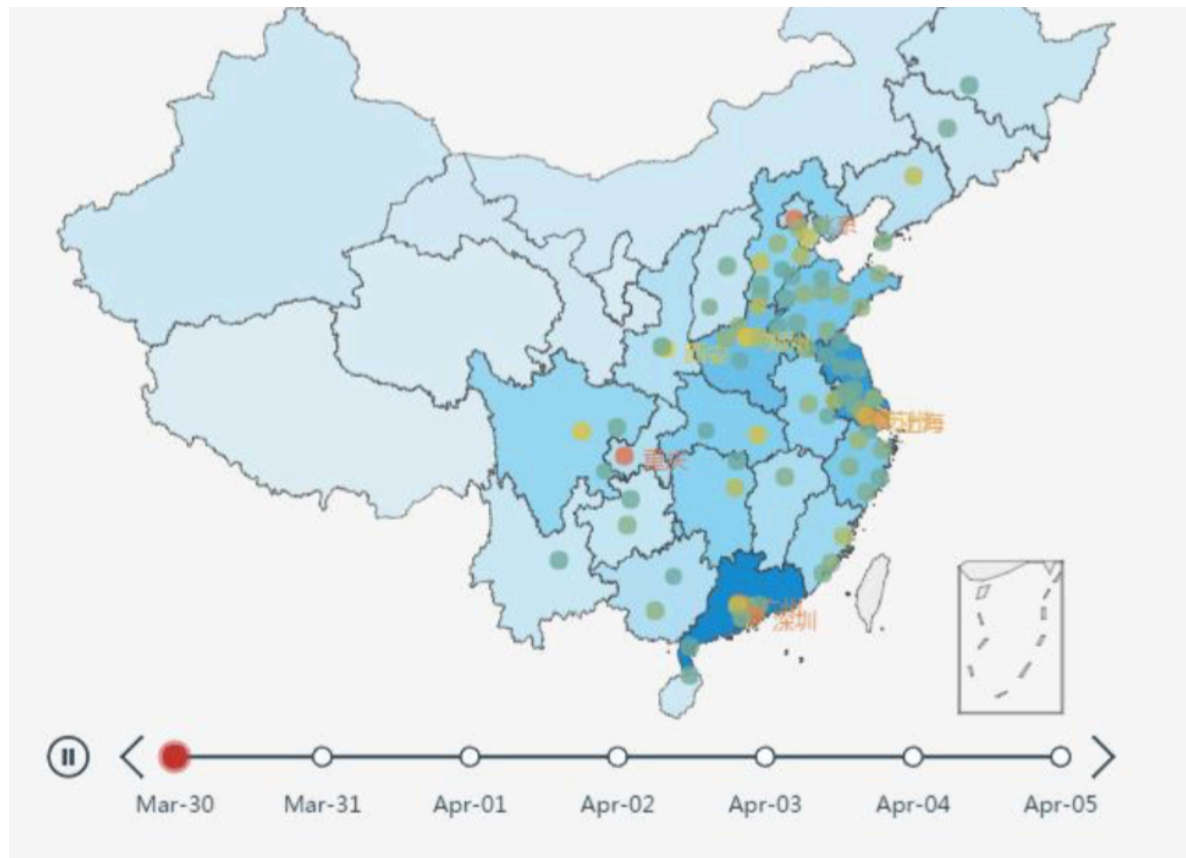
1亿+ 用户

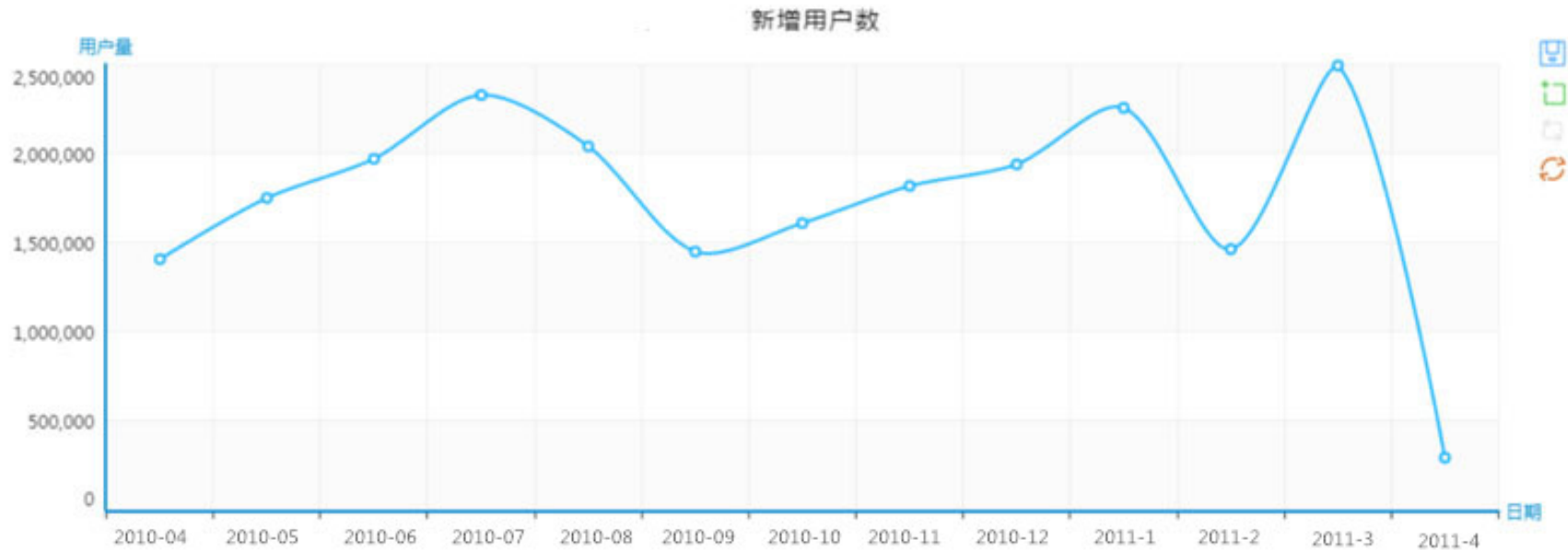


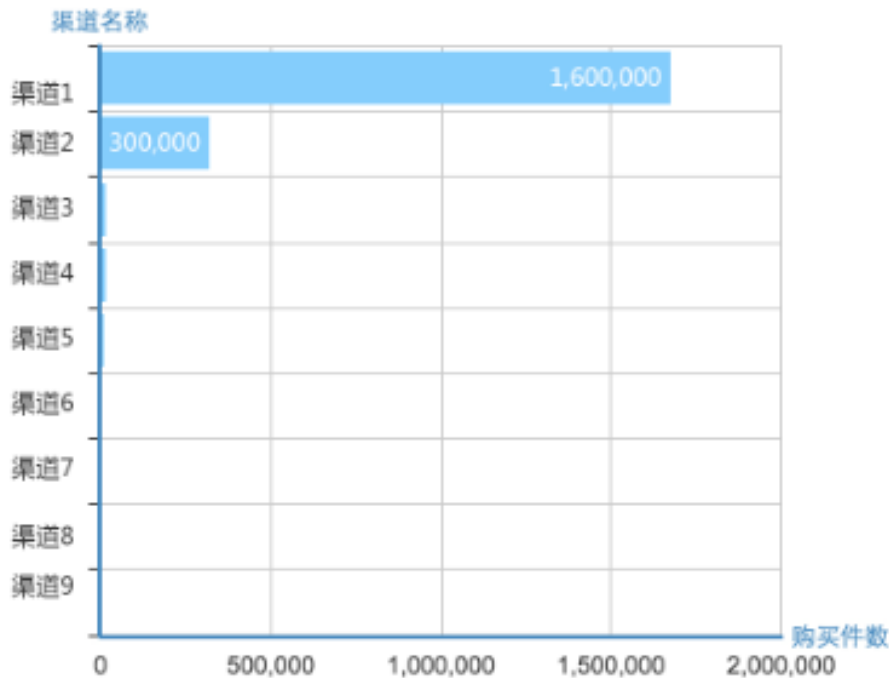
20亿 关联数据

500+GB







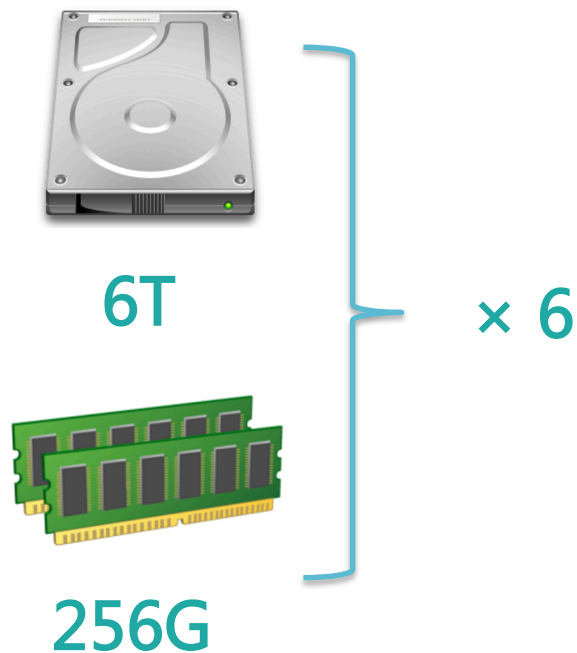


渠道名	次数
渠道1	1600000
渠道2	310000
渠道3	15000
渠道4	15000
渠道5	7000
渠道6	3000
渠道7	800
渠道8	700
渠道9	1



02

发现问题



1亿+ 用户

20亿 关联数据
500+GB

客户要求：响应小于**2000ms**

Elasticsearch **连接 es_dmp 集群健康值: green**

概览 索引 数据浏览 基本查询 [+] 复合查询 [+]

历史记录

查询

POST

```
{
  "terms": {
    "field": "productInfo.timeStr",
    "size": 13,
    "min_doc_count": 0,
    "order": {
      "_term": "desc"
    },
