

百亿数据下ES性能优化

袋鼠云高级运维工程师—河图

CONTENT



01 日志分析的价值

02 ES集群性能优化

03 云日志平台的应用

需求背景：

- 业务发展越来越快，系统间关联复杂，服务器越来越多。
- 各种访问日志、应用日志、错误日志的文件数量越来越多。
- 运维、开发人员排查问题时，需要到服务器上找日志，非常不方便。
- 运营小姐姐统计一些数据，也需要到服务器上分析日志。

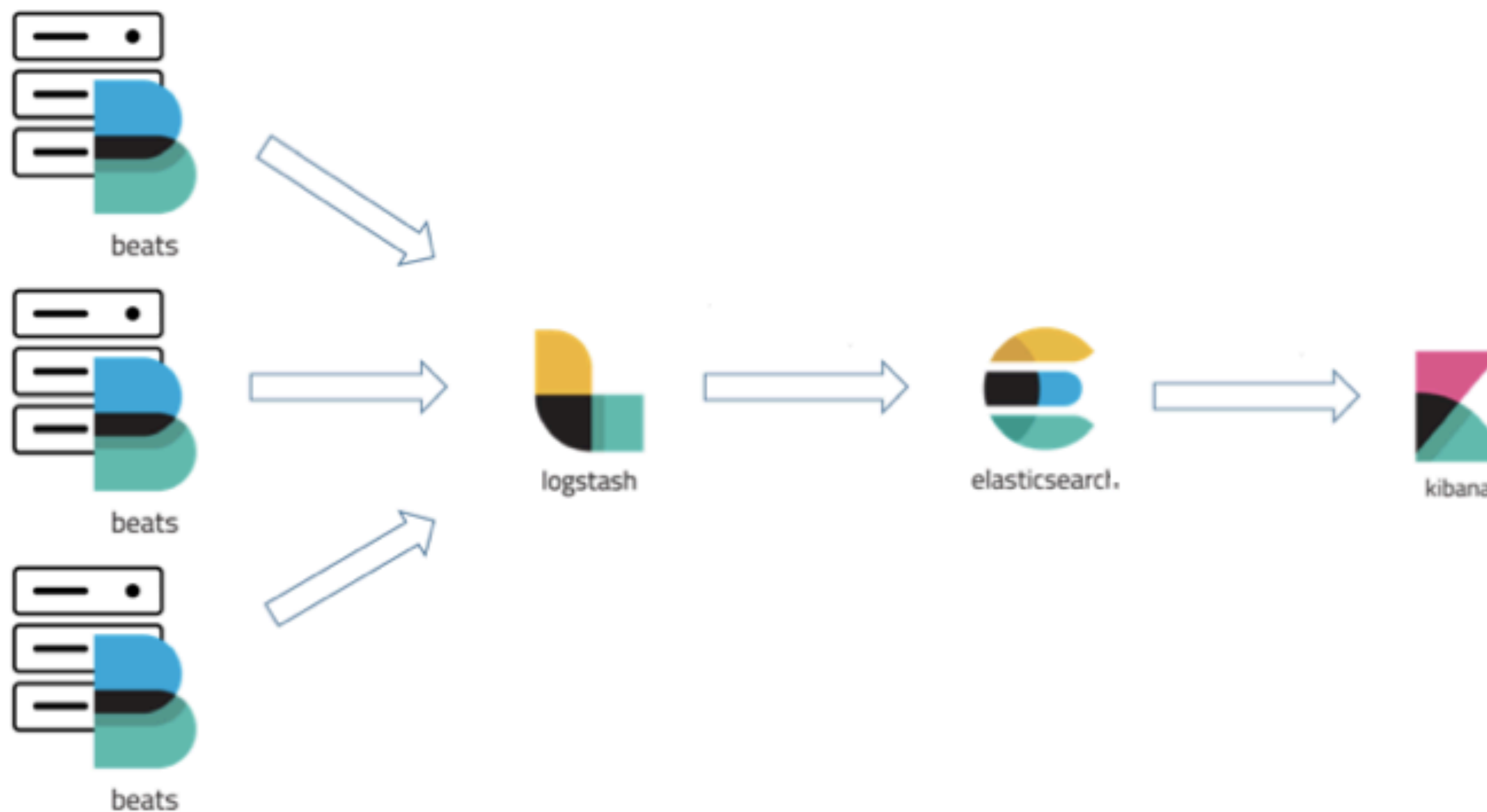
那咋办呢？



基础ELK架构-适用测试环境



ELKB架构-适用小规模日志场景



ELKB+Kafka架构-适用大规模日志场景



袋鼠云jlogstash开源项目-Logstash数据处理性能提升5倍

<https://github.com/DTStack/jlogstash>

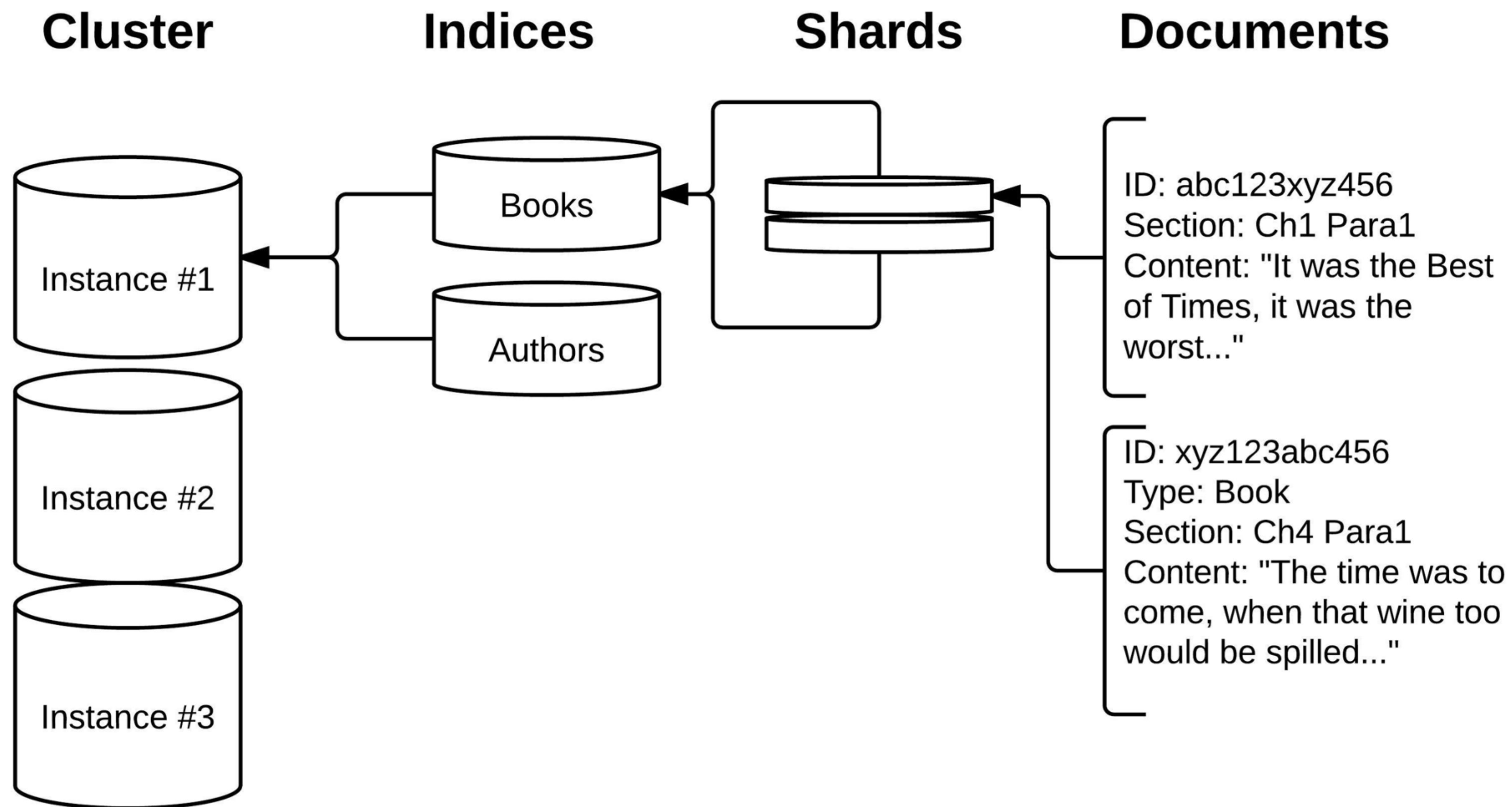
CONTENT



01 日志分析的价值

02 ES集群问题及优化

03 云日志平台的应用



某大型互联网传媒公司

客户情况：

- 日志数据日增2TB（约百亿条数据），峰值40万/秒
- 基于基础ELK架构、拥有80个节点。
- 索引写入速度不稳定
- 大搜索时搜索超时，但ES持续负载飆高
- 频繁young gc、full gc



问题1：百亿数据ES集群的规划

核心问题

- 日志数据增量大小
- 日志保留时长

小规模数据量方案：

日志增量	索引推荐个数	保存天数	ES集群配置
10G	1天1~2个	5天~10天	3台4c8g 200G~500G
100G	1天10个~20个	5天~10天	master/data分离 master: 2c4g data: 5台8c 16g 1T~2T
500G	1天30~40个	5天~10天	Hot-Warm Hot: 5台16c 32g 4T Warm: 4C 16G 4T
1T+++			

那大规模数据数据ES集群如何规划?

| 百亿数据ES集群规划解决方案及结果



规划方案

- 节点规划：master、data分离；冷热数据分离
- 内存规划：30G的Heap 大概能处理10T的索引数据
超过64G内存的机器运行多个ES实例
- 硬盘规划：多块磁盘；SSD
- 索引规划：划分业务系统，提前创建索引
大索引按天分隔
小索引使用aliases
- 分片规划：每个分片不超过40GB
每个节点不超过3个分片
- 刷新时间：5s（告警），10s（状态指标），
15~30s（文本日志）

