Discuz3 开启 Sphinx 全文搜索

作者：bangbang
原文链接：https://ld246.com/article/1521018075018
来源网站：链滴

Sphinx简介

Sphinx是一个基于SQL的全文检索引擎，可以结合MySQL,PostgreSQL做全文搜索，它可以提供比数库本身更专业的搜索功能，使得应用程序更容易实现专业化的全文检索。Sphinx特别为一些脚本语言计搜索API接口，如PHP,Python,Perl,Ruby等，同时为MySQL也设计了一个存储引擎插件。

Sphinx是独立的搜索服务端，不依赖MySQL，当Sphinx和MySQL结合部署时，Sphinx的数据来源MySQL。服务器安装Sphinx，由sphinx.conf配置文件指定Sphinx的数据源，如何读取MySQL的数内容，设置Sphinx对MySQL数据库的哪个表哪些字段建立索引，索引的返回数据必须是数值型。

Sphinx单一索引最大可包含1亿条记录，在1千万条记录情况下的查询速度为0.x秒（毫秒级）。Sphin创建索引的速度为：创建100万条记录的索引只需 3～4分钟，创建1000万条记录的索引可以在50分内完成，而只包含最新10万条记录的增量索引，重建一次只需几十秒。

正确使用Sphinx搜索数据的操作方式主要有三种：

1、命令行的search工具：/usr/local/webserver/sphinx/bin/search –i threads test

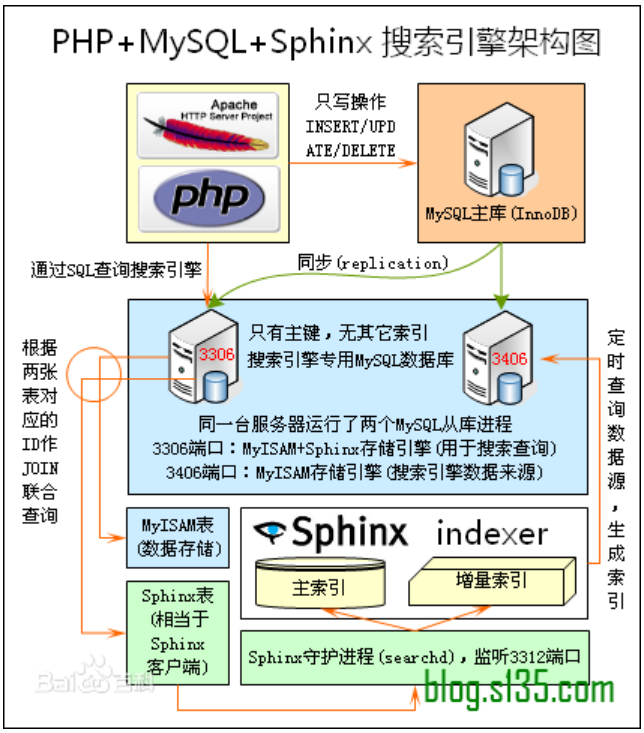
2、php的api接口查询：原理是直接用fsockopen连接端口，传递数据取得返回结果。 Sphinx官方经提供php的api接口，可以include api查询（本方案以该查询方法为主），也可以将其源代码编译成hp扩展而无需include。

3、在mysql中将Sphinx安装为SphinxSE存储引擎，通过SphinxSE方式调用Sphinx。

因Sphinx搜索结果只返回INT类型数据，部署Sphinx搜索的核心是由搜索入口（search.php）提交的键词到Sphinx中搜索，Sphinx返回对应的tid、pid等信息，再依据tid、pid到cdb_threads或者cdb_osts中搜索，得到结果集展示在页面上。