    "include": "2016-03|2016-02|2016-01|2015-12|2015-11|2015-10|2015-09|2015-08|2015-07|2015-06|2015-05|2015-04|2015-03"
  }
}
```

提交请求 验证 JSON 易读

结果转换器 重复请求 显示选项

4030ms

```
{
  "took": 4030,
  "timed_out": false,
  "_shards": {
    "total": 6,
    "successful": 6,
    "failed": 0
  },
  "hits": {
    "total": 9315,
    "max_score": 1,
    "hits": [
      {
        "_id": "1",
        "productInfo": {
          "doc_count": 9315,
          "filter": {
            "doc_count": 9315,
            "productInfo.time": {
              "doc_count_error_upper_bound": 0,
              "sum_other_doc_count": 0,
              "buckets": [
                {
                  "key": "2016-03",
                  "doc_count": 9315
                }
              ]
            }
          }
        },
        "key": "2016-03",
        "doc_count": 9315
      },
      {
        "key": "2016-02",
        "doc_count": 2001
      },
      {
        "key": "2016-01",
        "doc_count": 1180
      }
    ]
  }
}
```

Elasticsearch **连接 es_dmp 集群健康值: green**

概览 索引 数据浏览 基本查询 [+] 复合查询 [+]

历史记录

查询

POST

```
{
  "field": "productInfo.province",
  "size": 100,
  "aggregations": {
    "productInfo.city": {
      "terms": {
        "field": "productInfo.city",
        "size": 100
      }
    }
  }
}
```

