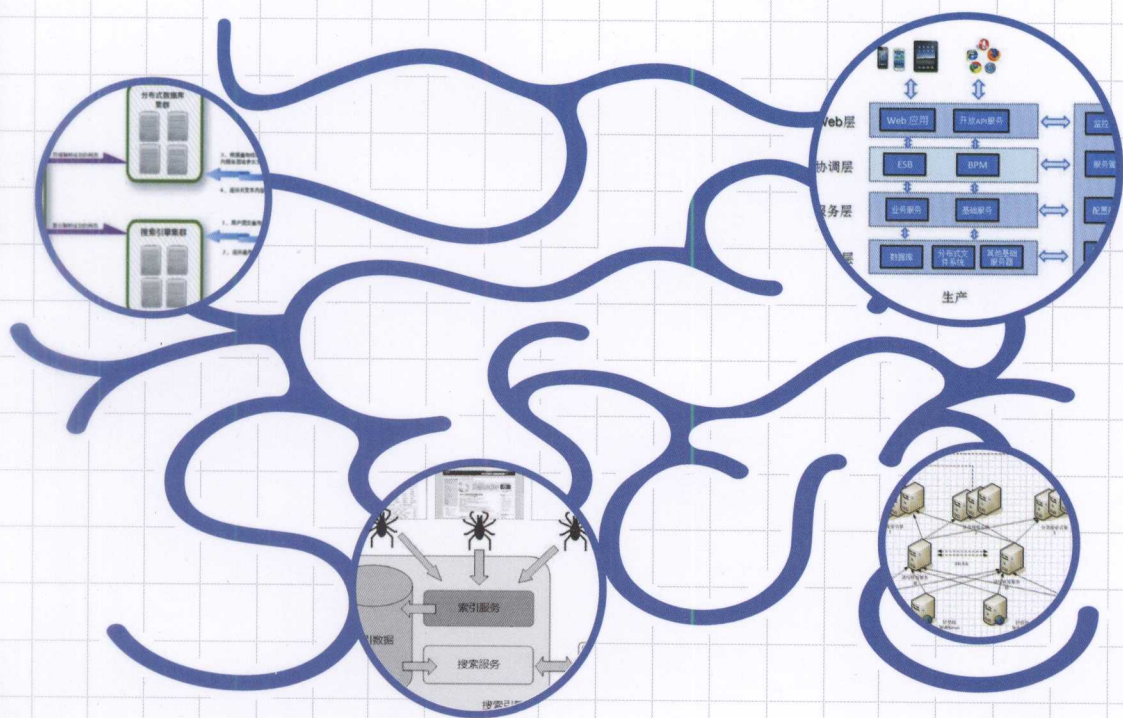


版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。



自己动手写 分布式搜索引擎

罗 刚 崔智杰 编著

- ◆ 猎兔搜索工程师数十年技术积淀与升华
- ◆ 讲解分布式搜索引擎的基本算法与原理
- ◆ ElasticSearch分布式集群实践经验的总结
- ◆ 带领读者快速入门分布式搜索引擎开发



内容简介

本书为分布式搜索引擎的权威指南，主要介绍 Java 语言开发的搜索引擎。本书共分 10 章，第 1 章介绍搜索引擎的基本概念，第 2 章介绍搜索引擎的组成，第 3 章介绍搜索引擎的索引，第 4 章介绍搜索引擎的搜索，第 5 章介绍搜索引擎的分布式，第 6 章介绍搜索引擎的集群，第 7 章介绍搜索引擎的优化，第 8 章介绍搜索引擎的部署，第 9 章介绍搜索引擎的维护，第 10 章介绍搜索引擎的测试。

自己动手写分布式搜索引擎

罗刚 崔智杰 编著

本书配套的光盘提供了相关的源代码，有的来源于作者多年开发经验积累，有的是参考算法实现，其中很多源代码可以直接用于项目开发。

本书适合需要具体实现搜索引擎的程序员使用，对于信息检索领域的研究人员也有一定的参考价值。同时，搜索引擎技术团队已经开发出来以本书为蓝本的商业软件，目前的一些分布式搜索引擎软件仍然包含很多功能。

本书由罗刚、崔智杰编写，清华大学出版社出版。

清华大学出版社 发行

地址：北京清华大学学研大厦A座

邮编：100084 电话：(010) 62770175

2017年9月第1版 2017年9月第1次印刷

ISBN 978-7-302-47000-0

定价：39.00元

010-62770175

清华大学出版社

北京

内 容 简 介

本书介绍了分布式搜索引擎开发的原理与 Java 实现, 主要包括全文检索的原理与实现、分布式算法与代码实现、SolrCloud 和 Elasticsearch 的使用与原理等内容, 并着重介绍了一种实现分布式中文搜索引擎的方法。

本书适合有 Java 程序设计基础的开发人员或者对分布式搜索引擎技术感兴趣的从业人员使用。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。
版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

自己动手写分布式搜索引擎/罗刚, 崔智杰编著. —北京: 清华大学出版社, 2017
ISBN 978-7-302-47708-2

I. ①自… II. ①罗… ②崔… III. ①搜索引擎—程序设计 IV. TP391.3

中国版本图书馆 CIP 数据核字(2017)第 162194 号

责任编辑: 杨作梅

装帧设计: 杨玉兰

责任校对: 张彦彬

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 26.75 字 数: 650 千字

版 次: 2017 年 9 月第 1 版 印 次: 2017 年 9 月第 1 次印刷

印 数: 1~3000

定 价: 59.00 元

产品编号: 075226-01

前言

搜索引擎成为人们获取信息不可或缺的工具。大数据技术的发展推动了多机集群的分布式搜索引擎技术走向成熟。普通的机器就可以搭建分布式搜索引擎。一些开源的分布式搜索引擎系统在数据存储、数据分析等方面的功能越来越强大。本书希望用通俗易懂的语言，让任何对分布式搜索引擎技术感兴趣的读者都能够有所收获。

