基于 Open Source 的全文检索框架

张以维 （总后指挥自动化站 北京 100842）

摘要：本文在开源(Open Source)项目 Jakarta Lucene 的基础上，结合多种开源文档格式分析工具，设计和实现了一种可扩展的全文检索框架，该框架可高效地对 XML、HTML、MS Word、PDF 等格式的文档进行全文检索。整个框架完全基于开源工具包，可以有效地对信息系统的开发进行支持。

关键词：全文检索 Lucene 设计模式 程序框架

1 引言

全文检索提供面向全文的数据检索功能，它可以用原文中任何有意义的字或词作为检索条目，提取原文。随着信息化建设水平的进一步提高，信息系统中积累的数据不断膨胀，这些数据基本上可以分为两类：结构化的数据和非结构化的数据。同数据库中的数据查询技术相比，全文检索可以更好地处理非结构化的数据，例如文本数据、图形文件等多媒体数据。在 Internet 日益普及的今天，全文检索的应用领域更加广泛，例如搜索引擎、桌面搜索等。本文基于开源的全文检索核心工具包 Lucene，设计与实现了一种可以对多种文件格式进行高效全文检索的程序框架，可以有效地支持 XML、Word、PDF、RTF、HTML 等格式，在增加对新的文本格式支持时，该框架具有良好的可扩展性。

2 全文检索和 Lucene

Lucene 是一个开放源码的全文信息检索工具包，作为 Apache 基金会下的一个开源项目，越来越来的应用以其作为后台的全文检索引擎。除了最成熟的 Java 语言版本，Lucene 还推出了 C、Delphi 等语言的版本。Lucene 全文检索与数据库中的模糊查询不同，前者的性能更高，并能够进行相似度计算和去重操做[1]。实现全文检索的过程分为两步：首先必须建立索引，然后才能进行检索。建立索引是一个非常重要的过程，如果要从大量的文档中检索包含某个关键词的文档，如果没有索引就需要把这些文档顺序地读入内存，然后逐个分析是否含有要查找的关键词，将会耗费大量的时间；而通过使用索引，搜索引擎能够在毫秒级的时间里查找到需要的结果。

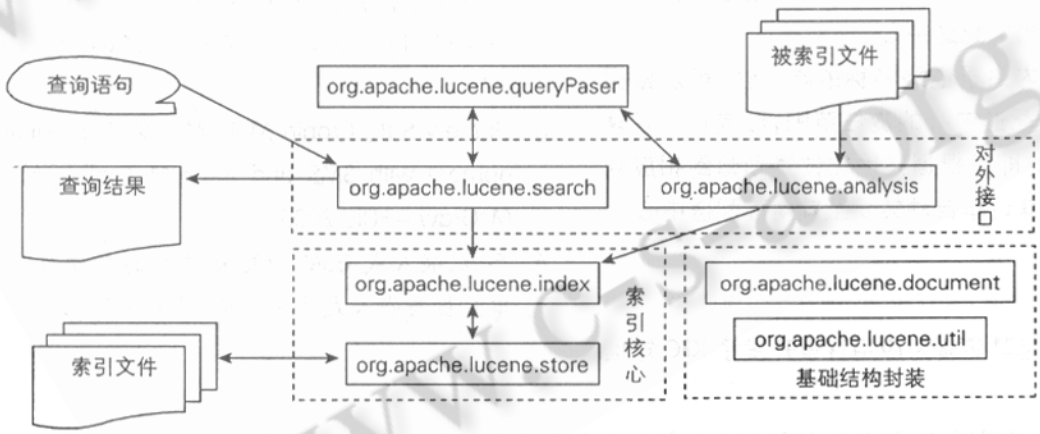

图 1 Lucene 的工作机制

Lucene 在建立索引的过程中，使用的是一种称为

反向索引(inverted index)的机制[2]。在这种机制中，维护着一个词/短语表，对于该表中的每个词/短语，都通过一个链接表来描述哪些文档包含了这个词/短语。