提交请求 验证 JSON 易读

结果转换器 重复请求 显示选项

2404ms

```
{
  "took": 2404,
  "timed_out": false,
  "_shards": {
    "total": 6,
    "successful": 6,
    "failed": 0
  },
  "hits": {
    "total": 168038,
    "max_score": 1,
    "hits": [
      {
        "_id": "1",
        "productInfo": {
          "doc_count": 168038,
          "filtered": {
            "doc_count": 168038,
            "productInfo.province": {
              "doc_count_error_upper_bound": 0,
              "sum_other_doc_count": 0,
              "buckets": [
                {
                  "key": "广东省",
                  "doc_count": 168038,
                  "productInfo.city": {
                    "doc_count_error_upper_bound": 0,
                    "sum_other_doc_count": 0,
                    "buckets": [
                      {
                        "key": "广州市",
                        "doc_count": 36485
                      },
                      {
                        "key": "深圳市",
                        "doc_count": 34746
                      }
                    ]
                  }
                }
              ]
            }
          }
        },
        "key": "广东省",
        "doc_count": 168038
      }
    ]
  }
}
```




03

分析问题

Bad case



Bad case

Field: startTime

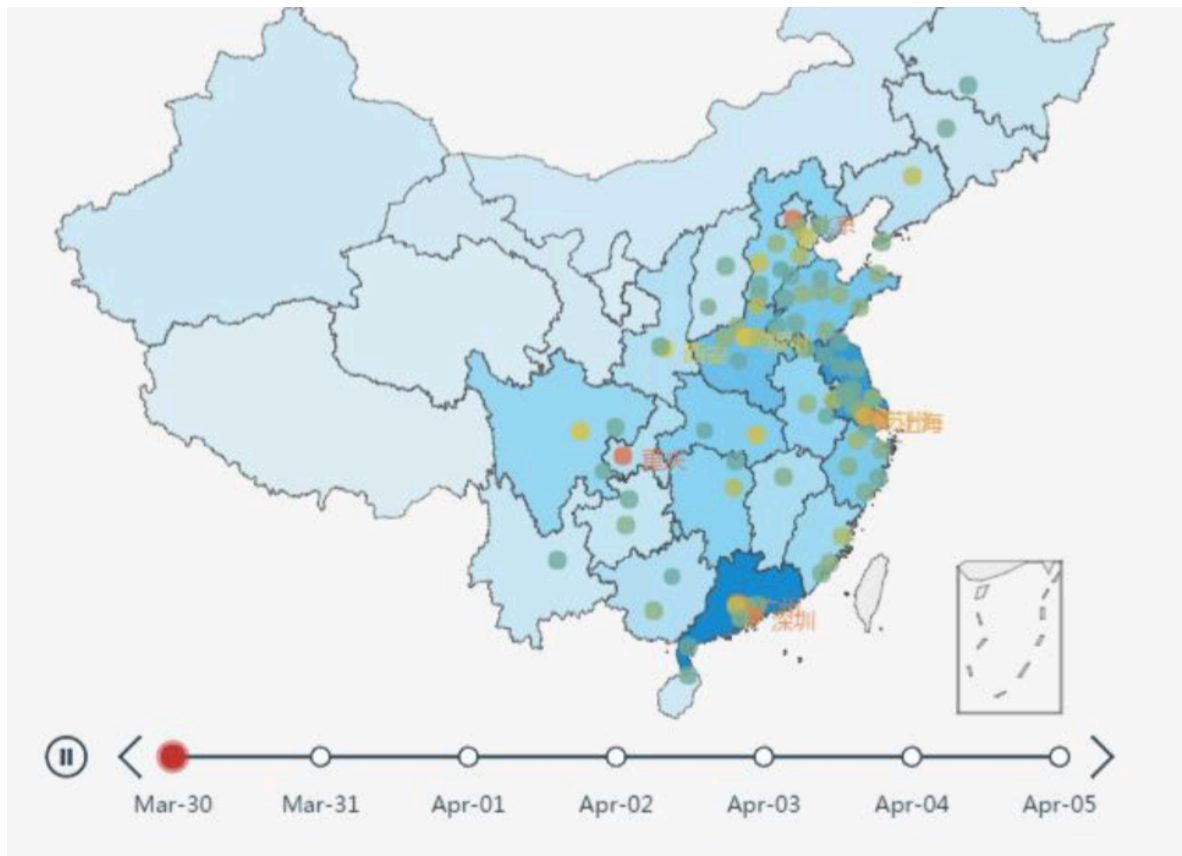
Aggregation:

