Two-Way Refinement Approach For Extra Corrupted Shard Removal In Elastic Search With Lucene And Translog

Subhani Shaik[a], Dr. Nallamothu Naga Malleswara Rao[b]
[a] Research Scholar, CSE Dept, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.
[b]Professor, IT Dept, RVR & JC College of Engineering, Guntur, Andhra Pradesh, India.
Email: [a]subhanishaik2613@gmail.com, [b]nnmr3654@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Elasticsearch is most popular search engine which is based on Apache Lucene. Many advantages are identified with the Elasticsearch. For every inserted document or record the Elasticsearch's auto-generated id values are created. But this may leads to increasing the duplicate values. To overcome this various duplicate methods are introduced by various researchers. Indexing is very important for the elastic search removing duplicates in elastic is based on indexing. For this Lucene Index and Translog are used. This can be used for all types of data in Elastic search. Many researchers working on removing duplicates and shards from the data. But still there is lot of corrupted shards is present in output. To overcome this, A Two way Refining Algorithm (TWRA) is introduced to remove the extra corrupted shards for extra refinement of data. The TWRA consists of two refinements of data such as Advanced Advanced Data Cleaning and Advanced Data Filtering Algorithm. Experimental results show the performance of the proposed methodology.

Keywords: Elastic search, apache Lucene, Translog, TWRA, indexing

1. Introduction

ElasticSearch is an open source, RESTful web crawler based over Apache Lucene and discharged under the Apache permit. It is Java-based, and can look and record report documents in various organizations. Elasticsearch underpins best replication over different datacenters with chop down inactivity. Particular affiliations like Netflix, Linkedin, Accenture depends exceedingly on Elasticsearch for securing, asking for and looking the information because of its segments like mind blowing advancement show up, modules, modified sharding and replication. Netflix alone has sent more than 15 groups including 800 focus focuses [3].Shard affirmation is a change for appropriated web crawlers. The pondering is that a demand ought to just be set up by focus guides which are likely going toward return enormous outcomes. Shard choice is a particularly investigated issue to which diverse blueprints have climbed all through late years. This part is worried over giving a comprehension of why this issue exists and why it is basic to research.

1.1.Shard:

As a distributed search server, the concept of elastic search shard is used to distribute index documents across all nodes. The index stores large amounts of information that can surpass the hardware limit for a single node. For example, an index of one of a billion files that take up 1 TB of space in the disk may not fit on a single node disk, or it may be very less speed to ask questions from just one node.

To provide solution to this problem, elastic search furnishes the capacity to split the index into several parts called segments. When creating an index, you can specify the required number of parts. Documents are placed on shards, and shards are permits on nodes in your cluster. As the cluster grows or contracts, the flexible search automatically moves shards between the nodes to keep the cluster balanced. A shard can be either main or duplicate. Every document in your index has a place with a similar essential shard, so the quantity of essential shards you have decides the most extreme measure of information your file can contain. A repeating shard is just a copy of the original shard.

1.2.Replica:

To avoid data loss in case of hardware failure, Replica Shard is a copy of Primary Shard. Flexible search allows you to call duplicate shards or dedicated copies of your index shards. An index can be duplicated from zero (which has no duplicate) or several times. The number of shards and replicas of each index can be determined at the time the index is made. After the index is generated, you can dynamically change the number of copies whenever, however, you can't change the number of shards after the incident is finished. As a matter of course, every index is in Elasticsearch is allocated 5 main Shards and 1 copy which implies that in the event that you have at least two nodes in your cluster, your index will have 5 main shards and another 5 imitation shards (1 complete replica) for an aggregate of 10 shards for each index.

1.3.Corrupted Shards:

Elastic search is the most ideal approach to run different versatility tests. We can scale long-running tasks (which will require days to complete) to finish within hours which needs to deal with near 13 million reports.

During this case, when we are expanding the number of nodes in statics search, there is a chance of shutting down one of the nodes. Far and away more terrible we ended that shard. As a request one of the nodes took with it an enter shard. For a model if we have two c5.2xlarge nodes serving 13 threads each taking 5000 questions per minute sooner it was taking just 100 every moment which is each low speed.

However we needed to build the no of inquiries every moment further thus, we added, two additional nodes to the cluster now, the elastic search will begin moving one of the 8 shards to this third and fourth node.

