数据分析与决策技术丛书
HZ BOOKS
华章IT

Principles and Practice of Elasticsearch

Elasticsearch
技术解析与实战

朱林 编著

包含Elasticsearch 5最新功能，凝聚了作者多年开发经验

分布式大数据全文搜索与数据挖掘必备工具

机械工业出版社
China Machine Press

数据分析与决策技术丛书

Elasticsearch 技术解析与实战

朱林　编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Elasticsearch 技术解析与实战 / 朱林编著 . —北京：机械工业出版社，2016.12
（数据分析与决策技术丛书）

ISBN 978-7-111-55327-4

Ⅰ. E… Ⅱ. 朱… Ⅲ. 互联网络 - 情报检索 Ⅳ. G354.4

中国版本图书馆 CIP 数据核字（2016）第 274894 号

Elasticsearch 技术解析与实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：吴 怡 责任校对：殷 虹
印 刷：北京市荣盛彩色印刷有限公司 版 次：2017 年 1 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：27.25
书 号：ISBN 978-7-111-55327-4 定 价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

Preface 前 言

Elasticsearch 是目前全球最受欢迎的全文搜索引擎。初识 Elasticsearch 是在 2012 年的一个项目中，当时 Elasticsearch 还是 0.19.0 版本，但是功能已经比较强大，只是接口稍微有点复杂。到了 2015 年年初，公司开发了一款日志分析产品，它实时不间断地采集用户网络中各种不同系统的日志，然后从中分析系统的安全情况、系统情况、业务情况。最初所有的数据都存储在 MySQL 中，随着日志的不断增加，MySQL 搜索速度越来越慢。后来在更换技术架构选型的时候又想到了 Elasticsearch，这个时候 Elasticsearch 已经是 1.6.0 版本了。我们对此进行了简单的测试，在上亿条的数据搜索中很多都在一秒内完成，在上亿条的数据中进行统计分析大多也是在秒级完成，它展示了强大实力。我们顺势就把 Elasticsearch 整合到了现在的产品中，取得了很好效果。到了 2016 年 3 月的时候，Elasticsearch 发布了 2.3.0 版本，各方面更加成熟，我们的产品又再一次升级到这个新版本上。

Elasticsearch 产品的更新变化非常快，在我们开发研究的过程中基本上找不到新版本的中文资料，目前市场上介绍 Elasticsearch 的中文书籍都是在版本 1.0 左右，甚至更早，这些书的很多内容尤其是开发接口相关的部分都已经过时，没有办法在新版本中使用。所以我们开发的过程中基本上都是研究官方文档，有时候甚至研究它的源码才能解决问题。在接口选择的时候我们在 HTTP JSON 接口和 Java 接口中做了取舍，我们当时分析 HTTP JSON 接口最终还是要转换成 Java 接口，不如直接使用 Java 接口，一是效率可能更高，二是在部署实施的时候减少一个端口，三是对后续的升级更有利，比如后续增加权限认证等。但这些东西都没有资料，我们基本上都是研究系统源码来克服的。在后续研究过程中，我们发现 HTTP 接口转换到 Java 接口是有规律的，所以对 HTTP 接口的掌握对后续 Elasticsearch 的开发和扩展也有很大的帮助。在持续研究的过程中，我们积累了大量经验，并想把这些经验分享给更多需要的人。后来我把这个想法给出版社的吴怡编辑做了沟通，她非常支持我们的想法，便有了这本书。

本书首先介绍 Elasticsearch 的相关基础知识，然后由浅入深地介绍 Elasticsearch 索引查

询相关的知识，包括索引、映射、搜索、聚合，接着介绍 Elasticsearch 的集群、分词、重要的配置等高级功能，以及 Elasticsearch 相关的其他产品，包括告警、监控、权限管理，最后通过一个 ELK 示例结束本书。在写作的时候考虑到读者的接受能力，由浅入深地进行讲解，建议读者从前往后阅读。

