ElasticSearch Training

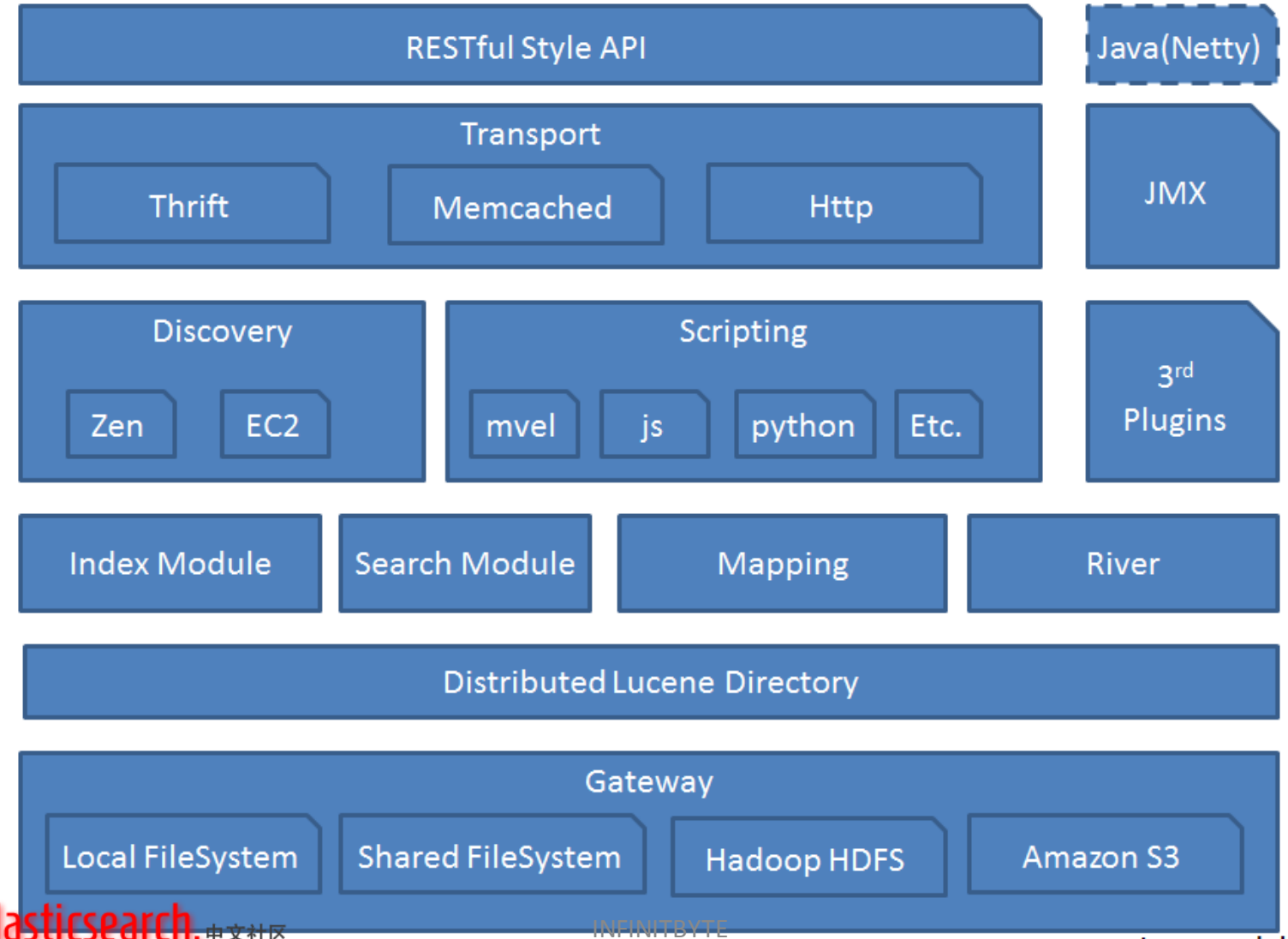
Advanced Concepts

Medcl,2013.1.20

你将学到什么？

- 模块介绍
- 设计理念
- 流程剖析
- 各种调优
- 监控

整体架构


RESTful Style API
Java(Netty)
Transport
Thrift
Memcached
Http
JMX
Discovery
Zen
EC2
Scripting
mvel
js
python
Etc.
3rd Plugins
Index Module
Search Module
Mapping
River
Distributed Lucene Directory
Gateway
Local FileSystem
Shared FileSystem
Hadoop HDFS
Amazon S3

Transport

- 传输层、可扩展
  - Native Java\Groovy API (by elasticsearch team)
  - Http API (by elasticsearch team)
  - Servlet transport (by elasticsearch team)
  - Memcached transport plugin (by elasticsearch team)
  - Thrift Transport (by elasticsearch team)
  - ZeroMQ transport layer plugin (by Tanguy Leroux)
  - Jetty HTTP transport plugin (by Sonian Inc.)
  - WebSocket transport plugin (by jprante)
  - etc.

