

1 - Overview

Sunday, 2 February, 2025 11:28 Manhã

Baseado em Apache Lucene (Java)

Relevância da busca é importante para um negócio

Dados estruturados (tem schema):

- Resultados de busca são binários - sim ou não.
- Filtros são precisos.
- Não tem score de relevância.

Dados não-estruturados (e.g. texto):

- Sem *schema*
- Pode ser semi-estruturado (i.e. texto com alguns metadados)
- Elastic é bom nisso - procura termos iguais e parecidos.

Elastic consegue unir busca de estruturado com não-estruturado (Não é um banco de dados relacional)

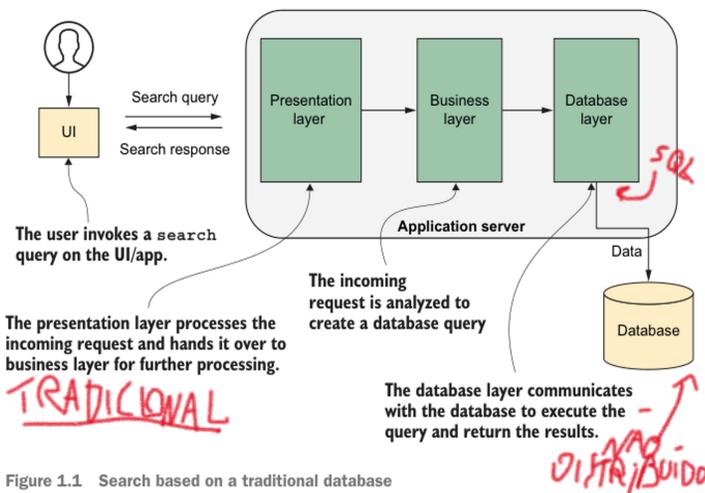


Figure 1.1 Search based on a traditional database

Modelo tradicional com SQL é pouco eficiente para texto não-estruturado.

São melhor adaptados a dados estruturados.

Devagar mesmo quando suporta.

Ainda assim pode ser uma boa solução dependendo do contexto.

- Bancos de dados comuns não conseguem indexar grandes quantidades de dados eficientemente.
- Não costumam ter recursos de busca como fuzzy logic, sinônimos, etc.
- Busca de texto não pode ser por matches exatos apenas.
- Normalização de bancos de dados torna difícil buscar em vários campos diferentes de uma vez.

Um bom buscador de texto deve:

- Ter suporte para dado estruturado e não-estruturado
- Ser flexível com o user input
- Auto-correct
- Escalável, rápido, confiável, sempre-disponível
- Poder buscar geolocalização (?)
- Permitir machine-learning

CHAT GPT

Fazer buscável = quebrar texto em tokens e mapear cada token ao documento completo.

Solr e **OpenSearch** são alternativas ao **ElasticSearch**, todos vindos do Lucene.

Solr costuma ser usado em datasets grandes e estáticos.

ElasticSearch já vem com features de análise, monitoramento, segurança e machine learning.

Ecosistema Elastic:

What does the book mean by "Search capabilities on geolocations, when talking about ElasticSearch?"

When the book refers to "Search capabilities on geolocations" in the context of **Elasticsearch**, it means that Elasticsearch provides built-in support for **geospatial search**, allowing users to store, index, and query data based on geographic locations (latitude/longitude coordinates). This functionality is useful for applications that need location-based searching, filtering, and ranking.

Key Geolocation Features in Elasticsearch:

- Geo-Point Data Type**
 - Elasticsearch allows you to store location-based data as a **geo_point** field type, which represents latitude and longitude.
- Geo Distance Queries**
 - Users can search for documents within a specified radius from a given point (e.g., "Find all restaurants within 10 miles of my location").
- Bounding Box Queries**
 - Searches can be restricted within a rectangular area (bounding box), which is useful for map-based applications.
- Geo-Shape Queries**
 - Elasticsearch supports **geo_shapes** (polygons, lines, etc.), allowing complex spatial queries (e.g., "Find all delivery zones that cover a specific address").
- Sorting and Ranking by Distance**
 - Results can be sorted by proximity to a given point, which is useful for "nearest location" searches.

Example Use Case:

A ride-hailing app like Uber or a food delivery service like DoorDash can use Elasticsearch's **geo capabilities** to:

- Find the closest drivers to a passenger.
- Locate restaurants within a delivery range.
- Show users businesses near their current location.