本书主要内容包括：

第 1 章 “Elasticsearch 入门”，介绍 Elasticsearch 是什么、Apache Lucene 的基础知识、Elasticsearch 的术语、JSON 介绍、Elasticsearch 的安装运行、Elasticsearch 的 HTTP 接口和 Elasticsearch 的 Java API 接口。

第 2 章 “索引”，介绍和 Elasticsearch 索引相关的接口，包括索引管理、索引映射管理、索引别名、索引设置、索引监控、索引其他重要接口以及文档管理。

第 3 章 “映射”，介绍 Elasticsearch 文档的内部结构，Elasticsearch 支持的字段类型，除此之外，本章还将展示 Elasticsearch 内置的元字段，映射的参数和动态映射功能。

第 4 章 “搜索”，详细介绍和搜索相关的知识，包括搜索的详细参数，搜索的评分机制、滚动查询、系统内部隐藏内容的查询、搜索模板等；接着介绍 Elasticsearch 的领域查询语言 DSL（Domain-specific Language）相关的知识点；最后介绍 Elasticsearch 的精简查询接口。

第 5 章 “聚合”，聚合可以对文档中的数据进行统计汇总、分组等，通过聚合可以完成很多的统计功能，该章介绍聚合相关的知识，包括度量聚合、分组聚合和管道聚合。

第 6 章 “集群管理”，详细介绍和集群相关的内容，包括集群的监控、集群分片迁移、集群的节点配置、集群发现、集群平衡的原理和配置。

第 7 章 “索引分词器”，介绍 Elasticsearch 的分词器和分词的原理，以及如何添加新的分词器等；还介绍 Elasticsearch 的插件相关知识，包括插件安装等。

第 8 章 “高级配置”，介绍 Elasticsearch 的高级配置，包括网络配置、脚本配置、快照和恢复配置、线程池配置和索引配置。

第 9 章 “告警、监控和权限管理”，介绍 Elasticsearch 官方支持的几个比较好的插件：Watcher、Marvel、Shield，它们可以对 Elasticsearch 进行告警、监控和权限管理。

第 10 章 “ELK 应用”，介绍 Elasticsearch 与另外两个产品 Logstash 和 Kibana 如何组合使用，Logstash 是对日志进行收集和处理，Kibana 是对存储在 Elasticsearch 中的索引进行展示和报表分析；最后通过一个简单的示例来介绍 ELK 几个产品是如何关联的。

在编写本书的时候，Elasticsearch 的最新版本是 2.2.0，但本书准备正式出版的时候，Elasticsearch 发布了最新的 5.0 版本。所以本书增加了一个附录专门介绍 5.0 版本的特性与改进。本书前面的部分截图是 2.2.0 版本的，书中所有的例子和功能都可以在 Elasticsearch 2.3.3 下运行，大部分的功能都可以在 5.0 下运行，详细的新版本差别请参考附录部分。本书中的例子大部分都是 HTTP 接口的，这些接口的测试使用了 Elasticsearch Head 插件。如果你想使用另一种工具，请注意修改 HTTP 请求的格式和编码，以便适合你所选择的工具。书中例子的结构大多是 JSON 格式，美化后的 JSON 格式比较容易阅读，但美化后的 JSON 格式比较

长，所以我们在不影响阅读的情况下，对美化后的格式做了简单调整。书中还有一小部分是Java 接口，我们在实验时用的是 Eclipse 工具，其他主流的 Java 开发工具都适用。

本书的目标读者是对全文检索和 Elasticsearch 有兴趣的读者，如果你是一个初学者，通过本书你将学到 Elasticsearch 的基础知识，以及如何使用一些高级功能。如果你已经知道并使用了 Elasticsearch 但又想深入了解其本身，想了解如何改进查询相关性，如何使用 Elasticsearch Java API 等，也会发现本书的实用性。

