International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Survey Paper on Elastic Search

Pragya Gupta[1], Sreeja Nair[2]

Abstract: ***Elastic search is a way to organize data and make it easily accessible. It is a server based search on Lucene. It is a highly scalable, distributed and full-text search engine. Elastic search is developed in Java. It is published as open source under the terms of the Apache License. Elasticsearch is the most popular enterprise search engine. Elasticsearch includes all advances in speed, security, scalability, and hardware efficiency. Elastic search is a tool for querying written words. It can perform some other smart tasks, but its principal is returning text similar to a given query and statistical analyses of a quantity of text. Elasticsearch is a standalone database server, which is written in Java and using HTTP/JSON protocol,it's takes data and optimized the data according to language based searches and stores it in a sophisticated format. Elastic search is very convenient, supporting clustering and leader selection out of the box. Whether it's searching a database of trade products by description, finding similar text in a body of crawled web pages, Elasticsearch is imaginary excellent. Elasticsearch is an excellent tool.***

Keywords: Elastic Search, Lucene

1. Introduction

The principal of Elasticsearch's intelligent search engine is basically another software project called Lucene. It is possibly easiest to understand elasticsearch as a part of infrastructure built nearby Lucene's Java libraries. In elasticsearch everything is related to the actual algorithms for matching text and storing optimized indexes of query terms is executed by Lucene. Elasticsearch itself provides a more functional and compact API, scalability, and operational tools overhead Lucene's search implementation.

Lucene is ancient in internet years, seeing back to 1999. It's also extremely widespread and established. Lucene is used by inexpressible numbers of companies, running the scope from huge corporations such as Twitter, to small startups. Lucene is demonstrated, tested, and is widely considered best-of-breed in open-source search software

Most of the rational effort users of elasticsearch allocate to the task of search will be related to using the Lucene APIs elasticsearch exposes.

2. The Value Adds

While Lucene is eccentric tool, it is unwieldy to use directly, and provides few features for scaling previous a single machine. Elasticsearch provides a more instinctive and simple API than the basic Lucene Java API. Critically, lucene also provides substructure story that makes scaling across machines and data centers relatively simple. Some of the features Elasticsearch brings to basic Lucene:

- A simpler API
- Inter-operation with non-Java/JVM languages
- Operational ease of use
- Clustering and replication
- Good defaults for complex Lucene classes

3. Basic Concepts

There are a few concepts that come with Elastic Search and their understanding is critical to fully understand how Elasticsearch works and operates

3.1 Index

Elasticsearch stores its data in one or more indices. Using similarities from the SQL world, indexing is similar to a database. It is used to store the documents and read them from it. Elasticsearch uses Apache Lucene library to write and read the data from the index. Elasticsearch index may be built of more than a single Apache Lucene Index by using "Shards".

3.2 Document

Document is the main entity in the Elasticsearch world. At the end, all use cases of using Elasticsearch can be carried at a point where it is all about searching for documents and analyzing them. Document consists of fields, and each field identified by its name and can contain one or multiple values. Each document may have different set of fields; there is no schema or imposed structure.

3.3 Type

Each document in Elasticsearch has its type defined. This allows us to store various document types in one index and different mapping for different documents types.

3.4 Mapping

All documents are analyzed before being indexed. The input text is divided into tokens, which tokens should be filtered out, or what additional processing, such as removing HTML tags, is needed. This is where mapping comes into play; it holds all the information about the analysis chain.

3.5 Node

The single instance of the Elasticsearch server is called a node. A single node in Elasticsearch deployment can be sufficient for many simple use cases.Elasticsearch is designed to index and search our data, so the first type of node is the data node. Such nodes hold the data and search on them. The second type of node is the master node a node that works as supervisor of the cluster controlling other nodes' work. The third node type is the Tribe node, which is new and was introduced in Elasticsearch. The tribe node can join multiple Clusters and thus act as a bridge between them,

allowing us to execute almost all Elasticsearch functionalities on multiple clusters.

3.6 Cluster

Cluster is a set of Elasticsearch nodes that works together. The distributed nature of Elasticsearch allows us to easily handle data that is too large for a single node to handle.

3.7 Shard

Clustering allows storing information volumes that exceed abilities of a single server. To achieve this requirement, Elasticsearch spreads data to several physical Lucene indices these indices are called Shards, and all the parts of the index is called sharding. Elasticsearch can do this automatically and all the parts of the index (shards) are visible to the user as one big index.

