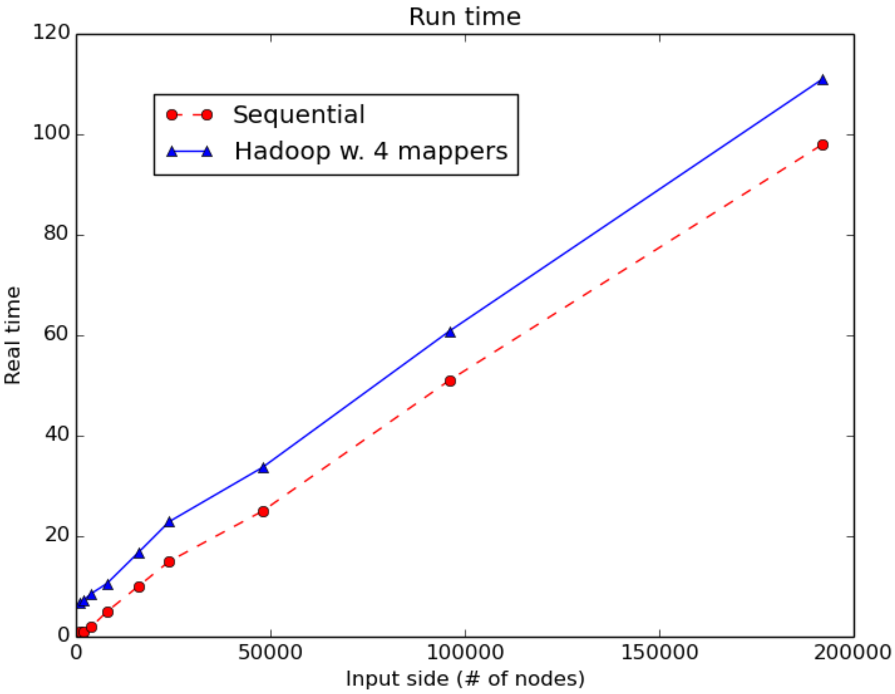
CSC494 Spring 2015 final report

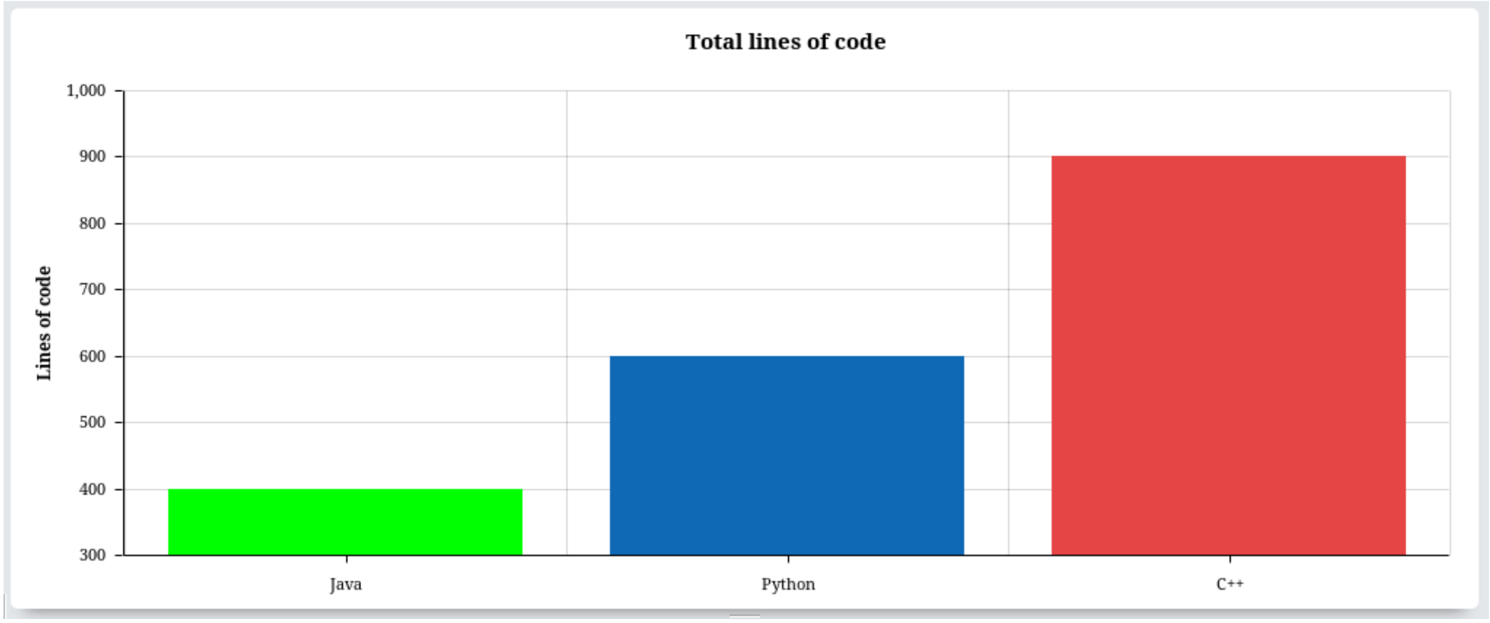
Supervisors: Renée J. Miller, Fatemeh Nargesian

