New SUMO Search Engine – High Level Summary

Features

- Uses Sphinx as the search engine http://www.sphinxsearch.com/
- Batch job indexing process (not real time indexing) for increased real-time search performance
- Caching of results in PHP session built in and the same caching hooks can be used to implement memcache very easily (adding some memcache hooks and flick a switch)
- Merging of forum and kb search results
- "Did you mean" spelling correction (need to be configured into template)
- Automatic "description" generation from first 200 characters of content (if manual description is not included)
- Configurable weights:
  - articles vs. forum posts
  - content vs. description vs. page name vs. tags
  - poll results (not yet support new CSAT polls)
  - answered vs. not answered status for forums (integration not yet complete)
- Ability to add/remove words that should not be indexed
- Can search multiple languages of term and then merge results showing pages in users' locale first (through automatic use of Google translation, for languages supported by Google translation) (need to check terms of use of Google translation...)
- Search filters supported:
  - articles vs. forums only
  - language
  - last modification
  - help vs. troubleshooting article
  - answered vs. not answered thread (integration not yet complete)
  - author/contributor

Test installation environment and results

- Currently tested on Nelson's development server (a Xen virtual machine running CentOS 5, MySQL 5.0.45, and PHP 5.1.6) which is very close to Mozilla's machines so no platform incompatibility problems are expected.
- We used Sphinx version 0.9.8 RC2. The current latest version of Sphinx is 0.9.8 which should essentially work the same.

Outstanding Issues

- Some characters in the following languages (Korean, Finnish, Polish, Russian, Slovak, Ukrainian, Japanese) causes the indexing process to crash. A workaround (which we have tested) is to write a logging system that finds out which characters these are whenever it crashes and to exclude them from indexing using a filter. This may take a few days work. We might also find out the underlying reason for this problem (but deeper investigation into this problem is not yet done).

- Scalability and performance testing is not done yet, but we know that search performance will be better than current Forum search (which uses default TikiWiki search) simply because of architectural improvements, but we are not sure how it compares to the KB search using Google. We know that using Memcache to cache results should be a viable way to scale this (see below).

- Need to test this with Memcache as we do not support $_SESSION variables for non-logged in users for scalability reasons (setting this up should be easy, involving adding a few memcache hooks and flick a switch, on the Mozilla test machines). But of course, testing will take a few days.

- The indexing process cannot be incremental, but has to be complete each time. Attempts to do incremental update of the indexes crashes Sphinx. The full indexing using the test database took about 4 minutes on the test. Given the growth in size of content, this might have increased, but I think to no more than 15 minutes. Proposed solution is to remove old forum posts from indexing to limit the time taken to index.

- The following items should take no more than a few days to do:

  - Need to implement more generic filter by categories. It is supported as right now it is based on categoryId for help vs troubleshooting article, but needs to be generalized a better.

  - Need to re-theme the search results page (and to add any stylistic markers necessary, e.g. to indicate if it is a forum result or a kb page).

  - "Did you mean" spelling correction needs to be configured and set into template

  - "More like this" functionality (if still required) needs to be integrated

  - Use of poll results as weights need to be modified to support new CSAT polls

  - Use of answered vs. not answered status of forums as weight need to be integrated

  - Need to check if use of Google Translation to search in multiple languages is a problem (legally or technical dependency reasons).

Installation procedure

- Download Sphinx (version 0.9.8) from http://www.sphinxsearch.com/ , and compile from source on target machine. Follow Sphinx installation procedure to setup Sphinx, including database schema.

- Follow the 6 step process in the documentation in the package to copy the additional files to the correct location, to configure configuration files, execute some SQL commands to modify indexing database schema.

- Configure weights and stopwords in admin-search.php

Package and Documentation

- This can be obtained from http://vps11.etazo.net/package%20GSOC.zip, and detailed documentation is included in the package

- The package does not include Sphinx source code, which has to be obtained from http://www.sphinxsearch.com/