AI 人工智能系列

Elasticsearch

大数据搜索引擎 Search

◎ 罗 刚 编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Elasticsearch

人工智能系列

Elasticsearch大数据搜索引擎

罗　刚　编著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

Elasticsearch 搜索集群系统在生产和生活中发挥着越来越重要的作用。本书介绍了 Elasticsearch 的使用、原理、系统优化与扩展应用。本书用例子说明了 Java、Python、Scala 和 PHP 的编程 API，其中在 Java 搜索界面实现上，介绍了使用 Spring 实现微服务开发。为了扩展 Elasticsearch 的功能，本书以中文分词和英文文本分析为例介绍了插件开发方法。本书介绍了使用 Elasticsearch 作为数据管理平台的日志监控与分析方法，介绍了使用 OCR 从图像中提取文本以及问答式搜索的开发方法。

本书适用于有程序设计基础的开发人员或者对 IT 运维技术感兴趣的从业人员。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Elasticsearch 大数据搜索引擎 / 罗刚编著. —北京：电子工业出版社，2018.1
（人工智能系列）
ISBN 978-7-121-33233-3

Ⅰ. ①E… Ⅱ. ①罗… Ⅲ. ①搜索引擎—程序设计 Ⅳ. ①TP391.3

中国版本图书馆 CIP 数据核字（2017）第 306154 号

策划编辑：张 迪
责任编辑：底 波
印 刷：三河市华成印务有限公司
装 订：三河市华成印务有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：14.25 字数：364.8 千字
版 次：2018 年 1 月第 1 版
印 次：2018 年 1 月第 1 次印刷
定 价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254469，zhangdi@phei.com.cn。

前言

<<<<< PREFACE

智慧生物的大规模协作造就了惊人的进化奇迹。大规模机器集群造就机器系统进化成为强大的智能系统。Elasticsearch 作为大数据与搜索引擎技术的结合体，随着社会对大规模开源分布式搜索引擎的需求迅速成长。

由于其良好的易用性，Elasticsearch 早在 1.0 版本之前就加速了大规模搜索集群的普及。本书从基本概念开始熟悉 Elasticsearch，接下来介绍了 Elasticsearch 在 Windows 和 Linux 操作系统下的安装。除了 CURL 命令，本书还介绍了使用常用的编程工具和 Elasticsearch 搜索服务交互，包括 Java、Python、Scala 和 PHP，以及使用 SQL 语句查询 Elasticsearch 索引的方法。自然语言文本理解往往以插件形式存在于 Elasticsearch 集群中，第 2 章介绍了如何开发与测试插件。因为 Elasticsearch 经常用于实时搜索或分析，所以性能优化很重要，第 3 章介绍了如何管理 Elasticsearch 集群。为了更合理地使用和扩展 Elasticsearch，第 4 章简单分析了 github 中托管的 Elasticsearch 源代码。对于搜索引擎来说，返回结果的相关性是一个重要的话题，第 5 章讨论了这个问题。第 6 章介绍了使用 Java 开发搜索引擎 Web 用户界面的几种方法。

随着人工智能领域技术的发展，让搜索引擎智能加速变成现实。智能搜索引擎需要能够检测到并识别出图像中的文字，第 7 章介绍了结合 OpenCV 使用 Tesseract 识别文字的方法。第 8 章介绍了根据问题返回搜索结果的问答式搜索。

目前 Elasticsearch 是实时系统监控的首选，第 9 章介绍了使用 Elasticsearch 监控与分析日志，也介绍了通过物联网监控系统的方案。

本书相关的参考软件和代码在读者 QQ 群 471033528 的附件中可以找到。Elasticsearch 及其底层依赖的软件，其复杂程度已经超越了一个人所能掌握的程度。一些具体的细节也可以在读者 QQ 群中讨论。感谢早期合著者、合作伙伴、员工、学员、读者的支持，给我们提供了良好的工作基础。就像玻璃容器中的水培植物一样，这是一个持久可用的工作基础。技术的融合与创新无止境，欢迎读者一起探索。

本书适合需要具体实现搜索引擎的程序员使用，对于信息检索等相关领域的研究人员也有一定的参考价值，同时猎兔搜索技术团队已经开发出以本书为基础的专门培训课程和商业软件。

