International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

An Analysis on the Comparison of the Performance and Configuration Features of Big Data Tools Solr and Elasticsearch

Mustafa Ali AKCA*[1], Tuncay AYDOĞAN[2], Muhammer İLKUÇAR[3]

Accepted 3rd September 2016

Abstract: Today, every kind of text, audio and visual data, which are thought to be transformed into pieces of information, are stored for long periods of time for processing. The concept of Bid Data is not only associated with the data stored, but also with the system involving hardware and software that collects, processes, stores, and analyzes the data. As the data grows bigger, their physical storage options must be provided in a distributed architecture. Solr and Elasticsearch are among the most preferred tools which makes this storage process easier. As a part of Apache Lucene project, Solr is a software which was started to be developed in 2004 with the searching features of full text, multiple search, dynamic clustering, database-integrated, open source and elasticity. Similarly, Elasticsearch is a new open-source tool for real-time, full-text and distributed search, which was launched in 2010 using the Lucene library. Although Solr and Elasticsearch have similar features, there are many parameters that differentiates one from the other such as intended use, type of use, and query and indexing performances. This study researches and analyzes the differences between Solr and Elasticsearch with regards to their query and indexing speeds, ease and difficulties of use, configuration forms, and architectures in light of the literature, and the results are discussed regarding these tools' performances.

Keywords: *Big data, Elasticsearch, Solr*

1. Introduction

Huge amount of data stack which cannot be stored nor processed by traditional methods is called Big Data [1]. Based on the fact that most of the smart devices communicate on the Internet and the technological devices and computers will communicate through cloud computing, it is easy to predict that much bigger data is to be produced in the future [2]. It is impossible to store, process, and analysis this big data by conventional tools. Therefore, many tools were developed for the data which can be regarded as Big Data. It was observed that Big Data tools and techniques that are used in processing this data provided better performance results in comparison with traditional tools and techniques [3].
One of these tools is Solr which is an open source, Java-based tools that uses Apache Lucene search library and it can be used on servers such as Tomcat, Jetty, etc. [4]. According to July, 2016 figures of DB-Engines Ranking of Search Engines, Solr is the second most popular search engine software [5]. With its 5th version, Solr has become a separate software instead of being a Java package [6].
Elasticsearch is one of the other tools that is used in analyzing and process Big Data. It is an open-source, Lucene-based content search and analysis tool that was developed in Java. This tool, which began to be developed in 2010, is used by many domestic/foreign organizations. It is also a tool that can be set up in Cloud servers purchased through Google for Works, and it is in a sense supported by Google [7]. According to July, 2016 figures of DB-Engines Ranking of Search Engines, Elasticsearch is the most popular search engine software [5].
Because of the fact that both of Solr and Elasticsearch tools are based on Apache Lucene library, and they have similarities between the intended use and methods, supported protocols, and coding languages, today they are compared and contrasted by users in different categories. Users try to decide which one is the best tool for them based on their own needs and system requirements.
This study, therefore, collects comparative data conducted so far between Solr and Elasticsearch. These collections involve technical configurations, performance tests, etc. This study is thought to contribute to software developers in using Solr and Elasticsearch.

2. Solr

Solr is an open source enterprise search platform, written in Java, from the Apache Lucene project an its major features include full-text search, hit highlighting, faceted search, real-time indexing, dynamic clustering, database integration, NoSQL features and rich document (e.g., Word, PDF) handling [6][8].

