

# 使用 Lucene 构建强大的 discuz 论坛搜索模块

版本：内测版 1.0

文档编写：[sunboyu@gmail.com](mailto:sunboyu@gmail.com)

发布博客：<http://www.sunboyu.cn>

讨论 QQ 群：41886598

## 目录

1、Lucene 介绍.....	1
2、IKAnalyzer 中文分词器介绍.....	2
3、针对 discuz 论坛的索引设计和索引创建.....	3
4、检索系统.....	4
5、PHP 调用 lucene 检索结果(Php-java-bridge 安装配置).....	5
6、自动任务的设计和部署(编写中).....	6
7、自定义词库(编写中).....	7

### 文章目的：

康盛旗下 discuz 论坛在国内社区中占有相当用户，随着论坛数据的增长，会出现各种的性能瓶颈。

其中中文论坛数据模糊检索在数据量超过一定水平后，其 mysql like 检索的方式显然不适合。本文既针对此问题而作，使用开源的分词检索工具：Lucene 进行改造，解决中文检索方面的问题。

# 1、Lucene 介绍

## 1、什么是 lucene

Apache Lucene 是一个开放源程序的搜寻器引擎，利用它可以轻易地为 Java 软件加入全文搜寻功能。Lucene 的最主要工作是替文件的每一个字作索引，索引让搜寻的效率比传统的逐字比较大大提高，Lucene 提供一组解读，过滤，分析文件，编排和使用索引的 API，它的强大之处除了高效和简单外，最重要的是使使用者可以随时应自己需要自订其功能。Lucene 是 apache 软件基金会项目组的一个子项目，是一个开放源代码的全文检索引擎工具包，即它不是一个完整的全文检索引擎，而是一个全文检索引擎的架构，提供了完整的查询引擎和索引引擎，部分文本分析引擎。Lucene 的目的是为软件开发人员提供一个简单易用的工具包，以方便的在目标系统中实现全文检索的功能，或者是以此为基础建立起完整的全文检索引擎。

## 2、Lucene 的作者

Lucene 的原作者是 Doug Cutting，他是一位资深全文索引/检索专家，曾经是 V-Twin 搜索引擎的主要开发者，后在 Excite 担任高级系统架构设计师，目前从事于一些 Internet 底层架构的研究。

## 3、Lucene 的历史

早先发布在作者自己的 <http://www.lucene.com/>，后来发布在 SourceForge，2001 年年底成为 apache 软件基金会 jakarta 的一个子项目。现在则是 apache 的顶级项目 <http://lucene.apache.org/>

## 4、Lucene 应用

apache 软件基金会的网站使用了 Lucene 作为全文检索的引擎；IBM 的开源软件 eclipse 也采用了 Lucene 作为帮助子系统的全文索引引擎；相应的 IBM 的

商业软件 Web Sphere 中也采用了 Lucene; 著名的 Jive 论坛使用了它; Eyebrows (EyeBrows 是目前 APACHE 项目的主要邮件列表归档系统) 邮件列表 HTML 归档/浏览/查询系统也使用了它。

Lucene 以其开放源代码的特性、优异的索引结构、良好的系统架构获得了越来越多的应用。

Cocoon:基于 XML 的 web 发布框架, 全文检索部分使用了 Lucene 到现在 lucene 已经有 C++、C#、Python 和 Perl 的版本更多关于 lucene 的应用见 这里:  
<http://wiki.apache.org/lucene-java/PoweredBy>

## 5、Lucene 能做什么

Lucene 使你可以为你的应用程序添加索引和搜索能力(这些功能将在 1.3 节中描述)。Lucene 可以索引并能 使得可以转换成文本格式的任何数据能够被搜索。在图 1.5 可以看出, Lucene 并不关心数据的来源、格式甚至它的语言, 只要你能将它转换为文本。这就意味着你可经索引并搜索存放于文件中的数据: 在远程服务器上的 web 页面, 存于本地文件系统的文档, 简单的文本文件, 微软 Word 文档, HTML 或 PDF 文件或任何其它能够提取出文本信息的格式。同样, 利用 Lucene 你可以索引存放于数据库中的数据, 提供给用户很多数据库没有提供的全文搜索的能力。一旦你集成了 Lucene, 你的应用程序的用户就能够像这样来搜索: +George +Rice - eat - pudding, Apple - pie +Tiger, animal:monkey AND food:banana 等等。利用 Lucene, 你可以索引和搜索 email 邮件, 邮件列表档案, 即时聊天记录, 你的 Wiki 页面……等等更多。

