LIÈGE université
Sciences Appliquées

Université de Liège
Faculté des Sciences Appliquées

Elasticsearch: NoSQL for Scalable Data Management

INFO9016-1 - Advanced Databases

Authors
ABDELALEEM Aly - s206111
DELPORTE Guillaume - s191981
HOORELBEKE Jordi - s175615

May 27, 2023

Contents

1 Introduction

1.1 Overview

With the continuous growth of Internet usage and data proliferation in the contemporary era, the need for efficient data solutions has become increasingly critical, contributing to the rapid expansion of this field. Traditional database management systems, while having been the preferred choice for decades, may face challenges in managing the substantial volumes of structured or unstructured data generated daily.

NoSQL databases, offering flexibility, scalability, and high performance, have emerged in response to these evolving market requirements. They are designed to handle large data sets characteristic of the current digital landscape. Of the various types of NoSQL databases, document-oriented databases have attracted notable interest due to their proficiency in managing diverse data formats. **Elasticsearch**, an open-source full-text search and analytics engine, and a document-oriented NoSQL database management system, aligns with these specifications, positioning it as a robust tool in the current data ecosystem.

Document-oriented NoSQL databases are designed to manage complex and diverse data sets. They store data in a semi-structured format, promoting adaptability to changes and a more intuitive, human-readable data model. Therefore, Elasticsearch, with its scalability, near real-time search, and Apache Lucene foundation, exemplifies these new types of databases.

1.2 What This Paper Will Cover

This tutorial will initially explore the features of Elasticsearch, highlighting both its benefits and considerations. The aim is to elucidate the distinctive capabilities of Elasticsearch. Then, we will examine real-life applications of Elasticsearch through case studies from prominent companies like Airbus, Cisco, GitHub, and eBay.

The tutorial is designed to provide a comprehensive understanding of Elasticsearch. By the end, readers are expected to gain a deeper insight into Elasticsearch's potential and its role in contemporary data management and analysis.

In addition, a practical, hands-on tutorial[1] accompanies this paper. This tutorial will utilize Elasticsearch to demonstrate a real-world example and apply some of the concepts discussed in the paper.

[1] The tutorial is presented in a Jupyter Notebook. We recommend using JupyterLab or the Jupyter extension for VSCode to view the file.

2 Features

In this section, we will discuss the features that make Elasticsearch a well-known search and analytic engine with unique capabilities that make it especially helpful for analytics, investigations, monitoring, and alerts.

2.1 Document-Based Model

Similar to ***MongoDB***, ***Elasticsearch*** stores the data in a collection of documents that contain fields. A document is a JSON object that is schema-flexible and is stored with a unique ID. There is a version field to keep up with the changes that happened to the document; it is incremented every time the document is subjected to a change.

Related documents are grouped by an index for faster retrieval time. An index is just a virtual pointer to where data is stored, which is inside a shard, which is where ***Elasticsearch*** searches for data, and it's an important feature that we will discuss next.

2.2 Sharding

When ***Elasticsearch*** creates an index, a shard is created as well by default, and as we explained earlier, a shard is where data is stored. This concept allows you to distribute data across multiple nodes on different machines, which consequently allows for horizontal scaling, meaning you can quickly add shards to adapt to the increase in demand. This also allows for quicker retrieval time because you can take advantage of the data being distributed and query the shards simultaneously in parallel, speeding up the search time.

A shard can be one of two types: a primary shard, in which case the shard has the ability to perform write operations in its documents, or a replica shard, which is an exact copy of the primary shard. The redundancy of replica shards is used as a backup for the data in case of data loss or corruption. This redundancy gives rise to another advantage, which is load balancing. This is a very good way to improve the performance of the search, as the requests can be distributed among the other shards to help keep up with the increasing demand.

Elasticsearch does not support indices sharing shards; a shard belongs to one index and one index only. To share data across multiple indices, the data must be replicated in their respective indices. On the other hand, Elasticsearch has several APIs to manage data across multiple indices:

- Index aliases: This API creates a pointer for one or more indexes and enables you to apply operations to multiple indices as if they were one.
- Re-index API: This API allows you to re-index data from one or multiple indices; this can help in case you want to refactor the structure of your data.

2.3 REST API

Everything in ***Elasticsearch*** can be done over an exposed, well-documented REST API. This feature allows for easy integration with other technologies, as they can just use the exposed API to do or get what they need from the database without having to worry about how to implement it or compatibility issues.

