基于 Solr 的机构知识库检索系统构建研究

杨 洁

华中师范大学信息技术系　武汉 430079

[摘要] 介绍 Solr 的概念、特性以及体系结构，并使用它设计和构建机构知识库的检索系统，该系统初步具备简单检索、高级检索、分面检索、相似资源检索、访问统计等功能，，实现了机构知识库的个性化检索。

[关键词] Solr　机构知识库　检索系统

1 引　言

伴随着计算机技术、网络技术以及数字化技术的迅速发展，数字化资源的发布和共享不再受时间、空间的限制。在研究机构和大学里，由研究人员和教师通过网络收集、保存、处理的一些会议论文、期刊论文、专著、教学课件、声音动画等数字化资源成为重要的学术资源。研究机构和大学将这些重要的学术资源从分散存储在研究人员和师生员工等的计算机上集中起来构成了机构知识库（Institutional Repository, IR)。机构知识库通过校园网甚至校际之间的协议，得到开放利用，一些发达国家的大学图书馆使用这种机构知识库共享学术资源。与此同时，机构知识库也承载了学术传播、电子出版、场次保存、知识管理、促进教育、科研评价、共享利用等诸多功能[1]。但是在互联网信息技术高速发展的今天，人们更加关注如何以最快的时间获取最有价值的信息。传统仅基于关键字的检索已不能满足人们现在的要求，如何能以最快的时间在机构知识库中找到自己想要的资源是现今亟需解决的问题。

如今，开源软件本着自由、共享的理念，作为一种新兴的软件模式正在迅速深入人心。开源软件能够使开发者根据自身的需求进行二次开发来实现功能定制，从而提高其创新能力和针对性服务能力[2]。为了能够结合用户的需求提供多层次、高性能、多方面的信息服务，应充分利用一些优秀的开源软件。

本文提出了基于Solr的机构知识库检索系统的模型，实现了对机构知识库的高效查询和浏览并提供了相似资源的推荐。

2　Solr 概述

2.1 Solr 的概念及特性

Apache Solr 是一个开源的基于 Lucene 的搜索服务器。它使用 Java 语言开发，主要基于 HTTP 和 Apache Lucene 来实现，在全文索引工具 Lucene 的基础上进行了封装和功能扩展。Solr 提供了分面搜索、高亮显示等功能并且支持多种输出格式（XML 和 JSON 格式)。它是一个较为稳定和成熟的全文检索服务器，易于安装和配置，而且有自己独特的管理界面，是一个高性能的、可独立运行的企业级全文搜索引擎服务器[3]。

Solr 的特性主要包括：①高级的全文搜索功能，高亮显示检索结果；②专为高通量的网络流量进行优化；③基于开放接口（XML 和 HTTP）的标准；④综合的 HTML 管理界面；⑤具有很强的可伸缩性，能够

