COMS 4995 PFP Project Proposal
Parallel Word Autocomplete

Aswin Tekur (at3584) and Vikrant Satheesh Kumar (vs2778)

Summary

Given a very large input data file (words.txt), another text file (incomplete.txt) that needs to be autompleted, a number k, generate a text file (auto-completed.txt) that contains the auto-completed version of incomplete.txt based on the k-th most frequent word that matches the incomplete word pattern from words.txt. If the k-th word doesn't exist or the word exists in words.txt, then do not replace it.

For example:
words.txt:
This is the words the file file using which the thing other files will be autocomplete.

{"this":2, "is":1, "the":3, "words":1, "file":2, "using": 1, "which": 1,
"other": 1, "files": 1, "will": 1, "be": 1, "autocomplete": 1, "thing": 1}

incomplete.txt
Thi is th words fil using which many other files wi b auto.

auto-completed.txt with k = 1
This is the words file using which many other files will be autocomplete.

auto-completed.txt with k = 2
thing is this words files using which many other files wil b auto.

Approach

(words.txt): Use parallel MapReduce to find the frequency count of all the words in words.txt and build a trie of words.

(incomplete.txt): Read the words and generate the auto-complete words based on the trie from words.txt. The list of auto-completed words can also be generated in a parallel manner. Finally, write the output to auto-completed.txt