Gateway

- 允许部分集群故障
- 允许整个集群故障
- Apple's TimeMachine

Gateway:

Local,本地分布式存储，推荐
Shared FS，共享集中式文件存储
Hadoop，分布式文件系统存储
S3，网络云文件系统存储

```
# gateway.type: local
# gateway.recover_after_nodes: 1
# gateway.recover_after_time: 5m
# gateway.expected_nodes: 2
```

http://log.medcl.net/item/2010/09/translation-search-engine-and-the-time-machine/

索引存储及持久化

- 为什么需要持久化到Gateway？
  - 节点（集群）重启之后的索引数据恢复
- Gateway与WorkDir
  - Gateway存储完整的索引信息
  - WorkDir对外提供相应查询操作
  - WorkDir可以是内存、本地文件系统或两者结合
  - Gateway可以是本地文件系统、共享文件系统或HDFS等云存储
- WorkDir被假设是不安全的运行环境，数据允许随意丢失
- Gateway被假设是可靠的，持久化的数据存储

Discovery

- 节点自动发现
- Zen Discovery
  - MultiCast
  - Unicast
- EC2（ plugin cloud-aws is required）

- Zen 用来做节点自动发现和master选举，master用来处理节点的加入和退出，以及shard的重新分配，注意master不是单点的，当前master挂了之后，其它节点自动选举产生新的master。
- 节点不需要每次请求都通知master，所以没有任何单点故障的瓶颈
- 只有当节点“准备就绪”的时候，该节点才会被通知可被使用。(即等待该节点完全初始化完成）

Scripting

- scripting模块用来允许用户通过自定义脚本来进行评分计算，从而影响最终的搜索结果。

- 支持多种脚本语言：
  - mvel, js, groovy, python, and native java
- 脚本可以存放在相应目录实现预加载
  - config/scripts/group1/group2/test.py
- 或直接写在查询里面（inline）

River

- 什么是River？
  - River是一个运行在elasticsearch集群内部的可插拔的服务，主要用来从外部pull数据，然后往elasticsearch里面创建索引。

- CouchDB River Plugin (by elasticsearch team)
- Wikipedia River Plugin (by elasticsearch team)
- Twitter River Plugin (by elasticsearch team)
- RabbitMQ River Plugin (by elasticsearch team)
- RSS River Plugin (by David Pilato)
- MongoDB River Plugin (by Richard Louapre)
- Open Archives Initiative (OAI) River Plugin (by Jörg Prante)
- St9 River Plugin (by Sunny Gleason)
- Sofa River Plugin (by adamlofts)
- Amazon SQS River Plugin (by Alex B)
- JDBC River Plugin (by Jörg Prante)
- FileSystem River Plugin (by David Pilato)
- LDAP River Plugin (by Tanguy Leroux)
- Dropbox River Plugin (by David Pilato)
- ActiveMQ River Plugin (by Dominik Dorn)

Mapping

- Mapping 是定义搜索引擎如何处理一个索引文档的过程，包括文档的搜索特征，比如那些字段可被搜索，是否切词，如何切等等,一个索引下面能存储不同 “mapping types” 的索引文档.

集群

- 节点类型
  - IndexNode：既提供读也提供写
  - DataNode：只提供数据存储和访问（负载均衡)
- 节点之间是对等关系，去中心化
- Master节点（弱化）
  - 只不过多了维护集群状态
  - 每个节点上面的集群状态数据都是实时同步的
  - 挂掉master，没有任何问题，任意一台自动顶上
  - #cluster.name: elasticsearch
  - #discovery.zen.minimum_master_nodes: 1
    - split brain

Node

- # node.name: "Franz Kafka"
- # node.rack: rack314
- “workhorse”
  - # node.master: false
  - # node.data: true
- “coordinator”
  - # node.master: true
  - # node.data: false
- “search load balancer”
  - # node.master: false
  - # node.data: false

分布式索引目录

- ES实现了一个分布式索引目录
- 索引级别的灵活配置
  - 每个索引可拆分为多个分片（Shard）
  - 每个分片可以拥有0个或多个副本
  - 索引级别的灵活配置
- 任何一个节点都能提供索引和查询操作
- 任何一个分片或其副本都可进行查询、搜索操作（一个完整功能的lucene索引）

Transaction log

- Indexed / deleted doc is fully persistent
- No need for a Lucene IndexWriter#commit
- Managed using a transaction log / WAL
- Full single node durability (kill dash 9)
- Utilized when doing hot relocation of shards
- Periodically “flushed” (calling IW#commit)

Partitioning

- 基于文档的分区（Document Partitioning）
  - Each shard has a subset of the documents
  - A shard is a fully functional “index”
- 基于词条的分区（Term Partitioning）
  - Shards has subset of terms for all docs

Partitioning - Term Based

- 优点:
  - 包含K个Term的查询，需要参与的Shard数量为K
- 优点: 对包含K个Term的查询条件，所需要的磁盘访问的复杂度为O(K)
- 缺点:
  - 网络流量高，数据量大
  - 每台机器的term都需要统一拿到一个地方来
- 缺点:
  - 无法获取单篇文档的完整信息（ per doc information ）
  - 以至于很难实现(facets / sorting / custom scoring)

Partitioning - Term Based

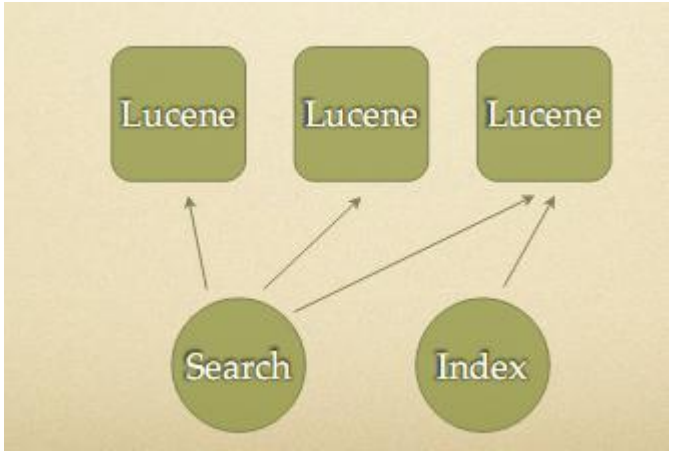
- Riak Search - Utilizing its distributed key-
- value storage
- Lucandra (abandoned, replaced by Solandra)
  - Custom IndexReader and IndexWriter to
  - work on top of Cassandra
  - Very very “chatty” when doing a search
  - Does not work well with other Lucene
  - constructs, like FieldCache (by doc info)

