● 王思丽[1,2]，马建玲[1,2]，姚晓娜[1,2]
（1. 中国科学院　国家科学图书馆兰州分馆，甘肃　兰州　730000；2. 中国科学院　资源环境科学信息中心，甘肃　兰州　730000）

开放知识资源分类检索机制优化研究*

摘　要： 针对传统分类检索技术存在的局限性，结合开放知识资源的分类检索需求，对开放知识资源的分类检索机制进行了优化研究，包括分类索引配置的优化、分类检索过程的优化、分类检索访问的优化和客户端分类检索功能的实现。最终将研究方案应用于开放知识资源项目建设，提高了开放知识资源分类检索的响应速度和效率。

关键词： 开放知识资源；分类检索；优化

Abstract: Based on the limitations of the traditional classification retrieval technologies, and combined with the classification retrieval demands of open knowledge resources, this paper researches on the optimization of classification retrieval mechanism of open knowledge resources, including optimization of classification indexing allocation, optimization of classification retrieval process, optimization of classification retrieval access, and realization of client classification retrieval function. Finally, the research scheme is applied to the construction of open knowledge resources project, which improves the response speed and efficiency of classification retrieval of open knowledge resources.

Keywords: open knowledge resources; classification retrieval; optimization

开放知识资源主要是指数字化科学研究过程中所涉及的科学数据、种质资源、教学资源、软件、计划与项目、学术组织、学术会议等各类非文献资源[1]。目前，中国科学院国家科学图书馆针对中国科学院数字化科研创新中对各类综合科技资源的需求，已将建立综合数字资源服务体系建设任务列入“十二五”战略规划布局，正在积极推进针对开放知识资源建立系统化发现、规范化遴选、知识化描述与组织、集成化揭示的开放知识资源服务体系，并提供相应分类检索和浏览服务。

笔者在参与开放知识资源项目建设的过程中发现，原有系统的分类浏览导航技术主要采用 Dtree 树形菜单组件技术[2]。该组件使用 JavaScript 编写，优点是简单灵活，支持从关系型数据库动态引入数据，只需少量改编原有代码即可配置应用，不用添加任何页面而可以直接用代码实现多个栏目，同时支持框架或非框架的页面，并能在不同页面间记录当前状态，实现形式上的无限分级。虽然在前期的应用中，基于 Dtree 的分类浏览技术未见不妥，但 Dtree 太过于轻量级，随着项目登记数据量的不断增加以及大量采集与收割数据的集成，开放知识资源需要分级的类别越来越多，越来越细，Dtree 每一次分级的导航数据都需直接和数据库进行交互，频繁地直接访问和读取数据，给服务器造成了很大的压力，使得页面浏览和检索的响应速度并不是很理想。同时 Dtree 只是提供了灵活分级的机制，并没有提供对分级栏目命中结果数据的获取接口，无法满足用户对项目后期的改进需求。因此，考虑到 Solr 高效灵活的缓存和索引机制对海量数据的良好支持和分面查询检索的快速响应性能，本文采用 Solr 进行了功能改进和索引升级，优化了开放知识资源的分类检索机制，为进一步提供更为优质的知识组织服务奠定基础。

1　分类检索机制

1.1　分类检索的概念

分类检索，又称分面检索、分面浏览、分类导航等，是信息检索的一个分支[3]。分类检索主要是指将各种不同的信息资源或概念按照一定的分类法，如学科、主题或其他专业标准等进行知识重组，并通过等级与层次的直观显示，为用户浏览和获取信息提供知识导航服务[4-5]。在分类检索中，用户通过选择有效的限定条件，在结果集中逐步缩小范围直到获取所需信息。其主要特点是：限定条件由用户任意选择组配，保证永远不会检索不到结果，同时能够动态地多维度提示方便用户获取最相关的结果。分类检索的结果是上下文相关的，用户选择某个条件后，分类检索引擎会在该条件限定下的结果集中动态获取，再次从

* 本文为中国科学院文献情报能力专项基金项目“开放知识资源登记系统（二期）”的研究成果，项目编号：Y300051001。

不同的角度对数据进行重新归类整合，帮助用户进一步过滤他们需要获取的资源信息。总的来说，分类检索机制有一定的层次性和系统性，可以帮助用户快速地查找和定位到相关类目的全部资源，避免因输入关键词不充分而引起漏检误检，提高了知识获取的查全率和查准率。

