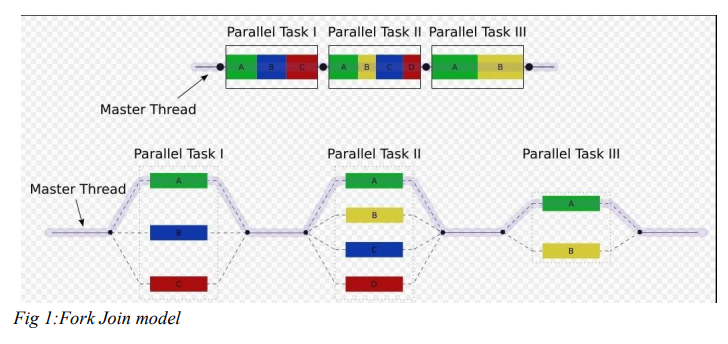
**PARALLEL WORD SEARCH USING OPENMP**

------Shubham Kumar

**INTRODUCTION:**

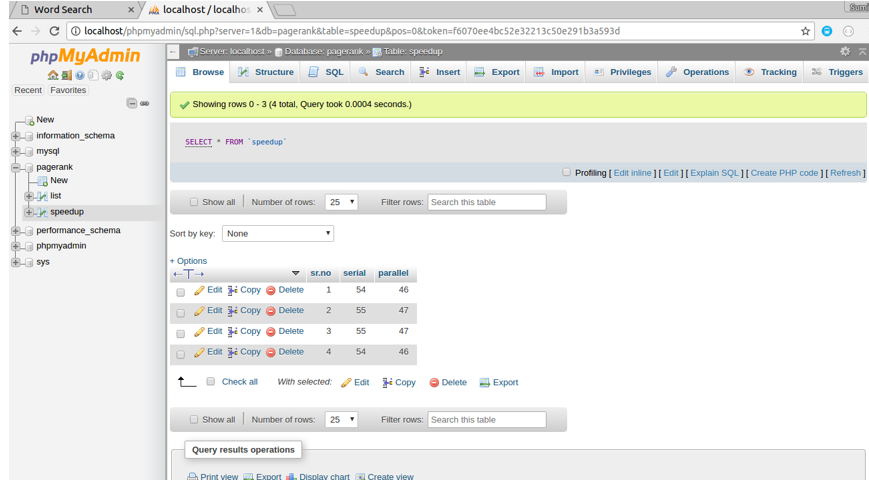
Word search searches for words in multiple text files parallely using openmp concepts.This reduces the execution time of the code as compared to the sequential word search.The documents are scanned word by word and a dictionary is maintained so that words can be searched. Since documents can have several thousands of terms scanning each term can take a lot of time and hence there is a lot of scope for parallelization. KEYWORDS: OpenMP, Parallelization, Multithreaded, Wordsearch, Sequential



