

Ecenaz Ozmen (eo2419)

Yefri Gaitan (yg2548)

PFP Project Proposal: MapReduce to Find Mutual Friends

We would like to implement a parallel MapReduce library and test it on social network data to identify mutual friends given 2 people. For the library, we will include a sequential and parallel implementations for map and reduce functions. We're thinking of making this library generic so we could use it for multiple applications, eg. word counter, identifying mutual friends. For the parallel implementation, we will run both the map and reduce functions concurrently as they are called by the user.

For testing purposes and implementing the MapReduce framework in a social network context, we plan to use data we found [online](#) that includes a person and a list of their friends represented by ids. The computationally expensive part of identifying mutual friends is that every friend has to be compared to each other and the computation becomes exponential. This is why being able to run this in parallel with the use of the map reduce framework makes this a perfect test case for our library.

The steps to identifying mutual friends include, first, reading and parsing the file, and calling the map function on every line of the input file (formatted as PersonID tab and comma separated list of friends). The MapReduce library will save the intermediate steps to disk in case the file is too long and we run out of space to store the intermediate values. These intermediate values will be fed into the reduce function. Then, the reduce function should give a list of mappings for every pair of friends, a list of mutual friends they have. The result can also be written to disk.