本书的很多内容来源于搜索引擎、自然语言处理、金融等领域的项目开发和教学实践。在此感谢开源软件的开发者们，他们无私的工作丰富了本书的内容。

本书的第 1 章介绍开发分布式搜索引擎所需要的基本算法；第 2 章介绍如何从头开始自己动手写一个简单的全文检索软件包；第 3 章介绍 Lucene 的基本使用方法及其原理；第 4 章介绍使用 JSP 或者 Struts 2 开发搜索引擎用户界面，以及用户界面常用的 Taglib；第 5 章介绍 Solr 实现分布式搜索引擎的解决方案——SolrCloud，以及它对 SQL 查询的支持；第 6 章介绍如何使用基于 Lucene 的 ElasticSearch 实现分布式搜索引擎。

鉴于 ElasticSearch 处于快速发展中，一些新版本的具体使用情况可以加入 QQ 群 460405445，进行讨论。

本书配套的光盘中提供了相关的源代码，有的来源于猎兔搜索多年的开发经验积累，有的是经典算法实现。其中很多源代码都可以直接用于项目实践。

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。目前的一些分布式搜索引擎软件仍然有很多功能有待完善，作者真诚地希望通过本书把读者带入分布式搜索引擎开发的大门并认识更多的朋友。

感谢早期合著者、合作伙伴、员工、学员的支持，给我们提供了良好的工作基础。在将来，希望我们的分布式搜索引擎代码和技术能够像雨后春笋一样快速生长。

本书由罗刚、崔智杰编著，另外参与本书编写的还有张晓斐、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。

编者

目 录

第 1 章 搜索引擎.....1	3.2 创建和维护索引库..... 37
1.1 搜索引擎基本模块.....2	3.2.1 设计索引库结构..... 37
1.2 开发环境.....3	3.2.2 创建索引库..... 38
1.3 搜索引擎工作原理.....4	3.2.3 向索引库中添加索引文档..... 40
1.3.1 网络爬虫.....5	3.2.4 删除索引库中的索引文档..... 43
1.3.2 全文索引.....5	3.2.5 更新索引库中的索引文档..... 44
1.3.3 搜索用户界面.....8	3.2.6 关闭索引库..... 45
1.3.4 分布式计算.....9	3.2.7 索引的优化与合并..... 45
1.3.5 文本挖掘.....9	3.2.8 灵活索引..... 46
1.4 算法基础.....9	3.2.9 索引文件格式..... 47
1.4.1 折半查找.....10	3.2.10 定制索引存储结构..... 49
1.4.2 排序.....10	3.2.11 写索引集成到爬虫..... 54
1.4.3 最小生成树.....12	3.2.12 多线程写索引..... 56
1.5 软件工具.....15	3.2.13 分发索引..... 58
1.6 单元测试.....15	3.2.14 修复索引..... 61
1.7 本章小结.....17	3.3 查找索引库..... 61
1.8 术语表.....18	3.3.1 查询过程..... 61
第 2 章 自己动手写全文检索.....19	3.3.2 常用查询..... 64
2.1 构建索引.....22	3.3.3 基本词查询..... 65
2.2 生成索引文件.....23	3.3.4 模糊匹配..... 65
2.3 读入索引文件.....25	3.3.5 布尔查询..... 67
2.4 查询.....26	3.3.6 短语查询..... 69
2.5 有限状态机.....29	3.3.7 跨度查询..... 71
2.5.1 运算.....29	3.3.8 FieldScoreQuery..... 74
2.5.2 编辑距离有限状态机.....30	3.3.9 排序..... 77
2.6 本章小结.....32	3.3.10 使用 Filter 筛选搜索结果..... 81
第 3 章 Lucene 的原理与应用.....33	3.3.11 使用 Collector 筛选搜索结果..... 82
3.1 Lucene 快速入门.....34	3.3.12 遍历索引库..... 85
3.1.1 创建索引.....34	3.3.13 关键词高亮显示..... 88
3.1.2 查询索引库.....35	3.3.14 列合并..... 91
3.1.3 创建文档索引.....36	3.3.15 关联内容(BlockJoinQuery)..... 92
3.1.4 查询文档索引.....36	3.3.16 查询大容量索引..... 94
	3.4 读写并发..... 95