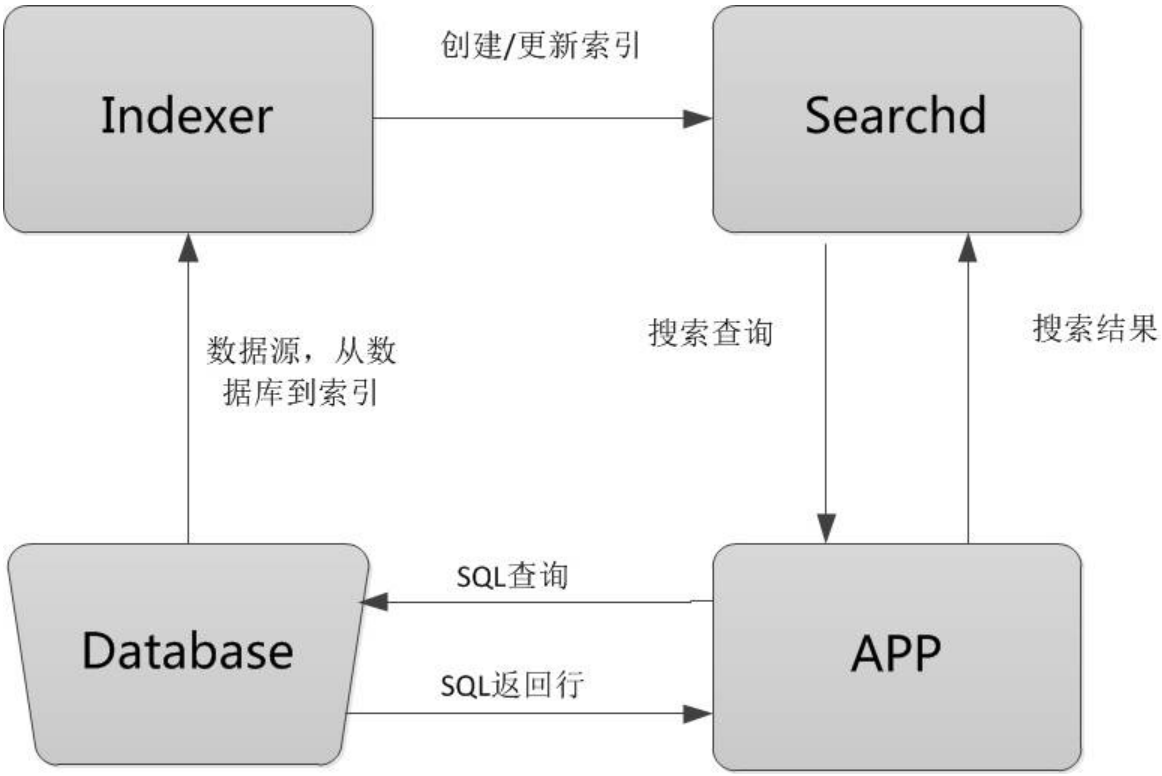
Sphinx的搜索速度非常快，而tid/pid都是主键查询，总体来说虽然用了多次查询，但是速度仍然非快。

Sphinx全文检索方案架构图



原文链接：Discuz3 开启 Sphinx 全文搜索

Sphinx工作流程图



1. Database

数据源，是Sphinx做索引的数据来源。因为Sphinx是无关存储引擎、数据库的，所以数据源可以是MSQL、PostgreSQL、XML等数据。

2. Indexer

索引程序，从数据源中获取数据，并将数据生成全文索引。可以根据需求，定期运行Indexer达到定更新索引的需求。

3. Searchd

Searchd直接与客户端程序进行对话，并使用Indexer程序构建好的索引来快速地处理搜索查询。

4. APP

客户端程序。接收来自用户输入的搜索字符串，发送查询给Searchd程序并显示返回结果。

Sphinx工作原理

Sphinx的整个工作流程就是Indexer程序到数据库里面提取数据，对数据进行分词，然后根据生成的词生成单个或多个索引，并将它们传递给searchd程序。然后客户端可以通过API调用进行搜索。

Sphinx中文分词

中文的全文检索和英文等latin系列不一样，后者是根据空格等特殊字符来断词，而中文是根据语义来词，起搜索中应以词为依据，独立存在的单个汉字搜索几乎没有意义。目前大多数数据库尚未支持中全文检索，如Mysql。故，国内出现了一些Mysql的中文全文检索的插件，做的比较好的有hightman