## 6、Lucene 资料

Lucene 主页: <http://lucene.apache.org/>

中文的 lucene 教程: <http://www.chedong.com/tech/lucene.html#intro>

写的很好的 lucene 书: lucene in action

luceneAPI: <http://lucene.zones.apache.org:8080/hudson/job/Lucene-Nightly/javadoc/>

lucene in action 示例代码:<http://www.manning.com/hatcher2>

lucene 的 wiki:

<http://wiki.apache.org/lucene-java/FrontPage?action=show&redirect=FrontPageEN>

## 7、Lucene 的优点

(1) 索引文件格式独立于应用平台。Lucene 定义了一套以 8 位字节为基础的索引文件格式，使得兼容系统或者不同平台的应用能够共享建立的索引文件。

(2) 在传统全文检索引擎的倒排索引的基础上，实现了分块索引，能够针对新的文件建立小文件索引，提升索引速度。然后通过与原有索引的合并，达到优化的目的。

(3) 优秀的面向对象的系统架构，使得对于 Lucene 扩展的学习难度降低，方便扩充新功能。

(4) 设计了独立于语言和文件格式的文本分析接口，索引器通过接受 Token 流完成索引文件的创立，用户扩展新的语言和文件格式，只需要实现文本分析的接口。

(5) 已经默认实现了一套强大的查询引擎，用户无需自己编写代码即使系统可获得强大的查询能力，Lucene 的查询实现中默认实现了布尔操作、模糊查询、分组查询等等。

## 8、Lucene 的周边

### Nutch vs Lucene

Lucene 不是完整的应用程序，而是一个用于实现全文检索的软件库。

Nutch 是一个应用程序，可以以 Lucene 为基础实现搜索引擎应用。

### Nutch vs Larbin

"Larbin 只是一个爬虫，也就是说 larbin 只抓取网页，至于如何 parse 的事情则由用户自己完成。另外，如何存储到数据库以及建立索引的事情 larbin 也不提供

### Nutch vs Larbin

"Larbin 只是一个爬虫，也就是说 larbin 只抓取网页，至于如何 parse 的事情则由用户自己完成。另外，如何存储到数据库以及建立索引的事情 larbin 也不提供

Nutch 则还可以存储到数据库并建立索引。

## 2、IKAnalyzer 中文分词器介绍

### 1、IKAnalyzer 中文分词器介绍

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。从 2006 年 12 月推出 1.0 版开始，IKAnalyzer 已经推出了 3 个大版本。最初，它是以开源项目 Luence 为应用主体的，结合词典分词和文法分析算法的中文分词组件。新版本的 IKAnalyzer3.0 则发展为面向 Java 的公用分词组件，独立于 Lucene 项目，同时提供了对 Lucene 的默认优化实现。

IKAnalyzer3.0 特性:

采用了特有的“正向迭代最细粒度切分算法“，具有 60 万字/秒的高速处理能力。

采用了多子处理器分析模式，支持：英文字母（IP 地址、Email、URL）、数字（日期，常用中文数量词，罗马数字，科学计数法），中文词汇（姓名、地名处理）等分词处理。

优化的词典存储，更小的内存占用。支持用户词典扩展定义

针对 Lucene 全文检索优化的查询分析器 IKQueryParser(作者吐血推荐)；采用歧义分析算法优化查询关键字的搜索排列组合，能极大的提高 Lucene 检索的命中率。

