ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

Personalize Search Engine Using Elastic Search

Sachin G. Patil[1], Ramanuj R. Kabra[2], Vishal P. Nikum[3], Prachi V. Patil[4]

U.G. Student, Department of Computer Engineering, SSBT's COET Bambhori, Jalgaon, India

ABSTRACT: As the volume of electronically available information grows, relevant items become harder to find. This work presents an approach to personalizing search results in scientific publication databases. A personalized Web search can provide different search results for different users or organize search results differently for each user, based upon their interests, preferences, and information needs. Personalized web search differs from generic web search, which returns identical research results to all users for identical queries, regardless of varied user interests and information needs. Big data offers the solution for analysing large amount of data. Using technique of Elasticsearch, access to data can be made faster. Elasticsearch uses the concept of indexing to make the search faster. This paper elaborates the search technique of Elasticsearch.

KEYWORDS: Personalize search engine; Elasticsearch; Search uing ElasticSearch.

I. INTRODUCTION

Unstructured or semi- structured data may need to be indexed. RDBMS support services that have a schema. Elasticsearch is a big data technology which is schema less. It is a tool used to search big data. Elasticsearch uses the concept of denormalization for search. Elasticsearch uses the indexing concept. It is a document oriented tool.[1]

Elasticsearch represents data in the form of structured JSON documents, and makes full-text search accessible via RESTful API and web clients for languages like PHP, Python, and Ruby. It's also elastic in the sense that it's easy to scale horizontally.[2] Elasticsearch is a schema less big data technology that uses the indexing concept. It is a document oriented tool. That means once the document is added, it can be searched within a second. Elasticsearch can be used for many use cases like analytics store, auto completer, spell checker, alerting engine, and as a general purpose document store; Full text search is one of it. It is a robust search engine that provides a quick full text search over various documents. It searches within full text fields to find the document and return the most relevant result first. The relevancy of documents is good as Elasticsearch uses boolean model to find document. As soon as a document matches a query, Lucene calculates its score for that query, combining the scores of each matching term. The relevance of the document can be calculated using practical scoring function.

Personalized web search differs from generic web search, which returns identical results to all users for identical queries, regardless of varied user interests and information needs. When queries are issued to search engine, most return the same results to users. In fact, the vast majority of queries to search engines are short and ambiguous. Different users may have completely different information needs and goals when using precisely the same query.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

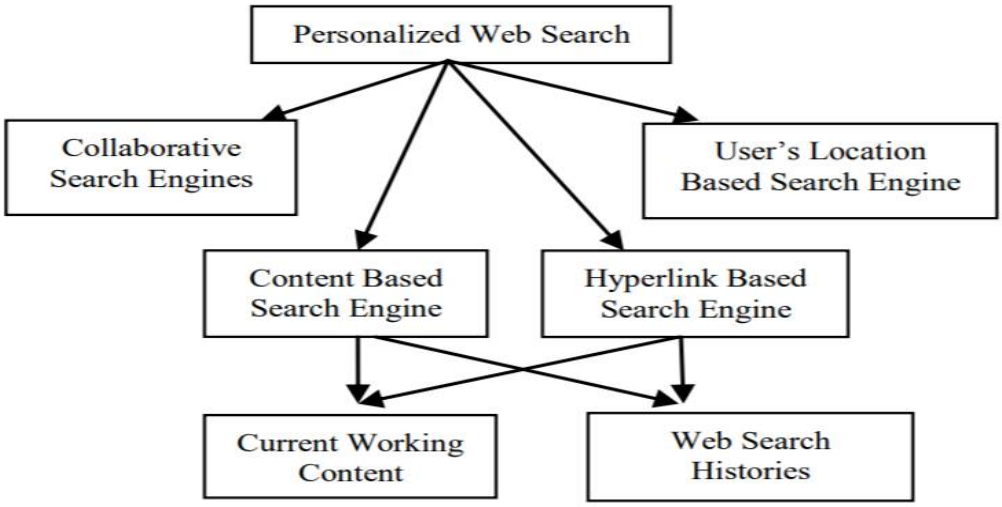
International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

II. APPROACHES TO PERSONALIZE SEARCH RESULTS



Personalize Search Result Structure 1

Personalized web search can be achieved by checking content similarity between web pages and user profiles. Some work has represented user interests with topical categories. User's topical interests are either explicitly specified by users themselves, or can be automatically learned by classifying implicit user data. Search results are filtered or re-ranked by checking the similarity of topics between search results and user profiles.

III. SEARCHING WITH ELASTIC SEARCH

Elastic search database is document oriented. By default, the full document is returned as part of all searches. The Elastic search uses the inverted index to search the term. The terms are sorted in ascending order. Elastic search provides the ability to subdivide the index into multiple pieces called shards. When a new document is stored and indexed, Elastic search server defines the shard responsible for that document. When an index is created, user can simply define the number of shards. Each shard is in itself a fully-functional and independent "index" that can be hosted on any node in the cluster.

