International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-6, August 2020

Optimizing Search and Data Analytics of Twitter Data using Elastic Search Algorithms

Subhani Shaik , Nallamothu Naga Malleswara Rao

***Abstract:** Quick data acquisition and analysis became an important tool in the contemporary era. Real time data is made available in World Wide Web (WWW) and social media. Especially social media data is rich in opinions of people of all walks of life. Searching and analysing such data provides required business intelligence (BI) for applications of various domains in the real world. The application may be in the area of politics or banking or insurance or healthcare industry. With the emergence of cloud computing, volumes of data are added to cloud storage infrastructure and it is growing exponentially. In this context, Elasticsearch is the distributed search and analytics engine that is very crucial part of Elastic Stack. For data collection, aggregation and enriching it Beats and Logstash are used and such data is stored in Elasticsearch. For interactive exploration and visualization Kibana is used. Elasticsearch helps in indexing of data, searching efficiently and performing data analytics. In this paper, the utility of Elasticsearch is evaluated for optimising search and data analytics of Twitter data. Empirical study is made with the Elasticsearch tool configured for Windows and also using Amazon Elasticsearch and the results are compared with state of art. The experimental results revealed that the Elasticsearch performs better than the existing ones.*

***Index Terms:** Elastic search, indexing, searching, data analytics, cloud computing, Amazon Elastic search*

I. INTRODUCTION

Elasticsearch is scalable search, an open source and a data analytics engine. It facilitates us to store, perform search and data analytics over large volumes of data in almost real time. It enables applications in the real world with complex search and analysis features. It is a distributed system that works on top of Lucene tools for various purposes like guessing, indexing and so on. It is based on REST API that provides powerful interface to different real world applications. It is at the heart of Elastic Stack. The stack includes Kibaba (visualization), Elasticsearch (search and data analytics), Beats and Logstash (collection and aggregation of data besides enriching it). The Elasticsearch is capable of supporting different kinds of data such as geospatial data, numerical data, structured and unstructured data. It has capabilities to store data effectively and index it for near real time search features. It facilitates not only obtaining data needed, but also supports data aggregation and discovery of patterns or trends from data that are otherwise hidden. It supports deployment in distributed environment and is made scalable to meet growing volumes of queries.

Revised Manuscript Received on August 15, 2020.
* Correspondence Author
Subhani Shaik *, Research Scholar, Department of CSE, Acharya Nagrjuna University, Guntur, India.
Nallamothu Naga Malleswara Rao , Professor, Department of IT, RVR & JC College of Engineering, Chowdavaram, Guntur, (A.P.), India.

It promotes flexibility and speed and serves plenty of use cases. The use cases include its usage as search app embedded into web sites, storage and analysis of security events, metrics and logs;modelling behaviour of data in real time using machine learning techniques; business workflow automation; integrating, managing and analysing besides using Elasticsearch as a system known as Geographic Information System (GIS); work as bioinformatics tool. It is being used by enterprises in amazing ways and means with different unprecedented use cases. Elasticsearch search engine provides plenty of solutions to different real world problems. In this paper, it is used as a tool for extracting Twitter data, index it, search and perform data analytics. Thus it is used to optimize search and data analytics of Twitter data.

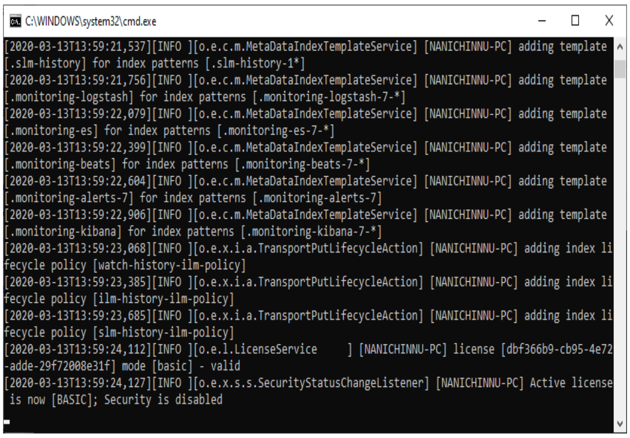

Fig 1: Locally configured Elasticsearch

Fig 2: Cluster configuration details in the local machine

An empirical study is made using it configured in Windows Operating System (OS) (shown in Figure 1 and Figure 2) and also as Amazon Elasticsearch. From the literature, different other search engines are found and they are compared with the Elasticsearch. The results revealed that for the cloud based distributed systems, Elasticsearch outperforms other search engines. Our contributions in this paper are as follows.

