Comparison of Search Servers for Use with Digital Repositories

Aluísio Augusto Silva Gonçalves[a] and Marcos Sfair Sunye[b]
Department of Informatics, Federal University of Paraná, Curitiba, Brazil

Keywords: Digital Libraries, Information Retrieval, Benchmark, Lucene.

Abstract: Search is a fundamental operation of storage systems, such as digital repositories, and their performance and quality directly impact the user's opinion of a system. This paper evaluates two different search engines, Apache Solr and Elasticsearch, for the same repository and reports the pros and cons of each for that specific use case. In particular, we identify that although Elasticsearch consumes less resources and responds to most queries more quickly, it may also take longer to respond in some scenarios.

1 INTRODUCTION

Search is an essential use case of databases and document storage systems, and is the focus of an entire discipline of computer science: information retrieval, defined by Manning et al. (2009) as the act of locating materials (usually documents) of an unstructured nature (such as plain text) present in large collections (usually computers) that satisfy an information need, in opposition to traditional mechanisms of data retrieval that rely on an unique identifier associated with a particular datum.

One kind of document storage system is the digital repository (or digital library), an specialized system that manages preservation and access to digital files, and which is publicly used mostly by universities and research institutes as a tool for the dissemination of scientific knowledge (Ferreira and Sunye, 2017).

To Schatz (1997), one of the fundamental aspects of digital repositories is the ability to search and filter their contents according to the user's needs. Particularly for user-facing operations, the performance of searches, as measured both in response time and quality of the results, influences the perception of the system as a whole (Joseph et al., 1989). Thus, it is in the interest of digital repository operators to evaluate different search engines in the context of a single system, so as to select the most user-friendly option.

Per the Directory of Open Access Repositories (Jisc, 2019) and the Registry of Open Access Repositories (University of Southampton, 2019), the DSpace[1] platform is the one used by the majority of public digital repositories. DSpace uses the Apache Solr[2] search server for its document search and organization needs, and Solr in turn wraps the Lucene[3] search library, which does much of the heavy lifting involved in a search.

A similar project in the same space as Solr is Elasticsearch[4], that works with new paradigms and technologies and takes advantage of today's distributed computing environments (Banon, 2010; Gospodnetić, 2015). By virtue of sharing the Lucene core library with Solr, Elasticsearch can be used, with some adaptation, in lieu of the Apache project's search server. This makes both projects candidates for the kind of comparison previously considered in this work.

With this context, the aim of this work is to evaluate the quality of results returned by these two textual information retrieval systems, Apache Solr and Elasticsearch, when operating in conjunction with a DSpace document repository. To do so, we will review basic concepts of information retrieval and examine the software to be tested, then present the evaluation methodology and test setup, then conclude with a discussion of the results and of future works stemming from them.

[a] https://orcid.org/0000-0002-1844-6540
[b] https://orcid.org/0000-0002-2568-5697

[1] https://duraspace.org/dspace/
[2] https://lucene.apache.org/solr/
[3] https://lucene.apache.org/core/
[4] https://www.elastic.co/elasticsearch

Gonçalves, A. and Sunye, M.
Comparison of Search Servers for Use with Digital Repositories.
DOI: 10.5220/0009577102560260
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 256-260
ISBN: 978-989-758-423-7
Copyright © 2020 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

2 CONCEPTS FROM INFORMATION RETRIEVAL

Current information retrieval systems work with collections of arbitrary documents that can be automatically or manually associated with the information present in each document represented as *terms* (Baeza-Yates and Ribeiro-Neto, 1999). This association is usually maintained in an inverted index, a correspondence between terms and documents.

Two main metrics are used to assess the quality of results returned by a query to an information retrieval system: *precision*, defined as the proportion of the results that are considered relevant to that query, i.e. those that are correctly present in the results; and *recall*, that considers how many relevant documents have been returned as search results, and is computed as the ratio between the number of relevant documents returned and the total number of relevant documents in the base. These metrics are used as building blocks for many others, and are henceforth denominated *primitive* metrics for the rest of this article.

Mandl (2008) notes that these metrics alone, as well as combinations such as *measure-F*, work better when results are collected in an unordered set; when the results are presented in an ordered fashion, it is possible to take this new information into account during the evaluation. Jones et al. (1995); Latha (2016) discuss two measures of particular interest to digital repositories: *precision-at-k*, which limits the set of results to those appearing on the first page shown to the user (i.e. the first *k* results); and *mean precision*, given by the arithmetic mean of the precision calculated for each relevant document as it is found in the list of results.