本方案在 IK Analyzer 3.2.3 and Lucene3.0 基础上进行部署。

发布地址：<http://linliangyi2007.javaeye.com/blog/667095>

## 2、作者相关信息

作者笔名：[linliangyi2007](#)

作者主页：<http://linliangyi2007.javaeye.com/>

作者邮箱：[linliangyi2005@gmail.com](mailto:linliangyi2005@gmail.com)

项目主页：<http://code.google.com/p/ik-analyzer/>

## 3、针对 discuz 论坛的索引设计和索引创建

Discuz 是国内用户量相当大的基于 PHP 语言开发的论坛系统，其优势自然不用我多说。但随着网站的发展和数据量的增大，一些弊病也逐渐凸现：

1、数据库性能的瓶颈

2、数据存储的瓶颈

3、数据检索的瓶颈

对于瓶颈 1，使用 mysql 的 replication 分担压力已经是很常见的方法。

对于瓶颈 2，使用分表，也能很大提升数据存储量。

对于瓶颈 3，似乎还没有什么好的方法。部分站长使用 google 的站内搜索，不过搜索的结果自己完全不可控。

在 discuz 最新的产品 discuz! X1 中，后台已经提供 sphinx 的支持，不过对于一些站长，sphinx 的部署和使用似乎也是很大的问题，在官方并没有看见详细的部署文档。

本文着重针对 discuz 论坛系统使用 lucence 对其主题、内容、发帖人进行检索，并附详细源码和服务器部署文档，希望大家能提出宝贵意见，我会针对大多数人的意见进行本方案的完善。大餐开始！

能看到这里的同学，我衷心感谢，希望在看分割线下边内容时候，先理解一下关键词的含义：分词、索引。

所有的 java 代码，大家不要纠结代码风格，这是很 PHP 的 java 代码，我会尽量写得规范，并且，我也会尽量发布比较兼容的 jar 包。如果大家有什么问题，可以进我的 QQ 群进行提问，待工作不忙的时候我会一一解答。

-----万恶的分割线-----

### 1、需要索引的字段

此方案只是一个演示方案，我们只针对普通帖子进行索引。

根据我公司论坛的经验，用户操作最多的搜索，即：1、针对标题的搜索 2、针对内容的搜索 3、针对某用户的搜索。因为，只要能针对以上三个条件进行索引即可。

在这里，我的索引包含以下几个字段：

Pid	postid	回复的 id, int 型
Fid	forumid	版块分类 id, int 型
Tid	threadsid	主题 id, int 型
Author	作者账号,	string 型
Subject	主题,	string 型
Dateline	日期,	int 型
Message	内容,	string 型

## 2、Java 代码

WebSearch.java

```
import java.io.*;
import java.lang.String;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.WhitespaceAnalyzer;
import org.apache.lucene.store.FSDirectory;
import org.wltea.analyzer.lucene.IKAnalyzer;
import org.wltea.analyzer.lucene.IKQueryParser;
import org.wltea.analyzer.lucene.IKSimilarity;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.util.Version;
import org.apache.lucene.search.Collector;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.Query;
import org.apache.lucene.search.ScoreDoc;
import org.apache.lucene.search.Scorer;
import org.apache.lucene.search.Searcher;
import org.apache.lucene.search.TopScoreDocCollector;

public class Index
{
    private MySQLConn Db = null;
    //构造函数
    public void Index()
    {
```

```

}
//建立 mysql 连接
public void getConn( String conndsn )
{
    this.Db = new MySqlConnection();
    this.Db.SetDsn( conndsn );
}
public static void main( String args[] )
{
    if( args.length==0 || args[0].equals("help") ){
        System.out.println("欢迎使用 Discuz-Lucene 索引创建程序。\\n 作者: sunboyu
email:sunboyu@gmail.com blog:http://www.sunboyu.cn");
        System.out.println(" 帮助 ( 参 数 说 明 ):java Index host=**** port=***
database=**** username=**** password=*** dateline=**** [help]");
        System.out.println("    host:主机地址, 默认 localhost");
        System.out.println("    port:主机地址, 默认 3306");
        System.out.println(" database:论坛数据库");
        System.out.println(" username:数据库账号");
        System.out.println(" password:数据库密码");
        System.out.println(" datelines:记录时间戳起始");
        System.out.println("    help:帮助");
        System.out.println("祝您使用愉快! ");
        System.exit(0);
    }else{
        //System.out.println( args[0] );
        //System.exit(0);
    }
    String[][] _args = new String[6][2];
    String host = "localhost";
    String port = "3306";
    String database = "";
    String username = "";
    String password = "";
    String datelines = "0";
    for(int i=0;i<args.length;i++){
        _args[i] = args[i].split("=");
        if(_args[i][0].equals("host")){
            host = _args[i][1];
        }
        if(_args[i][0].equals("port")){
            port = _args[i][1];
        }
        if(_args[i][0].equals("database")){
            database = _args[i][1];

