Ain Shams University
Faculty of Computer and Information Sciences
Department of Scientific Computing

Parallel Text Search

High-Performance Computing Course Project - 2016

Team Members' Names in Arabic:
ندى علاء عبد الوهاب العواد
محمد اغيد اسامة حالول
عمر معتز عبد الواحد احمد عطيه
اسلام مجدى سعيد حسن
عبد الرحمن عثمان محمد سعيد

Project Description:

We created a program that determines whether a string is found in a paragraph or not. The algorithms implemented search the file in 3 ways:

- Sequential search algorithm: a normal linear search algorithm that runs sequentially.
- Parallel search algorithm: the work load is divided among N threads in the processor.
- Parallel-GPU search algorithm: uses the computer's graphics processing unit to perform the search in a very high speed.

Input:

- Text file (up to 5,000,000 characters)
- A search key (up to 1,000,000 characters)

Output: Found or Not, The search key positions

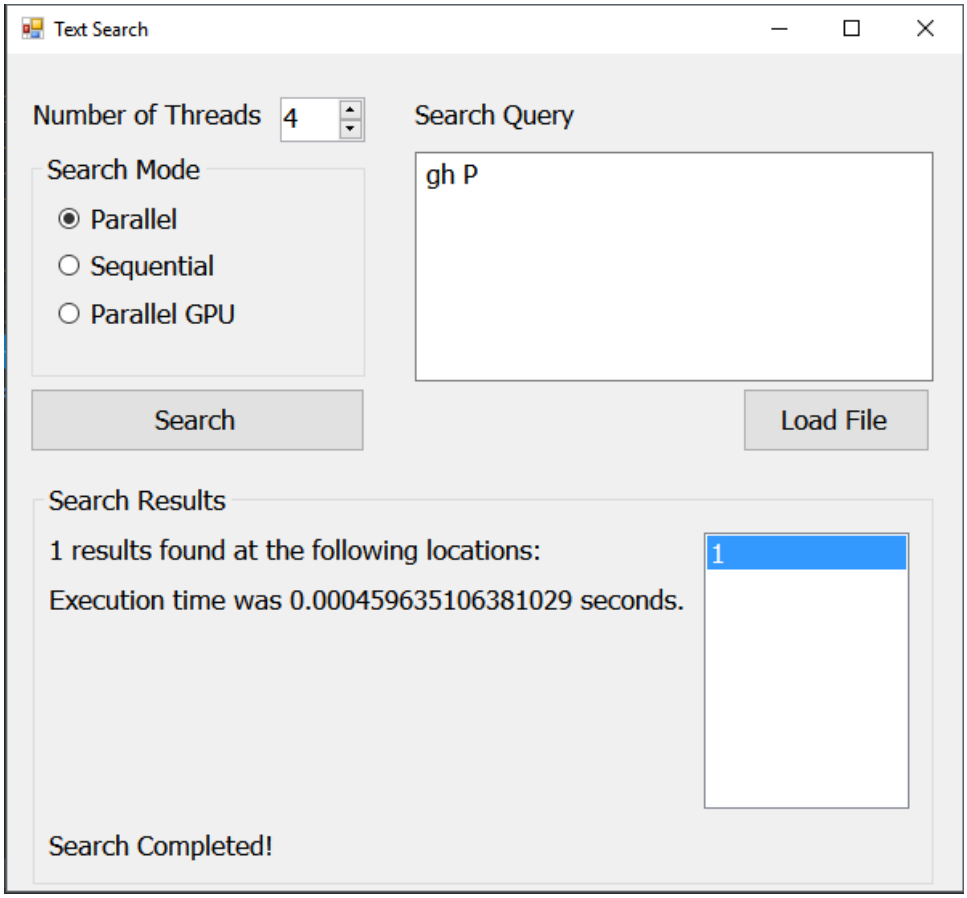
Examples

Input:

- Text: "High Performance Computing Project"
- Search key: "gh P"

Output:
Found, at position 2.

