ROSA 设计文档

2009-8-26

目录

1 需求说明

现有的搜索引擎 Thephy 存在的问题:

1. 不支持复杂搜索，只是简单的字符串匹配。这样的搜索匹配效率不高，最严重的缺点是不能支持用户的自定义搜索，比如用户只想搜后缀名为 rmvb 的某个视频，现有的搜索引擎 Thephy 是无能为力的。

2. 搜索结果冗余。比如对于”/movie/ 蓝莓之夜 /1.rmvb”这样的资源，当搜索“蓝莓之夜”时结果会有两条：

   - /movie/ 蓝莓之夜
   - /movie/ 蓝莓之夜 /1.rmvb

   当用户输入的关键词匹配条目很多的时候，这种冗余会带来很大的混乱，迫使用户浪费很多时间来区分条目。

3. 现有的搜索引擎不支持任何形式的排序。对于很多资源，比如软件，电影，用户往往喜欢选择最新的资源下载，在没有排序支持的情形下需要用户人工区分资源更新时间。

4. 考虑到同一关键字可能会同时出现软件，电影等资源名称中，所以分类搜索也是用户十分需要的功能。

5. 现有的搜索引擎用专门的论坛负责站点登记，当用户想把自己的站点登入的时候，要在论坛发帖，然后由管理员手工添加。这种方式过于浪费人力和时间。我们需要一个 FTP 服务器站点的自主管理平台。

6. 现有的搜索引擎无法确定结果页中的站点是否可以访问。我们希望能在搜索结果页面中给出站点连接性能提示，以便用户选择。

ROSA 搜索引擎要实现的功能:

1. 使用当代搜索引擎技术，抛弃简单的字符串匹配，而使用基于分词的倒排表保存索引数据和提供搜索服务。

2. 细致的可用搜索域，可以满足用户各种搜索需求。

3. 针对常用搜索域的升序降序排列。方便用户排序搜索资源。

4. 提供针对路径名和文件名的多种搜索方式，以使搜索结果简洁美观。

5. 对资源根据文件后缀进行自动分类，并提供相应的搜索接口，方面用户进行分类搜索。

6. 自主的站点登录、管理和登出系统。方面 FTP 管理员管理站点。

7. 资源访问和站点访问统计，方便用户分享各自的兴趣和选取热点站点。

8. 站点可连接性提示，方便用户选择站点下载资源。

9. 多点下载程序，减轻用户站点选取的负担，因为用户关注的是资源而非站点。所以我们需要一个多点下载程序可以根据用户选取的资源信息同时从多个站点来分片下载资源。

2 开发工具

开发语言 ROSA 后台使用 Java 语言开发，前台使用 JSP 和 Struts 架构构建。

开发工具 开发工具为 MyEclipse 6.0.1 和 MySql 5.0。

服务器 使用的服务器为 Apache Tomcat 6.0.18。

3 使用的开源软件

3.1 Lucene

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 提供了一个简单却强大的应用程式接口，能够做全文索引和搜寻，在 Java 开发环境里 Lucene 是一个成熟的免费开放源代码工具；就其本身而论，Lucene 是现在并且是这几年，最受欢迎的免费 Java 资讯检索程式库[1]。

3.2 Struts

Struts 是 Apache 软件基金会（ASF）赞助的一个开源项目。它最初是 Jakarta 项目中的一个子项目，并在 2004 年 3 月成为 ASF 的顶级项目。它通过采用 Java Servlet / JSP 技术，实现了基于 Java EE Web 应用的 Model-View-Controller〔MVC〕设计模式的应用框架〔Web Framework〕，是 MVC 经典设计模式中的一个经典产品[2]。

3.3 Log4j

log4j 是一个开源的日志系统，它允许开发者以任意的粒度控制日志语句的输出。通过使用外部配置文件，log4j 在运行时也可以自由的配置。最重要的是 log4 简单易学，非常容易上手[3]。

