

Elasticsearch的大杂烩分享

分享者——赖鸿智

欢聚时代



- 1 elasticsearch&本人介绍
- 2 elasticsearch在欢聚时代的使用
 - 2.1 搜索服务架构介绍
 - 2.2 实时大数据查询引擎介绍
- 3 遇到的一些坑
- 4 优化技巧
- 5 工具推荐

因前公司需要对分布式搜索选型，2011年9月开始接触**elasticsearch**，当时还是0.16版本，作为对比的有**solr**，**solandra**，**solrcloud**，**lily**等。

目前任欢聚时代搜索工程师，负责欢聚时代搜索服务的搭建及维护。

后陆续写了一系列**es**方面的博客

csdn专栏地址：<http://blog.csdn.net/column/details/elasticsearch.html>

很多过时的博文，不看也罢



Elasticsearch是个开源分布式搜索引擎
它的特点有：

- 近实时索引
- 高级分析聚合查询
- 分布式
- 零配置
- 自动发现
- 索引自动分片
- 索引副本机制
- restful风格接口
- 多租户
- schema free
- 自动搜索负载等



elasticsearch在数据库中的排名



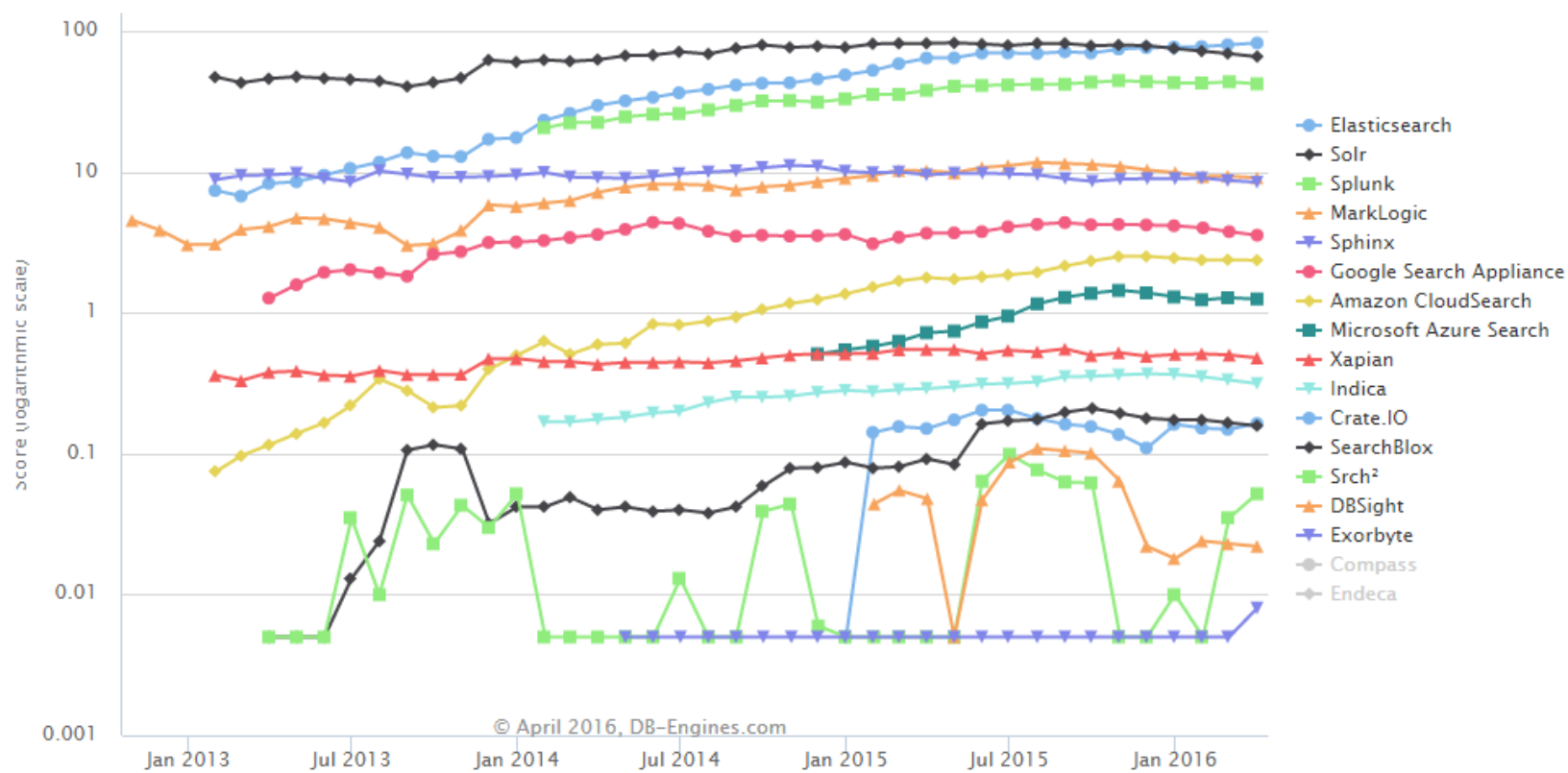
303 systems in ranking, April 2016

Rank			DBMS	Database Model	Score		
Apr 2016	Mar 2016	Apr 2015			Apr 2016	Mar 2016	Apr 2015
1.	1.	1.	Oracle	Relational DBMS	1467.53	-4.48	+21.40
2.	2.	2.	MySQL +	Relational DBMS	1370.11	+22.39	+85.53
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1135.05	-1.45	-14.07
4.	4.	4.	MongoDB +	Document store	312.44	+7.11	+33.85
5.	5.	5.	PostgreSQL	Relational DBMS	303.73	+4.10	+35.41
6.	6.	6.	DB2	Relational DBMS	184.08	-3.85	-13.56
7.	7.	7.	Microsoft Access	Relational DBMS	131.97	-3.06	-10.22
