PostgreSQL 中文全文索引技术研究与实现

战 疆　冯月利　王 珊

(中国人民大学 信息学院，北京 100872)

摘要：开放源码关系数据库 PostgreSQL 目前还不支持对中文的全文索引.文章研究和分析了 PostgreSQL 的全文索引技术(TSearch2)，通过对其核心函数的重写和配置文件的修改，将中文分词技术引入到了 PostgreSQL 的全文索引技术中，并加入了去除中文无用词的功能，从而首先实现了 PostgreSQL 的中文全文索引.

关 键 词：中文全文索引；中文分词；PostgreSQL；TSearch2

中图分类号：TP316.9　　**文献标识码**：A　　**文章编号**：1671-4512(2005)S1-0213-04

Research and implementation of full text index on Chinese in PostgreSQL

Zhan Jiang　Feng Yueli　Wang Shan

Abstract: Open-source RDBMS PostgreSQL does not support Full Text Index (FTI) on Chinese at present. The FTI technology (TSearch2) in PostgreSQL is analyzed in this paper. Further, the Chinese Word Segmentation technology is merged into TSearch2 by rewriting its key functions and modifying its configuration files. Also the function of omitting frequent and content-free Chinese stop words is implemented. The Full Text Index on Chinese in PostgreSQL is firstly implemented.

Key words: full text index; Chinese word segmentation; PostgreSQL; TSeaech2

Zhan Jiang　Doctoral Candidate; School of Information, Renmin University of China, Beijing 100872, China.

全文检索技术通过发送关键词来实现对信息的查询[1].开放源码的关系数据库系统 PostgreSQL 提供了和数据库引擎紧密集成的文本搜索引擎扩展，可以对数据库中的文本建立全文索引(即倒排表索引).但它提供的全文索引均不能对中文进行有效的支持.本文在 PostgreSQL 的全文索引机制的基础上，引入中文分词功能到该数据库系统中，实现了 PostgreSQL 数据库的中文全文索引功能.

1 PostgreSQL 全文索引的建立和使用

1.1 PostgreSQL 的全文索引模块——TSearch2

PostgreSQL 的全文索引功能由其 TSearch2 模块实现[2].TSearch2 模块包含一种新型的数据类型——tsvector，利用 tsvector 来实现全文索引(即倒排索引)，还可对关键词进行查找.tsvector 是由单词及其在文档中的位置组成的记录，用一种特殊的结构加以组织，便于快速访问和查找.

1.2 PostgreSQL 全文索引的建立

为数据库建立全文索引，即为全文索引做准备包括两个步骤①：**a**. 将文档划分为一系列的单词.这一步将进行词干抽取，同时抽取出词在整个字段中出现的所有位置，叫词位 lexeme.将 lexeme 存放在表的类型为 tsvector 的新的一列中.在建立全文检索时，使用函数 to_tsvector()将文本段分解成词位.**b**. 按 lexeme 对文档建索引.用 GIST 对表的 tsvector 列建索引②.

1.3 执行全文索引查询

文档被索引后，执行搜索操作包括：**a**. 在进

收稿日期：2005-07-10.

作者简介：战　疆(1970-)，男，博士研究生；北京，中国人民大学信息学院 (100872).

E-mail: zhanjiang@ruc.edu.cn

基金项目：国家自然科学基金资助项目 (60473069).

① http://www.sai.msu.su/~megera/oddmuse/index.cgi/tsearch-v2-intro

② http://www.sai.msu.su/~megera/postgres/gist/TSearch/V2/docs/TSearch_V2/Readme.html

行查询之前,还要将要查询的关键词变成 tsquery 类型.数据类型 tsquery 类似于 tsvector,用于表示查询关键词的词位. TSearch2 只在 tsquery 和 tsvector 之间进行精确匹配查找.将查询关键字变成词位可用 to_tsquery()函数完成. **b**. 取回符合查询条件的文档.

用操作符@@进行全文检索的查询,使用方法是:tsvector @@ tsquery.

2 TSearch2 的实现机制

TSearch2 模块提供文件"TSearch2.sq"为全文索引做准备.在数据库为全文索引作完准备以后就会生成 4 张表:pg_ts_cfg,pg_ts_cfgmap,pg_ts_dict 和 pg_ts_parser.