A. Technical Specifications of Solr

Each data in Solr is named as Document. A simple analogy between Solr and Java Database is as follows:

Java Class = Solr Document
Java Class Attribute = Document Field
and
Database table = Solr Document
Column in the database table = Document Field

A Solr Document can be generated by method of creating a table

[1] *Süleyman Demirel University, Faculty of Education, Isparta, TURKEY*
[2] *Süleyman Demirel University, Faculty of Technology, Isparta, TURKEY*
[3] *MAKU, Vocational School of Technical Sciences, Burdur, TURKEY*
** Corresponding Author: Email: mustafaakca@sdu.edu.tr*
Note: This paper has been presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.

in a database or building a Java class.
Data indexed by Solr provides a fast response based on query type through Request method on HTTP. In addition that query results can be instantly listed on the screen, one can also print out in XML, JSON, CSV, etc. formats.
Solr uses Lucene library in indexing, analyzing, and searching processes, which involves filtering on existing structures, faceting, caching, and distributed architecture support. While Solr can be used as an API server similar to REST, it can also operate on protocols that are supported by HTTP/XML or JSON coding languages. Additionally, it supports libraries that work independently in Java and enables specifications.
Rules regarding the Documents designed in Solr are stored in Schema.xml file. There are <field> settings in Schema file which corresponds to "column" in traditional databases. These settings are composed of the parameters of field name, whether it is indexed in Solr search, whether it is stored and responds to findings, whether it is multi valued, and what type of data it includes [9].
A sample schema file involves the following data:

```
  <field name="id" type="string" indexed="true"
stored="true" required="true" multiValued="false" />
  <field name="sku" type="text_en_splitting_tight"
indexed="true" stored="true" omitNorms="true"/>
  <field name="name" type="text_general" indexed="true"
stored="true"/>
<field name="manu" type="text_general" indexed="true"
stored="true" omitNorms="true"/>.
```

B. How to Use Solr

After the installation of Solr, some sample data sets in "exampledocs" file can be used. A sample data file is as follows:

Table 1. Sample document

```
TWINX2048-3200PRO
CORSAIR  XMS 2GB (2 x 1GB) 184-Pin DDR SDRAM
Unbuffered
    Corsair Microsystems Inc.
    corsair
    electronics
    memory
    CAS latency 2, 2-3-3-6 timing, 2.75v
    185
    5
    true
    37.7752,-122.4232
    2006-02-13T15:26:37Z
    electronics|6.0 memory|3.0


    VS1GB400C3
    CORSAIR ValueSelect 1GB 184-Pin
    Corsair Microsystems Inc.
    corsair
    electronics
    memory
    74.99
    7
    true
    37.7752,-100.0232
    2006-02-13T15:26:37Z
    electronics|4.0 memory|2.0
```

Data in Table 1 can be listed by the following methods on sonar browser after indexed in Solr.

Request:
http://localhost:8983/solr/select?q=*:*&wt=json

- Lists all data
- wt=json Listed in JSON format

Request:
http://localhost:8983/solr/select?q=id:VS1GB400C3&wt=json&fl=id,name

- Lists data with VS1GB400C3 id parameter value
- wt=json Listed in JSON format
- fl=id,name Lists only id and name information in results page.

3. Elasticsearch

Elasticsearch is also another distributed and open-source analysis and search tool which was developed using Lucene library. It was designed for Scalability, security, and easy management. By means of its query language that indexes structural and non-structural, and time-based data, provides rapid search and powerful analysis capabilities.

A. Technical Specifications of Elasticsearch
Elasticsearch is a document-based tool. Each entry in Elasticsearch is a structured JSON document. In other words, each data sent to Elasticsearch for indexing is a JSON document [10]. It is necessary to know some basic concepts to understand the structure of Elasticsearch. These are as follows:

Index: An index is like a database in a relational database. Each Elasticsearch index is a structured JSON document.

Type: A type is like a table in a relational database. Elasticsearch indexes can include more than one type.

Mapping: This is the process of managing how the data transferred types on indexes should be handled. It is equal to identification of data type (string, integer, double, boolean) before adding the data in the traditional databases. Elasticsearch automatically realizes mapping based on the data, and however this can also be done manually.

Shard/Sharding: It is the horizontal distributing of data to other machines on the network in database technologies. Thanks to sharding technology, data can be divided into pieces based on the existing physical facilities. Elasticsearch divides indexes into 5 shards by default to store them.

