doi:10.3969/j.issn.1673-9833.2015.05.015

基于云存储的网络文档共享系统

杜红刚，吴岳忠

（湖南工业大学 计算机与通信学院，湖南 株洲 412007）

摘 要：针对非结构化的海量文档获取困难的问题，设计和开发了基于云存储的网络文档共享系统。该系统采用了Hadoop和Lucene以及Mahout来实现对文档存储、全文检索和推荐。通过测试证明，网络文档共享系统可以使用户更快速高效地获取文档。

关键词：云存储；文档共享；全文检索；推荐

中图分类号：TP338.8 **文献标志码**：A **文章编号**：1673-9833(2015)05-0072-05

Network Document Sharing System Based on Cloud Storage

Du Honggang，Wu Yuezhong

（School of Computer and Communication，Hunan University of Technology，Zhuzhou Hunan 412007，China）

Abstract：In view of the difficult problem of massive document acquisition, the network document sharing system based on cloud storage is designed and developed. The system uses Hadoop，Lucene and Mahout to achieve the document storage, full-text search and recommendation. The test shows that the network file sharing system can be used to obtain the documents more quickly and efficiently.

Keywords：cloud storage；document sharing；full-text retrieval；recommendation

随着计算机硬件的飞速发展，处理速度的不断提高，人们在获取快速的处理速度同时也产生着大量的文档和用户数据。然而随着互联网的到来和快速发展，这种数据呈指数级增长。在工作中，人们从这些大量的文档中获取自己想要的文档时，就变得越加困难而缓慢。

为了解决这个问题，本文提出并设计实现了基于云存储的网络文档共享系统。主要研究工作及创新点为：

1）系统将文档存储在以Hadoop为基础的文件存储集群，它可以使人们摆脱对移动存储设备的依赖，只要有网络就可以随时随地的访问自己的文件，而且成本低廉，安全稳定；

2）系统配合使用Lucene作为全文搜索引擎，提取文档关键词特性，以矢量的方式来标识文档的特征，从而为用户快速查找到有效文档；

3）系统通过Mahout数据挖掘来为用户做智能推荐。

1 相关技术

1.1 云存储

云存储是在云计算（cloud computing）概念上延伸和发展出来的一个新的概念，是一种新兴的网络

收稿日期：2015-08-15

基金项目：国家自然科学青年科学基金资助项目(61502163)，湖南省教育厅科研基金资助项目(14C0323)，湖南工业大学科研基金资助项目（2014HZX16，KFK1402）

作者简介：杜红刚(1994-)，男，安徽宿州人，湖南工业大学学生，主要研究方向为云计算，Web技术，E-mail：970255897@qq.com

通信作者：吴岳忠(1981-)，男，江苏江阴人，湖南工业大学讲师，硕士，主要研究方向为云计算，推荐系统和大数据，E-mail：5174979@qq.com

存储技术，是指通过集群应用、网络技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。如今存储的解决方案很多，其中Hadoop是比较成熟的一种方案。Hadoop是Apache软件基金会所研发的开放源码项目。它是一个能够让用户轻松架构和使用的分布式计算平台。如今很多知名的IT企业都在使用Hadoop，如京东、百度等公司在存储、分析日志、数据挖掘和机器学习中都使用了Hadoop。在文献[1−2]中通过实战开发，详细阐述了Hadoop的程序开发方式。在文献[3]中通过实验，证明Hadoop在大数据处理发面的优势。在文献[4−5]中，介绍了Hadoop的分布式文件系统HDFS的搭建。本文使用Hadoop搭建分布式文件存储系统，为本文设计的系统提供低廉、安全的云存储。