1. We configured Elasticsearch locally and in Amazon cloud for empirical study and different search and analysis operations are made on Twitter data.
2. Observations are made in terms of search time and throughput and the performance of the Elasticsearch is compared with other tools.

The rest of the paper structure has been prepared as follows. Review the literature related to section 2, distributed search engines like Elasticsearch. Section 3 provides important information related to Elasticsearch. Section 4 presents the Elasticsearch architectural overview and its Amazon's version. Section 5 presents experimental results. Section 6 concludes on paper and provides instructions for future work.

II. RELATED WORK

This section reviews literature on different distributed search engines including the Elasticsearch. Langi *et al*. [1] focused on the usage of Elasticsearch along with Logstash and Twitter River for analysing Twitter data. They found the improvements with Elasticsearch. Bendechache *et al*. [2] explored the importance of Elasticsearch and its usage in different real world applications. They implemented Elasticsearch in Linknovate.com and leverage it with a fault tree model. Then the tree model is associated with Stochastic Petri Net (SPN). They employed metrics such as availability ad MTTF to know its reliability. They found best results with Elasticsearch. However, their mode needs to be evaluated yet. Betke and Kunkel [3] studied many real time applications such as FUSE, Grafana, Elasticsearch, HPC applications and I/O monitoring in real time. With different search engines they built HPC online monitoring application and found its effectiveness. They found that Elasticsearch performance was better.

Nagi [4] explored different search engines that are distributed in nature. They opined that big data needs to be handled well in cloud platforms with effective search engines. The cloud platforms they explored include Microsoft Azure and Amazon EC2 to mention few. They used Hadoop and Solr distributed programming frameworks as well. They studied different factors such as petitioning, replication, consistency, fault-tolerance, manageability and high performance. They intended to fix memory leakage problem associated with LuMongo in future. Thacker *et al*. [5] on the other hand investigated on Elasticsearch in distributed computing environments like cloud. They used two indexing techniques for empirical study. They are known as non-nGram indexing and nGram indexing.

Nanamanthrao and Thejaswini [6] proposed a framework to study clickstream data analytics along with visualization techniques. They used different tools like Apache Hadoop, Spark, Kafka, Elasticsearch and Kibana. They made a recommendation engine and found the utility of Elasticsearch. Shah *et al*. [7] used Elasticsearch for exploring storage, indexing and searching of social media data. They proposed an architecture for search and data analytics. It includes Kibana as frontend for making queries and providing data visualization. They found the usefulness of Elasticsearch in making DSS in different applications. Jin *et al*. [8] explored different data analytics engines and usage of SQL on the Hadoop kind of systems. They intended to explore the tools using GPGPU environments in future.

Gonzalez-Dominguez *et al*. [9] proposed an Automatic Speech Recognition (ASR) system with multi-language support. Such works can be integrated well with Elasticsearch and provide a dashboard and visualization for better access performance. Hammou *et al*. [10] on the other hand explored on real time data processing frameworks for faster text search and data analytics. Solomon [11] proposed methodology for data analytics and security in web applications. The Elasticsearch and other tools like Kibana and Logstash are integrated in order to have an affordable solution to web based integration of search and data analytics. Zamfir *et al*. [12] focused on systems monitoring and data management in a large scale including analytics. They employed Elastic search in order to have empirical study on the system monitoring practice. They intended to integrate the DevOps monitoring with the Elasticsearch. Wei *et al*. [13] focused on data gathering, data management and data visualization with Elasticsearch. They used scientific data for the research. Bove and Muller [14] studied the characteristics of Elasticsearch and employed it for monitoring public cloud systems and the data pertaining to different applications. They also focused on the security possibilities. In future, they intended to work on the log entries data using Elasticsearch. Berral-Garcia [15] studied big data storage and analytics with different tools like Kibana, Logstash and Elasticsearch. They found the usage of Elasticsearch to handle big data in different perspectives. Elasticsearch is available in [16]. From the literature, it is understood that there is need for evaluation of Elasticsearch for optimizing Twitter data in terms of search, indexing and data analytics.