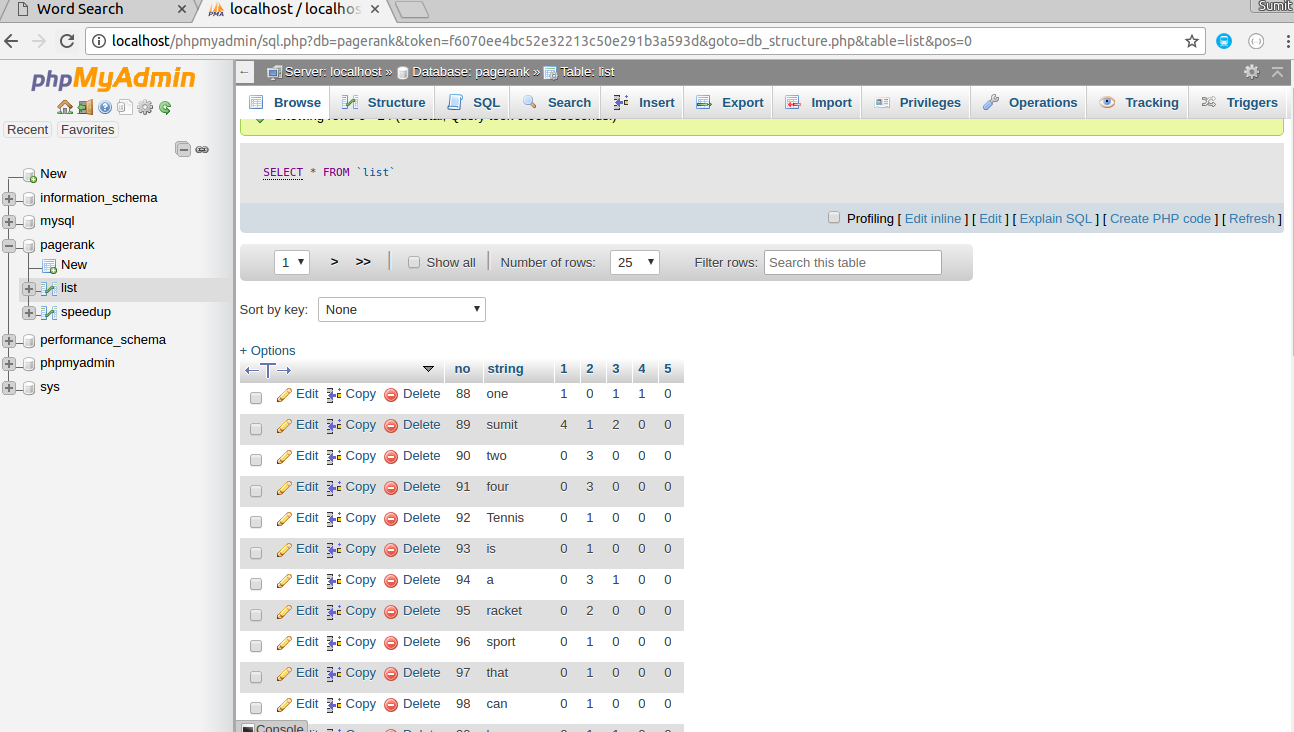
## **Programming Languages and Web Technologies Used**:

C++ (using Standard Template Library) , Openmp (for parallelization of the code), PHP , HTML ,CSS , Javascript ( Jquery) , Bootstrap , Mysql(Database).

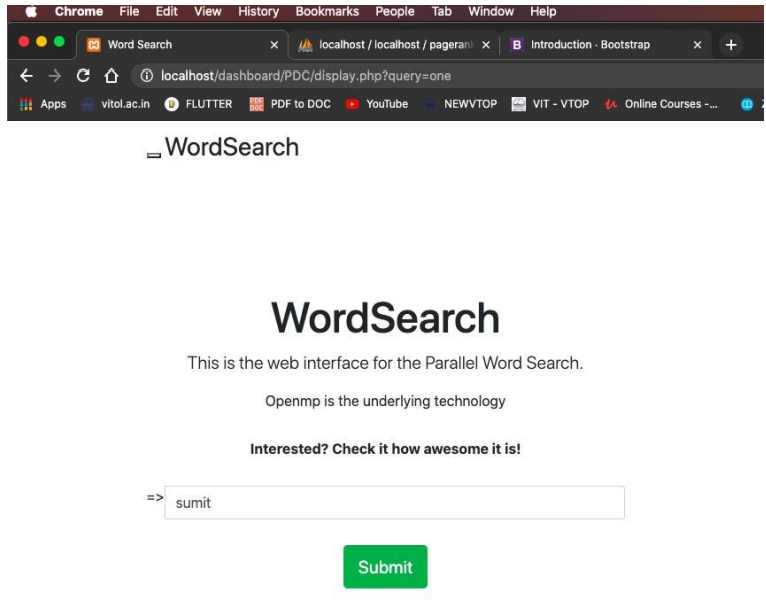
Following table shows the serial Vs Parallel time of execution.

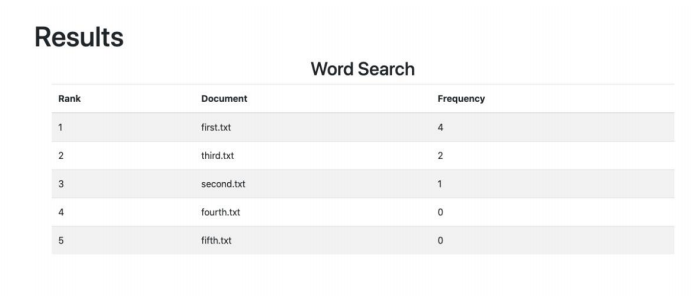


Following diagram shows the dictionary which is made as result of running the code in mysql database.



