HipgRap

Bicheng Gao (bg2640) \ Kangwei Ling (kl3076)

Introduction

Grep is a very useful command line tool for search patterns in text-based files. We would like to implement such a tool in haskell, called HipgRap, for purposes of both practising programming with haskell and gaining experience on building system tools focusing on performance. A typical grep tool has three parts: find the files to search, search patterns in those files, and print the result. Our implementation will focus on the search of string literals.

Motivation

Current haskell implementation of grep in the library is not efficient enough, especially when you compare it to other implementation in C(GNU grep) and Rust(ripgrep). The basic motivation is to use the parallelism feature of Haskell to boost the execution of reading files, finding matches in each line.

Potential Improvements

1. File level parallelism: Suppose you are searching some patterns in a directory, you definitely don't want to search file by file, if you can search different files simultaneously, and aggregate the results, it would be a great boost to the performance.
2. Line level parallelism: Think about a big file of several gigabytes, it will take time to go through the whole contents. What if we split this file into multiple chunks in the granularity of lines. We will have some small chunks and search the same pattern on them at the same time.
3. Apply the different matching algorithm when searching for string literals: Boyer-Moore Fast String Searching Algorithm and KMP algorithm are the tools that we can use to improve the brute-force search of matchings. It will reduce the time complexity from O(nm) to linear or sublinear.