2.4 The Elastic Stack

Elasticsearch is part of the Elastic Stack, formerly known as the ELK Stack, which consists of:

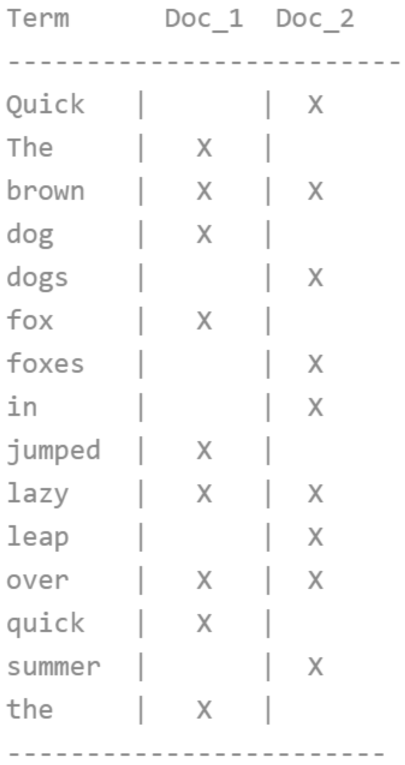
1. **E**lasticsearch a search and analytic engine
2. **L**ogstash a tool for data collection and ingestion
3. **K**ibana a tool for data visualization
4. Beats a tool to send data from hundreds or thousands of machines and systems to Logstash or Elasticsearch.

Together, they create a very flexible data management and analysis platform. Capable of monitoring, alerting, and analyzing data, especially log data.

2.5 Full-text Search

This feature is what gives ***Elasticsearch*** somewhat of a competitive edge. First, let's define what a "full-text search" is. It is the ability to search for words in a database and retrieve complete, partial, combinations, or misspelled matches in the database or any other data source.

Full-text search in ***Elasticsearch*** is very fast, due to the utilization of an inverted index, what it does is tokenize every unique word in a document, similar to that of a lexer in a compiler, and create a list in which each word appears, as well as for each word a list of the documents it appears in, as we can see in the below Figure 1



```
Term      Doc_1  Doc_2
-------------------------
Quick   |       |  X
The     |   X   |
brown   |   X   |  X
dog     |   X   |
dogs    |       |  X
fox     |   X   |
foxes   |       |  X
in      |       |  X
jumped  |   X   |
lazy    |   X   |  X
leap    |       |  X
over    |   X   |  X
quick   |   X   |
summer  |       |  X
the     |   X   |
------------------------
```

Figure 1: Result of inverted index example

Elasticsearch's full-text search returns the data sorted by a generated score. Based on the relevancy of the word in the document, it gives a higher score to exact matches and a lower score to partial matches, and so on. This is a genuinely new concept compared to that used in a traditional relational database, which searches only for exact matches.

This feature is convenient for search engines in which misspelling is common, speed and relevancy are important, and the search requires searching through a huge amount of data.

3 Benefits & Drawbacks of Elastic Search

While every technology comes with its own set of advantages and limitations, the effectiveness of its application can significantly vary depending on the context. Consequently, understanding the potential benefits and drawbacks of Elasticsearch is crucial to optimally leverage its features and anticipate any challenges that might arise in its usage. Hence, in this section, we will outline and elaborate on some of the key benefits and drawbacks of Elasticsearch.

3.1 Benefits

- **Developed in Java**: The fact that Elasticsearch is developed in Java confers a level of portability to the software.
- **Full-Text Search**: Elasticsearch provides robust full-text search capabilities, which can allow for efficient indexing and retrieval of large volumes of data. Its functionality includes identifying and retrieving documents that align with search criteria, thereby enhancing data search and organization.
- **Scalability**: With its design, Elasticsearch demonstrates scalability, capable of managing large quantities of data and efficiently distributing this across multiple nodes.
- **Real-Time Analytics**: Elasticsearch supports real-time analytics, which facilitates the extraction of insights from data immediately following its ingestion.
- **Speed**: Elasticsearch's efficiency in processing queries swiftly can be advantageous in applications designed to serve a large number of users globally.
- **Integration**: As a part of the Elastic Stack, which includes Logstash and Kibana, Elasticsearch benefits from seamless integration with other tools dedicated to data collection, visualization, and management.

3.2 Drawbacks

It is noteworthy that, in addition to its numerous benefits, Elasticsearch also presents certain considerations that might be viewed as drawbacks under specific circumstances.