函数 to_tsvector()将文档分解成 tsvector 的过程分为两个阶段:第一阶段用解析器(parse)将文档分成短的叫做 token 的文本序列. Token 通常是单词、空格和标点符号.第二阶段每个 token 或者被抛弃或者通过字典(dictionary)转换成词位.词位被收集起来形成 vector 并返回.

使用哪个解析器和使用哪个字典由这 4 张表定义.用户可以通过 4 张表对解析器和字典进行配置以实现需要的功能.

A. pg_ts_cfg 记录了 TSearch2 现有的配置:用哪个解析器(parser)将文档分解成 tokens;用哪个字典(dictionary)对 tokens 进行词干抽取.改进以前,该表中有 3 种配置:**a**. default:使用 en_stem 处理拉丁语系文字,使用 simple 字典处理其他.**b**. default_russian:使用 en_stem 处理拉丁语系文字,使用 ru_stem 处理非拉丁文字,使用 simple 字典来处理其他.**c**. simple:用 simple 字典处理单词和数字,既不去掉无用词也不转换它们.

在 TSearch2 中,只有缺省(default)的一种 parser,它能处理大多数 plaintext 和 HTML 文档.

B. pg_ts_cfgmap. TSearch2 通过 pg_ts_cfgmap 表来查找当前所选配置的具体信息.表中 prs_name 列定义该配置使用哪一个 parser.一旦 parser 把文档分解成 tokens,没有列出类型的 token 被抛弃掉.用整数来记录其余 tokens 的位置,并通过字典提取词干.在某种配置下,哪种类型的 token 抽取词干时参照的是哪个字典的配置定义在 pg_ts_cfgmap 表中.

在 default 类型的配置下,拉丁文字使用{en_stem}来抽取词干,并参照 contrib/english.stop 去掉无用词.

C. pg_ts_dict. 字典把文本 tokens(通常由 parser 生成)作为输入.返回在 token 基础上生成的词干. TSearch2 带有的词典包括:**a**. simple——只是把大写字母变成小写字母.**b**. en_stem——用 English Snowball stemmer 进行词干抽取.**c**. ru_stem——用 Russian Snowball stemmer 进行词干抽取.

D. pg_ts_parser. PostgreSQL 仅有一种名为 default 的 parser.

3 中文全文检索

要把中文语句按"词"进行索引,首先应将语句切分成词①.本方法是利用已有的中文分词系统的分词功能,通过将其引入 TSearch2 模块,从而实现 PostgreSQL 的中文全文索引.经过比较,决定使用中科院计算所的汉语词法分析系统 ICTCLAS.中科院分词系统提供的是一个动态链接库 ICTCLAS.dll 和相应的概率词典——data 目录.开发者可以完全忽略汉语词法分析,直接在自己的系统中调用 ICTCLAS.ICTCLAS.dll 中提供函数 ParagraphProcess(),用于对汉语句子和段落进行分词,而分词的依据就是字典——data 目录.data 目录必须和 ICTCLAS.dll 放在同一目录下.

3.1 ICTCLAS 的 API 函数加入

在没有中文分词功能的 TSearch2 系统中,用空格和西文标点符号来区分单词,因此会将没有空格的中文句子识别成一个单词.

让 TSearch2 支持中文分词,就要在 TSearch2 的函数库中增加一个函数(不妨也叫做 ParagraphProcess())实现中文分词功能,使得该函数像 to_tsvector()一样在 SQL 语句中使用.这样事先将中文字符串进行分词,词与词之间加上空隔,就可以像英文单词一样对它进行索引和匹配了.

TSearch2 中使用的 to_tsvector()等函数均定义在文件 contrib/tsvector.c 中,因此在 tsvector 中加入了一个新的函数 ParagraphProcess(),该函数调用 ICTCALS.dll 中的分词函数 ParagraphProcess().