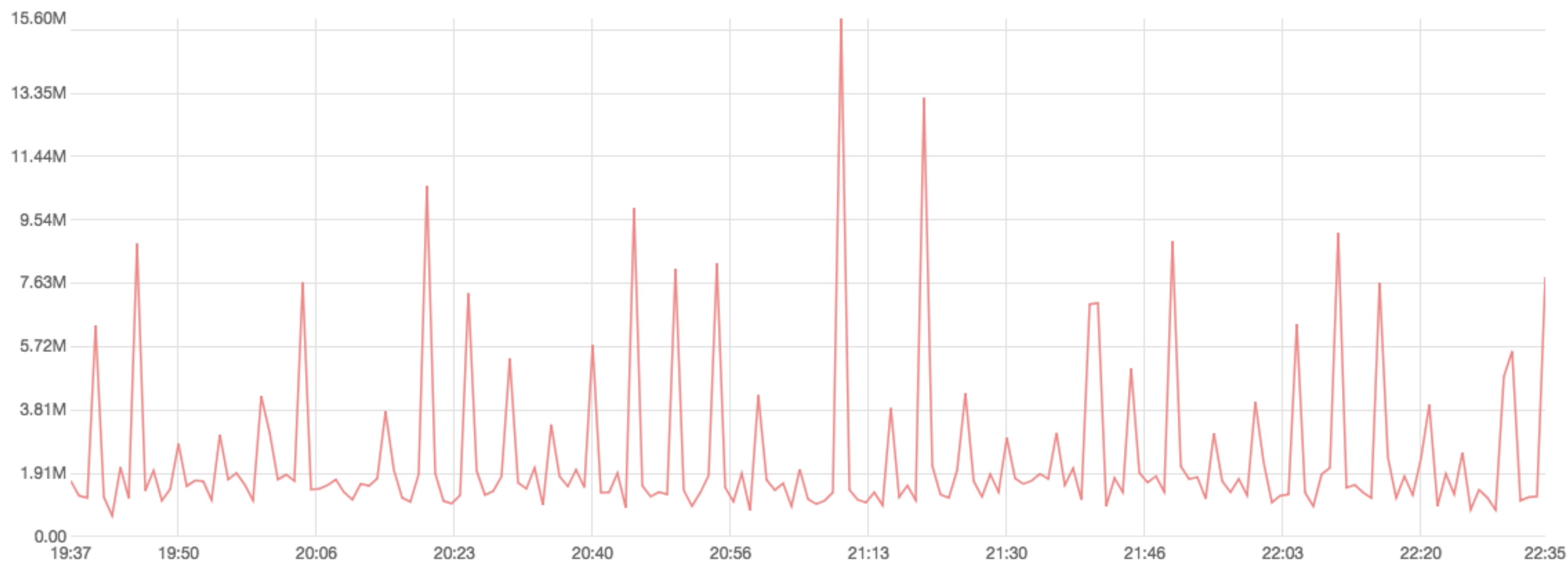
规划结果

- 80+机器优化到30+
- 接入400+个服务器，日增80+个索引，
3TB+索引
- 120亿+条/天，峰值40w条/s，日志保留10天

问题2：索引写入不稳定

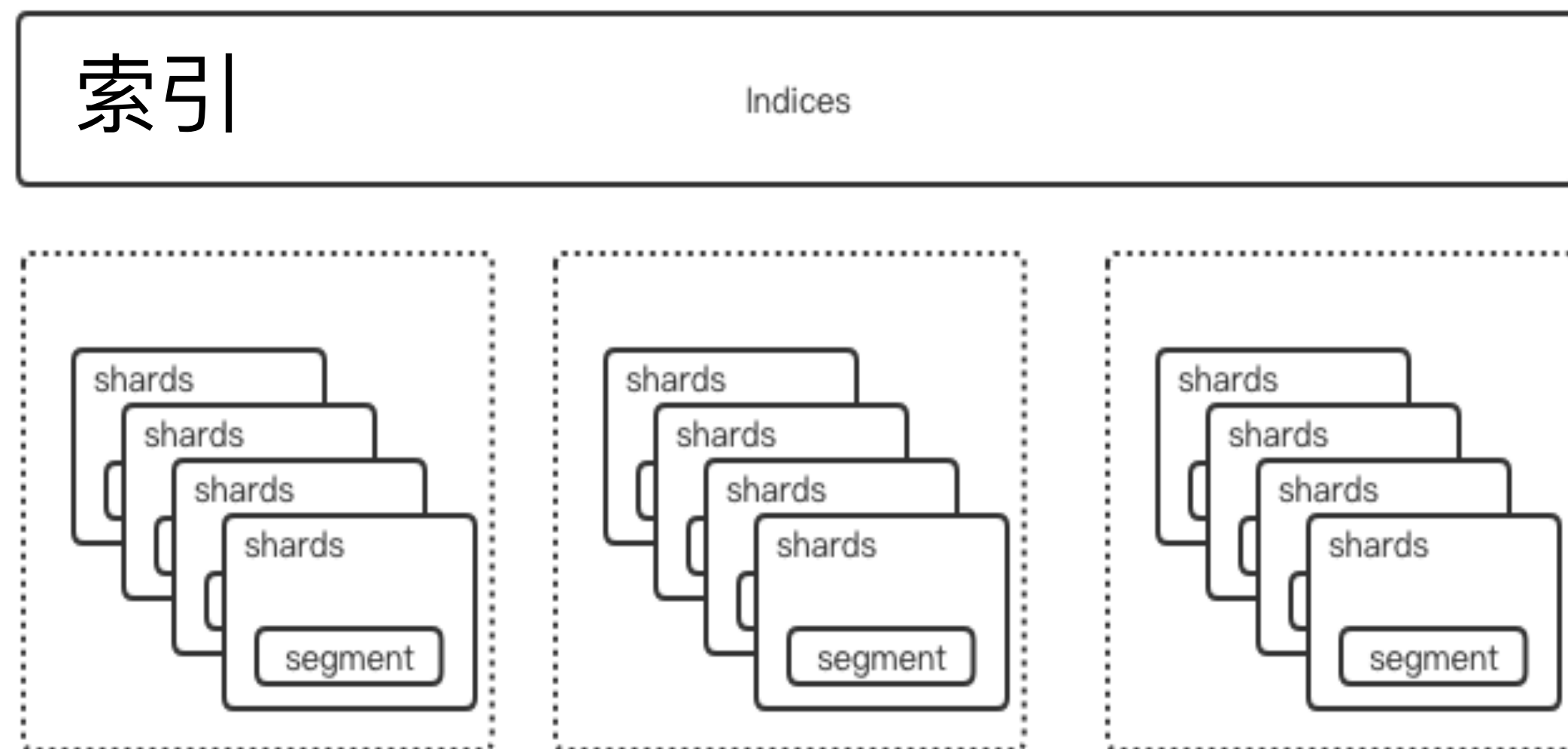
现象描述：

- 索引的写入抖动非常剧烈



问题2：索引写入不稳定

ES索引写入原理：从小文件合并成大文件



合并结果：

```
"merges": { -  
  "current": 0,  
  "current_docs": 0,  
  "current_size": "0b",  
  "current_size_in_bytes": 0,  
  "total": 2957,  
  "total_time": "23.6h",  
  "total_time_in_millis": 85093636,  
  "total_docs": 1592718203,  
  "total_size": "1.3tb",  
  "total_size_in_bytes": 1450593079700,  
  "total_stopped_time": "47.6m",  
  "total_stopped_time_in_millis": 2860319,  
  "total_throttled_time": "0s",
```

```
"segments": { -  
  "count": 335,  
  "memory": "1gb",  
  "memory_in_bytes": 1179061150,  
  "terms_memory": "980.9mb",
```

合并过程



问题2：索引写入不稳定

Merge默认参数：

```
"merge" : {  
  "scheduler" : {  
    "auto_throttle" : "true",  
    "max_merge_count" : "9"  
  },  
  "policy" : {  
    "reclaim_deletes_weight" : "2.0",  
    "floor_segment" : "2mb",  
    "max_merge_at_once_explicit" : "30",  
    "max_merge_at_once" : "10",  
    "max_merged_segment" : "5gb",  
    "expunge_deletes_allowed" : "10.0",  
    "segments_per_tier" : "10.0"  
  }  
},
```

