

深入理解Elasticsearch（原书第2版）



[深入理解Elasticsearch（原书第2版）_下载链接1_](#)

著者:Rafal Kuc

出版者:机械工业出版社

出版时间:2017-5

装帧:

isbn:9787111568254

作者介绍:

ElasticSearch是一个优秀的开源分布式搜索引擎，同时有良好的社区和商业支持。对于中小型的垂直搜索引擎，ElasticSearch是一个不错的选择。本书是一本ElasticSearch的进阶教材，深入剖析DSL、索引控制、分布式实现、系统运维等高级内容，特别适合深入研究ElasticSearch。

—— 徐川 明星衣橱CTO，前雅虎高级工程师

ElasticSearch的出现，让开源搜索产品真正进入分布式时代。本书是一本不可多得的关于ElasticSearch的著作，既对ElasticSearch的全文索引、IR模型、分布式机制有深入剖析，又有生动翔实的示例，能帮助读者快速提升在该领域的技术水平。

—— 高剑林 腾讯（架构平台部）资深技术专家

除了用于搜索，ElasticSearch也是日志存储、离线数据分析挖掘的利器。本书深入浅出，案例丰富，在信息检索模型、准实时搜索、分布式架构、系统优化等诸多方面都有精彩的论述。

—— 李伟博士 微软（bing）数据挖掘组高级工程师

很高兴看到《Mastering ElasticSearch》中文版面市，本书对ElasticSearch的分布式系统架构、系统调优有较深入的探讨，是一本进阶的好读物，其中一些系统设计思维对于文件系统研发人员也是有所裨益的。

—— 许加强 前IBM(GPFS)资深工程师

尽管ElasticSearch是一个开源搜索产品，它在百度也被广泛应用。目前已经覆盖到20多个业务线。这本书针对性较强，既不乏典型实例，也有一定的理论深度。非常适合进阶用户阅读。

—— 陈铁兵 百度网页搜索部高级工程师

目录: 译者序

作者简介

评审者简介

前言

第1章 Elasticsearch简介 1

1.1 Apache Lucene简介 1

1.1.1 熟悉Lucene 2

1.1.2 Lucene的总体架构 2

1.1.3 分析数据 4

1.1.4 Lucene查询语言 5

1.2 何为Elasticsearch 8

1.2.1 Elasticsearch的基本概念 8

1.2.2 Elasticsearch架构背后的关键概念 10

1.2.3 Elasticsearch的工作流程 10

1.3 在线书店示例 14

1.4 小结 17

第2章 查询DSL进阶 18

2.1 Apache Lucene默认评分公式解释 18

2.1.1 何时文档被匹配上 19

2.1.2	TF/IDF评分公式	19
2.1.3	Elasticsearch如何看评分	21
2.1.4	一个例子	21
2.2	查询改写	24
2.2.1	前缀查询示例	24
2.2.2	回到Apache Lucene	26
2.2.3	查询改写的属性	28
2.3	查询模板	30
2.3.1	引入查询模板	31
2.3.2	Mustache模板引擎	33
2.3.3	把查询模板保存到文件	35
2.4	过滤器的使用及作用原理	36
2.4.1	过滤及查询相关性	36
2.4.2	过滤器的工作原理	40
2.4.3	性能考量	41
2.4.4	后置过滤和过滤查询	42
2.4.5	选择正确的过滤方式	44
2.5	选择正确的查询方式	45
2.5.1	查询方式分类	45
2.5.2	使用示例	50
2.6	小结	65
第3章	不只是文本搜索	66
3.1	查询二次评分	66
3.1.1	什么是查询二次评分	67
3.1.2	一个查询例子	67
3.1.3	二次评分查询的结构	67
3.1.4	二次评分参数	70
3.1.5	总结	70
3.2	多匹配控制	71
3.3	重要词项聚合	78
3.3.1	一个例子	79
3.3.2	选择重要词项	81
3.3.3	多值分析	81
3.3.4	额外的配置	84
3.3.5	使用限制	89
3.4	文档分组	89
3.4.1	top_hits聚合	90
3.4.2	一个例子	90
3.5	文档关系	95
3.5.1	对象类型	95
3.5.2	嵌套文档	98
3.5.3	parent-child关系	99
3.5.4	其他解决方案	102
3.6	Elasticsearch各版本中脚本的变化	102
3.6.1	脚本变迁	102
3.6.2	Groovy简单介绍	103
3.6.3	全文检索中的脚本	108
3.6.4	Lucene表达式	115
3.7	小结	118
第4章	改善用户搜索体验	119
4.1	改正用户拼写错误	119
4.1.1	测试数据	120
4.1.2	深入技术细节	121
4.1.3	suggester	121