Replica: A replica is a copy of the primary shard. It is mainly used to prevent data loss in index documents as well as to distribute the load of the indexes in distributed database architectures. Elasticsearch indexes creates one copy of each shard document when they are created at first.

Schema: Being schema-free, Elasticsearch automatically realizes index, type, filed type descriptions and does not require the users to do so.
Elasticsearch can be managed by JSON commands over HTTP with RESTful API. Thereby, it can operate on many coding languages with no problem.

B. How to Use Elasticsearch

Elasticsearch can be used in computers with JVM installed after downloading. It does not require configuration for first use. Data adding/query can be conducted on curl (client URL library) as it is for RESTful API.

Table 2. Sample of adding data

```
mustafa:~ mustafa$ curl -XPUT
localhost:9200/deneme/article/1 -d '
> {
> title: "ElasticSearch",
> content: "ElasticSearch is developed in Java, open source,
lucene-based, scalable full-text search engine and data analysis
tool.",
> date: "2016-07-07T12:00:00",
> author: "Mustafa Ali AKCA"
> }'
{"ok":true,"_index":"deneme","_type":"article","_id":"1","_v
ersion":1}
```

Through commands shown in Table 2, type 1 id data named "article" found in "deneme" index was added. Command and query result in order to query this command is shown in Table 3.

Table 3. Query sample

```
mustafa:~ mustafa$ curl -XGET
localhost:9200/deneme/article/1?pretty=true
{
  "_index" : "deneme",
  "_type" : "article",
  "_id" : "1",
  "_version" : 1,
  "exists" : true, "_source" :
{
title: "ElasticSearch",
content: "ElasticSearch is developed in Java, open source,
lucene-based, scalable full-text search engine and data
analysis tool.",
date: "2016-07-07T12:00:00",
author: "Mustafa Ali AKCA"
}
}
```

While using Elasticsearch indexes, curl library can be used in addition to "Sense" plugin in Google Chrome.
Elasticsearch provides client support for many platforms including Java, Ruby, PHP, Python, and .NET, and it can be managed through coding languages of these platforms.

4. Technical Comparison Between Solr And Elasticsearch

A. Overview

Table 4 provides a general overview for the features of Solr and Elasticsearch tools.

B. Access and Data Processing

Detailed data can be found in Table 5 regarding the comparison between Solr and Elasticsearch based on their data processing methods and direct or indirect access.

Table 4. Overview

	Solr	Elasticsearch
Inital Release	**2004**	**2010**
Current Release	6.1.0 June 2016	2.3.3, May 2016
Licence	Open Source	Open Source
Database as a Service	No	No
SQL	No	No
Implementation language	Java	Java
Server operating systems	All OS with a Java VM and a servlet container (Tomcat, Jetty, etc)	All OS with a Java VM
Based	Apache Lucene	Apache Lucene
Admin Interface	Embedded Available	Plugin Needed (Bigdesk, Kopf, ElasticHQ, Paramedic, Marvel, etc)

Table 5. Access and data processing

	Solr	Elasticsearch
Access Methods	RESTful http API, Java API	RESTful http API, Java API
Supported client language	.Net, Erlang, Java, JavaScript (XML or JSON), Perl, PHP, Python, RubyScala	Clojure, Cold Fusion, Erlang, Go, Groovy, Haskell, Java, Javascript, kotlin, .NET, Ocaml, Perl, PHP, Phyton, R, Ruby, Scala, Smalltalk, Vert.x
Data Import	JDBC, MySQL, CSV, XML, Tika, URL, Flat File	Wikipedia, MongoDB, CouchDB, RabbitMQ, RSS, Sofa, JDBC, Filesystem, Dropbox, ActiveMSQ, LDAP, Amazaon SQS, St9, OAI, Twitter, DynamoDB, Git, GitHub, Hazelcast, JMS, Kafka, neo4j, Redis, Solr, St9, Subversion

C. Data and Distributed Architecture

Data storage, data processing, and distributed architecture features of Lucene-based Solr and Elasticsearch are found in Table 6.

