CS302

Assignment 3: Long addition in parallel

Description

Addition. Such a simple task, you might have learned the algorithm for long addition when you were in grade school. Now you will have the opportunity and re-live this childhood experience by adding two large numbers (so large that they cannot be stored into an `int` or even an `unsigned long long`), thus you will need to store the number into a dynamic array or a `vector`. Thus each element will be a digit in its appropriate place value, so you can write the iterative code to implement long addition. To make matters worse, we will do this in parallel using a `vector` of threads.

Suppose we want to add the two 32 digit numbers: 56781257987110923417082367650001 and 74321168999909850024784138764000. Of course this can be done sequentially in $O(n)$ time where n is the size of the two numbers. Suppose we had 8 threads that could run in parallel, we can break the problem where each thread adds a section of size 4 and add them up in parallel since the each thread can perform its own addition without the result of any other thread.

	5678	1257	9871	1092	3417	0823	6765	0001
+								
	7432	1168	9999	0985	0024	7841	3876	4000

1 3110 **0** 2425 **1** 9870 **0** 2077 **0** 3441 **0** 8664 **1** 0641 **0** 4001

As you can see above, each portion was added (using long addition) and the number in red is the carry over that would be added to the section to the left of where the carry over bit was computed, and so on. Once all the carry overs are added to each section, the numbers can be concatenated to obtain the final answer.

Your program will read two large numbers (64 digits and larger), and you will need to store each digit into an array or `vector`. Then you will need to determine the maximum number of threads that can run concurrently that is also a power of 2 (You can assume the number of digits for the numbers used will also be powers of 2). You then need to divide up the work such that each thread performs the addition on a specific portion of the number, compute its sum and store the carry overs for each portion. Then at the end you will perform add the carry overs to the correct section of the number, this will be done sequentially.

Written Portion

You will submit a written portion for this assignment by answering the following questions

1. If we were adding two n digit numbers in parallel using n amount of threads, where each thread only needs to add two 1 digit numbers, what would be the overall asymptotic runtime assuming each thread completes its work at the same time if no carry over is present? Please explain.
2. What would be the asymptotic runtime if we had a carry over that would be added from the rightmost digit all the way to the left most digit? Please explain.
3. Which parts of this algorithm (for this assignment) cannot be done in parallel? Please explain why.

Specifications

- Document your code
- Use the C++ `thread` library to implement the parallel addition algorithm
- Do not spawn more threads than your system can handle, i.e. do not spawn more threads than the amount returned by `std::thread::hardware_concurrency()`
- Be careful with passing in reference parameters into a thread function, that causes compiling errors, you can however have the parameters as pointers and pass an address (reference) as an actual parameter (this is not required just FYI)
- Global variables are ok in this assignment especially since the threads might share some memory
- If you use the remote server to compile and run code **USE CARDIAC.CS.UNLV.EDU**

Example Output

```
% g++ -pthread main.cpp
%./a.out

Enter a number: 337736459843544577880990361200105860338230585677454661410231349 3

Enter a number: 808021259648381470450494357235067396448883459032241682588640378 1

Result = 1145757719491926048331484718435173256787114044709696343998871727 4

% ./a.out

Enter a number: 999999999999999999999999999999999999999999999999999999999999999 9

Enter a number: 000000000000000000000000000000000000000000000000000000000000000 1

Result = 100000000000000000000000000000000000000000000000000000000000000 0
```

Submission

Upload your source code to the code grade submission and submit your write up into the canvas submission by the deadline

References

- Link to the top image can be found at $https://www.seekpng.com/idown/u2q8q8y3u2a9q8r5_naturalcrit-mister-meeseeks-home-brewery-rick-and-rick/$
- Supplemental Video https://youtu.be/gCyg8Tuh0Dw