- 4.2 改善查询相关性 142
 - 4.2.1 数据 142
 - 4.2.2 改善相关性的探索之旅 145
- 4.3 小结 157
- 第5章 分布式索引架构 159
 - 5.1 选择合适的分片和副本数 159
 - 5.1.1 分片和过度分配 160
 - 5.1.2 一个过度分配的正面例子 161
 - 5.1.3 多分片与多索引 161
 - 5.1.4 副本 161
 - 5.2 路由 162
 - 5.2.1 分片和数据 162
 - 5.2.2 测试路由功能 162
 - 5.2.3 索引时使用路由 166
 - 5.2.4 别名 169
 - 5.2.5 多个路由值 169
 - 5.3 调整默认分片的分配行为 170
 - 5.3.1 部署意识 171
 - 5.3.2 过滤 173
 - 5.3.3 运行时更新分配策略 174
 - 5.3.4 确定每个节点允许的总分片数 175
 - 5.3.5 确定每个物理机器允许的总分片数 175
 - 5.4 查询执行偏好 179
 - 5.5 小结 181
- 第6章 底层索引控制 182
 - 6.1 改变Apache Lucene的评分方式 182
 - 6.1.1 可用的相似度模型 183
 - 6.1.2 为每字段配置相似度模型 183
 - 6.1.3 相似度模型配置 184
 - 6.1.4 选择默认的相似度模型 185
 - 6.2 选择适当的目录实现—store模块 188
 - 6.3 准实时、提交、更新及事务日志 191
 - 6.3.1 索引更新及更新提交 192
 - 6.3.2 事务日志 193
 - 6.3.3 准实时读取 194
 - 6.4 控制索引合并 195
 - 6.4.1 选择正确的合并策略 196
 - 6.4.2 合并策略配置 197
 - 6.4.3 调度 199
 - 6.5 关于I/O调节 200
 - 6.5.1 控制I/O节流 200
 - 6.5.2 配置 200
 - 6.6 理解Elasticsearch缓存 202
 - 6.6.1 过滤器缓存 203
 - 6.6.2 字段数据缓存 204
 - 6.6.3 查询分片缓存 212
 - 6.6.4 使用circuit breaker 213
 - 6.6.5 清除缓存 214
 - 6.7 小结 215
- 第7章 管理Elasticsearch 216
 - 7.1 发现和恢复模块 216
 - 7.1.1 发现模块的配置 217
 - 7.1.2 主节点 218
 - 7.1.3 网关和恢复模块的配置 223

7.1.4 索引恢复API	226
7.2 使用人类友好的Cat API	229
7.2.1 基础知识	230
7.2.2 使用Cat API	231
7.2.3 一些例子	232
7.3 备份	232
7.4 联盟搜索	236
7.4.1 测试用的集群	236
7.4.2 建立部落节点	237
7.4.3 通过部落节点读取数据	238
7.4.4 通过部落节点写入数据	239
7.4.5 处理索引冲突	240
7.4.6 屏蔽写操作	241
7.5 小结	242
第8章 提高性能	243
8.1 使用doc values来优化查询	243
8.1.1 字段缓存存在的问题	244
8.1.2 使用doc values的例子	245
8.2 了解垃圾回收器	247
8.2.1 Java内存	248
8.2.2 解决垃圾回收问题	249
8.2.3 在类UNIX系统上避免内存交换	254
8.3 对查询做基准测试	255
8.3.1 为基准测试配置集群	256
8.3.2 进行基准测试	256
8.3.3 控制运行中的基准测试	259
8.4 热点线程	261
8.4.1 热点线程的使用说明	261
8.4.2 热点线程API的响应	262
8.5 扩展Elasticsearch	263
8.5.1 垂直扩展	263
8.5.2 水平扩展	264
8.5.3 在高负载的场景下使用Elasticsearch	271
8.6 小结	283
第9章 开发Elasticsearch插件	284
9.1 创建Maven项目	284
9.2 了解基本知识	285
9.2.1 Maven Java项目的结构	285
9.2.2 POM的理念	285
9.2.3 执行构建过程	286
9.2.4 引入Maven装配插件	287
9.3 创建自定义REST行为	289
9.3.1 设定	289
9.3.2 实现细节	289
9.4 创建自定义分析插件	295
9.4.1 实现细节	295
9.4.2 测试自定义分析插件	302
9.5 小结	304
• • • • •	(收起)

标签

es

计算机科学

计算机

评论

书本身内容可以，但是不得不说，翻译得狗屁不通。

内容很垃圾，翻译也很垃圾。所有涉及到原理的东西草草了事，接着就开始说实操，看得我一脸懵逼。

1.比如2.2章节提到的查询改写，前缀查询中提到有四个j开头的范例，但在分析里说只有三个，也不说为什么，错漏百出。

2.3.4章涉及到事务日志的更新，csdn上有许多描述底层写过程的文章，从shared到buffer再到cache和segment

file。这本书就说了几句用事务日志保证一致性，当出现异常的时候会检索事务日志进行恢复。？？这谁不知道啊。关键流程呢？更新算法呢？统统没说，我还不如去搜帖子看呢

一般般，并不算深入

翻译比较糟糕，内容老旧

[深入理解Elasticsearch（原书第2版）_下载链接1](#)

书评

1. 这本书应该还是比较早的中文Elasticsearch书籍，现在的ES已经到了5.1版本，而书中讲解的版本则是0.9，所以阅读时经常会检查下这个特性是否废弃过改进过。 2. 除此之外，本书应该来说是一本比较优秀的书籍，书中介绍的内容适合我这种对ES有一定了解又想深入理解其中机制的读者...

现在的ES版本是6.5,这本书里面提到的ES版本是1.4。另外这本书叫深入理解……，其实有点不太符合深入二字。书中很多内容其实都是官方文档中提到的东西，只是官方文档是英文，如果英文不太好，理解起来不到位，结合一下中文书看看，理解起来效率会高一点。比如说：书中第二章对L...

感觉大部分内容是直接用百度翻译而来，很多地方的语言比较晦涩难懂。。。比如：上面提到的过滤器中，最后三个本身并不使用字段缓存，但由于它们操作其他过滤器，因而它们不缓存。翻译地太生硬。。。还有一些地方，说明数据明显与案例对应的数据不符。。。希望作者下次翻译书籍...

内容是基于ElasticSearch 0.9版本的 19页的例子好像是没有和前面完全对应上
（抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉...）

我是从网上找到的这个例子，它跟我的需求差不多。但是我发现它的实现方式跟我的想法不太一样。我想知道的是：为什么在代码中要使用`getters`方法？以及为什么要用`@Getter`注解来修饰成员变量呢？

```
@Getter private String name; @Getter private Integer age;
```

谢谢！

深入理解Elasticsearch (原书第2版) 下载链接1