

Tech Review: Apache Lucene vs Elasticsearch

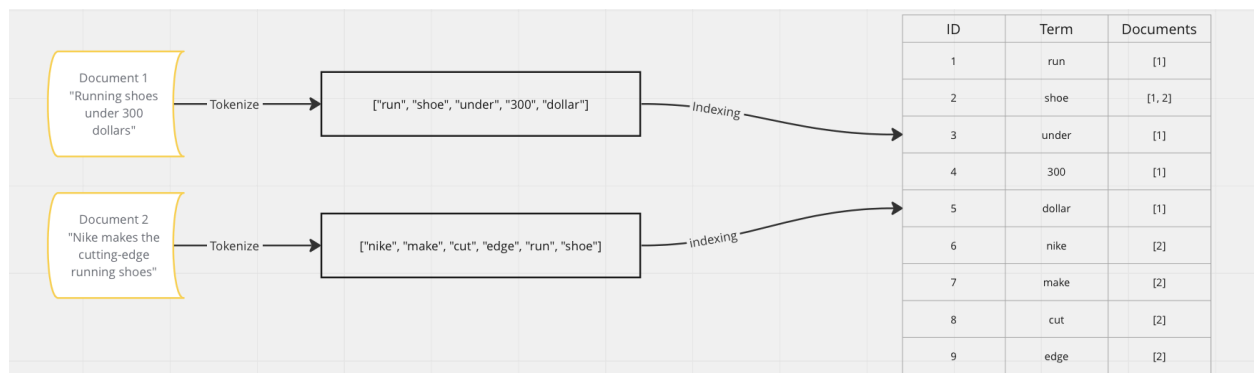
By Xinlei Huang, xhuang84@illinois.edu, Nov 3, 2022

Introduction

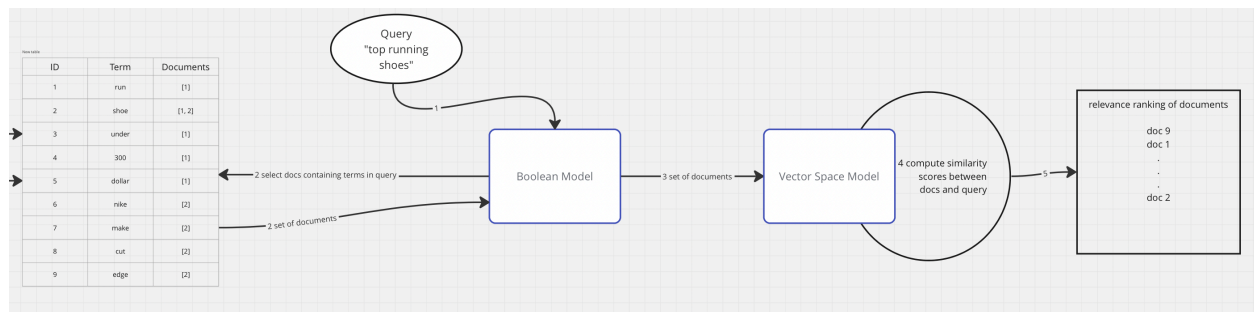
Elasticsearch is one of the cutting-edge open-source searching technologies. The fundamental methodology is built upon Apache Lucene which leverages inverted indexing to accelerate searching, but Elasticsearch is more than a shell of Apache Lucene. It provides scalability and resilience like other industry level distributed systems. This review will illustrate the similarities and variances between them.