Table 6. Data and distributed architecture

	Solr	Elasticsearch
Distributed Architecture	Supports	Supports
Sharding	Yes	Yes
Replication	Yes	Yes
Shard Splitting	Yes	No
Changing Number of Shards	Yes	No
Relocating	Yes	Yes
Routing	Yes	No
Schemaless	Yes (After 4.4)	Yes
Map Reduce	No	No
Automatic Shard Rebalancing	No	Yes
Distributed Group By	Yes	No

D. Searching and Indexing

Solr and Elasticsearch have different indexing and searching features based on their basic structures and the plugins developed by users (*see Table 7*).

Table 7. Searching and indexing [11]

	Solr	Elasticsearch

Custom Analyzers and Tokenizers	Yes	Yes
Multiple Indexes	Yes	Yes
Realtime Search/Indexing	Yes	Yes
Multiple document types per schema	No	Yes
Schema Changes	Yes	Yes
Field Copying	Yes	Yes
Distributed Search	After 2012	Yes
Term Vectors API	Yes	Yes
Autocomplete	Yes	Yes
Highlighting	Yes	Yes
Structured Query DSL	No	Yes

5. Performance Comparison Between Solr And Elasticsearch

One of the issues that Solr and Elasticsearch tools are compared most is their performance figures. Even though these tools are quite similar to one another in terms of being Lucene-based, different figures can be obtained related to their performance. In studies on performance comparisons, it was found out that Solr and Elasticsearch do not greatly have superiority over each other [12][13]. However, sometimes Solr had better performance over Elasticsearch based on intended use, place of use, and changes in some parameters, and vice versa. In this part of this study, some of the performance results conducted so far on Solr and Elasticsearch are listed.

A. Flax.co.uk Tests

Flax.co.uk design, build and support open source powered search applications based in Cambridge U.K. Flax has completed many performance tests on Solr and Elasticsearch. Cluster specifications used in a test are as follows:

- Two machines, each with 96GB RAM
- Two instances of SolrCloud or Elasticsearch on each
- Each instance has 24GB JVM heap
- Four shards
- No replicas

In the first stage of this test, a data consisting of 40 million pieces with 5 to 20 words was created. The results of Solr and Elasticsearch indexes are listed below:

- Elasticsearch indexed them in 30 minutes
- Total index size was 8.8 GB (easily cacheable)
- Solr indexed them in 43 minutes
- Total index size was 7.6 GB

Then, a data consisting of 40 million pieces with 200 to 1000 words was indexed. The results of Solr and Elasticsearch indexes are listed below:

- Elasticsearch indexed them in 179 minutes
- Total index size was 363 GB (not completely cacheable)
- Solr indexed them in 119 minutes
- Total index size was 226 GB

Based on these results, it can be concluded that Elasticsearch performs better with short documents, while Solr had better performance with long documents.

B. Search Test with Indexing

In another study, the first test was designed for conducting a search on servers with similar features (4 nodes, 4 index shards, no replication, 16 GB per node) with 20 million documents. In this process, no extra procedure was followed regarding Solr and Elasticsearch nodes. As seen in Figure 1, it was found out that most of the Elasticsearch queries were completed between 0.10 sec and 0.22 sec as a result of 1000 search inquiries sent to nodes. The same test provided the results of 0.12 sec and 0.54 sec for Solr tool. [13].