在 TSearch2 中添加的分词函数的代码如下:

Datum ParagraphProcess (PG_FUNCTION_

① [GiST development site] http://www.sai.msu.su/~megera/postgres/gist.

```
ARGS)
{text * in=PG_GETARG_TEXT_P(0);
  int len;
  char * s;
  len=VARSIZE(in)-VARHDRSZ;
  s=palloc(len+1);
  memcpy(s, VARDATA(in), len);
  *(s+len)='\0';
  bool end1=true;
  HINSTANCE hinstLib;
  MYPROC1 ProcAddr1;
  MYPROC2 ProcAddr2;
  int fFreeResult;
  bool fRunTimeLinkSuccess=FALSE;
  char * splitWords=(char *) palloc(4 *
strlen(s)+1);
  text * out;
//Get a handle to the DLL module.
  hinstLib=LoadLibrary("ICTCLAS.dll");
  //If the handle is valid, try to get the func-
tion address.
  if (hinstLib ! =NULL)
  {
ProcAddr1 = (MYPROC1) GetProcAddress
(hinstLib, "Init");
  if (fRunTimeLinkSuccess=(ProcAddr1 !
=NULL))
    {
    end1=(ProcAddr1)(0,0);
  if (end1)
  {
  ProcAddr2=(MYPROC2) GetProcAddress
(hinstLib, "ParagraphProcess");
  if (fRunTimeLinkSuccess=(ProcAddr2 !
=NULL))
    (ProcAddr2) (s, splitWords);
  }
}
//Free the DLL module.
fFreeResult=FreeLibrary(hinstLib);
}
out=char2text(splitWords);
pfree(splitWords);
PG_RETURN_TEXT_P(out);
}.
```

此外，还要在 TSearch2.sql 中定义并创建该函数.在 tsearch2.sql 中加入这段：

```
CREATE FUNCTION ParagraphProcess
(text)
RETURNS text
AS '$libdir/TSearch2', 'ParagraphProcess'
LANGUAGE 'C' with (isstrict, iscachable).
```

作了上述改动，并重新编译 TSearch2 模块后，函数 ParagraphProcess 就真正地作为库函数被加入了 TSearch2 模块，在使用前，还必须把 ICTCLAS 的概率辞典(data 目录)和 ICTCLAS.dll 放在数据库的当前目录下.

要检验中文全文索引的可用性，执行下面的查询：

```
SELECT to_tsvector('default', Paragraph-
Process('巴拿马和美国都是国家地区，汉族是一
个民族.')).
```

查询结果如下：

```
to_tsvector
.....
'.':13 ',':8 '都':4 '和':2 '是':5 '地
区':7 '国家':6 '汉族':9 '美国':3 '民族':12
'一个':11 '巴拿马':1.
```

说明中文分词功能已经是可用的了.能够进行正确的分词，是中文全文索引的关键所在，此后，中文全文索引的创建、查询和维护都跟英文是一样的.

3.2　TSearch2 功能改进

TSearch2 在建立英文全文索引时，会去掉无用词，如 is,a,our 等.中文中也存在许多无用词，如"的"、"一个"、"是"和标点符号等.要增加中文的无用词过滤功能，需要改动上面介绍的 TSearch2 模块中与配置相关的 4 张表：

a. pg_ts_cfg

在 pg_ts_cfg 表中新建一个用于处理中文的配置 default_ch，parser 和 locale 分别是 default 和 C.

```
insert into pg_ts_cfg values('default_ch','
default','C');
```

b. pg_ts_dict

仿照 contrib/english.stop 的格式，在 contrib/下创建并书写文件 chinese.stop，把中文无用词(标点符号，助词等)加进去，每个词或符号占一行.

在 pg_ts_dict 中新建一个词典 cn_stem，用于中文抽取词干，仿照 en_stem，cn_stem 各列的值如下：

Insert into pg_ts_dict values ('cn_stem', 'snb_en_init(text)', 'contrib/Chinese.stop', 'snb_lexize(internal,internal,integer)', 'Chinese Stemmer. Snowball.');

c. pg_ts_cfgmap

修改pg_ts_cfgmap,将新配置default_ch的每种类型所对应字典信息插进去.部分操作如下:

insert into pg_ts_cfgmap values('default_ch','lword','{en_stem}');

insert into pg_ts_cfgmap values('default_ch','nlword','{cn_stem}');

insert into pg_ts_cfgmap values('default_ch','word','{cn_stem}');

insert into pg_ts_cfgmap values('default_ch','email','{simple}');

insert into pg_ts_cfgmap values('default_ch','nlpart_hword','{cn_stem}');

insert into pg_ts_cfgmap values('default_ch','lpart_hword','{en_stem}');

……

注意:nlword, word, nlpart_hword, hword, nlhword等类型的词都要参照cn_stem字典;lword, lpart_hword, lhword等类型都要参照en_stem字典;其他的类型都参照simple字典.可见,TSearch2可以支持中英文混合查找,中文和英文类型的词分别参照相应cn_stem和en_stem字典即可.

d. pg_ts_parser

使用原来的parser,所以对该表无需作改动.