```

```

    }
    if(_args[i][0].equals("username")){
        username = _args[i][1];
    }
    if(_args[i][0].equals("password")){
        password = _args[i][1];
    }
    if(_args[i][0].equals("datelines")){
        datelines = _args[i][1];
    }
}

Index index = new Index();
String dsn = "jdbc:mysql://" + host + ":" + port + "/" + database;

index.getConnection( dsn ); // "jdbc:mysql://localhost:3306/newbbs"
index.Db.SetUserPass( username , password );
index.Db.Conn();
String sql = "SELECT pid,fid,tid,author,subject,dateline,message,authorid FROM posts
ORDER BY pid ASC LIMIT 100000";

index.Db.sqlQuery( sql );
try
{
    File d = new File("./post");
    if (d.exists()==false) {
        d.mkdir();
    }
    IndexWriter writer = new IndexWriter( FSDirectory.open(new File("./post")) , new
IKAnalyzer() , IndexWriter.MaxFieldLength.LIMITED );
    writer.setUseCompoundFile(true);
    while( index.Db.rs.next() )
    {
        String pid      = index.Db.rs.getString( "pid" );
        String fid      = index.Db.rs.getString( "fid" );
        String tid      = index.Db.rs.getString( "tid" );
        String author   = index.Db.rs.getString( "author" );
        String authorid = index.Db.rs.getString( "authorid" );
        String subject  = index.Db.rs.getString( "subject" );
        String dateline = index.Db.rs.getString( "dateline" );
        String message  = index.Db.rs.getString( "message" );
        Document doc = new Document();
        Field f1 = new Field("pid",pid,Field.Store.YES,Field.Index.NOT_ANALYZED);
        Field f2 = new Field("fid",fid,Field.Store.YES,Field.Index.NOT_ANALYZED);

```

```

        Field f3 = new Field("tid",tid,Field.Store.YES,Field.Index.NOT_ANALYZED);
        Field          f4          =          new
Field("author",author,Field.Store.YES,Field.Index.NOT_ANALYZED);
        Field          f5          =          new
Field("authorid",authorid,Field.Store.YES,Field.Index.NOT_ANALYZED);
        Field f6 = new Field("subject",subject,Field.Store.YES,Field.Index.ANALYZED);
        Field          f7          =          new
Field("dateline",dateline,Field.Store.YES,Field.Index.NOT_ANALYZED);
        Field          f8          =          new
Field("message",message,Field.Store.YES,Field.Index.ANALYZED);

        doc.add( f1 );
        doc.add( f2 );
        doc.add( f3 );
        doc.add( f4 );
        doc.add( f5 );
        doc.add( f6 );
        doc.add( f7 );
        doc.add( f8 );
        System.out.println( pid );
        try
        {
            writer.addDocument( doc );
        }
        catch (Exception e)
        {
            System.out.println("Error : " + e.toString());
        }
    }
    writer.close();
}
catch(Exception e)
{
    System.out.println("Error : " + e.toString());
}
}
}

```

## 4、检索系统

### 1、主题查询

标题搜索

根据标题搜索是 dz 最常用的搜索，此动作占据搜索动作一半以上。

搜索条件：where first = 1 and subject like '%keywords%'