1.2　全文检索

全文检索是一种将文件中所有文本与检索项匹配的文字资料检索方法。普通的文档系统只能通过标题这种结构化的数据来进行搜索，而全文检索不同，它是以文档内容为分析对象，通过分词器以及分词库将文档内容中的各个词汇汇总统计，这样便能很好地标识这个文档的特征，从而更智能地匹配用户的搜索要求[5]。通过全文检索就可以更深入地为用户推荐用户所需内容，因为文档的命名无法覆盖文档表达的所有方面，如果用输入的搜索关键词比较少，那么就无法将那些没有通过命名来体现文档内容的文档查询出来。所以，通过全文检索直接分析文档内容更直接、更有利于搜索和推荐。

Lucene是一套用于全文检索和搜寻的开源程式库，由Apache软件基金会支持和提供。Lucene提供了一个简单强大的应用程式接口，能够实现全文索引和搜寻。很多国外的企业已经将Lucene投入使用，例如推特、FaceBook等。文献[6]详细介绍了Lucene的开发步骤如索引的更新、文档搜索等。本系统中使用Lucene对文档进行高效的全文检索。

1.3　智能推荐

文档推荐是基于用户兴趣点对用户的文档需求进行预测，然后将预测结果推送给用户。文档推荐和搜索在系统内实现基本一致，不同的是文档搜索只有在用户主动发起请求的时候系统才会执行搜索动作，这是用户主动的行为；而文档推荐是系统根据用户的兴趣主动去文档库去搜索，然后将结果显示给用户，这时用户是被动的。在文档智能推荐算法中，常用的有UserCF和ItemCF。UserCF是用户间类比推荐的一种横向推荐，它的作用是推荐那些和该用户有共同兴趣的用户所感兴趣的文档，反应群体内的热门程度；ItemCF是基于文档特性的纵向推荐，它的作用是推荐那些该用户之前喜欢的文档，反应用户本人的兴趣爱好[7]，这2种算法都可以通过Mahout来实现。Mahout是一个强大的数据挖掘工具，是一个分布式机器学习算法的集合，其最大的优点就是可以基于Hadoop实现，把很多以前运行于单机上的算法，转化为MapReduce模式，这样大大提升了算法可处理的数据量和处理性能[8]。本系统将通过Mahout把UserCF和ItemCF这2种算法结合使用为用户提供智能推荐。

2　系统设计与实现

2.1　系统功能设计

系统功能分为4个模块：文档存储模块、文档搜索模块、文档推荐模块、用户登录和权限模块。这4个模块组成系统的整个核心业务，系统功能结构图如图1所示。

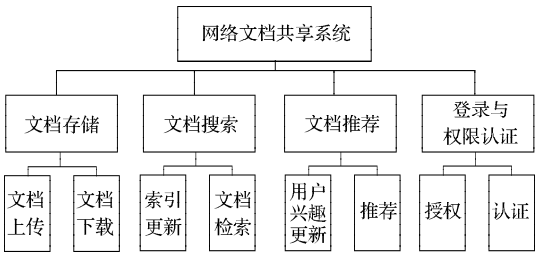


图1　系统功能结构图

Fig. 1　System function structure

具体模块功能如下：

1）文档存储模块。包括文档上传了文档下载2个功能。主要实现了用户的文档上传到云存储服务器，以及从云存储服务器下载文档。

2）文档搜索模块。文档搜索模块是整个系统的核心，文档推荐模块也依托于文档搜索模块。包括索引更新和文档检索功能，由于文档库是实时变化的，所以要对整个文档库的所有文档进行索引的更新，这样才能在文档搜索的时候得到最新的结果。

3）文档推荐模块。系统根据用户的兴趣有针对的对用户进行智能推荐。它包括用户兴趣更新和推荐2个功能，其中用户兴趣更新是更新用户的兴趣，为推荐做基础。

4）登录和权限认证模块。它是系统的基础，包括授权和认证2个部分，其中授权是对用户在系统中所能进行的操作进行授权，而认证是对用户在系统内的操作进行认证的过程。

2.2 系统架构设计

系统在J2EE平台开发，代码基本采用Java语言编写。服务器使用Tomcat作为Web容器，数据库采用MySQL，分布式文件存储使用Hadoop的HDFS。在代码框架上使用Struts2作为MVC框架，Spring作为注入功能和事务控制，Hibernate作为数据库存储层的框架，安全控制使用Spring Security[9–12]。系统架构图如图2所示。