3.5 Lucene 深入介绍	95	3.10.3 PForDelta.....	176
3.5.1 整体结构.....	96	3.10.4 VSEncoding.....	178
3.5.2 索引原理.....	97	3.10.5 前缀压缩	179
3.5.3 文档值.....	100	3.10.6 差分编码	180
3.5.4 FST.....	102	3.10.7 静态索引裁剪	182
3.6 查询语法与解析.....	102	3.11 搜索中文	182
3.6.1 JavaCC	104	3.11.1 Lucene 切分原理	185
3.6.2 生成一个查询解析器.....	114	3.11.2 Lucene 中的 Analyzer.....	186
3.6.3 简单的查询解析器.....	114	3.11.3 自己写 Analyzer	188
3.6.4 灵活的查询解析器.....	114	3.11.4 Lietu 中文分词.....	191
3.7 检索模型.....	119	3.11.5 字词混合索引	191
3.7.1 向量空间模型.....	121	3.12 搜索英文	196
3.7.2 DFR.....	125	3.12.1 英文分词	196
3.7.3 BM25 概率模型	130	3.12.2 词性标注	199
3.7.4 BM25F 概率模型	136	3.12.3 原型化	201
3.7.5 统计语言模型.....	138	3.13 索引数据库中的文本	202
3.7.6 相关性反馈.....	140	3.14 优化使用 Lucene	204
3.7.7 隐含语义索引.....	140	3.14.1 系统优化	204
3.7.8 学习评分.....	141	3.14.2 查询优化	205
3.7.9 查询与相关度.....	142	3.14.3 实现时间加权排序	207
3.7.10 使用 Payload 调整相关性.....	142	3.14.4 词性标注	210
3.8 查询原理.....	146	3.14.5 个性化搜索	213
3.8.1 布尔匹配.....	147	3.15 实时搜索	213
3.8.2 短语查询.....	150	3.16 语义搜索	215
3.8.3 索引统计.....	150	3.16.1 发现同义词.....	215
3.8.4 相关性.....	152	3.16.2 垂直领域同义词.....	219
3.9 分析文本.....	155	3.16.3 同义词扩展.....	219
3.9.1 Analyzer.....	156	3.16.4 语义标注	225
3.9.2 TokenStream	162	3.17 本章小结	225
3.9.3 定制 Tokenizer.....	164	3.18 术语表	226
3.9.4 重用 Tokenizer.....	166		
3.9.5 有限状态转换.....	167	第 4 章 搜索引擎用户界面	227
3.9.6 索引数值列.....	168	4.1 实现 Lucene 搜索	228
3.9.7 检索结果排序.....	171	4.1.1 测试搜索功能.....	228
3.9.8 处理价格.....	171	4.1.2 加载索引	229
3.10 Lucene 中的压缩算法	172	4.2 搜索页面设计	231
3.10.1 变长压缩.....	172	4.2.1 Struts2 实现的搜索界面.....	232
3.10.2 Gamma	174	4.2.2 用于显示搜索结果的 Taglib.....	234

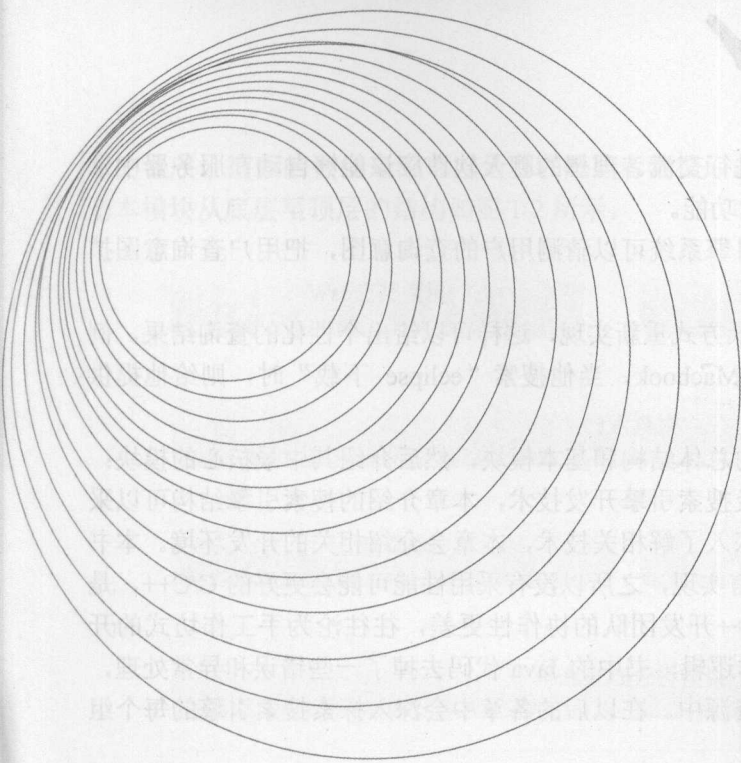
4.2.3 实现翻页.....	235	4.11.1 部署到 Web 服务器.....	290
4.3 实现搜索接口.....	238	4.11.2 防止攻击.....	292
4.3.1 编码识别.....	238	4.12 手机搜索界面.....	295
4.3.2 布尔搜索.....	241	4.13 本章小结.....	296
4.3.3 指定范围搜索.....	241	第 5 章 Solr 分布式搜索引擎.....	297
4.3.4 搜索结果排序.....	242	5.1 Solr 简介.....	298
4.3.5 索引缓存与更新.....	243	5.2 Solr 基本用法.....	299
4.4 实现分类统计视图.....	249	5.2.1 Solr 服务器端的配置与中文支持.....	300
4.4.1 单值列分类统计.....	255	5.2.2 数据类型.....	304
4.4.2 侧钻.....	256	5.2.3 解析器.....	306
4.5 实现相似文档搜索.....	257	5.2.4 把数据放进 Solr.....	307
4.6 实现 AJAX 搜索联想词.....	259	5.2.5 删除数据.....	312
4.6.1 估计查询词的文档频率.....	259	5.2.6 查询语法.....	313
4.6.2 搜索联想词总体结构.....	259	5.3 使用 SolrJ.....	313
4.6.3 服务器端处理.....	260	5.3.1 Solr 客户端与搜索界面.....	313
4.6.4 浏览器端处理.....	265	5.3.2 Solr 索引库的查找.....	315
4.6.5 拼音提示.....	267	5.3.3 分类统计.....	317
4.6.6 部署总结.....	267	5.3.4 高亮.....	319
4.7 推荐搜索词.....	268	5.3.5 同义词.....	322
4.7.1 挖掘相关搜索词.....	268	5.3.6 嵌入式 Solr.....	322
4.7.2 使用多线程计算相关搜索词.....	270	5.3.7 Spring 实现的搜索界面.....	323
4.8 查询意图理解.....	271	5.3.8 索引分发.....	331
4.8.1 拼音搜索.....	271	5.3.9 Solr 搜索优化.....	333
4.8.2 无结果处理.....	272	5.4 从 FAST Search 移植到 Solr.....	336
4.9 集成其他功能.....	272	5.5 Solr 扩展与定制.....	337
4.9.1 拼写检查.....	272	5.5.1 缺省查询.....	337
4.9.2 分类统计.....	276	5.5.2 插件.....	338
4.9.3 相关搜索.....	281	5.5.3 Solr 中字词混合索引.....	338
4.9.4 再次查找.....	284	5.5.4 相关检索.....	340
4.9.5 搜索日志.....	284	5.5.5 搜索结果去重.....	341
4.10 查询分析.....	286	5.5.6 定制输入输出.....	344
4.10.1 历史搜索词记录.....	286	5.5.7 聚类.....	348
4.10.2 日志信息过滤.....	286	5.5.8 分布式搜索.....	348
4.10.3 信息统计.....	287	5.5.9 分布式索引.....	352
4.10.4 挖掘日志信息.....	289	5.5.10 SolrJ 查询分析器.....	353
4.10.5 查询词意图分析.....	290	5.5.11 扩展 SolrJ.....	360
4.11 部署网站.....	290		

