

Manticore Search

The community reaction in public and private is very enthusiastic. So proposal is accepted and work started in January 2022. We [hit our first bug but it was already fixed](#) for the next version 😊

Latest code: <https://gitlab.com/kroky/tiki/-/commits/feature/manticore/> Will be merged in [Tiki25](#)

Wow! Manticore 5.0 will have so many new features!

<https://github.com/manticoresoftware/manticoresearch/blob/master/manual/Changelog.md>

What

This is a proposal to add [Manticore Search](#) support to Tiki. [Manticore is a fork of Sphinx Search](#)

Why

As of 2021-11-28, we have 2 good options for the Unified Index:

- MariaDB/MySQL Full Text Search is the default and built-in.
- Elasticsearch for advanced projects

Details are here: [Unified Index Comparison](#).

While MariaDB/MySQL Full Text Search is great for small projects, and is impressively fast, we hit limits on more advanced project. For example, [Faceted search](#) is not supported. We *could* add support, but the feature requests will just keep coming. ([Stored Search](#), [“Did you mean?”](#), etc.)

Elasticsearch is great for big projects. But it requires beefy servers, and can be complex to manage. And [Elasticsearch is no longer available under an OSI approved license.](#), so a fork has emerged: [OpenSearch](#). We can speculate that they will try to stay close, to make it easy to support both (like MariaDB vs MySQL). However, sooner or later, we can also expect them to diverge.

This is a good time to re-explore our abstraction layer, and add support for Manticore.

Benefits of Manticore

- Tons of features
- [Consulting services available](#)
- Open community ([forum](#), [courses](#), [chat](#), [issues](#), etc.)
- Will be less work and more future-proof to add support for Manticore than to add missing features to our MariaDB/MySQL implementation.
- Easy to install on all the main platforms: <https://manticoresearch.com/downloads/>
- [Can be installed without root access](#) so shared hosting should be OK (They will need SSH though).
- Uses significantly fewer resources (RAM) than Elasticsearch/OpenSearch
- Official support for PHP: <https://packagist.org/packages/manticoresoftware/manticoresearch-php>

- <https://manticoresearch.com/2019/10/22/manticore-search-indexes-and-document-storage/>
- <https://manticoresearch.com/2018/02/16/fuzzy-matching-and-2nd-pass-query/>
- Generally a great project as per these [criteria](#).

Planned enhancements

- Secondary indexes → higher performance.
- Docstore for columnar attributes → higher performance.
- Read-only listeners → better security.
- Bulk insert/replace via HTTP JSON → higher performance.
- Keepalive support in HTTP for multi-queries → ease of use.
- Further columnar storage performance optimizations.
- Making full-text optional. Manticore is not only about full-text, but still requires at least one full-text field in each index. It's time to change it.

Drawbacks

- This is a major project. It will take away resources from other projects.

Risks

- A risk of having multiple engines is to have diverging results (which has been the case Since Tiki12). Thus, we created https://gitlab.com/tikiwiki/tiki/-/merge_requests/940

Questions

- Is it possible to develop this, without any risk for current setup? This being said, we don't want to maintain too many things in parallel in the long term.
- Do we need extra conversion or publishing a guide for parameters like:
 - ft_min_word_len (or innodb_ft_min_token_size)
 - ft_stopword_file (or INNODB_FT_DEFAULT_STOPWORD)
 - ...

About revamping Unified Index

- In the course of this project
 - Make it clear which engine supports what

Some notes from Victor (who lead the analysis)

“

Did quite a review of their docs, sources and manual/courses. Forked from Sphinx and based on C++ sounds like a win on the performance part.

SQL search and indexing language is a huge win over elastic. Integration will be much easier. They even have SphinxSE kind of extensions to store the index inside existing mysql db but that's more of an edge case to explore.

I see they support sub-queries which kind of work like sql joins - e.g. ability to get all tasks with a timesheet not in category XYZ - this is a classic example of a join we currently don't support but Manticore seems to support which is very nice.

In terms of existing ES functionality, we seem mostly covered. Full text, range queries, boolean operators, geo spatial search, faceted search, ranking, sorting, filtering, everything seems there. There is also NLP module for lemming, stemming and other pre-processing which might allow newer usages. Our current mysql index stores everything in one big table and has a huge performance overhead because of this. ES stores also all mappings in the index at once. Manticore, on the contrary, allows us to define a schema - different "tables" for different types of documents which might open even more use-cases to support. We can potentially store per-tracker index tables which makes searching even faster, support sub-queries and joins to retrieve data from different document types more easily, etc. I'd be excited to integrate that into Tiki if we have the chance to do that...

“

1. We can use their PHP lib but they also support SQL query interface which we already support with our Mysql search index, so it will be a little effort to bypass their PHP lib.

2. Federated search could be implemented relatively easily with Manticore search. It uses a mysql-like table per index and we can just search in multiple tables to combine results. Furthermore, we can even search in different Manticore servers and combine results. I think we are good to go here.

3. I don't think we need to have separate Manticore processes running on the same machine. We can do logical segmentation and use index prefixes for different clients. We can also do the other way around - https://manual.manticoresearch.com/Creating_a_cluster/Creating_a_cluster - use distributed index and load balancing to improve performance on high-traffic sites.