III. ELASTICSEARCH AND ITS CAPABILITIES

As discussed in Section 1, Elasticsearch is a Java based distributed search engine that provides different functionalities. Essentially it is open source tool that can be integrated with real time applications for search and data analytics capabilities. It supports data aggregation with categories like bucketing, metric, matrix and pipeline and these aggregations can be effectively nested. The basic structure of aggregations can be captured as in Listing 1.

```
"aggregations" : {
  "<aggregation_name>" : {
    "<aggregation_type>" : {
      <aggregation_body>
    }
    [,"meta" : {  [<meta_data_body>] } ]?
    [,"aggregations" : { [<sub_aggregation>]+ } ]?
  }
  [,"<aggregation_name_2>" : { ... } ]*
}
```

Listing 1: Nesting of aggregations

Elasticsearch supports Domain Specific Language (DSL) queries which is based on JSON.

Retrieval Number: F1535089620/2020©BEIESP
DOI: 10.35940/ijeat.F1535.089620
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

The queries of two types namely leaf query clauses and compound query clauses. Elasticserch supports cross-cluster search. It is useful in the context of federated clouds and the clouds ability to provide multiple clusters. Listing 2 provides the means of remote cluster setup.

```
curl -X PUT "localhost:9200/_cluster/settings?pretty" -H 'Content-Type: application/json' -d'
{
 "persistent": {
  "cluster": {
   "remote": {
    "cluster_one": {
     "seeds": [
      "127.0.0.1:9300"
     ]
    },
    "cluster_two": {
     "seeds": [
      "127.0.0.1:9301"
     ]
    },
    "cluster_three": {
     "seeds": [
      "127.0.0.1:9302"
     ]
    } } } } }'
```

Listing 2: Remote cluster setup

There is support for scripting with which Elasticsearch is capable of evaluating custom expressions. It supports general purpose and special purpose languages for this. It has mapping feature that allows us to determine how a document needs to be stored and indexed so as to enable the Elasticsearch to manage it. It has text analysis feature that enables to build search engine, perform mining on the unstructured data, fine tune characteristics of search and also support linguistic and lexicographic search. There are different modules in Elasticsearch such as Gateway, HTTP, indices, network, plugins, thread pools, transport, remote clusters etc. There is index related modules that are used to control aspects pertaining to indices. Ingest node is used in order to pre-process documents prior to indexing them. Index Lifecycle Management (ILM) policies are used to manage indices and improve performance and resiliency. There is possibility of executing SQL queries against indices of Elasticsearch. Elasticsearch cluster monitoring provision is with the Elastic Stack. The older indices that are searched rarely are known as frozen indices. Data manipulation is done with Elasticsearch in two ways namely transforming data and rolling up data. Elasticsearch has its replica shards that ensure high availability of data. Clusters associated with Elastic search can be secured using security features of Elastic Stack. Different command line tools are made available with Elasticsearch. These tools are used to provide security and perform other important tasks. Elasticsearch can also be used as part of testing infrastructure. Elasticsearch provides required REST API for user interface components.

IV. ARCHITECTURAL OVERVIEW OF ELASTIC SEARCH

This section presents overview of Elasticsearch and where it fits into the Elasticsearch. It covers various architectural aspects of Elasticsearch and the usage of it with clusters and also in the Amazon EC2 cloud platform.

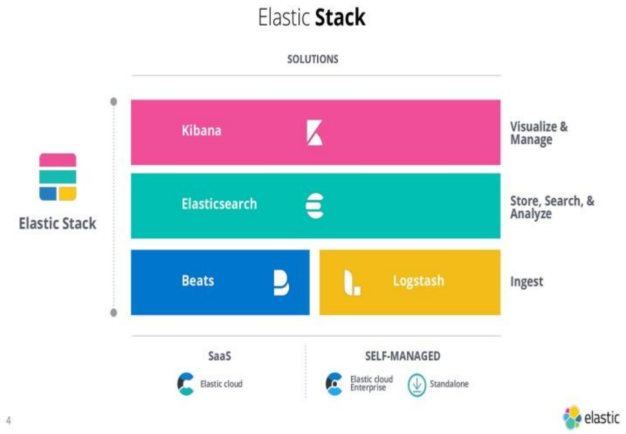

Fig 3: Architectural overview of Elasticsearch

As presented in Figure 3, the Elastic Stack contains Elasticsearch. Elastic Stack is made up of Beats, LogStash, Kibana along with Elasticsearch. For data collection, aggregation and enriching it Beats and Logstash are used and such data is stored in Elasticsearch. For interactive exploration and visualization Kibana is used. Elasticsearch helps in indexing of data, searching efficiently and performing data analytics. There are different layers involved. They include solutions, visualization, storing, search and analysis, ingest and deployment.