8.	8.	8.	Cassandra +	Wide column store	129.67	-0.66	+24.78
9.	9.	↑ 10.	Redis +	Key-value store	111.24	+5.02	+16.69
10.	10.	↓ 9.	SQLite	Relational DBMS	107.96	+2.19	+5.67
11.	11.	↑ 14.	Elasticsearch +	Search engine	82.58	+2.41	+17.92
12.	12.	↓ 11.	SAP Adaptive Server	Relational DBMS	73.32	-3.33	-13.37
13.	13.	13.	Teradata	Relational DBMS	72.26	-1.81	+2.00
14.	14.	↓ 12.	Solr	Search engine	66.02	-3.35	-15.98
15.	15.	15.	HBase	Wide column store	51.49	-0.92	-9.65
16.	16.	↑ 17.	Hive	Relational DBMS	49.08	-1.43	+6.33
17.	17.	↓ 16.	FileMaker	Relational DBMS	46.10	-1.83	-5.72
18.	18.	18.	Splunk	Search engine	42.35	-1.38	+4.32
19.	19.	↑ 21.	SAP HANA +	Relational DBMS	40.35	+0.36	+7.01
20.	20.	↑ 22.	Neo4j +	Graph DBMS	31.91	-0.44	+3.50

elasticsearch在搜索引擎中的排名



DB-Engines Ranking of Search Engines



那些国外企业在用elasticsearch



Who is Using Us



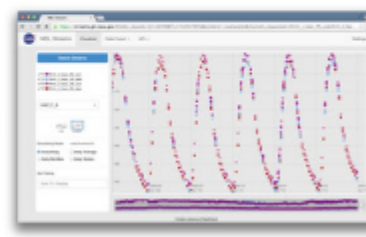
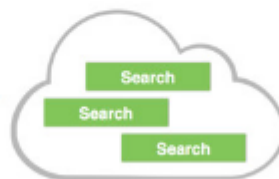
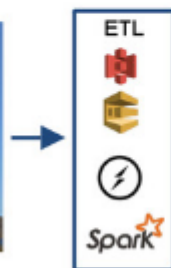
NASA

Data Pipeline and Indexing

FLIX

mozi

FIC



sen



XING

GitHub



LOGMATIC.IO

GitHub你们最常用的功能是什么？

"Search is at the core of
GitHub"

