UDC 378.147.134:37.013.77

Tulchak L.V.,
Senior teacher of Foreign Languages Department,VNTU
Shevchuk V.V.,
Student, VNTU

IMPLEMENTING ELASTICSEARCH IN SOCIAL NETWORK

The current period of the civilized society is characterized by process of informatization. Informatization of the society is a global social process, which in its turn characterized by the dominant activity in the sphere of social production particularly by collection, storage, production, processing, transmission and use of the information. It is transmited on the basis of modern microprocessor and computer technology, as well as on various means of information exchange. That‘s why, quite valuable characteristic of information in terms of material processing, transmission and search are considered here. Nowadays, there is a complex problem about working with very large scale data sets. So a new technology such as Elasticsearch comes to help us to solve this problem.

Elasticsearch is a search server based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. Elasticsearch can be used to search all kinds of documents. A document is a basic unit of information that can be indexed. For example, it can have a document for a single customer, another document for a single product, and yet another for a single order. This document is expressed in JSON (JavaScript Object Notation) which is an ubiquitous internet data interchange format [1, c. 4].

Within an index/type, it can store as many documents as you want. Note that although a document physically resides in an index, a document actually must be indexed/assigned to a type inside an index.

Elasticsearch is a near real time search platform. This means that there is a slight latency (normally one second) since the time it indexes a document until the time it becomes searchable. It provides scalable search, and supports multitenancy. Elasticsearch is distributed, which means that indices can be divided into shards and each shard can have zero replicas or morethan one. Each node hosts one or more shards, and acts as a coordinator to delegate operations to the correct shards. Rebalancing and routing are done automatically [1, c. 5].

Elasticsearch.Net is a very low level, dependency free, client that has no opinions about how you build and represent your requests and responses. It has abstracted enough so that all the Elasticsearch API endpoints are represented as methods but not too much to get in the way of how you want to build your json/request/response objects. It also comes with built in, configurable/overridable, cluster failover retry mechanisms [2, c. 507].

NEST is a high level client that has the advantage of having mapped all the request and response objects, comes with a strongly typed query DSL that maps 1 to 1 with the Elasticsearch query DSL, and takes advantage of specific .NET features such as covariant results and auto mapping of POCOs. NEST internally uses and still exposes the low level Elasticsearch.Net client.

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. You use Kibana to search, view, and interact with data stored in Elasticsearch indices. You can easily perform advanced data analysis and visualize your data in a variety of charts, tables, and maps [3, с. 1].

Kibana makes it easy to understand large volumes of data. Its simple, browserbased interface enables you to quickly create and share dynamic dashboards that display changes to Elasticsearch queries in real time.

Setting up Kibana is a snap. Install Kibana and start exploring your Elasticsearch indices in minutes —no code, no additional infrastructure required [3, с. 2].

I created a software module that works in addition to the social network. As it was expected, there is a large flow of users in order to reduce the load on backend operation and to make the site more resistant to requests Elasticsearch was chosen.

According to the statistics of other social networks in search transaction, there is a pattern, which is completely not linear and depends on many factors, such as geolocation, time, well known events and others. Therefore, in addition to slow database I added fast NoSQL, which is the basis for Elasticsearch, then the main database asynchronously synchronized when new data received (when new users registered). According to the given preliminary analysis it can be concluded the application has been developed that allowed to speed up the exchange of information between a client and the server side. This application can be used for educational, scientific, and other purposes with appropriate modification if required.

Literature:

1. Elasticsearch Reference: [Електронний ресурс] / S. A. Schuurman, U. O. Bones, A. S. Banon // definitive guide. – 2016. – № 5. – С 80. – Режим доступу до журн. : https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html.

2. Э. В. Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. В. Троелсен // The Expert's Voice in .NET. – 2013. – № 6. – С. 1311.

3. Kibana Reference: [Електронний ресурс] / N. T. White, G. O. Gupta, // definitive guide. – 2016. – № 3. – С 40. – Режим доступу до журн. : https://www.elastic.co/guide/en/kibana/current/index.html.