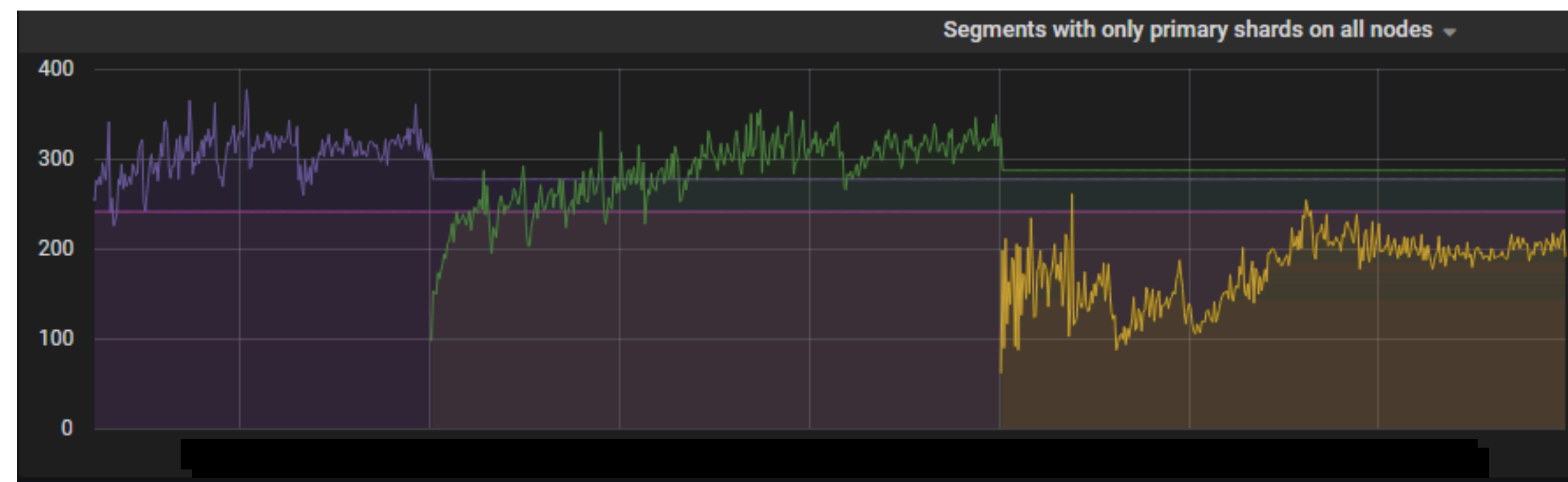
默认大小

默认个数

那Merge应该如何调整



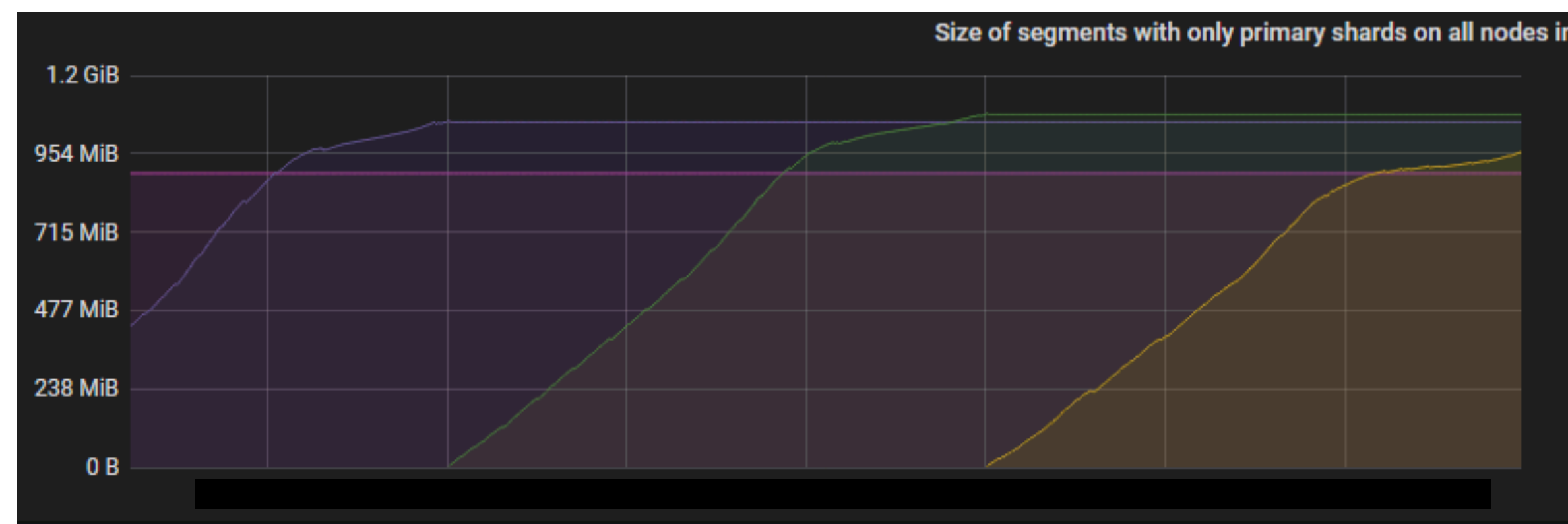
Segment数量情况



默认大小
默认个数

大小增大10倍
默认个数

个数增加3倍
默认大小



Segment内存情况

I 问题2：索引写入不稳定

调整Merge策略

- 针对大索引：

max_merge_at_once: 10
max_merged_segment: 10gb
segments_per_tier: 10
floor_segment: 20mb

- 针对小索引：

max_merge_at_once: 30
max_merged_segment: 5gb
segments_per_tier: 30
floor_segment: 10mb