Apache Lucene Methodology

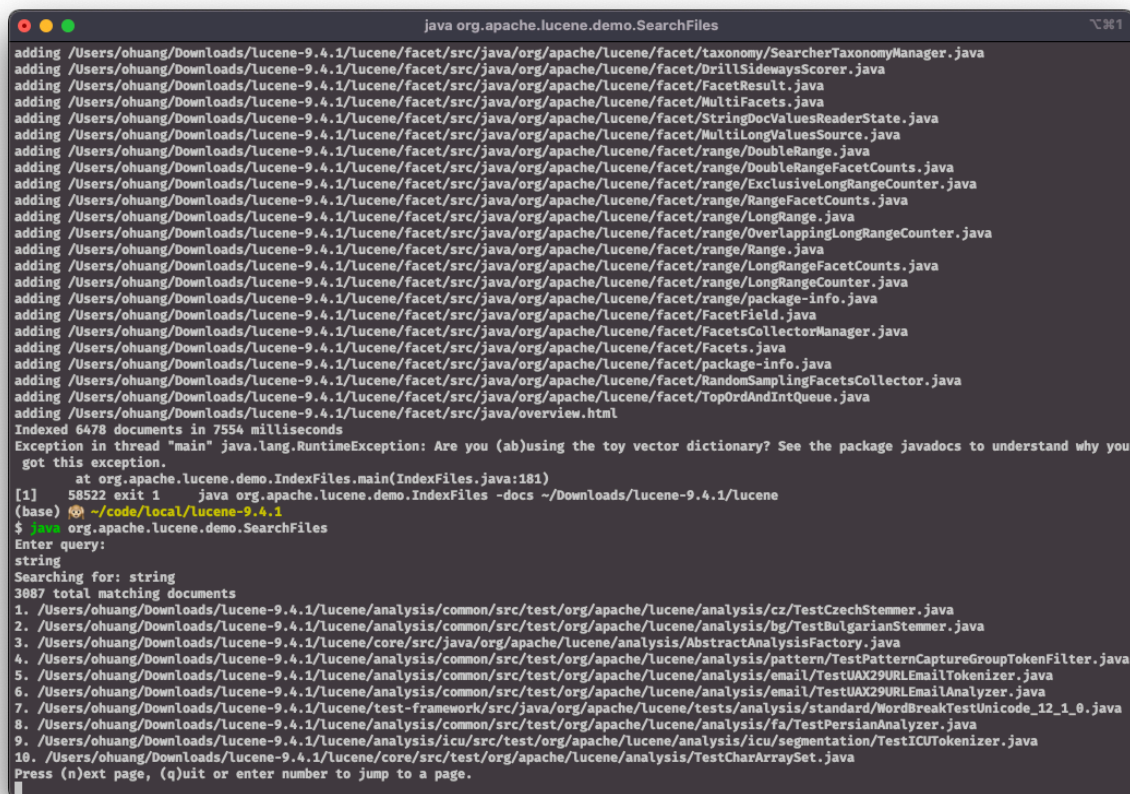
Apache Lucene is a high-performance, full-featured text search engine library leveraging inverted indexing. The oversimplified indexing process can be described as following diagram



After indices are generated, it's ready to perform text search with a query. Lucene scores documents with Boolean model and Vector Space Model to filter out ones relevant to the query. Overview of this scoring process can be found in following diagram



Lucene adopts Boolean model to assemble a set of documents which contains terms in the searching query via indices built previously. Secondly, it will compute the similarity between each document and query string with Vector Space Model. The similarity computing algorithm is configurable, for instance Okapi BM25 and classic TF-IDF cosine similarity. One thing worth mentioning is Lucene actually builds its SimilarityBase scoring algorithm upon one of Professor's Zhai's papers, *A Study of Smoothing Methods for Language Models Applied to Information Retrieval*, 2004. To better illustrate the indexing and searching process, a small demo was conducted with Apache Lucene Java distribution.