Amidst this whole cycle, there was a side dir measure pulverizing these studies with different such inquiries through 21 strings. The pure size it had to exhaust was only one million. So before the elastic search could finish moving the shard to the new node the whole cycle was finished to make the job quicker. Now these two additional nodes were pointless and those nodes can be ended. Yet, one of those two nodes took a shard alongside those. That implies there were just 7 shards with 3 million reports and the eighth shard having near 400k records is missing. The unused shards were currently in a corrupt state. If we want to bring it back to life and run a re-index on the missing reports.

1.4.Elastic Search Shard:

Now and again, the Lucene index or translog of a shard copy can get corrupted. The Elastic search shard inquires to remove corrupted parts of the shard if a good copy of the shard cannot be recovered automatically or restored from back-up.

At the point when Elastic search identifies that a shard's information is ruined it fairs that the shard copy and will not utilize it. All in all, that is naturally recovered from another copy. If there is no good copy available and the shard isn't accessible and if there is no chance of recovering or restoring we can utilize Elastic search shard to eliminate the undermined data and reestablish admittance to any reassigning data in the unaffected segment.

- To eliminate corrupted shard data utilize the use of the sub covered remove-corrupted-data.
- Here, we can indicate the path in two ways :

❖ Specify the index value and shard value with – index and – shard_id
❖ Use the – dir alternative to indicate the full way to the corrupted index or translog files.

1.5.Translog:

Lucene commits are too costly to even consider performing on every single individual change, so every shard copy likewise composes tasks into its transaction log known as the" translog". All index add delete tasks are composed to the translog in the wake of being prepared by the inside Lucene index met before they are recognized. If there is a crash, late tasks that have been recognized however excluded from the last Lucene commit are rather recovered from the translog when the shard recovers. Flushing an information stream or file is the way toward ensuring any information that is presently stored in the' translog' is likewise permanently stored in the 'Lucene file'. Elastic search naturally triggers flushes varying utilizing heuristics that compromise the size of the unflushed transaction log against the expense of performing each flush.

When restarting, elastic search replays any unflushed tasks from the translog into the Lucene record to bring once more into the express that it was before the restart.

Once every activity has been flushed it is permanently stored in Lucene index. That implies there is no need of keeping an extra copy of it in translog. Translog is comprised of different documents, called generations and the elastic search will erase any generation records once they are not required, freeing up the disk space. Elastics search flush is the way toward performing out a Lucene comprise and beginning another translog generation. The information in the translog is possibly endured to disk when the translog is fsynced and committed. If an equipment disappointment or working framework crash or a JUM crash or a shard failure any information composed since the past translog comprise will be lost. Index.translog.durability is set to 'request' as a matter of course implying that elastic search will only repeat the success of an index, delete, update or break request to the customer after the translog has been effectively fsynced and committed on the essential and on each allocated replica. If record.Translog.durability is set to 'async' at that point elastic search fsyncs and comprises the translog just every index.translog.sync-interval; which implies that any tasks that were performed not long before a crash might be lost when the node recovers.

2.Literature Survey

When data is linked to multiple sources, it contains a large portion of the dirty data. Report this dirty data to record price errors, record duplication, incorrect spelling, movable values, disobedience reference integrity and inconsistency in records. This statistic can be used to verify dirty data. Therefore, it is important to clear the data [2].

Data quality management is a problem for organizations because they have the power to manipulate decisions [9]. Therefore, data quality is monitored as a barrier problem in businesses and industries [10]. Operational databases and online analytics processing systems cannot avoid data quality problems by consolidating data. These problems are caused by a non-standard set of standards in a distributed database. Data cleanup plays an important role in providing quality data by detecting and eliminating discrepancies in data [11].

3. Proposed Methodology

3.1 Advanced Data Cleaning

Advanced Data Cleaning (ADC) is most important task to clear the unwanted, irrelevant data in the dataset. ADC can also performs managing missing data, managing structural errors, eliminating unwanted experiments etc. Especially in Elastcisearch, the Advanced Data Cleaning plays the major role to overcome the performance issues with the unprocessed data. ADC can used to develop a better model to handle the unstructured and missing data. This paper aims to spent lot of efforts to remove the unwanted and corrupted shards in the dataset after preprocessing. The ADC in this paper includes three stage cleaning such as Eliminating unwanted data scrutiny, managing unwanted outliers and the unhealthy shards got removed. Unhealthy shards removed in three steps.

- Step 1: Scan Health of a cluster in elastic Search
- Step 2: Sort all the unhealthy shards denoted in red
- Step 3: Delete all the unhealthy shards

Eliminating unwanted data scrutiny consists of removing duplicates, redundant or irrelevant shards from the selected dataset. These types of data will fit for the particular problem that we are trying to solve. Redundant data also effects the results, because if the data repeated these many times then this may cause unfaithful results. With certain types of models the outliers becomes more problem. Without any genuine reason one can't remove the outliers. Sometimes removing of outliers can effects the performance or may not.