- 优化结果：

数据写入非常平稳



问题3：大数据搜索超时，但ES持续负载飚高

现象描述：

- 在进行历史数据搜索时发现频繁搜索超时
- 超时后ES负载依旧很高
- 且偶尔伴随着OOM

土豪方案：加内存！

这个问题充钱就能解决！



- 调整Linux系统参数：
- `vm.swappiness=1`

然而土豪方案没有从根本解决问题
还有别的方案吗

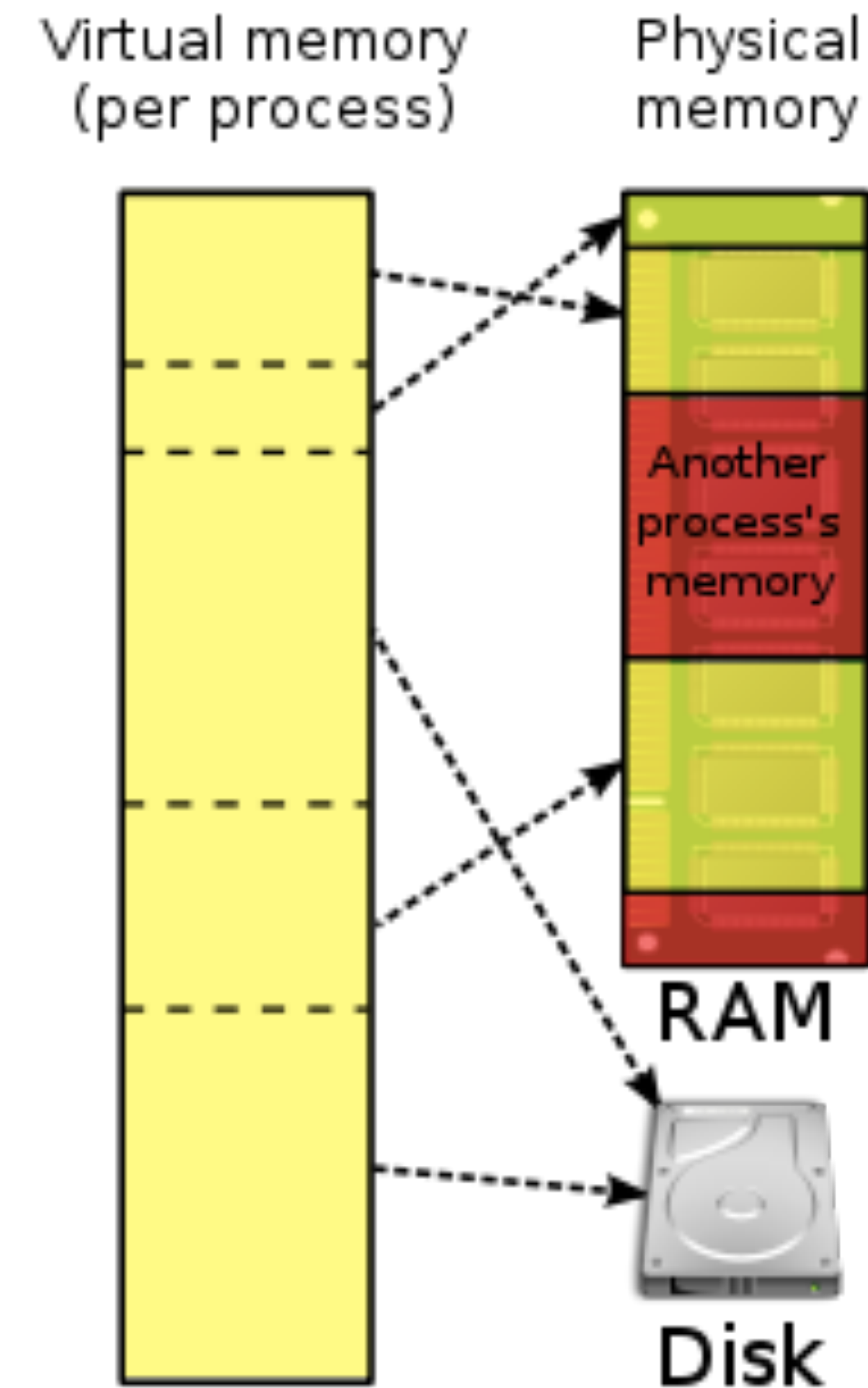


问题3：大数据搜索超时，但ES持续负载飚高

高性价比的方案——调整索引存储策略

`index.store.type`

- `mmapfs`（默认） -- 适用于小索引
- `niofs` -- 适用于大索引、历史索引



I 问题4：频繁的young gc和Full gc

观察到的现象：

- 新生代垃圾回收频繁；
- full gc耗时长，导致节点失联

调整策略1：

调整cms gc开始时间

-XX:CMSInitiatingOccupancyFraction=70

调整后问题：

Full gc 间隔时间长但是节点失联更严重

调整策略2：

调整jvm heap比例

新生代:老年代 == 1:4

-XX:NewRatio=4

调整后问题：

需要不断优化，动态调整该值
每次调整需要重启生效

I 问题4：频繁的young gc和Full gc

终极策略—调整GC

CMS  G1

调整值：

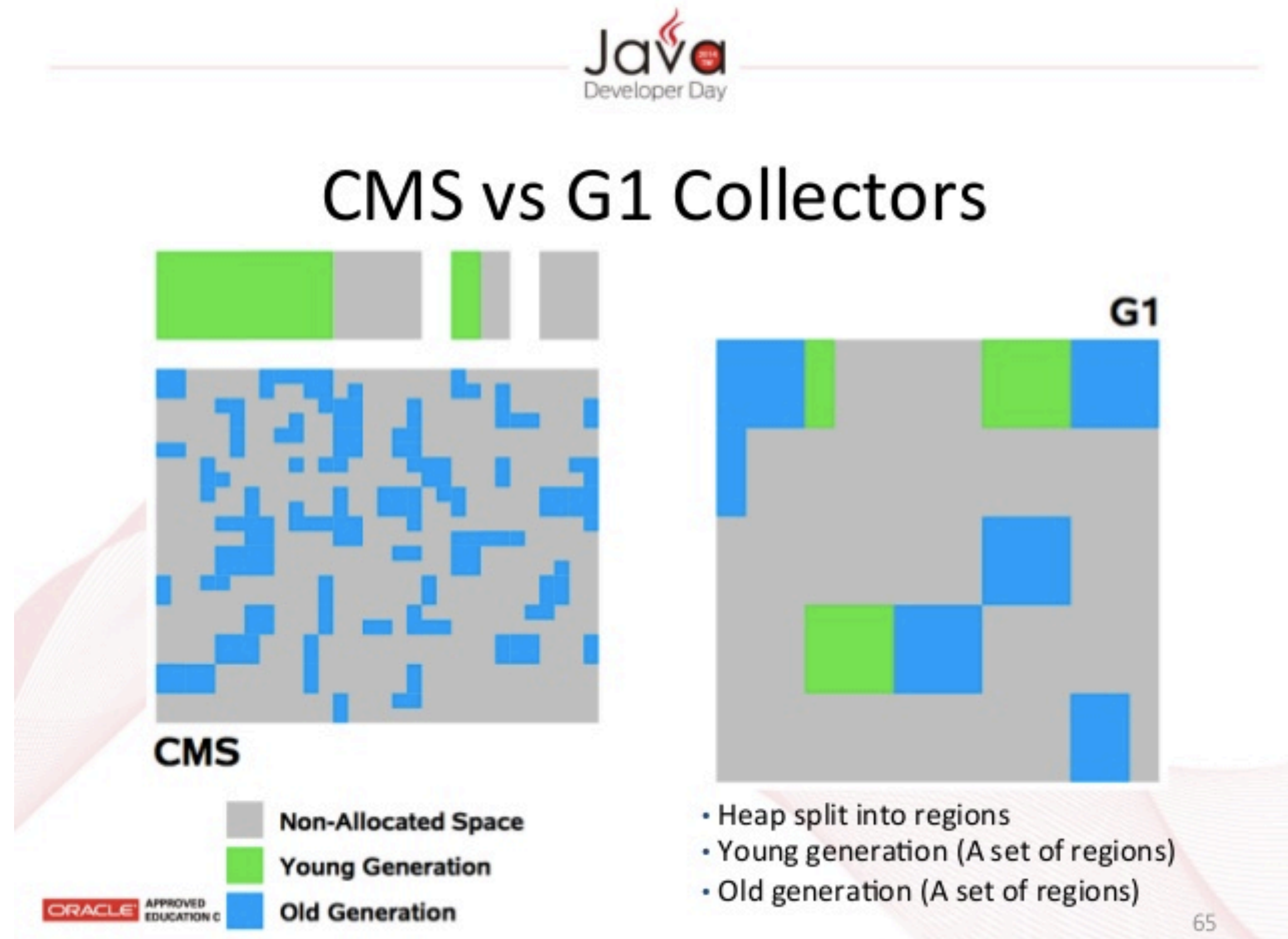
-XX:+UseG1GC

-XX:MaxGCPauseMillis=100

-XX:GCPauseIntervalMillis=1000

-XX:InitiatingHeapOccupancyPercent=35

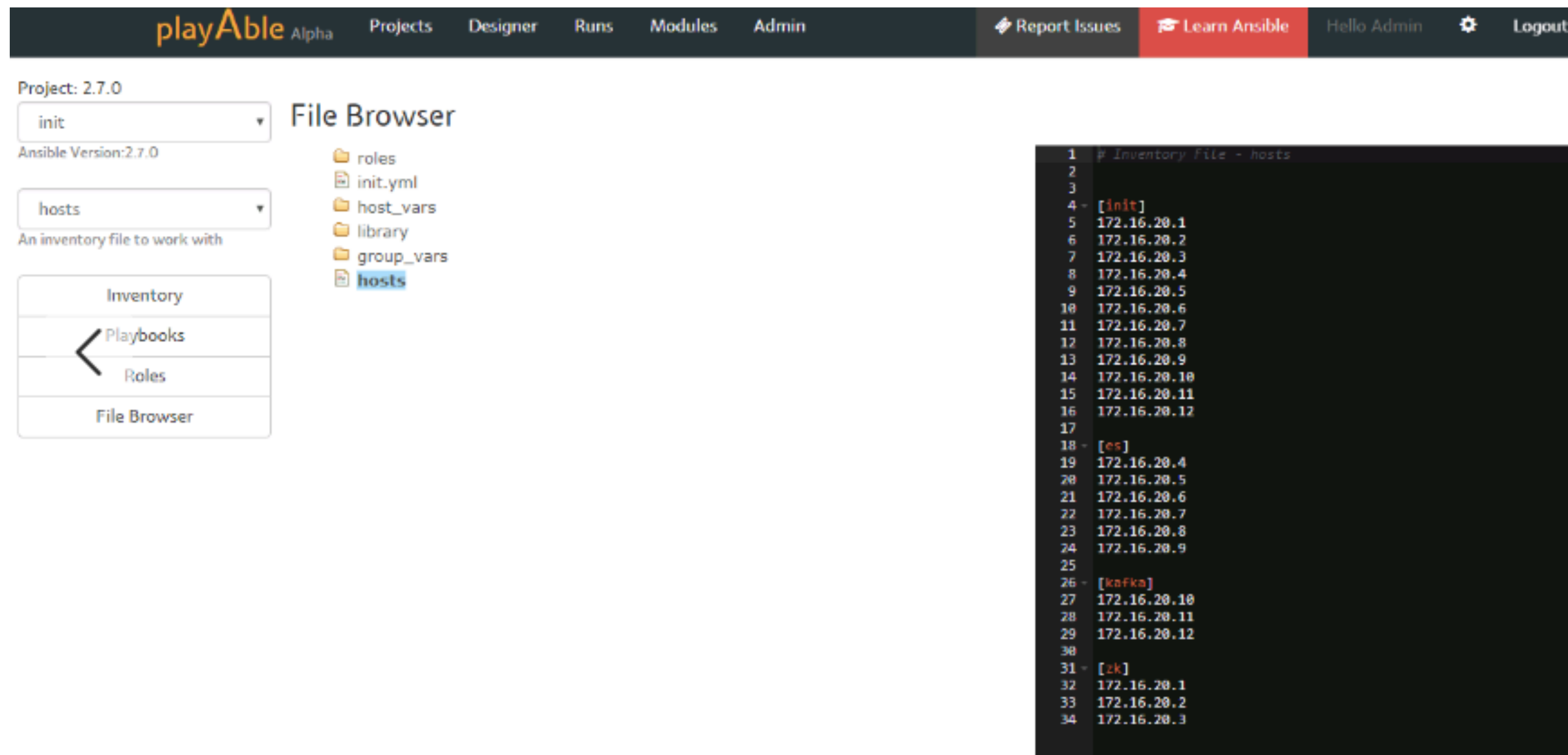
➤ 调整后目前还未出现full gc



| 其他：ES集群部署

部署&扩容：

- Ansible
- Ansible-playable



The screenshot displays the playAble web interface. The top navigation bar includes links for Projects, Designer, Runs, Modules, Admin, Report Issues, Learn Ansible, Hello Admin, and Logout. The main content area shows the 'File Browser' for a project named '2.7.0'. The 'init' dropdown menu is selected, and the 'hosts' file is highlighted in the file list. Below the file list, a sidebar shows a navigation menu with 'Inventory', 'Playbooks', 'Roles', and 'File Browser'. The 'File Browser' is currently selected. To the right of the file list, a terminal window displays the contents of the 'hosts' file, which lists IP addresses for three groups: 'init', 'es', and 'zk'.

```
1 # Inventory file - hosts
2
3
4 - [init]
5 172.16.20.1
6 172.16.20.2
7 172.16.20.3
8 172.16.20.4
9 172.16.20.5
10 172.16.20.6
11 172.16.20.7
12 172.16.20.8
13 172.16.20.9
14 172.16.20.10
15 172.16.20.11
16 172.16.20.12
17
18 - [es]
19 172.16.20.4
20 172.16.20.5
21 172.16.20.6
22 172.16.20.7
23 172.16.20.8
24 172.16.20.9
25
26 - [kafka]
27 172.16.20.10
28 172.16.20.11
29 172.16.20.12
30
31 - [zk]
32 172.16.20.1
33 172.16.20.2
34 172.16.20.3
```


| 其他：ES集群管控



进程管控：

- supervisor
- cesi

Cesi Dashboard Nodes Environments Groups Users Settings

Connected 5

- ☒ node1
- ☐ node2
- ☐ node3
- ☐ node4
- ☐ node5

Not-Connected 0

Processes for node1 2 Start All Stop All Restart All

Name	Group	Pid	Uptime	State	Action
elasticsearch	elasticsearch	16326	6 days, 18:18:08	RUNNING	<button>Start</button> <button>Stop</button> <button>Restart</button> <button>Log</button>
kafka	kafka	16024	3 days, 2:52:44	RUNNING	<button>Start</button> <button>Stop</button> <button>Restart</button> <button>Log</button>

