Document Search Engine

Deepak Gupta, Rohan Chandavarkar, Laxmikant Kishor Mokadam

North Carolina State University

1. Definition

The project 'Document Search Engine' is a platform which aims to return relevant documents for a given input query. It is a distributed data-intensive computing platform which quickly returns related documents from the dataset while balancing the load at the back end. The platform also updates the dataset in real time.

2. Justification

This project gives us an opportunity to deal with large real-time data in a distributed environment. It will help us learn how to efficiently compute large volumes of data in the minimum possible time.

We will also learn to tackle load balancing of large data to maintain efficiency irrespective of data size. The project deals with load balancing in real time when new data is added to the database.

This project is scalable enough to include the three Vs of data-intensive computing. The large dataset of documents ensures high volume. We will deal with a variety of documents like pdf, text, and images. The velocity of data is generated through automated scripts sending queries from client to system. While this high-velocity data is handled at the same time the load is balanced during the real-time updation of the database.

Along with the use cases, the project is ideal as it will give us an exposure to technologies like Apache Spark for distributed computing and Apache Streaming for load balancing.

3. Overview

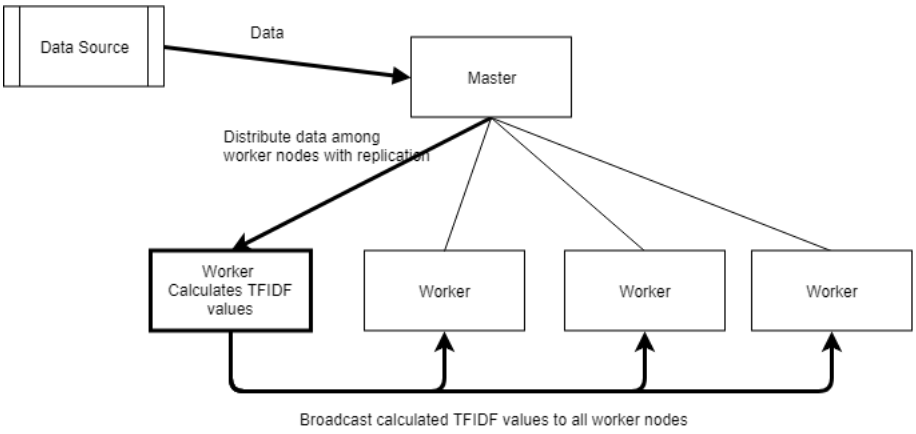
The document dataset will be distributed among workers. Apache Spark will be used for distributed computing. Each worker will then use Indexing method like TF-IDF or LSI on each document to extract the most important words representing the document. Further, all workers will then share the results with each other in order to get a cumulative metadata. This metadata will then be used by workers to return the result of queries.

The load balancing of the queries will be done by using Apache Streaming. This will ensure that all workers get an equivalent number of queries to handle. The worker will then return the result directly to the client.

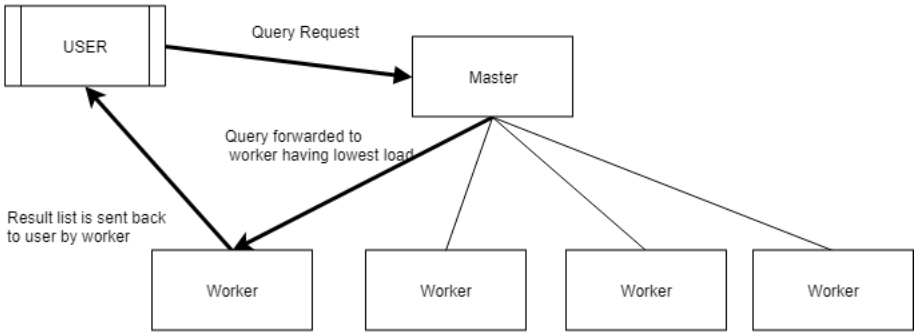
Further, the real-time updation of the document database will be implemented. This will also be implemented by keeping load balancing in mind. When a new document data set is assigned to a worker, no queries will be assigned to it. Once it completes calculation of metadata, it will resume its normal handling of queries. After every fixed interval, these workers will broadcast and share the results with each other. The system will be further enhanced to handle a variety of datasets like pdfs and images.

The project will be considered complete if relevant documents are returned for each query, with the huge number of searches queried at high velocity.

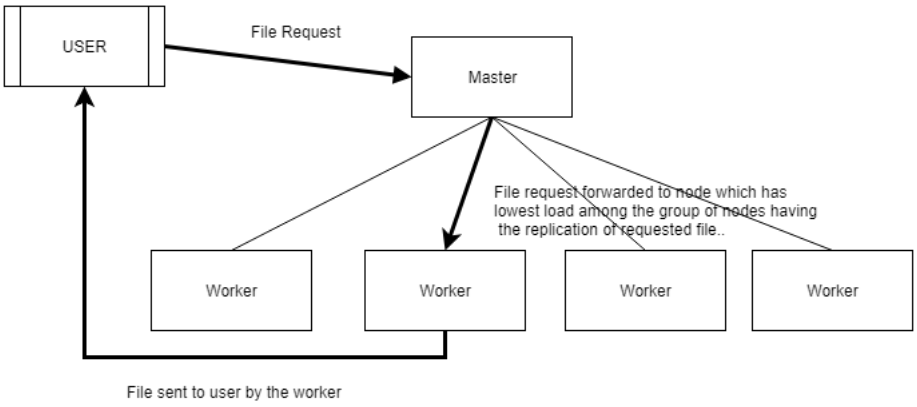
4. Architecture



Data Distribution and TFIDF Calculation



Query Workflow



File Access Workflow

5. Timeline

Sep 28 - Connection setup between Client-Master and Master-Workers. Master must be able to distribute files among workers and Client must be able to send query to the Master.

Oct 12 - Distributed TFIDF Implementation

Oct 26 - Query workflow implementation

Nov 9 - File Access workflow

Nov 23 - Runtime database updation with load balancing

Nov 30 - Testing with 3 Vs and documenting results

6. Resources

We will require a minimum of 8 VCL images running simultaneously.

Dataset : We will use documents in text and pdf form taken from various sources like Wikipedia, Kaggle etc.