3.8 Replica

Sharding allows pushing more data into Elasticsearch that is possible for a single node to handle. The idea is simple create an additional copy of shard, which can be used for queries just as original, primary shard.

3.9 Key Concepts behind Elasticsearch Architecture

Elasticsearch was built with few concepts in mind. The development team wanted to make it easy to use and highly scalable. These core features are visible in every corner of Elasticsearch. The main features are as follows:

1) Reasonable default values that allow users to start using Elasticsearch just after installing it. This includes built-in discovery and auto- configuration.
2) Working in distributed mode by default. Node assumes that they are or will be a part of the cluster.
3) Peer to peer architecture without single point of failure (SPOF).Nodes automatically connect to other machines in the cluster for the data interchange and mutual monitoring. This covers automatic replication of shards.
4) Easily scalable both in terms of capacity and the amount of data by adding new nodes to the cluster.
5) Elastic search dose not impose restrictions on the data organization in the index. This allows user to adjust to the existing data model.Near Real Time (NRT) searching and versioning. Because of the distributed nature of elastic search, it is impossible to avoid delay and temporary differences between data located on the different nodes. Elastic search tries to reduce these issues and provide additional mechanisms as versioning.

4. Working of Elastic Search

The following section will include information on key Elasticsearch features, such as bootstrap, failure detection, data indexing, querying, and so on.

4.1 The start-up Process

When Elasticsearch node starts, it uses the discovery module to find the other nodes on the same cluster and connect to them. By default the multicast request is broadcast to the network to find the other Elasticsearch nodes with same cluster name.In the preceding figure, the cluster, one of the nodes that is master eligible is elected as master node. This node is responsible for the managing the cluster state and the process of assigning shards to node in reaction to changes in cluster topology. The master node reads the cluster state and, if necessary, goes into the recovery process. During this state, it checks which shards are available and decides which shards will be the primary shards. After this the whole cluster enters into a yellow state. This means that a cluster is able to run queries, but full throughput and all possibilities are not achieved yet. The next thing to do is to find duplicated shards and treat them as replicas. When a shard has too few replicas, the master node decides where to put missing shards and additional replicas are created based on a primary shard.

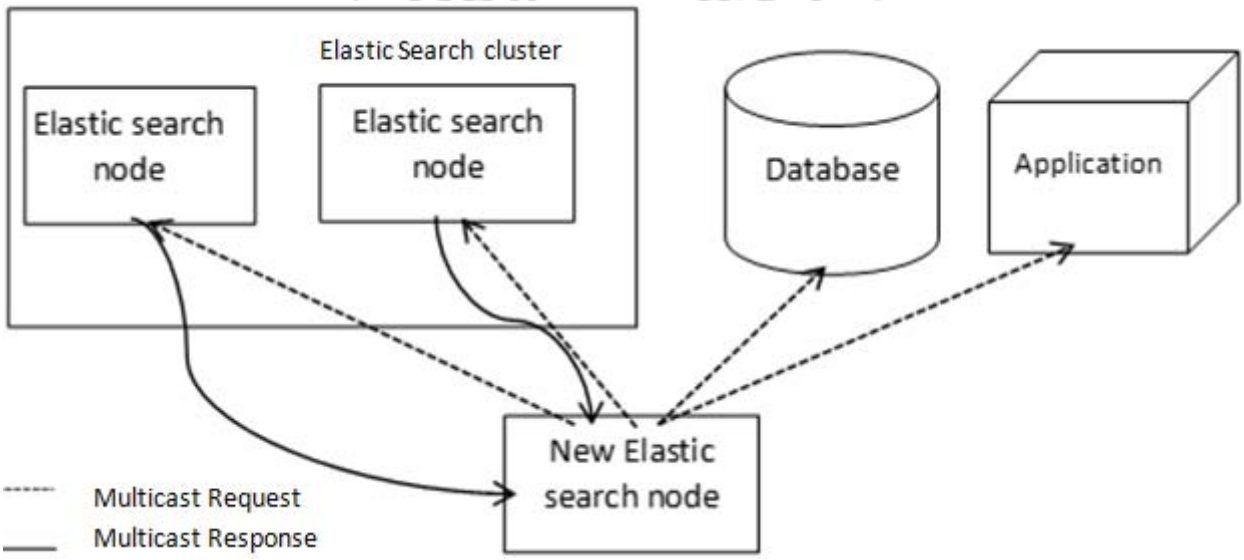

Figure 4.1: The Startup Process of Elastic Search

4.2 Indexing Data

There are few ways to send data to Elasticsearch. The easiest way is using the index API, which allows sending a single document to particular index. The second way allows sending many documents using the bulk API and the UDP bulk API. The difference between these two methods is the connection type. Common bulk command sends documents by HTTP protocol and UDP bulk sends this using connection less datagram protocol. This is faster but not so reliable.