索引管理：

- curator
- jiacrontab

jiacrontab 主页 说明

计划任务 常驻任务 运行信息

首页 / 计划任务列表 / 编辑

添加脚本

地址 localhost:20001

脚本名称 forcemerge

命令 docker run --rm --net=host curator /home/admin/.curator/forcemerge.yml pipe +

脚本超时 14400

超时操作 ☐ 邮件通知 ☒ api通知 ☐ 强杀 ☐ 邮件通知并强杀 ☐ 忽略

邮箱地址 monitor@dtstack.com

api地址

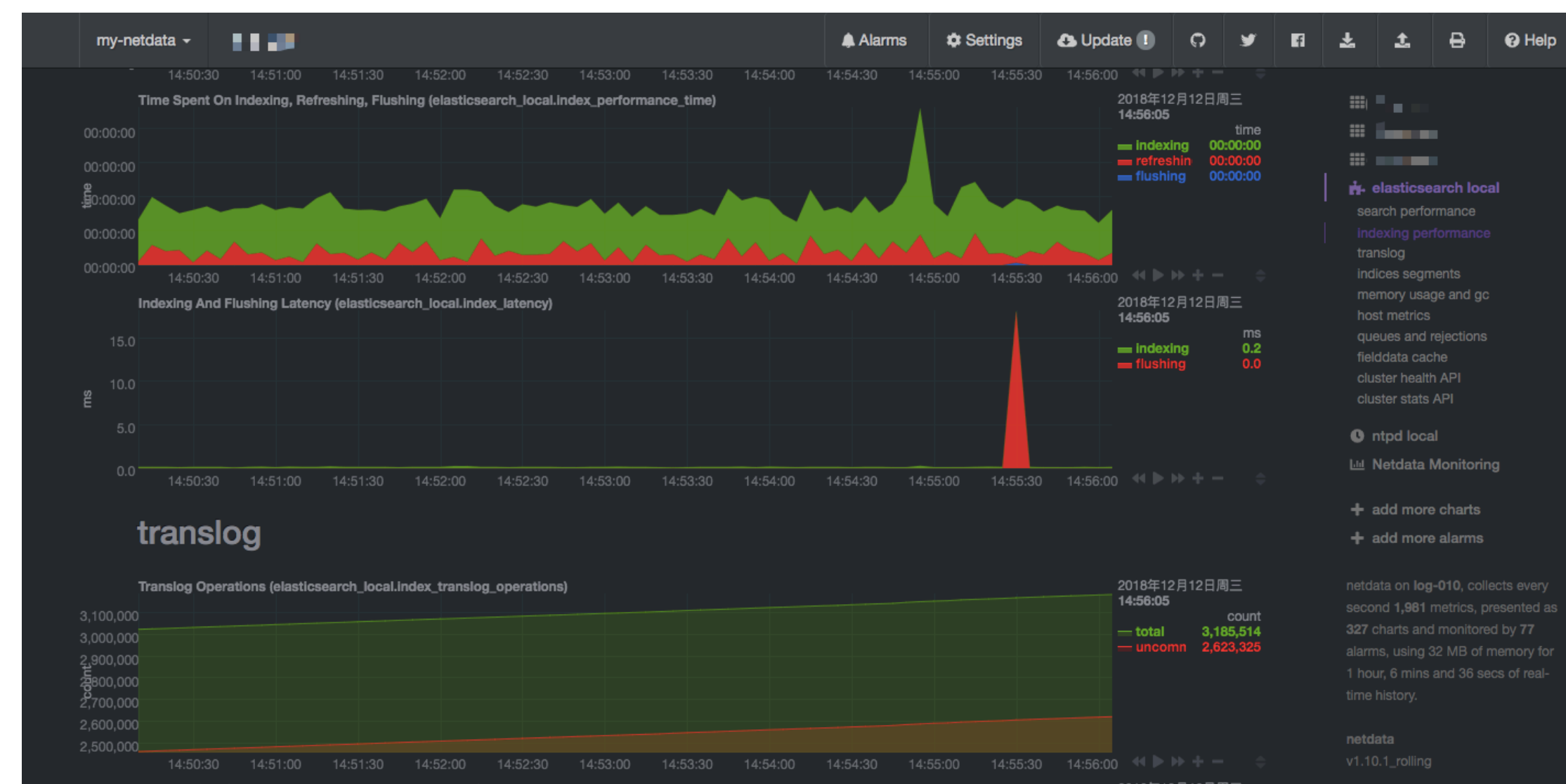
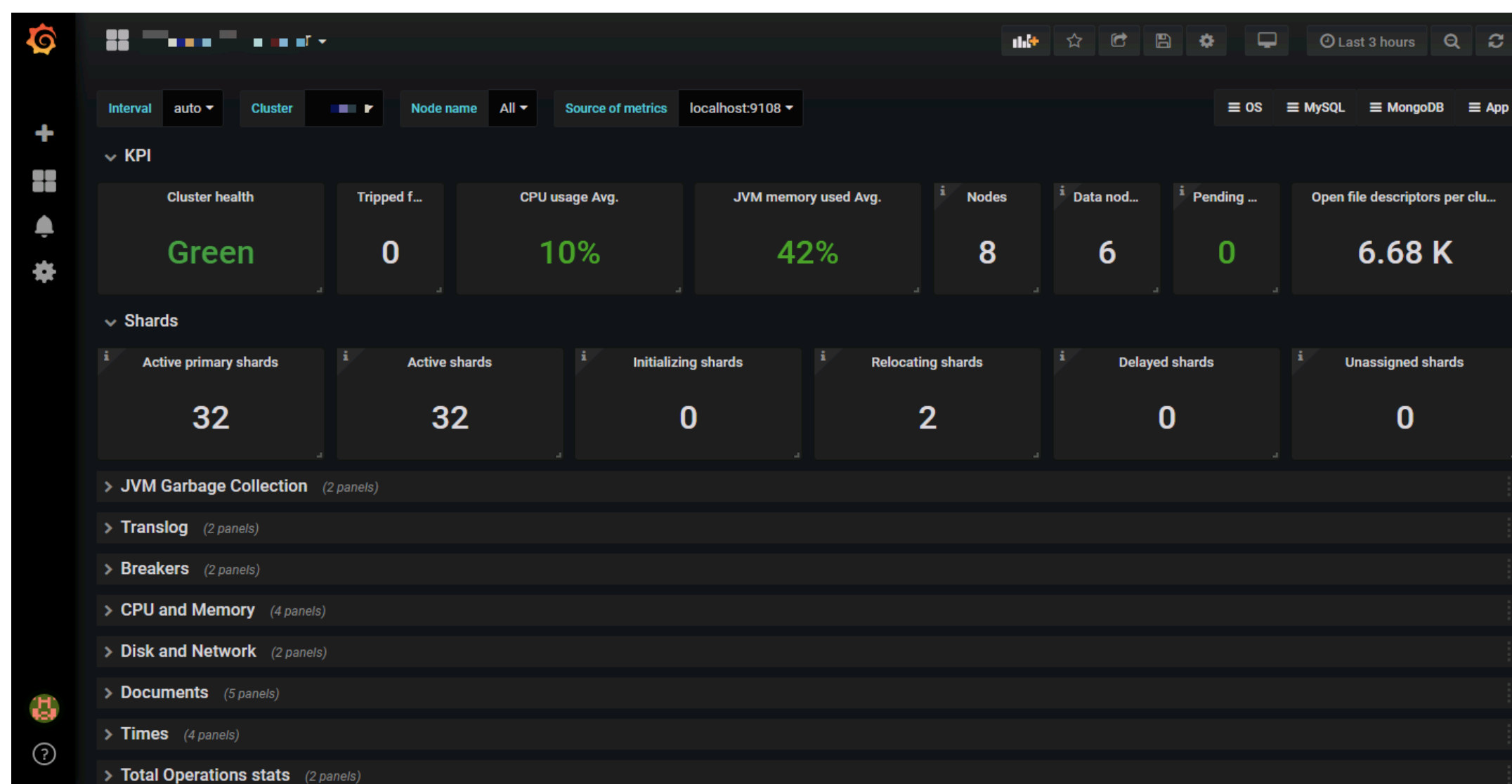
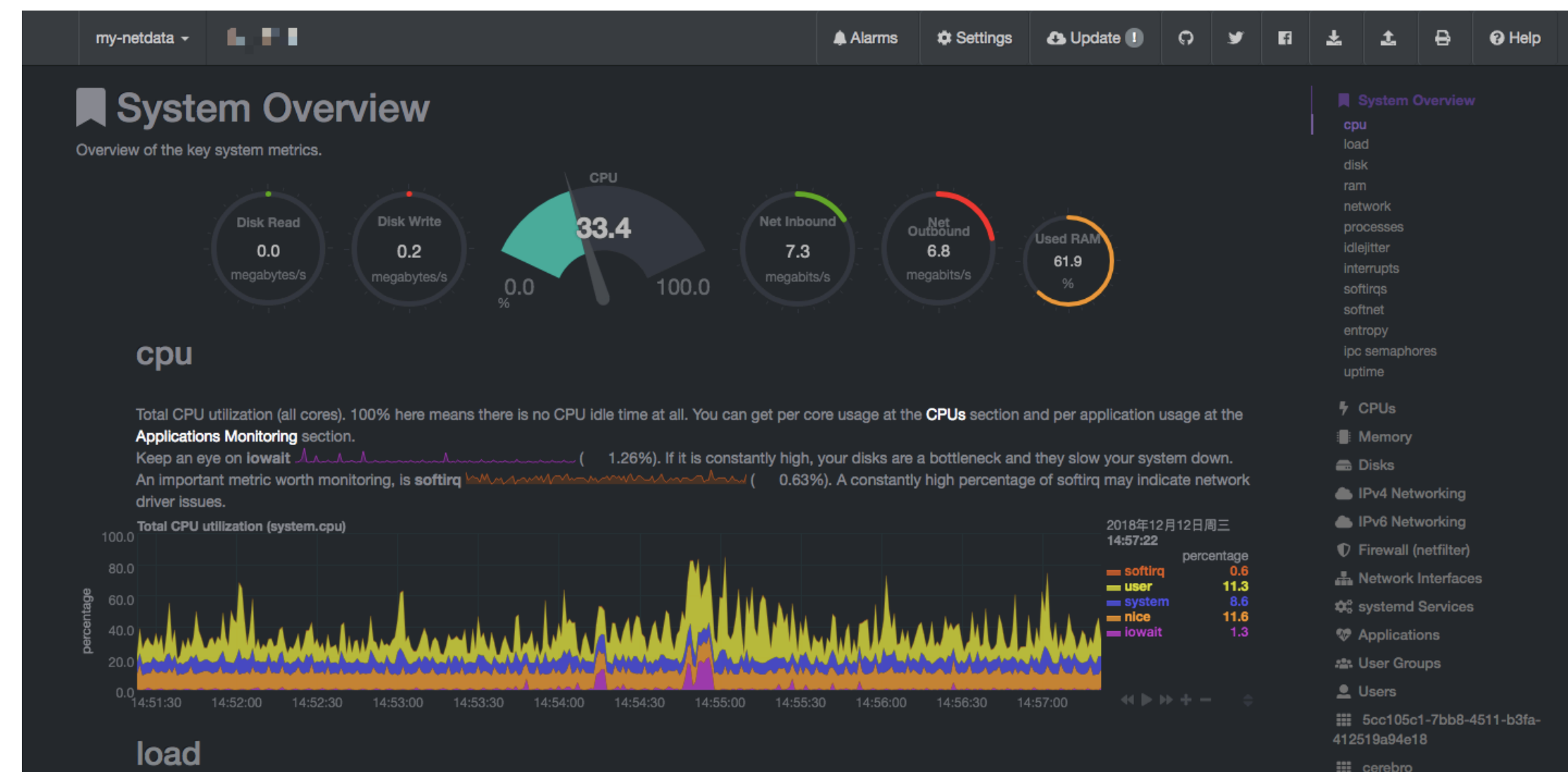
最大并发数 1