3.4 JDom

JDOM 是一个开源项目，它基于树型结构，利用纯 JAVA 的技术对 XML 文档实现解析、生成、序列化以及多种操作。JDOM 直接为 JAVA 编程服务。它利用更为强有力的 JAVA 语言的诸多特性（方法重载、集合概念以及映射），把 SAX 和 DOM 的功能有效地结合起来[4]。

3.5 Commons Net

Jakarta Commons Net 实现了很多客户端的网络协议。这个程序库的目的在于提供基本的协议使用而非高层抽象。它支持的协议包括：

- FTP/FTPS
- NNTP
- SMTP
- POP3
- Telnet
- TFTP
- Finger
- Whois
- rexec/rcmd/rlogin
- Time (rdate) and Daytime
- Echo
- Discard
- NTP/SNTP

3.6 Jakarta Oro

Jakarta-ORO 是一套兼容 Perl5 和类 AWK 正则表达式的文本处理 API[5]。

4 软件总体架构

考虑到缺乏专用的服务器，如果将所有的爬虫程序放在某一台机器上的话会影响正常的使用，所以我们考虑了分布的方式来进行索引和提供搜索服务。我们通过“站点编号 mod 站点数”来给每个索引节点分配任务。每个节点都同时提供搜索服务，采用 RMI 的实现方式。独立设计的爬虫程序，采用线程池并发的从多个 FTP 服务器站点获取资源列表，然后提交给索引器建立索引。图1给出了 ROSA 的三层架构。

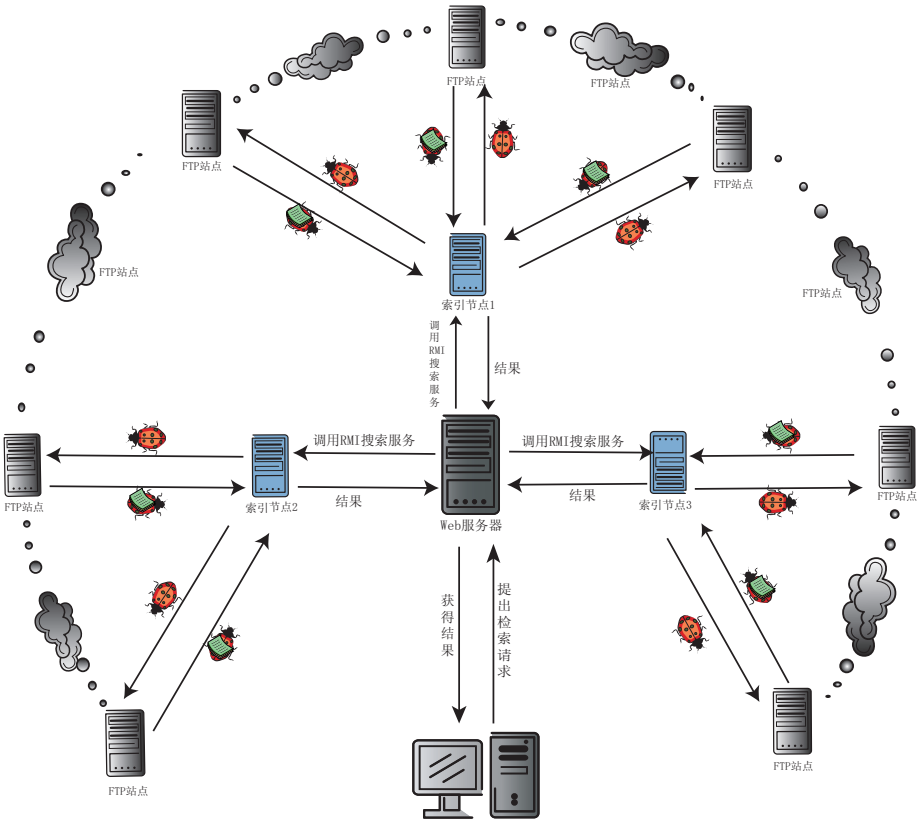

图 1: ROSA 架构

5 爬虫程序设计

5.1 流程简介

从效率方面考虑，我们使用了线程池来负责从各个 FTP 站点收集资源信息。每个站点对应一个资源收集任务，通过将任务放入线程池，我们可以最大程度的利用机器的性能和带宽。对于每个资源收集任务，工作流程如下：

