

# **Evaluation of the energy-saving effect of a new database software**

A report within the project P2023-00192, "Evaluation of energy-saving effect of new database software", financed by the Swedish Energy Agency.

## Evaluation of energy saving effect

**Reference:** P2023-00192

**Done on request by Vesiro within the project P2023-00192, financed by the Swedish Energy Agency.**

### Contact persons RISE

Joachim Johansson, Daniel Olsson, Tina Stark  
ICE Data center  
+46730829340  
Tina.stark@ri.se

### RISE Research Institutes of Sweden AB

Postal address  
Box 857  
501 15 BORÅS  
SWEDEN

Office location  
Björkskataleden 112  
SE-973 47 Luleå

Phone / Fax / E-mail  
+46 10-516 50 00  
+46 33-13 55 02  
info@ri.se

Confidentiality level  
C2 - Internal

This document may not be reproduced other than in full,  
except with the prior written approval of RISE AB.

## Table of contents

<b>Assignment</b>	<b>3</b>
<b>Object and material</b>	<b>3</b>
Test environment	3
Software	4
Test data	4
<b>Test setup</b>	<b>5</b>
Part 1	5
Part 2	5
<b>Test results</b>	<b>6</b>
Part 1	6
Part 2	6
<b>Summary</b>	<b>7</b>
Conclusion	8

## Assignment

The ongoing digitization of society has led to that companies worldwide are storing more and more data. Requiring an increased need for servers that can store the data and be used to analyse and create value from the data collected.

Servers are often kept in large data centers and projections show that the world's data centers are expected to go from representing 1-2% of global energy consumption in 2020 to 10% in 2030. Vesiro's algorithms are based on 8 years of R&D and enable faster searches in databases, leading to ensure that the server hardware where the databases are stored is used more efficiently.

The goal of this assignment has been for us at RISE to test Vesiro's software and to validate and quantify the performance and energy saving effect of Vesiro's algorithm.

## Object and material

This section describes the server setup, software, and test data.

### Test environment

The test environment at ICE data center at RISE can be seen in Figure 1 below. It contains a test client, a VM with 4 core 4GB RAM and 500GB disk and three ASUS servers with the specification described in the list below.