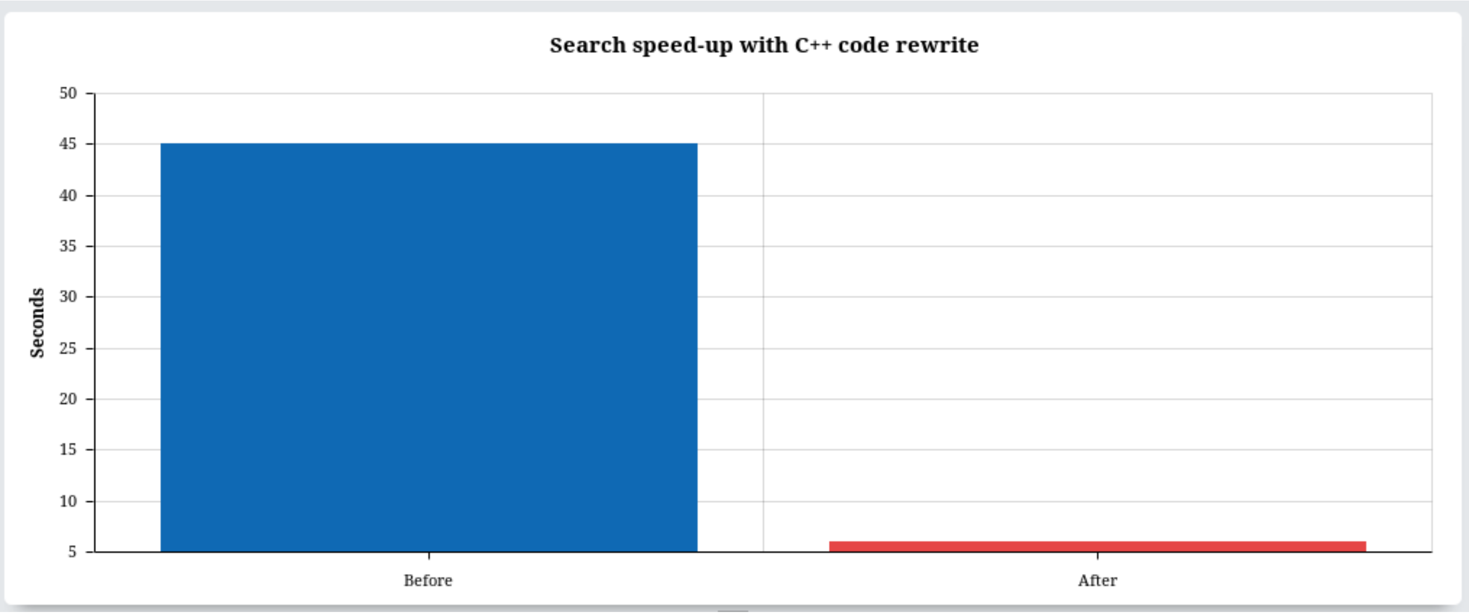
Authors: Huy Bui, Mihai Nicolae

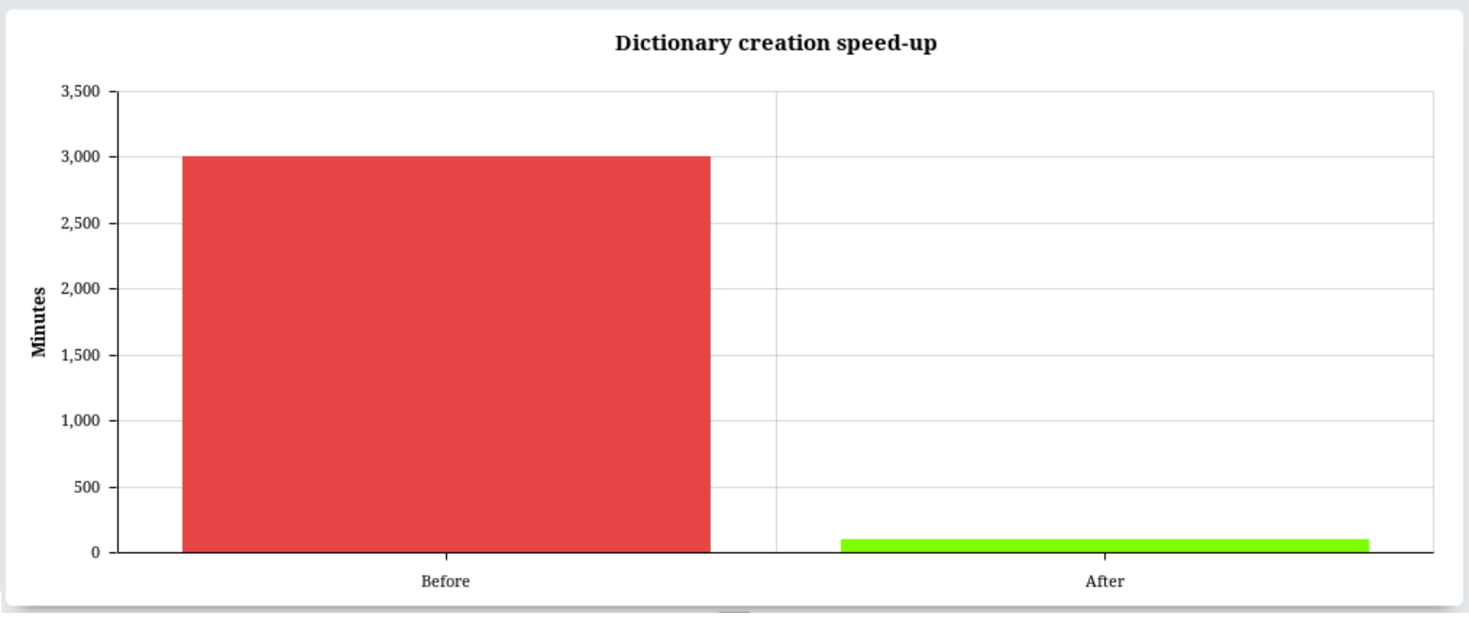
Research highlights

1. Gained working knowledge of Apache Hadoop framework.
2. Implemented and experimented with 2 different ways to parallelize search in Hadoop. Concluded that Hadoop is too slow given the small input size.
3. Searching graph in parallel in order to achieve acceptable web application response time
   - researched different C++ graph libraries and chosen Boost Graph Library (BGL)
   - Rewrote search algorithm in C++
   - Found the bottleneck of the search algorithm: finding the initial rootQueue. We parallelized this part and achieved 8x speed-up.
   - Removed irrelevant search results by adding logic to calculate similarity of concepts to the query.
   - Researched and applied other Boost libraries to add helpful features such as profiling.
   - Total lines of code: 900+
   - Future to-dos: serialize search DAG to disk, experiments against high quality dictionary.
4. Creating high quality dictionary
   - Experimented with more sophisticated heuristics, e.g., part of speech tagging to help remove unwanted words and stemming to reduce duplications, etc.
   - Parallelized dictionary creation. Achieved 180x speed-up.
   - This helps facilitate experiments of different heuristics to improve dictionary quality and, indirectly, search result.
   - Total lines of code: 600+


Run time
Sequential
Hadoop w. 4 mappers
Real time
0
20
40
60
80
100
120
0
50000
100000
150000
200000
Input side (# of nodes)


Total lines of code
Lines of code
1,000
900
800
700
600
500
400
300
Java
Python
C++


Search speed-up with C++ code rewrite
Seconds
50
45
40
35
30
25
20
15
10
5
Before
After


Dictionary creation speed-up
Minutes
3,500
3,000
2,500
2,000
1,500
1,000
500
0
Before
After