Partitioning - Document Based

- 优点:
  - 每个shard能够独立的处理查询请求
- 优点: 很方便的为每个文档添加信息
  - 很方便的实现(facets, sorting, custom scoring)
- 优点:很少的网络开销
- 缺点： 每个shard都需要处理查询请求
- 缺点: shard为N，处理包含K个term的查询，磁盘访问的复杂度为O(K*N)

Distributed Lucene Doc Partitioning

- Shard Lucene into several instances
- Index a document to one Lucene shard
- Distribute search across Lucene shards



分布式演示

- In a Nutshell -
Distributed

索引流程

- 索引提交到节点
- 节点上面有shard group
- 什么是shard group？
- 什么是primary shard？
- hash并定位到primary shard，如果该节点上没有primary shard，则跳转到有primary shard的节点
- primary shard处理索引请求，分发replica请求到replica group，默认同步执行
- 写完translog，索引操作返回
- real get，是在未refresh前，直接从buffer和translog里面读取数据
- refresh之后，索引数据可见，可被搜索

查询流程

- get 和索引不一样，不需要在primary shard上执行
- hash
- 然后shard的replica group，挑选任意shard，来获取数据，可通过设置preference参数来进行控制
- 查询，是在任意节点上面做scatter和gather的过程，查询类型searchtype
  - Query And Fetch
  - Query Then Fetch
  - Dfs, Query And Fetch
  - Dfs, Query Then Fetch
  - Count
  - Scan
- 允许部分shard请求失败

除了跑起来，还得跑得快

调优

如何调优

- Zabbix
- iostatus
- netstat
- iotop
- sar
- htop
- application profiler
- query log analyzer
- … …

工具不重要，重要的是思想

收集，观察，可视化，
定位问题，解决问题

Rebalancing

- 什么情况下会发生
  - 集群故障恢复
    - 节点挂掉
  - 副本分配
    - 动态调整副本数
  - 索引动态均衡
    - 新增机器
    - 挂掉机器

相关参数配置：
http://www.elasticsearch.org/guide/reference/modules/cluster.html

服务器优化

– JAVA环境
– * - nofile 20480，调整文件打开数
– swap off，关闭swap
– * - memlock unlimited，调整memlock
– ulimit -n 204800，调整每个进程可打开文件数

– vi /etc/fstab ，关闭磁盘文件访问时间

/dev/*sdb* /var/elasticsearch ext3 noatime,nodiratime 0 0

JVM配置

vi elasticsearch\bin\service\elasticsearch.conf

set.default.ES_HEAP_SIZE=1024

HEAP_SIZE设置为物理内存的60%左右，
其余剩下内存留给操作系统做文件系统分页缓存等

Never Swaps
bootstrap.mlockall:true

`ulimit -l unlimited`

vi /etc/security/limits.conf
* - memlock unlimited

GC优化

- 淘汰JDK6，使用JDK7、8
- 合理设置HEAP大小
  - 太小，频繁GC，OOM
  - 太大，delay，攒个大的，回收压力大，时间长，内存占用高，可能造成swapping
- 默认HEAP使用率达到75%触发GC
- 提高吞吐还是提高速度，tradeoff
- 控制IndexMerge压力
  - #index.merge.policy.segments_per_tier:10
- 记录并分析GC日志
- http://jprante.github.com/2012/11/28/ElasticSearch-Java-Virtual-Machine-settings-explained.html

集群优化

- 经常遇到的问题
  - 集群恢复太慢
  - 集群恢复时，数据需要重新平衡
    - # discovery.zen.minimum_master_nodes: 1
    - # discovery.zen.ping.timeout: 3s
    - # cluster.routing.allocation.node_initial_primaries_recoveries: 4
    - # cluster.routing.allocation.node_concurrent_recoveries: 2
    - # indices.recovery.max_size_per_sec: 0
    - # indices.recovery.concurrent_streams: 5
  - 每个节点上面的shard数不均衡
  - 有些节点上面的shard都是热门数据，而有些刚好相反

集群优化

- 节点设置：
  - node.tag: tag1
- 控制集群的shard存放位置
  - index.routing.allocation.include.tag
  - index.routing.allocation.exclude.tag
- 通过节点ip来控制shard存放位置
  - cluster.routing.allocation.include._ip
  - cluster.routing.allocation.exclude._ip
- 然后在索引或者集群范围内应用设置
  - 实时动态修改，动态生效

http://www.elasticsearch.org/guide/reference/index-modules/allocation.html

Shard Allocation

- Cluster级别的设置

```
curl –XPUT localhost:9200/_cluster/setting –d’
{
   "persist":{
      "cluster.routing.allocation.exclude._ip":
"192.168.1.1"
   }
}’
```

- Index级别的设置

```
curl –XPUT localhost:9200/medcl/ -d’ {
   "index.routing.allocation.include.tag":"node1,node2"
}’
```