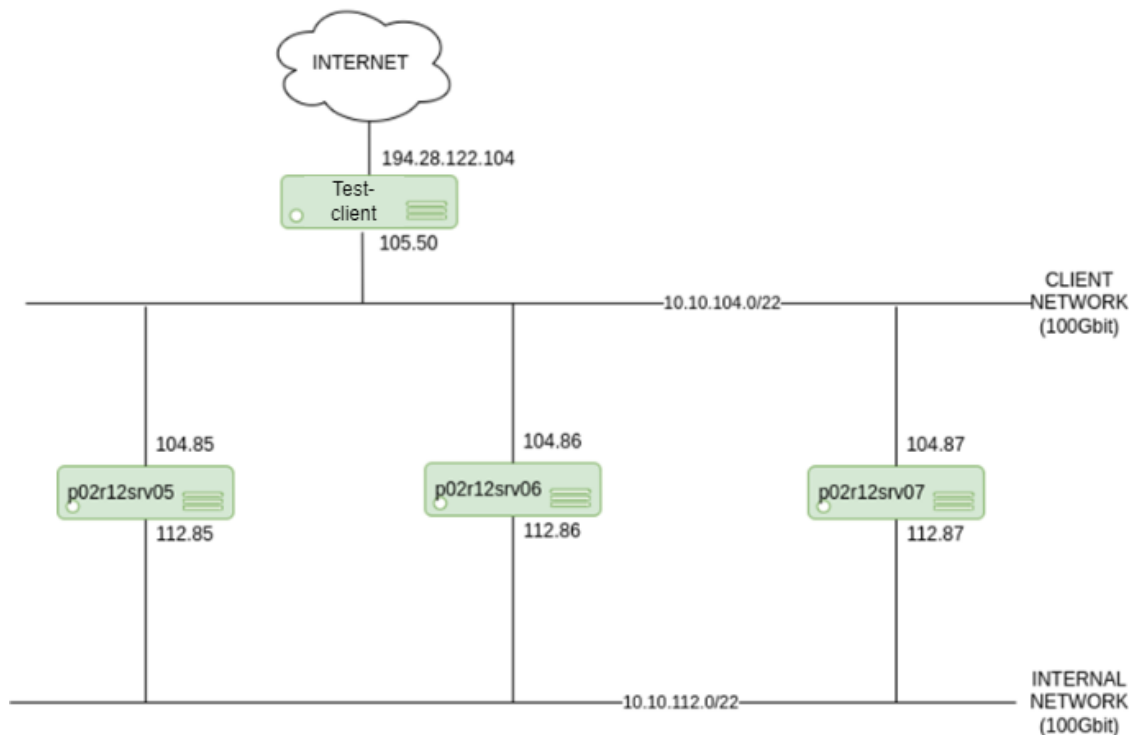


Figure 1. Test environment provided by ICE Data center at RISE.

Specification for the server setup:

- Asus RS520A-E11-RS24U
- CPU: AMD EPYC 7313 16-Core Processor (32 Thread)
- Memory: 512 GB
- System disk: 2x256 GB NVMe
- Storage disk: 13x7.68GB PCI NVMe (total 100TB)
- Network: 2x100GbE

Due to that the Vesiro plugin is in a prototype stage, Vesiro decided to use the setup describe in Figure 2 for the tests. The Elasticsearch data base was installed on one single disk on one server, p02r12srv07. For part 2 of the testing, the test client was run on a separate server, p02r12srv06.



Figure 2. Hardware setup for the tests.

## Software

Elasticsearch is a primarily text-based search engine that is built upon Apache Lucene. It uses what is known as an “inverted index” which makes it possible to search for large bodies of text and get the top scoring documents that includes the search term. Vesiro’s software is installed as an SO, which Lucene accesses when searching. This access has been made possible through a small modification of Lucene. Accessing Vesiro’s software enables shorter search times, which means that Elasticsearch will also benefit from this.

For the moment, Vesiro's plugin can handle two search types:

- Term Search (Search for documents that include the term)
- Boolean Search (with “SHOULD”-clauses)

In the tests Term search is using Elasticsearch’s “match” query and the one called Boolean search is using 5 “match” queries with “SHOULD” (i.e., “should appear”) on each. To clarify, all Boolean search types has not been tested.

For the moment, the plugin is compatible with Lucene version 9.4.2 and the plugin is exclusively used to speed up search times, by utilizing Vesiro’s search algorithms each time a query is processed. The way the original data is stored (the index) remains the same as in the original software, but with the addition of extra data files that are specific to Vesiro’s plugin.

The plugin is currently in its prototype phase but under development. As a result, the primary focus is on its core functions, leading to some limitations in its capabilities. A key limitation is the restricted variety of search types it supports. Presently, the plugin doesn’t support continuous updates to the index. Instead, it can only search data available upon starting Elasticsearch. The test is therefore prepared with the addition of data files.

## Test data

The data that is used during testing, is an Elasticsearch index with an approximate size of 500GB. The data is collected from Common Crawl, which is an open repository of web crawl

data, famously used by many of the LLM's like ChatGPT. The terms used in the tests comprise of 100,000 words and symbols extracted from the entire Wikipedia database, provided by Vesiro.

## Test setup

The scope of the test is to evaluate and compare the Elasticsearch with and without Vesiro's plugin, with the focus on search time (server-side processing time). The test is divided into two parts, the first part where Lucene is used and the second part where a complete instance of Elasticsearch is used. For each part tests for both Term search and Boolean search were performed.

### Part 1

The test was configured with these parameters:

- 2 000 000 queries for Boolean and 10 000 000 queries for Term
- Number of threads: 32
- Mode: "lucene" (for standard Lucene) or "vesiro" (to use Vesiro's software)
- Search type: "TERM" for term searches or "BOOL" for Boolean searches

The test was then initiated, and four tests were performed:

- Term Search with standard Lucene
- Boolean Search with standard Lucene
- Term Search with Vesiro's plugin
- Boolean Search with Vesiro's plugin

Each test will generate files which includes data of:

- Verification Data – data that validates the correctness of Vesiro's plugin.
- Average queries/s
- Average search time – time for the "query phase" in Lucene, including searching and scoring

### Part 2

The test was configured with these parameters:

- Terms file – terms that are used to construct queries
- 2 000 000 queries for Boolean and 10 000 000 queries for Term
- Search type: "TERM" for term searches or "BOOL" for Boolean searches
- For Boolean searches only, clauses were set to 5
- Number of clients: 50
- Mode: "standard" (Elastic without plugin) or "vesiro" (Elastic with Vesiro's plugin)

The test was then initiated, and four tests were performed:

- Term Search with standard Elasticsearch
- Boolean Search with standard Elasticsearch
- Term Search with Vesiro's plugin
- Boolean Search with Vesiro's plugin

Each test will generate files which includes data of:

- Verification Data – data that validates the correctness of Vesiro’s plugin.
- Response time – time for a query to travel to the server and back.
- Server time - Time on the server to perform the query, including parsing, starting threads, searching, and fetching the data. Does not include network times and encode/decode to and from JSON format. Reported as took time by Elasticsearch.

## Test results

In this section the results from the tests will be presented. The tests were performed with 2 million queries for Bool search type and 10 million queries for Term search type. In addition to that the tests were configured as mentioned in the Test setup section.

### Part 1

In Table 1 the results from the four tests for part 1 can be seen.

Table 1. Results from the four tests for part 1.

	Bool			Term		
	Lucene	Vesiro	Improvement	Lucene	Vesiro	Improvement
<b>Elapsed time [s]</b>	1472	934	36.6%	942	549.47	41.6%
<b>Queries/s</b>	1359	2142	57.6%	10 620	18199	71.4%
<b>Average search time [µs]</b>	22 837	14 313	37.3%	2419	1181	51.2%

### Part 2

In Table 2 the results from the four tests of part 2 can be seen. The power of the server was measured during the tests. A slight decrease of power was seen on the Vesiro tests compared to the standard setup. The energy was then calculated for the summed total server time for all 50 clients and the average server power.

Table 2. Results from the four tests for part 2 where standard stands for Standard Elasticsearch and Vesiro for Elasticsearch with Vesiro’s plugin.

	Bool			Term		
	Standard	Vesiro	Improvement	Standard	Vesiro	Improvement
<b>Elapsed time [s]</b>	1894	1133	40.2%	2209	1850	16.3%
<b>Queries/s</b>	1109	1921	73.2%	5459	6505	19.2%
<b>Response time (median) [µs]</b>	38724	21721	43.9%	7865	6396	18.7%
<b>Server time (median) [µs]</b>	35960	19122	46.8%	5097	3539	30.6%
<b>Server time (average) [µs]</b>	42359	23451	44.6%	56836	41825	26.4%
<b>Energy for total server time [kWh]</b>	12.70	6.88	45.8%	8.36	6.06	27.5%

An analysis of the response and server time for all indexes shows that tests with Vesiro’s plugin was faster than standard setup with around 65-78%, see Table 3.

Table 3. Results of how much faster Vesiro’s plugin was compared with standard setup.

	<b>Bool</b>	<b>Term</b>
<b>Response time</b>	77.3%	65.1%
<b>Server time</b>	77.5%	74.7%

An additional test was made where the test data was replaced with the Oxford 3000 word list from the Oxford Learner's Dictionaries. The keywords of the list have been carefully selected by a group of language experts and experienced teachers as the words which should receive priority in vocabulary study because of their importance and usefulness.

Expressions in the list consisting of two words, example “no one”, was sorted out for the tests. This since they are interpreted as a Boolean search term, “no” or “one” and only terms wanted to be tested in the Term search type tests.