中文分词。Sphinx如果需要对中文进行全文检索，也得需要一些插件来补充。其中我知道的插件有 coeseek 和 sfc 。

1. Coreseek

Coreseek是现在用的最多的sphinx中文全文检索，它提供了为Sphinx设计的中文分词包LibMMSeg并提供了多个系统的二进制发行版，其中有rpm,deb及windows下的二进制包。另外，coreseek也为phinx贡献了以下事项：

- GBK编码的数据源支持
- 采用Chih-Hao Tsai MMSEG算法的中文分词器

2. sfc

sfc（sphinx-for-chinese）是由网友happy兄提供的另外一个中文分词插件。其中文词典采用的是xdit。据其介绍,经过测试，目前版本在索引速度上(Linux 测试平台)基本上能够达到索引UTF-8英文的一，即官方宣称速度的一半。（时间主要是消耗在分词上）。 现提供了与sphinx最新版(sphinx 0.9.10同步的sphinx-for-chinese-0.9.10-dev-r2006.tar.gz 。此版本增加了sql_attr_string，经过本人的试。其安装和配置都非常方便。happy兄在分词方面还有另外一个贡献——php-mmseg，这是php中文分词的一个扩展库。

Sphinx安装步骤

1.下载源码

1.下载地址

https://github.com/zwxhenu/coreseek

2. 下载命令

```
git clone https://github.com/zwxhenu/coreseek
```

2.安装mmseg3中文分词

** 1. 安装依赖**

```
yum -y install gcc gcc-c++ autoconf python python-devel libiconv libtool
```

** 2. 编译安装mmseg-3.2.14**

```
cd mmseg-3.2.14
./configure --prefix=/usr/local/mmseg3
```

```
make
make install
```

** 3. 遇到的问题**

1). 遇到error：cannot find input file： src/Makefile.in

```
yum -y install libtool
aclocal
libtoolize --force
automake --add-missing
autoconf
autoheader
make clean
./configure --prefix=/usr/local/mmseg3
make
make install
```

2). 没有规则可以创建“all-am”需要的目标“data/uni.lib

```
删除Makefile.am中的data/uni.lib
automake
./configure --prefix=/usr/local/mmseg3
make
make install
```

3). 生产uni.lib

```
cd /usr/local/mmseg3/etc/
/usr/local/mmseg3/bin/mmseg -u unigram.txt
cp unigram.txt.uni uni.lib
```

3.安装coreseek

** 1.安装依赖软件 **

```
yum -y install expat expat-devel
```

2. 检查环境

```
sh buildconf.sh
```

** 3. 检查环境出问题**

1. 在 csft-4.1/buildconf.sh 文件&& aclocal \ 后加入:

```
 && automake --add-missing \
```

2) 把csft-4.1/configure.ac 文件中的AM_INIT_AUTOMAKE([-Wall -Werror foreign])改为:

```
AM_INIT_AUTOMAKE([-Wall foreign])
```

3. 在csft-4.1/configure.ac 文件中的AC_PROG_RANLIB后面加上:

```
m4_ifdef([AM_PROG_AR], [AM_PROG_AR])
```

4. 在 csft-4.1/src/sphinxexpr.cpp 文件中， 替换所有:

```
T val = ExprEval ( this->m_pArg, tMatch ); 为T val = this->ExprEval ( this->m_pArg, tMatch );
```

** 4. 配置编译选项 **

./configure --prefix=/usr/local/coreseek --without-unixodbc --with-mmseg --with-mmseg-incudes=/usr/local/mmseg3/include/mmseg/ --with-mmseg-libs=/usr/local/mmseg3/lib/ --with mysql

** 5. 配置编译选项问题解决**

1). 遇到MySQL include files... configure: error: missing include files.解决办法:

```
yum install mysql-community-embedded-devel.x86_64 mysql-community-devel.x86_64 mysq++-devel.x86_64
```

2). 提示libiconv无法找到，需要修改vi src/Makefile 文件，找 LIBS = 开头的行:

将LIBS = -lm -lz -lexpat -L/usr/local/lib –lpthread修改成:

LIBS = -lm -lz -lexpat -liconv -L/usr/local/lib -lpthread

** 5. 编译&安装**

make;

make install

4.命令行测试mmseg分词，coresekk搜索

** 1. 进入测试目录 **

cd testpack

** 2. 检查终端是否能显示中文 **

cat var/test/test.xml #此时应该正确显示中文

** 3. 进行分词和创建索引 **

/usr/local/mmseg3/bin/mmseg -d /usr/local/mmseg3/etc var/test/test.xml

/usr/local/coreseek/bin/indexer -c etc/csft.conf --all

** 4. 检查搜索结果 **

1. 命令:

/usr/local/coreseek/bin/search -c etc/csft.conf 网络搜索

2. 结果:

Coreseek Fulltext 3.2 [Sphinx 0.9.9-release (r2117)]

Copyright (c) 2007-2011,

Beijing Choice Software Technologies Inc (http://www.coreseek.com)

```
 using config file 'etc/csft.conf'...

index 'xml': query '网络搜索 ': returned 1 matches of 1 total in 0.003 sec

displaying matches:

1. document=1, weight=1, published=Thu Apr 1 22:20:07 2010, author_id=1

words:

1. '网络': 1 documents, 1 hits

2. '搜索': 2 documents, 5 hits
```

5.配置sphinx与mysql

** 1. threads源定义**

```
source threads

{

 type = mysql

 sql_host = localhost

 sql_user = root

 sql_pass = mysql57@fangstar

 sql_db = discuz

 sql_port = 3306 # optional, default is 3306

 sql_sock = /var/lib/mysql/mysql.sock

 sql_query_pre  = SET NAMES utf8

 #sql_query_pre = SET SESSION query_cache_type=OFF

 sql_query_pre  = CREATE TABLE IF NOT EXISTS pre_common_sphinxcounter ( indexid INTEGE
 PRIMARY KEY NOT NULL,maxid INTEGER NOT NULL)

 sql_query_pre  = REPLACE INTO pre_common_sphinxcounter SELECT 1, MAX(tid)-10 FROM p
e_forum_thread

 sql_query = SELECT t.tid AS id,t.tid,t.subject,t.digest,t.displayorder,t.authorid,t.lastpost,t.specia
 \

 FROM pre_forum_thread AS t \
```

```
 WHERE t.tid>=$start AND t.tid<=$end

 sql_query_range = SELECT (SELECT MIN(tid) FROM pre_forum_thread),maxid FROM pre_com
on_sphinxcounter WHERE indexid=1

 sql_range_step = 4096

 sql_attr_uint = tid

 sql_attr_uint = digest

 sql_attr_uint = displayorder

 sql_attr_uint = authorid

 sql_attr_uint = special

 sql_attr_timestamp = lastpost

 sql_query_info = SELECT * FROM pre_forum_thread WHERE tid=$id

}
```

** 2. threads index定义**

```
index threads

{

 source = threads

 path = /usr/local/coreseek/var/data/threads

 docinfo  = extern

 mlock  = 0

 morphology = none

 min_word_len = 1

 charset_type = zh_cn.utf-8

 charset_dictpath = /usr/local/mmseg3/etc/

 min_prefix_len  = 0

 min_infix_len = 1

 ngram_len = 0

 html_strip  = 0
```

```
}
```

** 3. threads_minute 源定义**

```
source threads_minute: threads
{
 sql_query_pre =
 sql_query_pre = SET NAMES utf8
 sql_query_pre  = SET SESSION query_cache_type=OFF
 sql_query_range = SELECT maxid+1,(SELECT MAX(tid) FROM pre_forum_thread) FROM pre_c
mmon_sphinxcounter WHERE indexid=1
}
```

** 4. threads_minute 索引定义 **

```
#threads_minute
index threads_minute : threads
{
 source = threads_minute
 path = /usr/local/coreseek/var/data/threads_minute #windows下最好用全路径
}
```