Would you like a specific Elasticsearch query example for geolocation search?

🔍 📄 🔄 🗑️ 📌



Elastic Enterprise Search
Workplace, website, and app search



Elastic Observability
Unified logs, metrics, and APM data



Elastic Security
SIEM, endpoint, and threat hunting

Stack Elastic:

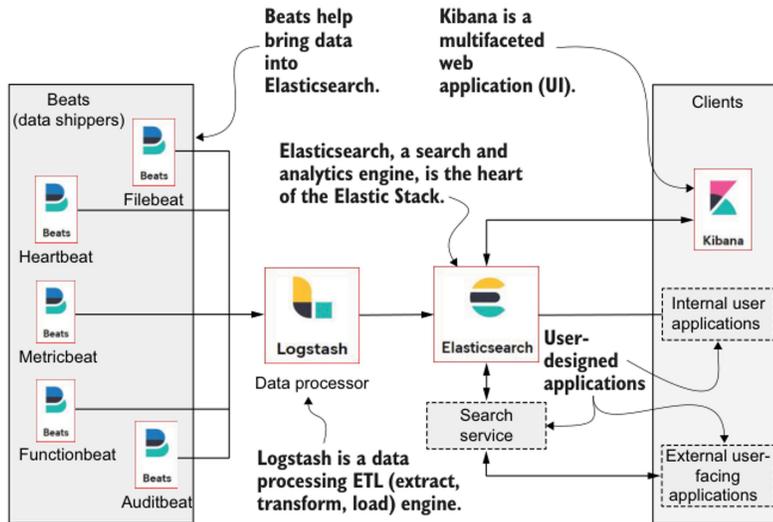


Figure 1.3 The Elastic Stack: Beats, Logstash, Elasticsearch, and Kibana

Beat: carrega dado vindo de algum lugar, consome dados

Logstash: Transforma e transfere dados. Pipeline de ingestão de dados.

Kibana: visualizar e interagir com dados.

Elastic Search é usado em várias aplicações:

- Como busca, três escopos:
 - **App Search** (busca dados internos para uma aplicação)
 - **Enterprise Search** (busca intra organização)
 - **Site Search** (busca em um website)
- Business Analytics
- Security Analytics
- Logging & Monitoramento

É usado por várias empresas grandes como Uber e Netflix.

Elastic Search **NÃO** deve ser usado para:

- Busca em dados relacionais
- Dados transacionais
- Dados geoespaciais em alta escala
- Muitos writes, é bom apenas para muitos reads.
- OLAP análise
- Muitos arquivos binários como imagem e vídeo
- Analytics em tempo real
- Baixa latência
- Séries temporais, grafos, etc.

Não substitui todas as outras soluções de busca.

Gerenciamento do Elastic não é simples

LLMs como ChatGPT têm sido um concorrente de buscadores de texto.

Trazem potencial de tornar as buscas mais intuitivas, personalizadas e preditivas.

Summary

- Search is the new normal and the most sought-after functionality for organizations, enabling competitive advantage.
- Search engines built using relational databases as the backend used to serve our search purposes but can't fulfill the full-fledged search functionality found in modern search engines.
- Modern search engines provide multifaceted, full-text search capabilities and multifold benefits from basic search to advanced search and analytical functions, all with split-second performance. They are also expected to handle terabytes to petabytes of data and scale if needed.
- Elasticsearch is an open source search and analytics engine built over Apache Lucene. It is a highly available server-side application developed in Java.
- Because Elasticsearch was designed as a programming-language-agnostic product, communication with the server takes place over HTTP using rich RESTful APIs. These APIs receive and send data in JSON format.
- The Elastic Stack is a suite of products composed of Beats, Logstash, Elasticsearch, and Kibana. Beats are single-purpose data shippers, Logstash is a data-processing ETL (extract, transform, load) engine, Kibana is the administrative UI tool, and Elasticsearch is the heart and soul of the stack.
- The Elastic Stack enables an organization to position itself in three core areas: search, observability, and security.
- Elasticsearch has become popular over the last few years due to its structured/unstructured search and analytics capabilities; rich set of RESTful APIs; its schema-free nature; and performance, high-availability, and scalability characteristics.
- AI-powered search is here. With the advent of generative AI and ChatGPT, the search space will be explored further, and search will become more intuitive and predictive.