Name: Harpreet Singh Siddhu
Course – CS 410 Text Information Systems
Technology Review – Full Text Search Engines – Lucene, Solr, Elastic Search

Introduction –

What is full text search?
Full-text search is widely used in the typical search engine. The search basically works by scanning every word in the article, creating an index on each word, indicating the number and location of occurrences of the word in the article, and when the user queries, the retrieval program searches based on the pre-established index and feeds the search results back to the user's search method. This process is similar to the process of looking up a word in a dictionary using the search Word table.

Problem – What type of data are we dealing with?
The data in our lives is basically divided into two types –

1. Structured data
2. Unstructured data

The third possible category of the data is semi-structured.

Based on the data structure, the search is divided structured data search and unstructured data search. For structured data search, we generally store them in relational databases.

But for unstructured data, there are two methods for full text search –

1. Sequential Scanning
2. Full Text Retrieval

Sequential Scanning Vs. Full Text Retrieval –
Sequential scanning is undoubtedly very slow and time consuming. But can we optimize it? Question is yes we can optimize the search by introducing some kind of structure to the unstructured data. Full text search or text retrieval engines basically structures the unstructured data to search data and that structure is called Index. Indexing is the way to generate keywords out of the text or string and to utilize them in the search.

Lucene, Solr, Elastic Search?
Lucene, Solr and Elastic Search are the modern mainstream search engines which are used highly for information retrieval from the unstructured data/datasets. They are indexed based on the inverted index. Inverted index (English: Inverted index), also often referred to as a reverse index, place file, or reverse file, is an indexed method that is used to store the mapping of a word in a document or group of documents under a full-text search. It is the most commonly used data structure in the document retrieval system.

Body –

What is Lucene?

Lucene is a Java full-text search engine that is written entirely in Java. Lucene is not a complete application, but rather a code base and API that can be easily used to add search functionality to an application.

Lucene is scalable, provides high performance indexes thru its powerful yet simple API. It is capable of indexing 150GB/hour over modern hardware with small RAM requirements (1 MP heap).

Lucene search algorithm is powerful, efficient and accurate. It uses ranking search first to return the best results. It supports many query types like phrase query, wild character queries, proximity queries, range queries etc. It provides pluggable ranking models like Okapi BM25 and vector space models.

Best part is that it is available as open source software under Apache licensing. It is 100% java based but implementation in other programming languages are also available.

Lucene is a framework and it requires a lot of learning to understand how it works and skilled use of Lucene is very complex.

What is Solr?

Apache Solr is an open source search platform and it built on Java based framework called Lucene. Solr is a matured product with strong and extensive user community. It provides distributed indexing, replication, load-balancing queries, and automatic failover and recovery. If it is properly deployed and well managed, it can become a highly reliable, scalable, and fault-tolerant search engine. Many internet giants, such as Netflix, eBay, Instagram and Amazon (Cloud Search), use SOLR because it is able to index and search multiple sites.

The important features of Solr are Full-Text search, faceted search, real time indexing, dynamic cluster, database integration, NoSQL support and rich document processing such as MS word and PDF files.

What is Elastic Search?

Elastic search is an open source (Apache 2 license) and is a restful search engine built on the Apache Lucene library. Elastic search was launched in the years after SOLR. It provides a distributed, multi-tenancy full-Text search engine with HTTP Web interface (REST) and no schema JSON documents.

Elasticsearch is a distributed, RESTful search and analytics engine which include indexes that can be partitioned into shards, and each shard can have multiple replicas. Elasticsearch is fast. Really, really fast. It is useful when you need –

1. Rapid results – Get Answer instantly
2. Scalability – Scales horizontally and can support jillions of events per second
3. Relevance – Rank you search based on term frequency or recency to popularity or beyond.
4. Resiliency – Cluster are always safe and available. It all works in distributed environment.

Comparison – Elasticsearch Vs. Solr

Due to the complexity of Lucene, it is rarely considered as the first choice for search, so here I will focus on comparing Elasticsearch and SOLR.

Elasticsearch vs. Solr. Which one is better? What's the difference between them? Which one should you use?

Apache Solr is approximately 15 years old matured project with a large number of active development and user community. Solr was first released as an open source in 2006 and has been a search engine for anyone who need search functionalities specially in websites. Solr provides rich search functionality not just simple text search but faceted search, powerful filtering, pluggable rankers like Okapi BM25, language detection and much more.