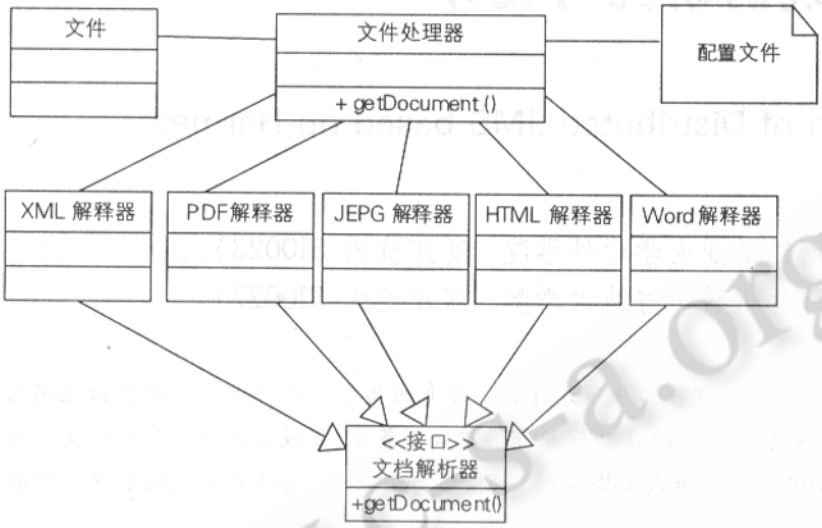

图 2　可扩展的全文检索框架

Lucene 在对文档建立好索引后，就可以通过索引进行检索了。搜索引擎首先会对搜索的关键词进行解析，然后再在建立好的索引上面进行查找，最终返回和用户输入的关键词相关联的文档。整个过程如所图 1 示[3]，涉及的主要功能调用：封装文本文档、文档分词、建立索引、检索索引，均由 Lucene 工具包中相应的 Java 包(Package)提供。

3　全文检索框架

Lucene 根据纯文本文档建立索引文件和全文检索的效率非常高，它的出现极大地推动了全文检索技术在各个行业的应用[4]。但是它仅仅是一个核心库，无法满足实际应用中对多种文件格式建立全文检索的需求。Lucene 具备非常优秀的面向对象设计的优点，每一个被添加到索引库中的文件都要用一个文档对象(Document)来表示，在图 2 中针对每一种文件格式都创建了对应的文件解释器(在 Java 编程中实际是一个类)，这些文件解释器实现了共同的接口，对文档进行处理，统一转换为 Lucene 可以处理的文档对象。以广泛使用的 PDF 文档为例，在对应的 PDF 解释器类中使用 PDFBox 开源库，提取文本内容、文本元数据，最终构造成一个 Lucene 的文档对象。

在上图所示的可扩展全文检索程序框架中，各种文档解释器作为子系统实现一种具体的功能，为了向应用程序提供了一个简单的接口，根据设计模式中的 Fa? ade 模式，设计了文件处理器类。作为框架的核心，文件处理器根据文件的后缀名调用相应的文档解释器，处理机制如下述代码所述：

```
Public Document getDocument (File file) {
    Properties props = new Properties( ); //开始载入配置文件
    props. load ( new FileInputStream (handle. properties) ;
    String ext = getfileExt( file) ; //得到文件扩展名
    String handlerClassName = props. getProperties( ext) ;
Class handlerClass = Class. forName ( handlerClassName) ;
    DocumentHandler handler =
        ( DocumentHandler ) handlerClass. newInstance( ) ;
    //根据配置文件获得文档处理类
    Return handler. getDocument ( new FileInputStream( file) ) ;
}
```

在这段代码中，程序根据配置文件中指定的某种扩展名的文件与其解释器类的对应关系，实例化一个解释器对象，然后对文件进行处理，最后返回一个可由 Lucene 建立索引的文档对象(Document)。

4　文件解释器的构建

在实际应用中，存在多种类型的文件，例如 PDF、Word、HTML、XML、RTF 等，它们都有各自独立的格式和元数据。为了提取这些文本内容，在框架的实现中，利用已有的开源软件包进行开发是一种非常好的解决方案[5]。

(下转第 61 页)

(上接第 57 页)

表 1 文件解释器开源工具包

文件类型	开源软件包	资源地址
PDF	PDFBox	http://www.pdfbox.org/
WORD	POI	http://jakarta.apache.org/poi/
HTML	JTidy	http://jtidy.sourceforge.net/
XML	Xerces2	http://xml.apache.org/xerces2-j/index.html

表 1 描述了对常见的文件类型进行解释的开源软件包名字和它们的 Web 地址，在这些软件包中都提供了强大易用的编程接口对文件进行处理，并且针对一种格式的文件有几种不同的工具包可以进行选择。

5 结束语

本文提出了一种基于开源软件工具包 Lucene 的全文检索框架，这种框架具有良好的扩展性，方便增加对各种文件格式的支持，并且通过配置文件进行配置，不涉及对用户应用程序的修改。此外在框架中，使用开源工具实现文件解释器，有效的利用了已有资源，增加了开发效率，因而本文描述的全文检索框架可以有效地支持信息系统的开发。

参考文献

1 郎小伟、王申康，基于 Lucene 的全文检索系统研究与开发，计算机工程，2006-02.

2 Cutting D. The Lucene Search Engine Powerful Flexible and Free: JavaWorld [M]. John Wiley Sons, 2000-09.

3 潘以锋，基于 Lucene 的网站全文检索系统的开发，广西教育学院学报，2006-05.

4 李刚，征服 Ajax + Lucene 构建搜索引擎[M]，北京：人民邮电出版社，2006-04.

5 Erik H, Otic G. Lucene in action. Manning Publications Co, 2005-09.