Output Snapshots:





**Instructions:**

The basic code is written in C++ which uses openmp to parallelize the code.

1. The documents are located in a folder and file names are given as some collection

2. The files are taken in a parallel way and all the data is collectively stored in a vector

3. After this the vectors are sent to a search method that parallizes the search and concurrently browses through the working documents

4. After this, the documents are ranked based on highest word searches.

5. A mapping is done and the text files are displayed accordingly based on those with highest occurrence.

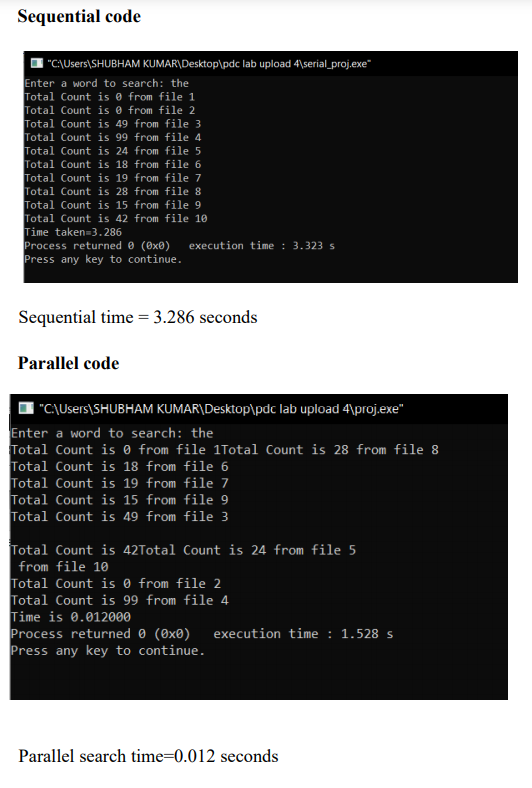
## **Application**:

Word search is used everywhere from local page search ( Cntrl + F) to searching words on document viewer like “reader” in windows. In fact a whole branch called Information Retrieval was developed for this. This project was actually inspired by Information Retrieval. It has a lot of application in real word.

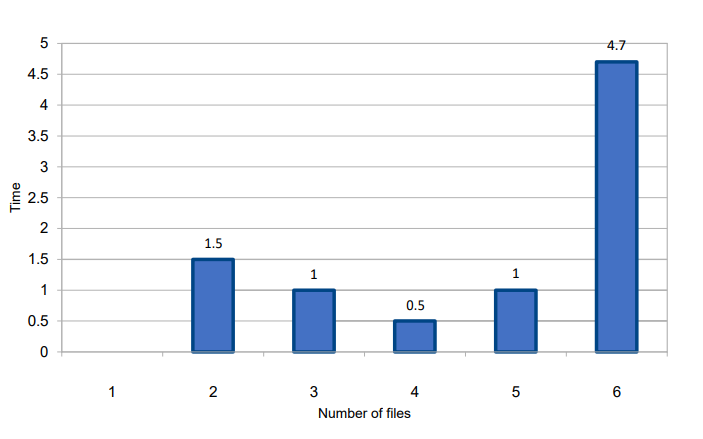
**Note**:

Following files are the part of the project

1. pagerank.cpp (Parallel Code) //connected to php file
2. pagerank2.cpp (Serial Code) //connected to php file
3. insert.php
4. config.php
5. display.php
6. File1.txt (Input file) //create in the local device
7. File2.txt (Input file) //create in the local device
8. File3.txt (Input file) //create in the local device
9. File4.txt (Input file) //create in the local device
10. File5.txt(Input file) //create in the local device
11. proj (Sample code for Parallel searching) //just run in the C++ compiler to get the time
12. serial\_proj (Sample code for Serial searching) //just run in the C++ compiler to get the time



The graph can be plotted using ‘plotly’ function by collecting the data configured while running the serial and parallel codes simultaneously in a C++ compiler.



The following data is displayed from manual observations.