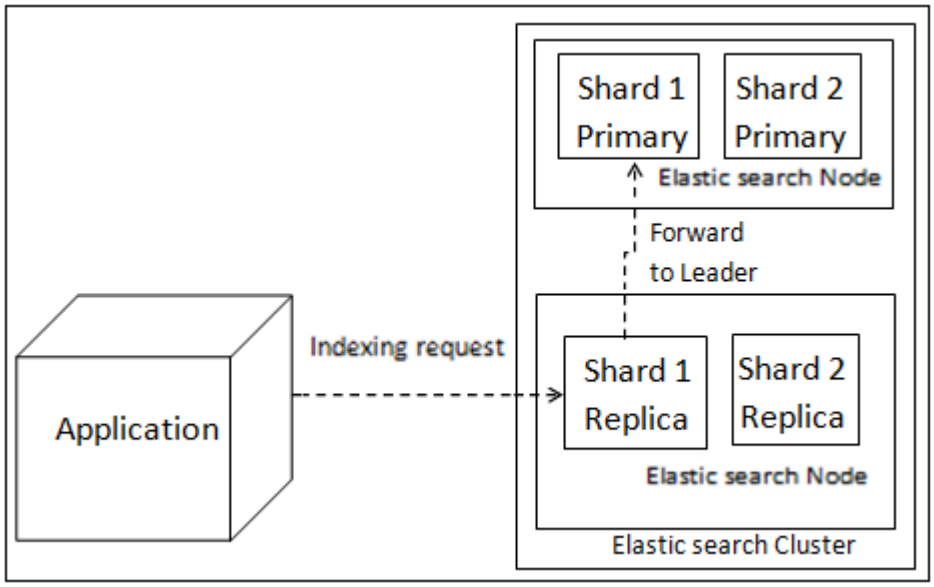

Figure 4.2: The Indexing Process of Elastic Search

4.3 Querying Data

The Query API is a big part of Elastic Search API.The Query process can be divided into two phase: the scatter phase and the gather phase. The scatter phase is about querying all the relevant shards of the index. The gather phase is about gathering the results from the relevant shards, combining them, sorting, processing and returning to the client.

