College of Engineering,
Trivandrum

Computer Science and Engineering

Elastic Search

Submitted By:
Mohammed Nisham K
mnishamk1995@gmail.com
Roll No 39

Guide:
Vipin Vasu

July 17, 2016

Contents

List of Figures

Abbreviations

API Application Programming Interface

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

REST Representational State Transfer

SQL Structured Query Language

TM Trademark

Abstract

A search server based on Lucene, Elastic Search is a way to organise data and make it readily accessible. A highly scalable, distributed, full-text search engine. Coded completely in java and published as open source under the terms of Apache Licence, it is the most popular enterprise search engine used by giants on the web like facebook, wikipedia, stumbleupon etc. It includes advances in speed, security (with shield plugin), scalability and hardware efficiency out of the box.

Elastc Search is a tool for querying words, its principla task being to return text similar to a given query and statistical and liguistic analysis of it. A standalone database server, communable only through RESTful API's, it takes data and optimises the data according to language based searches and stores it in a sophisticated manner. It supports clustering, and multiple shards out of the box. It makes for an excellent tool.

1 Introduction

Conventional relational databases fail to work for bigdata applications. NoSQL addresses this problem. But again it fails to incorporate full text search on the saved database. Another issue is real-timing, conventional database techniques do not ensure a real time implementation, so a search engine database implementation is required which addresses these issues. [1]

Elastic Search is an open-source realtime distributed search and analytics engine built as an abstraction layer on top of Apache Lucene $^{\mathrm{TM}}$. Lucene is an advanced full text search engine with high performance. But since Lucene is written as a library, it is available only when working with Java and it is very complex to use. [2]

Elastic provides an abstraction layer implemented in Java, which uses Lucene internally for its operations, but provides a simple method to access them via RESTful API's. Since access to operations is via RESTful API's, the usage of Elastic Search does not require coding in java, *ie* it can be used from any language.

Elastic search is widely used by giants on the web. Facebook, Wikipedia, and Github being examples. It was ranked as the most popular search engine database, outranking competitors like Solr and Sphinx. [3]

2 History

Shay Banon, started working with Lucene, and finding its interface tricky started building an abstraction layer over it. It was released as an open-source library for Java called Compass. [2]

Later, working in high performance distributed environments revealed the need for distributed solution to search. So Compass libraries were rewritten from scratch with distributed usage in mind. Making it available to different languages was easy as JSON became an accepted standard of representing complex objects serially and RESTful API's the standard interface to access functionality via HTTP connections, thus its implementaion, serialisation and deserialisation, being available in all languages

3 Features

Elastic Search provides myriads of features on top of the abstraction of and simple interface to Lucene. Some of the features, excluding search are explored here

3.1 Document Oriented

Elastic search is document oriented. Instead of trying to create columns where the data can fit, the data is stored as a JSON object. Not only that, but each field is indexed for searchability. *ie* instead of searching on rows, searching is done on documents. [2]

JSON is a format of representing complex objects serially. It is the standard format accepted by almost all languages and conversion to and from JSON can be done easily. Representing a document as JSON instead of a native object makes it serial, thus having the added benefit of being a viable parameter in a HTTP request, or a RESTful API.

3.1.1 Index

Documents are stored under indices. An index is a name given to a store of data. In actuality it is a collective name representing a group of shards which contain the documents. Indices can be considered as analogous to databases in a relational model. Index$_{(noun)}$ is not to be confused with Index$_{(verb)}$ which means to add a document to an index.

3.1.2 Type

An Index can have documents of multiple types. A type is a logical grouping of documents that are similar, or have content that have most of their fields common. A type can be compared to a table of a relational model.

Each type has its own mapping or schema definition which defines how the fields of the documents of that type must be indexed. For example, a date field would be indexed to allow a range filter to be used on it, while a string field would be indexed for full search capability.

3.1.3 Id

Each document is indexed with an identification field id. Analogous to a row in relational databases, id can be used to get a document from the database, provided you have the index and the type in which it is stored.

Id also helps in routing, or determining which shard the document will be stored in, the details of which are provided in the next section

3.1.4 Dynamic Mapping

Type mappings are optional, if mappings are not provided explicitly, Elastic tries to guess the mapping of the fields provided based on the first document that it is given to index. String fields map to string type, and a standard analyzer is tacked on it while a field with value like '2014/01/01' would be mapped to a date type.

References

[1] O. Kononenko, O. Baysal, R. Holmes, and M. Godfrey, "Mining modern repositories with elasticsearch," *University of Waterloo, Waterloo, ON, Canada*, 2014.

[2] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide.* O'Reilly Media.

[3] "Db-engines ranking - popularity ranking of database management systems." http://www.db-engines.com/en/ranking.