5.5.12	扩展 Solr	361
5.5.13	日文搜索	364
5.5.14	查询 Web 图	365
5.6	SolrNet	367
5.6.1	使用 SolrNet 实现全文搜索	367
5.6.2	实现原理	370
5.6.3	扩展 SolrNet	371
5.7	Solr 的 PHP 客户端	373
5.8	Solr 的其他客户端	376
5.9	为网站增加搜索功能	376
5.10	SolrCloud	377
5.10.1	Zab 协议	377
5.10.2	ZooKeeper	377
5.10.3	使用 SolrCloud	379
5.10.4	SQL 查询	380
5.11	Solr 原理	381
5.11.1	支持 Solr 的中文分词	381
5.11.2	缓存技术	383
5.12	本章小结	384

第 6 章 ElasticSearch 分布式搜索引擎 387

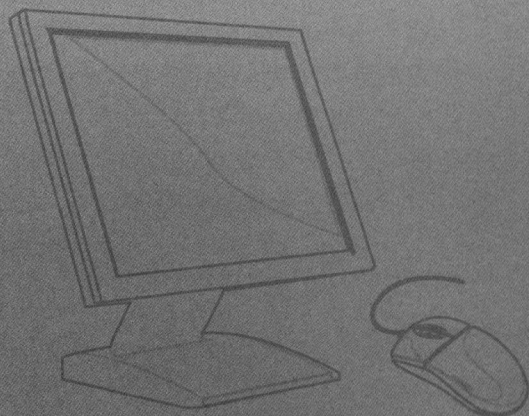
6.1	安装	389
-----	----	-----

6.2	搜索集群	390
6.2.1	Zen 发现机制	390
6.2.2	JGroups	391
6.3	创建索引	393
6.4	Java 客户端接口	396
6.4.1	创建索引	398
6.4.2	插入数据	398
6.4.3	索引库结构	400
6.5	查询	401
6.6	高亮显示	405
6.7	分页	406
6.8	中文搜索	407
6.8.1	中文 AnalyzerProvider	407
6.8.2	字词混合索引	409
6.9	分组统计	412
6.10	与爬虫集成	413
6.11	Percolate	413
6.12	权限	414
6.13	SQL 支持	415
6.14	本章小结	419



第1章

搜索引擎



每天都有很多人通过聊天软件进行交流。理想的聊天软件应该能够自动在服务器中永久记录聊天记录，而且有强大的搜索功能。

当用户只输入个别词时，搜索引擎系统可以猜测用户的查询意图，把用户查询意图扩展成整句。

搜索界面可以用 WebSocket 登录方式重新实现，这样可以给出个性化的查询结果。例如，如果已经知道某个用户在使用 Macbook，当他搜索“eclipse 下载”时，则给他提供 Mac 版本的 Eclipse 下载地址。

本章首先概要地介绍搜索引擎的总体结构和基本模块，然后介绍其中最核心的模块：全文检索的基本原理。为了尽快普及搜索引擎开发技术，本章介绍的搜索引擎结构可以采用开源软件实现。为了通过实践来深入了解相关技术，本章会介绍相关的开发环境。本书所介绍的搜索技术使用 Java 编程语言实现，之所以没有采用性能可能会更好的 C/C++，是因为 Java 代码的可维护性更好。C++ 开发团队的协作性更差，往往沦为手工作坊式的开发。另外，为了集中关注程序的基本逻辑，书中的 Java 代码去掉了一些错误和异常处理，实际可以运行的代码在本书附赠的资源中。在以后的各章中会深入探索搜索引擎的每个组成模块。