### 2、内容查询

大部分的网站都屏蔽了帖子内容的查询，因为这个动作——太浪费资源了，极易锁表，所以，此功能基本阉割。不过对于用户来说，此功能是进行网站信息检索最好的工具，因为，我们一定要把此功能做上。

搜索条件：where message like '%keywords%'

### 3、根据作者账号查询

### 4、根据作者 uid 查询

同上，根据 author 或者 authorid 字段进行查询，此功能在查看某用户的主题和回复的时候常用。既然索引了，也改造这两条语句、

搜索条件：where author = '\$author'      where authored = \$authorid

## 5、Java 代码

```
import java.io.*;
import java.lang.Math.*;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.index.IndexReader;
import org.apache.lucene.search.Searcher;
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.util.Version;
import org.apache.lucene.search.Query;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.TopScoreDocCollector;
import org.apache.lucene.search.TopDocs;
import org.apache.lucene.search.ScoreDoc;
```

```

import org.apache.lucene.document.Document;
import org.wltea.analyzer.lucene.IKAnalyzer;
import org.wltea.analyzer.lucene.IKQueryParser;
import org.wltea.analyzer.lucene.IKSimilarity;

import org.apache.lucene.search.BooleanClause;
import org.apache.lucene.search.BooleanQuery;

//import org.apache.lucene.search.TermQuery;
import org.apache.lucene.search.BooleanClause.Occur;

public class WebSearch
{
    public String search(String words,String field,int pagesize,int start) throws Exception
    {
        String result = "<?xml version=\"1.0\" encoding=\"utf-8\"?>\r\n";
        result = result + " <root>\r\n";

        FSDirectory dir = FSDirectory.open(new File("./post"));

        IndexSearcher searcher = new IndexSearcher(dir);
        searcher.setSimilarity(new IKSimilarity());

        //Query query = IKQueryParser.parse(field, words);

        BooleanQuery resultQuery = new BooleanQuery();
        Query query = null;
        if (field.equals("subject"))
        {
            query = IKQueryParser.parse("subject", words);
            resultQuery.add(query,Occur.MUST);
            //query = IKQueryParser.parse("first","1");
            //resultQuery.add(query,Occur.MUST);
        }
        else if (field.equals("message"))
        {
            query = IKQueryParser.parse("message", words);
            resultQuery.add(query,Occur.MUST);
        }
        else if (field.equals("author"))
        {
            query = IKQueryParser.parse("author", words);

```

```

        resultQuery.add(query,Occur.MUST);
    }
    else if (field.equals("authorid"))
    {
        query = IKQueryParser.parse("authorid", words);
        resultQuery.add(query,Occur.MUST);
    }

    TopDocs collector = searcher.search(query ,start+pagesize);
    result = result + "    <hit>" + collector.totalHits+"</hit>\r\n";
    ScoreDoc[] hits = collector.scoreDocs;

    if(collector.totalHits<start){
        result = result + "    <root>\r\n";
    }
    result = result + "    <recordlist>\r\n";
    for(int i=start;i<Math.min(start+pagesize,collector.totalHits);i++)
    {
        Document doc = searcher.doc(hits[i].doc);
        result = result + "        <item>\r\n";
        result = result + "            <pid>" + doc.get("pid") + "</pid>\r\n";
        result = result + "            <subject>" + doc.get("subject") + "</subject>\r\n";
        result = result + "            <message><![CDATA[" + doc.get("message") + "]]></message>\r\n";
        result = result + "            <authorid>" + doc.get("authorid") + "</authorid>\r\n";
        result = result + "        </item>\r\n";
    }
    result = result + "    </rs>\r\n";
    result = result + "    <list>" + collector.totalHits + "</list>\r\n";
    result = result + "    <root>\r\n";
    return result;
}
}

```

# 5、PHP 调用 lucene 检索结果 (Php-java-bridge 安装配置)

## 1、环境初始化

环境初始化细节暂不细数，标准的 linux+nginx+php+mysql 环境，具体可参考如下文章：  
<http://blog.s135.com/post/366/>