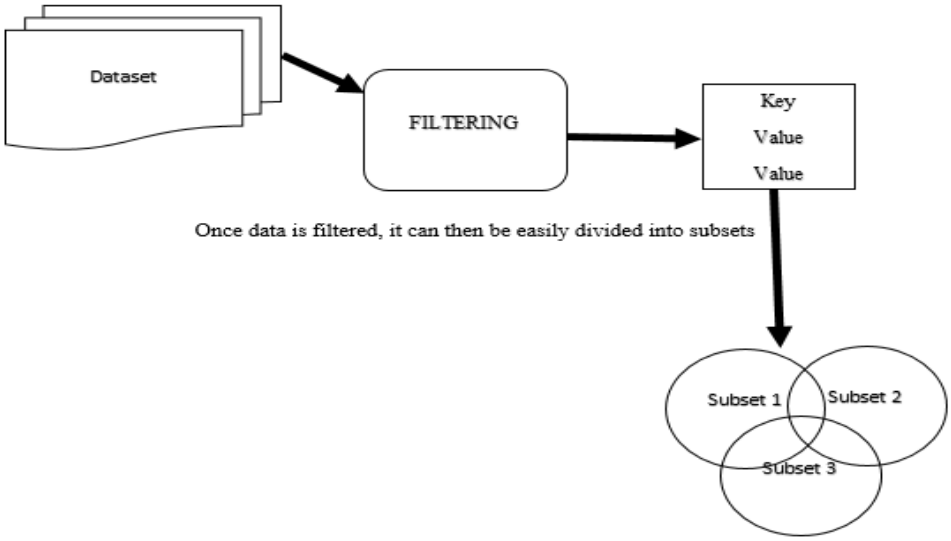
3.2 A Two way Refining Algorithm (TWRA)

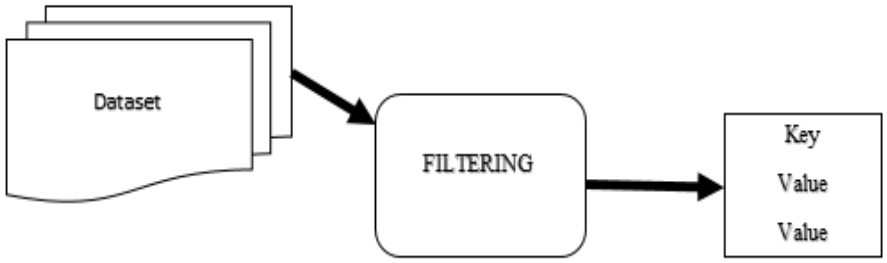
The proposed system TWRA is two ways refining of data and this consists of two stages. In the first stage, an advanced Data Cleaning algorithm is proposed and in the second stage an improved Advanced Data Filtering algorithm is proposed to filter the shards in the output data.

Based on the architecture, the client request the query and the query is checked in data base. The elastic search comes to the picture and the file is created an indexing documents. And the documents are shared to shards that are created in node. Every shared is a Lucene index and the Lucene collects all the updates on documents of adding the document, deleting the document and manages the indexes. There the operations get committed. Due to cost effective, committing the transactions or operations every time is not possible. So to overcome the issue here we use translog. There the translog will stores the commit and uncommitted transactions that happened in Lucene. The Lucene indexes will shares the data to segments and again segments get merged and allows to store in the disk. After committing the data there will be duplicate and invalid shards. To overcome this a "Quality Assorted Algorithm [12] is considered and identified the duplicate and invalid shards. After identifying the semi and fully duplicated shards gets removed by using a technique called EDDR Algorithm [13]. This algorithm removes the duplicated shards that are identified using Quality assorted Algorithm. But due to the invalid and duplicate shards we had a possibility to occurrence of extra corrupted shards in the elastic search cluster. To overcome from this problem we had proposed a solution Two Way Refining Algorithm, where advanced Data Cleaning is applied on the dataand in the next stage data filtration algorithm is applied to remove corrupted shards,where the extra corrupted shards willget removed and finally we get an elastic search cluster with only good shards.

3.3 Advanced Data Filtering

Data filtering is the process of selecting a small portion of your dataset and using that substrate for display or analysis. Filtering is usually (but not always) temporary - the entire data set is secure, but only part of it is used for calculations. Exclude false or "bad" notes from your analysis. This is done to make it easier to focus on specific information in a larger dataset or spreadsheet. Filtering does not remove or change data, it only changes the rows or columns that appear in the active Excel worksheet.