自定义属性

elasticsearch.yml配置

```
node.group1: group1_value1
node.group2: group2_value4
```

使用自定义参数

```
curl -XPUT localhost:9200/test/_settings -d '{
"index.routing.allocation.include.group1" : "xxx"
"index.routing.allocation.include.group2" : "yyy",
"index.routing.allocation.exclude.group3" : "zzz",
}'
```

每节点shard数

- 能够设置每个节点上面最多承载的shard数量
- index级别，分别设置，实时设置，实时生效

```
curl  -XPUT localhost:9200/medcl –d’
{
   "index.routing.allocation.total_shards_per_node": 2
}’
```

索引优化

- 影响索引速度的因素
  - shard数量
  - 节点数量
  - 集群同步操作
  - 索引操作
    - 合并、优化
    - 索引写操作，每个lucene目录每次只有一个写操作
  - 磁盘io
    - io次数及速度
  - translog和data目录放ssd

索引优化

- client端减少频繁建立连接
  - 使用tcp长连接，而非http
  - 多线程
  - 连接池
- client减少请求次数，合并索引操作
  - 使用bulk接口
- 尽量减少索引大小，索引前预处理、过滤等
- 合理规划mapping，合理使用分词，减少索引量
- store、_source合理启用，减少索引量
- mapping、analyzer合理化设置

索引优化

- 批量索引时，ES相关优化
  - 关闭_refresh或调大刷新时间，默认1s
  - 各节点同时接收索引
  - 索引replica设置为0
  - 索引segments个数,optimize来合并
  - translog参数调整
    - index.translog.flush_threshold_ops ,默认5000
  - merge参数调整：
    - index.merge.policy.merge_factor，默认10

查询优化

- 影响查询时间的因素
  - 服务器硬件环境
  - 索引量
  - 索引shard数量
  - 索引shard里面小文件数目
  - 索引存放位置
  - 查询条件
  - 是否有缓存

查询优化

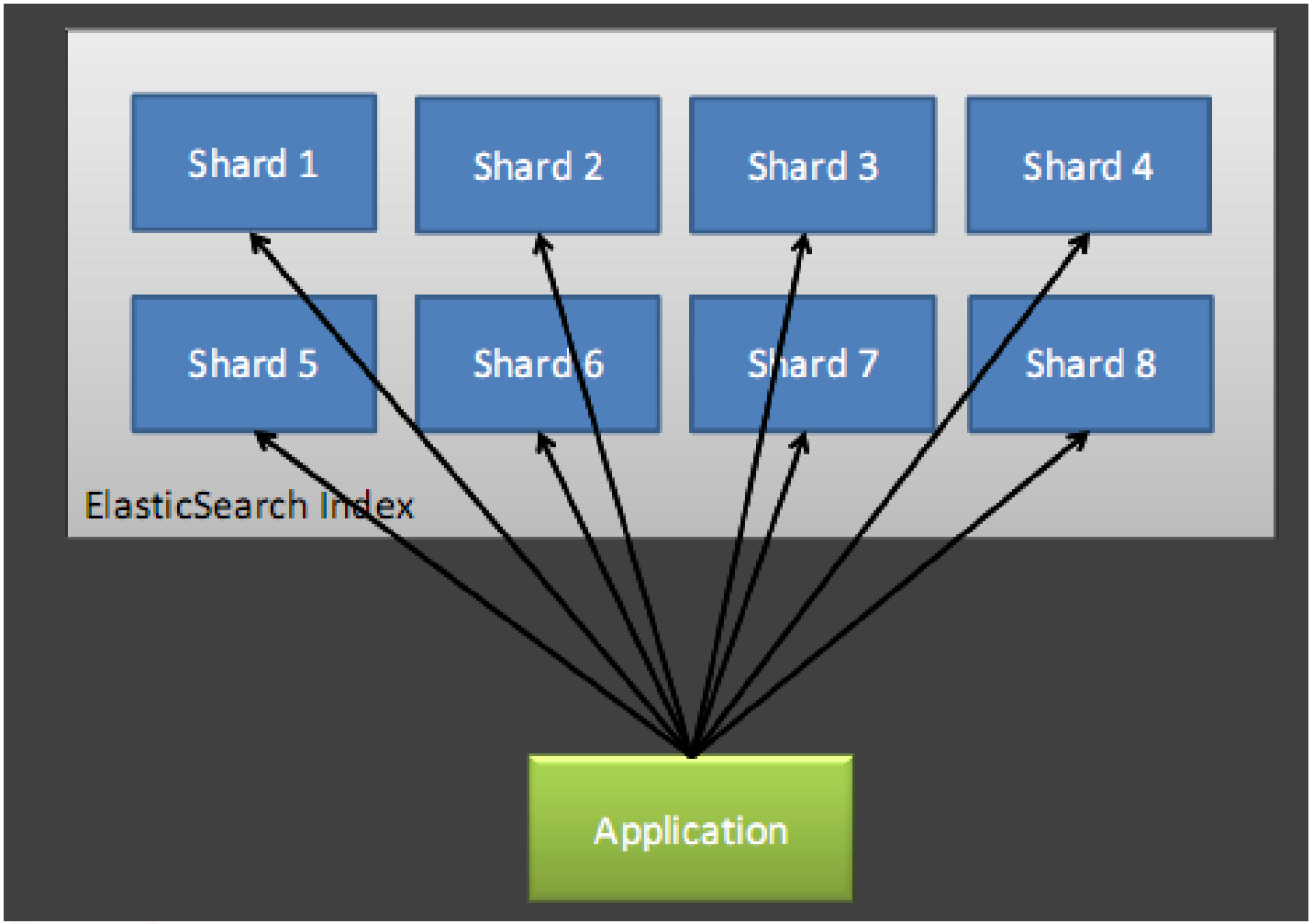
- 查询条件设计优化
  - 尽量使用Filter，合理设置Cache大小
  - 字段分词规则设计
- 执行Optimize接口优化索引库
  - 合并Segments
- 合理规划Index和Shard
  - 使用Routing
  - 按照数据特征划分索引