由于时间紧，能力有限，编写的过程中难免有不当之处，还请各位读者不吝指出。

致谢

在此，首先我想感谢我的家庭，多年以来，因为工作关系我照顾他们太少，他们为我付出太多；我在全身心投入本书写作的时候，他们同样表现出了极大的耐心，是我最坚强的后盾。

其次还要感谢赛克蓝德公司以及公司的同事：周忠立、万荣慧和夏海华，是他们的辛勤工作才完成本书的编写工作，尤其是周忠立在很多章节的编写上贡献了很多的内容；同时要感谢本书的出版团队，尤其是吴怡编辑，写书是件非常艰巨的任务，很多内容需要反复推敲才能表达准确的意思，吴怡在检查错误、校稿、消除表达歧义等方面做出了很多贡献。

最后，非常诚挚地感谢所有 Elasticsearch 项目的创建者和开发者，感谢他们杰出的工作和对开源项目的热情。没有他们，就没有本书的诞生，没有他们，开源搜索引擎就不会有现在这种活力。再次感谢！

朱林

2016 年 10 月于南京

目 录 *Contents*

第 1 章 Chapter 1
Elasticsearch 入门

欢迎来到 Elasticsearch 世界，它目前是全球最受欢迎的全文搜索引擎。你对 Elasticsearch 和全文搜索有没有经验都不要紧。我们希望你可以通过这本书走进 Elasticsearch 的大门。这本书是为初学者准备的，当然对于中高级的人员也有参考作用。我们首先介绍一些和 Elasticsearch 相关的基础内容。接着介绍一下 Elasticsearch 的安装和配置。此外本章还会介绍如何简单使用 Elasticsearch，包括 HTTP JSON 接口和 Java 接口。在学习这些接口的过程中，不用陷入太多的细节，后面的章节会逐步展开并细化接口内容。读完本章，你将学到以下内容：

- Elasticsearch 介绍
- 全文搜索
- Elasticsearch 的基础知识
- 安装和配置 Elasticsearch
- HTTP REST API 接口
- Java 开发接口

1.1 Elasticsearch 是什么

Elasticsearch（ES）是一个基于 Lucene 构建的开源、分布式、RESTful 接口全文搜索引擎。Elasticsearch 还是一个分布式文档数据库，其中每个字段均是被索引的数据且可被搜索，它能够扩展至数以百计的服务器存储以及处理 PB 级的数据。它可以在很短的时间内存储、搜索和分析大量的数据。它通常作为具有复杂搜索场景情况下的核心发动机。

Elasticsearch 就是为高可用和可扩展而生的。可以通过购置性能更强的服务器来完成，

称为垂直扩展或者向上扩展（Vertical Scale/Scaling Up），或增加更多的服务器来完成，称为水平扩展或者向外扩展（Horizontal Scale/Scaling Out）。

尽管 ES 能够利用更强劲的硬件，垂直扩展毕竟还是有它的极限。真正的可扩展性来自于水平扩展，通过向集群中添加更多的节点来分担负载，增加可靠性。

在大多数数据库中，水平扩展通常都需要你对应用进行一次大的重构来利用更多的节点。而 ES 天生就是分布式的：它知道如何管理多个节点来完成扩展和实现高可用性。这也意味着你的应用不需要做任何的改动。

我们举几个例子来说明 Elasticsearch 能做什么？

当你经营一家网上商店，你可以让你的客户搜索你卖的商品。在这种情况下，你可以使用 Elasticsearch 来存储你的整个产品目录和库存信息，为客户提供精准搜索，可以为客户推荐相关商品。

当你想收集日志或者交易数据的时候，需要分析和挖掘这些数据，寻找趋势，进行统计，总结，或发现异常。在这种情况下，你可以使用 Logstash 或者其他工具来进行收集数据，当这些数据存储到 Elasticsearch 中。你可以搜索和汇总这些数据，找到任何你感兴趣的信息。

