

Manticore Search

Manticore Search is an Open Source database that was created in 2017 as a continuation of Sphinx Search engine (Started in 2001). It's a modern, fast, light-weight and full-featured database with outstanding full-text search capabilities. <https://manticoresearch.com/>
You can check out benchmarks at: <https://db-benchmarks.com/>

It is a [Unified Index](#) option starting in [Tiki25](#) (experimental) and will be production ready in [Tiki26](#).
[Manticore Search is very appreciated by its community.](#)

As of 2022-11-16, the results between the various engines are not identical in all cases. We will write a script to compare the results of MariaDB/MySQL vs Elasticsearch vs Manticore Search, and review all the discrepancies. Also, different engines have different features. If Manticore doesn't work for you, you can revert to your previous engine.

Installation

See instructions on how to install it server side:

- <https://manticoresearch.com/install/>
- <https://wikisuite.org/Manticore-Search>
- [How to Install Tiki on Ubuntu GNU Linux](#)

How to indicate to Tiki to use Manticore Search

Once Manticore is installed and running:

1. Pick Manticore as **Unified search engine** at `tiki-admin.php?page=search`
2. Rebuild index
 - Small sites, via GUI
 - Medium and large sites, via CLI `php console.php index:rebuild --log`
3. Enjoy!

Please see development information: [Manticore Search](#) and <https://wikisuite.org/Manticore-Search>

It has many features including:

- [Faceted search](#)
- [Stored Search](#)
- [Federated Search](#)

See also

- [Manticore Buddy](#)

Dev notes from Victor

In short:

1. Introducing a new setting in manticore search settings - you can enumerate the fields you want to be always indexed - e.g. important fields or fields that search too slowly with regex. Default here is title and contents - you will always want contents be in the full-text index for the whole site search.
2. Keep the search setting of the tracker fields as it works until now - work in combination with the preference to exclude non-searchable fields from the index.
3. When building the index, we try to fit all text fields as full-text but if this is more than 255, we prefer

the preferenced ones and then start with the rest until we fill the limit. If we go over the limit, the rest are indexed as string columns and allow regex search.

4. Changed multivalue attributes to MVA native manticore implementation to spare some fulltext fields.

So, most of the sites shouldn't notice this change. Bigger sites will probably need some configuration to allow faster lookups on big trackers - ensure the searched fields end up in the preference.

Move dev notes from Victor

https://gitlab.com/tikiwiki/tiki/-/merge_requests/2735

Had to unify the way we gather and store stats as well as building it for reporting. Now each reporting place uses a single stats builder, so they should be in sync (those are: rebuild console command output, rebuild log file output, last stats displayed via module).

Due to the way we are indexing wiki pages and other wiki content, the content is parsed but there is no easy way to gather the fields used in that specific place, so I added a catch-all type of a rule: manticore query builder logs all field usages in queries (either fulltext or not) and stores these. Then, after rebuild finishes, we display the following additional stats:

indexed fulltext fields

text fields converted to strings (those are supposed to be indexed but the 256 was limit hit)

indexed fields that are not used in fulltext queries (note that this might not be a complete picture as we only take the wiki texts; you can possibly do other search queries even with API or customsearch templates residing in themes that are not taken in to account)

non-indexed fields that are used in queries

If you want a more sophisticated report of where each indexed/non-indexed field is used, then I think a console command is needed with proper table display or might be part of the admin UI?

Mac OS Notes

The only way I could get this to run locally on macOS (on Apple Silicon) was by using Docker, and i needed to add these config items, pasted here for reference)

Environment

PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

GOSU_VERSION

1.11

MCL_URL

https://repo.manticoresearch.com/repository/manticoresearch_focal/dists/focal/main/binary-amd64/manticore-columnar-lib_1.15.4-220522-2fef34e_amd64.deb

DAEMON_URL

https://repo.manticoresearch.com/repository/manticoresearch_focal/dists/manticore_5.0.2-220530-348514c86_amd64.tgz

LANG

C.UTF-8

LC_ALL

C.UTF-8

Mounts

/path/to/your/tiki/temp

██

/var/lib/manticore/

██

Ports

9306/tcp

0.0.0.0:██████

9308/tcp

0.0.0.0:██████

9312/tcp

0.0.0.0:██████

Manticore Federation

We had a federation of two servers (each with Manticore) which were handling multiple sites. A third server was created and one of the sites was moved there.

Moving a site from a Manticore federation to a new server has several side effects which all need to be fixed. What I did for dev.tiki.org and the new dogfood6.evoludata.com server: 1. Mirror manticore config from dogfood5 server. Most importantly, make it listen on the external inet interface - namely, dev.tiki.org, so it is accessible from the outside. 2. Open ports 9306 and 9312 to all other hosts from the federation. dogfood servers use nftables, so I did this in /etc/nftables.conf and restarted the service. 3. Change search config in the new server to use the same hostname as manticore config - dev.tiki.org - this was already done. 4. Go to Admin->Tools->external wikis and update the federation index locations. We were using local index locations for doc and themes and I had to change this to remote for dev.tiki.org. 5. Go to external wikis of ALL other sites participating in the federation and update dev.tiki.org index to the new correct remote location. 6. Open any ports on the remote hosts to the new host. I sent an email to Oliver to do this for tiki.org server. Also opened the ports for dogfood5.evoludata.com server in /etc/nftables.conf After all this is done, all federated indices need to be rebuilt to re-configure each other.