有效地被复制到另外一个 Solr 搜索服务器；⑥使用 XML 配置达到灵活性和适配性，并具有可扩展的插件体系。

2.2 Solr 体系架构

Solr在Lucene的基础上，重在数据之间内在关联关系的挖掘。作为一个完整的全文检索服务系统，Solr具有良好的体系架构[4]，如图1所示：

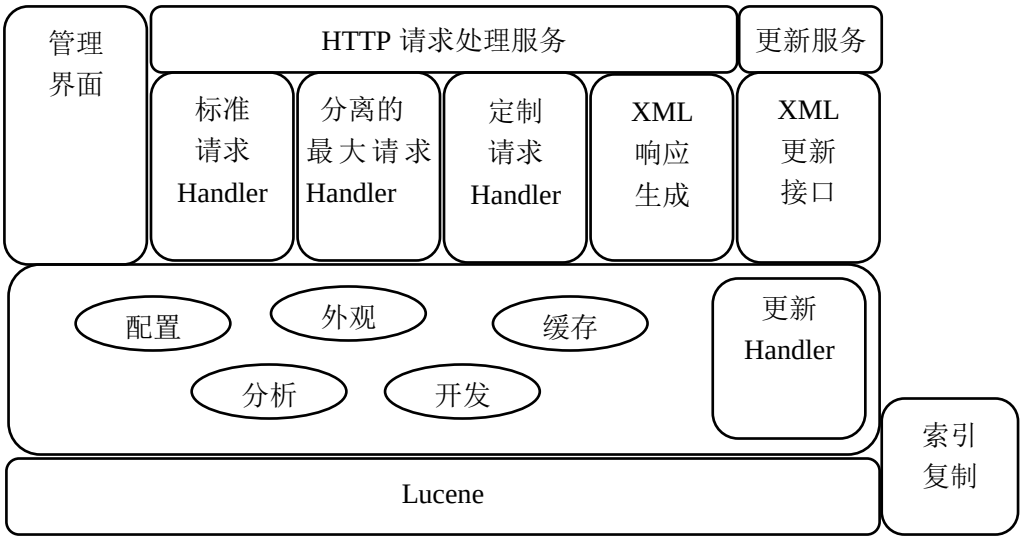

图1 Solr体系架构

上层主要包括管理员界面(Admin Interface)、索引更新处理器(Update Servlet)和HTTP请求处理器(HTTP Request Servlet) 三大模块。管理员、用户和其他系统通过HTTP接口，向Solr发送HTTP请求，HTTP请求处理器根据接受到的不同请求，分析要使用的不同SolrRequestHandler，然后通过中间层即Solr的核心层处理这些请求，并以XML、JSON 等格式返回请求结果。索引更新处理器主要为XML数据的导入提供相应的可视化界面。

中间层为Solr的核心层，由多个独立模块组成，负责整个系统配置(Config) 和索引参数(Schema)的加载与解析，索引文档及查询请求的分析(Analysis)，提供建立索引和读取索引的并发控制(Concurrency)和分面、文档缓存机制(Caching) 。更新处理器(Update Handler) 负责对XML、CSV 和数据库等来源的索引请求进行处理。

底层为全文索引工具Lucene，负责具体的文本分析、创建索引，并对索引文件进行高效查询。此外，索引复制功能(Replication) 是一个独立的模块，可以通过脚本程序、异步处理程序等完成，用于支持分布式索引和检索。

3 基于 Solr 的机构知识库检索系统的设计

3.1 系统总体功能

基于高性能的 Solr 构建机构知识库检索系统，可以对机构知识库进行深度开发和综合利用，为用户提供高效、稳定的检索服务平台。该系统的功能结构如图 2 所示：

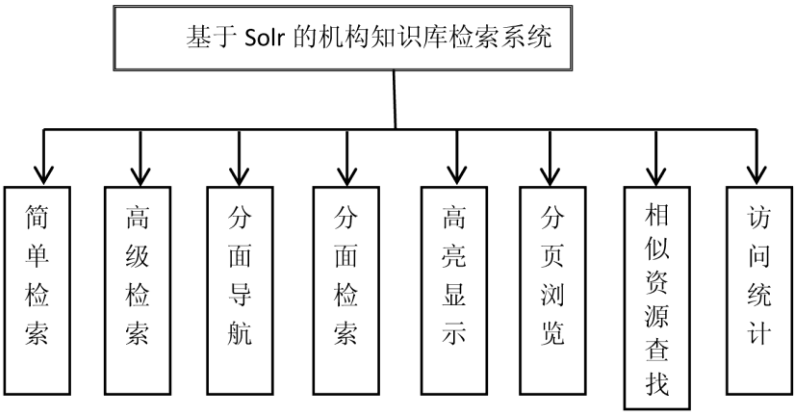

图 2　系统总体功能

该系统的主要功能分为 8 个部分。其中，简单检索指依据用户需求，提供所有字段的检索；高级检索是指对资源的作者、题名、关键字、出处等进行多入口的组合检索；分面导航是从论文、专著、教学课件、其他资源这 4 个维度对检索结果进行聚类统计分析；分面检索是用户依据分面导航进入相关链接时，系统将对当前结果进行分面过滤，从而实现分面检索[5]。另外，为了实现系统和用户之间的高度友好，系统提供检索结果的高亮显示和分页浏览功能；相似资源查找则是为用户提供更多有用的资源，在为用户显示检索结果的同时提供相似资源查找的功能；访问统计是系统为了帮助管理者更好地了解机构知识库被访问和利用的情况提供的资源访问的统计功能。

