后盾网 人人做后盾

www.houdunwang.com

Coreseek
全文检索

后盾网 2011-2016

您在运营知识分享型的社区(Forum)

- 大量宝贵的答案、精辟的见解被淹没在回复中，通过帖子主题检索不到；
- 为了降低服务器负载，您不得不限制用户每30秒才能使用搜索功能一次，用户很难受，您很难过；

您在运营社会化社交网络（SNS）

- 如何让您的用户找到志同道合的朋友?
- 如何帮助用户更好的管理自己的Blog?

您在运营电子商务网站(e-Shop)

- 如何让每个访客尽快找到他/她感兴趣的商品，达成销售?

您每天需要处理大量的电子文档(CMS)

- 如何在最短的时间内找到您需要的文件
- 如何利用现有文件的内容加速您新文档的撰写工作?

何时，您需要全文检索

- Sphinx是由俄罗斯人Andrew Aksyonoff开发的一个全文检索引擎。意图为其他应用提供高速、低空间占用、高结果相关度的全文搜索功能。
- 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，它可以提供比数据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设计搜索API接口，如PHP,Python,Perl,Ruby等。
- 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）。Sphinx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分钟内完成，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。

Sphinx是什么

coreseek介绍

- Coreseek 是一款中文全文检索/搜索软件，以GPLv2许可协议开源发布，基于Sphinx研发并独立发布，专攻中文搜索和信息处理领域，适用于行业/垂直搜索、论坛/站内搜索、数据库搜索、文档/文献检索、信息检索、数据挖掘等应用场景。

下载地址

- http://www.coreseek.cn/news/14/52/

Coreseek中文全文检索

csft(Sphinx)

- Sphinx 是一个在 GPLv2 下发布的一个全文检索引擎，Sphinx 是一个独立的搜索引擎,意图为其他应用提供高速、低空间占用、高结果 相关度的全文搜索功能。Sphinx 可以非常容易的与 SQL 数据库和脚本语言集成。

libMMSeg

- LibMMSeg 是Coreseek.com为Sphinx 全文搜索引擎设计的中文分词软件包，其在GPL协议下发行的中文分词法。

Coreseek包含组件

1. 把老师提供的Sphinx文件夹上传到/root目录

2. 安装相关软件：

   yum –y install make gcc g++ gcc-c++ libtool autoconf automake imake mysql-devel libxml2-devel libtool.i686 expat-devel php-devel

3. tar zxvf /root/sphinx/coreseek-4.1-beta.tar.gz

4. cd /root/coreseek-4.1-beta/mmseg-3.2.14/

5. ./bootstrap #测试安装环境

6. ./configure --prefix=/usr/local/mmseg3 #指定安装目录

7. make & make install

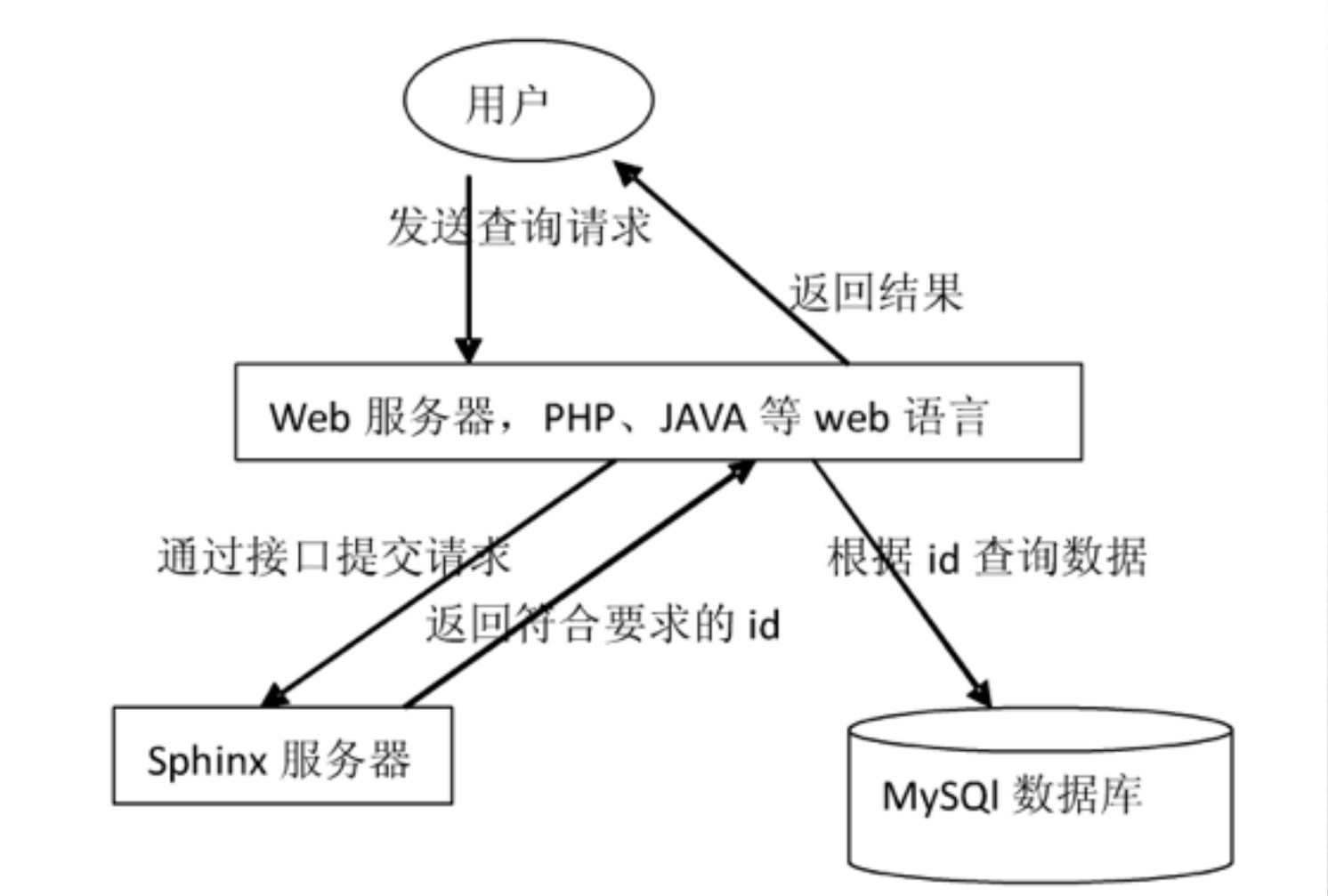
安装mmseg

/usr/local/mmseg3/bin/mmseg -d /usr/local/mmseg3/etc /roots/coreseek-4.1-beta/mmseg-3.2.14/src/t1.txt

中文分词测试

1. cd /root/coreseek-4.1-beta/csft-4.1
2. sh buildconf.sh
3. ./configure --prefix=/usr/local/coreseek --without-unixodbc --with-mmseg --with-mmseg-includes=/usr/local/mmseg3/include/mmseg/ --with-mmseg-libs=/usr/local/mmseg3/lib/ --with-mysql
4. make & make install

安装csft



PHP操作Sphinx

索引生成器(indexer)

- 查询数据库，为结果的每行中的指定字段建立索引，并且将每个索引条目绑定到行的主键上。

搜索引擎(searchd)守护程序

- 搜索引擎是名为 searchd 的守护程序。该守护程序将接收搜索词和其他参数，快速遍历一个或多个索引，并返回结果。如果找到匹配，searchd 将返回一个主键数组。Searchd 默认将在端口 3312 上通过套接字连接与应用程序进行通信。

命令行search实用程序(search)

- search 实用程序使您可以从命令行构造搜索而无需编写代码。如果 searchd 返回匹配，则 search 将查询数据库并显示匹配集中的行。search 实用程序对于调试 Sphinx 配置和执行临时搜索十分有用。

分词命令

1. 使用提供的建表sql.txt文件创建数据表
2. 复制配置文件：cp /root/sphinx/csft.conf /usr/local/coreseek/etc/
3. 创建索引文件：/usr/local/coreseek/bin/indexer --all
4. 分词搜索：/usr/local/coreseek/bin/search 后盾

Mysql数据分词实验

要连接的 SQL 服务器主机地址

- sql_host = localhost

SQL 服务器的 IP 端口(mysql 端口 3306, pgsql 端口 5432)

- sql_port = 3306

sql_host 时使用的 SQL 用户名

- sql_user = test

SQL 用户密码

- sql_pass =

使用的 SQL 数据库

- sql_db = test

Coreseek配置说明

主查询之前执行的预先查询

- sql_query_pre = SET NAMES=utf8
- sql_query_pre = SET SESSION query_cache_type=OFF

获取文档的主查询（数据源中只能有一个主查询）

- 文档 ID 必须是第一列,而且必须是唯一的正整数值，所有既不是文档ID(第一列)也不是属性的列的数据会被用于建立全文索引，可以指定32个列。
- sql_query = SELECT id, group_id, title, content FROM documents

用来获取和显示记录详细信息, 仅用于调试目的

- sql_query_info= SELECT * FROM documents WHERE id=$id

Coreseek配置说明

charset_dictpath=/usr/local/mmseg3/etc/

- 必须设置，表示词典文件的目录，该目录下必须有uni.lib词典文件

charset_type=zh_cn.utf-8

- 必须设置，表示启用中文分词功能；否则中文分词功能无效，使用sphinx的其他处理模式。启用中文分词功能后，需要source数据源之中，读取的数据编码字符集为UTF-8，否则无法正确处理

索引文件的路径和文件名

- path = /var/data/test1

索引过程内存使用限制

- 最大可能的限制是 2047M。太低的值会影响索引速度,但256M 到 1024M 对绝大多数数据集(如果不是全部)来说足够了。
- mem_limit = 256M

Coreseek配置说明

安装libsphinxclient否则无法安装php的sphinx扩展

1. cd /roott/coreseek-4.1-beta/testpack/api/libsphinxclient
2. ./configure
3. make & make install

php的sphinx扩展安装

4. cd /root/sphinx/
5. tar zxvf php-module-sphinx-1.3.2.tgz
6. cd sphinx-1.3.2/
7. phpize (如果不存在此命令，执行yum –y install php-devel安装)
8. ./configure --with-php-config=/usr/bin/php-config --with-sphinx
9. make & make install

修改PHP配置文件

1. cd /etc/php.d/
2. cp gd.ini sphinx.ini
3. 修改内容为extension=sphinx.so

安装php扩展

启动Sphinx

1. /usr/local/coreseek/bin/searchd -c /usr/local/coreseek/etc/csft.conf
2. netstat -tlp

   #t:tcp l:Listen (监听)服务 p程序名

停止搜索服务

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

已启动服务，要更新索引

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

注：默认配置文件为csft.conf，如果文件没改名时，不用设置-c选项

启动Sphinx

1. 将php文件夹放入web访问目录
2. 运行index.php脚本进行测试

编写php代码

SphinxClient::query (string $query [, string $index = "*" [, string $comment = ""]])

参数说明：

- query　　查询字符串.
- index　　索引名称 (多个索引用逗号分割，或者为“*”表示全部索引)

SphinxClient::setMatchMode (int $mode)

$mode说明：

- SPH_MATCH_ALL　　匹配所有查询词（默认模式）
- SPH_MATCH_ANY　　匹配查询词中的任意一个
- SPH_MATCH_PHRASE　　将整个查询看作一个词组，要求按顺序完整匹配

PHP操作Sphinx函数

SphinxClient::buildExcerpts (array $docs , string $index , string $words [, array $opts])

参数说明：

- $docs　　要高亮的内容（数组类型，查找到的记录结果）
- $index　　索引名称
- $words　　高亮字符串（主是用户搜索的字符串）
- $opts　　选项

$opts选项说明：

- before_match　　在匹配的词前插入内容
- after_match　　在匹配的词后插入内容

PHP处理查询结果

- 有这么一种常见的情况:整个数据集非常大,以至于难于经常性的重建索引,但是每次新增的记录却相当地少。一个典型的例子是:一个论坛有 1000K 个已经归档的帖子,但每天只有1000 个新帖子。
- 在这种情况下可以用所谓的“主索引+增量索引”(main+delta)模式来实现“近实时”的索引更新。
- 这种方法的基本思路是设置两个数据源和两个索引,对很少更新或根本不更新的数据建立主索引,而对新增文档建立增量索引。在上述例子中,那 1000000 个已经归档的帖子放在主索 引中,而每天新增的 1000 个帖子则放在增量索引中。增量索引更新的频率可以非常快,文档可以在出现几分种内就可以被检索到。

实时索引更新

重新建立索引

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

重建增量索引

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

注:

主数据源与增量数据源字段数量要一致

实时索引更新

创建日志文件

- touch /usr/local/coreseek/var/log/main.log
- touch /usr/local/coreseek/var/log/delta.log

创建shell脚本

- mkdir /usr/local/coreseek/sh/
- cd /usr/local/coreseek/sh
- touch main.sh
- touch delta.sh
- chmod a+x –R /usr/local/coreseek/sh/

创建日志与脚本文件

mainx.sh：主索引脚本

```
#!/bin/sh
    /usr/local/coreseek/bin/indexer main --rotate >> /usr/local/coreseek/var/log/
    main.log
```

delta.sh#增量索引

```
#!/bin/sh
/usr/local/coreseek/bin/indexer delta --rotate >> /usr/local/coreseek/var/log/
    delta.log
```

实时索引更新

增量索引每5分钟更新，主索引凌晨3:30点更新

crontab -e 来编辑 crontab文件

1. */5 * * * * /bin/sh /usr/local/coreseek/sh/delta.sh
2. 30 3 * * * /bin/sh /usr/local/coreseek/sh/main.sh

实时索引更新（计划任务）