1. 使用 Apache Common Net[6]工具中的 FTP API 连接站点。

2. 向站点发送 ls -lR 命令。

3. 得到文件列表，并逐条的返回给调用者（索引器），同时要统计一些属性，如文件数，视频文件数之类的。

图2给出了爬虫程序的架构。

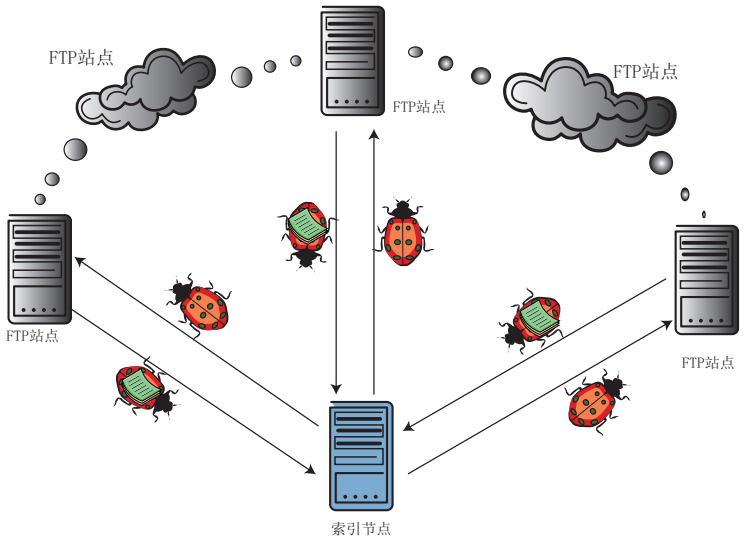

图 2: ROSA 爬虫

5.2 问题解决

这儿有一个问题，目前还无法很好解决，就是有些站点不支持 ls -lR 命令。对于这些站点目前的方法是检查在根目录发送 ls 命令和发送 ls -lR 命令返回条目数的关系，如果接近则认为不支持 ls -lR 命令，并将此信息存入数据库，下次就使用程序手动递归方式进行索引。但这种程序手动递归方式存在一个严重的偶发性超时问题，目前无法完美解决。

6 索引程序设计

6.1 分布式索引结构

考虑到缺乏专用的服务器，如果将所有的爬虫程序放在某一台机器上的话会影响正常的使用，所以我们考虑了分布的方式来进行索引和提供搜索服务。我们通过站点编号 mod 站点数 来给每个索引节点分配任务。每个节点都同时提供搜索服务，采用 RMI 的实现方式。之所以选择 RMI，是考虑到 RMI 的如下三个优点：

- 对其他框架的依赖性弱，RMI 框架一直都是 JDK 的标准组件
- 经济性好，通过整合开源工具，避免购买昂贵的授权
- 运行效率高，具有一定的可扩展性。比如 RMI 的序列化和压缩机制都可以进行自由的自定义扩展。

6.2 索引过程

此过程分为如下几个子过程。

1. **检查任务列表**

   对于每个站点都有更新周期，默认是一天，如果某个站点上次更新时间超过这个周期的话就进行一次全新索引。此处还有另外一个考虑，对于那些索引文件总数为 0 的站点不考虑索引周期而进行全新索引（这种情况可能是由于第一次索引失败或服务器站点故障）。索引一共有两份，一份是正在提供索引服务的索引，另一份是其备份。更新的时候是在备份上完成，更新完成后一次性将更新反映到正在使用的索引中。通过使用两份索引，我们的系统获得了较高的可靠性。