1.1 搜索引擎基本模块

一个最简单的搜索引擎由索引和搜索界面两部分组成，相对完整的搜索结构如图 1-1 所示。

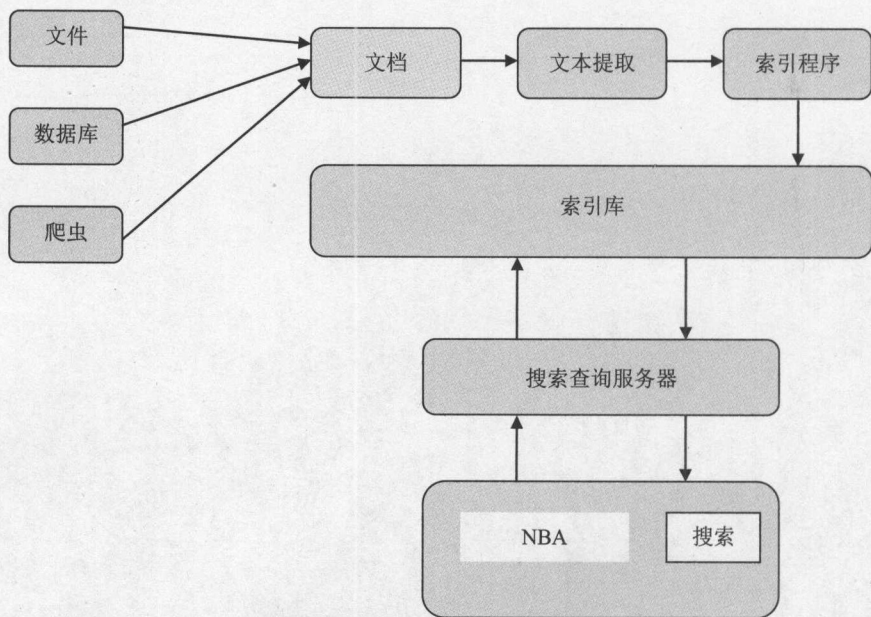


图 1-1 搜索引擎的简单结构

实现按关键字快速搜索的方法是建立全文索引库，所以最基础的程序是管理全文索引

库的程序。搜索的数据来源可以是互联网或者数据库，也可以是本地路径等。搜索引擎的基本模块从底层至顶层的结构如图 1-2 所示。

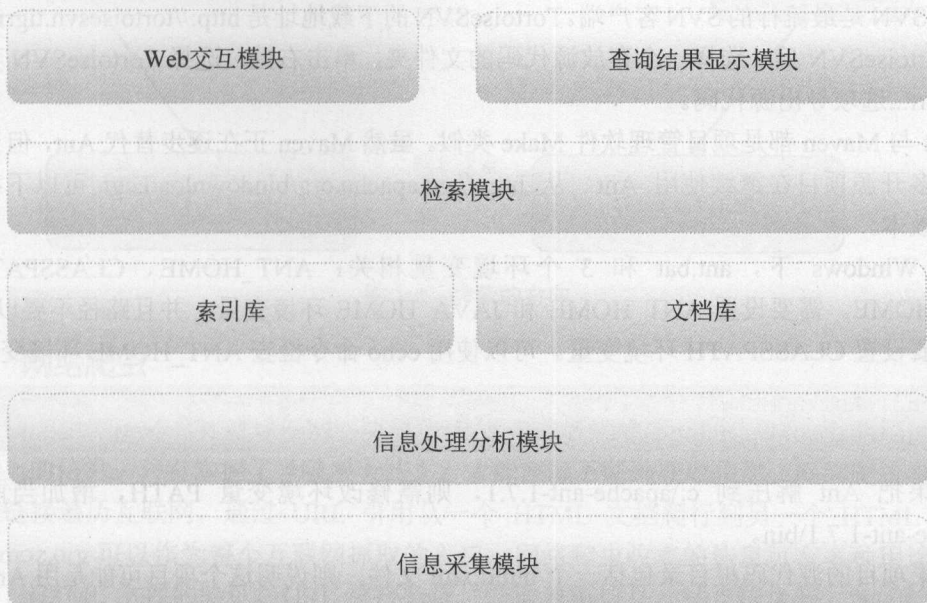


图 1-2 搜索引擎中的主要模块

1.2 开发环境

由于开源软件的迅速发展，可以借助开源软件简化搜索引擎开发工作。很多开源软件用 Java 语言编写，例如最流行的全文索引库 Lucene，所以本书采用 Java 来实现搜索。

为了实现一个简单的指定目录文件的搜索引擎，首先要准备好 JDK 和集成开发环境 Eclipse。当前可以使用 JDK 1.6。JDK 1.6 可以从 Java 官方网站 <http://java.sun.com> 下载得到，使用缺省方式安装即可。本书中的程序在附赠的资源中都能找到，可以直接导入 Eclipse 中。Eclipse 默认是英文界面，如果习惯用中文界面可以从 <http://www.eclipse.org/babel/downloads.php> 下载支持中文的语言包。

Lucene 是一个 Java 实现的 jar 包，用来管理搜索引擎索引库。可以从 <http://lucene.apache.org/java/docs/index.html> 下载到最新版本的 Lucene，当前最新的版本是 3.0。

如果需要用 Web 搜索界面，还要下载 Tomcat，可以从 <http://tomcat.apache.org/> 下载，推荐使用 Tomcat 6 以上的版本。使用开源的全文检索包 Lucene 作索引后，要把实现搜索的界面发布到 Tomcat。