Tim Pease
an Operations Engineer at
GitHub

400万用户

800万repositories

20亿代码文件

300请求一分钟

26 datanode 8 client node

routing on repository ID

那些国内企业在用elasticsearch



百度 400+机器 700+节点

新浪 50+机器

携程 30个数据节点

京东

百姓网 20+机器

芒果TV

日志易

欢聚时代

数说故事

。。。。还有很多！！！！

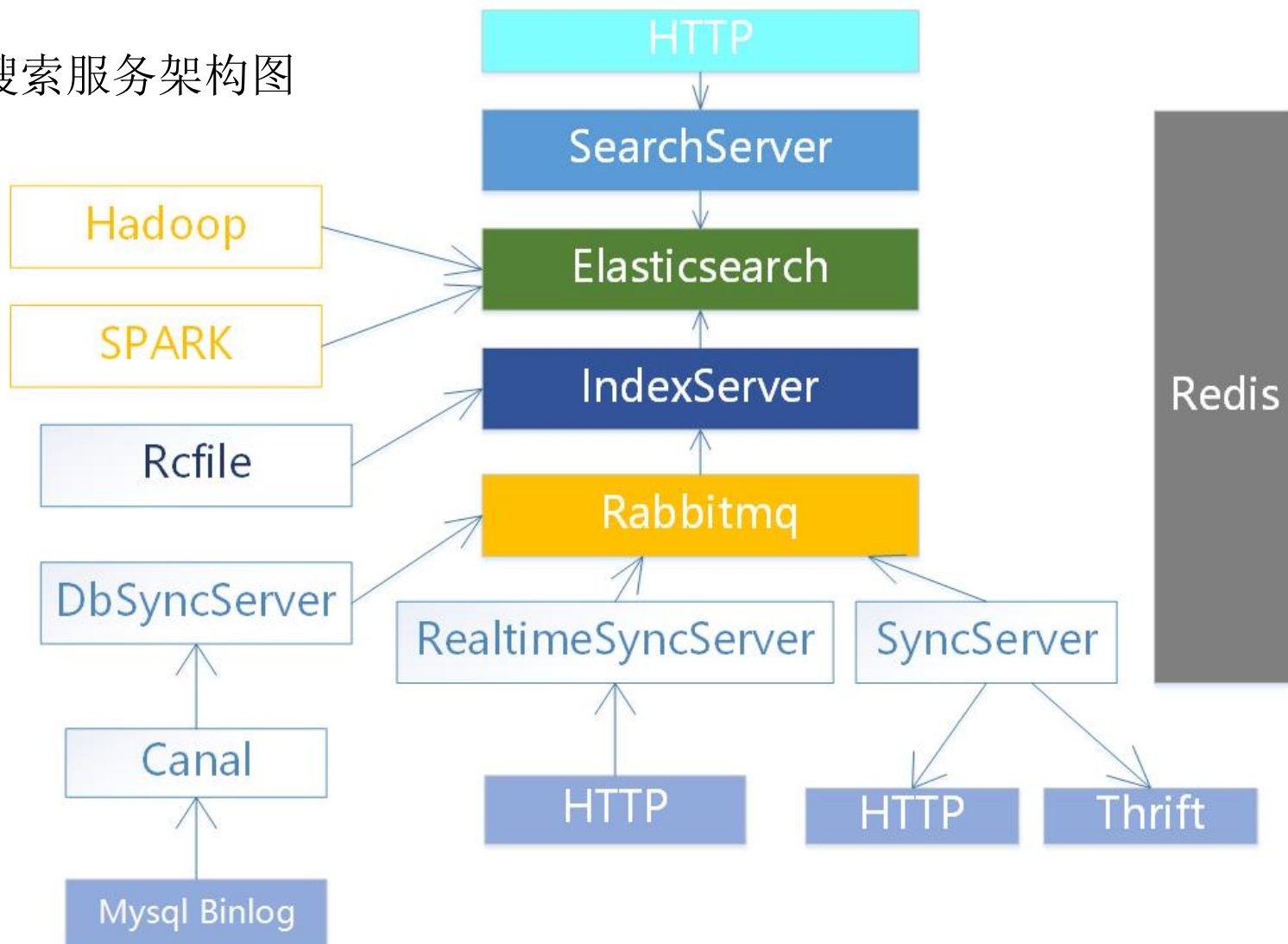
支撑公司大大小小二十多个业务的搜索

Solr to Elasticsearch

欢聚时代搜索服务介绍



搜索服务架构图



20节点，双机房互备

30+个索引

200+个分片

10亿+条 100g数据

峰值1k请求一秒，3w更新一秒

实时大数据分析服务介绍



业务方只想更快、更快、更快

目标：

实现千亿级数据秒级返回

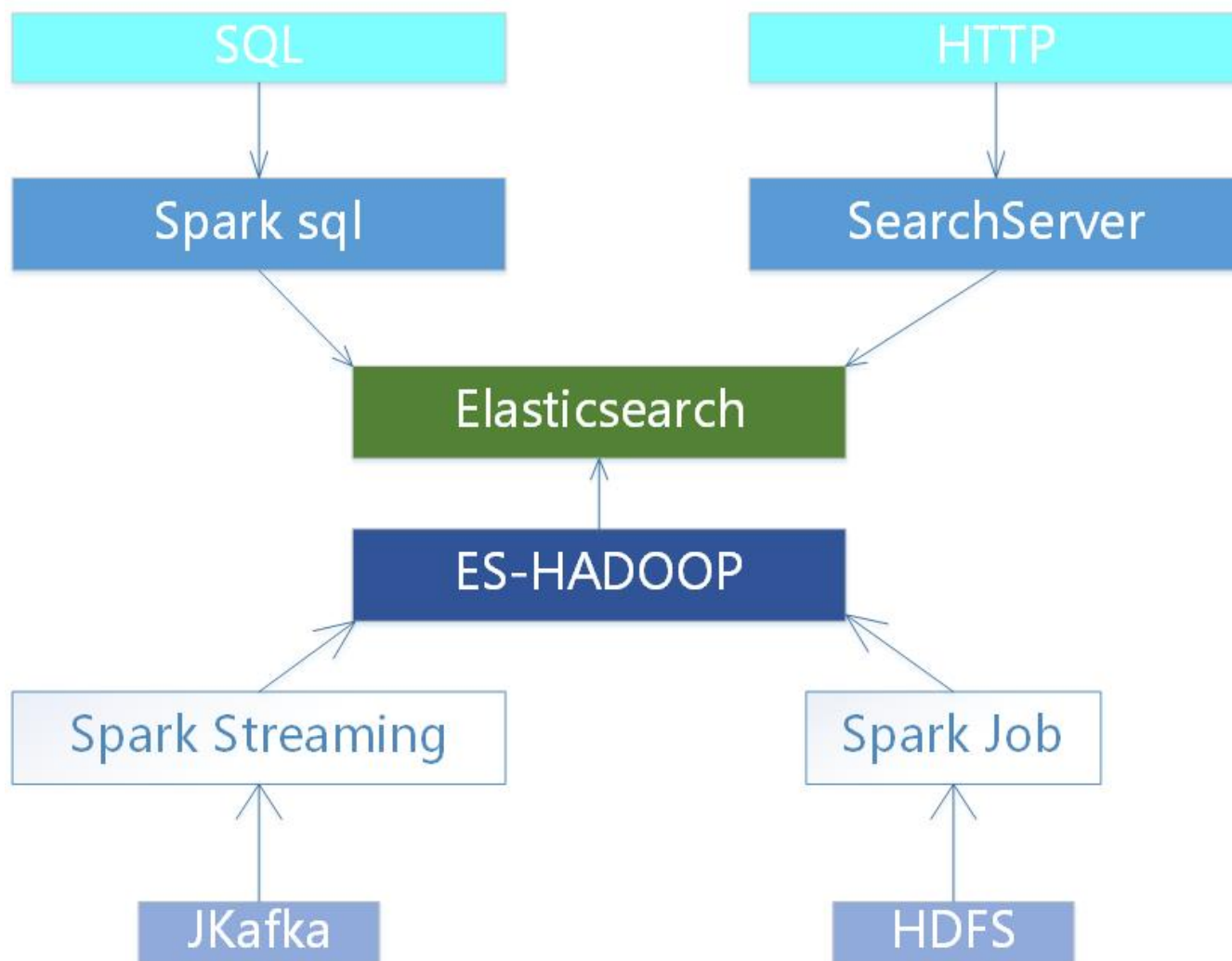
实现：

spark+es，es作为数据存储，通过spark实现es没法提供的多表join功能

实时大数据分析服务介绍



实时大数据分析服务架构图



27节点

600+索引

10000+分片

900亿+数据

8w/s 写入

elasticsearch一些使用经验，以及优化方法

merge速率自动化

- `index.merge.policy.type`
- `index.merge.policy.min_merge_size`