date_range

---> date_histogram

```
{
  "aggs": {
    "one_year": {
      "date_range": {
        "field": "startTime",
        "ranges": [
          {
            "from": "<start_timestamp>",
            "to": "<to_timestamp>"
          }
        ]
      },
      "aggs": {
        "user_trend": {
          "date_histogram": {
            "field": "startTime",
            "interval": "1m",
            "format": "yyyy-MM",
            "min_doc_count": 0,
            "order": {
              "_key": "desc"
            }
          }
        }
      }
    }
  }
}
```


Bad case





Field:
product.time &
product.province

Aggregation:
nested
---> date_range
-----> date_histogram
-----> terms



product的字段
time
org
segment
.....

基础字段
startTime
birthdate
.....

address的字段
province
city
.....

- 我在什么字段上做aggregation？这些字段是否会导致查询变慢？
- 这些字段的什么属性导致了查询变慢？
- 我的query或filter写错了吗？
- 我的aggregation写错了吗？
- 那么大的内存我用好了吗？

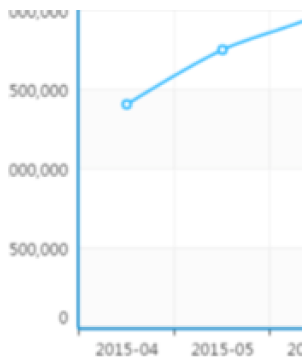


03

优化方案

优化方案—1、转range为terms

- 增加冗余字段，为date类型的startTime，增加一个string类型的格式为“**yyyy-MM**”的startTimeStr字段



```
"startTimeStr": {  
  "type": "string",  
  "index": "not_analyzed",  
  "store": "no",  
  "doc_values": true,  
  "fielddata": {  
    "loading": "eager"  
  }  
},
```

Aggregation:
date_range
---> date_histogram



Aggregation:
terms aggs

优化方案—2、为字段增加配置



- 为所有需要用到字段增加一个配置，让数据尽量缓存到内存

```
"fielddata": {  
  "loading": "eager"  
}
```

```
"provinceName": {  
  "type": "string",  
  "index": "not_analyzed",  
  "store": "no",  
  "doc_values": true  
},
```



```
"provinceName": {  
  "type": "string",  
  "index": "not_analyzed",  
  "store": "no",  
  "doc_values": true,  
  "fielddata": {  
    "loading": "eager"  
  }  
},
```




能用filter就不用query

- ◆ filter拿到相应的doc后不计算score不用排序
- ◆ query会对符合条件的doc计算score并进行排序
- ◆ filter的查询速度比query快很多