定时 分: 1 时: 1 日: * 月: * 周: *

| 其他：ES集群监控&性能问题排查

监控方案：

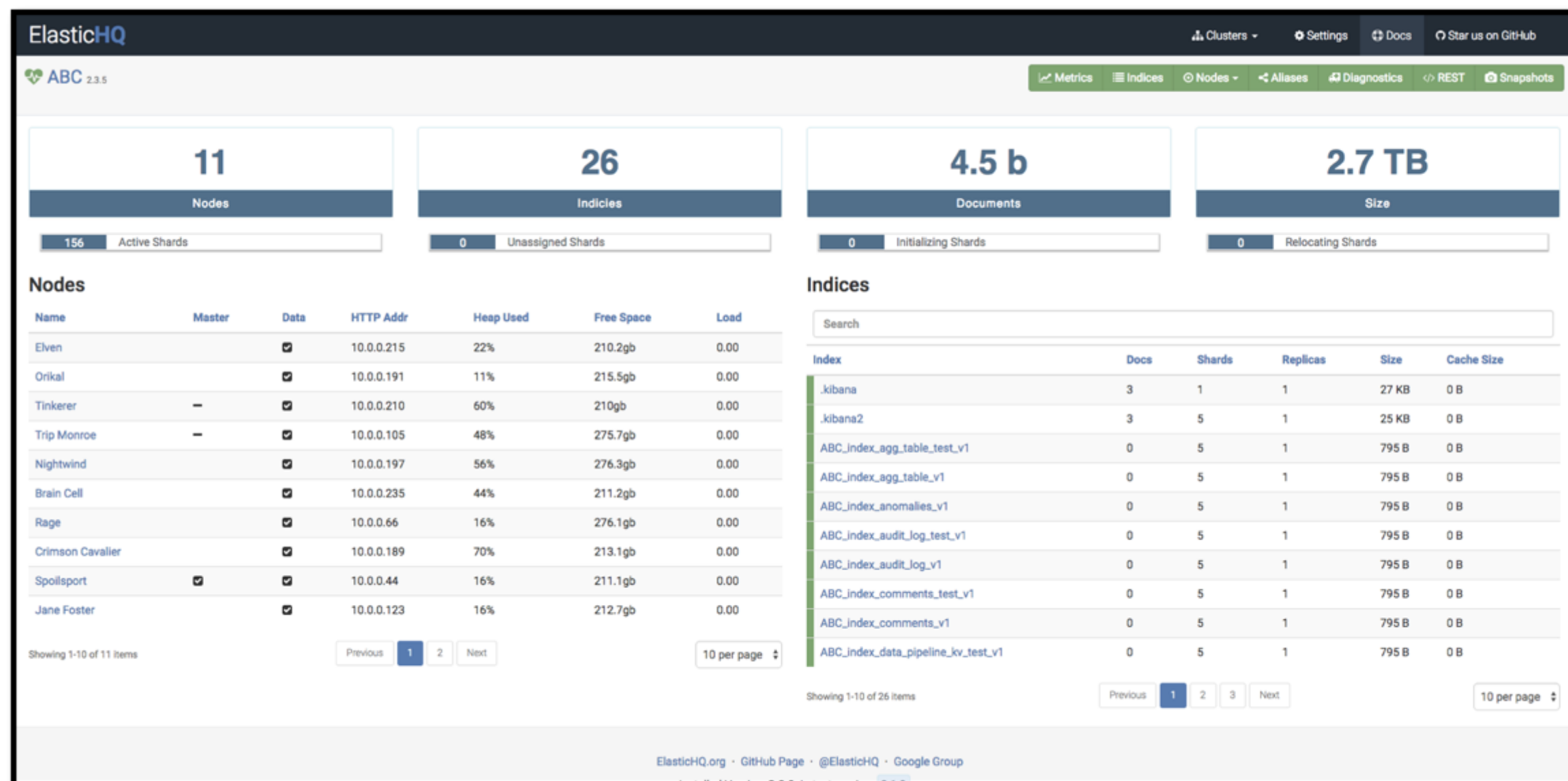
- netdata - 单机
- es_exporter + prometheus + grafana-集群



| 其他：ES集群监控&性能问题排查

性能优化：

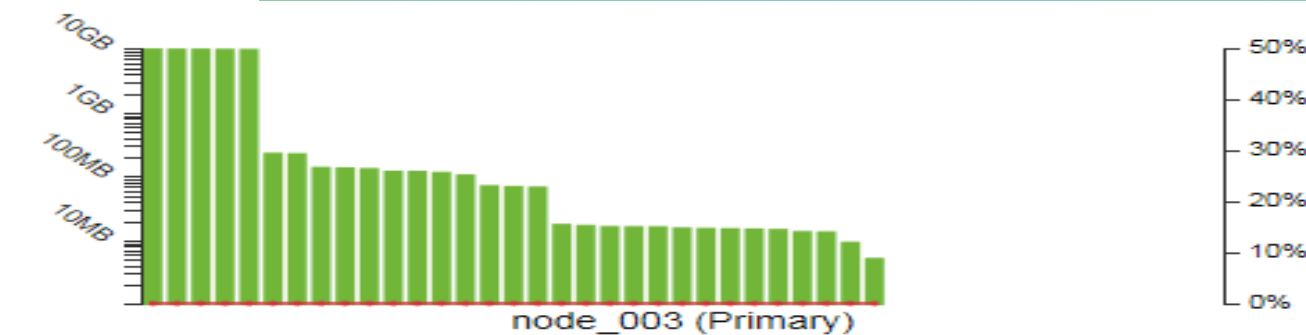
- elastic-hq：性能分析
- cerebro：索引管理、settings
- whatson：segment分析
- es cat api



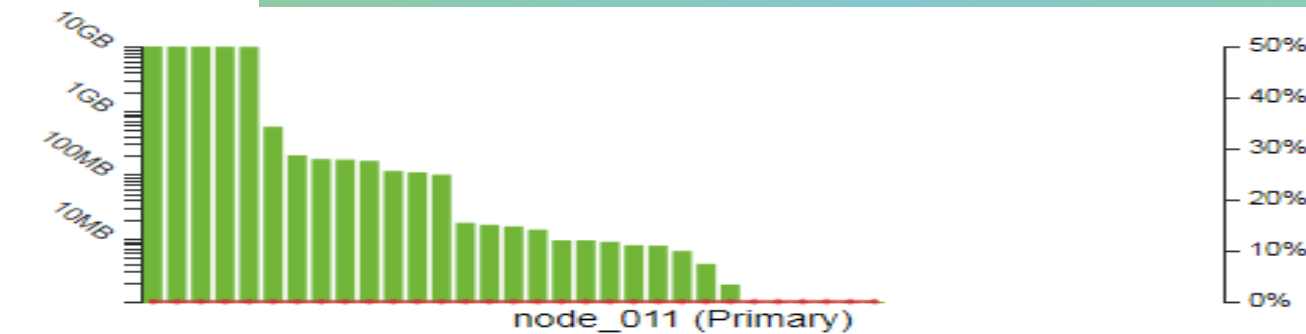
Index:	Shard Status	Primary Size	Total Size	Primary Docs	Deleted
	green	430GB	430GB	532M	0.00%
Shard:	—	—	—	—	—

Segments (Select an Index or Shard Above)

Index: — Shard:0



Index: — Shard:1



Node :

- GC count/GC time
- Query time 查询耗时
- Index time 索引耗时
- Merge time 合并耗时
- Segment count
- Segment memory

Indices :

- Query time 查询耗时
- Index time 索引耗时
- Merge time 合并耗时
- Index writer memory 写入内存
- Segment count
- Segment memory

- Infrastructure UI
- Logs UI

袋鼠云日志产品对以上功能有成熟的解决方案



Infrastructure / Logs

BETA

Q Search for log entries... (e.g. host.name:host-1)

Customize

11/07/2018 11:08:39 AM

Stream live

2018-11-07 11:08:18.000	apache2 107.15.22.242 - GET / HTTP/1.1 200 826	
2018-11-07 11:08:22.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet%2C+consectetur+adipiscing+elit.+B%27erat%27+b%27mi%27+b%27a%27+b%27cum%27+b%27fames%27 HTTP/1.1" 200 255	Wed 07
2018-11-07 11:08:22.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	
2018-11-07 11:08:23.000	REPLCONF ACK 135204	03 AM
2018-11-07 11:08:26.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet.+B%27nunc%27+b%27ve%27+b%27a%27+b%27cum%27+b%27a%27.+B%27eros%27+b%27ut%27+b%27a%27+b%27 HTTP/1.1" 200 255	
2018-11-07 11:08:27.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	06 AM
2018-11-07 11:08:33.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	
2018-11-07 11:08:33.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet.+B%27diam%27+b%27ac%27.+B%27nih%27+b%27mi%27+b%27ac%27+b%27nih%27+b%27platea%27+b%27 HTTP/1.1" 200 255	09 AM
2018-11-07 11:08:39.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet%2C+consectetur.+B%27nisi%27+b%27id%27+b%27ad%27+b%27a%27+b%27vel%27+b%27a%27+b%27a%27 HTTP/1.1" 200 255	
2018-11-07 11:08:39.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	12 PM
2018-11-07 11:08:45.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet%2C+consectetur+adipiscing+elit+b%27nisi%27+b%27mi%27.+B%27erat%27+b%27mi%27+b%27a%27+ HTTP/1.1" 200 255	
2018-11-07 11:08:46.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	03 PM
2018-11-07 11:08:53.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet%2C+consectetur+adipiscing.+B%27nisi%27+b%27et%27.+B%27quam%27+b%27ut%27.+B%27ante%27+ HTTP/1.1" 200 255	
2018-11-07 11:08:53.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	06 PM
2018-11-07 11:09:02.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	
2018-11-07 11:09:02.000	apache2 107.15.22.242 - "GET /guestbook.php?cmd=set&key=messages&value>Lorem+ipsum+dolor+sit+amet%2C+consectetur+adipiscing+elit+b%27diam%27+b%27mi%27.+B%27orci%27+b%27eu%27+b%27a%27+ HTTP/1.1" 200 255	
2018-11-07 11:09:11.000	apache2 107.15.22.242 - "GET / HTTP/1.1" 200 826	09 PM