- **Complexity**: The usage and management of Elasticsearch might be perceived as complex due to its diverse range of capabilities, potentially necessitating a learning curve for full utilization.
- **Lack of Built-In Multilingual Support**: In contrast to some competing technologies such as Apache Solr, Elasticsearch does not currently include built-in support for handling request and response data in multiple languages.
- **Document Oriented**: As a document-oriented technology, Elasticsearch may not be the optimal choice for applications that require the management of relational data.
- **No Support for traditional SQL-Style JOIN**: Elasticsearch does not support traditional SQL-style joins due to its scalable design, which focuses on avoiding potentially resource-intensive operations across large amounts of distributed data.

If JOIN-like operations are necessary, nested objects can serve as a potential alternative. However, it should be noted that Elasticsearch is primarily designed for diverse search operations and optimized for search queries. Thus, it may be beneficial to structure data to capitalize on the strengths of Elasticsearch, rather than attempting to use it in the same manner as a traditional SQL database.

4 Real World Examples

To gain a comprehensive understanding of the immense potential and capabilities of Elasticsearch, it is essential to explore ways in which organizations are using this technology. The prominence of Elasticsearch among companies leveraging its advantages becomes evident even without delving too deeply into real-world examples. In fact, simply having a look at the list of some companies using Elasticsearch from the official documentation would suffice to convince anyone of the importance of Elasticsearch. However, to delve deeper into its significance, we will highlight some prominent organizations that rely on Elasticsearch to drive their operations and achieve their goals.

4.1 Airbus

Airbus[2], the leading European aeronautics manufacturer, utilizes Elasticsearch as a search and indexing engine to power its ADNS (Advanced Data Navigation Services) platform. With the objective of providing near real-time access to aircraft technical documents, Airbus developed ADNS to replace their previous document digitization solution. Airbus ensures that approximately 80000 internal and external users can quickly search and access a vast database of technical documents related to Airbus aircraft models thanks to Elasticsearch's search capability to fulfill up to 3000 requests per minute in less than 2 seconds.

The success of ADNS demonstrates the significant role Elasticsearch plays in enabling organizations like Airbus to streamline operations, enhance customer service, and efficiently manage vast amounts of critical data.

4.2 Cisco

Cisco Systems[3], a renowned leader in the technology world, adopted the Elastic Stack to improve its search functions across various applications. Given that over 87% of Fortune 500 companies depend on Cisco technology, the company understood the necessity of an efficient content search system.

By using Elasticsearch as the primary search and indexing tool, engineers can now rapidly locate pertinent documentation and address service requests, resulting in an impressive monthly time saving of 5,000 hours. The enhanced search functions have improved the customer experience and increased engagement through faster response times and accurate search results.

By combining the resilience, scalability, and AI capabilities of Elastic, Cisco has built a powerful search platform that is relevant to a wide audience across its enterprise. The company persists in innovating and expanding its search capabilities, capitalizing on the abundant features and versatility provided by Elastic.

[2] According to the case study of Airbus [1]

[3] According to the case study of Cisco Systems [2]

4.3 GitHub

GitHub[4] provides a clear demonstration of the advantages Elastic Search offers in terms of scalability. GitHub, being the biggest hosted version control system, utilizes Elasticsearch to drive its search function and provide a powerful search experience for its over 4 million technical users. Through Elasticsearch, GitHub has the capacity to index and query a vast amount of public data, which includes code repositories and other resources. The flexibility of Elasticsearch to store and retrieve both structured and loosely structured data makes it a perfect fit for GitHub's search system. It allows users to execute full-text searches or searches based on specific parameters, such as projects using a certain programming language and recent activity. Moreover, Elasticsearch offers profound programmatic search for developer applications, and it employs analytics to draw insights from search data. The performance, scalability, and sophisticated search features of Elasticsearch have established it as an invaluable tool in handling GitHub's massive code repository and supporting its large user base.

4.4 eBay

eBay[5], millions of sellers, 162 million active buyers, and 800 million listings. An already immense volume of data that continues to expand led the company to adopt Elasticsearch to handle all of its search functionalities. With the implementation of Elasticsearch, eBay has successfully confronted the challenges associated with processing and retrieving data on such an immense scale. Leveraging the advanced features and scalability of Elasticsearch, eBay now possesses a reliable and efficient solution to navigate its vast catalog of listings and deliver precise search results to its millions of users worldwide.

4.5 And Numerous Others

Elasticsearch's immense potential and capabilities are demonstrated through its use by prominent organizations such as Airbus, Cisco, GitHub, and eBay.