Shard&Routing

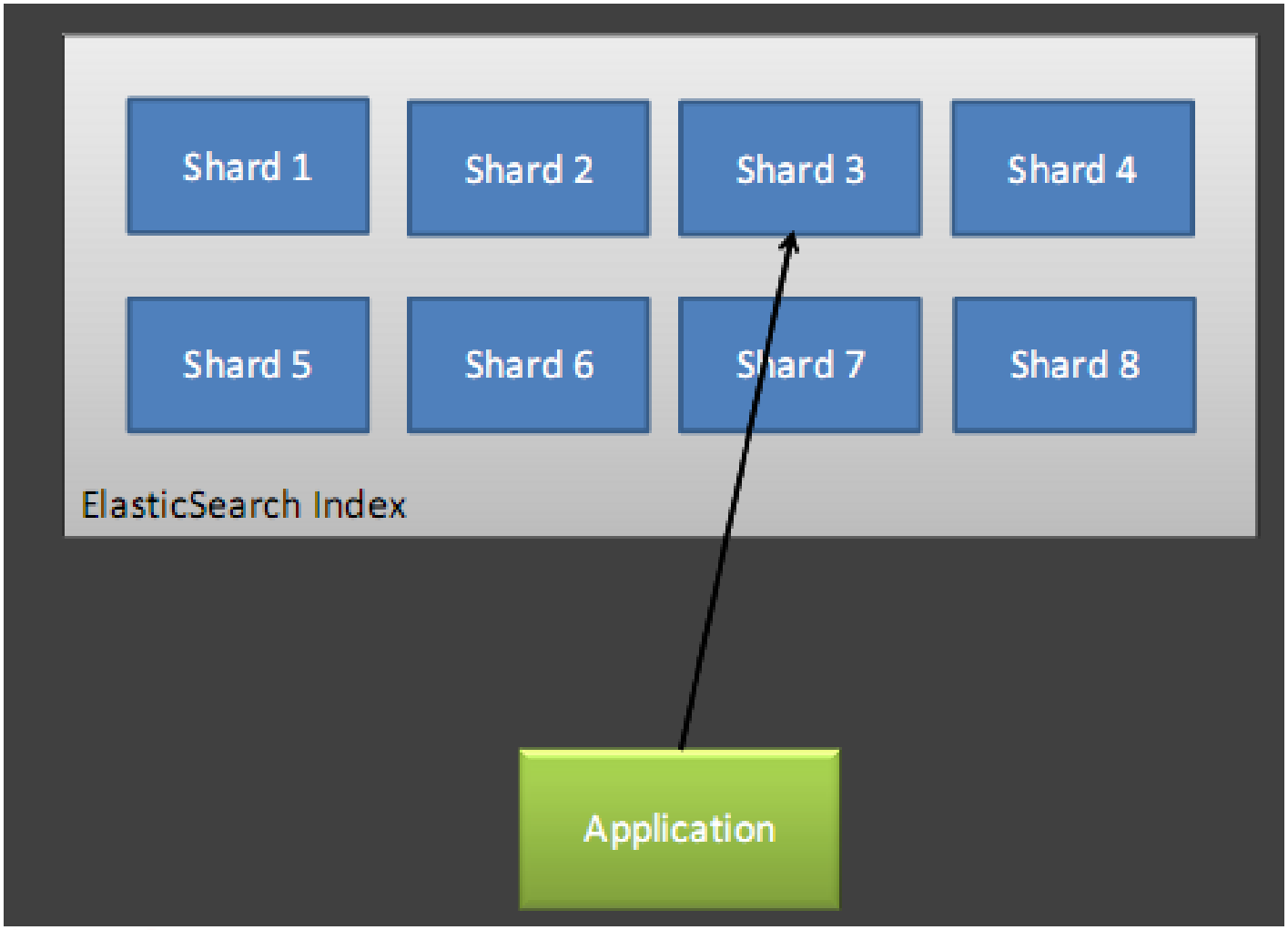
- 索引和查询的路由策略
- 默认情况下
  - type+id：哈希取模
  - 保证数据
- 使用routi
  - 数据存放
  - 查询的时
  - 从四处撒网到特定的放大

```
{
  "comment" : {
    "_routing" : {
      "required" : true,
      "path" : "blog.post_id"
    }
  }
}
```

No Routing


Shard 1
Shard 2
Shard 3
Shard 4
Shard 5
Shard 6
Shard 7
Shard 8
ElasticSearch Index
Application

With Routing



查询优化

- Shard容量有上限
  - 根据具体的服务器及相关配置找到单个shard的瓶颈
- 索引大小同样有上限
  - shards*max_shard_size
- Replica同样有上限，当replica超过阀值，只是多占一份磁盘空间而已
- shard数目要看具体的数据和使用场景，没有公式
  - 硬件、文档数、文档大小、查询（是否使用sort、facet等）