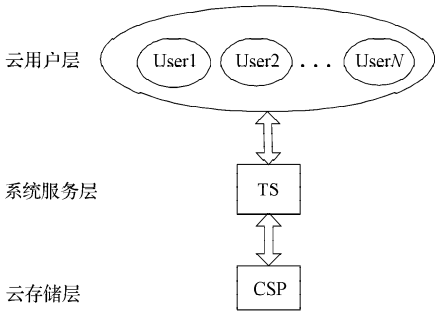


图2 系统架构图

Fig. 2 System architecture

如图2所示，本系统具有3层系统架构，包括：云用户层、系统服务层和云存储层。在该架构中，由用户组成的云用户层，直接使用应用，进行文档的上传、检索等资源共享服务。由可信服务器（trust servers，TS）作为系统服务层，支持用户和云存储的交互，一方面是可以对源文件进行创建索引，再把源文件和索引文件上传到云存储服务器上，还可以将用户提供的关键词提交云存储层进行搜索，并将获得结果返回用户，另一方面是处理用户兴趣模型，便于系统进行内容的按需推荐；由云服务器提供商（cloud servers provider，CSP）作为云存储层，主要与系统服务层交互，对文档资源进行存储，并可以提供云的超强计算能力，如对上传海量数据源文件进行存储、搜索、文件提取等操作；

2.3 数据库设计

系统的数据库采取MySQL数据库，系统的结构化数据存储在MySQL数据库中，包括登录权限认证数据、用户的基本信息、文档的基本信息、文档的分享信息以及用户的搜索记录。系统功能涉及部分数据表设计如下。

1）用户信息表。包括用户ID、用户名、角色ID以及用户密码。

2）角色表。包括角色ID、角色名。

3）文档信息表。包括文档ID、文档实际名、文档存储名、文档URL地址、文档大小、文档类型、文档状态等。

4）兴趣信息表。包括兴趣ID、兴趣关键词、关键词热度。

5）权限信息表。包括权限ID、权限名。

系统ER图如图3所示。

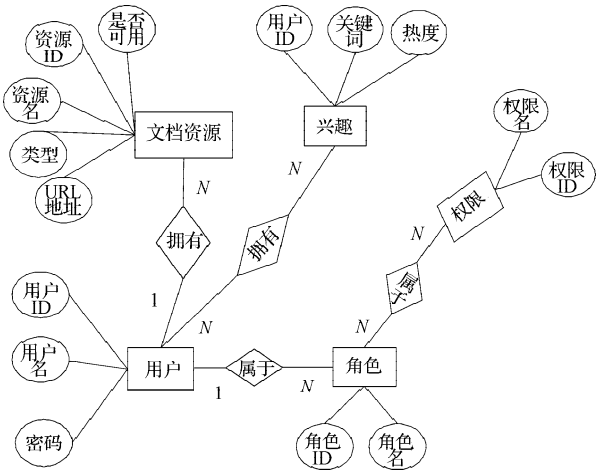


图3 系统ER图

Fig. 3 The ER chart of system

2.4 文档搜索和推荐核心算法设计

搜索和推荐，其本质就是检索文档内容是否匹配用户的需求。本文所采用匹配算法设计思路为：每个文档都是由很多词组组成的，通过Lucene索引，可以将每个文档的词组统计出来，并且可以计算每个词组占整个文档词组总量的比例，称之为该词组在此文档内的权重，将所有的词组权重有序的排放成一个数列，称此数列为该文档的文档空间向量，该向量可以表示一个文档的内容特征。同理用户有很多感兴趣的关键词，而这些关键词组的权重所组成的有序数列，也表示了用户的兴趣特征。其中词组权重 W 的计算公式为式（1），即