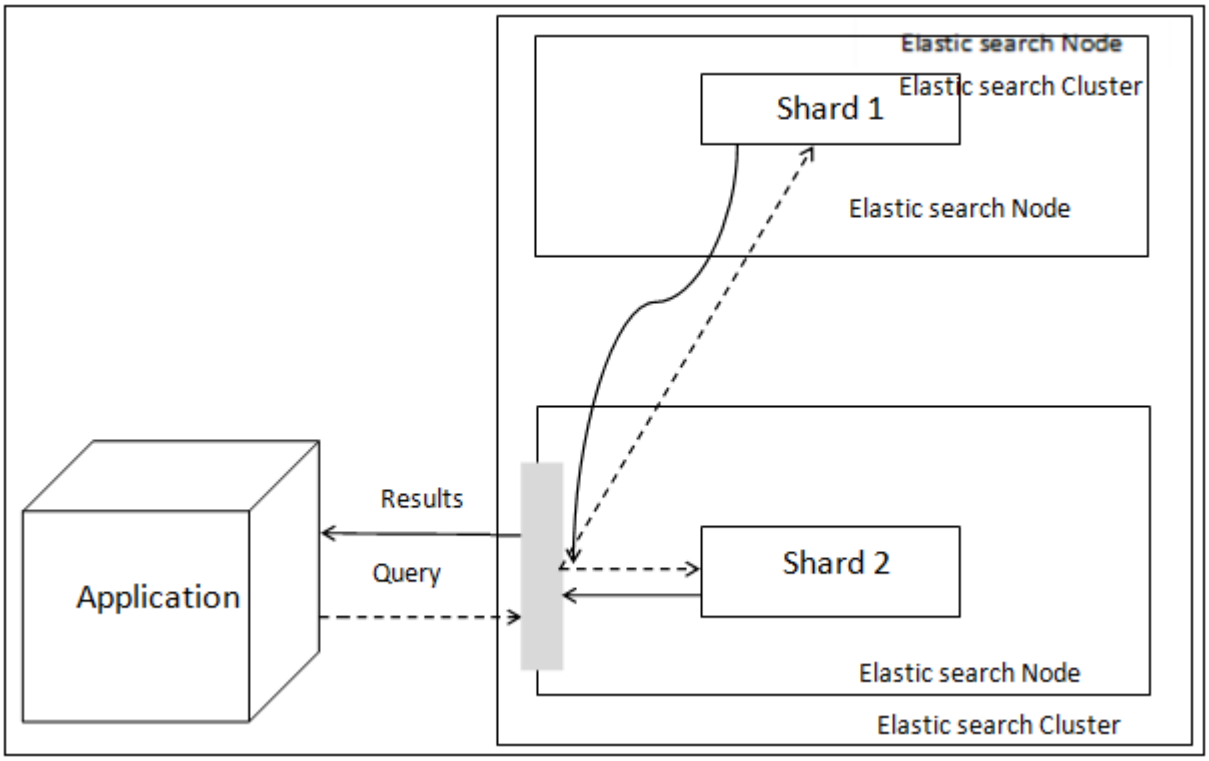

Figure 4.3: The Querying Process of Elastic Search

5. Amazon Elastic Search Service

Amazon Elasticsearch Service is a managed service that makes it easy to deploy, operate, and scale Elasticsearch in the AWS Cloud. Amazon Elastic search Service provision all the resources for cluster and launches it. The service automatically detects the Elastic search nodes and replaces failed nodes, reducing the overhead associated with self-managed Elastic search software and infrastructure. Amazon Elasticsearch Service allows easily scale cluster via a single API call or a few clicks in the AWS Management Console.

5.1Benefits of Amazon Elasticsearch

5.1.1 Simple to Deploy

Amazon Elasticsearch Service easy to deploy Elasticsearch in the AWS cloud. Use the AWS Management Console (API calls) to access a production-ready Amazon Elasticsearch cluster in minutes without worrying about infrastructure provisioning, or installing and maintaining Elasticsearch software.

5.1.2 Easy to Administer

Amazon Elasticsearch Service simplifies time-consuming management tasks such as ensuring high availability, patch management; failure detection and node replacement, backups, and monitoring allowing pursue higher value application development.

5.1.3 Scalable

Amazon Elasticsearch Service enables to monitor cluster through Amazon CloudWatch metrics and resize cluster up or down via a single API call or a few clicks on the AWS Management Console.

5.1.4 Integrated with Logstash and Kibana

Logstash is an open-source data pipeline that helps user process logs and other event data, and loads them into Elasticsearch. Kibana is an open-source analytics and visualization platform that helps user get a better understanding of their data in Elasticsearch.

5.1.5 Cost Effective

Amazon Elasticsearch Service saves user the administrative costs of setting up and managing Elasticsearch. User can scale up and scale down their cluster to deliver optimum performance as data and usage patterns change, paying only.The on-demand pricing allows user to pay for resources by the hour with no long-term commitments and frees user from the costs and complexities of planning, purchasing, and maintaining hardware.

5.1.6 Secure

User can control access to the Elasticsearch APIs using AWS Identity and Access Management (IAM) policies. Using IAM policies user can allow applications to access their Amazon Elasticsearch clusters securely.

6. Conclusion

Elastic search allows to store, search, and analyze huge amount of data quickly and in near real time.Elastic Search is a great open-source search tool that's built on Lucene (like SOLR) but is natively JSON + RESTful. It is been used quite a bit at the Open Knowledge Foundation over the last few years. Plus, as it is easy to setup locally it is an attractive option for digging into data on local machine. Elasticsearch is a standalone database server, written in Java that takes data in and stores it in a sophisticated format optimized for language based searches.

References

[1] Amazon Web Service. [https://aws.amazon.com/elasticsearch-service/]
[2] ElasticSearch Tutorial [http://www.elasticsearchtutorial.com/]
[3] Data Big Zone [https://dzone.com/articles/elasticsearch-getting-started]
[4] https://www.elastic.co/products/elasticsearch