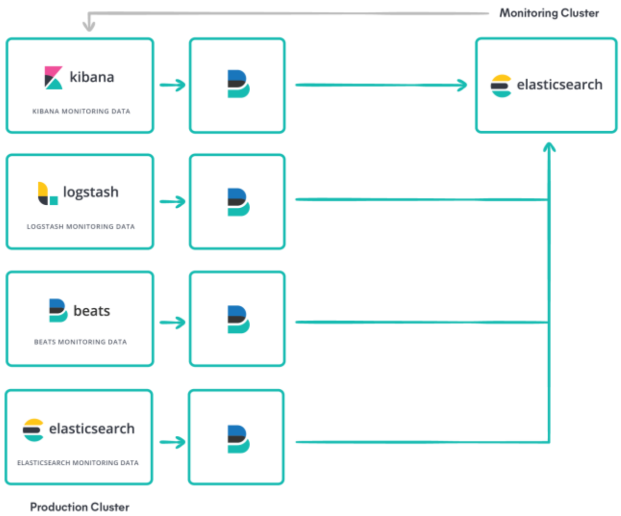

Fig 3: Illustrates separation of dashboards by groups

As presented in Figure 3, the Kiban configuration is used for visualization that is used by the groups of people. In fact, dashboards are provided for different groups. The results of Ellasticsearch are used to present in dashboards with visualization done by Kibana.

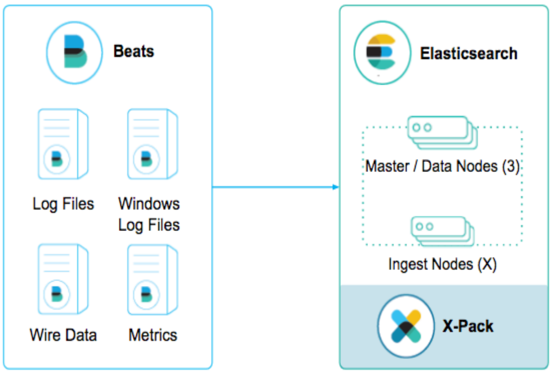

Fig 4: Shows inside details of an Elasticsearch cluster

As presented in Figure 4, the Elasticsearch supports clustering. Elasticsearch cluster has different components involved. It has minimum of 3 master nodes. It has many ingest nodes, many coordinating nodes, many data-hot nodes, many data-warm nodes, many alerting nodes and 2 or more nodes used for data analytics using ML techniques.

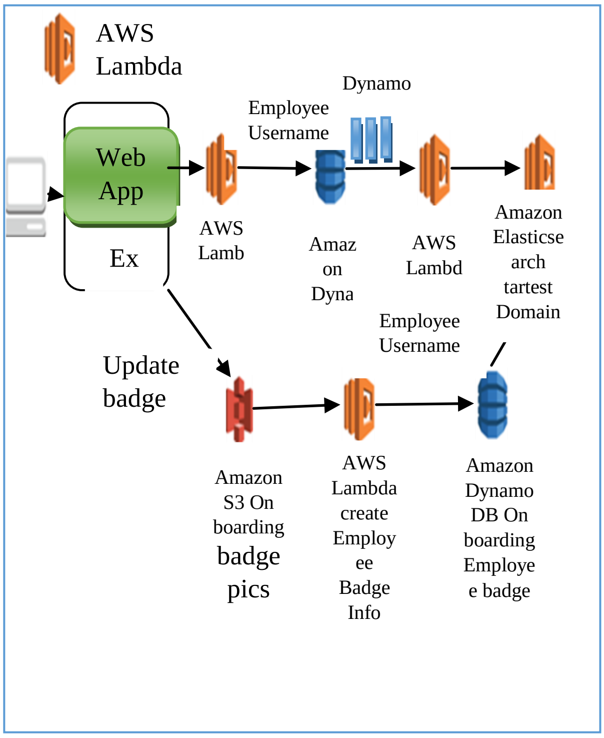

Fig 5: Architecture of Amazon Elasticsearch Service

As presented in Figure 5, Elasticsearch is configured in AWS cloud. When Amazon EC2 is used to configure Elasticsearch, it has benefits of Amazon cloud and its scalable and available infrastructure. It provides compete environment that helps in data acquisition, data indexing, data improvisation, data search, data visualization and data analytics. This configuration is used in this paper in order to capture and store Twitter data and then perform search and data analytics on the optimized data.

