

Assignment 2: Part 2

Medha Kant, Namrata Priyadarshini

In this assignment we were supposed to find the pagerank of different pages given the input, a page and which other pages it references to. We first calculated the transpose for the links and then also found out the leaf nodes. The block of code that we wish to apply map reduce to is:

```
for(int j=0;j<n;j++){
    vector<int> page = incominglinks[j];
    double dot = 0;
    for(int k=0;k<page.size();k++){
        dot += iold[page[k]]/numlinks[page[k]];
    }
    pgr_intermediate[j] = dot*alpha + oneav + oneiv;
}
```

As we can see, we can use j values as keys and intermediate pagerank values as the values for our map reduce algorithm.

Implementation of the Pagerank mechanism

1. Using MAPREDUCE LIBRARY.
2. Implementing our own mapreduce using MPI
3. MPIMRE LIBRARY

Compiling and running the program

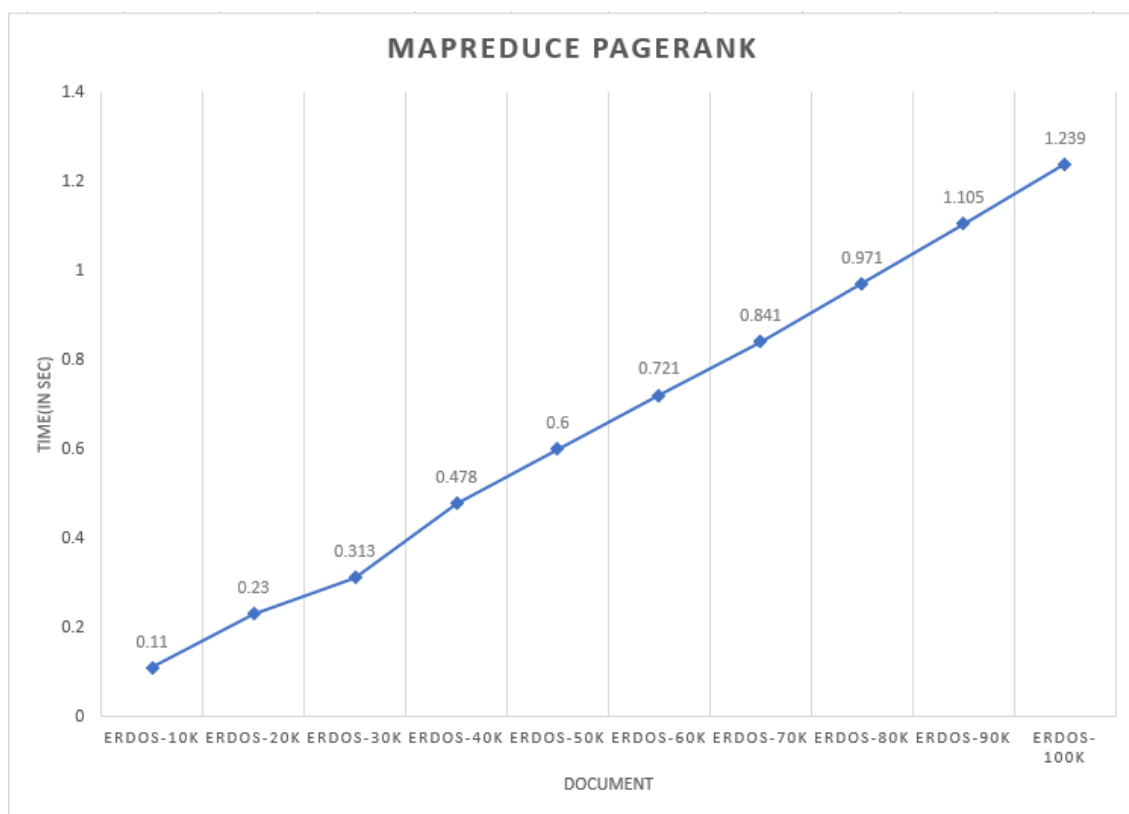
To compile a pagerank algorithm, go to the directory and run the makefile and it would produce an executable with the name- mr-pr-cpp.o and pr-cpp.o for the code that doesn't use map reduce. To run the non-MPI version, type
`./mr-pr-cpp.o $infile -o $outfile`
 and for the MPI version, run `mpirun -np 1 ./mr-pr-cpp.o $infile -o $outfile`.
 The code in the difference folder can be used to calculate the difference generated from two pagerank algorithms.