2. **索引准备**

   (a) 检查索引文件是否正常关闭，否则进行索引文件恢复。

   (b) 删除原有的此站点的索引。

   (c) 初始化 IndexWriter 对象并设置一些参数。

   (d) 初始化线程池。

3. **将任务放入线程池**
   每个任务负责一个站点的索引，索引成功则更新数据库信息，索引失败则记录此失败节点。

4. **善后工作**

   (a) 等待线程池任务执行结束。

   (b) 关闭 IndexWriter。

(c) 回滚失败节点的索引。因为我们在索引前已经将待索引站点的索引删除，所以此时我们需要从备份中（这个备份其实就是正在使用的索引库）恢复这部分索引。

(d) 更新索引库。为了不影响正在进行的查询，我们不是简单的将索引文件删除，而是使用 IndexReader 首先将更新了的站点的索引删除，然后再讲更新的站点的索引加入到这个库中。所以此处多次使用 IndexReader 和 IndexWriter。注意对以同一个索引文件这两个不能同时使用。具体过程如下：

i. 在正在使用的索引文件中删除被更新了的索引项。

ii. 使用 IndexWriter 重建索引库（不存在重建，存在只是写入）。

iii. 将上述备份中更新的信息加入到索引库中。

iv. 关闭 IndexReader 和 IndexWriter。

6.3 分词器

关于索引过程中比较重要的要数分词器，目前使用的是我在 StandardAnalyzer 基础上修改的分析器，并加上了 Snowball 词干提取过滤器。从目前的搜索来看似乎还有些不是很理想的因素，但对于一般的需求似乎已经足够了。

对 StandardAnalyzer 的修改主要是将一些原分词器没有分开的词分开，如 FTP 站点上经常在资源名称中包含句点，比如对于 [先知].Knowing.2009.BDRip.X264-TLF 这样的字符串，原分词器的分词结果为：“先” “知” “knowing.2009.bdrip.x264-tlf”，修改后的分词器的分词结果为：“先” “知” “know” “2009” “bdrip” “x” “264” “tlf”。通过这样的修改，能更针对地将 FTP 资源名称和路径分词，也便利了用户使用多种变化和自己喜好或擅长的搜索条件。

6.4 高级搜索功能设计

6.4.1 资源分类

分类是基于文件后缀，在爬虫获取资源时根据文件后缀自动将文件的分类属性置为相应分类。索引器会将这些分类信息存入索引文件中，这样用户就可以根据文件分类来进行搜索。具体的分类依据参看表1。

表 1: 分类说明

分类	意义	相应的文件后缀
video	视频	rm; rmvb; mpg; mpeg; mov; mtv; dat; wmv; avi; 3gp; dmv; divx; asf; vob; flv; mkv; swf
subtitle	字幕或歌词	srt; idx; sub; smi; ssa; lrc
audio	音频	wav; mp3; ra; rma; wma; asf; mid; midi; rmi; ogg; mod; ape; aiff; au; voc; vox
document text	文档 文本	pdf; doc; docx; chm; ppt; pptx; xls; xlsx; rtf tex; txt; html; htm; xml; log; ini; properties; prop; url; css
program	源码	java; c; cc; c++; cpp; cxx; h; hh; hpp; hxx; lisp; perl; pas; sh; pyo; a; so; lo; la; sql; class; js; o
image	图片	gif; jpg; jpeg; jpe; ico; mng; pbm; pgm; psd; png; pnm; ps; ppm; tif; tiff; bmp; xpm; eps; dcm; dicom
compress	压缩文件	gz; bz2; tar; zip; rar; ar; ear; jar; war; iso
executable	可执行文件	bat; exe; msi; dll; cab; com; sys; deb
unknown	未知	
directory	文件目录	