1.2 传统分类检索技术的局限性

一个良好的分类检索机制，不仅要能分类，分类体系可视化，而且要能够快速响应分类请求，缩短分类时间，准确返回分类结果。传统的分类检索处理方法大多采用一些轻量级的组件技术，如 Dtree，jQuery Plugin，TreeView 等进行二次开发或实例化后实现，甚至有的不采用任何组件技术，直接根据分类体系的元数据描述规范，仅仅实现形式上的、视觉上的分类检索。这些分类检索技术在本质上来讲，都是依赖于客户端程序与关系数据库的直接交互，且大多属于单一分类类目检索，在数据量越来越大，分类类目越来越多，分类层次越来越细化的情况下，远远不能满足用户期望多类目交叉分类、快速响应分类检索请求、及时详细返回分类检索结果的需求。而且传统的分类检索技术还会对数据库服务器造成很大的负载，频繁的分类检索会拖慢服务器的运行速度，甚至会造成内存溢出等现象，影响整个系统的正常运行和知识服务能力。

1.3 分类检索技术的发展优化

目前，针对传统分类检索技术存在的不足，取而代之的是以 Solr 技术为核心的分类检索机制的优化发展与应用。Solr 是独立的开放源码的企业级搜索服务器软件，由 Apache 软件基金会所研发[6]。Solr 自身采用 Java 5 开发，使用 Lucene 程式库[7] 以及需要 Servlet 容器作执行环境，如 Tomcat，Jetty，Resin 等。Solr 服务器使用标准的 HTTP 与 XML 通信协议，对外提供 HTTP/XML 与 JSON 的应用程式接口，用户可以通过 HTTP 协议与 Solr 搜索引擎服务器进行交互，分析文档、创建、更新索引和提交查询请求，并得到 XML 或 JSON 格式的返回结果。

Solr 最显著的特性就是提供了分面检索组件 Facet-Component 实现对分类检索的优化处理，通过自建的集合类来提供高效的分面结果计算反馈给用户。同时提供了翔实的查询语言和语法规则，具有高度的可扩展、可配置性能，灵活的动态集群和信道共用机制，独立的主—从分布式索引复制机制支持分面搜索和索引的快速复制，能在大规模的数据量中分散应用服务器的查询负载，并能挖掘出交叉分类数据之间的部分内在联系，提高分类检索的响应速度和效率。如果说，传统的分类检索技术重在提供“分类功能”，专注于分类表层的建设，而 Solr 则重在处理“分类检索过程和结果”，专注于知识服务应用。此外，Solr 还提供了一个基于 Web 的可视化的客户端功能管理界面，方便项目研发人员进行分类查询、文本分词等功能的应用配置和测试。目前，Solr 已经在很多数字图书馆开源软件、大型 Web 数据库系统甚至包括政府门户网站在内的综合性网站服务系统中得到广泛应用[8]，如内容管理工具 Drupal[9]、学术期刊开源系统 JSTOR[10]、世界医学期刊索引库 Pubget[11]、数字图书馆开源系统 VuFind[12]、数字图书馆开源集成系统 ClavisNg[13]、互联网档案馆 Archive. org、开源软件平台 SourceForge. net、美国联邦通信委员会网站 FCC. gov、美国在线 AOL，eBay 等。