当你运行一个价格提醒的平台，可以给客户提供一些规则，例如客户有兴趣购买一个电子设备，当商品的价格在未来一个月内价格低于多少钱的时候通知客户。在这种情况下，你可以把供应商的价格，把他们定期存储到 Elasticsearch 中，使用定时器过滤来匹配客户的需求，当查询到价格低于客户设定的值后给客户发送一条通知。

当有大量数据（千万条以上的记录）时，你有商业智能分析的需求，希望快速调查、分析和可视化。在这种情况下，你可以使用 Elasticsearch 来存储你的数据，然后用 Kibana 建立自定义的仪表板或者任何你熟悉的语言开发展示界面，你可以使用 Elasticsearch 的聚合功能来执行复杂的商业智能与数据查询。

对于程序员来说，比较有名的案例是 GitHub，GitHub 的搜索是基于 Elasticsearch 构建的，在 github.com/search 页面，你可以搜索项目、用户、issue、pull request，还有代码。共有 40 ~ 50 个索引库，分别用于索引网站需要跟踪的各种数据。虽然只索引项目的主分支（master），但这个数据量依然巨大，包括 20 亿个索引文档，30TB 的索引文件。

1.1.1 Elasticsearch 的历史

网上流传的故事是：多年前，一个叫作 Shay Banon 的刚结婚不久的失业开发者，由于妻子要去伦敦学习厨师，他便跟着也去了。在他找工作的过程中，为了给妻子构建一个食谱的搜索引擎，他开始构建一个早期版本的 Lucene。

直接基于 Lucene 工作会比较困难，所以 Shay 开始抽象 Lucene 代码以便 Java 程序员可以在应用中添加搜索功能。他发布了第一个开源项目，叫作“Compass”。

后来 Shay 找到一份工作，这份工作处在高性能和内存数据网格的分布式环境中，因此

高性能的、实时的、分布式的搜索引擎也是理所当然需要的。然后他决定重写 Compass 库使其成为一个独立的服务叫作 Elasticsearch。

第一个公开版本发布于 2010 年 2 月，在那之后 Elasticsearch 已经成为 GitHub 上最受欢迎的项目之一，代码贡献者超过 300 人。直到 2016 年 3 月 30 日，Elasticsearch 已经发布了 2.3.0 版本。目前已经成为全球最受欢迎的全文搜索引擎。

那 Elasticsearch 为什么会有如此的魅力呢？我们首先看一下 Elasticsearch 的优点：

- **横向可扩展性**：只需要增加一台服务器，做一点儿配置，启动一下 Elasticsearch 进程就可以并入集群。
- **分片机制提供更好的分布性**：同一个索引分成多个分片（sharding），这点类似于 HDFS 的块机制；分而治之的方式可提升处理效率。
- **高可用**：提供复制（replica）机制，一个分片可以设置多个复制，使得某台服务器在宕机的情况下，集群仍旧可以照常运行，并会把服务器宕机丢失的数据信息复制恢复到其他可用节点上。
- **使用简单**：只需一条命令就可以下载文件，然后很快就能搭建一个站内搜索引擎。

1.1.2 相关产品

Beats：它是一个代理，将不同类型的数据发送到 Elasticsearch 中。它可以直接将数据发送到 Elasticsearch。Beats 由三部分内容组成：Filebeat、Topbeat、Packetbeat。Filebeat 用来收集日志。Topbeat 用来收集系统基础设置数据，如 CPU、内存、每个进程的统计信息。Packetbeat 是一个网络包分析工具，统计收集网络信息。这三个工具是官方提供的。

Shield：它为 Elasticsearch 带来企业级的安全性，加密通信，认证保护整个 Elasticsearch 数据，它是基于角色的访问控制与审计。当今企业对安全需求越来越重视，Shield 可以提供安全的 Elasticsearch 访问，从而保护核心的数据。注意：Shield 是收费的产品。