** 5. posts 源定义**

```
#posts
source posts
{
 type = mysql
 sql_host = localhost
 sql_user = root
 sql_pass = mysql57@fangstar
```

```
 sql_db = discuz

 sql_port = 3306

 sql_query_pre = SET NAMES utf8

#  sql_query_pre  = SET SESSION query_cache_type=OFF

 sql_query_pre = REPLACE INTO pre_common_sphinxcounter SELECT 2, MAX(pid)-2 FROM pre
forum_post

 sql_query = SELECT p.pid AS id,p.tid,p.subject,p.message,t.digest,t.displayorder,t.authorid,t.la
tpost,t.special \

 FROM pre_forum_post AS p LEFT JOIN pre_forum_thread AS t USING(tid) \

 WHERE p.pid>=$start AND p.pid<=$end

 sql_query_range = SELECT (SELECT MIN(pid) FROM pre_forum_post),maxid FROM pre_comm
n_sphinxcounter WHERE indexid=2

 sql_range_step = 4096

 sql_attr_uint = tid

 sql_attr_uint = digest

 sql_attr_uint = displayorder

 sql_attr_uint = authorid

 sql_attr_uint = special

 sql_attr_timestamp =lastpost

 sql_query_info = SELECT * FROM pre_forum_post WHERE pid=$id

}
```

6. posts index定义

```
#posts

index posts

{

 source = posts

 path = /usr/local/coreseek/var/data/posts #windows下最好用全路径
```

```
 docinfo = extern
 mlock = 0
 morphology  = none
 min_word_len = 1
 html_strip  = 0
 charset_dictpath = /usr/local/mmseg3/etc/ #BSD、Linux环境下设置，/符号结尾
 charset_type  = zh_cn.utf-8
 #charset_debug  =  0
 ngram_len = 0
}
```

** 7. posts_minute 源定义**

```
 #posts_minute
source posts_minute : posts
{
 sql_query_pre =
 sql_query_pre = SET NAMES utf8
 #  sql_query_pre  = SET SESSION query_cache_type=OFF
 sql_query_range = SELECT maxid+1,(SELECT MAX(pid) FROM pre_forum_post) FROM pre_co
mon_sphinxcounter WHERE indexid=2
}
```

8. posts_minute index定义

```
#posts_minute
index posts_minute : posts
{
 source = posts_minute
 path = /usr/local/coreseek/var/data/posts_minute #windows下最好用全路径
```

```
}
```

9. 全局indexer定义

```
indexer
{
 mem_limit  = 256M
}
```

10. searchd服务定义

```
searchd
{
 listen = 9312
 listen =  /tmp/sphinx.sock
 log  = /var/log/spinhx/searchd.log
 query_log  = /var/log/spinhx/query.log
 read_timeout = 5
 client_timeout = 300
 max_children = 30
 pid_file = /var/run/searchd.pid
 max_matches  = 1000
 seamless_rotate  = 1
 preopen_indexes  = 0
 unlink_old = 1
 mva_updates_pool = 1M
 max_packet_size  = 8M
 max_filters  = 256
 max_filter_values  = 4096
```

}

6. 启动服务，创建索引

1. 创建索引

```
/usr/local/coreseek/bin/indexer -c /usr/local/coreseek/etc/sphinx.conf -all
```

2.启动后台服务searchd

```
/usr/local/coreseek/bin/searchd -c /usr/local/coreseek/etc/sphinx.conf
```

3. 后台服务测试

```
/usr/local/coreseek/bin/search -c /usr/local/coreseek/etc/sphinx.conf aaa
```

4. 自动化命令

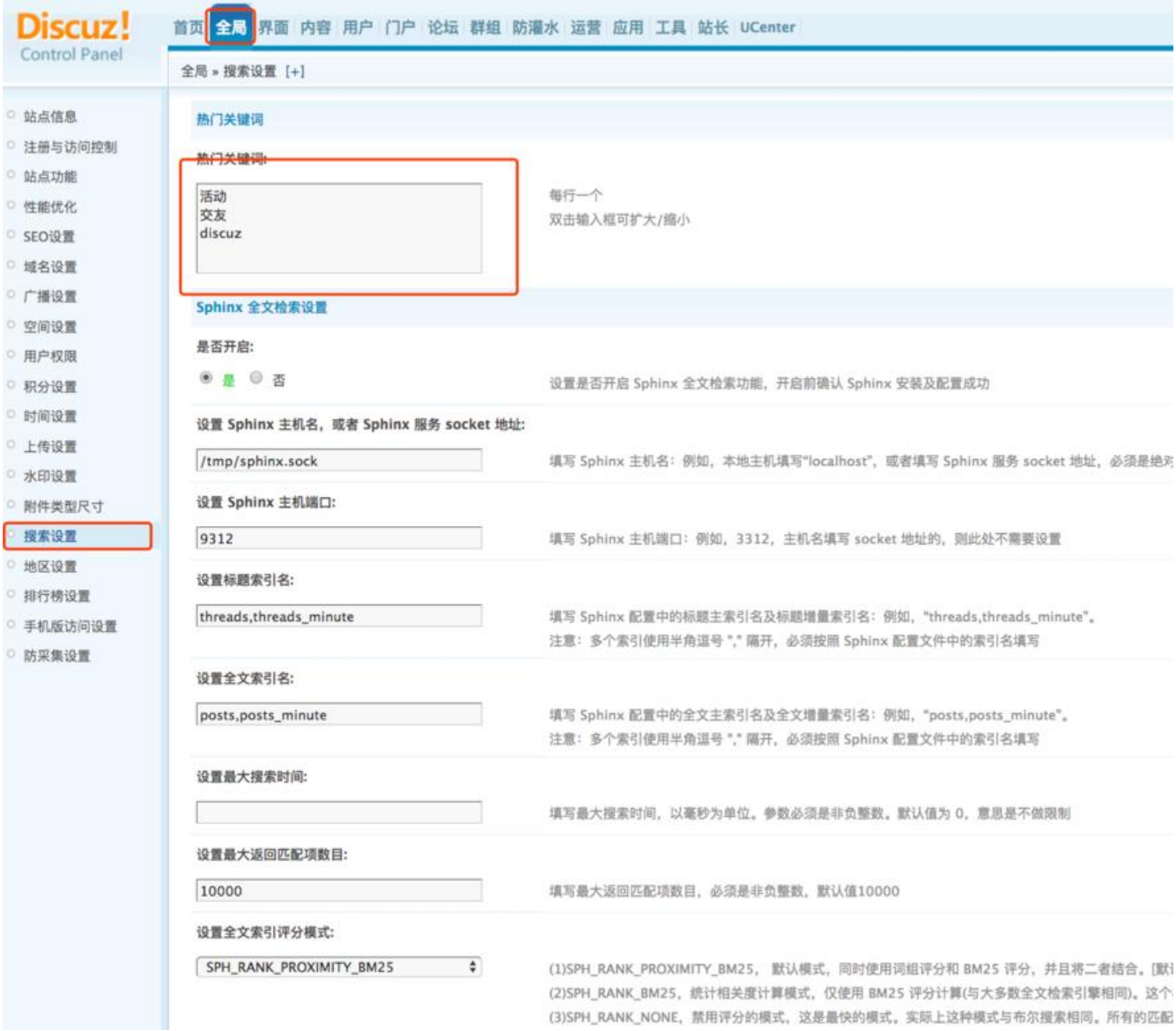
```
crontab -e
* */4 * * * /usr/local/coreseek/bin/indexer -c /usr/local/coreseek/etc/sphinx.conf -all --rotate
```

5. 关闭后台服务searchd

```
/usr/local/coreseek/bin/searchd -c /usr/local/coreseek/etc/sphinx.conf --stop
```

#discuz后台配置

1. 按照下图进行搜索设置



2. 点击提交

参考文献

1. discuz论坛配置开启Sphinx全文搜索

2. coreseek sphinx mmseg mysql 全文检索 安装 配置

3. Coreseek/Sphinx安装测试配置指南
4. Discuzx3 使用sphinx实现全文搜索功能
5. 千万级Discuz!数据全文检索方案(Sphinx)
6. centos安装coreseek

原文链接：Discuz3 开启 Sphinx 全文搜索