3.2　系统体系架构

基于 Solr 的机构知识库检索系统的体系架构，如图 3 所示：

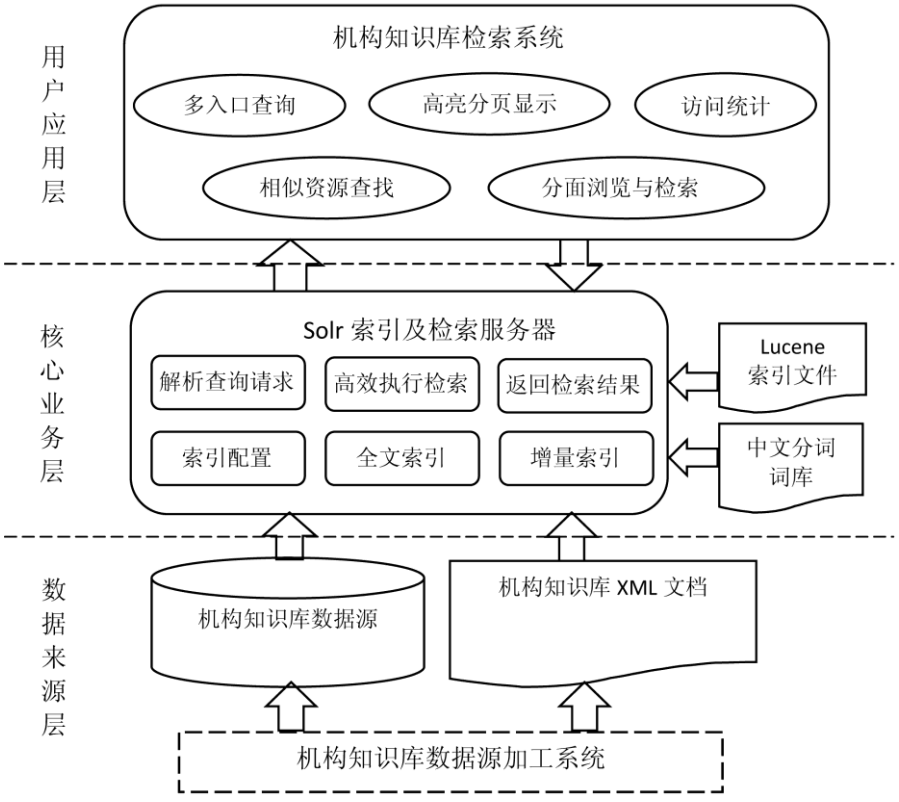

图 3　系统体系架构

用户应用层是该系统中开发的重点，主要是基于 Solr 进行的二次开发，所完成的工作是提供用户界面，实现系统与用户间的友好交互，对于用户输入的查询条件进行高效灵活的检索，并且对检索的结果进行动态的分面导航，实现分面检索。另外，该层还提供检索结果的高亮显示、分页浏览以及相似资源的查找。最后，该层的一个特色功能就是提供知识库资源的访问统计，如哪些资源是用户比较感兴趣的、哪些资源是引用较多的、哪些时间段访问的人数比较多等。这样可以促进管理者更好地了解机构知识库的利用情况，以便改善。