On the other hand, Elasticsearch was introduced in 2010 as an open source Apache project in the market. Since it was launched in more modern world, it is built on more modern principles and for more modern use cases. Elasticsearch is easier to handle and but can do fast search than Solr.

Comprehensive comparison

1. **Trends** – If we look at the Google search trends for both of these products, Google trends suggest that Elasticsearch is attractive compared to SOLR, but that does not mean that Apache Solr has died. While some may not think so, SOLR is still one of the most popular search engines, with strong community and open source support.

2. **Installation and Configuration** – The Elasticsearch is easy to install and very lightweight compared to SOLR. In addition, you can install and run Elasticsearch within minutes. However, this ease of deployment and use can be a problem if Elasticsearch is poorly managed. The JSON-based configuration is simple, but if you want to specify a comment for each configuration in the file, it is not for you. Overall, if your app is using JSON, Elasticsearch is a better choice. Otherwise, use SOLR, because its schema.xml and solrconfig.xml are well documented.

3. **Community** – SOLR has a larger, more mature user, developer and contributor community. ES has a small but active user community and a growing community of contributors. SOLR is the real open source community code. Anyone can contribute to SOLR and select the new SOLR developer (also known as the submitter) based on merit. Elasticsearch is technically open source, but not so important in spirit. Anyone can see the source, anyone can change it and contribute, but only Elasticsearch employees can actually make changes to Elasticsearch. SOLR contributors and submitter come from many different organizations, and the Elasticsearch submitter is from a single company.

4. **Maturity level** – SOLR is more mature, but grows fast and I think it's stable.

5. **Document** – Solr scored very high here. It is a very well-documented product with clear examples and API use case scenarios. Elasticsearch's documentation is well-organized, but it lacks good examples and clear configuration instructions.

6. **Popularity** – Elasticsearch is more popular among new developers due to its ease of use. But, if you are used to working with SOLR, continue to use it because migrating to Elasticsearch does not have a specific advantage. Elasticsearch is a better choice if you need it to handle analytic queries in addition to searching for text. If you need a distributed index, you need to select Elasticsearch. Elasticsearch is a better choice for cloud and distributed environments that require good scalability and performance.

7. **Domination** – Elasticsearch dominates the open source log management use case, and many organizations index their logs in Elasticsearch to make them searchable. Although SOLR can now be used for this purpose, it just misses the idea.

8. **Orientation** – SOLR is still more oriented towards text search. On the other hand, Elasticsearch is typically used to filter and group-analyze query workloads-not necessarily text searches. Elasticsearch developers put a lot of effort into the Lucene and Elasticsearch levels to make such queries more efficient (lower memory footprint and CPU usage). Therefore, Elasticsearch is a better choice for applications that require not only text search but also complex search time aggregation.

9. **Easy To Start** – Elasticsearch is easier to get started with, a download and a command to start everything. SOLR has traditionally needed more work and knowledge, but SOLR has recently made great strides in eliminating this and is now only trying to change its reputation.

10. **Performance** – In terms of performance, they are roughly the same. "Roughly," because no one had done a full and unbiased benchmark test.

11. **Operation** – In operation, Elasticsearch is relatively simple to use-it has only one process. SOLR relies on Apache Zookeeper in its fully distributed deployment mode. Zookeeper is super mature, super widely used and so on, but it is still another active part. If you are using Hadoop, Hbase, Spark, Kafka or some other newer distributed software, you may already be running zookeeper somewhere in your organization.

12. **Indicators** – If your organization like monitoring and indicators, then forgot about Solr and just with Elasticsearch and you will go to heaven. Elasticsearch has more indicators than the New Year's Eve can squeeze in Times Square! SOLR exposes key indicators, but far less than Elasticsearch.

Conclusion

So, what is SOLR or Elasticsearch? Sometimes it's hard to find a definite answer. Whether you choose Solr or Elasticsearch, you first need to understand the right use cases and future requirements. But in short, both are feature-rich search engines. And if you design and implement them properly they will provide the same performance more or less.

References:

1. https://www.datanami.com/2015/01/22/solr-elasticsearch-question/
2. https://www.elastic.co/cn/
3. https://logz.io/blog/solr-vs-elasticsearch/
4. https://sematext.com/blog/solr-vs-elasticsearch-differences/