Domain Specific Code Repository using Elasticsearch

Mrs. Chetna Achar[1] Pravin Kumar Kamlashankar Mishra[2]
[1]Professor
[1,2]Department of MCA MET-ICS
[1,2]University of Mumbai, Mumbai-400050, India

Abstract— Various software development companies have their own code repository for future reference. This code repository consist of previous projects and code worked on, by company employees. Taking up a new project or working on a module of some current project, developers come across coding challenges which may have been faced by some other employees previously in the same project or some other project. These employees may not be the part of current project or may have already left the company. But the probability that they found out a solution for the problem is absolute. The solution to this problem exists in the code repository in form of chunks of code in files. However searching for a specific peace of code in the repository may be a tedious job. And sometimes may even take more time than writing the code itself, from scratch. This paper proposes the use of elastic search for indexing the code repository to enormously accelerate the search proses.

Key words: Structured, Semi-structured, Un-structured

I. Introduction

Software development companies have huge amount of data stored in the form of projects and chunks of code in several thousand files. These companies may have been working on specific software development domain for years. The data these companies may have, can be in Giga-Bytes or Tera-bytes. And these type of companies take up similar projects, because this is their domain of expertise. The main aim of these companies are to increasing the speed of development process. So as to decrease the cost of development, by reusing their previous code repositories.

Writing any piece of code from scratch is not only time consuming but may lead to entire iterative cycle to make that piece of code stable. The code stored in their repositories and archives from the previous projects are like a mine of gold to them. The only thing is that you need to mine it, i.e. you need to find the code that suites your purpose in the current project. And this can be easily achieved if the data in the repositories can be stored and indexed properly. Which will make it easy to search for the developers for solutions of coding challenges.

The reason for opting for Elasticsearch is its storage and indexing technique and NoSQL structure. Since the code repository stores files (unstructured-data) and the idea is to organize files with the problem statement, problem domain, code documentation and solution. i.e. (structured-data) the elastic search is best solution as it follows NoSQL structure.

II. An Overview of Current System

The current system simply stores the code in the form of files and/or archives in one or more server inside the company premises. These files and archives may have documentations by requires human efforts to read and understand it. As it is also stored in text based files. The effort required to go through an old documentation is quite tedious and die to human tendency most of the Developers in the company would be reluctant to do so. And in case of new employees, they may not even know if such module ever existed in any of the project ever before. In spite of having the solution right at their disposal, most of them are unaware of it and take up an effort to write the code from scratch. This not only leads to time inefficient utilization of time but also makes the practice of maintaining the code repository useless. As no code is reused.

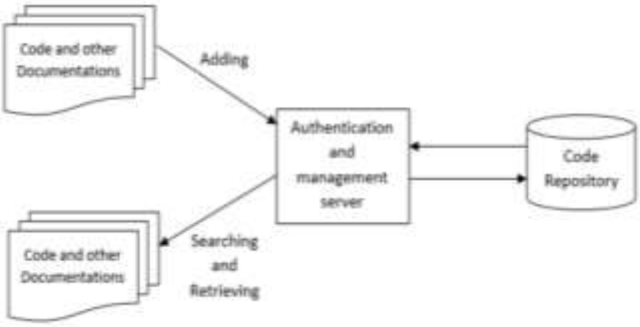
Code and other Documentations
Adding
Authentication and management server
Code Repository
Code and other Documentations
Searching and Retrieving

Fig. 1: Current code repository management system.

III. An Overview of Proposed System

The proposed system can be implemented in two ways wiz:

A. *Open For Public Use:*

The open for public use method can be used where the developer or the company wants to share their work with others. Some developers like to avail their projects and work for public.

B. *Private or Internal To the Company:*

This type of code repository is for the company's private or internal use. This type of code repository is for companies personal use and not available for public use.

In both scenarios the implementation and structure of the system will remains constant.

1) Structure of the repository is that, there will be two sections. In one section the user can add the code solution along with the details of the problem such as the problem statement, problem domain (weather the problem belongs to ERP, CRM or any other domain) and author of the code solution along with an elaborated documentation. This data will be indexed using Elasticsearch, based on the keywords in the documentations provided along with the code.
2) In the other section the user can search the code solution for the problem. As the elastic search provides full text search the user can search for the code solution based on the problem domain, problem statement and also based on author of code.

3) In the result of the search there will be set of solutions returned by the system. The developer needs to select and further explore the solution. The results will have problem statement, problem domain, solution or code author name, code documentation and the code or solution itself.

IV. WORKING OF ELASTICSEARCH

Elasticsearch is search engine server based on Apache Lucene and written in java. Elasticsearch is completely open source. Elasticsearch is fast, scalable and distributed. Each instance of Elasticsearch is called as node. The Elasticsearch indexes the text, binary and other type of files. Elastic search uses JSON as primary data format.

Elasticsearch can be distributed across multiple machines which makes it horizontally scalable. Elasticsearch is capable of providing full text search along with handling huge amount data and a simple yet powerful API. Elasticsearch can be horizontally distributed makes it fault tolerant and high uptime (availability).

Elasticsearch provides support for connecting to database with different programming languages or the programming frameworks with the help of connection libraries or drivers. Elasticsearch provides its own query language for users to access and manipulate data name Query DSL (Domain Specific Language).