$$W_{t,d} = tf_{t,d} * \log\left(n / df_t\right), \tag{1}$$

式中：$W_{t,d}$ 表示关键词 t 在文档 d 中的权重；

$tf_{t,d}$ 表示关键词 t 在文档 d 中的频率；

n 表示文档集 $Data$ 中文档总个数；

df_t 表示包含关键词 t 的文档个数。

在进行搜索和推荐的时候，系统会计算文档空间向量和用户兴趣空间向量的相似度 $Score_{q,d}$，其计算公式为式（2），即

$$Score_{q,d} = \frac{\sum_{i=1}^{n} W_{i,q} W_{i,d}}{\sqrt{\sum_{i=1}^{n} W_{i,q}^2} \sqrt{\sum_{i=1}^{n} W_{i,d}^2}}, \tag{2}$$

式中：$W_{i,q}$ 表示用户兴趣向量 $\boldsymbol{q}$ 中的关键词 i 的权重；

$W_{i,d}$ 表示文档向量 $\boldsymbol{d}$ 中的关键词 i 的权重；

n 是文档集中关键词的个数。

用户在进行搜索和推荐的时候，会根据关键词和用户兴趣分计算出其与每个文档的相似度，然后

将相似度最高的 N 个文档作为结果返回给用户。

2.5　算法实现

2.5.1　文档上传到HDFS

文件上传至 HDFS 中，是通过 Hadoop 提供的一套 API 来操作，由于 Hadoop 封装性较好，所以使用较为简单。HDFS 的核心类是 FileSystem，通过该类实现文件在 HDFS 中的存储、读取、删除等操作。上传流程如图 4 所示。

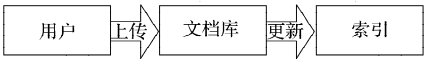


图4　文档上传流程图

Fig. 4　Document upload flowchart

2.5.2　文档索引建立

文档索引的建立时，索引文件是存储在可信服务器上的，文档量逐渐变大，系统更新索引也会随之变慢，所以系统会在午夜进行索引的更新。实现步骤有创建索引目录对象、创建索引的写入器、创建 Document、为 Document 添加 Field、获取 HDFS 的 FileSystem 实例，遍历 HDFS 中的文档、设置所要索引的域、通过 IndexWriter 添加文档到索引中、关闭索引目录和索引写入器。建立索引的过程如图 5 所示。

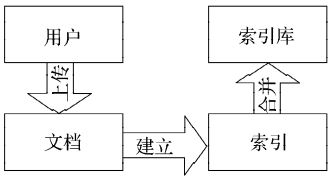


图5　文档索引建立流程图

Fig. 5　Document index flowchart

2.5.3　文档搜索

搜索流程如图 6 所示。文档搜索的时候，在用户输入关键词后系统会通过 Lucene 加载已经建立好的索引，索引在加载后通过关键词来获取满足搜索要求的文档。实现步骤为：创建目录对象、创建索引读取实例、创建搜索的搜索对象、搜索并返回最符合条件的前 n 条、根据 TopDocs 获取 ScoreDoc 对象，然后遍历所搜索到的项、根据 Seacher 和 ScoreDoc 对象获取具体的 Document 对象、根据 Document 对象获取需要的值。

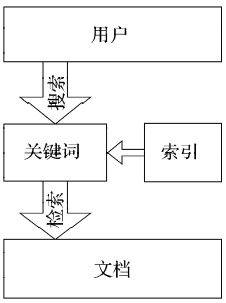


图6　文档搜索流程图

Fig. 6　Document search flowchart

2.5.4　文档推荐

用户每次搜索后，系统都将存储用户本次搜索的关键词，但关键字在系统中不存在的时候将在系统中添加该关键词，如果已经有了就会把这个关键词的 hot 指数加一，然后根据用户使用最频繁的几个关键词为用户做推荐。推荐流程如图 7 所示。