To shorten the test time to around the same as for the previous test, the number of queries was altered to 200 000 for Boolean and 1 000 000 queries for Term.

Table 4. Results for the four tests for part 2 with test data from Oxford dictionary where standard stands for Elastic search standard and Vesiro for Elastic search with Vesiro’s plugin.

	<b>Bool</b>			<b>Term</b>		
	<b>Standard</b>	<b>Vesiro</b>	<b>Improvement</b>	<b>Standard</b>	<b>Vesiro</b>	<b>Improvement</b>
<b>Elapsed time [s]</b>	1389	896	35.5%	589	509	13.2%
<b>Queries/s</b>	147	228	55.7%	4398	5124	16.5%
<b>Response time (median) [µs]</b>	308501	196353	36.4%	9192	7691	16.3%
<b>Server time (median) [µs]</b>	304895	193023	36.7%	5519	4248	23.0%
<b>Server time (average) [µs]</b>	337463	215638	36.1%	6308	4756	24.6%
<b>Energy for total server time [kWh]</b>	9.94	6.46	35.0%	1.89	1.40	25.9%

## Summary

This section summaries the findings and results made during this project.

Overall Vesiro’s plugin has performed better in all the test for both part 1 and 2 and decreased the server time with up to 45% compared with the standard configuration.

For the tests in part 1 the results show that the Vesiro plugin decreases the search time with 37% for Boolean search type and 51% for term search compared to standard Lucene setup. It performs 58% more queries/s for Boolean and 71% more for term search.

The results for part 2 shows that Vesiro’s plugin uses 45% less server time for Boolean search type and 26% less for term search type. Looking at the server time data for all the indexes Vesiro’s plugin was faster than standard setup in 78% of the indexes for Boolean search type and 75% for term search type. Looking at the energy from the tests in part 2, Vesiro’s plugin



then uses 46% less energy for the Boolean search type and 28% less for term search. With more efficient searches, both less energy is used, and less hardware is needed for the same amount of work.

For the tests in part 2 the results showed a higher improvement for the Boolean search type compared to the term search type. For example, queries/s performed around 50 percentage points better for Boolean than term and around 15 percentage points better for server time. While in the results from part 1 it was the opposite, here term search performed better than Boolean.

For the additional tests with the Oxford test data the results show that the Vesiro plugin uses 36% less server time for Boolean search type and 25% less for term type. The slight decrease in improvement for the Oxford tests compared with the first tests could be due to that the test data is the most common words in the English language, which results in a bigger search match and a longer time for the rating. This means that the improvement in search time is smaller in relation to the total server time which includes fetching and rating. This is also affecting the energy savings results since it is calculated with the total server time. Comparing the Oxford data with the test data containing 100 000 words provided by Vesiro, many of the expressions in the first test data seem to occur less frequently in the documents than the Oxford list.

## Conclusion

The results show a significant improvement with the Vesiro plugin compared to standard setup with an energy decrease of 26-46%, for these test setups. The results have been given with the configuration described in this report and with the data received from the tests. The tests in part 1 have not been verified since the code could not be seen by us due to a premade script including both client, search engine and plugin. In part 2 the code for the test client could be reviewed, the configuration of Elasticsearch could be verified, and logs were available.

As mentioned in this report the Vesiro software is in a prototype stage at the moment and under development and the tests have been focused on the performance of the search algorithm. When more functionality is available these results can be complemented with the impact of dynamic updates, different search types etc. When the product is more mature, it is recommended to broaden the tests and move to a more production-like environment, with Elasticsearch setup in a cluster etc, in order to conduct more real-world like scenarios.

A deeper analysis is also recommended to see how different terms affect the test result, for example why the trend of the results for Boolean versus term is the opposite for part 1 and part 2.

**RISE Research Institutes of Sweden AB**  
**ICE Data center**

Written by

---

Joachim Johansson, Daniel Olsson, Tina  
Stark

**A report within the project P2023-00192, "Evaluation of energy-saving effect of new database software", financed by the Swedish Energy Agency.**