不过我稍加修改，使用软件的版本有所不同。我习惯使用 mysql5.0.22,php5.2.6,安装目录皆为/opt/\*\*\*

Java 环境：CLASSPATH 变量里需要加载以下类库：

```
mysql-connector-java-3.1.14-bin.jar  
IKAnalyzer3.2.3Stable.jar  
lucene-core-3.0.0.jar
```

## 2、Php-java-bridge 安装

下载 Php-java-bridge 安装包：

[http://downloads.sourceforge.net/project/php-java-bridge/RHEL\\_FC%20SecurityEnhancedLinux/php-java-bridge\\_5.5.4.1/php-java-bridge\\_5.5.4.1.tar.gz?use\\_mirror=cdnetworks-kr-2&ts=1280125079](http://downloads.sourceforge.net/project/php-java-bridge/RHEL_FC%20SecurityEnhancedLinux/php-java-bridge_5.5.4.1/php-java-bridge_5.5.4.1.tar.gz?use_mirror=cdnetworks-kr-2&ts=1280125079)

```
wget http://labs.renren.com/apache-mirror/ant/binaries/apache-ant-1.8.1-bin.tar.gz  
tar -zxvf apache-ant-1.8.1-bin.tar.gz  
mv apache-ant-1.8.1 /opt/ant  
  
tar -zxvf php-java-bridge_5.5.4.1.tar.gz  
cd php-java-bridge-5.5.4.1/  
/opt/php-5.2.6/bin/phpize  
./configure --with-php-config=/opt/php-5.2.6/bin/php-config --with-java=/opt/jdk  
make && make install  
  
#安装完，在 php 扩展目录里应该出现以下文件： java-bright.jar java-bright-war java.so  
php-script.jar Runjavabright script-api.jar  
  
修改 php.ini
```

```
extension_dir = "/opt/php-5.2.6/lib/php/extensions/no-debug-non-zts-20060613/"
extension=java.so
```

启动 java-bright 后台进程 注意，要以 htdocs 为当前目录启动

```
nohup java -jar /opt/php-5.2.6/lib/php/extensions/no-debug-non-zts-20060613/JavaBridge.jar
SERVLET_LOCAL:8080 3 >> /var/log/javabridge.log &
```

重启 php-fpm 后，查看 phpinfo(),应该会出现以下信息:

```
java
java support  Enabled
java bridge   5.5.4.1
```

在网站根目录下建立以下文件，运行，如果出现预期结果，则配置成功:

```
require_once("http://127.0.0.1:8080/JavaBridge/java/Java.inc");
$system = new Java('java.lang.System');
$s = new Java("java.lang.String", "php-java-bridge config...<br><br>");
print 'Java version='.$system->getProperty('java.version').' <br>';
print 'Java vendor='.$system->getProperty('java.vendor').' <br>';
print 'OS='.$system->getProperty('os.name').' ' .
$system->getProperty('os.version').' on ' .
$system->getProperty('os.arch').' <br>';
$formatter = new Java('java.text.SimpleDateFormat',
"EEEE, MMMM dd, yyyy 'at' h:mm:ss a zzzz");
print $formatter->format(new Java('java.util.Date'));
```

### 3、php 调用程序

PHP 调用，需要把 WebSearch.java 打包为 WebSearch.jar，放到网站根目录下，方可调用。

```
<?php
header("content-type:text/html; charset=utf-8");
if(empty($_GET['keyword'])){
    echo "<form method='get'><input type='text' name='keyword' /><input type='hidden' name='start' value='0'
/><input
type='hidden' name='count' value='10' /><input type='submit' value='ok'></form></form>";
    exit;
}
require_once("http://127.0.0.1:8080/JavaBridge/java/Java.inc");
java_require("/opt/nginx7/html");
error_reporting(2047);
$search = new Java("WebSearch");
$rs = java_values(
    $search->search( $_GET['keyword'] , "message" , $_GET['count'] ,
```

```
$_GET['start']    );  
echo $rs;
```