Watcher：它是 Elasticsearch 的警报和通知工具。它可以主动监测 Elasticsearch 的状态，并在有异常的时候进行提醒，还可以根据你的数据变化情况来采取不同的处理方式。注意：Watcher 也是收费的产品。

Marvel：它是 Elasticsearch 的管理和监控工具。它监测 Elasticsearch 集群索引和节点的活动，快速诊断问题。注意：Marvel 也是收费的产品。

1.2 全文搜索

全文搜索是指计算机搜索程序通过扫描文章中的每一个词，对每一个词建立一个索引，指明该词在文章中出现的次数和位置，当用户查询时，搜索程序就根据事先建立的索引进行查找，并将查找的结果反馈给用户。这个过程类似于通过字典中的搜索字表查字的过程。Lucene 是目前全球使用最广的全文搜索引擎开源库。

1.2.1 Lucene 介绍

Lucene 是 Apache 软件基金会中一个开放源代码的全文搜索引擎工具包，是一个全文搜索引擎的架构，提供了完整的查询引擎和索引引擎，部分文本分析引擎。Lucene 的目的是为软件开发人员提供一个简单易用的工具包，以方便在目标系统中实现全文检索的功能，或者是以此为基础建立起完整的全文搜索引擎。

Lucene 最初是由 Doug Cutting 所撰写的，是一位资深全文索引 / 搜索专家，曾经是 V-Twin 搜索引擎的主要开发者，后来在 Excite 担任高级系统架构设计师，目前从事于一些 Internet 底层架构的研究。

1.2.2 Lucene 倒排索引

倒排索引源于实际应用中需要根据属性的值来查找记录。这种索引表中的每一项都包括一个属性值和具有该属性值的各记录的地址。由于不是由记录来确定属性值，而是由属性值来确定记录的位置，因而称为倒排索引（inverted index）。带有倒排索引的文件我们称为倒排索引文件，简称倒排文件（inverted file）。

倒排索引中的索引对象是文档或者文档集合中的单词等，用来存储这些单词在一个文档或者一组文档中的存储位置，是对文档或者文档集合的一种最常用的索引机制。

搜索引擎的关键步骤就是建立倒排索引，倒排索引一般表示为一个关键词，然后是它的频度（出现的次数）、位置（出现在哪一篇文章或网页中，及有关的日期，作者等信息），好比一本书的目录、标签一般。读者想看哪一个主题相关的章节，直接根据目录即可找到相关的页面。不必再从书的第一页到最后一页，一页一页地查找。

Lucene 使用的是倒排文件索引结构，下面用例子介绍该结构及相应的生成算法。

假设有两篇文章 1 和文章 2。

文章 1 的内容为：Tom lives in Guangzhou,I live in Guangzhou too.

文章 2 的内容为：He once lived in Shanghai.

1. 取得关键词

由于 Lucene 是基于关键词索引和查询的，首先我们要取得这两篇文章的关键词，通常我们需要如下处理措施：

- 我们现在有的是文章内容，即一个字符串，我们先要找出字符串中的所有单词，即分词。英文单词由于用空格分隔，比较好处理。中文单词间由于是连在一起的，所以需要特殊的分词处理。
- 文章中的“in”“once”“too”等词没有什么实际意义，中文中的“的”“是”等字通常也无具体含义，这些不代表概念的词是可以过滤掉的。
- 用户通常希望查“He”时能把含“he”和“HE”的文章也找出来，所以所有单词需要统一大小写。

- 用户通常希望查“live”时能把含“lives”和“lived”的文章也找出来，所以需要把“lives”，“lived”还原成“live”。
- 文章中的标点符号通常不表示某种概念，也可以过滤掉。

在Lucene中以上措施由Analyzer类完成。经过上面处理后，得到如下结果：

文章1的所有关键词为：[tom] [live] [guangzhou] [i] [live] [guangzhou]

文章2的所有关键词为：[he] [live] [shanghai]