参与本书编写的还有张子宪、崔智杰、张晓斐、石天盈、张继红、张进威、刘宇、何淑琴、任通通、高丹丹、徐友峰、孙宽，在此一并表示感谢。

试读结束：需要全本请在线购买： www.ertongbook.com

目录

使用 Elasticsearch

在信息时代，可供获取的数据加速涌现，我们可以通过搜索引擎来挖掘大数据的价值，百度就是一个大的数据搜索引擎。

Lucene 是一个 Java 语言开发的开源全文检索引擎工具包。Lucene 穿了一件 json 的外衣，就是 Elasticsearch。Elasticsearch 内置了对分布式集群和分布式索引的管理，所以相对 Solr 来说，更容易分布式部署。使用 Elasticsearch 的搜索系统整体架构如图 1-1 所示。

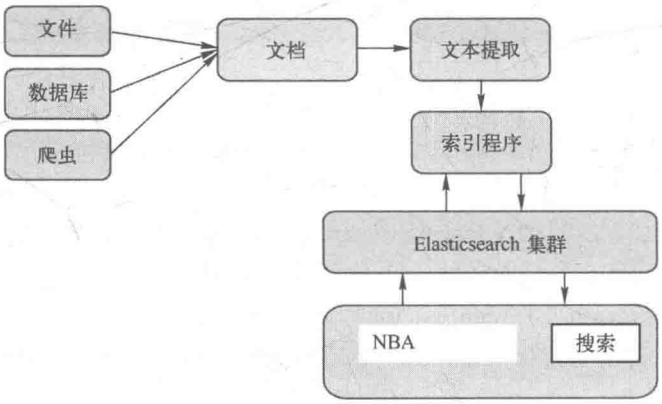

图 1-1　Elasticsearch 的外部结构

基本概念

每一个运行实例称为一个节点，每一个运行实例既可以在同一个机器上，也可以在不同的机器上。

在一个分布式系统里面，可以通过多个 Elasticsearch 运行实例组成一个集群（Cluster），这个集群里面有一个节点叫作主节点（master），Elasticsearch 是去中心化的，所以这里的主节点是动态选举出来的，不存在单点故障。

在同一个子网内，只要在每个节点上设置相同的集群名，Elasticsearch 就会自动把这些集群名相同的节点组成一个集群。节点和节点之间的通信以及节点之间的数据分配和平衡全部由 Elasticsearch 自动管理。

试读结束：需要全本请在线购买：　www.ertongbook.com

Elasticsearch 会把一个索引（Index）分解为多个小的索引，每一个小的索引就叫作分片（Shards）。

Elasticsearch 的每一个分片都可以有 0 到多个副本（Replicas），而每一个副本也都是分片的完整复制品，其好处是可以用它来增加速度的同时也提高了系统的容错性。

一旦 Elasticsearch 的某个节点数据损坏或服务不可用时，那么就可以用其他节点来代替坏掉的节点，以达到高可用的目的。

当有节点加入或退出时，它会根据机器的负载对索引分片进行重新分配，当“挂掉”的节点再次重新启动时也会进行数据恢复（Recovery）。

通过网关（Gateway）来管理集群恢复，可以配置集群需要加入多少个节点，才能够启动恢复。网关配置用于恢复任何失败的索引。当节点崩溃并重新启动时，Elasticsearch 将从网关读取所有索引和元数据。

Transport 代表 Elasticsearch 内部的节点或集群与客户端之间的交互方式。默认使用 TCP 协议来进行交互，同时它支持 HTTP 协议（json 格式）、Thrift、Servlet、Memcached、ZeroMQ 等多种的传输协议（通过插件方式集成）。

为了让集群能够在运行时动态附加额外的功能，使用插件机制加载实现公共接口的程序集。Elasticsearch 插件用于以各种特定的方式扩展基本的 Elasticsearch 功能。

安装

首先介绍 Elasticsearch 在 Windows 下的安装，然后介绍在 Linux 下的安装。

安装包下载网址：http://www.elasticsearch.org/download/。这里使用的版本为 5.1.2。得到文件：elasticsearch-5.1.2.zip