Screenshot:



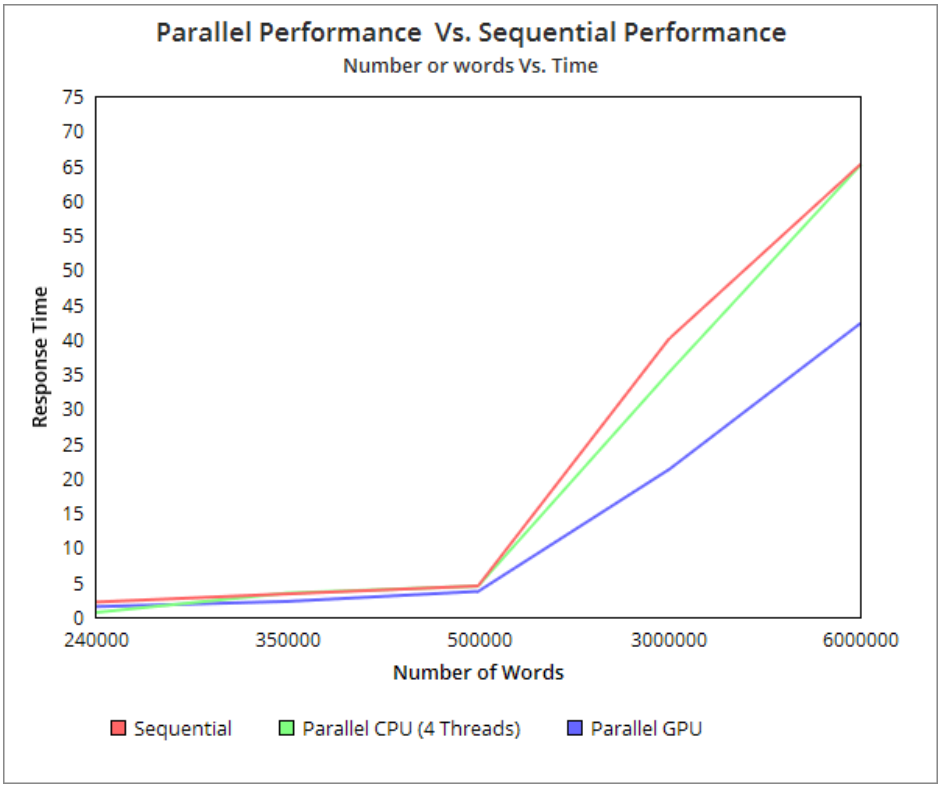
Performance:

Searching for a word in a big farm of words is an important problem. Performance is a major parameter in it. This can make enormous differences if the search query is huge. Wildcard search cases also should be fast to output a result. Consequently, performance is important.

Parallel Search Vs. Sequential Search

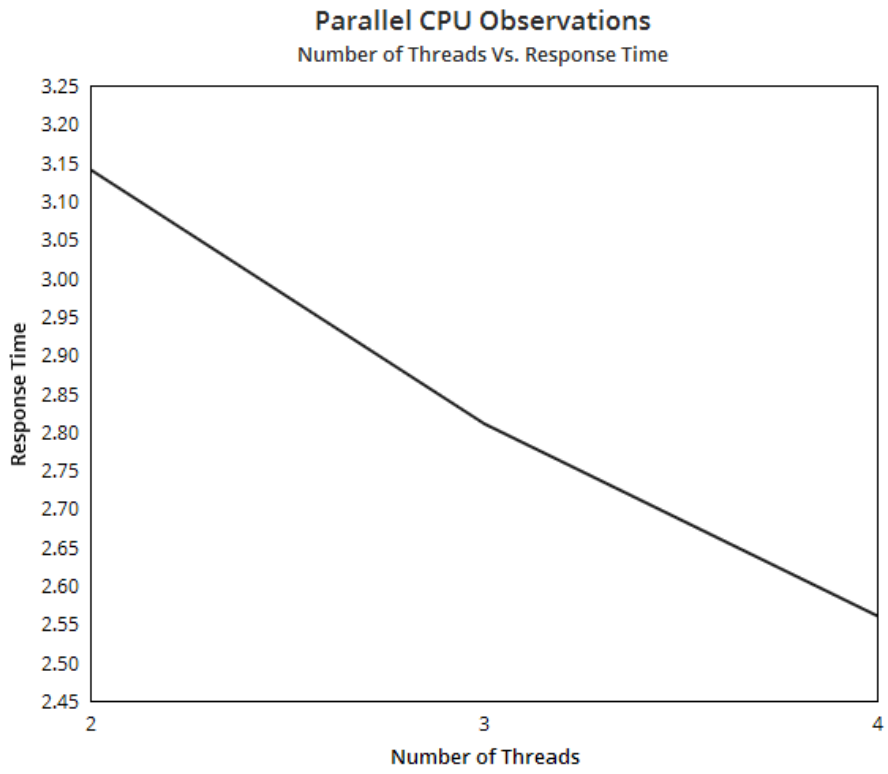
We made a 3-option program that can search in sequential mode, CPU parallel mode, and GPU parallel mode. You can choose the number of threads in the CPU Parallel version.

We've tested the program on different input sets and we've generated this graph that illustrates the results. Response time is in seconds.