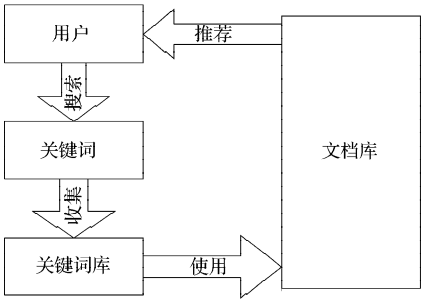


图7　文档推荐流程图

Fig. 7　Document recommendation flowchart

2.6　系统设计特点

1）采用 HDFS 摆脱了传统目录文件系统的深层次结构，HDFS 可采用扁平的文件系统结构，而且 HDFS 容错性高，支持自动备份，使得文件更安全。

2）系统采用 Lucene 为文档库建立索引，当文档量很大的时候势必会带来性能问题。但是由于系统采用分布式架构，可以将建立索引和检索索引的工作以 MapReduce 的方式进行处理，即将任务分割成许多的小任务进行并行计算，然后再将运算结果合并成最终的运算结果，从而大大缩短索引的时间。

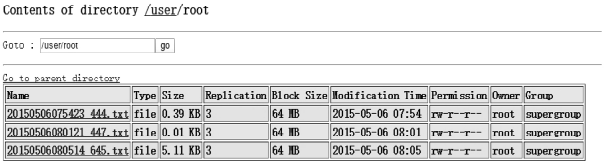
3）系统使用 Mahout 推荐算法可以根据用户在系统内的操作记录，抽象出来一个用户的兴趣趋向，从而为用户进行有针对性、有价值的推荐，而且随着用户的数据的增多，推荐越准确。

3　实验分析

本文采用的开发平台和工具等如 2.2 节所述。开发工具使用 IDEA，版本控制工具为 Git，项目编译发布工具使用 Maven。

3.1　文档存储

用户上传文档的功能，在上次文档界面，选择所要上传的文件然后点击上传。这部分主要是对数据进行验证，首先查看数据可得 Document 表里面是否有新的文档记录，然后查看 HDFS 里的是否有该文件。效果图如图 8 所示。


Contents of directory /user/root

Goto : /user/root　go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
20150506075423_444.txt	file	0.39 KB	3	64 MB	2015-05-06 07:54	rw-r--r--	root	supergroup
20150506080121_447.txt	file	0.01 KB	3	64 MB	2015-05-06 08:01	rw-r--r--	root	supergroup
20150506080514_645.txt	file	5.11 KB	3	64 MB	2015-05-06 08:05	rw-r--r--	root	supergroup

图8　文档上传

Fig. 8　Document upload

3.2 文档搜索

系统事先对整个文档库进行索引的更新，在系统内的搜索输入框内输入要搜索的关键词，点击搜索，系统根据关键词查找文档库中匹配该关键词的文档。效果如图 9 所示

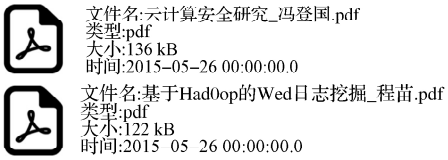


图9 文档搜索

Fig. 9 Document search

3.3 文档推荐

系统是根据用户的兴趣进行测试，即系统收集用户的搜索关键词，然后根据关键词出现的频率来进行文档的推荐，这样的测试的时候只需要不断的更新用户的关键词频率即可，例如可以不断地搜索某一个关键词，使改用的这个关键词的频率升高，然后去查看系统为该用户的推荐结果。效果图如图 10 所示。

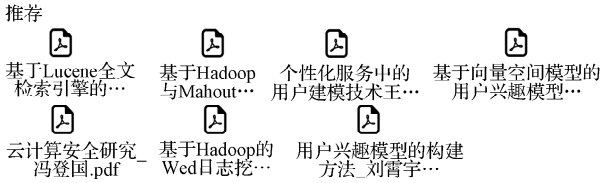


图10 文档推荐

Fig. 10 Document recommendation

4 结语

通过搭建基于 Hadoop 的分布式文件系统，配合 Lucene 的全文检索以及基于 Mahout 数据挖掘的智能推荐，完成基于云存储的网络文档共享系统的设计与开发。经测试，系统提高了用户的文档管理效率，使用户可以方便、高效且智能地获取文档。

参考文献：

[1] Tom White. Hadoop权威指南[M]. 北京：清华大学出版社，2011：42-117.
Tom White. Hadoop Authority Guide[M]. Beijing：Tsinghua University Press，2011：42-117.