IV. ELASTIC SEARCH - WORKING MODEL

A. Processing Nodes

Once Elastic search node starts, it uses the discovery module to find the other nodes on the same cluster and connect to them. The master node reads the cluster state and 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

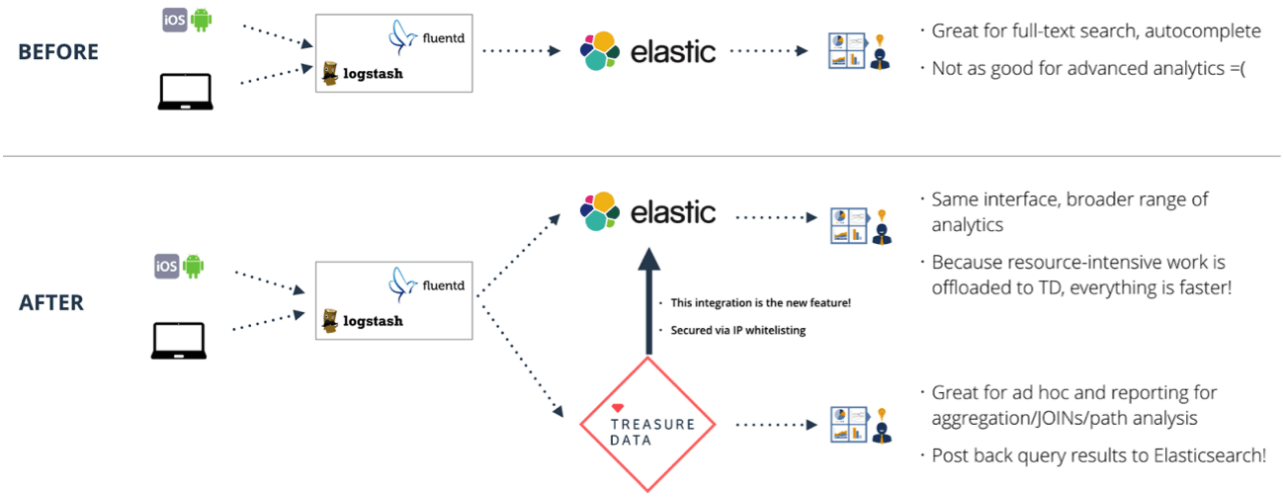
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

B. Indexing Data and Querying Data

The Query API is an important section of Elastic search API. The Query process is divided into two phase: the scatter phase is about querying all the relevant shards of the index and the gather phase is about gathering the results from the relevant shards, combining them, sorting, processing and returning to the client.



Indexing Data and Querying Data 1

1) Query DSL: The query DSL is a flexible, expressive search language that Elastic search uses to expose most of the power of Lucene through a simple JSON interface. It makes your queries more flexible, more precise, easier to read, and easier to debug. The Query DSL is Elastic search's way of making Lucene's query syntax accessible to users, allowing complex queries to be composed using a JSON syntax. Like Lucene, there are basic queries such as term or prefix queries and also compound queries like the bool query.[4]

A query clause typically has this structure:

```
{
        QUERY_NAME:
        {  ARGUMENT: VALUE,
        ARGUMENT: VALUE,
        }
}
```

