CS/SE 4F03 Assignment 2

28 January 2016

Due date: 8 February

Problem 1 (50 points) An RSA public key (see e.g. https://en.wikipedia.org/wiki/RSA_(cryptosystem)) is the product of two distinct large prime numbers. Cracking such a key involves finding the two prime numbers.

Given a key n, a brute force approach for finding its two prime factors is to multiply prime numbers p and q to see if $pq = n$.

For example, suppose $n = 85$. We can try

$$\begin{array}{l} 3 \times 5 = 15 \\ 3 \times 7 = 21 \\ 3 \times 11 = 33 \\ 3 \times 13 = 39 \\ 3 \times 17 = 51 \\ 3 \times 19 = 57 \\ 3 \times 23 = 66 \\ 3 \times 31 = 93 \end{array} \quad \text{and then} \quad \begin{array}{l} 5 \times 7 = 35 \\ 5 \times 11 = 55 \\ 5 \times 13 = 65 \\ 5 \times 17 = 85 \end{array}$$

That is, we have $p = 5$ and $q = 17$.

A simple serial algorithm producing the above is

```
found = false;
p  = some prime number
q  = p
while p*p < n
        q  = nextprime(q)
        if  p*q == n
                found = true
                break
        if p*q > n
                p = nextprime(p)
                q = p
```

Implement an MPI program that takes as an input a number n that is the product of two prime numbers and finds (given sufficient time) these numbers.

You program should run as

```
mpirun -np P ./factor n
```

where `P` is the number of processes, `factor` is the name of your executable, and `n` is a decimal string representing an integer number that is the product of two prime numbers.

- You can use any of the number theoretic functions from GMP, the GNU Multiple Precision Arithmetic Library http://gmplib.org/. See e.g. `mpz_probab_prime_p`, `mpz_nextprime`.

  To link your program with the GMP library use `-lgmp`.

- It will be difficult to factor large numbers within the time we have for this assignment. One of the goals here is to distribute the work such that each process takes about the same amount of time.

- When your program finishes, it should create a text file with name `time_n`. For example, if $n = 323$, the file name is `time_323`. This file should contain two columns, where each row is process number and the time in seconds this process takes to finish. For example, with 4 processes, your file should look like

```
0    1.12e1
1    1.34e1
2    2.01e1
3    0.91e1
```

Grading

1. (5 points) Describe your algorithm in words and using pseudo code.

2. (35 points) Choose a number n such that with $P = 1$ your program runs in about 15 minutes. Produce speed up and efficiency plots for $P = 2, 4, 8, 16, 32, 64$.

   Your mark will depend on the efficiency E you achieve with $p = 32$ and will be calculated as

   $$(E + 0.05) \times 35$$

   For example, if $E = 0.5$, then $(0.5 + 0.05) \times 35 = 19.25$.

   (5 points) Discuss the speed up and efficiency you have obtained. In particular, explain why you obtain or do not obtain good speed up.

3. (5 points) For an input number (key) of your choice, and total execution time of your program of at least 3 hours on 16 processes, submit a plot of CPU time per versus process number. Submit this key and the time it takes to factor on 16 processes.

Submit also

- All your program files to SVN under directory A2.

  In this directory, there must be a `makefile`, such that when `make` is typed the executable `factor` is created.

- Hard copy of your programs.

You may find these links interesting and helpful:

- https://en.wikipedia.org/wiki/RSA_Factoring_Challenge.
- http://www.calculatorsoup.com/calculators/math/prime-factors.php
- http://primes.utm.edu/

Problem 2 (10 points) Derive a formula for the speedup and formula for the efficiency of the program implementing the trapezoidal integration (as discussed in class). Consider both versions: (a) with send and receive and (b) with broadcast and reduce.

For a message of size m words, assume that a send or a receive takes $t_s + mt_w$ time and a broadcast or a reduce takes $(t_s + mt_w) \log p$ time.

Discuss if each of (a) and (b) is strongly scalable.

Problem 3 (5 points) A vendor gives you quotes for processors

- Intel Xeon E5-2630 V3 20MB 8 CORE 2.40GHZ LGA2011 8.00GT/S — $978.98 and
- Intel Core I7-5820K 3.30GHZ (3.6GHZ Turbo Mode) Six Core 15MB Hyperthreading LGA2011-V3 — $529.99

Assuming that they will be used mostly for intensive floating-point computations, speed is important, and money is not a serious issue, which one would you buy? Give sufficient details supporting your claim.