[2] 陆嘉恒. Hadoop实战[M]. 2版. 北京：机械工业出版社，2012：162-186.
Chen Jiaheng. Hadoop Practice[M]. 2nd ed. Beijing：China Machine Press，2012：162-186.

[3] 董西成. Hadoop技术内幕：深入解析MapReduce架构设计与实现原理[M]. 北京：机械工业出版社，2013：76-120.
Dong Xicheng. Hadoop Technologies：In-Depth Analysis of MapReduce Architecture Design and Implement Principle [M]. Beijing：China Machine Press，2013：76-120.

[4] 吴岳忠，周训志. 面向 Hadoop 的云计算核心技术分析[J]. 湖南工业大学学报，2013，27(1)：77-80.
Wu Yuezhong，Zhou Xunzhi. The Core Technology of Hadoop-Oriented Cloud Computing[J]. Journal of Hunan University of Technology，2013，27(1)：77-80.

[5] 吴岳忠，刘 琴，李长云，等. 基于云存储的网络文档共享研究[J]. 小型微型计算机系统，2015，36(1)：95-99.
Wu Yuezhong，Liu Qin，Li Changyun，et al. Research on Cloud Storage Based Network Document Sharing[J]. Journal of Chinese Computer Systems，2015，36(1)：95-99.

[6] 成 龙. Lucene搜索引擎开发进阶实战[M]. 北京：机械工业出版社，2015：45-132.
Cheng Long. Lucene Search Engine Development and Advanced Practice[M]. Beijing：China Machine Press，2015：45-132.

[7] 樊 哲，Dmitry Babenko. Mahout算法解析与案例实战[M]. 北京：机械工业出版社，2014：56-100.
Fan Zhe，Dmitry Babenko. Mahout Algorithm Analysis and Cases Practice[M]. Beijing：China Machine Press，2014：56-100.

[8] 迈纳，舒克. MapReduce设计模式[M]. 北京：人民邮电出版社，2014：82-125.
Miner Donald，Shook Adam. MapReduce Design Patterns [M]. Beijing：Post & Telecom Press，2014：82-125.

[9] 杨 波，刘 渊，冷文浩. 基于Struts+Hibernate+Spring架构的船舶数字化平台的设计与实现[J]. 计算机应用与软件，2008，25(2)：178-180.
Yang Bo，Liu Yuan，Leng Wenhao. Ship Digitized Platform Based On Architecture of Struts & Hibernate & Spring[J]. Computer Applications and Software，2008，25(2)：178-180.

[10] Sierra K，Bates B. 深入浅出 Java [M]. 2版. 南京：东南大学出版社，2005：50-130.
Sierra K，Bates B. Explaining Java[M]. 2nd ed. Nanjing：Southeast University Press，2005：50-130.

[11] 李 刚. 轻量级JavaEE企业应用Struts2+Spring+Hibernate整合开发[M]. 北京：电子工业出版社，2008：210-400.
Li Gang. Lightweight JavaEE Enterprise Application：Struts 2 & Spring & Hibernate Integration Development [M]. Beijing：Publishing House of Electronics Industry，2008：210-400.

[12] Eckel Bruce . Java编程思想[M]. 4版. 北京：机械工业出版社，2007：103-215.
Eckel Bruce . Java Programming Thinking[M]. 4th ed. Beijing：China Machine Press，2007：103-215.

（责任编辑：申 剑）