V. EXPERIMENTAL RESULTS

Experiments are made with Elasticsearch and the results are compared with the stat of the art distributed search engines. Tweets scrapped from Twitter and the Elasticsearch is used to analyse the same and the data is stored in inverted index for text search optimization. Search word pizza is used to evaluate search performance. The geographical distribution of the tweets associated with the word "pizza" is shown in Table 1.

Table I: Tweet distribution that resulted in search with Elasticsearch for word "pizza"

Unite d States	Turke y	Philippine s	United kingdo m	Brazi l	Othe r
47%	3%	3%	8%	9%	30%

As presented in Table 1, the resultant tweets of the search word "pizza" distributed across different countries such as Brazil, United Kingdom, Philippines, Turkey and United States besides other countries.

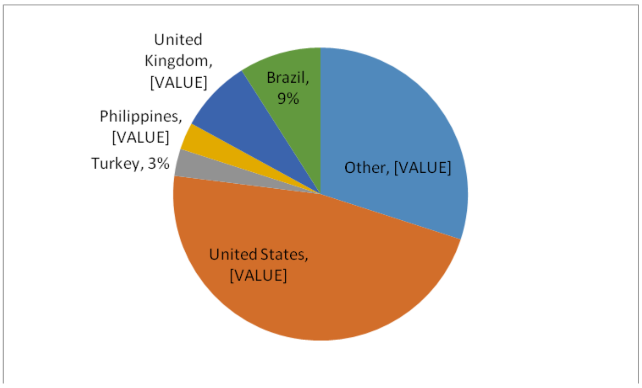

Fig 6: Shows resultant tweet distribution for "pizza"

As presented in Figure 6, the results of the search word "pizza" using Elasticsearch revealed that the tweets distribution is interestingly more with United States with 47%. Turkey showed 3%, Philippines 3%, UK 8% and Brazil 9%. The other countries showed 30%.

Table II: Language distribution of the result with search word "pizza"

French	Japanese	Portuguese	Spanish	English
3%	2%	6%	12%	77%

As presented in Table 2, the language distribution in terms of % is shown with different languages like English, Spanish, Portuguese, Japanese and French.

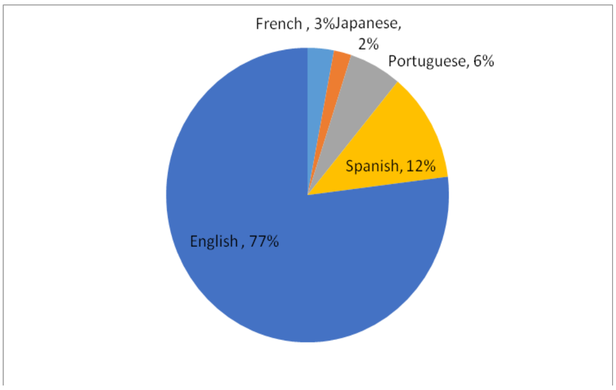

Fig 7: Shows resultant tweet distribution for "pizza" in terms of languages

As presented in Figure 7, the results of the search word "pizza" using Elasticsearch revealed that the tweets distribution is more in English language with 77%. Spanish showed 12%, Portuguese 6%, French 3% and Japanese 2%.

Table III : Source distribution of the result with search word "pizza"

Twitter Lite	Tweet Deck	Web Client	Android	iPhone	Other
3%	4%	11%	29%	38%	15%

As presented in Table 2, the source distribution in terms of % is shown with different sources like Twitter Lite, Tweet Deck, Web Client, Android, iPhone and other.

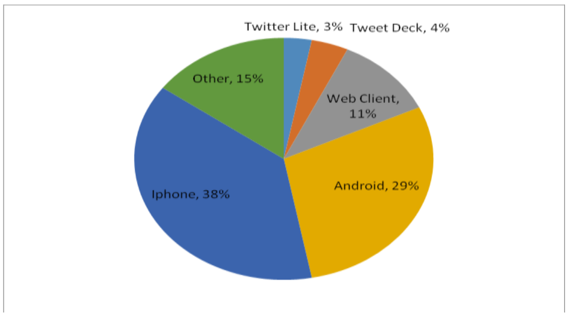

Fig 8: Shows resultant tweet distribution for "pizza" in terms of sources

As presented in Figure 8, the results of the search word "pizza" using Elasticsearch with respect to source revealed that the tweets distribution is more in iPhone with 38%. Web Client showed 11%, Tweet Deck 4%, Twitter Lite 3%, Android 29% and Other15%.