As data is filtered, key and values are determined

In Proposed Advanced Data Filtering we process the data for data quality and also for data analysis. Collaborating filtering algorithms have been improved for recommendation systems, also, applied in some real applications, for example, Elasticsearch, YouTube, Amazon and Netflix. How do you analyze such a large collection of data is important. Matrix and tensor factorization with or without potential treatment, it is possible to implement effective algorithms to deal with this problem. In general, the observed matrix has a rating value for M users and N items (movies, videos, or products), which is definitely non-negative. Some entries in X may be missing. A sparse matrix can be collected. Collaborative filtering aims to analyze the user's preferences and past history of ratings and use the analyzed information to predict the user's future rankings on a particular item. In Salakhutdinov and Mnih (2008) [11], the probabilistic matrix factorization (PMF) was proposed to approximate an observed rating matrix $X \in R^{MxN}$ using a product of a user matrix $B \in R^{MxK}$ and an item matrix$W \in R^{KxN}$. This is differentiated with the standard source separation method by using NMF where the observed matrix $X = \{X_{mn}\}$ is collected as the log-magnitude spectrograms over different frequency bins m and time frames n. The nonnegative matrix X is factorized as a product of basis matrix B and weight matrix W. Let the matrices B and W be represented by their corresponding row and column vectors, i.e.

$B = [B_1^T \ldots . B_{M:}^T]^T$----[1]

$$W = [W_{:1} \ldots W_{:N}] - -[2]$$

PMF is constructed according to probabilistic assumptions. The propability function of generating the rating matrix is assumed to follow a Gaussian distribution

$$p(X|B, W, \sigma^2)$$

$= \prod_{m,n}[N(X_{mn}|B_{m:}W_{:n}, \sigma^2]^{I_{mn}}$ -- [3]

$$= \prod_{m,n}[N(X_{mn}|\sum_k B_{mk}\, W_{kn}, \sigma^2]^{I_{mn}} - -[4]$$

Where B_m and $W_{:n}$ denote the K-dimensional user-specific and item-specific latent feature vectors, respectively, σ^2 is a shared variance parameter for all entries in X and *lmn* denotes the indicator, which is one when X_{mn} is observed and zero when X_{mn} is missing. The prior densities of PMF parameters are assumed as zero-mean Gaussians with the shared precision parameters α_b and α_w for all entries in matrices B and W given by.

$$p(B/\sigma_b^2 = \prod_{m,k} N(B_{mk}|0,\sigma_b^2), --[5]$$

$p(W/\sigma_w^2 = \prod_{n,k} N(W_{nk}|0,\sigma_w^2)$,-- [6]

The maximum *a posteriori* (MAP) estimates of **B** and **W** are evaluated by maximizing the logarithm of the subsequent distribution over the user and item matrices given by the fixed variances {σ2, σb2, σw2}. Accordingly, maximizing the subsequent distribution logarithm is tantamount to minimizing the following target function:

$$\frac{1}{2}\sum_{m,n} I_{mn}\,(X_{mn} - \sum_{k} B_{mk}W_{kn})^2 + \frac{\lambda_b}{2}\sum_{m,k} B_{mk}^2 + \frac{\lambda_w}{2}\sum_{m,k} W_{nk}^2 --[7]$$

Since regulation parameters are used using different contrast parameters,

$$\lambda_B = \frac{\sigma^2}{\sigma_B^2}\ and\ \lambda_w = \frac{\sigma^2}{\sigma_w^2}$$

In practice, select the appropriate hyper parameters {σ2, σb2, σw2} or regularization parameters {λb, λw} is crucial for model regularization.

3.4 Twra Processing Steps

In the Two Way Refining process there are seven steps to make a data set into corrupt shards free data which makes the searching more efficient and with less processing time. The seven steps are explained below. Step one will collect and initialize the data set and removes the corrupted shard and finally gives the output file.