A. Indexing:

The Elasticsearch performs the indexing of documents when it is inserted, the indexing is preformed based on id and document type. If the data does not contain natural id Elasticsearch auto generates id for the data. If the data is updated then its version number is incremented. If id and document type for data already exist in database then it is overridden.

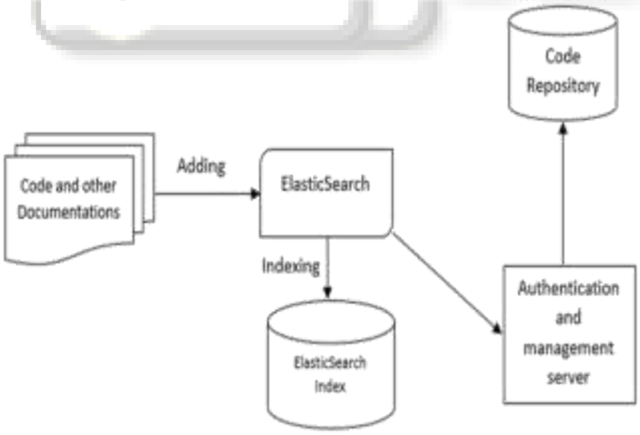

Fig. 2: Indexing of Codes Repository using Elasticsearch.

B. Searching:

It provides to types of search mechanisms query based and filter based. The filter based search mechanism is considerably faster than query based search. The filter based search is faster because it does not give the calculated ranking or score to each and every result as of query based search.

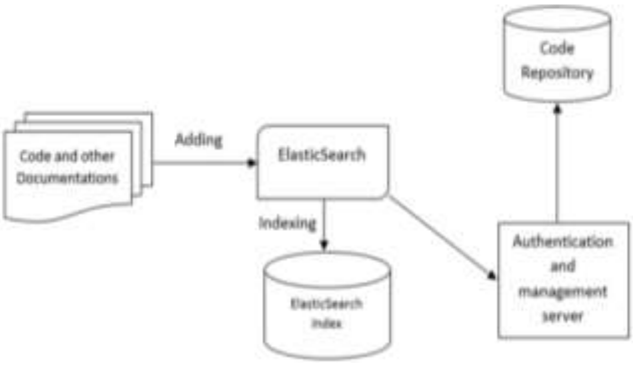

Fig. 3: Searching in code repository using Elasticsearch

V. WORKING OF ELASTICSEARCH INDEXING IN PROPOSED SYSTEM

1) Authentication: In case of open for public use, the user authentication is not required since it is available to all for access. But in case of repository that is private or internal to the company the authentication mechanism is required as the projects and code should not be accessed to all.
2) The next step is inserting the code solution in the repository. In this step the code or solution is entered in the system along with some other information. The information related to code or solution such as the problem domain, so that the user can identify, whether the problem domain is related to the type of solution he/she is searching for. The next information is the problem statement which specifies the exact purpose of the code or solution. But the most important piece of information is the code documentation, assuming that the code documentation is available. It not only saves developers time to understanding the code solution. But also makes the search result more precise, as these documentations are also indexed by Elasticsearch. And lastly, the code itself is stored as text files. Which facilitates searches using code snippets as indexed keywords for searching.
3) The code repository can have different types of files or documents, such as normal text files (html or css files rendered by browser), code file and the compiled binary files. The elastic search performs the indexing at two times, first at the time of insertion of the documents and the second is when the document is searched. Elasticsearch adds a version number to the index file for particular document after each search. The time when the user will insert the code solution, Elasticsearch will add a document id in the index for that document (if natural document id is not given by user) and a version number. Each time the document is added to a search result its version number is updated. Elastic search uses analysers, tokenisers and token filters for indexing the code documents.
4) Next step is searching the document in the repository. When user searches a code solution using problem domain, problem statement, code documentation or the author name, Elasticsearch searches the keyword in the indexes (shards). It then takes the corresponding document id, document type and

version number and search the document or code file and return the result.

5) Proposed system uses query based filter search as it gives the ranking to each search result based on the latest added code solution and the number of times the solution is searched. Each time the search is performed, elastic search will give ranking to the search result and update the version number of the document or code file.
6) After each search Elasticsearch will make use of analysers, tokenisers and token filters to update the index file for the particular document and search entry.

VI. Conclusion and future work

The system proposed in this paper provides a mechanism for utilising the code and projects that are store on the machines which remain unused. Even though the companies allocate some resources such as the machine, memory or drive space. But due to lack of proper mechanism for storage and indexing the code or solution is (which is an asset for the company) remains unused. And the projects and code solution become just peace of data.

So the proposed system makes the unused assets and resources useable.

References

[1] Elasticsearch - The Definitive Guide by Clinton Gormley (Author), Zachary Tong (Author)
[2] Mastering ElasticSearch by Rafal Kuc (Author), Marek Rogozinski (Author)
[3] Elasticsearch Server: A Practical guide to building fast, scalable, and flexible search solutions with clear and easy-to-understand examples, 2/E by Marek Rogozinski (Author)
[4] ElasticSearch Server by Rafal Kuc (Author), Marek Rogozinski (Author)
[5] Elasticsearch tutorial :http://www.elasticsearchtutorial.com
[6] Exploring Elasticsearch: http://www.exploringelasticsearch.com
[7] Querying Elasticsearch: http://okfnlabs.org/blog/2013/07/01/elasticsearch-query-tutorial.html