No additional entries found Load again

CONTENT

01 ES集群概况

02 ES集群优化



03 云日志平台的应用

云日志 是基于日志数据的运维分析产品



数据采集

全栈式日志数据及范日志数据采集汇总管理，不需要登录服务器即可批量部署 Agent 并进行配置管理。



搜索和分析

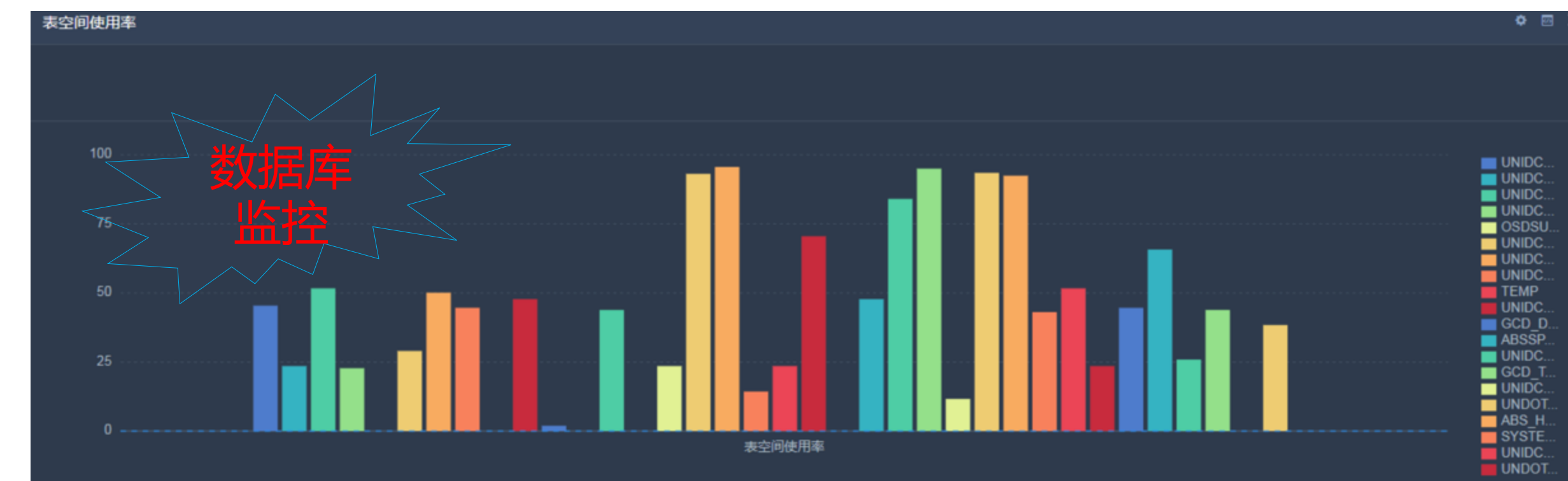
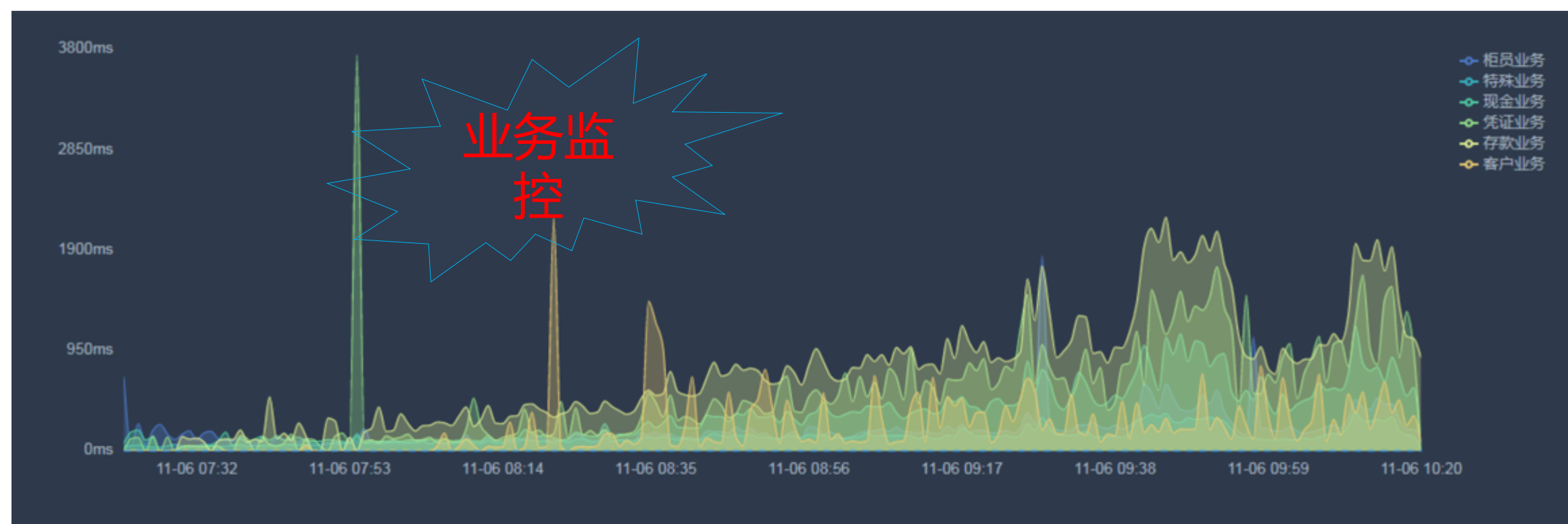
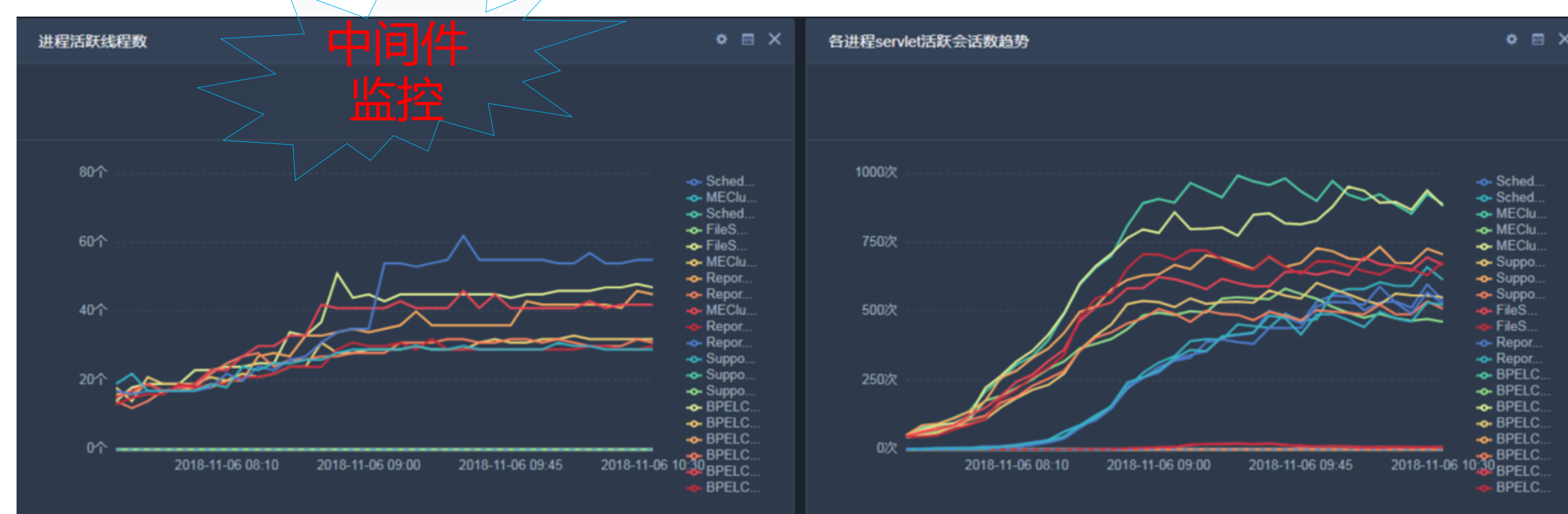
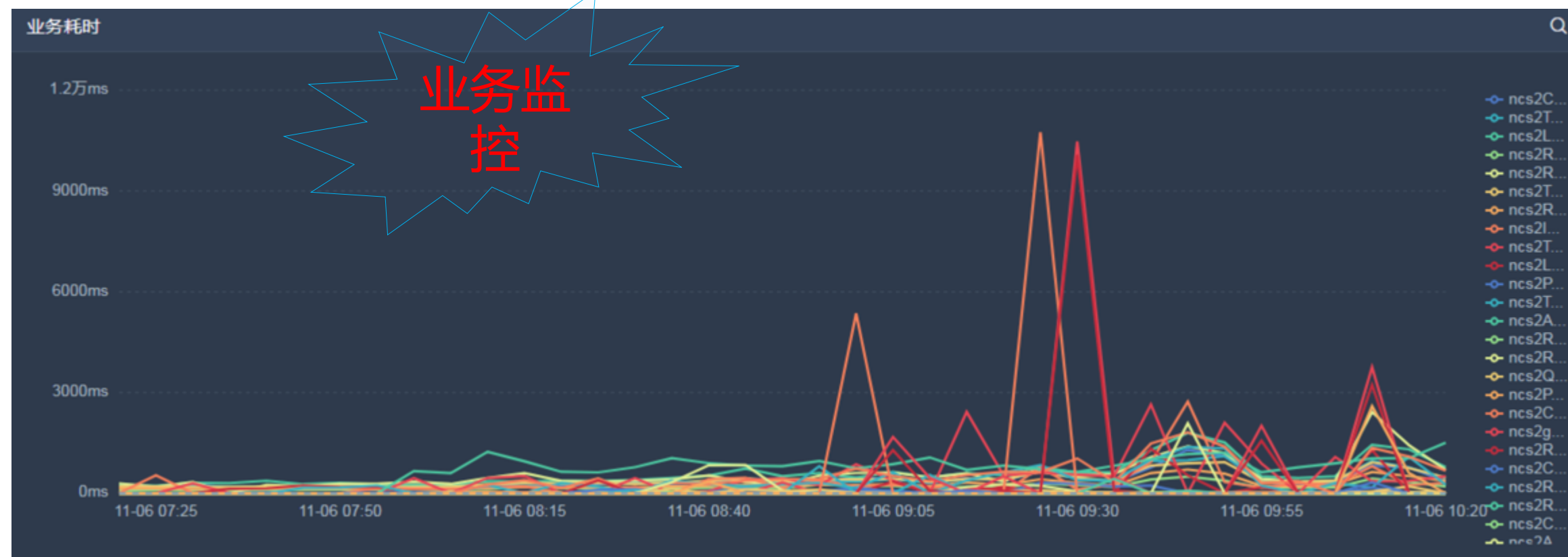
海量数据秒级搜索，并对大量数据源进行格式化分析整合解析，可视化配置解析规则。