6.4.2 访问控制

如果你有一个 FTP 站点，但你不想将这个站点注册上来然后能被所有人搜索，那么你可以在我们的注册页面上将访问控制设定为你想要的字串，那么在高级搜索页面中可以输入这个字串来搜索你的站点，而不输入这个字串的用户是搜索不到你的站点的。默认的字串是 anybody，所以如果你想让你的站点只在小范围共享的话请确保你的字串不要为 anybody。

6.4.3 站点快照

通过在索引中加入不分词的父目录（parent）域，我们设计实现了 ROSA 的站点快照功能。这个功能便于用户在不登陆的情况下快速浏览 FTP 服务器上的资源文件。

6.5 ROSA 的索引域

表 2: ROSA 可搜索域

搜索域名称	意义
id	FTP 服务器在数据库中的 ID
server	FTP 服务器域名
username	登陆用户名
password	登陆密码
location	站点位置：校内，教育网，公网
port	端口号
access	访问控制
path	文件路径域
parent	文件父目录
name	文件名称域
date	文件更新时间
size	文件大小
ext	文件后缀名
cat	文件分类
updatetime	索引更新时间

其中搜索域 access 只有使用高级搜索功能时才可使用。updatetime 域目前没有提供高级搜索接口。默认的是 10 天内得到更新的索引都可以被搜索到。此外端口号，密码对搜索的帮助不大，用户名域可以用来区分同一机器上 FTP 服务器上的不同用户的索引。比较常用的是 name 和 path 域。

7 分布式搜索设计

由于我们采用的是分布式结构，所以需要一个所有提供搜索 RMI 服务的节点描述文件。这个文件我们是这样定义的：

```
<?xml version="1.0" encoding="UTF-8"?>
<nodes version="1.0">
    <node>
        <name>nodename1</name>
        <address>ipaddress1</address>
```

```
        <port>portnum</port>
    </node>
    <node>
        <name>nodename2</name>
        <address>ipaddress2</address>
        <port>portnum</port>
    </node>
</nodes>
```

从上面的例子可以容易的看出搜索节点的定义方式。rmi://address:port/name 的组合就是我们使用 lookup 时给出的参数。搜索的过程大致如下：

1. 获取所有节点的 Searchable 对象并封装到 ParallelMultiSearcher 中。
2. 使用 QueryParser 进行分析，这儿的分析器使用的是 PerFieldAnalyzerWrapper，它对除 path 和 name 之外的项使用关键字分析器（KeywordAnalyzer），而 path 和 name 使用索引时使用的分析器。
3. 从搜索参数中获取排序方式。
4. 从搜索参数中获取过滤器（分类查询，访问控制都是用过滤器方式实现的）。
5. 进行搜索。
6. 提取所需结果集（因为网页的分页显示方式，所以每次只是返回搜索结果的一部分）。
7. 在逐条提取结果集的时候同时进行高亮处理。

