Benjamin Snyder
bs3148

Par-Grep

I think I would like to take on the challenge of implementing a parallel grep. I'm fairly familiar with regular expressions, and think that I could probably write a simple lexer/parser to understand what they want to find, possibly use threads to generate the permutations (not sure if that's going to be efficient yet) and then use a divide and conquer where each thread attempts to scan a line. Doing this and "chunking a file" I can divide the work and conquer the text they give me faster. Combining the results at the end into a clean output.

https://github.com/PatricZhao/ParallelGrep
There's an implementation on github for doing this now in C, and I would just either use this as a basis or look at other implementations and scale this up to work in haskell

https://github.com/PatricZhao/ParallelGrep/blob/master/ParallelGrep.c
Considering that they were able to accomplish this in C in about 500 lines, it should not be too difficult to write in Haskell. I don't imagine it will be as clean, but could possibly be shorter than that.

I think this is good because there are multiple places where it makes sense to use parallelism that they discuss in their readme, both to scan directories and divide files into chunks to scan.

I imagine that it will be a challenge to beat GNU grep since it's highly optimized, but maybe I could beat this C implementation, or if I can find a simple grep implementation in haskell elsewhere, and compare myself to that. So that I can fairly gauge the speedup that my parallelism offers.