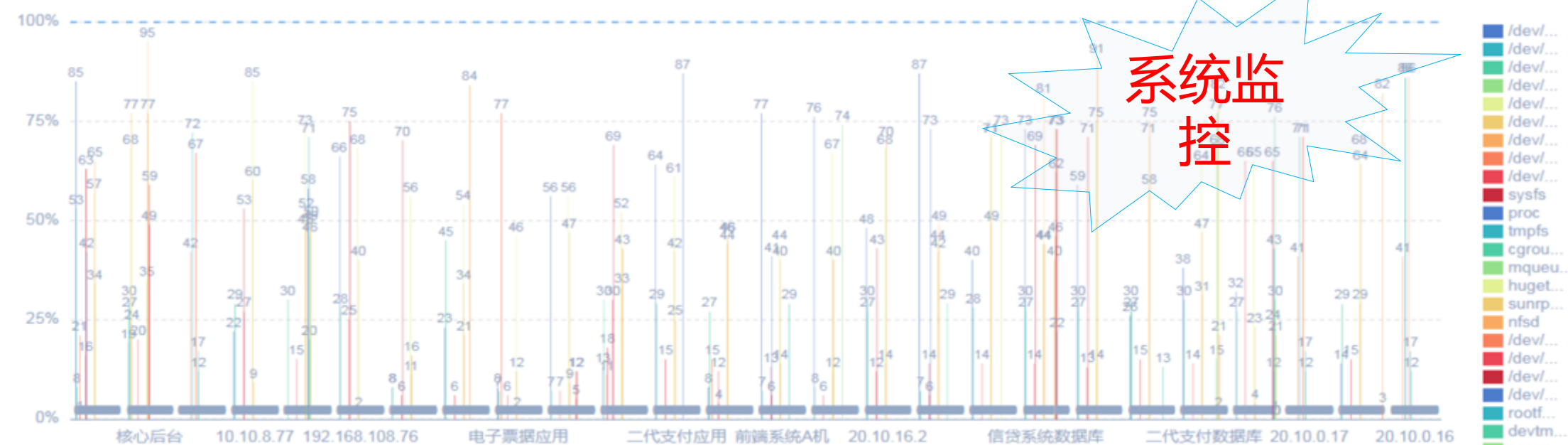
数据可视化

日志数据展示，可通过自定义设置业务仪表盘展示数据，生动直观展示机器情况和业务情况。

云日志平台的应用



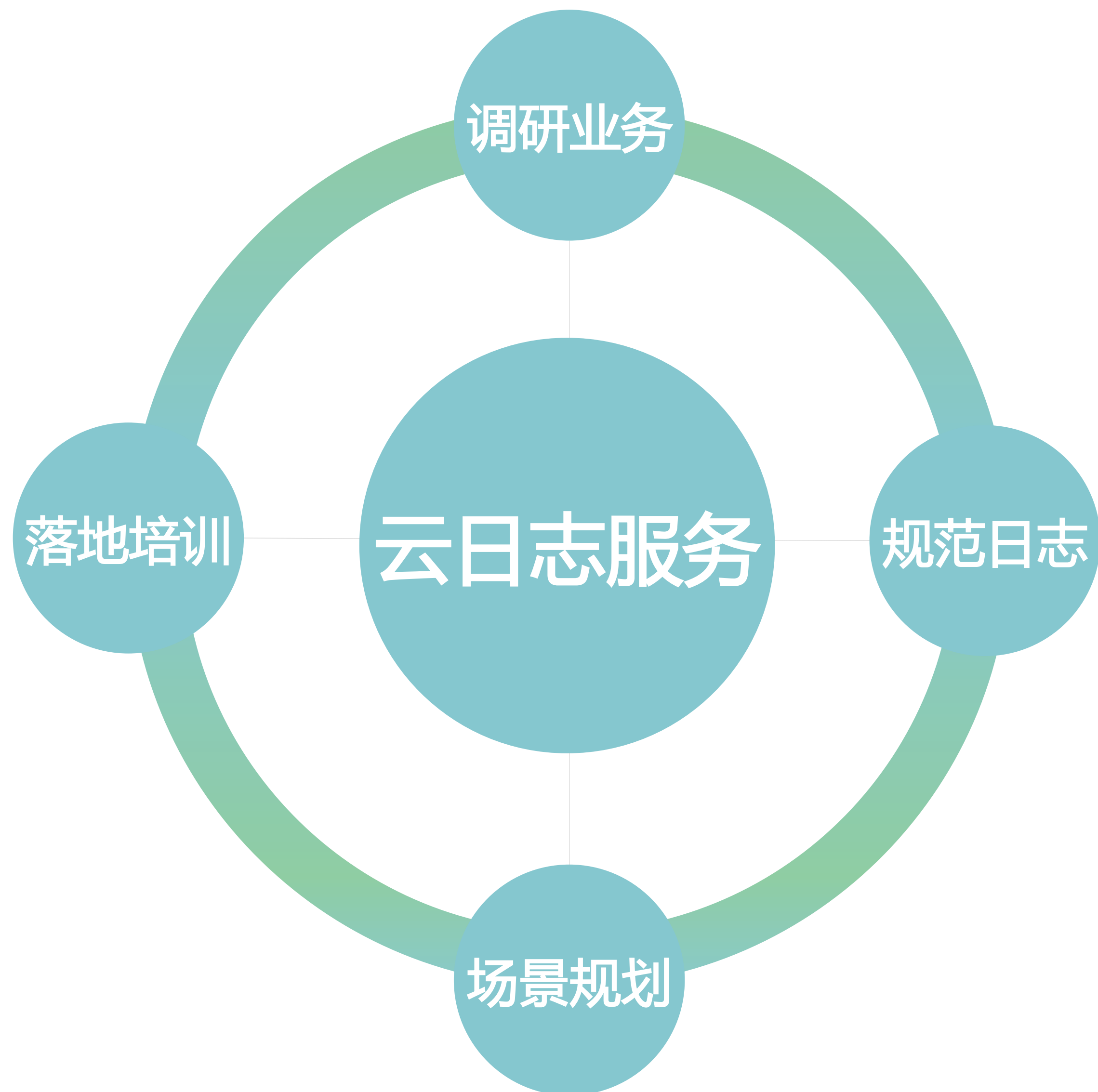
主机性能指标



SaaS版云日志使用场景图
扫码注册体验SaaS版本云日志

<https://account.dtstack.com/#/login>





调研业务

深耕业务模式，了解具体操作

规范日志

对需求日志进行字段提取和规范化

场景规划

对于日志进行场景的规划

落地培训

对企业人员进行培训和落地实施，让方案得以实现

公司简介



袋鼠云 | 数据智能，让未来变成现在

袋鼠云成立于2016年，是国内领先的数据智能践行者。自创立以来，袋鼠云始终坚持“让数据产生价值”的理念，将数据智能的先进理念和技术传播和应用到传统行业中，在“一切业务数据化”的基础上，通过帮助挖掘客户的数据价值，实现“一切数据业务化”，帮助提升生产效率，促进产业创新和社会



1亿

一年半内累计融资1个亿，入选2018年杭州“准独角兽”榜单

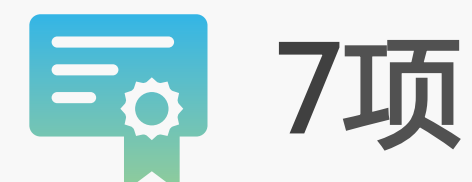
4地



总部坐落于人间天堂杭州，在北京、上海、贵州均设有分公司或办事处



客户选择和袋鼠云数智同行



7项

作为阿里云战略合作伙伴，阿里云生态技术先锋，拥有7项阿里云合作伙伴认证，超过10位阿里云全球MVP

80%



公司总人数超过200人，其中80%为技术人员，每年数千万研发投入

使命：让数据产生价值

愿景：做最领先的数据智能践行者

价值观：客户第一、团队合作、专业、担当

THANKS

电话：400 002 1024 网址：www.dtstack.com



微 信 公 众 号



专业、垂直、纯粹的 Elastic 开源技术交流社区
<https://elasticsearch.cn/>