搜索结构图参看图3。

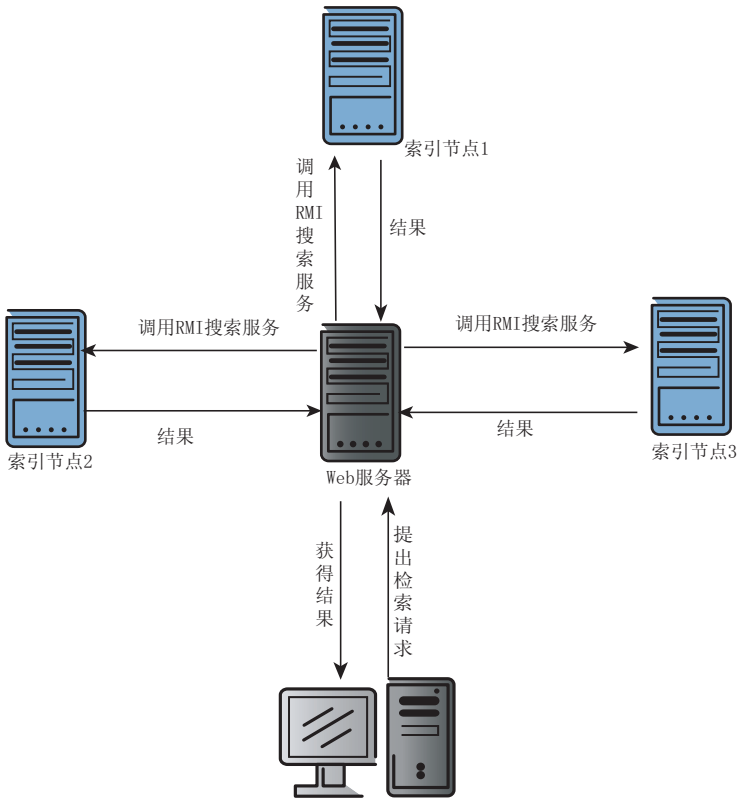

图 3: ROSA 分布式搜索架构简图

8 多点下载客户端设计

考虑到同一资源在不同 FTP 服务器上的多处出现，我们设计实现了多点下载客户端。通过对请求下载的资源进行查询资源文件的出现站点（目前根据完整文件名和文件大小进行判断），进行分片多点下载。热门的资源通常会有多份拷贝在不同服务器上，通过使用多点分片下载，可以极大提高下载速度。前期的 RMI 搜索服务在这个程序中得到了充分应用，下载程序可以通过 RMI 调用查询候选资源，实现多点下载。

8.1 设计思路

1. 获取下载链接接。如：ftp://ftp.askrosa.cn/test.txt。

2. 根据下载连接生成任务列表，如果下载链接是文件夹则递归生成所有待下

载文件。

3. 初始化线程池。
4. 将每个下载任务放入线程池。
5. 对于每个下载任务
6. (a) 使用 RMI 服务查询这个任务的候选站点资源。
   (b) 任务分片。
   (c) 初始化分片任务线程池。
   (d) 将每个分片放入分片任务线程池。
7. 等待传输结束，在等待过程中随时更新下载进度条显示。

对于每个分片任务的传输，如果发生失败，将其重新放回分片任务线程池。注意这儿有两种线程池，一种是针对每个下载任务（文件）的，另一个是负责下载每个任务（文件）的若干分片的。

8.2 设计要点

1. **FTP 连接池**
   由于程序中多处使用多线程下载，所以同时设计了 FTP 连接池，当需要 FTP 连接时，向连接池取出连接使用。连接池中请求的站点连接不存在，就初始化一个连接。当用完 FTP 连接时，将其放回连接池

2. **Applet 前端**
   考虑需要将程序放入网页中，所以使用了 Applet。由于 Applet 的沙箱问题，所以还使用了数字签名[7]，否则 Applet 没有权限访问本地资源和网络连接。

3. **线程池的使用**
   程序中要使用两个线程池，一个线程池针对每个下载任务，而每个下载任务都还有一个线程池来处理每个分片任务。

4. **文件并发写入**
   由于文件分片下载所以需要对文件并发写入，这通过结合 RandomAccessFile 和 FileChannel 可以实现[8]。仅仅使用 RandomAccessFile 是不可以并发写入的。

8.3 遗留问题

1. URL 编码问题，发现在 IE 中会出现 URL 乱码。
2. 对单个文件进行了分片，但对多个文件使用多线程目前还有问题没有解决，所以将线程数设为 1。在后续开发中需要解决。

9 系统配置文件设计

我们将系统的配置参数全放入到了配置文件中。用户可以通过修改配置文件而不需要修改代码来实现系统的参数设置。可配置的主要参数包括：

- 线程池大小
- 代理服务器
- 数据库连接
- 可用缓存
- 文件日期格式
- 索引文件存放地址
- 网络连接超时时间
- 索引文件超时时间（当某站点在设定时期内未能够更新的话它的资源将不会出现在搜索结果中，默认超时时间为 10 天）