A limitation of these metrics is their dependence on the correct and complete identification of relevant and non-relevant documents, in particular those derived from the recall primitive. When it is not possible to evaluate all the documents in the collection from this perspective, it is advantageous to use the binary preference metric, or *bpref* (Buckley and Voorhees, 2004), defined simply as the number of times documents deemed non-relevant are retrieved before documents assumed relevant by the system.

Metrics such as *precision-at-k* and *bpref* are used, among others, as one of the metrics for benchmarks in the Text Retrieval Conference (Hersh et al., 2006), an event whose publications center on the development and evaluation of both generic and task-specific information retrieval systems (Stokes, 2006).

3 TESTED SOFTWARE

Jantz and Giarlo (2005) defines a digital repository as a platform to catalogue, preserve, and access digital files, whether they are digitized (e.g. books or newspaper issues) or native digital objects (such as spreadsheets, e-books, and multimedia files).

From the multiple digital repository systems available presently, the DSpace software originally developed by the Massachusetts Institute of Technology and Hewlett-Packard (Smith et al., 2003) can be considered the leading solution in use for publicly available repositories (Jisc, 2019; University of Southampton, 2019). That makes it a prime candidate for which to analyze the quality of information retrieval.

Underlying DSpace's search capabilities is the Apache Solr enterprise search server, which receives commands and queries over the HTTP protocol and can operate over multiple distinct document bases called *cores* (Veenhof, 2012).

Actual interaction with the cores, including insertions and queries, is done through the Apache Lucene library. Lucene indexes documents composed of key-value pairs (termed *fields*) into an inverted index structure. When responding to a query, results are sorted with the BM25 function described by Robertson et al. (1995), and relevant parts can be indicated for highlighting by the client application (Gospodnetić and Hatcher, 2005).

For DSpace, the indexed fields include file metadata such as title, author, subject, among others; the transcribed textual content of the file, when available; and the location of the file within the repository.

Despite operating in a client-server model, Solr did not originally have any support for multiple servers, which would enable features such as load balancing, database replication, and fail-over. The Elasticsearch server was created to fill these shortcomings while taking advantage of new paradigms developed after Solr's origin (Gospodnetić, 2015).

4 RELATED WORKS

Because Elasticsearch too uses Lucene as its underlying search engine, multiple comparisons can be found between it and Solr. Few of them, however, explore the performance and search result quality differences between both systems.

Coviaux (2019) presents a detailed framework for optimization of Elasticsearch search results after choosing it over Solr due to developer friendliness; however, no discussion is had about the runtime performance and requirements of either search servers, nor there is a comparison of their search results.

Correia (2016), while developing a clinical database system, ranks Solr higher due to superior documentation of its query language. Once again, the comparison does not reach a stage in which the search servers are tested in a live environment.

Kılıç and Karabey (2016) compare the replication and security aspects of Elasticsearch and Solr, favouring the former due to its greater flexibility in regards to distributed configurations.

Luburić and Ivanović (2016) perform a direct comparative analysis of both search servers directly and conclude that they perform similarly for plain text indexing, while Elasticsearch outperforms Solr on analytical queries such as those present on big data workloads.

Akca et al. (2016) provide a direct and extensive performance study of Elasticsearch and Solr, outside of any particular usage or workload, that finds that Elasticsearch has a pronounced advantage over Solr in response time with more than 100 concurrent queries, particularly when the underlying document set is still being indexed. We have found that, in the context of digital repositories and our particular document set, the actual difference is half of what was reported.

5 METHODOLOGY

Inspired by the methodology used at the Text Retrieval Conference, we designed a full benchmark for evaluating the search servers, as outlined by Dekhtyar and Hayes (2006).

The document set upon which the searches were performed is based on the contents of the Digital Archive of the Federal University of Paraná[5] (UFPR) on February 21, 2019. The document set upon which the searches were performed is based on the contents of the Digital Archive [REDACTED] on February 21, 2019.

To select the queries to be made part of the benchmark, we analysed all searches made in the Digital Archive between 2015 and August 2019. 30 queries were used, that cover the composition of over 99 % of searches. The fields queried in each of the selected search patterns are listed on Table 1.