对于 Web 搜索界面，建议使用 MyEclipse 开发。对于其他普通的非 Web 开发工作则不建议使用 MyEclipse。例如，开发爬虫建议只使用 Eclipse，而不要用 MyEclipse。MyEclipse 开发普通的 Java 项目时，速度慢。

Lucene 及一些相关项目的源代码由版本管理工具 SVN 管理，如果要构建源代码工程，

可以使用 Ant 和 Maven 工具。

如果需要导出 Lucene 的最新开发版本,就需要用到 SVN 的客户端。“小乌龟” TortoiseSVN 是最流行的 SVN 客户端。TortoiseSVN 的下载地址是 <http://tortoisesvn.tigris.org/>。安装 TortoiseSVN 后,选择一个存放源代码的文件夹,单击右键,选择 TortoiseSVN 菜单中的 Export...选项导出源代码。

Ant 与 Maven 都是项目管理软件 Make 类似。虽然 Maven 正在逐步替代 Ant,但当前仍然有很多开源项目在继续使用 Ant。从 <http://ant.apache.org/bindownload.cgi> 可以下载 Ant 的最新版本。

在 Windows 下,ant.bat 和 3 个环境变量相关:ANT_HOME、CLASSPATH 和 JAVA_HOME。需要设置 ANT_HOME 和 JAVA_HOME 环境变量,并且路径不要以\或/结束,不要设置 CLASSPATH 环境变量。可以使用 echo 命令检查 ANT_HOME 环境变量:

```
>echo %ANT_HOME%  
D:\apache-ant-1.7.1
```

如果把 Ant 解压到 c:\apache-ant-1.7.1,则需修改环境变量 PATH,增加当前路径 c:\apache-ant-1.7.1\bin。

如果项目的源代码根目录包括一个 build.xml 文件,则说明这个项目可能是用 Ant 构建的。大部分用 Ant 构建的项目只需要如下一个命令:

```
#ant
```

生成 jar 包。

可以从 <http://maven.apache.org/download.html> 下载最新版本的 Maven,当前最新版本是 Maven 2.2.1。解压下载的 Maven 压缩文件到 C 盘根目录下,将创建一个 c:\apache-maven-2.2.1 路径。修改 Windows 系统环境变量 PATH,增加当前路径 c:\apache-maven-2.2.1\bin。如果一个项目的源代码根目录包括一个 pom.xml 文件,则说明这个项目可能是用 Maven 构建的。大部分用 Maven 构建的项目只需要如下一个命令:

```
#mvn clean install
```

盖大楼的时候需要搭建脚手架,虽然最终不会交付使用。很多单元测试代码也不会在正式环境中运行,但是必须写。可以使用 JUnit 做单元测试。

如果参与开发的人很多,就需要一个项目管理软件,例如 Redmine 或者 JIRA。Lucene 和 Solr 的开发管理就是用的 JIRA。

搜索集群往往运行在 Linux 操作系统,需要熟悉一些 Linux 命令,例如使用 crash 命令分析 Linux 内核崩溃转储文件。

1.3 搜索引擎工作原理

一个基本的搜索包括采集数据的爬虫和索引库管理以及搜索页面展现等部分,如图 1-3 所示。

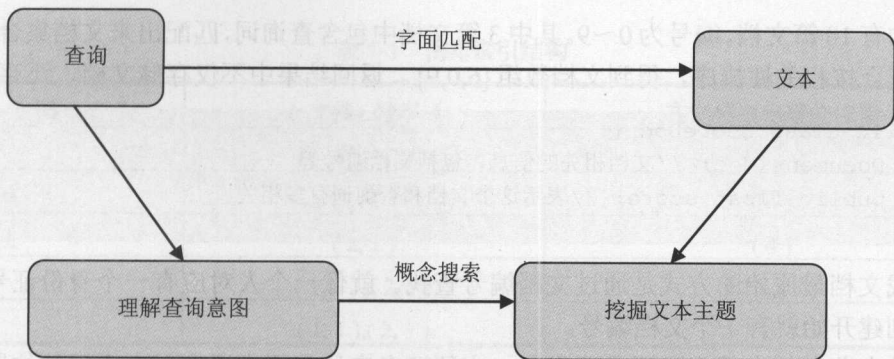


图 1-3 搜索的方向

1.3.1 网络爬虫

网络爬虫(Crawler)又被称作网络机器人(Robot)或者蜘蛛(Spider)，它的主要用途是获取互联网上的信息。只有掌握了“吸星大法”，才能源源不断地获取信息。网络爬虫利用网页中的超链接遍历互联网，通过 URL 引用从一个 HTML 文档爬行到另一个 HTML 文档。<http://dmoz.org> 可以作为整个互联网抓取的入口。网络爬虫收集的信息可有多种用途，如建立索引、HTML 文件的验证、URL 链接验证、获取更新信息、站点镜像等。为了检查网页内容是否更新过，网络爬虫建立的页面数据库往往包含根据页面内容生成的文摘。

在抓取网页时大部分网络爬虫会遵循 Robot.txt 协议。网站本身可以用两种方式声明不想被搜索引擎收入的内容：第一种方式是在站点的根目录增加一个纯文本文件 <http://www.yourdomain.com/robots.txt>；另外一种方式是直接在 HTML 页面中使用 Robots 的 <meta>标签。