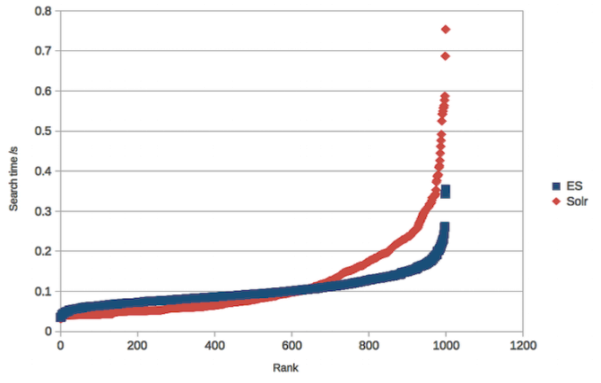

Figure 1. Search Test [15]

C. Search Test with Indexing Load

In the second test, servers were requested to operate search inquiries while engaged in indexing process. These tests were also conducted with 20 million entries on servers with similar features. As seen in Figure 2, Elasticsearch had results varying between 0.14 sec and 0.34 sec, Solr completed search inquiry between 0.24 sec and 0.68 sec. Elasticsearch was slightly better than Solr in search tests with indexing load.

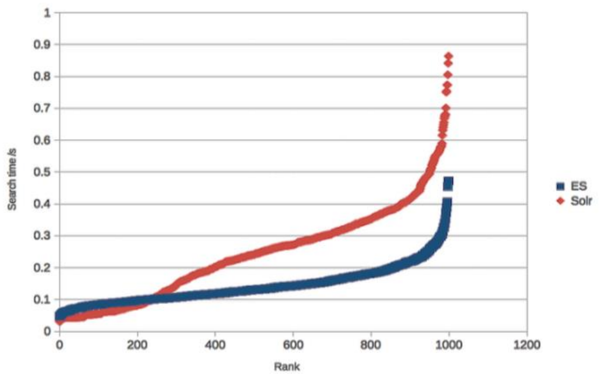

Figure 2. Search Test with Indexing Load [15]

D. QPS Test

Queries per second (QPS) is a common measure of the amount of search traffic an information retrieval system, such as a search engine or a database, receives during one second [14].

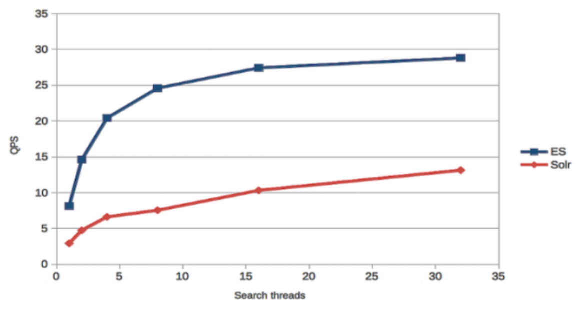

Figure 3. QPS Test [15]

In QPS test conducted between Elasticsearch and Solr, indexing was continued as a continuation of the previous test. QPS figures were measured for both tools while indexing process was active.

As seen in Figure 3, Elasticsearch reached at 30 QPS speed, while Solr was remained at 15 QPS speed when sent similar search requests.
In another QPS test conducted with 40 million pieces of data (between 200 and 1000 words), it was observed that Solr had better performance (Figure 4) [12].

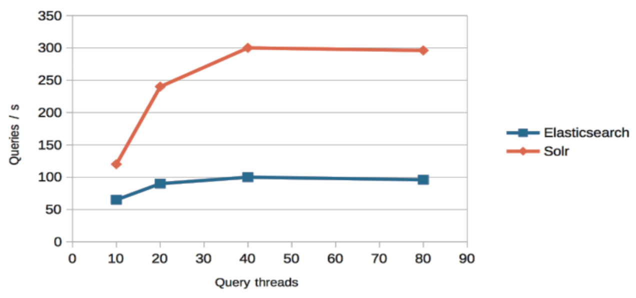

Figure 4. QPS Test

6. Conclusions

Solr and Elasticsearch are two similar search platforms which are both Lucene-based, has similar coding languages, intenden use and styles. Solr is one of the search tools that could maintained its popularity even in today's world.
Elasticsearch, one the other hand, was developed as a way of completing the deficiencies of Solr, and it had a significant place in markets in such a short span of 6 years (2010-2016).