Step 1: Initialize dataset

Step 2: Pre-processing

Step 3: Advanced Data Cleaning

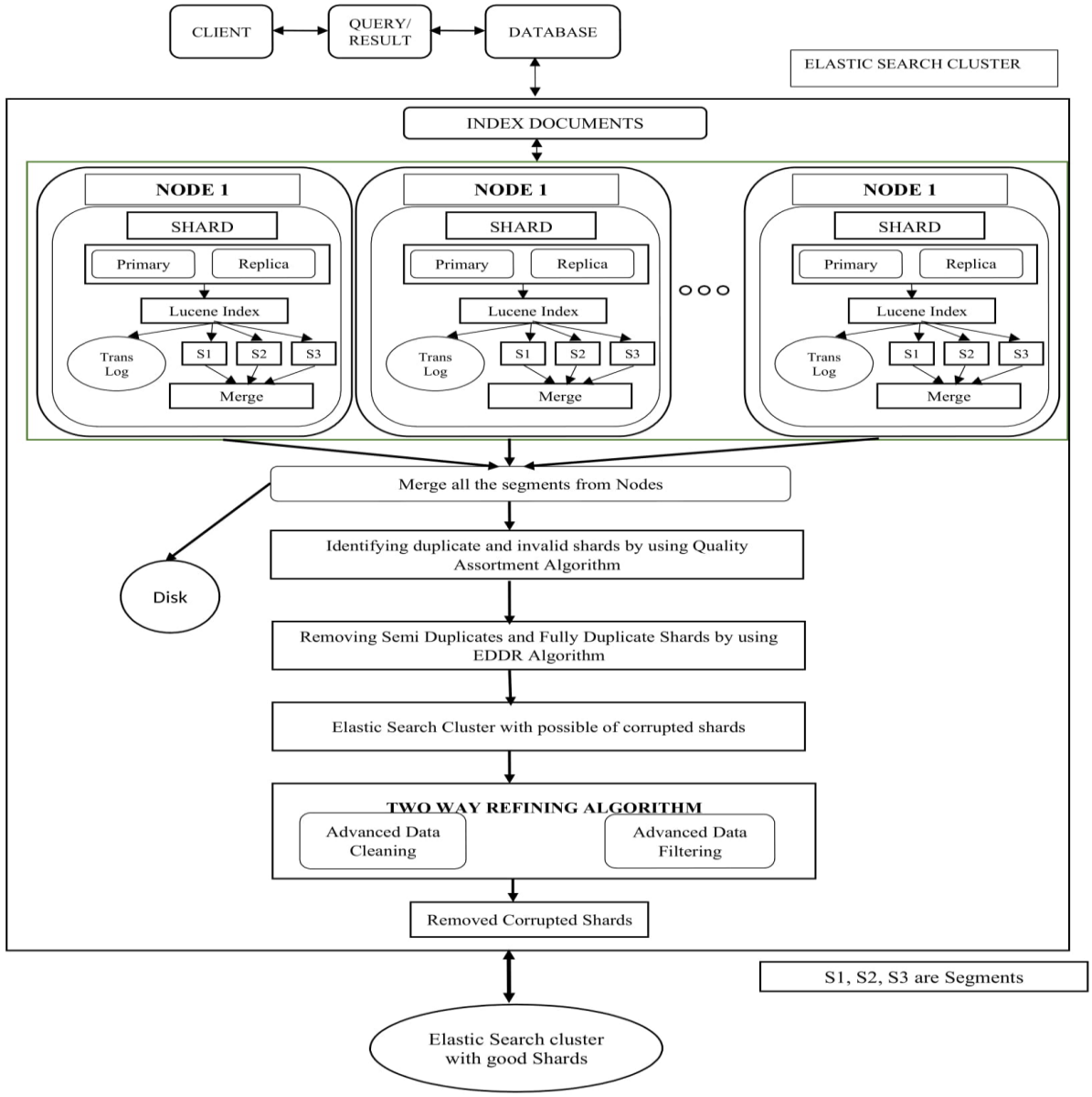
Step 4: Feature Extraction from equation 3, 4

Step 5: Advanced Data Filtering

Step 6: Removing Corrupted Shards

Step 7: Results

TWRA Algorithm:



Input: Dataset Having Extra Corrupted Shards

Output: Removal Of Extra Corrupted Shards

1.
tart INITIALIZE
2.
elect the Dataset to perform the elastic search( )
3.
nitialize or load the Dataset
4.
re-Processing of dataset -
Split the Dataset into ‘n’ Shards through index document.
5.
n Shards we have **“Primary Data Block”** and **“Replica Data Block”.**

6.
pply **"Lucene Index"** (Lucene Search) on Primary data.
To reducing the commit cost in Lucene index we are using **"Translog"**
7.
tart of **Two-Way Refining Algorithm**
(i) Advanced Data Cleaning (ADC)
The ADC includes **Eliminatingunwanted data scrutiny** such as removing duplicates, redundant or irrelevant shards from the selected dataset and **managing unwanted outliers**.
(ii) Advanced Data Filtering (ADF)
ADF The process of selecting a small portion of the dataset and using that substrate for presentation or analysis **(Exclude erroneous or "bad" observations).** It retrieves selected rows or columns to appear in the active worksheet for analysis.
8.
emoving Corrupted Shards:
If there is a Single Cluster

```
Start
c= read Number of clusters
if(c==1 && Corruption==True)
{
Elastic Search is useless
goto initial state
}
else
continue search
end.
```

If there are multiple nodes in a cluster

```
start
c=read the cluster
sn=Random(node from a single cluster)
if(sn==datanode&& Corruption==True)
{
execute the cluster
To store the outcome-rearrange the node
goto initial state
}
else
{
if(sn==MAX &&datanode==false)
remove corrupted node
else
continue search
}
end.
```

9.