Table IV: Search time with different number of threads employed on small cluster

Number Of Searching Threads	Search Time (ms)			
	Solr Cloud S3R3	**Lucene-MongoDB R3S3**	**Solr Cloud S3 HDFS R3**	**Elasticsearch**
32	300	200	350	150
64	300	300	500	200
96	400	390	750	300
128	550	420	1000	350
160	620	600	1200	500
192	700	780	1350	600
224	750		1500	600
256	800		1600	750
288	830		1700	700
320	900		1800	800

Search time, as can be seen in Table 4 is recorded with different search engines and the results are recorded in terms of search time when different number of searching threads.

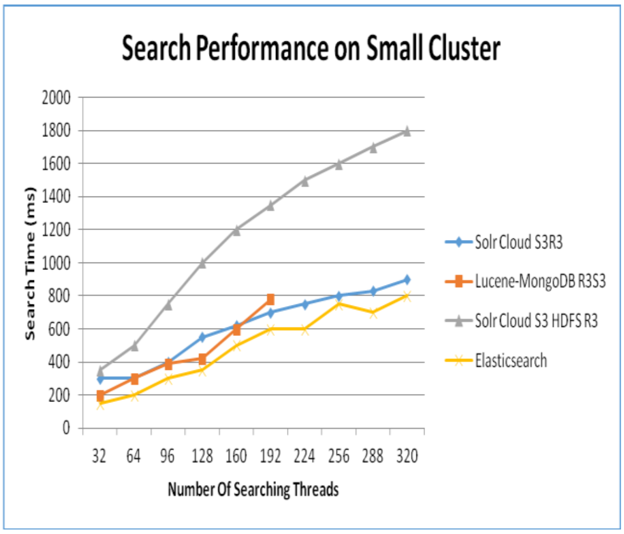

Fig 9: Search performance on small cluster

As presented in Figure 9, the number of searching threads is presented in horizontal axis. The number of searching threads is taken from 32 and increased gradually up to 320 threads. The vertical axis shows search time in milliseconds. Different search engines are compared along with the performance of Elasticsearch. The results revealed that the number of search threads has its impact on the search performance. Another important observation is that the Elasticsearch outperformed other methods. These results are when small cluster is used for experiments.

Table V: Search time with different number of threads employed on large cluster

Number Of Searching Threads	Search Time (ms)			
	Solr Cloud S7R3	**Lumongo S7R3**	**Solr Cloud S3**	**Elasticsearch**
32	200	200	250	150
64	300	300	400	200
96	350	450	600	300
128	450	500	800	400
160	550	620	950	400
192	600	800	1050	500
224	700	780	1150	600
256	750	900	1300	700
288	800	1000	1350	700
320	810		1400	750

Search time, as can be seen in Table 5 is recorded with different search engines and the results are recorded in terms of search time when different number of searching threads.

Retrieval Number: F1535089620/2020©BEIESP
DOI: 10.35940/ijeat.F1535.089620
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

The results are observed with the large cluster.

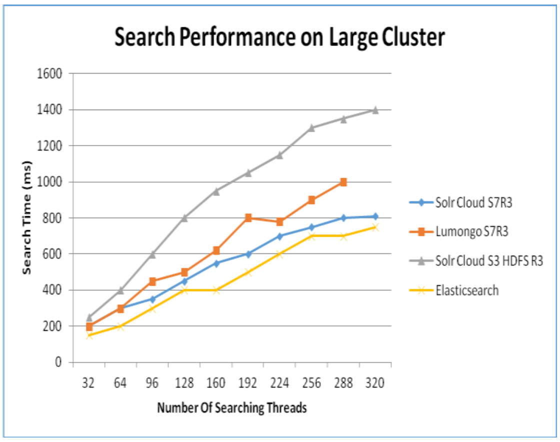

Fig 10: Search performance on large cluster

As presented in Figure 10, the number of searching threads is presented in horizontal axis. The number of searching threads is taken from 32 and increased gradually up to 320 threads. The vertical axis shows search time in milliseconds. Different search engines are compared along with the performance of Elasticsearch. The results revealed that the number of search threads has its impact on the search performance. Another important observation is that the Elasticsearch outperformed other methods. These results are when large cluster is used for experiments.