If it references one particular field, it has this structure:

```
{ QUERY_NAME:
        { FIELD_NAME:
        { ARGUMENT: VALUE,
         ARGUMENT: VALUE,
         }
 }
}
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

In Elastic Search, searching is carried out by using query based on JSON. Query is made up of two clauses.

2) Leaf Query Clauses: These clauses are match, term or range, which look for a specific value in specific field.

3) Compound Query Clauses: These queries are a combination of leaf query clauses and other compound queries to extract the desired information. Elastic Search supports a large number of queries. A query starts with a query key word and then has conditions and filters inside in the form of JSON object.

V. BASICS OF PERSONALIZED SEARCH

1) Creation of User Profile

To provide personalized search results to users, personalized web search maintains a user profile for each individual. A user profile stores information about user interests and preferences. It is generated and updated by exploiting userrelated information. Such information may include:

- Information about the user like age, gender, education, language, country, address, interest areas, and other information
- Search history, including previous queries and clicked documents.
- Other user documents, such as bookmarks, favorite web sites, visited pages, and emails.

2) Content Based Personalized Search

By checking content similarities between web pages and user profile personalized search can be improved [5]. User's interests can be automatically learned by classifying implicit user data. Search results are filtered or re-ranked by checking the similarity of topics between search results and user profiles. User-issued queries and user-selected documents are categorized into concept hierarchies that are accumulated to generate a user profile. When the user issues a query, each returned result is also classified. The documents are re-ranked based upon how well the document categories match user interest profiles. Chirita et al. [7] use the ODP (Open Directory Project, http://www.dmoz.org/) hierarchy to implement personalized search. User favorite topics nodes are manually specified in the ODP hierarchy. Each document is categorized into one or several topic nodes in the same ODP hierarchy. The distances between the user topic nodes and the document topic nodes are then used to re-rank search results.

VI. SIMULATION AND RESULTS

Elastic search can be used to search all kinds of documents. It provides scalable search, has near real-time search, and supports multitenancy. Elastic search is distributed, which means that indices can be divided into shards and each shard can have zero or more replicas. Each node hosts one or more shards, and acts as a coordinator to delegate operations to the correct shard(s). Rebalancing and routing are done automatically". Related data is often stored in the same index, which consists of one or more primary shards, and zero or more replica shards. Once an index has been created, the number of primary shards cannot be changed.

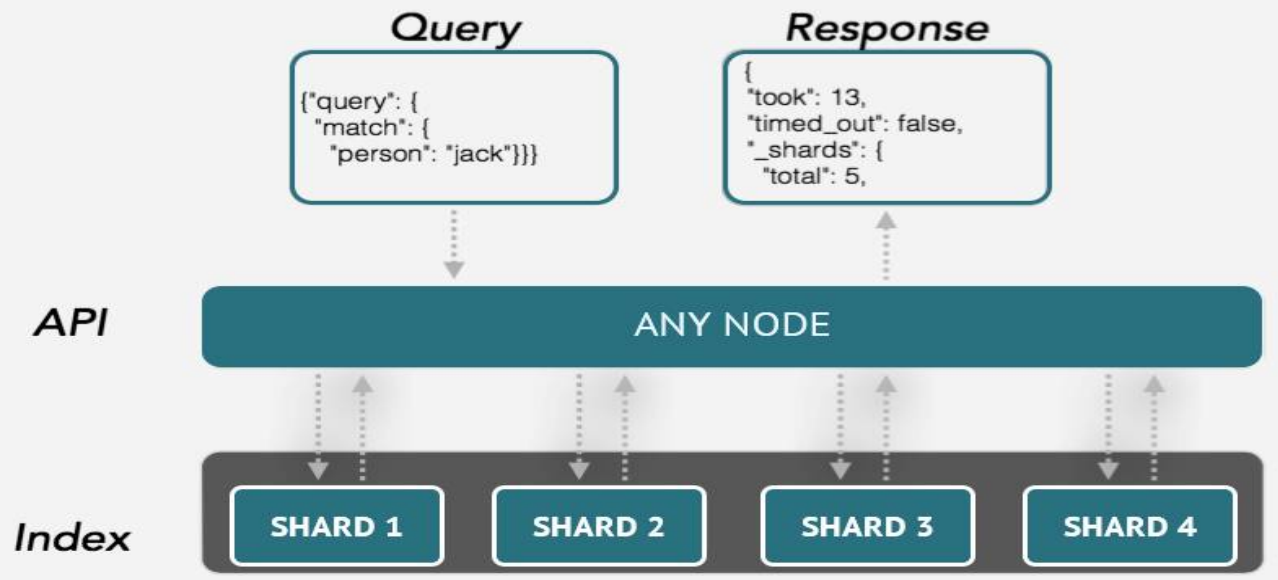
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019



Response With Search Results 1

Elastic search does not have tables, and a schema is not required. Elastic search stores data documents that consist of JSON strings inside an index. The field is like the columns of the SQL database and the value represents the data in the row cells

VI. CONCLUSION AND FUTURE SCOPE

This article presents an automatic approach to personalizing Web searches given a set of user interests. Elastic search can scale out to hundreds or even thousands of servers and handle megabytes of data. Elastic-search is distributed by nature, and it is designed to hide the complexity that comes with being distributed. The distributed aspect of Elastic search is largely transparent. The approach is well suited for a workplace setting where information about professional interests and skills can be obtained automatically from an employee's resume or a database using an IE tool or database queries.

REFERENCES

[1] Darshita Kalyani, Dr. Devarshi Mehta," Paper on Searching and Indexing Using Elasticsearch" International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 6 Issue 6 June 2017.
[2] Subhani shaik, Nallamothu Naga Malleswara Rao, "Enhancement of Searching and analyzing the document using Elastic Search", International Research Journal of Engineering and Technology.
[3] Chanchala Joshi , Teena Jaiswal, Himanshu Gaur," An Overview Study of Personalized Web Search", International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.
[4] Zhongming Ma, Gautam Pant and Olivia R. Liu Sheng. "Interest-Based Personalized Search," ACM Transactions on Information Systems, Vol 25, Issue 2, Article 5, February 2007.
[5] Subhani shaik , Nalamothu Naga Malleswara Rao,"A Conceptual Review of Elastic Search – Survey Paper", International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887 Volume 5 Issue XI November 2017.
[6] P.A. Chirita, C. Firan, and W. Nejdl, "Summarizing Local Context to Personalize Global Web Search", Proc. ACM Int'l Conf. Infor. and Knowledge Management (CIKM), 2006.
[7] Chirita P.A., Nejdl W., Paiu R., and Kohlschu¨tter C. "Using ODP metadata to personalize search". In Proc.31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005, pp. 178–185.