ND

3.5 Removing Corrupted Shards:

When you utilize Elastic search shard to eliminate the corrupt data, The Shard's assignment ID changes. After restarting the node, you should utilize the cluster reroutes API to feel Elastic search to utilize the new ID. We can utilize the –truncate-clean-translog to shorten the Shard's translog even if does not appear to be corrupt. What happens when Elastic inquiry node/file/shard gets corrupt: When there is a single cluster, and if that there is corruption, at that point the whole Elastic search arrangement is useless. We need to set up again from the earliest starting point.

If there are multiple nodes in the cluster:

If we configure a single node as a data node and if that node is corrupted, the cluster will be running however queries would not restore any outcome. In such cases, we need to re-arrange the node as a data node and restart the cluster. On the off chance that there are various data nodes, at that point, a corruption/failure of a node will be removed simply after that node. The remaining nodes and Elastic search will accomplish their work of course. Yet, the only issue here is that the data stored in the corrupted node won't be accessible. The shards in the corrupted node will become unassigned shards and must be reassigned to some other data node.

```
PUT{
 "commands": [
  {
   "allocate_empty_primary": {
    "index": "index_name_masked",
    "shard": 0,
    "node": "*.*.*.*",
    "accept_data_loss": true
   }
  }
 ]
}
```

If there are "copies" empowered, at that point, there will be no impacts as far as information misfortune. It would require the unassigned shards to be needed to some new information node. It is ideal to have a multi-node cluster with at least 2 data nodes and copies empowered to mitigate shards/data nodes corruption. Even however Elastic search has more prominent security, there is as yet a chance of getting a cluster into a "red" state because of a corrupted index. An index gets corrupt because of a sudden loss of intensity, equipment failure, or running out of disk space.

In this paper we are talking about how to carry the cluster to a good state with insignificant or no information loss. If there is some kind of problem with Elastic search, the primary activity is to check cluster health. It is extremely simple to distinguish which indices are at fault because those will have the health status as “red”. One-way recovery is simply to eliminate the folder with Elastic search data and start from the beginning yet even the data loss may be accepted, it isn't at all a decent arrangement as there are a couple of indices that are to fault. So there should be an approach to recover with loss/no damage.

```
bin/elasticsearch-shard remove-corrupted-data
 ([--index <Index>] [--shard-id <ShardId>] | [--dir <IndexPath>])
 [--truncate-clean-translog]
 [-E <KeyValuePair>]
 [-h, --help] ([-s, --silent] | [-v, --verbose])
```

- Elasticsearch-shard with subcommand eliminate corrupted shards
- The primary objective is to fix the corrupted shards - the activity is dangerous - in this way no any fix or restore, keep away from truncate considered a long way from Lucene.
- Available alternatives for eliminate corrupted shards:

- -- indexfile name index_name and - shard-id shard_id (required)
- ❖ alternative:-d path _ to _ index _ folder or – dirpath _ to _ index _ folder
- - dry-run do quick check without the real dropping of corrupted shards
- no alternatives implies exercise - connected keyboard authentication is required
- integrate elasticsearch-translog into elasticsearch-shard
- elasticsearch-translog becomes elasticsearch-shard truncate translog
- elasticsearch-translog has just - d alternative to determine folder - it is ideal to have - indexfile name index_name and - shard-id shard_id
- Exit quickly if there is no defilement marker document
- for the two cases
- Literally missed shards are the unrecoverable case with check Index.