Table VI: Throughput with different number of threadsemployed on small cluster

Number Of Searchin g Threads	searches/second			
	Solr Clou d S3R3	Lucene-MongoD B R3S3	Solr Cloud S3 HDFS R3	Elasticsearc h
32	110	140	70	150
64	230	180	120	240
96	230	170	120	240
128	230	150	125	240
160	230	140	120	240
192	230	140	120	240
224	230		120	240
256	230		125	240
288	230		120	240
320	230		125	240

As can be seen in Table 6, the throughput is recorded with different search engines and the results are recorded in terms of throughput when different number of searching threads are employed. The results are observed with the small cluster. As presented in Figure 11, the number of searching threads is presented in horizontal axis. The number of searching threads is taken from 32 and increased gradually up to 320 threads. The vertical axis shows throughput. Different search engines are compared along with the performance of Elasticsearch.

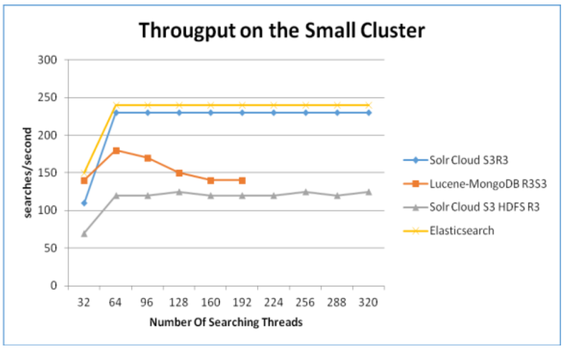

Fig 11: Throughput performance on small cluster

The results revealed that the number of search threads has its impact on the throughput performance. Another important observation is that the Elasticsearch outperformed other methods. These results are when small cluster is used for experiments.

Table VII: Throughput with different number of threads employed on large cluster

Number Of Searching Threads	searches/second			
	Solr cloud S7R3	Lumongo S7R3	Solr Cloud S7	Elasticsearch
32	130	150	120	160
64	180	170	140	190
96	230	180	140	240
128	240	190	140	250
160	250	210	150	260
192	240	180	140	260
224	230	170	140	240
256	250	160	150	260
288	230	160	140	240
320	250		150	260

As can be seen in Table 7, the throughput is recorded with different search engines and the results are recorded in terms of throughput when different number of searching threads are employed. The results are observed with the large cluster.

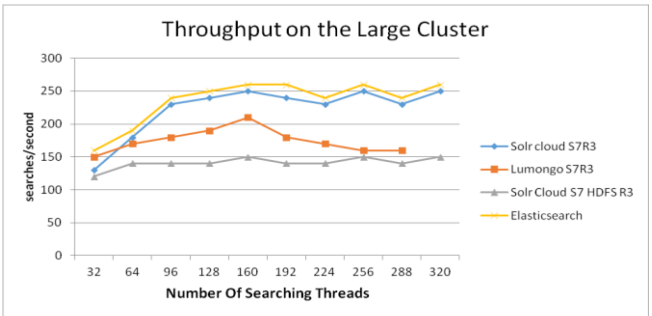

Fig 12: Throughput performance on large cluster

Retrieval Number: F1535089620/2020©BEIESP
DOI: 10.35940/ijeat.F1535.089620
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

As presented in Figure 12, the number of searching threads is presented in horizontal axis. The number of searching threads is taken from 32 and increased gradually up to 320 threads. The vertical axis shows throughput. Different search engines are compared along with the performance of Elasticsearch. The results revealed that the number of search threads has its impact on the throughput performance. Another important observation is that the Elasticsearch outperformed other methods. These results are when large cluster is used for experiments.