2 开放知识资源分类检索机制优化研究

2.1 开放知识资源分类检索需求

开放资源按元数据描述框架体系可分为三大类：资源、机构、学术会议。对于资源，缺省状态需要实现按资源类型、学科分类、语种、Subject、访问控制方式、资源来源来分类检索。同时资源类型又包含了子一级的资源内容类型，即当点击资源类型进行分类检索时，需将资源内容类型作为新的条件和上述条件重新结合进行重新分面并显示结果。而 Subject 包含了中文主题词和英文主题词，需要根据语种的变化而变化。缺省 Subject 按英文主题词进行分面，当语种被选中为“汉语”时，Subject 需变化为按中文主题词进行分面。对于机构，缺省状态需要实现按机构类型、学科分类、国家、Subject、数据来源进行分类检索。同时国家是个 3 层级的级联条件，它包含了子一级的地区条件，地区又包含了更下一级的省市条件，当点击国家进行分类检索时需要地区作为新的分面条件和上述条件结合重新进行分面。当点击地区时，如果地区包含有相应的省市，则省市也必须列为分面条件之一。Subject 缺省按英文主题词进行分面，当国家被选中为“中国”时，需变化为按中文主题词进行分面。对于学术会议则比较简单，分面条件的变化性比较小，缺省按照学科分类、会议时间、会议地点和会议级别进行分面。此外，资源、机构、学术会议在分类检索的同时，还必须能够支持用户输入任意检索词对分类结果进行二次过滤分类。

考虑到以上的分类检索需求，为了实现关系数据库和 Solr 服务器的 HTTP 通信，必须在原有项目中嵌入 Solr 客户端。Solrj 是 Solr 基于 Java 的客户端，它在底层封装了 httpClient 方法，并进行了相应的扩展，提供了更为完善的操作 Solr 的 API[14]。具体应用时需要把相关的 Lib 库加入开放知识资源的 Lib 库。

2.2 分类索引配置的优化

为改进性能，提高索引效率，并不是所有的关系数据库字段都需要被 Solr 服务器索引和存储。核心优化策略是将所有只用于搜索过程的，而不需要作为结果显示的字段

Field（特别是一些比较大的 Field）的 Stored 设置为 False；将不需要被用于搜索的而只是作为结果返回 Field 的 Indexed 设置为 False；在服务器端运行 JVM 时使用尽可能高的 Log 输出等级，减少日志量等。但用于分类检索的字段必须被索引。Solr 索引的基本性能参数见表 1。

表 1　Solr 索引的性能参数

性能参数	描　述
useCompoundFile	通过将很多 Lucene 内部文件整合到单一文件来减少使用中的文件数量。这可有助于减少 Solr 使用的文件句柄数目，代价是降低了性能。除非是应用程序用完了文件句柄，否则 False 的默认值应该就已经足够
mergeFactor	决定低水平的 Lucene 段被合并的频率。较小的值（最小为 2）使用的内存较少但导致的索引时间也更慢。较大的值可使索引时间变快但会牺牲较多的内存
maxBufferedDocs	在合并内存中文档和创建新段之前，定义所需索引的最小文档数。段是用来存储索引信息的 Lucene 文件。较大的值可使索引时间变快但会牺牲较多的内存
maxMergeDocs	控制可由 Solr 合并的 Document 的最大数。较小的值（＜10000）最适合于具有大量更新的应用程序
maxFieldLength	对于给定的 Document，控制可添加到 Field 的最大条目数，进而截断该文档。如果文档很大，就需要增加这个数值。然而，若将这个值设置得过高会导致内存不足错误
unlockOnStartup	unlockOnStartup 告知 Solr 忽略在多线程环境中用来保护索引的锁定机制。在某些情况下，索引可能会由于不正确的关机或其他错误而一直处于锁定，这就妨碍了添加和更新。将其设置为 True 可以禁用启动锁定，进而允许进行添加和更新

2.3　分类检索过程的优化

Solr 提供了优化分类检索过程的机制。其实分类检索过程和上述的索引配置息息相关。其中最重要的就是定义字段 Field 的字段类型 FieldType 在进行分类查询时要使用的分析器 Analyzer，包括复杂的分词和相关的过滤。一般常用到的 FieldType 主要有 String，Date 和 Text 三种。其中 String 和 Data 类型可以直接使用 Solr 索引配置文件中已定义好的。Text 类型必须根据分类需求自定义。考虑到分类检索中需要支持用户任意输入检索词对分类结果进行二次过滤，因此 Text 类型应定义为能够支持中文分词。Solr 能够控制分类检索过程是否需要应用分词，分词的粒度是 AND 还是 OR 等逻辑的分类关系，但自身并不支持中文分词，必须加入辅助的中文分词包来辅助实现分词功能。因此，本文引用了 IKAnalyzer 来进行中文分词，IKAnalyzer 是一个开源的，基于 Java 语言开发的轻量级的中文分词工具包[14]。它采用了特有的“正向迭代最细粒度切分算法”，支持细粒度和智能分词两种切分模式，具有 60 万字/s 的高速处理能力。同时采用了多子处理器分析模式，优化的词典存储，更小的内存占用，采用歧义分析算法优化查询关键词的搜索排列组合，能极大地提高分类检索的命中率。同时，还定义了一个 Text 类型的 Atext 多值字段 CopyField，作为相关需要加入分类检索字段，以获取更为精确的检索结果，但字段自身又是 String 类型，被要求不能够被分词的字段的目标域，Solr 在创建索引时，会自动复制上述字段索引合并到 Atext 索引中。在分类检索实现的时候，对 Solr 来说，只需在 Atext 字段中检索，即相当于同时在多个需求字段中分类检索，便可满足用户对多个分类类目交叉检索，实现“一次输入，多方搜索，统一输出”，极大地提高了分类检索的效率。