1.3.2 全文索引

假如你想看书，说出你的要求后，有经验的图书管理员可以从他们的书库中直接给你推荐几本。查询的基本过程与此类似，如图 1-4 所示。

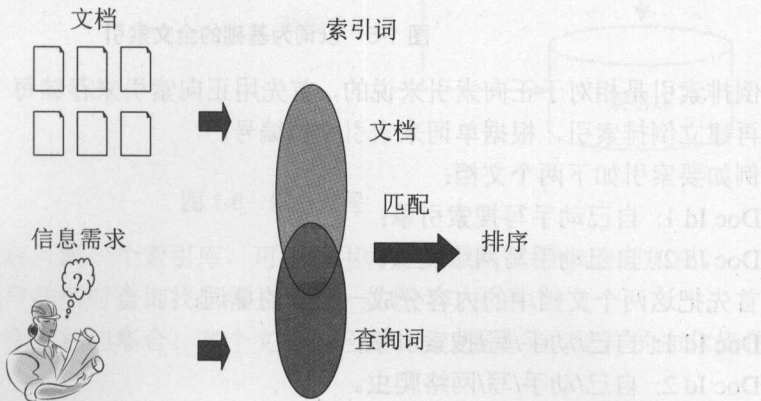


图 1-4 全文检索的基本过程

例如有 10 篇文档, 编号为 0~9。其中 3 篇文档中包含查询词, 匹配出来文档集合 {0,6,9}。对文档集合按相关性排序, 得到文档数组 {6,0,9}。返回结果中不仅存储文档, 还存储分值。

```
public class ScoreDoc {  
    Document doc; //文档相关的信息, 包括文档编号等  
    public float score; //表示这个文档和查询词有多相关  
}
```

查找文档最原始的方式是通过文档编号查找。就像一个人对应有一个身份证号, 一个文档从创建开始就有一个文档编号。

有的专业书籍末尾有名词术语索引, 方便读者定位名词术语在书中出现的位置。为了按词快速查找文档, 不是采用字符串匹配的方法在文档中查询词, 而是按词建立文档的索引。以词为基础建立的全文索引, 也称倒排索引, 如图 1-5 所示。在这里, 索引中的文档用编号表示。

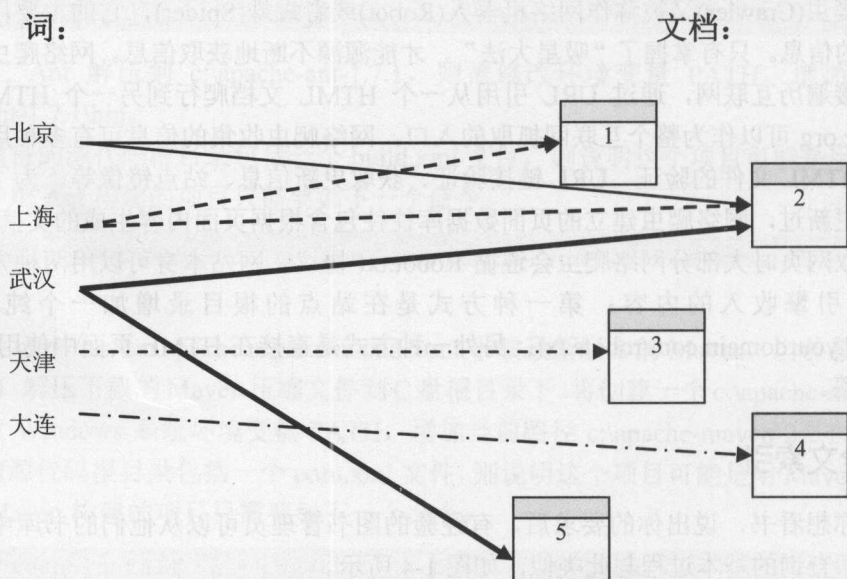


图 1-5 以词为基础的全文索引

倒排索引是相对于正向索引来说的, 首先用正向索引来存储每个文档对应的单词列表, 然后再建立倒排索引, 根据单词来索引文档编号。

例如要索引如下两个文档:

Doc Id 1: 自己动手写搜索引擎;

Doc Id 2: 自己动手写网络爬虫。

首先把这两个文档中的内容分成一个个的单词:

Doc Id 1: 自己/动手/写/搜索引擎;

Doc Id 2: 自己/动手/写/网络爬虫。

按词建立的倒排索引结构见表 1-1。

表 1-1 倒排索引结构

词	(文档, 频率)	在文档中出现的位置
动手	(1, 1), (2, 1)	(2), (2)
搜索引擎	(1, 1)	(4)
网络爬虫	(2, 1)	(4)
写	(1, 1), (2, 1)	(3), (3)
自己	(1, 1), (2, 1)	(1), (1)

每个词后面的文档编号(docId)列表称为投递列表(posting list)。除了在投递列表中记录文档编号外，还可以添加词频和位置信息。词频的添加有助于结果的排序。位置信息记录了一个索引词在文档中出现的位置，可以用于包含多个查询词的短语检索；此外，也可以用于快速高亮显示查询词。