VI. CONCLUSION AND FUTURE WORK

In this paper, Elasticsearch is used for investigating its search and analysis capabilities. Elasticsearch is configured in locally Windows OS and also in Amazon EC2 cloud. Since Elasticsearch can support data storage with cloud platforms, search in distributed environment and data analysis, it is investigated and compared with other search engines. Elasticsearch is Java based tool that can be integrated with real time applications for its intended operations. It can be used to search and analyse data for garnering BI. With the cloud computing technology and the scalable infrastructure associated with the cloud, Elasticsearch usage is dramatically increased. Since it can be used to have clusters in cloud to gather data, store it, index it and perform analytics with near real time, Elasticsearch assumes high significance. For data collection, aggregation and enriching it Beats and Logstash are used and such data is stored in Elasticsearch. For interactive exploration and visualization Kibana is used. Elasticsearch helps in indexing of data, searching efficiently and performing data analytics. In this paper, the utility of Elasticsearch is evaluated for optimising search and data analytics of Twitter data. Empirical study is made with the Elasticsearch tool configured for Windows and also using Amazon Elasticsearch and the results are compared with the state of the art. The experimental results revealed that the Elasticsearch performs better than the existing ones. In future, we intend to integrate Elasticsearch with different kinds of applications demonstrate its utility and usefulness in the contemporary era to real enterprises.

REFERENCES

1. Langi, P. P. I., Widyawan, Najib, W., & Aji, T. B. (2015). *An evaluation of Twitter River and Logstash performances as elasticsearch inputs for social media analysis of Twitter. 2015 International Conference on Information & Communication Technology and Systems (ICTS) p1-6*
2. Malika Bendechache. (2019). Analysing dependability and performance of a real-world Elastic Search application, p1-8.
3. Betke, E., & Kunkel, J. (2017). *Real-Time I/O-Monitoring of HPC Applications with SIOX, Elasticsearch, Grafana and FUSE. High Performance Computing, 174–186.*
4. Khaled Nagi (2015). Bringing Search Engines to the Cloud using Open Source Components. *In Proceedings of the 7th International Joint*
5. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).* P1-5.
6. Hanamanthrao, R., & Thejaswini, S. (2017). *Real-time clickstream data analytics and visualization. 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT).* P1-6.
7. Shah, N., Willick, D., & Mago, V. (2018). *A framework for social media data analytics using Elasticsearch and Kibana. Wireless Networks.* P1-9.
8. Jin ZH, Shi H, and Hu YX *et al*. CirroData: Yet another SQL-on-Hadoop data analytics engine with high performance. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(1): 194–208.
9. Javier Gonzalez-Domingue. (2015). A Real-Time End-to-End Multilingual Speech Recognition Architecture. *IEEE*. 9 (4), p1-11.
10. Ait Hammou, B., Ait Lahcen, A., & Mouline, S. (2020). *Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics. Information Processing & Management, 57(1),* p1-15.
11. Inigo Solomon, F. (2019). *Securing Websites & Webapplications Using Data Analytics. 2019 International Conference on Computational Intelligence in Data Science (ICCIDS).* P1-4.
12. Zamfir, V.-A., Carabas, M., Carabas, C., & Tapus, N. (2019). *Systems Monitoring and Big Data Analysis Using the Elasticsearch System. 2019 22nd International Conference on Control Systems and Computer Science (CSCS).* P1-6.
13. Wei, H., Wu, S., Zhao, Y., Deng, Z., Ersotelos, N., Parvinzamir, F., Dong, F. (2016). *Data Mining, Management and Visualization in Large Scientific Corpuses. Lecture Notes in Computer Science, 371–379*
14. Bove, D., & Muller, T. (2019). *Investigating Characteristics of Attacks on Public Cloud Systems. 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom).* P1-6.
15. Berral-Garcia, J. L. (2016). *A quick view on current techniques and machine learning algorithms for big data analytics. 2016 18th International Conference on Transparent Optical Networks (ICTON) p 1-4*
16. Elasticsearch (2020). Download Elastic Search. Available at https://www.elastic.co/downloads/ elasticsearch.
17. *Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Managemen*, p116-126.
18. Thacker, U., Pandey, M., & Rautaray, S. S. (2016). *Performance of*
19. *elasticsearch in cloud environment with nGram and non-nGram indexing.*

ATHOURS PROFILE

Mr. Subhani shaik is a par time research scholar in the department of Computer Science and Engineering at Acharya Nagarjuna University, and he has 15 years of teaching experience in the academics.

Dr. Nallamothu Naga Malleswara Rao is Working as a Professor in the Department of Information Technology at RVR&JC College of Engineering, Chowdavaram, Guntur,A.P. He has 28 Years of Experience in the field of Computer Science.