2.4　分类检索访问的优化

传统的分类检索技术依赖于程序和数据库的直接交互，一旦数据库出现问题会立刻造成分类检索功能的失效。基于 Solr 的分类检索机制封装了分类检索的目标数据，对数据库而言，分类检索的功能模块是完全独立于数据库而存在的，类似于数据库的一个镜像。虽然分类检索的数据是实时更新的，但开发人员在分类检索模块里更新、删除索引数据等都不会直接影响到数据库，用户在进行分类检索时也不会直接影响到数据库，很大程度上分担了主数据库服务器的负载和访问压力。这也是 Solr 分类检索功能能够得到广泛应用的先决优势之一。具体操作时仍通过基于 Solrj 的 Java 编程进行实现。需要实现一个核心的线程用于和 Solr 服务器进行私有通信，同时在该线程中实例化一些方法，创建索引文档对象列表，将数据库中的元数据依次映射到 Solr 的分类索引中进行优化存储。

2.5　客户端分类检索功能的实现

基本思想是创建分类检索的实例方法，并初始化一个查询条件字段，用于存放所有可能出现交互的分类检索条件。缺省状态设置为开启分类，按预定义的分类条件查询全部 Solr 索引并输出结果。在实际执行过程中，如果从客户端接收到某个新的分类检索请求，则依次在原有查询条件中添加该条件。然后将最终形成的新查询条件重新添加到最初创建的分类检索实例方法中，调用 Solr 服务端核心线程的公有方法，建立与 Solr 服务器端的通信，将全部查询请求提交到 Solr 服务器进行处理，获取到分类检索结果的列表 List，循环该 List 即可得到所有的分类类目和每个类目的分类结果数。

3　分类检索应用效果

在向索引库中添加全文检索类型的索引文档的时候，

Solr 会首先用空格进行分词，然后把分词结果依次使用指定的过滤器进行过滤。这其中包括使用过滤词过滤器（StopFilter）、拆字过滤器（WordDelimiterFilter）、小写过滤器（LowerCaseFilter）、英文相近词过滤器（English-PorterFilter）和去除重复词过滤器（RemoveDuplicatesTokenFilter），最后剩下的结果才会加入到索引库中以备查询。Solrf 分类检索分词的效果，以输入检索词“中国科学院国家科学图书馆”为例，去除重复词过滤器的过滤结果见图 1。

position	1	2	3	4	5	6	7	8	9	10	11
term text	中国科学院	中国	科学院	科学	学院	国家科学	国家	科学	图书馆	图书	书馆
keyword	false	false	false	false	false	false	false	false	false	false	false
startOffset	0	0	2	2	3	5	5	7	9	9	10
endOffset	5	2	5	4	5	9	7	9	12	11	12
type	word	word	word	word	word	word	word	word	word	word	word