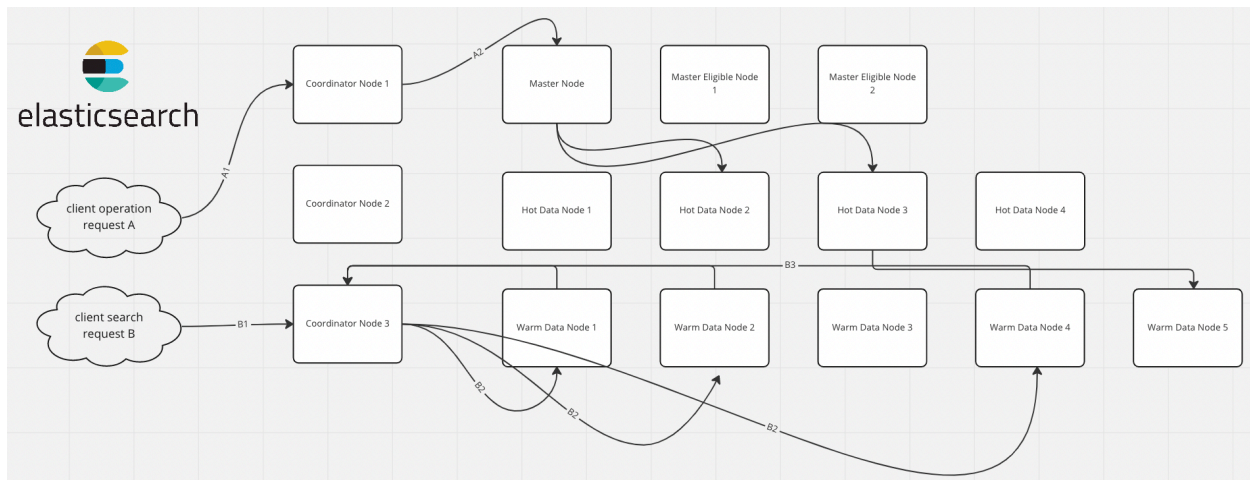


```
java org.apache.lucene.demo.SearchFiles
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/taxonomy/SearcherTaxonomyManager.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/DrillSidewaysScorer.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/FacetResult.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/MultiFacets.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/StringDocValuesReaderState.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/MultiLongValuesSource.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/DoubleRange.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/DoubleRangeFacetCounts.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/ExclusiveLongRangeCounter.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/RangeFacetCounts.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/LongRange.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/OverlappingLongRangeCounter.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/Range.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/LongRangeFacetCounts.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/LongRangeCounter.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/range/package-info.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/FacetField.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/FacetsCollectorManager.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/Facets.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/package-info.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/RandomSamplingFacetsCollector.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/org/apache/lucene/facet/TopOrdAndIntQueue.java
adding /Users/ohuang/Downloads/lucene-9.4.1/lucene/facet/src/java/overview.html
Indexed 6478 documents in 7554 milliseconds
Exception in thread "main" java.lang.RuntimeException: Are you (ab)using the toy vector dictionary? See the package javadocs to understand why you
got this exception.
    at org.apache.lucene.demo.IndexFiles.main(IndexFiles.java:181)
[1] 58522 exit 1      java org.apache.lucene.demo.IndexFiles -docs ~/Downloads/lucene-9.4.1/lucene
(base) ~ % cd ~/code/local/lucene-9.4.1
$ java org.apache.lucene.demo.SearchFiles
Enter query:
string
Searching for: string
3887 total matching documents
1. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/common/src/test/org/apache/lucene/analysis/cz/TestCzechStemmer.java
2. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/common/src/test/org/apache/lucene/analysis/bg/TestBulgarianStemmer.java
3. /Users/ohuang/Downloads/lucene-9.4.1/lucene/core/src/java/org/apache/lucene/analysis/AbstractAnalysisFactory.java
4. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/common/src/test/org/apache/lucene/analysis/pattern/TestPatternCaptureGroupTokenFilter.java
5. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/common/src/test/org/apache/lucene/analysis/email/TestUAX29URLEmailTokenizer.java
6. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/common/src/test/org/apache/lucene/analysis/email/TestUAX29URLEmailAnalyzer.java
7. /Users/ohuang/Downloads/lucene-9.4.1/lucene/test-framework/src/java/org/apache/lucene/tests/analysis/standard/WordBreakTestUnicode_12_1_0.java
8. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/common/src/test/org/apache/lucene/analysis/fa/TestPersianAnalyzer.java
9. /Users/ohuang/Downloads/lucene-9.4.1/lucene/analysis/icu/src/test/org/apache/lucene/analysis/icu/segmentation/TestICUTokenizer.java
10. /Users/ohuang/Downloads/lucene-9.4.1/lucene/core/src/test/org/apache/lucene/analysis/TestCharArraySet.java
Press (n)ext page, (q)uit or enter number to jump to a page.
```

Elasticsearch

Elasticsearch brands itself as an enterprise search solution. Deep down it's a distributed system where each node is running an Apache Lucene instance. Indexing and scoring methodologies are generally similar, thus this section will focus on how Elasticsearch achieves high scalability and resilience. It adopts a common architecture of distributed system, splitting tasks into multiple subtasks of different category such as load balancing, computation, and data storage. Each subtask is handled by a specific type of worker nodes.

Master nodes control the cluster. It over-watches the health of other nodes to make sure availability of services and data integrity. Coordinator nodes handles searching query by distributing indexing and searching tasks to data nodes containing relevant terms and then aggregating results sent back from data nodes. Data nodes hold indexed data and perform



data related operations. There are two types of data nodes. Hot nodes usually run on machines equipped with SSD handling write/read tasks, while warm nodes reside on machines with spin disks for read-heavy tasks. On data storage perspective, Elasticsearch splits indices into multiple shards, and create replica shards at the mean time. If any primary shards were corrupted, replicas would become primary shards and open write access to worker nodes. This will mitigate data loss caused by hardware failure significantly.

Summary

From methodology perspective, Apache Lucene and Elasticsearch share notable similarities. Both use inverted indexing and language models for scoring, however, Elasticsearch extends Lucene with distributed architecture to support large scale searching capacity. It doesn't mean Lucene cannot satisfy industry level usage. Evernote, Slack, and Twitter all build their searching services around Apache Lucene with architecture tweaks.

References

Apache Lucene - scoring. (n.d.). Retrieved November 2, 2022, from https://lucene.apache.org/core/3_5_0/scoring.html

Lucene 9.1.0 demo api. Overview (Lucene 9.1.0 demo API). (2022, March 22). Retrieved November 3, 2022, from https://lucene.apache.org/core/9_1_0/demo/index.html

Package org.apache.lucene.search.similarities. org.apache.lucene.search.similarities (Lucene 8.2.0 API). (2019, July 25). Retrieved November 3, 2022, from https://lucene.apache.org/core/8_2_0/core/org/apache/lucene/search/similarities/package-summary.html

Free and open search: The creators of Elasticsearch, Elk & Kibana. Elastic. (n.d.). Retrieved November 4, 2022, from <https://www.elastic.co/>