直接解压至某目录，例如，D:\elasticsearch-5.1.2。下载完成解压后有以下几个路径：bin 是运行的脚本，config 是设置文件，lib 中放依赖的包。

到目录 D:\elasticsearch-5.1.2\bin 下，运行 elasticsearch.bat。

如果显示 Java 虚拟机内存不够，则可以在 D:\elasticsearch-5.1.2\config\jvm.options 配置文件中调整内存大小。

成功启动 Elasticsearch 后，在浏览器中打开网址：http://localhost:9200/。

启动成功后，会在解压目录下增加两个文件夹：data 用于存储索引数据，logs 用于日志记录。因为创建索引耗时，所以将文档预先写入一个日志目录中。

通过 HTTP 协议发送指令来和 Elasticsearch 交互。curl 是一个知名的网络命令行工具，可以用来发送 GET 或 POST 命令。在 Linux 下默认已经安装了这个命令行工具，但也有 Windows 版本的 curl，下载地址为 http://www.paehl.com/open_source/，这是一个编译好的 curl.exe 文件。需要在 Windows 命令行下运行这个工具。

默认情况下，Elasticsearch 的 RESTful 服务只有本机才能访问，也就是说无法从主机访问虚拟机中的服务。为了方便调试，可以修改 config/elasticsearch.yml 文件，加入以下两行：

```
http.host: 0.0.0.0
transport.host: 127.0.0.1
```

但线上环境切忌不要这样配置，否则任何人都可以通过这个接口修改 ES 的数据。

为了看到索引内容，需要安装 head 插件。Elasticsearch 5.x 安装 head 插件需要随同 NodeJS 一起安装的包管理工具 npm。

在 Linux 下首先安装 JDK。

```
#wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie" http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.rpm
# rpm -i ./jdk-8u131-linux-x64.rpm
```

验证 Java 安装：

```
#java –version
```

输出如下：

```
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

启动时可能会出现一些错误，根据需要增加打开文件和进程的数量以及虚拟内存数量。

编辑 limits.conf：

```
vi /etc/security/limits.conf
```

添加如下内容：

```
* soft nofile 65536
* hard nofile 131072
* soft nproc 2048
* hard nproc 4096
```

编辑 90-nproc.conf。

```
vi /etc/security/limits.d/90-nproc.conf
```

修改如下内容：

```
* soft nproc 1024
```

修改为：

```
* soft nproc 2048
```

修改配置文件 sysctl.conf：

```
vi /etc/sysctl.conf
```

添加下面配置：

```
vm.max_map_count=655360
```

执行命令：

```
sysctl -p
```

在 Linux 下不能以 root 用户启动 Elasticsearch，所以需要先创建用户。

```
# adduser ops
```

设置密码：

```
# passwd ops
```

下载并解压缩 elasticsearch-5.1.2.tar.gz。启动：

```
# sh elasticsearch
```

执行 elasticsearch 默认会以控制台的方式执行，如果想脱离控制台运行，加上参数 -d 即可（./elasticsearch -d）。

对于旧版本的 Linux，启动时会提示警告：unable to install syscall filter，可以忽略这个警告。这个过滤器是 Elasticsearch 安全模块做出了额外的努力，撤销 Linux 进程权限，减少“攻击载体”恶意活动。

使用 HTTP 请求与 Elasticsearch 打交道。HTTP 请求包括请求的 URL 地址、HTTP 命令（GET、POST 等）等。为了简洁而一致地描述 HTTP 请求，Elasticsearch 文档使用 cURL 命令行语法。这也是在用户社区中描述对 Elasticsearch 的请求的标准做法。通过 cURL 命令发送 HTTP 请求给本地节点的例子如下。


```
# curl -XGET 'http://localhost:9200/'
```

使用 cURL 的简单搜索请求：

```
# curl -XPOST "http://localhost:9200/_search" -d'
    {
        "query": {
          "match_all": {}
        }
    }'
```

上述代码片段在控制台中执行时，使用 3 个参数运行 cURL 程序。第 1 个参数-XPOST 意味着 cURL 所做的请求应使用 HTTP 的 POST 请求；第 2 个参数 http://localhost:9200/_search 是请求的 URL；第 3 个参数-d'{...}'使用-d 来指示 cURL 发送跟随这个标记的 HTTP POST 数据。

可以用字符浏览器 links 访问：