核心业务层是联系用户应用层和数据来源层的核心纽带，该层一方面利用全文搜索引擎服务器 Solr 所提供的强大的功能，接收来自数据来源层的数据，进行系统参数以及索引参数的动态配置和管理，然后通过中文分词器和词库进行全量和增量索引，并且生成 Lucene 索引文件。另一方面，该层接收来自用户应用层的用户查询请求进行解析，高效、稳定地执行检索过程，并且将检索结果以标准的 XML 或 JSON 格式返回。

数据来源层是该系统的最底层，主要负责对初始数据的处理。该系统在获取来源数据时主要支持两种方式：一是直接来源于机构知识库的数据加工系统（主要负责将资源按照一定的标准进行规范化存储）；二是将机构知识库的数据加工系统中的数据转化为规范的 XML 文档。藉此，可以最大程度地实现数据处理的独立性，使得本系统可以在加工系统之外建立索引。

4 基于 Solr 的学科机构知识库检索系统的构建

4.1 Solr 的安装部署

Apache Solr 是开源的搜索服务器，本文拟在安装 JDK 和 tomcat 的基础上，进行 Apache Solr 的本地化安装、配置与测试。

首先需要在官方网站下载 JDK、tomcat 以及 Solr 的安装包。成功安装 JDK 和 tomcat 之后，重要的工作就是正确配置 Solr 的相关参数，实现 Solr 的本地化安装。其中需要做的工作主要有：①将下载的 Solr 包下面的 dist 文件夹中的 apache-solr-1.4.1.war 拷贝到 tomcat 的 webapps ，并且改名为 solr.war。该 WAR 文件是一个完整的 web 应用程序，包括了 Solr 的 Jar 文件和所有运行 Solr 所依赖的 Jar 文件、Jsp 和很多的配置文件与资源文件。②新建/opt/solr-tomcat/solr 文件夹，把下载的 solr 包中的 example/solr 文件夹下面的所有文件放入 /opt/solr-tomcat/solr。③配置添加 solr.home 环境变量。在 tomcat 的 conf 文件夹建立 Catalina 文件夹，然后在 Catalina 文件夹中再建立 localhost 文件夹，在该文件夹下面建立 solr.xml,代码如下：

```
<Context docBase="/usr/local/tomcat6/webapps/solr.war" debug="0" crossContext="true" >
    <Environment name="solr/home" type="java.lang.String" value="/opt/solr-tomcat/solr" override="true" />
</Context>
```

为验证Solr是否正确安装，可以访问Solr的管理界面http://localhost:8080/solr/admin/，如果出现Solr系统管理主界面则表示配置成功，如图4所示：

图 4　Solr 系统管理主界面

4.2 搜索引擎的配置

为了能够满足该系统对索引的需求，首先通过 solrconfig.xml 和 schema.xml 两个文件对索引性能参数和索引结构进行设置。schema.xml 文件是数据表配置文件，定义了加入索引的数据类型。主要包括 types、fields 和其他的一些缺省设置。在定义 fieldType 时，通常需要指定该类型的数据在建立索引和查询时需要用的分析器 Analyzer。该系统采用开源的 mmseg4j 中文分词器[6]，实现代码如下：

```
<types>
……
<!--mmseg4j field types-->
<fieldType  name="textComplex" class="solr.TextField" positionIncrementGap="100" >
<analyzer>
<tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="complex"
dicPath="/opt/solr-tomcat/solr/dic"/>
<filter class="solr.LowerCaseFilterFactory"/>
</analyzer>
</fieldType>
<fieldType name="textMaxWord" class="solr.TextField" positionIncrementGap="100" >
<analyzer>
<tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="max-word"
dicPath="/opt/solr-tomcat/solr/dic"/>
<filter class="solr.LowerCaseFilterFactory"/>
</analyzer>
</fieldType>
<fieldType name="textSimple" class="solr.TextField" positionIncrementGap="100" >
<analyzer>
<tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="simple"
dicPath="/opt/solr-tomcat/solr/dic"/>
```

```
<filter class="solr.LowerCaseFilterFactory"/>
</analyzer>
</fieldType>
……
</types>
```