10 系统服务安装

通过使用 Java Service Wrapper(Community)将索引过程和 RMI 远程调用服务封装成 Windows 系统服务，这样实现了系统的开机自动启动。双击 wrapper 文件夹下的 InstallRosa-NT.bat 可以安装这个系统服务。卸载使用 UninstallRosa-NT.bat。

11 日志系统

程序使用了 log4j 日志系统，日志存放在 logs 文件夹下面。rosa.log 是本程序的日志。wrapper.log 是 Java Service Wrapper 的日志。通过查看日志文件可以发现程序错误的位置和原因。对于有网络访问的程序，日志系统对系统维护有至关重要的作用。

12 查询统计和搜索提示

1. **热门搜索**
   通过对用户搜索历史 QueryStatistics 进行统计，给出一月，一年的搜索关键词频率排名。

2. **热门站点统计**
   通过统计用户对站点的点击操作，统计站点的热度。并给出站点热度的排名。

3. **访问统计**
   根据用户搜索历史 QueryStatistics，给出 24 小时系统负荷曲线图。

4. **搜索提示**
   根据数据库保存的用户搜索历史，我们设计实现了搜索建议功能。我们首先将这些搜索历史分词，然后使用 EdgeNGramTokenFilter 进行分割，最后对分割后的数据进行索引。在用户提交搜索关键字的时候我们可以使用上述的索引来搜索相似度最大的历史搜索作为搜索建议。

13 数据库设计

下面数据库中使用的字符串类型均为 VARCHAR 类型，括号中是最大的字符串长度。使用的数据库是 MySql。数据库自动生成工具是 lisptorq。

表 3: 站点信息(FtpSiteInfo)数据库表

名称	类型	说明
ID	自增整型	FTP 站点 ID，自动生成
server	字符串类型(128)	FTP 站点域名
address	字符串类型（128）	FTP 站点域名，目前长度不够，需要增长
verify	字符串类型（128）	用于验证站点添加者的权限，保证只有添加者可以修改站点信息
access	字符串类型（128）	默认为 anybody，否则在不给出这个字段的前提下是不会降这些站点的信息包含在搜索结果中
port	整型	FTP 站点端口号，默认 21
username	字符串类型（128）	FTP 站点用户名
password	字符串类型（128）	FTP 站点密码
encoding	字符串类型（128）	FTP 站点编码，默认“GBK”
admin	字符串类型（128）	FTP 站点管理员
contact	字符串类型（128）	FTP 站点管理员联系方式
description	字符串类型（5000）	FTP 站点描述信息
updateTime	Timestamp	站点最近一次更新时间
lastUpdateTime	Timestamp	站点最近一次上一次的更新时间
totalFileCount	整型	FTP 站点文件总数
crawlInterval	整型	FTP 站点爬行周期，默认 2 天
video	整型	FTP 站点视频文件总数
audio	整型	FTP 站点音频文件总数
subtitle	整型	FTP 站点字幕文件总数
document	整型	FTP 站点文档总数
text	整型	FTP 站点文本文件总数
program	整型	FTP 站点程序文件总数
image	整型	FTP 站点图像文件总数
compress	整型	FTP 站点压缩文件总数
executable	整型	FTP 站点可执行文件总数
directory	整型	FTP 站点目录文件总数
unknown	整型	FTP 站点未知文件类型的文件总数
speed	整型	FTP 站点限速
userslimit	整型	FTP 站点用户数限制
recursive	短整型	是否支持 ls lR 命令，默认 1

