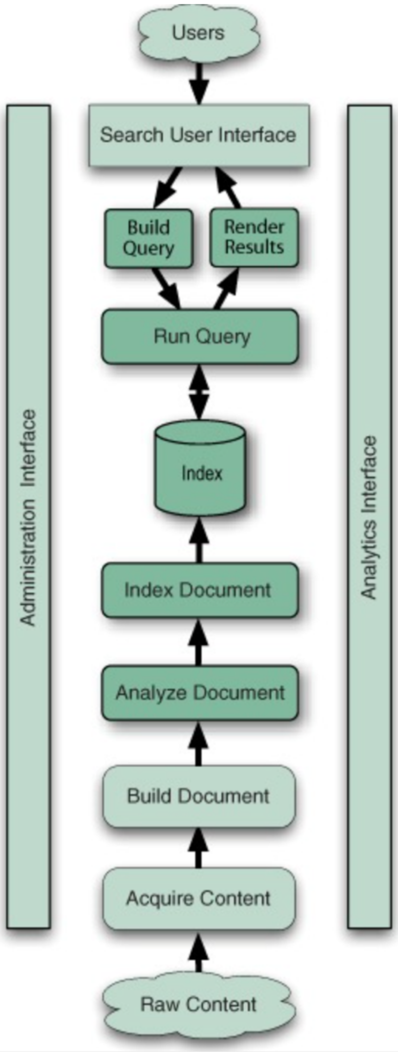
Apache Lucene

Apache Lucene is an open-source software library that is full featured, high-performance, scalable, cross platform data retrieval engine with powerful data retrieval capabilities. Lucene comes with powerful, efficient and accurate search algorithm along with plug-able ranking models including vector space model and Okapi BM25.
It is a software library and not a search application. It concerns itself with text indexing and searching, and it does those things very well. Lucene lets your application deal with business rules specific to its problem domain while hiding the complexity of indexing and searching behind a simple-to-use API. Lucene is the core that the application wraps around.

Components of search application:



1. Content: First step is to acquire raw content. The content can come from various sources such as database, crawling web sites, documents stored in a directory. The content here has to be incremental for efficient processing. Gathering the content itself is not something Lucene supports.
2. Build document: This step involves extracting the raw content into units called documents with searchable fields. Here consumers need to design which fields should be made searchable and what other content should the search engine bring back along with the search results. One can apply additional fields by running semantic analyzer to pull out fields such as proper names, location, date, times etc. If the content resides in database, additional integration projects can be implemented to build the document from the data base content.
3. Analyze: The document built in previous step is broken down into atomic units. Lucene provides an array of built-in analyzers that give control over fine grain process.
4. Index: During this step, document is added to the index. Lucene allows you to customize the indexing process. One must carefully design this process such that it allows for improving the search user experience. Indexes are stored on the disk to allow user to run queries.
5. Build queries: Lucene provides a powerful package called query parser that process the user text into a query object according to the common search syntax. The package also allows one to set up the filters on the query to restrict the search on the documents that the user has access.
6. Run Query: The process of consulting the search index and retrieving the documents matching the query, sorted in the requested sort order. Lucene is highly customizable and allows user to specify how the search query is executed, sorted, results are gathered. It supports the following search models:
    a. Boolean model: Documents either match or don't match the search query, retrieved documents are unordered. It is simplest form of search.
    b. Vector space model: Queries and documents are modeled as vectors in high dimensional space. Relevance and similarity is calculated by the magnitude of these vectors.
    c. Probabilistic model: Probability that the document is a good match for the query is computed in this model. Documents are retrieved based on sorted probability.

Lucene provides a number of configuration options such as tuning the size of RAM used during indexing, how many segments to merge at once, how often to commit changes, or when to optimize and purge deletes from the index.

Lucene provides usage analytics that allows one to gain insight into variety of queries executed by the users. It helps you answer the following questions:

1. How often which kinds of queries (single term, phrase, Boolean queries, etc.) are run
2. Queries that hit low relevance
3. Queries where the user didn't click on any results (if your application tracks click-throughs)
4. How often users are sorting by specified fields instead of relevance

5. The breakdown of Lucene’s search time
6. Documents indexed per minute or second

The above analytics is useful in implementing further improvements into your implementation.

Scaling:
Lucene indexing and searching throughput allows for a sizable amount of content on a single modern computer. However, it does not provide a way to scaling for high availability.

Overall, Lucene comes with powerful libraries that allows you to integrate with or build along with other open-source software such as Solr making search functionality with the application much more robust and performant. In addition, you can choose among numerous ways to access Lucene’s functionality from programming environments other than Java.

Work Cited

https://livebook.manning.com/book/lucene-in-action-second-edition/about-this-book/