Table 8. DB-Engines july, 2016 figures

Jul 2016	Jul 2015	DBMS	Jul 2016	Jun 2016	Jul 2015
1.	⬆2.	Elasticsearch	88.62	+1.21	+18.46
2.	⬇1.	Solr	64.69	+0.63	-14.59
3.	3.	Splunk	46.65	+1.42	+5.00
4.	4.	MarkLogic	9.75	+0.19	-1.38
5.	5.	Sphinx	8.05	-0.28	-1.68

Taking a look at the July 2016 figures published by most-respected DB-Engines web site by database users, Elasticsearch use/liking has increased compared to the previous year, and it moved ahead of Solr (*see Table 8*).
Considering the previous studies in literature and the technical data and performance tests of this study, it is seen that Solr and Elasticsearch were better than one another on different tests. These comparison results obtained through this study and the other studies conducted in literature are as follows:

- They are similar tools in terms of technical features.
- Elasticsearch supports more coding language than Solr.
- Solr uses less disk space considering the size of data after indexing.
- Regarding the indexing duration, Elasticsearch had better performance in short data, while Solr was better in long data.
- Both tools are rapid search tools.
- Close performance results were obtained in many different studies.
- Solr and Elasticsearch may have quite different performance characteristics in certain cases [12]
- QPS speed might vary depending on the type of data.
- It is quite hard to predict which one will have higher performance before the tests.
- If a choice shall be made between these two search engines, it is necessary to conduct a test based on the purpose of work.

Acknowledgment

We deeply send our regards and thanks to DB-Engines.com, www.solr-vs-elasticsearch.com, and ww.Flax.co.uk web sites for the information they shared with us regarding the collection of technical data and performance tests about Solr and Elasticsearch.

References

[1] F. Ohhorst, "Turning Big Data Into Big Money", Big Data Analytics, , New Jersey, AB.D., 2013.
[2] Science Clouds., https://portal.futuregrid.org/., Last Access : 13.07.2016
[3] S. Ramamorthy, S. Rajalakshmi, "Optimized Data Analysis in Cloud using BigData Analytics Techniques," 4th ICCCNT Conferense, Tiruchengode, India, 2013.
[4] C. Yeşilkaya, "Apache Solr Kurulumu", https://blog.kodcu.com/2013/03/apache-solr-kurulumu-ornek-sorgulama/ Last Access : 13.07.2016.
[5] DB-Engines Ranking of Search Engines, http://db-engines.com/en/ranking/search+engine, Last Access : 13.07.2016
[6] Apache Solr, https://tr.wikipedia.org/wiki/Apache_Solr, Last Access : 13.07.2016
[7] M.A. Akca, T. Aydoğan, "Elasticsearch Yük Dengeleme Işleminin Manuel Yapılandırılması Ve Başarım Ölçümü İçin Yazılım Geliştirilmesi", "Selcuk University, Journal of Engineering, Science & Technology, 4/2, 121-130, 2016
[8] Solr, "http://lucene.apache.org/solr", Last Access : 13.07.2016
[9] Solr'a Giriş ve Solarium, http://www.sonsuzdongu.com/blog/solr-a-giris-ve-solarium, Last Access : 13.07.2016
[10] H. Akdoğan, "Elasticsearch", https://blog.kodcu.com/2013/08/elasticsearch/, Last Access : 14.07.2016
[11] Apache Solr vs Elasticsearch, http://solr-vs-elasticsearch.com//, Last Access : 14.07.2016
[12] C. Hull, "Elasticsearch and SolrCloud a performance comparison", http://www.slideshare.net/charliejuggler/lucene-solrlondonug-meetup28nov2014-solr-es-performance?from_action=save", Last Access : 17.07.2016
[13] Tom, "Elasticsearch vs Solr Performance", http://www.flax.co.uk/blog/2015/12/02/elasticsearch-vs-solr-performance-round-2/, Last Access : 15.07.2016
[14] Queries per second, https://en.wikipedia.org/wiki/Queries_per_second, Last Access : 15.07.2016