The quality of the results was measured using the precision-at-10 and binary preference metrics. Precision-at-10 reflects the relevancy of the documents presented in the first page of search results seen by a user, and thus is a proxy measure for the perceived quality of those results. Binary preference was designed to measure the average relevancy of the search results while being resistant to incomplete relevancy judgements of the document set (Allan et al., 2005), thus being well suited for benchmarks composed of thousands of documents. These metrics are the same ones used, along with others, by the Text Retrieval Conference (Hersh et al., 2006), an event whose publications center on the development and evaluation of both generic and task-specific information retrieval systems (Stokes, 2006).

Tests were run on a DSpace instance hosted on a virtual machine with dedicated hardware, including 8 virtual cores and 19 GiB of RAM with no disk swap configured, mirroring the production setup of UFPR's Digital Archive. Tests were run on a DSpace instance hosted on a virtual machine with dedicated hardware, including 8 virtual cores and 19 GiB of RAM with no disk swap configured, mirroring the production setup of the Digital Archive. The instance could be switched to use either the built-in Solr server or a standalone Elasticsearch server for resolving search queries. Memory and resource usage during search index creation were tracked using the `pidstat` tool and information available through Linux's `/proc` file system. Resource usage during searches proved difficult to isolate from other tasks being done concurrently, and thus was not included.

6 RESULTS

Table 2 presents the overall resource consumption of the indexing process. It can be noted that Elasticsearch's resource consumption is 1.5 to 4 times smaller than Solr, for the same dataset. The fact that Solr runs in the same process as DSpace itself accounts for the increased memory usage, but not for the extra time spent (since DSpace was idle during this phase of the tests) nor for the disk space used by the search cores (which are wholly within Solr's purview).

Table 3 summarizes the metrics calculated for the search queries executed against both Solr and Elasticsearch through DSpace. When averaged over all search queries, the difference in search result quality is less than 5 %.

The response time for queries shows a less compromising story, however. While Solr starts responding to queries after about a second, Elasticsearch responses are sent back in under 500 ms. However, when many documents are returned from a query, as when one navigates the repository from DSpace's interface, Solr takes up to 5 seconds to finish its response, with Elasticsearch concluding after almost 15

[5] https://acervodigital.ufpr.br

Table 1: Fields included in each search query class.

no.	all	Has files?	Author	Date	Subject	Type
1			X	X	X	X
2			X	X	X	
3					X	
4			X		X	
5				X	X	
6					X	X
7				X	X	X
8			X			
9			X		X	X
10			X	X		
11		X		X	X	
12		X			X	
13		X		X	X	X
14		X			X	X
15		X	X		X	
16				X		
17		X	X	X	X	
18			X			X
19		X	X		X	X
20	X					
21		X	X	X	X	X
22		X	X			
23						X
24				X		X
25			X	X		X
26		X	X	X		
27		X				
28		X		X		
29		X	X			X
30		X	X	X		X

Table 2: Comparison of resource usage during indexing by the Solr and Elasticsearch search servers.

Resource	Solr	Elasticsearch
Total time	224 min	53 min
Maximum RAM usage	20 978 MiB	11 201 MiB
Index disk space	11 513 MiB	7 152 MiB

Table 3: Median precision-at-10 and binary preference metrics over all search queries, when using Solr or Elasticsearch as search server.

Metric	Solr	Elasticsearch
Precision at 10	0.3469	0.3315
b-pref	0.2148	0.2243

seconds. The 3x difference in the worst-case scenario must be carefully observed by repository operators, as it's directly noticeable by end users.

7 CONCLUSIONS

Searches in document storage systems are one of the facets through which users interact and ultimately perceive these systems, and the quality of search results has an impact on this perception. By comparing two similar search servers as part of the functioning of the same large application, this work aims to identify their strengths, weaknesses, and suitability for the domain of digital repositories.

We used the DSpace repository software as a platform for comparisons between the Solr and Elasticsearch search servers, both of which use the Lucene library as their search engine while offering different features and performance profiles.

Overall, even as it use the same underlying search engine as Solr, Elasticsearch offers significant savings in resource usage while keeping a similar level of quality in its results, but at the cost of occasional high latency while navigating the stored documents. This trade-off must be carefully observed by repository operators that might wish to switch their systems' search servers.

