Setting up a search engine for your website

Posted by Steve on Mon 29 Oct 2007 at 06:59

If you run a website and one people to be able to search it then installing a local spider to crawl your site and create a small database of your content which users may search is a relatively straightforward thing to do. Here we'll look at using mnoGoSearch - which is packaged for Debian and simple to install.

The mnogosearch packages come in three different flavors:

mnogosearch-mysql

> A package which will index your content and store it in a MySQL database.

mnogosearch-pgsql

> A package which will index your content and store it in a PostgreSQL database.

mnogosearch-sqlite

> A package which will index your content and store it in a local SQLite database.

Rather than getting bogged down with choosing one or the other right now I'll simply suggest you start with the `-sqlite` version. This is fast enough for the kind of sites that I've used it upon, and if it doesn't scale switching would be a very very simple process.

To get started you'll need to install the package:

```
etch32-builder:~# apt-get install mnogosearch-sqlite
Building dependency tree... Done
The following extra packages will be installed:
  libsqlite0 mnogosearch-common
The following NEW packages will be installed:
  libsqlite0 mnogosearch-common mnogosearch-sqlite
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 3196kB of archives.
After unpacking 9077kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Once installed you'll be prompted to answer some simple questions via `debconf`, answer them as you see fit - making sure you choose `sqlite` as the database site, and answering "`yes`" when asked if you want the database to be created.

After this you'll find that you have a new directory `/etc/mnogosearch/` containing configuration files which will allow you to configure the behavior of the software. Install the mnogosearch-doc package if you'd like to have details on all the options - but the basics we'll cover here.

The package contains two main components:

- The indexer, or spider, which will follow links from your starting point(s) to record page text in the database.
- A simple CGI script which will allow users to find content upon your site using that database.

To get started we'll need to configure the indexer part of the package so that it knows what to index, and configure the list of things to ignore, etc.

The indexer is configured via the file `indexer.conf` which is pretty well commented. The only option which is mandatory is the starting-point for the indexing. You can configure this by adding the following to the end of the file:

```
##########################################################################
#URL http://localhost/path/to/page.html
# This command inserts given URL into database. This is useful to add
# several entry points to one server. Has no effect if an URL is already
# in the database. When inserting indexer does not executes any checks
# and this URL may be deleted at first indexing attempt if URL has no
# correspondent Server command or is disallowed by rules given in
# Allow/Disallow commands.
#
URL http://example.com/
```

Once you've added your hostname to the file you can start the indexer process running by invoking:

```
etch32-builder:~# indexer
indexer[2040]: indexer from mnogosearch-3.2.37-sqlite started with '/etc/mnogosearch/indexer.con
indexer[2040]: [2040]{01} Writing words (0 words, 32 bytes, final).
indexer[2040]: [2040]{01} The words are written successfully. (final)
indexer[2040]: [2040]{01} Done (0 seconds, 0 documents, 0 bytes,  0.00 Kbytes/sec.)
```

In your case this will display a *lot* of output by default; a dump of all the URLs it has managed to find, starting from the `http://example.com/` page we specified earlier.

To keep the site index up to date you'll want to ensure that the indexer is executed on a regular basis. I suggest running this daily, unless you have a site which doesn't change too much. To do that you could create a small shell-script in `/etc/cron.daily/`. This script, `mnogo`, is what I use:

```
#!/bin/sh
/usr/sbin/indexer -a -l
```

Here `-a` causes the indexer to re-run even if documents haven't expired. The `-l` flag prevents the progress being written to stdout. The expiry of documents is something you may configure in the `indexer.conf`, as explained in the comments. But in brief if you have a website which doesn't change more than weekly you can cause indexing to terminate if the database has already been updated within the past week.

The only other thing that might need modification are the paths, files, and types of things to exclude from the indexing. There are many ways of doing this, each explained and demonstrated within the `indexer.conf` file. As a simple example here we disallow spidering of URLs containing `/tag/` in their path, and RSS files:

```
Disallow */tag/*
Disallow *.rss
Disallow *.xml
```

Once you have an index you're happy with the next thing is to ensure that your users may use it.

You should merely cause the CGI-script `/usr/lib/cg-bin/search.cgi` to be available for your virtual host - with something like this in your Apache configuration file:

```
  AddHandler cgi-script .cgi

  # CGI Handling
  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

  <Location /cgi-bin>
     Options +ExecCGI
  </Location>
```

(Don't forget to enable the handling of CGI scripts with `a2enmod cgi`" if appropriate!)

With this in place you should find that you may view http://example.com/cgi-bin/search.cgi to search.

The output page(s) which are returned to your users will be constructed via the template file `/etc/mnogosearch/search.htm` - so you may edit that to add your logo, etc.

For a sample of what this kind of search looks like please see this search page.

Database Recreation

> The indexer has many options which you can find in the manpage, readable via "`man indexer`". Two useful commands are the following:
>
> ```
> indexer -Edrop
> indexer -Ecreate
> ```
>
> These commands drop and re-create the default database tables. These can be invoked in succession to delete your database contents if you wish to start over.

This article can be found online at the **Debian Administration** website at the following bookmarkable URL:

- http://www.debian-administration.org/articles/557