查询优化

- wildcard
- sorting
- faceting
- query or filter
- analyzer
- warmup

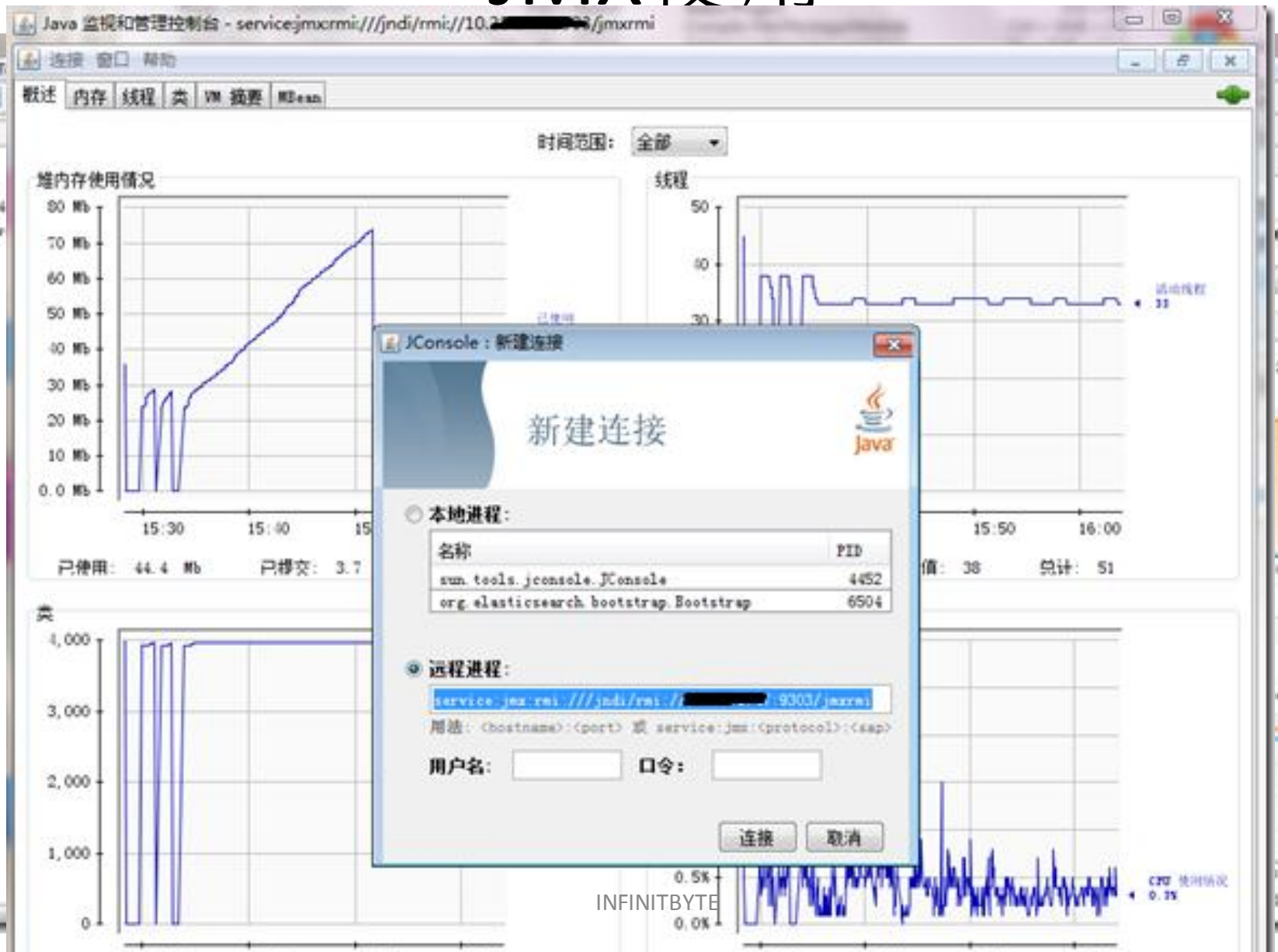
查询优化

- 在满足查询功能的情况下
  - 选择合理的查询类型
    - 掌握各个查询的特性和使用场景
  - 尝试调整查询参数
  - 分别记录并比较查询时间
  - 每次只修改一个参数
  - 查询尽量模拟真实场景

Cache

- Filter Cache
  - 缓存filters查询结果集
  - 类型
    - Node Filter Cache，节点全局共享，All Shard，LRU
      - indices.cache.filter.size：20%，或配置为具体容量，如512mb
    - Index Filter Cache
      - index级别，3种类型：resident、soft 、weak
      - index.cache.filter.max_size：-1
      - index.cache.filter.expire：5m
- Field Data Cache
  - 当字段做sorting 或者faceting，加载字段的值开销比较大
  - 2种类型：resident 、soft
  - index.cache.field.max_size：-1
  - index.cache.field.expire：5m
- Cache监控，Bigdesk
- http://www.elasticsearch.org/guide/reference/index-modules/cache.html

JMX使用


Java 监视和管理控制台 - service:jmx:rmi:///jndi/rmi://10.[redacted]/jmxrmi
连接 窗口 帮助
概述 内存 线程 类 VM 摘要 MBean
时间范围: 全部
堆内存使用情况
80 Mb
70 Mb
60 Mb
50 Mb
40 Mb
30 Mb
20 Mb
10 Mb
0.0 Mb
15:30
15:40
15:50
16:00
已使用: 44.4 Mb
已提交: 3.7
线程
50
40
30
活动线程 33
值: 38
总计: 51
类
4,000
3,000
2,000
1,000
0
0.5%
0.0%
CPU 使用情况 0.3%
JConsole : 新建连接
新建连接
Java
本地进程:
名称 PID
sun.tools.jconsole.JConsole 4452
org.elasticsearch.bootstrap.Bootstrap 6504
远程进程:
service:jmx:rmi:///jndi/rmi://[redacted]:9303/jmxrmi
用法: <hostname>:<port> 或 service:jmx:<protocol>:<sap>
用户名:
口令:
连接
取消