索引子系统把要搜索的文档先建立好索引，如图 1-6 所示。

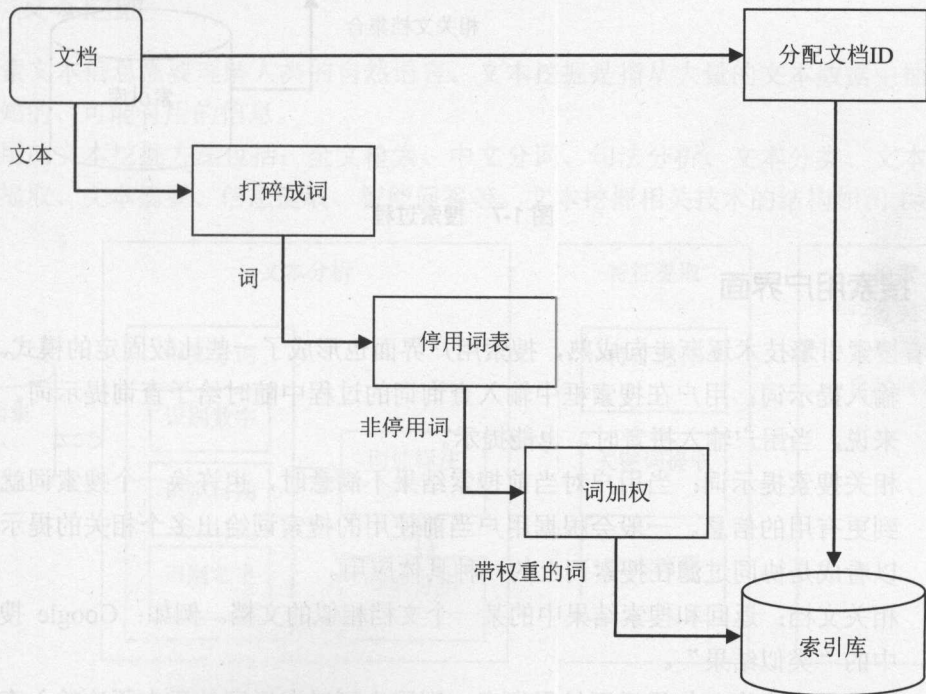


图 1-6 索引过程

待查询的很多文档放入同一个索引库，可以使用特征函数对文档中的词加权。搜索子系统针对用户的即时查询找出相关文档，如图 1-7 所示。每个词查询出来一个文档的集合。多个文档集合做 AND 或者 OR 这样的布尔操作。

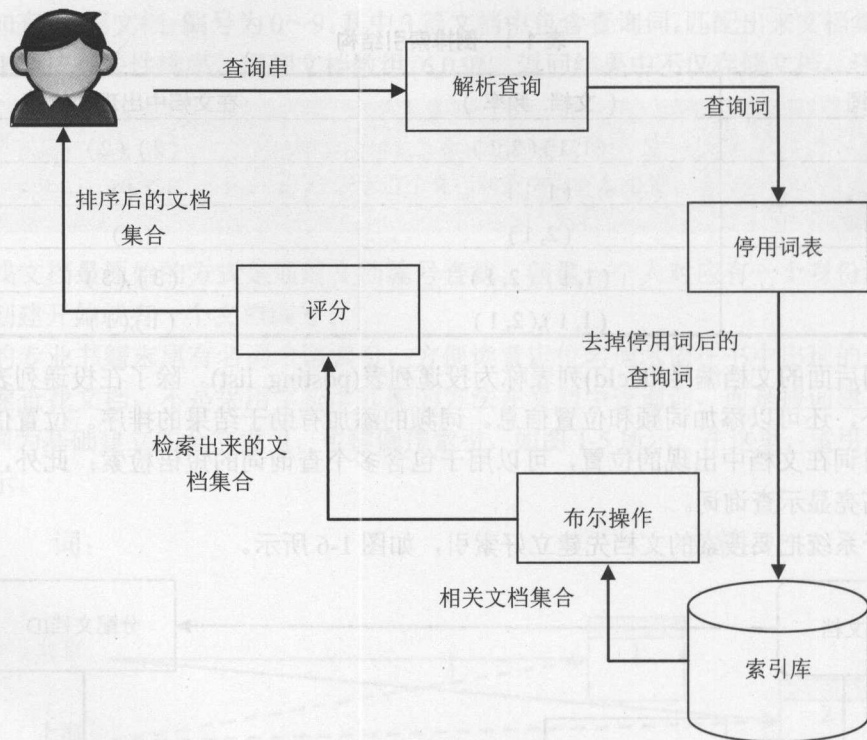


图 1-7 搜索过程

1.3.3 搜索用户界面

随着搜索引擎技术逐渐走向成熟，搜索用户界面也形成了一些比较固定的模式。

- 输入提示词：用户在搜索框中输入查询词的过程中随时给予查询提示词。对中文来说，当用户输入拼音时，也能提示。
- 相关搜索提示词：当用户对当前搜索结果不满意时，也许换一个搜索词就能够得到更有用的信息。一般会根据用户当前使用的搜索词给出多个相关的提示词。可以看成是协同过滤在搜索词上的一种具体应用。
- 相关文档：返回和搜索结果中的某一个文档相似的文档。例如：Google 搜索结果中的“类似结果”。
- 在结果中查询：如果返回结果很多，则用户可以在返回结果中再次输入查询词以缩小查询范围。
- 分类统计：返回搜索结果在类别中的分布图。用户可以按类别缩小搜索范围，或者在搜索结果中导航。有点类似于数据仓库中的向下钻取和向上钻取。
- 搜索热词统计界面：往往按用户类别统计搜索词，例如，按用户所属区域或者按用户所属部门等，当然也可以直接按用户统计搜索热词，如 Google 的 Trends。

搜索界面的改进都是以用户体验为导向。所以搜索用户界面往往还根据具体应用场景优化。所有这一切都是为了和用户的交互达到最大的效果。