发出以下查询对结果进行测试:

select to_tsvector('default_ch', ParagraphProcess('北京是中国的首都,位于中国的北部.')).

得到结果:

to_tsvector

.....

'北部':10 '北京':1 '首都':5 '位于':7 '中国':3,8'.

可以看到,无用词"是"、"的"和逗号等都已被去掉了.

参 考 文 献

[1] 文继军,王 珊.SEEKER:基于关键词的关系数据库信息检索[J].软件学报,2005,16(7):1 270—1 281

[2] 张华平,刘 群.基于*N*-最短路径方法的中文词语粗分模型[J].中文信息学报,2002(5):77—84

PostgreSQL中文全文索引技术研究与实现

作者： 战疆， 冯月利， 王珊， Zhan Jiang， Feng Yueli， Wang Shan
作者单位： 中国人民大学,信息学院,北京,100872
刊名： 华中科技大学学报（自然科学版） ISTIC EI PKU
英文刊名： JOURNAL OF HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY(NATURE SCIENCE)
年，卷(期)： 2005,33(z1)
被引用次数： 2次

参考文献(5条)

1. 文继军. 王珊 SEEKER:基于关键词的关系数据库信息检索[期刊论文]-软件学报 2005(07)

2. 张华平. 刘群 基于N-最短路径方法的中文词语粗分模型[期刊论文]-中文信息学报 2002(05)

3. 查看详情

4. 查看详情

5. 查看详情

本文读者也读过(10条)

1. 刘长浩. 孙玉芳. LIU Chang-Hao. SUN Yu-Fang PostgreSQL请求优化机制研究[期刊论文]-计算机科学2005,32(4)

2. 宋磊. 王静文. SONG Lei. WANG Jingwen PostgreSQL数据库性能优化[期刊论文]-电脑编程技巧与维护2009(16)

3. 郭龙江. 李金宝 PostgreSQL分析器的研究[期刊论文]-黑龙江大学自然科学学报2001,18(4)

4. 王黎维. 彭智勇. 林兰佳. 杨巍. 邹现军 PostgreSQL事务处理的分析与扩展[会议论文]-2003

5. 邵秀丽. 张琳. 田振雷. SHAO Xiu-li. ZHANG Lin. TIAN Zhen-lei PostgreSQL在异构数据集成中间件中的应用研究[期刊论文]-计算机工程与设计2006,27(21)

6. 罗昌明. 王朝坤. 王建民 基于PostgreSQL的执行计划缓存研究与实现[会议论文]-2006

7. 阮宏一. 彭智勇. 夏汇川. Ruan Hongyi. Peng Zhiyong. Xia Huichuan PostgreSQL数据库运行状态数据的统计收集机制分析[期刊论文]-计算机应用与软件2007,24(6)

8. 张孝. Zhang Xiao PostgreSQL中基于ACL的数据访问控制技术[期刊论文]-计算机应用与软件2007,24(9)

9. 胡巧巧. 王建民. 叶晓俊. HU Qiao-Qiao. WANG Jian-Min. YE Xiao-Jun PostgreSQL数据库预取算法研究[期刊论文]-计算机科学2006,33(3)

10. 古锐. 亓伟. 叶晓俊. GU Rui. QI Wei. YE Xiaojun 表空间存储策略在PostgreSQL中的研究与实现[期刊论文]-计算机工程2006,32(16)

引证文献(3条)

1. 刘淑梅. 夏亮. 许南山 Postgresql数据库集群在主题网络爬虫的应用[期刊论文]-计算机系统应用 2010(12)

2. 苗元亮. 滕至阳. 卢飞 Web文档管理系统与NAS结合的实现[期刊论文]-计算机技术与发展 2010(2)

3. 任宏萍. 黄晟 PostgreSQL存储引擎多线程化的研究与实现[期刊论文]-计算机与现代化 2013(9)