查看集群健康状态

- http://localhost:9200/_cluster/health


```
{
   "cluster_name": "elasticsearch",
   "status": "yellow",
   "timed_out": false,
   "number_of_nodes": 1,
   "number_of_data_nodes": 1,
   "active_primary_shards": 30,
   "active_shards": 30,
   "relocating_shards": 0,
   "initializing_shards": 0,
   "unassigned_shards": 30
```

其它接口

- Health
- State
- Update Settings
- Nodes Info
- Nodes Stats
- Nodes Shutdown
- Nodes Hot Threads

常用管理API

- 索引开关闭、只读
  - curl -XPOST 'localhost:9200/my_index/_close‘
  - curl -XPOST 'localhost:9200/my_index/_open'
- 优化、refresh
  - curl -XPOST 'http://localhost:9200/my_index/_optimize‘
  - curl -XPOST 'http://localhost:9200/my_index/_refresh'
- 调整replica数
  - curl -XPUT: http://localhost:9200/myindex/_settings

    { "index" : { "numberOfReplicas" :2 } }
- clear cache
  - curl -XPOST 'http://localhost:9200/twitter/_cache/clear'

elasticsearch | ElasticSearc ×

s.medcl.net/?query=elasticsearch&tab_name=1

elasticsearch.

elasticsearch | slideshare | weibo | other

elasticsearch

排序：时间↓ 相关性

找到相关结果约28369个,花费时间71ms

google groups (27661) elasticsearch.org (281) elasticsearch.cn (266) blog (89) slideshare.net (54) tutorial (13) web (2) translate (1) plugin (1) news (1)

Large Scale Performance Monitoring for ElasticSearch, HBase, Solr, SenseiDB, etc.

Desipio of how Semaex SPM Pefomae Moioig sevie is bui ad how i woks. Oigiay peseed a Bei Buzzwods 2012....

http://www.slideshare.net/sematext/otis-gospodnetic-spmbbuzz... 2012-06-12 07:08:20

Real-Time Analytics Webinar

Big Daa is hagig he game fo ageies ookig o ake hei soia media ehoogies ad usome isigh paies o he ex eve. Bu, maagig he massive veoiy, vou me, ad vaiey of soia media ad ohedaa ses a sae a be a huge haege. Ifo himps has bui he ages ope makepae of daa ses i wod. We've ow opeed up ou pafom o ageies ike yous.This webas oves how ageies a buid hei ow Big Daa pafom - eabig hem o go fom daa soues o seig isighs - i a fa io of he ime expeed, ad a a faio of he os.We wok wih some of he wod's op digia, adveisig, ad PR ageies, whih use he Ifohimps pafom o boade ad sae hei popieay daa offeigs hough- Seime ad Ifuee Aaysis- Cie Cusome Isighs- Rea-Time Soia Media Aayis- Ifogaphi ad Repo Geeaio- Topi ad Meme Takig - Web Taffi Aaysis- Pesoaized Campaigs- Mobie ad Coss-Chae Repoig...

http://www.slideshare.net/infochimps/realtime-analytics-webi... 2012-06-12 07:08:20

LibreCat::Catmandu

你不是一个人在学习，
ES交流QQ群：190605846(1群已满) 2116826
群欢迎）。

结果聚合

ElasticSearch Server (26)
ElasticSearch Setup (22)
ElasticSearch Cpu Usage (13)
ElasticSearch Connections (12)
ElasticSearch Expectable Performances (12)
ElasticSearch with JSONML (11)
ElasticSearch Multisort (9)
ElasticSearch OutOfMemory Exceptions (8)
Solr vs ElasticSearch (5)
ElasticSearch Administration Questions (4)
ElasticSearch Architecture Diagram (3)

最近更新

Re: has_parent query and routing (2013-01-19 07:24:49)
Re: Understanding Threadpools (2013-01-19 03:05:10)
Re: Text Search with more preference to prefix matches (2013-01-18 22:58:14)
Re: Short Questions (2013-01-18 22:36:02)
Re: Problem creating elasticsearch client unde