增加相关cache的配置

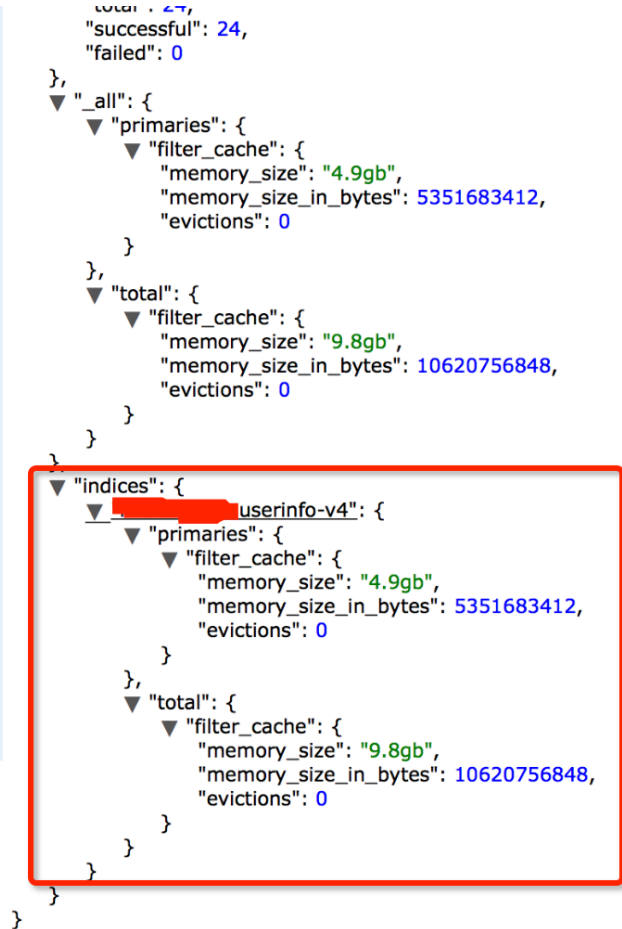
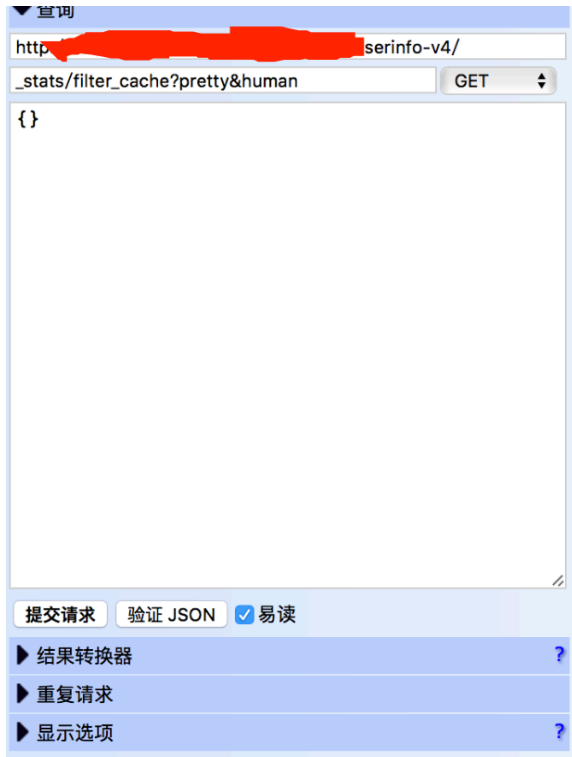
- ◆ `indices.cache.filter.size: 30%`
- ◆ `indices fielddata.cache.size: 60%`
- ◆ `index.cache.field.type: soft`
- ◆ `indices.breaker.fielddata.limit: 70%`



4030ms → 421ms → 1ms

2404ms → 308ms → 3ms

优化方案——效果





- ◆ Global ordinals
- ◆ Index warmer
- ◆ 考虑调整aggregation的collect_mode, breadth_first or depth_first



- 能用filter就不用query
- 增加冗余字段将部分range aggregation查询变成terms aggregation
- 为常用字段增加配置，将fielddata的loading设成eager，尽量多加载到内存
- 增加集群的缓存资源，把内存尽量多的用起来
- Global ordinals
- Index warmer
- 调整aggregation的collect_mode
- 上SSD



04

What next?



- ES源码改造
 - 改造查询接口，改造文档的权重计算，通过小文档计算大文档权重；根据文本中的某些词语的大小、颜色计算权重。
 - 修改highlight，定制特定场景下的highlight
- TB级索引优化，性能要求：单节点2000w文档，500ms响应时间

Thank you !
感谢聆听 !
We are hiring!

广州智索信息科技有限公司

邮编 : 510600 电话 : (020) 38891741 传真 : (020) 38891740

地址 : 广州市天河区珠江新城黄埔大道西100号富力盈泰广场A塔13A