Parallel CPU Comparisons (X Number of Threads)

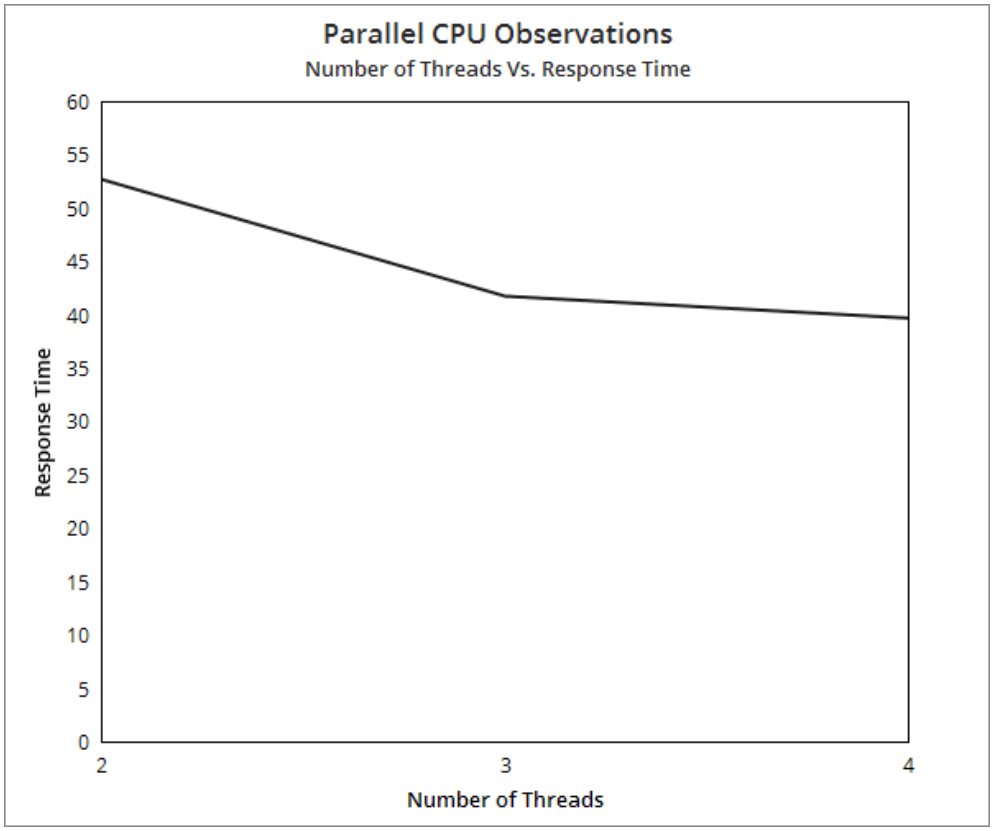
We tested the program on the same input set in the CPU parallel mode with changing the number of threads to record out observations. The results are illustrated in the following chart:



Sample test size was 240,000 words. Response time is in seconds.

It's clear that the time response decreases if we added more threads to do the computation.

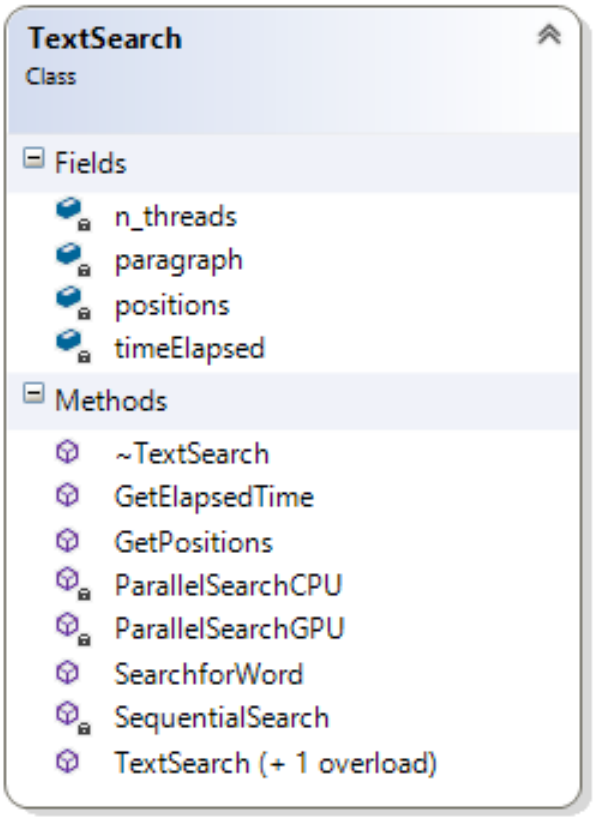
We made another test with 3000000 words, and these were the results (In the following chart). Response time is in seconds:



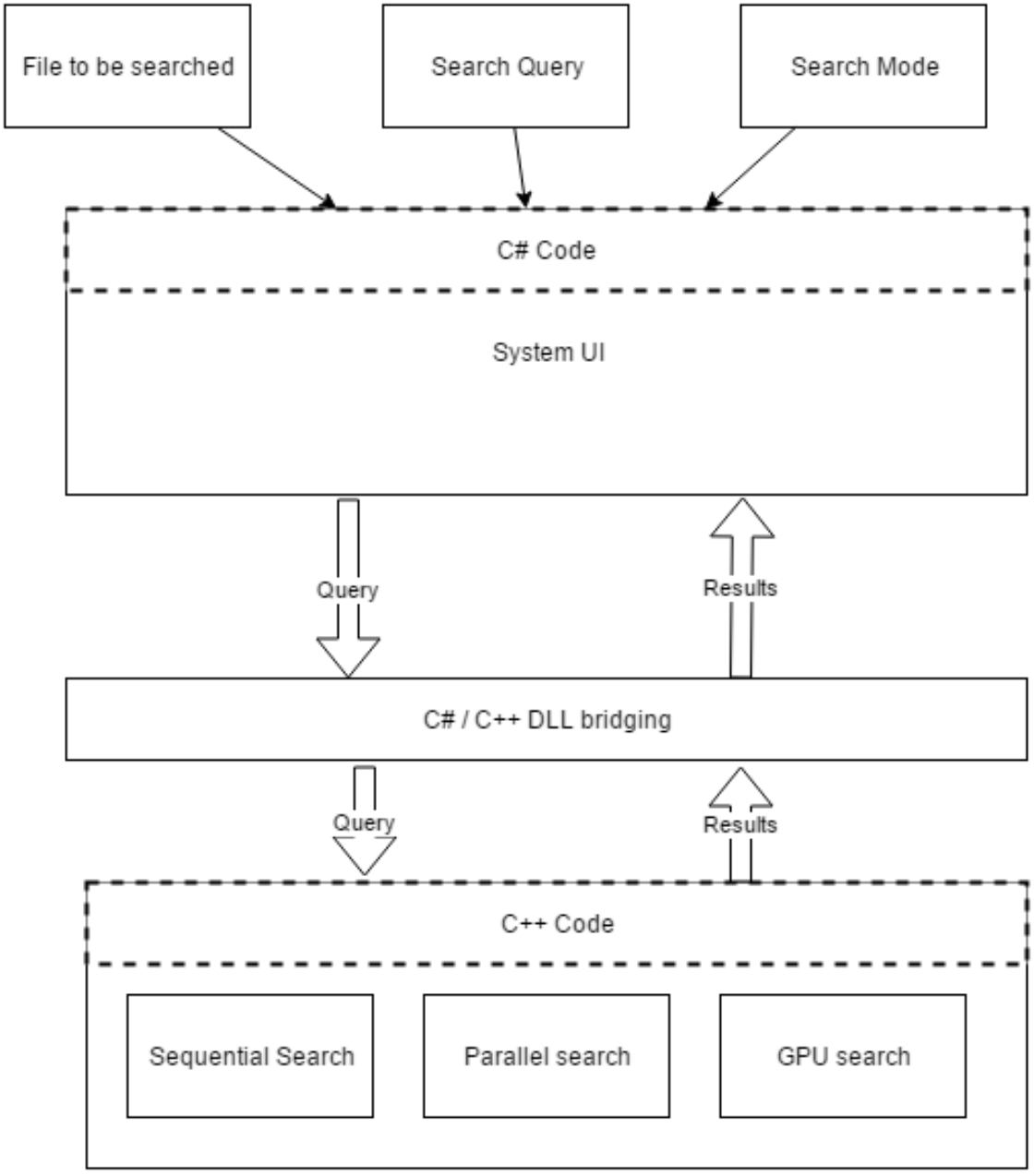
Observations:

- GPU mode works like a charm in large input sets. The difference doesn't appear clearly in small sets.
- GPU Search can add a little amount of latency time before the only first time search for each file.
- Sometimes, CPU parallel mode can reach the same peak time as the sequential mode.

Structure:



Class Diagram



Work Flow Diagram

References:

The charts were done using ChartGo.

The sample test files are in dwyl Github repository, some of them were doubled in data to measure the performance on large inputs.