表3中几个字段的说明如下：

verify 字段 为了减少用户注册站点的表格项目，我们没有设置用户名密码，只是简单的设置了这个字段来检查用户权限，用户在注册的时

候需要输入两次确认自己的 verify 字段，当需要修改站点信息的时候需要提供此字段。

access 字段 考虑到某些用户可能将站点注册上了是为了小范围的共享，设计了这个字段有助于以最小的复杂度实现这个目标。这个字段的

默认值是 anybody，也就是说如果用户不修改这个字段默认值的话，她或他注册的站点将可以被所有人检索到，否则检索用户

要输入这个字段才能将对应的站点的搜索结果包含在最终的搜索结果中。这个功能的实现使用的是 lucene 搜索中的过滤器

(Filter)功能。用户可以使用高级搜索页面使用这个功能。

recursive 字段 对于部分站点，ls lR 命令返回的结果和 ls 命令的结果一样，这样就不能用这个命令来递归枚举所有文件。这个字段用来表

示该站点是否支持 ls lR 命令。如果支持这个字段为 1，否则为 0。此外，爬虫在检索站点，会根据 ls 命令和 ls lR 命令返回结果的数目是否十分

接近自动检查站点对 ls lR 命令的支持。

表 4: 资源请求(ResourceRequest)数据库表

名称	类型	说明
id	自增整型资源请求 ID	
nickname	字符串类型(512)	请求者昵称
resourcename	字符串类型(1024)	资源名称
email	字符串类型(128)	请求者邮件地址
time	Timestamp	请求时间
deadline	Timestamp	请求过期时间
display	小整型	是否显示在页面上，默认显示
state	小整型	状态，是否已经回复

表 5: 留言(Article)数据库表

名称	类型	说明
id	自增整型	留言 ID
author	字符串类型（512）	留言作者
time	Timestamp 留言时间	
title	字符串类型（512）	留言标题
content	字符串类型（10000）	留言内容
clickcount	整型	点击次数
ip	字符串类型（128）	留言 IP
verify	字符串类型（64）	验证字符串，可以用来删除和修改留言

表 6: 回复(ResponsePost)数据库表

名称	类型	说明
postid	自增整型	回复 ID
id	整型	外键，指向 Article 表中的 id
author	字符串类型（512）	回复作者
content	字符串类型（10000）	回复内容
time	Timestamp	回复时间
ip	字符串类型（128）	回复 IP
verify	字符串类型（64）	验证字符串，可以用来删除和修改回复

表 7: 历史用户数统计(HistoryUsers)数据库表

名称	类型	说明
id	自增整型	ID，主键
count	大整型	记录历史用户总量

表 8: 会话统计(Sessions)数据库表

名称	类型	说明
id	自增整型	ID，主键
createTime	大整型	用大整型保存 Session 创建时间
destroyTime	大整型	Session 销毁时间

表 9: 查询统计(QueryStatistics)数据库表

名称	类型	说明
id	自增整型	ID，主键
keyword	字符串类型（512）	查询关键字
time	Timestamp	查询提交时间

表 10: 查询统计结果(QueryStatisticsResult)数据库表

名称	类型	说明
keyword	字符串类型（512）	查询关键字
frequencey	整型	查询频率

14 系统遗留问题

- 关于站点可访问性的功能还未能实现，因为这要求实时的站点检测，所以难度较大，目前正在考虑实现方式。
- ROSA FTP 多点下载程序还存在很多不稳定因素，还需要仅以测试和改进。

参考文献

[1] Wikipedia. http://zh.wikipedia.org/zh-cn/Lucene.

[2] Wikipedia. http://zh.wikipedia.org/zh-cn/Struts.

[3] Ceki Gülcü. http://logging.apache.org/log4j/1.2/manual.html, 2002.

[4] JDom Organization. http://www.jdom.org/.

[5] Apache. http://jakarta.apache.org/oro/.

[6] Apache. http://commons.apache.org/net/.

[7] Sun Microsystems. http://java.sun.com/developer/onlineTraining/Programming/JDCBook/signed.html.

[8] Sun Microsystems. http://java.sun.com/j2se/1.4.2/docs/api/java/nio/channels/FileChannel.html.