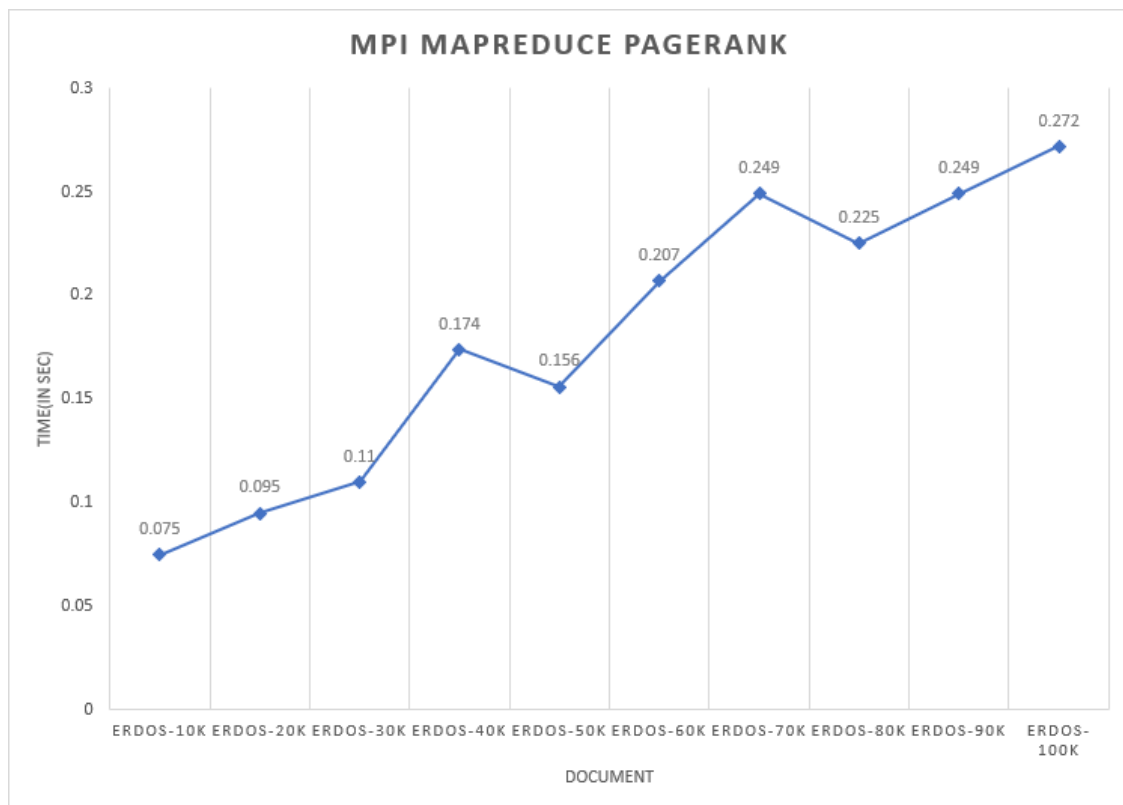
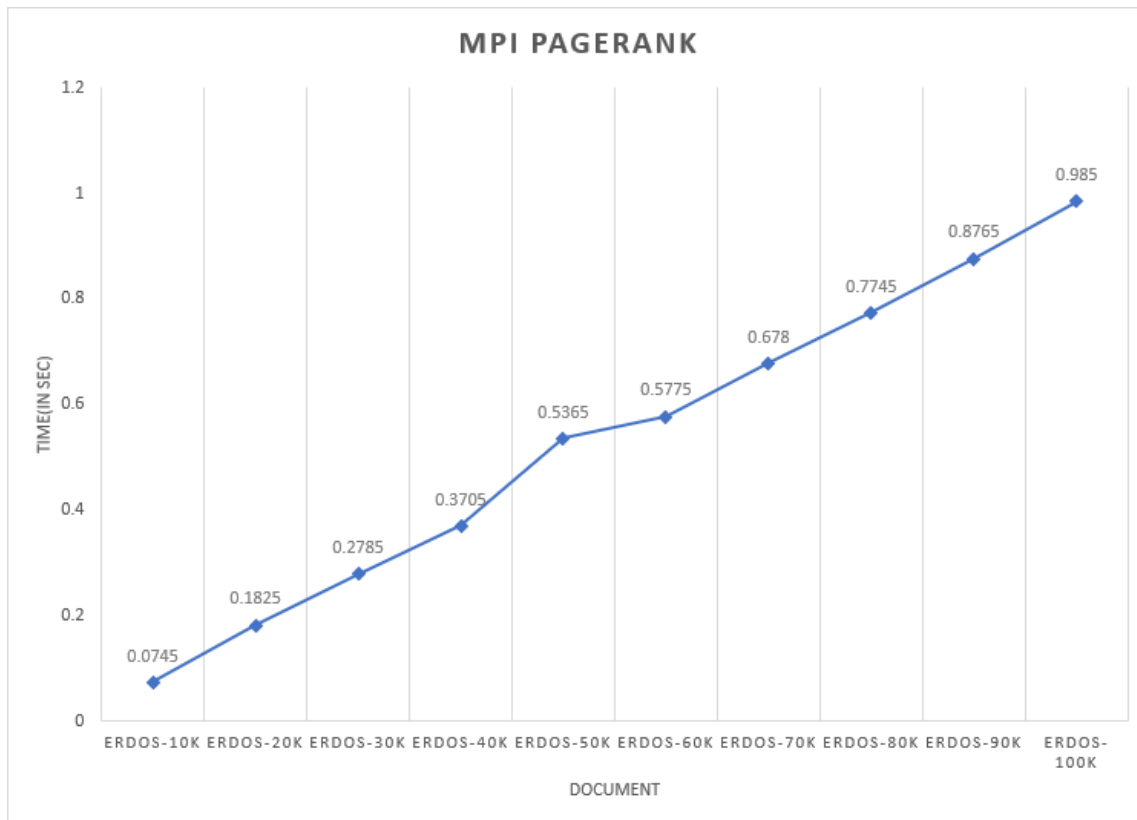
- MAPREDUCE > The provided map reduce library
- MPIMRE > MPI library implementation of map reduce
- MPI > Map Reduce functions using MPI Library
- PAGERANK > A simple implementation of Pagerank without map reduce
- DIFF > A file to calculate difference between 2 generated outputs