Proposed pseudo code for corrupted shard removal:

INPUT: {ESfile, br, rm} where ES is Elastic Search file, br is Buffered Reader, rm variable is used to read .recovery files

OUTPUT: Removal of corrupted Shard

Algorithm CSR(Corrupted Shard Removal)

1. StringBuilderreadESfile = new StringBuilder();
2. try
3. (BufferedReaderbr=new BufferedReader(new InputStreamReader(inputStream)))
4. if (br == NULL)
5. File Not Found Error
6. else
7. rm=read code_shards_translog_*.recover
8. while(readESfile)
9. if (rm=filename.recover)
10. remove file
11. corrupted shards cleaning completed
12. End if
13. End while
14. End if

Dataset Description

Bank Marketing Data Set: The figures relate to the Portuguese banking company's direct marketing campaign. The campaign is run by phone calls. Often, more than one contact with the same customer is required to access the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Advanced Data Cleaning can be done for the dataset after pre-processing:

•Change the columns with 'yes' and 'no' values to Boolean columns;

•Change the categorical columns into dummy variables.

4.Performance Evolution

Various performance measures are given below for the calculation of False Positive Rate (FPR), False Negative Rate (FNR), time and Accuracy.

FPR

Based on the given data the information is divided into normal and abnormal.

$$FPR = \frac{FP}{FP + TN}$$

FNR

The percentage of cases where the data was divided as abnormal, but in reality it was

$$FNR = \frac{FN}{FN + TN}$$

Accuracy: This will find out the overall accuracy of the data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

To calculate the accuracy based on the prevalence

$$Accuracy = \big((Sensitivity) * (Prevalence)\big) + ((Specificity) * (1 - Prevalence))$$

Sensitivity: $TP / (TP + FN)$

Specificity:$TN/ (FP + TN)$

Prevalence: The amount of queries

The **Data Processing Time** calculator computes the amount of time needed to process an amount of data (**S**) at a specified rate (**R**).

- (S) This is the size of the file or data object being processed.
- (R) This is the processing rate

$$A = \frac{S}{R}$$

Data is taken to compare the existing system IES with the proposed system TWRA.

Sno	Total No Of Employee's	Record Assigned To Every Employee	Result Show Based On Query	Time Taken	Accuracy Using IES
1	E-1	9876	4532	0.010	98.8
2	E-2	8976	3213	0.0054	98.8
3	E-3	9807	3421	0.0045	98.8
4	E-4	7890	4121	0.0034	98.8
5	E-5	8765	3211	0.0045	98.8
6	E-6	9098	2134	0.0056	98.8
7	E-7	8976	2321	0.0065	98.8
8	E-8	6785	2431	0.006	98.8
9	E-9	5674	1232	0.003	98.8
10	E-10	4140	876	0.004	98.8
Results	10 Employee's	79987	27492	0.054	98.8

Table: 1 Performance of Existing System IES

Same data is considered to identify the performance of TWRA after removing the corrupted shards

Sno	Total No Of Employee's	Record Assigned To Every Employee	Result Show Based On Query	Time Taken	Accuracy Using TWRA
1	E-1	9876	4725	0.087	99.11
2	E-2	8976	3512	0.0044	99.12
3	E-3	9807	3741	0.0040	99.12
4	E-4	7890	4368	0.0022	99.10
5	E-5	8765	3525	0.0023	99.11
6	E-6	9098	2523	0.0046	99.13
7	E-7	8976	2749	0.0045	99.12
8	E-8	6785	2931	0.005	99.11

9	E-9	5674	1652	0.003	99.14
10	E-10	4140	1205	0.003	99.14
Results	10 Employee's	79987	30931	0.044	99.12

Table2: performance of proposed system TWRA

After performing TWRA on the dataset thesensitivity, specificity, accuracy is increased and the execution time is decreased.

Algorithms	Sensitivity	Specificity	Accuracy	Time(sec)
Improved Elastic Search (IES) using Map Reduce	96.7	97.1	98.8	23.1
A Two Way refinement Algorithm (TWRA)	98.2	98.4	99.12	18.12