- `index.merge.policy.max_merge_size`
- `index.merge.policy.merge_factor`
- `index.merge.policy.max_merge_docs`
- `index.merge.policy.calibrate_size_by_deletes`
- `index.merge.policy.min_merge_docs`
- `index.merge.policy.max_merge_docs`

`indices.store.throttle.type`

`indices.store.throttle.max_bytes_per_sec`

`index.store.throttle.type`

`index.store.throttle.max_bytes_per_sec`

多数据文件目录支持

默认使用doc_value属性

支持lucene5的BEST_COMPRESSION特性

filter接口合并到query

ELK版本统一

Ingest node

Lucene6



踩的坑之1.7.3 升 2.x.x 注意问题



1.一定要在升级前进行兼容性测试

一定要在升级前进行兼容性测试

一定要在升级前进行兼容性测试

插件地址: <https://github.com/elastic/elasticsearch-migration/>

不然有可能出现mapping冲突导致无法启动es

2.如果索引过大, 并且使用多个磁盘的话, 要预留升级过程中索引文件复制的时间。

踩的坑之升级 2.x后索引速度下降问题



升级完后发现单表的导入速度从**2.5w/s**，降低到几百一秒，查了下文档发现**es2.x**的**translog**的写入方式变成了**fsync**，每次索引请求都要写入到磁盘，改成**async**默认是5秒刷一次，可以通过**index.translog.sync_interval**参数调整间隔。

```
curl -XPOST http://host:port/_template/  
template_defalut -d '{  
  "template": "*",  
  "order": 2,  
  "settings": {  
    "index.translog.durability": "async",  
    "index.translog.sync_interval": "60s"  
  }  
}'
```



踩的坑之节点重启后分片恢复缓慢



中BUG，2.3后修复了

踩的坑之gc时间过长导致节点离开



调大ping时间

优化gc设置，新手慎用

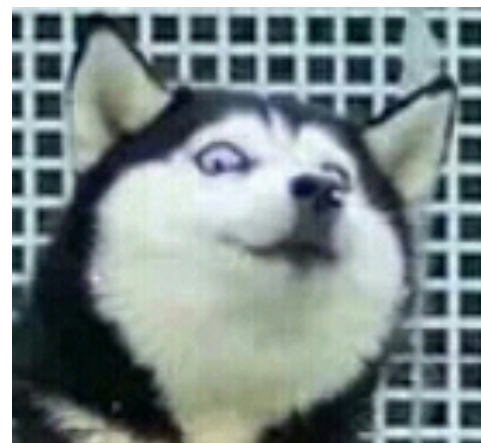
绝招G1 GC



Elasticsearch索引速度优化



index.refresh_interval :-1
index.number_of_shards : X
index.number_of_replicas : 0
index.translog.sync_interval : 30s
index.translog.durability : "async"
index.translog.flush_threshold_size: 4g
index.translog.flush_threshold_ops: 50000