2. 建立倒排索引

有了关键词后，我们就可以建立倒排索引了。上面的对应关系是：“文章号”对“文章中所有关键词”。倒排索引把这个关系倒过来，变成：“关键词”对“拥有该关键词的所有文章号”。

文章1和文章2经过倒排后的对应关系见表1-1。

表1-1 倒排索引关键词文章号对应关系示例

关键词	文章号
guangzhou	1
he	2
i	1
live	1，2
shanghai	2
tom	1

通常仅知道关键词在哪些文章中出现还不够，我们还需要知道关键词在文章中出现的次数和位置，通常有两种位置：

- 字符位置，即记录该词是文章中第几个字符（优点是显示并定位关键词快）。
- 关键词位置，即记录该词是文章中第几个关键词（优点是节约索引空间、词组查询快），Lucene中记录的就是这种位置。

加上“出现频率”和“出现位置”信息后，我们的索引结构参见表1-2.

表1-2 倒排索引关键词频率位置示例

关键词	文章号[出现频率]	出现位置
guangzhou	1[2]	3,6
he	2[1]	1
i	1[1]	4
live	1[2] 2[1]	2,5 2
shanghai	2[1]	3
tom	1[1]	1

以live这行为例，我们说明一下该结构：live在文章1中出现了2次，文章2中出现了一次，它的出现位置为“2,5,2”这表示什么呢？我们需要结合文章号和出现频率来分析，文章1中出现了2次，那么“2,5”就表示live在文章1中出现的两个位置，文章2中出现了一次，剩下的“2”就表示live是文章2中的第2个关键字。

以上就是Lucene索引结构中最核心的部分。我们注意到关键字是按字符顺序排列的

（Lucene 没有使用 B 树结构），因此 Lucene 可以用二元搜索算法快速定位关键词。

3. 实现

实现时，Lucene 将上面三列分别作为词典文件（Term Dictionary）、频率文件（frequencies）、位置文件（positions）保存。其中词典文件不仅保存了每个关键词，还保留了指向频率文件和位置文件的指针，通过指针可以找到该关键字的频率信息和位置信息。

Lucene 中使用了 field 的概念，用于表达信息所在位置（如标题中、文章中、URL 中），在建索引中，该 field 信息也记录在词典文件中，每个关键词都有一个 field 信息，因为每个关键字一定属于一个或多个 field。

4. 压缩算法

为了减小索引文件的大小，Lucene 对索引还使用了压缩技术。

首先，对词典文件中的关键词进行了压缩，关键词压缩为 < 前缀长度，后缀 >，例如：当前词为“阿拉伯语”，上一个词为“阿拉伯”，那么“阿拉伯语”压缩为 <3，语 >。

其次大量用到的是对数字的压缩，数字只保存与上一个值的差值（这样可以减少数字的长度，进而减少保存该数字需要的字节数）。例如当前文章号是 16389（不压缩要用 3 个字节保存），上一文章号是 16382，压缩后保存 7（只用一个字节）。

5. 应用场景

下面我们可以通过对该索引的查询来解释一下为什么要建立索引。

假设要查询单词“ live”，Lucene 先对词典二元查找、找到该词，通过指向频率文件的指针读出所有文章号，然后返回结果。词典通常非常小，因而，整个过程的时间是毫秒级的。

而用普通的顺序匹配算法，不建索引，而是对所有文章的内容进行字符串匹配，这个过程将会相当缓慢，当文章数目很大时，时间往往是无法忍受的。

1.3 基础知识

在 Elasticsearch 中有很多的术语和概念，为了后面更好地理解和阅读本书，本节先介绍一下这些术语和概念，然后介绍一下 Elasticsearch 存储的格式 JSON。

1.3.1 Elasticsearch 术语及概念

1. 索引词（term）

在 Elasticsearch 中索引词（term）是一个能够被索引的精确值。foo、Foo、FOO 几个单词是不同的索引词。索引词（term）是可以通过 term 查询进行准确的搜索。