在定义好相关字段类型之后，便可以通过提交 XML 等规范文档或连接数据库进行数据索引的构建。

4.3 索引的管理

该系统中索引的构建主要通过两种方式：一是基于 XML 文件的索引构建；二是基于数据库的索引构建。

● 基于 XML 文件的索引构建主要通过使用 4 种请求命令，首先要在 \apache-solr-1.4.1\example\exampledocs 目录下创建 mmseg4j-solr-demo-doc.xml 文档，样例数据如：

```
<add>
    <doc>
        <field name="id">1</field>
        <field name="title">面向 3G 手机的移动学习资源交互设计与实现</field>
        <field name="author">刘清堂</field>
        <field name="author">向丹丹</field>
        <field name="keyword">移动学习</field>
        <field name="keyword">3G</field>
        <field name="type">论文</field>
        <field name="abstract">移动学习是网络教育的最新发展，3G 手机被视为最具有发展前途的移动学习终端设备，面向 3G 手机的移动学习……</field>
        <field name="from">中国电化教育</field>
        <field name="year">2011</field>
    </doc>
</add>
```

然后在命令窗口切换到“<apache-solr-1.4.0\example\exampledocs>”，运行 post.jar 命令代码：

```
java -Durl=http://192.168.10.85:18080/solr/update -Dcommit=yes -jar post.jar mmseg4j-solr-demo-doc.xml
```

，若出现“Commintting Solr index changes”，则表明索引构建成功。

● 基于数据库的索引构建，首先要将对应数据库的 JDBC 驱动拷贝到 tomcat 下的 webapps\solr\WEB-INF\lib 目录下，接着配置数据库文件 db\data\config.xml 和 solrconfig.xml，使其与 DataImportHandler 进行映射即可。DataImportHandler 可以通过 Solr 提供的全量索引和增量索引两种方式直接从数据库获取数据来构建索引。

4.4 用户模块的设计与实现

用户模块的功能比较简单，主要包括页面设计显示以及分页辅助等工作。在设计用户界面时，需要保持美观大方、简洁友好的风格。

主页面的设计如图 5 所示：

图 5　用户模块主页面

当用户进入该系统时，默认提供的是简单检索的服务，用户可以依据自己的需求输入需要查询的内容进行查询。另外，还提供了高级检索、分面检索、相似资源检索以及访问统计的导航以方便用户进行其他方面的检索。

5　结　语

如今，利用优秀的开源软件进行二次开发和并且依据个性化需求进行适应性的加工已经成为一种趋势。Solr 作为一种开源的全文检索引擎应用到数字化资源服务的研究也备受关注。本文将 Solr 应用于机构知识库的检索系统，取得了较为理想的效果。该系统目前还存在一些问题，如检索结果不能重新排序、不能实现分布式检索等，下一步将要针对这些问题进行改进和完善，以提高系统的实用性和整体性能。

参考文献：

[1] 赵继海.机构知识库：数字图书馆发展的新领域[J].中国图书馆学报，2006,32(2):33-36,50.
[2] 鲜国建，赵瑞雪. 基于 Solr 的中文农业期刊文摘检索系统的构建研究[J].现代图书情报技术，2011(6):51-58.
[3] Apache Solr[EB/OL]. [2011-09-10].http://lucene.apache.org/solr/.
[4] 陈波.基于开源全文检索系统 Solr 的 OPAC 分面浏览[J].现代图书情报技术,2007(11):72-75.
[5] Goeschl S. SOLR: An open source enterprise search[EB/OL].[2011-03-22]. http://people.apache.org/-sgoeschl/ presentations/solr/index.html.
[6] mmseg4j - MMSEG for Java Lucene Chinese Analyzer,or for Solr-Google Project Hosting[EB/OL].[2011-04-22]. http://code.google.com/p/mmseg4j/.

[作者简介] 杨 洁,女，1986 年生，硕士生在读，发表论文 1 篇。