去掉_all字段可节省一半空间



开启索引压缩可节省空间，但会有10%-20%的性能损耗

不需分词的字符串字段设成not_analyzed

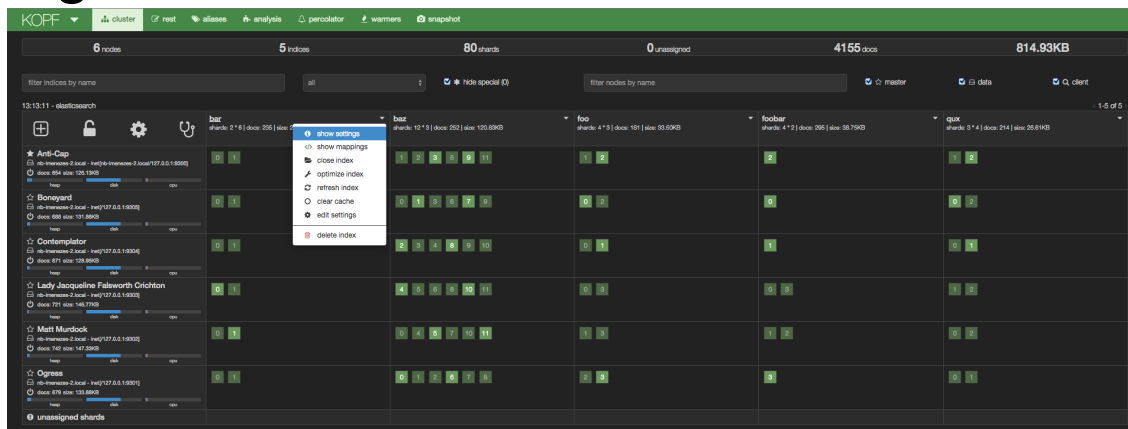
具体需求具体优化，没有万能法则！！！！

工欲善其事必先利其器——工具推荐

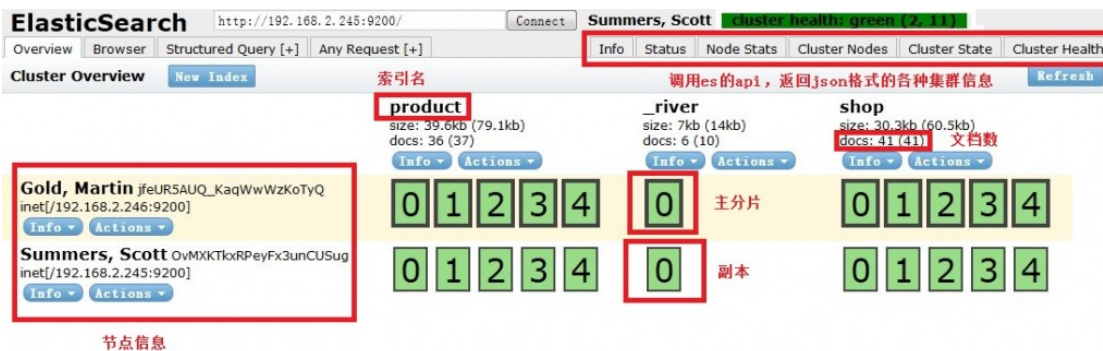


管理

kopf <https://github.com/lmenezes/elasticsearch-kopf>



head <https://github.com/mobz/elasticsearch-head>



工欲善其事必先利其器——工具推荐



监控

marvel <https://www.elastic.co/products/marvel>



bigdesk <https://github.com/lukas-vlcek/bigdesk>

ganglia https://github.com/ganglia/gmond_python_modules

总结



跳坑真言

es大法好，lucene保平安，码在写，天在看，心有疑惑看官网，遇到八哥问论坛，赶紧用es保平安，新手进群，诚心诚念唯一大神好，林总监答疑解惑普众生，es大法保平安，众生都是为es来，现世险恶忘前缘，es弟子说真相，教你脱困莫拒绝。

首选官方论坛，大神云集为你排忧解难

<https://discuss.elastic.co/>

中文社区适合英文小白

<http://elasticsearch.cn/>

Q群

Elastic技术交流群1: 190605846

Elastic技术交流群2: 211682609

个性化推荐攻城狮



谢谢 & QA