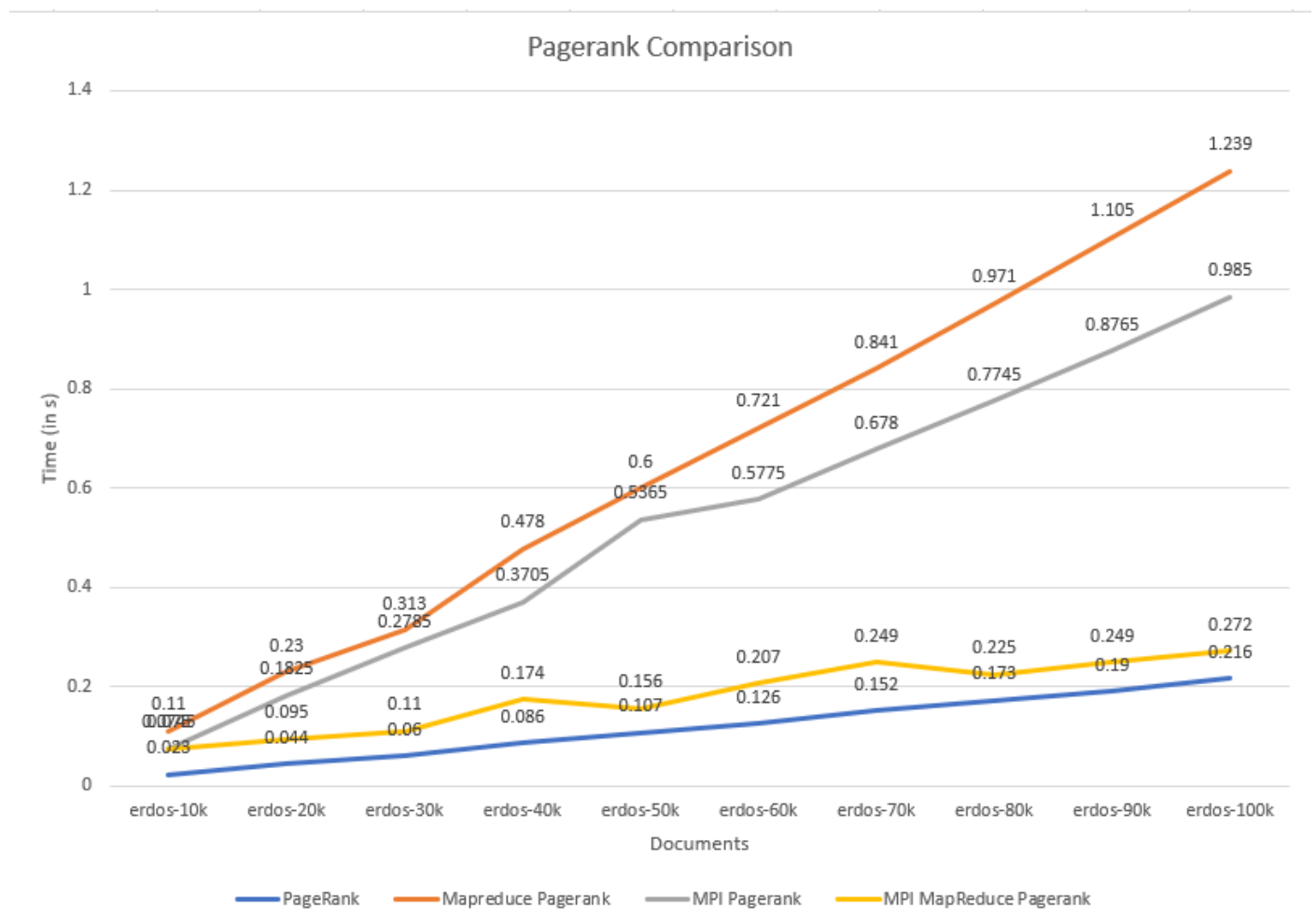
Plots and Results

The execution time of pagerank for different approaches is –

| <u>Document</u> | <u>PageRank</u> | <u>Mapreduce Pagerank</u> | <u>MPI Pagerank</u> | <u>MPI MapReduce Pagerank</u> |
|------------------------|------------------------|----------------------------------|----------------------------|--------------------------------------|
| erdos-10k | 0.023 | 0.11 | 0.0745 | 0.075 |
| erdos-20k | 0.044 | 0.23 | 0.1825 | 0.095 |
| erdos-30k | 0.06 | 0.313 | 0.2785 | 0.11 |
| erdos-40k | 0.086 | 0.478 | 0.3705 | 0.174 |
| erdos-50k | 0.107 | 0.6 | 0.5365 | 0.156 |
| erdos-60k | 0.126 | 0.721 | 0.5775 | 0.207 |
| erdos-70k | 0.152 | 0.841 | 0.678 | 0.249 |
| erdos-80k | 0.173 | 0.971 | 0.7745 | 0.225 |
| erdos-90k | 0.19 | 1.105 | 0.8765 | 0.249 |
| erdos-100k | 0.216 | 1.239 | 0.985 | 0.272 |







Plot comparing the runtime for different algorithms on the same documents.

Conclusion

Map Reduce is a really fast algorithm that can boost the speed of calculation of PageRank and other cases where it can be applied successfully.