```
# links http://localhost:9200/
```

head 插件是 Elasticsearch 集群的 Web 前端，作为一个单独的 Web 应用运行。在 Linux 下安装 head 插件的过程如下。

首先安装 npm。npm 是一个 node 包管理和分发工具，安装命令如下。

```
# yum install npm
```

然后下载 elasticsearch-head。用 git 命令复制一个小的本地仓库。

```
# git clone git://github.com/mobz/elasticsearch-head.git
```

```
# cd elasticsearch-head
```

安装包：

```
# npm install
```

启动：

```
# npm run start
```

为了避免出现跨域问题，在文件 elasticsearch.yml 中添加如下配置。

```
http.cors.enabled: true
http.cors.allow-origin: "*"
```

通过 CURL 发送 POST 请求停止节点。停止本地节点：

```
# curl -XPOST 'http://localhost:9200/_cluster/nodes/_local/_shutdown'
```

停止集群中所有的节点：

```
# curl -XPOST 'http://localhost:9200/_shutdown'
```


1.3 搜索集群

分布式索引需要考虑的问题如下。

- 确定一个集群中应包括哪些机器。
- 数据如何分布。
- 快速的分布式打分。

数据分布有两种方法：按文档分片或按词分片。不同的文档集合放在不同的机器上，不同的词列表集合放在不同的机器上。Elasticsearch 仍然是把不同的文档集合放在不同的机器上。

最简单的方法是手工管理集群。假设要手工建立两台机器组成的集群。修改 elasticsearch.yml 文件如下。

对于主机 esa.lietu.com：

```
cluster.name: OurCluster
node.name: "esa"
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["esb.lietu.com"]
```

对于主机 esb.lietu.com：

```
cluster.name: OurCluster
node.name: "esb"
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["esa.lietu.com"]
```

一般使用多播自动建立集群。只要其他节点与第一个节点有相同的 cluster name，它

就自动发现并加入第一个节点的集群。

1.4 创建索引

表的结构和相关设置的信息在 mapping 中设置。

索引模板也可以放在模板目录下的配置位置（path.conf）。注意，所有有主节点资格的节点都要放。例如，一个叫作 template_1.json 的文件可以放在 config/templates 目录下，如果它能够匹配一个索引，就会把它添加进去。放在 config/[index_name]/[some_name].json 的例子如下。

```
{
    "[type]" : {
        "properties" : {
            "title" : {
                "type" : "string",
                "boost" : 2.0
            }
            "tags": {
                "type" : "string"
            }
        }
    }
}
```

在命令行设置 Mapping 的例子如下。

```
curl -XPOST localhost:9200/wf_mds_org（索引名称） -d'{
    "settings": {
        "number_of_shards": 1,
        "number_of_replicas": 0,
        "index.refresh_interval": "-1",
        "index.translog.flush_threshold_ops": "100000"
    },
    "mappings": {
        "org": {    //(类型)
            "_all": {
                "analyzer": "ike"
            },
            "_source": {
                "compress": true
            },
            "properties": {
                "_ID": {
                    "type": "string",
                    "include_in_all": true,
                    "analyzer": "keyword"
```

```
},
                    "NAME": {
    "type": "multi_field",
    "fields": {
        "NAME": {
            "type": "string",
            "analyzer": "keyword"
        },
        "IKO": {
            "type": "string",
            "analyzer": "ike"
        }
    }
},
                    "SHORTNAME": {
    "type": "string",
    "index_analyzer": "pct_spliter",
    "search_analyzer": "keyword",
    "store": "no"
},
                    "OLDNAME": {
    "type": "multi_field",
    "fields": {
        "OLDNAME": {
            "type": "string",
            "analyzer": "keyword"
        },
        "IKO": {
            "type": "string",
            "analyzer": "ike"
        }
    }
},
                    "TNAME": {
    "type": "string",
    "analyzer":"custom_snowball_analyzer",
    "store": "no"
},
                    "TSNAME": {
    "type": "string",
    "index": "no",
    "store": "no"
},
                    "TONAME": {
    "type": "string",
    "index": "no",
```


试读结束：需要全本请在线购买：　www.ertongbook.com