Recent Discussions on Ela ×

es-bbs.medcl.net

ElasticSearch讨论板

控制面板 所有话题 动态 站内信 medcl 忽略新消息 退出

ES国内开发者第一次线下交流筹备中，欢迎献策献力。。。 QQ群：211682609

所有话题 我的话题 23 我的草稿 3 等待回答的问题 16

新插件发布：elasticsearch-analysis-string2int，如果你需要聚合很多facet，可以考虑一下

已回答 7条回复 medcl 最近回复 一月 15 经验分享

线下交流活动《ElasticSearch China Conference#1 》

1条回复 由 medcl 发起 一月 15 最新动态

routing怎么用

已解决√ 3条回复 1 条新消息 yaofaye 最近回复 一月 15 API使用

Java Client调用ElasticSearch搜索代码示例

已解决√ 5条回复 1 条新消息 yaofaye 最近回复 一月 15 API使用

请教不同网段的机器如何加入同一集群

提问 2条回复 2 新复数 asoqa 最近回复 一月 14 API使用

如何提高增加建立索引的速度？增加share的话，会提升添加文档的速度么？

提问 1条回复 1 条新消息 由 gemini 发起 一月 11 经验分享

master节点重新选举的问题

发起新话题

版块设置

所有话题

经验分享

最新动态

安装

API使用

插件使用

资料收集

文档翻译

Lucene相关

灌水区

站务相关

热门标签

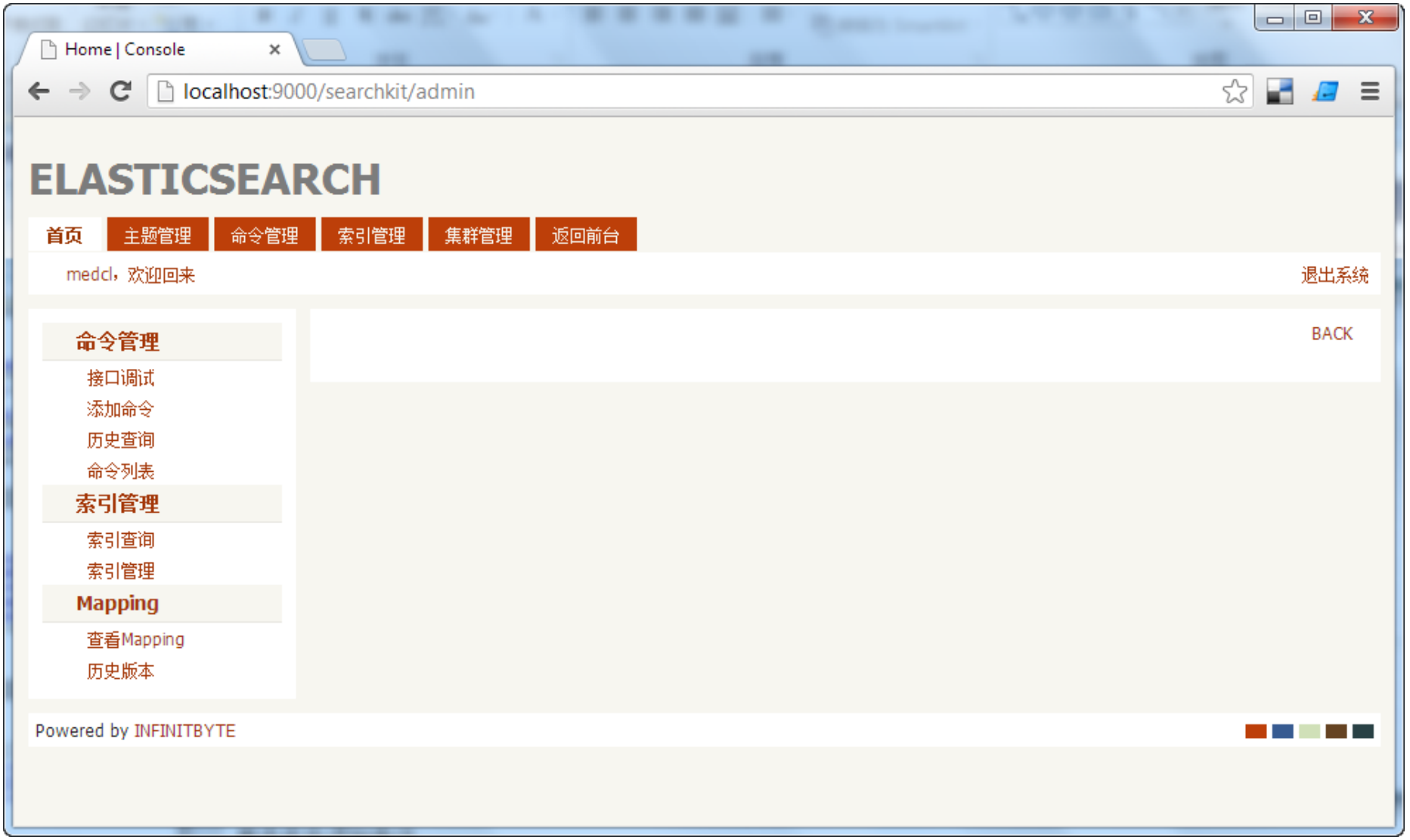
elasticsearch.cn

jdbc-river

介绍

ppt

Coming Soon~


Home | Console
localhost:9000/searchkit/admin
ELASTICSEARCH
首页
主题管理
命令管理
索引管理
集群管理
返回前台
medcl，欢迎回来
退出系统
命令管理
接口调试
添加命令
历史查询
命令列表
索引管理
索引查询
索引管理
Mapping
查看Mapping
历史版本
BACK
Powered by INFINITBYTE

一些工具

- elasticsearch-head: A web front end for an elastic search cluster.
- bigdesk: Live charts and statistics for elasticsearch cluster.
- paramedic: Live charts with cluster stats and indices/shards information.
- SPM for ElasticSearch: Performance monitoring with live charts showing cluster and node stats, integrated alerts, email reports, etc.

相关链接

- http://www.elasticsearch.org/ 项目网站
- https://github.com/elasticsearch/ 源码
- http://s.medcl.net/ 资源聚合
- http://elasticsearch.cn/ 中文站点
- http://es-bbs.medcl.net/ 中文论坛
- QQ群：211682609

Q/A

Thank you !


elasticsearch.中文社区


ESCC
ElasticSearch China Conference #1


INFINITBYTE