Airbus uses Elasticsearch in its ADNS platform to provide near real-time access to aircraft technical documents, benefiting both internal and external users. Cisco Systems relies on Elasticsearch to enhance search functions across various applications, resulting in significant time savings and improved customer experience. GitHub employs Elasticsearch to drive its search function, providing a powerful search experience for millions of technical users and handling its massive code repository. eBay, a global e-commerce giant, leverages Elasticsearch to optimize search and recommendation systems for customers, increasing user satisfaction and business growth.

These real-world examples illustrate the value and versatility of Elasticsearch in driving operations and achieving goals across a diverse range of industries.

[4] According to the case study of GitHub [3]

[5] According to the case study of eBay [4]

5 Conclusion

The rapid and relentless expansion of today's digital universe has created new challenges for data engineers and analysts. Traditional database management systems that have been the go-to for many years are now struggling to keep pace with the fast world we live in and the sheer amount of data produced each day by humanity. This has created a need for new means of collecting, storing, and analyzing data and Elasticsearch is one of the prime responses.
Elasticsearch NoSQL database management system excels in handling large volumes of unstructured or structured data. Its document-oriented design allows for high flexibility, scalability, and performance, making it a suitable choice for a wide range of applications.

This tutorial has explored some of the features that distinguish Elasticsearch, including its document-based model, sharding, REST API, full-text search, and its integral role in the Elastic Stack. The Elasticsearch document approach facilitates efficient storage and management of many different data types while sharding enables the possibility of fast horizontal scaling and high-speed data access. The REST API enables on this side seamless integration with various other technologies.

The Elastic Stack integration provides a complete data pipeline, from data ingestion to visualization, making Elasticsearch a comprehensive solution for data management and analysis. On top of that, the Full-text search is a precious feature as it permits the user to make some mistakes and still find an answer to their query.

However, we saw that this DBMS does not come without its own limitations. Its complexity, lack of built-in multilingual support, and suitability mainly for document-oriented applications are potential drawbacks that users must consider. Despite these limitations, some top leading companies across the globe still chose to use this type of database management system. This demonstrates the versatility and robustness of Elasticsearch.

Ultimately, having gained a comprehensive understanding of Elasticsearch, we encourage you to explore the accompanying Jupyter Notebook tutorial[6] that covers some of Elasticsearch's capabilities as well as deploys the complete Elastic stack, which is the next crucial aspect we recommend covering if you're looking to delve deeper into Elasticsearch.

This practical tutorial will guide you step-by-step through the process of setting up the Elastic Stack, establishing a connection to the Elasticsearch node, and performing essential CRUD operations while exploring data search capabilities.

[6] The tutorial is presented in a Jupyter Notebook. We recommend using JupyterLab or the Jupyter extension for VSCode to view the file.

References

[1] elastic. *Airbus ADNS: Powering the Search for Near Real-Time Access to Aircraft Technical Documents*. URL: https://www.elastic.co/customers/airbus (page 7).

[2] elastic. *Cisco chooses Elastic to power its enterprise search platform*. URL: https://www.elastic.co/customers/cisco (page 7).

[3] elastic. *GitHub: Accelerating software development*. URL: https://www.elastic.co/customers/github (page 8).

[4] elastic. *eBay and Elasticsearch: This is not small data*. URL: https://www.elastic.co/videos/ebay-and-elasticsearch-this-is-not-small-data (page 8).

[5] elastic. *Elastic customer stories of all shapes and sizes*. URL: https://www.elastic.co/customers/success-stories.

[6] ConaxInfotech. *What is ElasticSearch? Pros, Cons and Features List*. URL: https://conaxinfotech.com/blog/what-is-elasticsearch-pros-cons-and-features-list.html.

[7] Javatpoint. *Advantages and disadvantage of Elasticsearch*. URL: https://www.javatpoint.com/advantages-and-disadvantages-of-elasticsearch#:~:text=Elasticsearch%20is%20not%20a%20good,a%20bit%20difficult%20to%20learn..

[8] Apache Solr. *Apache Solr*. URL: https://solr.apache.org/.

[9] Elasticsearch. *Inverted Index*. URL: https://www.elastic.co/guide/en/elasticsearch/guide/current/inverted-index.html#inverted-index.

[10] Clinton Gormley and Zachary Tong. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

[11] Elasticsearch. *What is Elasticsearch?* URL: https://www.elastic.co/fr/what-is/elasticsearch.