图 1　Solr 去除重复词过滤分词效果

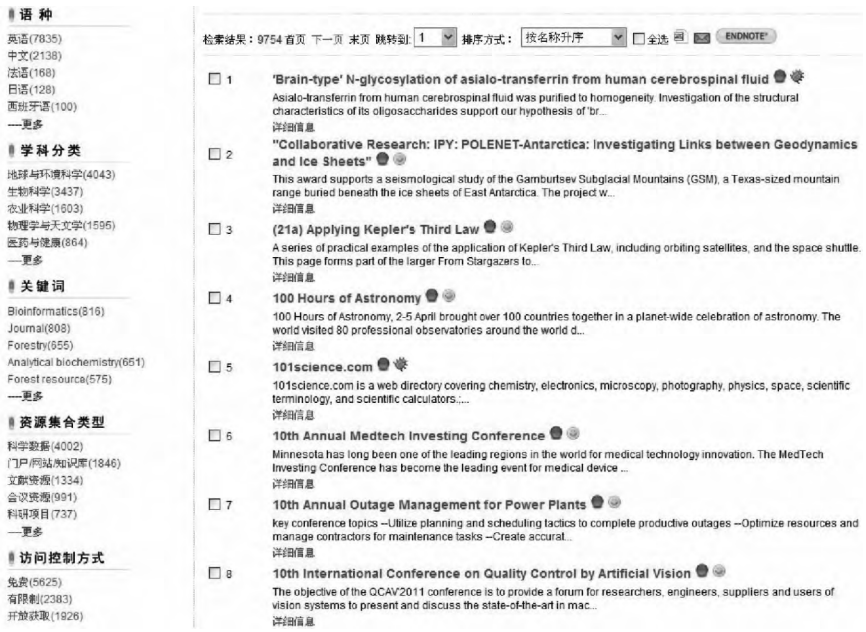

图 2　资源分类检索效果

客户端 JSP 页面的分类检索效果，以资源为例示例如图 2 所示。整体分面效果良好，准确实现了开放知识资源按照语种、学科分类、关键词、资源类型、访问控制方式等分类检索和导航功能，能够满足用户和管理人员对项目后期改进的初步需求。对于目前系统中登记的近 20000 多条数据来说，无论是创建分类索引，还是分类检索过程，与原来的基于 Dtree 的传统分类检索技术相比，经 Solr 优化后的分类检索机制的响应时间（Solr 参数为 QTime）基本都能控制在 16ms 内。今后，随着数据量的增加，响应时间虽然肯定会有所变长，但都会在一个可控的范围内。

4　结束语

本文通过分析传统分类检索技术的不足之处，对开放知识资源分类检索机制进行了优化研究，并将技术方案应用于开放知识资源建设项目，最终提高了分类检索的速度和灵活性，有助于开放知识资源服务质量的提升和改善。此外，由于 Solr 的可配置性、可扩展性和独立性等性能，基于 Solr 的分类检索模块独立于数据库服务器而存在，等同于一个数据接口，只需要获取相应的查询参数和方法，便可批量获取相关开放知识资源的摘要信息，有助于推动实现开放知识资源的共享和重利用。□

参考文献

[1] http://irsr.llas.ac.cn/aboutus/aboutus.jsp.

[2] http://www.destroydrop.com/javascripts/tree.

[3] http://en.wikipedia.org/wiki/Faceted_ browser.

[4] 宋乐平. 中文数据库分类检索能力研究 [J]. 图书馆学研究，2010（3）：63-66.

[5] RUSSELL-ROSE T，TATE T. Designing the search experience [M]. San Francisco：Morgan Kaufmann Publishers Inc，2013：167-218.

[6] http://lucene.apache.org/solr.

[7] http://lucene.apache.org.

[8] http://wiki.apache.org/solr/PublicServers.

[9] Drupal [EB/OL]. [2012-08-15]. http://drupal.org.

[10] JSTOR [EB/OL]. [2012-08-15]. http://www.jstor.org.

[11] Pubget [EB/OL]. [2012-08-15]. http://pubget.com.

[12] VuFind [EB/OL]. [2012-07-12]. http://vufind.org.

[13] ClavisNG [EB/OL]. [2012-08-24]. http://www.comperio.it/soluzioni/clavisng/un-gestionale-per-reti-di-biblioteche.

[14] http://wiki.apache.org/solr/Solrj.

[15] http://code.google.com/p/ik-analyzer.

作者简介：王思丽，女，1985 年生，硕士，馆员。

马建玲，女，1969 年生，研究馆员，硕士生导师。

姚晓娜，女，1985 年生，硕士，馆员。

收稿日期：2013－09－12