Table 3: The overall performance of existing and proposed methodology

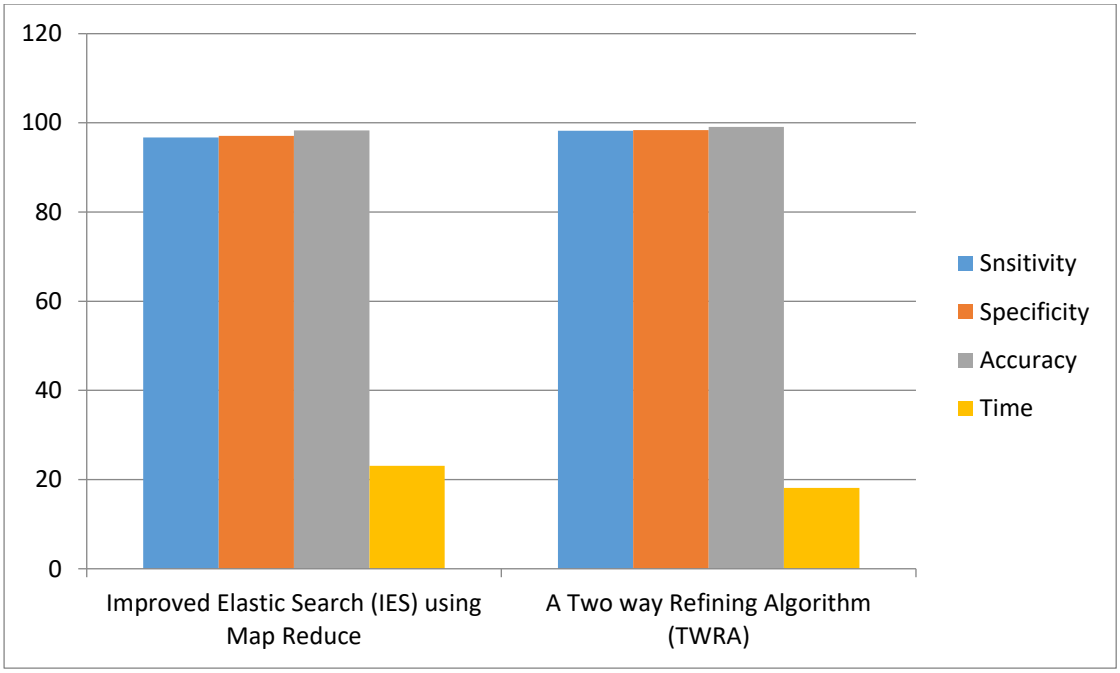


Figure 1: Performance comparison based on parameters

5.Conclusion

An improved Elastic Search is implemented with two way data refining techniques are integrated to get the better results. Improved Advanced Data Cleaning can be done before the pre-processing of data. The advanced data filters are utilized to remove the shards from the output. Elastic Search shard analyses the shard copy and provides an overview of the corruption found. To proceed you must then confirm that you want to remove the corrupted data. This can be done by using the TWRA

References

1. P. Ying, X. Jungang, C. Zhiwang, and S. Jian, "IKMC:An Improved K-Medoids Clustering Method for NearDuplicated Records Detection," in Computational Intelligence and Software Engineering, 2009. CiSE 2009.International Conference on, Wuhan, 2009, pp. 1 - 4
2. Z. Wei, W. Feng, and L. Peipei, "Research on Cleaning Inaccurate Data in Production," in Service Systems and Service Management (ICSSSM), 2012 9th International Conference on, Shanghai, 2012.
3. L. Zhe and Z. Zhi-gang, "An Algorithm of Detection Duplicate Information Based on Segment," in International Conference on Computational Aspects ofSocial Networks, 2010.
4. H., H. Shahri and Z., A., A. Barforush, "Data Mining for Removing Fuzzy Duplicates Using Fuzzy Inference," in Processing NAFIPS '04. IEEE Annual Meeting of the (Volume:1), 2004.

5. Elastic search refresh interval vs indexing performance. Available from: https://sematext.com/training/elasticsearch/. Date Accessed: 8/07/2013.
6. O. Baysal, R. Holmes, and M. W. Godfrey. Developer dashboards: The need for qualitative analytics. IEEE Software, 30(4):46–52, 2013.
7. Gormley C, Tong Z. Elasticsearch: The Definitive Guide. O'Reilly Books; 2015.
8. T. Prakash, M. Kakkar and K. Patel, "Geo-identification of web users through logs using ELK stack," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016, pp. 606-610.
9. K. Elmagrmid, P. G. Ipeirotis, and V. S. Verykois, "Replicated Shard Detection: A Survey," IEEE Transaction on Knowledge and Data Engineering, vol. 19, no. 1, IEEE, 2007.
10. Mining Modern Repositories with Elastic search:https://cs.uwaterloo.ca/~okononen/msr2014.pdf .
11. Salakhutdinov, R., & Mnih, A. (2008). Probabilistic matrix factorization. Advances in Neural Information Processing Systems 20. Cambridge, MA: MIT Press.
12. Subhani Shaik, N N Malleswara Rao (2018). Identification of Redundant Shards in the textual context Using Quality Assortment Algorithm in Elastic Search. Jour of Adv Research in Dynamical & Control Systems. Vol.10.01.
13. Subhani Shaik, N N Malleswara Rao (2019).Removal of Semi-Duplicated and Fully Duplicate Shards using Hadoop Techniques for Elastic Search. IJEAT, Vol.8.3.