8 FUTURE WORKS

We intend to continue development of the benchmark used in this work and to release it publicly so that it can be applied to other repository software and search engines. We also plan to increase its scope to cover search engines not based on inverted indexes, such as K^2Treap^{H}and WT1RMQ^H (Gog et al., 2017).

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

REFERENCES

Akca, M. A., Aydoğan, T., and İlkuçar, M. (2016). An analysis on the comparison of the performance and configuration features of big data tools Solr and Elasticsearch. *International Journal of Intelligent Systems and Applications in Engineering*, 6(special issue):8–12.

Allan, J., Carterette, B., and Lewis, J. (2005). When will information retrieval be "good enough"? In *Proceedings of the 28th annual international ACM conference on research and development in information retrieval*, pages 433–440. ACM.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley.

Banon, S. (2010). ElasticSearch, Sphinx, Lucene, Solr, Xapian. Which fits for which usage? Response from kimchy.

Buckley, C. and Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32. ACM.

Correia, J. d. S. B. C. (2016). Indexação de documentos clínicos. Master's thesis, Faculdade de Engenharia, Universidade do Porto.

Coviaux, Q. (2019). Optimization of the search engine ElasticSearch. Master's thesis, Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya.

Dekhtyar, A. and Hayes, J. H. (2006). Good benchmarks are hard to find: Toward the benchmark for information retrieval applications in software engineering. In *Proceedings of the 22nd International Conference on Software Maintenance*. IEEE.

Ferreira, E. and Sunye, M. S. (2017). A method for gathering and classification of scientific production metadata in digital libraries. In *Proceedings of the 19th International Conference on Enterprise Information System*, volume 1, pages 357–364. SCITEPRESS.

Gog, S., Konow, R., and Navarro, G. (2017). Practical compact indexes for top-k document retrieval. *Journal of Experimental Algorithmics*, 22.

Gospodnetić, O. (2015). Solr vs. Elasticsearch — how to decide?

Gospodnetić, O. and Hatcher, E. (2005). *Lucene in Action*. Manning.

Hersh, W., Cohen, A., Yang, J., Bhupatiraju, R. T., Roberts, P., and Hearst, M. (2006). TREC 2005 genomics track overview. In *NIST Special Publication 500-266: 14th Text Retrieval Conference*. NIST.

Jantz, R. and Giarlo, M. J. (2005). Digital preservation: Architecture and technology for trusted digital repositories. *Microform & Imaging Review*, 34(3):135–147.

Jisc (2019). Directory of Open Access Repositories. Statistics.

Jones, G. J. F., Foote, J. T., Jones, K. S., and J., Y. S. (1995). Video mail retrieval: the effect of word spotting accuracy on precision. In *Proceedings of the 1st International Conference on Acoustics, Speech, and Signal Processing*, pages 309–312. IEEE.

Joseph, B., Steinberg, E. R., and Jones, A. R. (1989). User perceptions and expectations of an information retrieval system. *Behaviour & Information Technology*, 8(2):77–88.

Kılıç, U. and Karabey, I. (2016). Comparison of Solr and Elasticsearch among popular full text search engines and their security analysis.

Latha, K. (2016). *Experiment and Evaluation in Information Retrieval Models*. CRC Press.

Luburić, N. and Ivanović, D. (2016). Comparing apache solr and elasticsearch search servers. In *Proceedings of the Sixth International Conference on Information Society and Technology*, volume 2, pages 287–291. Society for Information Systems and Computer Networks.

Mandl, T. (2008). Recent developments in the evaluation of information retrieval systems: Moving towards diversity and practical relevance. *Informatica*, 32(1):27–38.

Manning, C. D., Raghavan, P., and Schütze, H. (2009). *An Introduction to Information Retrieval*. Cambridge UP.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at TREC-3. In *NIST Special Publication 500-225*, volume 3, pages 109–126. NIST.

Schatz, B. R. (1997). Information retrieval in digital libraries: Bringing search to the net. *Science*, 275(5298):327–334.

Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., and Walker, J. H. (2003). Dspace: An open source dynamic digital repository. *D-Lib Magazine*, 9(1).

Stokes, N. (2006). Trec: Experiment and evaluation in information retrieval. *Computational Linguistics*, 32(4):563–567.

University of Southampton (2019). Registry of Open Access Repositories. Browse by repository software.

Veenhof, N. (2012). Improving Acquia search and the Apache Solr module. Master's thesis, Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS