Bardia Mojra – 1000766739

03/26/2022 – Assignment 03

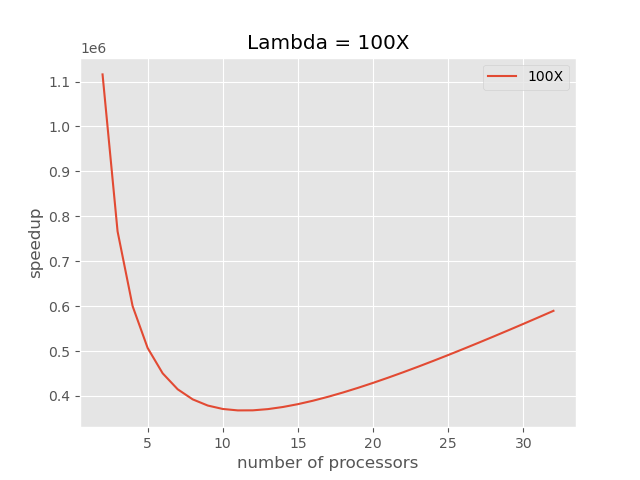
Q1. Assuming the strike to be equal to unit time, a sequential implementation of the Sieve of Earthosthenes takes about 2,122,048 unit time to mark 1,000,000 (n) numbers. This is obtained by

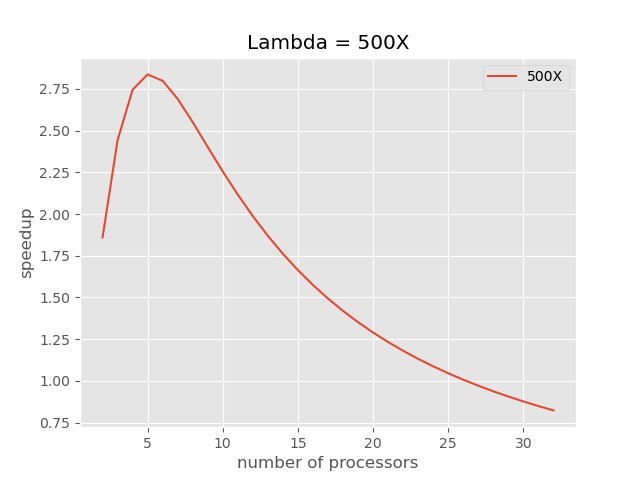
Since there are 78,498 primes smaller than n, a control-parallel algorithm would take equal time to transmit the number to another processor for the marking process, for a given unit communication time. Thus, 2,122,048+78,498=2,200,546 would be total time under Amdahl’s law. So, the upper bound on the speedup for p processors would be given by,

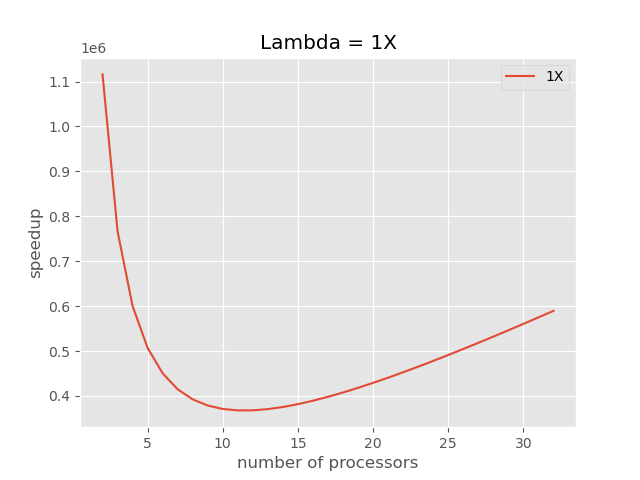
S = 1/(.0357+(.9643/p)), where 78,498/2,200,546 = 0.0357, and 1- .0357 = 0.9643.

Q2. The following formula was used to estimate total commute times and then speedup values.

| p | 1x | 100x | 500x |
| --- | --- | --- | --- |
| 2 | 1115840.06 | 1115840.06 | 1.86 |
| 3 | 766293.38 | 766293.38 | 2.44 |
| 4 | 599920.03 | 599920.03 | 2.75 |
| 5 | 506816.03 | 506816.03 | 2.84 |
| 6 | 450346.69 | 450346.69 | 2.80 |
| 7 | 414811.45 | 414811.45 | 2.69 |
| 8 | 392360.02 | 392360.02 | 2.55 |
| 9 | 378631.13 | 378631.13 | 2.40 |
| 10 | 371008.01 | 371008.01 | 2.26 |
| 11 | 367825.47 | 367825.47 | 2.12 |
| 12 | 367973.34 | 367973.34 | 1.99 |
| 13 | 370683.09 | 370683.09 | 1.87 |
| 14 | 375405.72 | 375405.72 | 1.76 |
| 15 | 381738.68 | 381738.68 | 1.66 |
| 16 | 389380.01 | 389380.01 | 1.57 |
| 32 | 589490.00 | 589490.00 | 0.82 |







Q3. A) Total strikes for a sequential algorithm for odd numbers under 1000 and 1000000 are 77619 and 1100226, respectively. Then, we add the total prime numbers to each, 77619+168=77787 and 1100226+78,498=1178724. Normal sequential algorithm would give us 155239+168=155407 and 2200453+78,498=2278951 for n=1000 and n=1000000, respectively. We would have the following speedups, 1.997 and 1.933 for n=1000 and n=1000000, respectively.

B) Total execution times for 1 to 16 processors for lambda = 100x for n=1,000 and n=1,000,000;

| p | n=1000 | n=1000000 |
| --- | --- | --- |
| 2 | 17582.85 | 1115840.06 |
| 3 | 34121.90 | 766293.38 |
| 4 | 50791.42 | 599920.03 |
| 5 | 67513.14 | 506816.03 |
| 6 | 84260.95 | 450346.69 |
| 7 | 101023.67 | 414811.45 |
| 8 | 117795.71 | 392360.02 |
| 9 | 134573.97 | 378631.13 |
| 10 | 151356.57 | 371008.01 |
| 11 | 168142.34 | 367825.47 |
| 12 | 184930.47 | 367973.34 |
| 13 | 201720.44 | 370683.09 |
| 14 | 218511.84 | 375405.72 |
| 15 | 235304.38 | 381738.68 |
| 16 | 252097.86 | 389380.01 |
| 32 | 520848.93 | 589490.00 |

C) We have following speedups,

| p | n=1000 | n=1000000 |
| --- | --- | --- |
| 2 | 8.84 | 2.04 |
| 3 | 4.55 | 2.97 |
| 4 | 3.06 | 3.80 |
| 5 | 2.30 | 4.50 |
| 6 | 1.84 | 5.06 |
| 7 | 1.54 | 5.49 |
| 8 | 1.32 | 5.81 |
| 9 | 1.15 | 6.02 |
| 10 | 1.03 | 6.14 |
| 11 | 0.92 | 6.20 |
| 12 | 0.84 | 6.19 |
| 13 | 0.77 | 6.15 |
| 14 | 0.71 | 6.07 |
| 15 | 0.66 | 5.97 |
| 16 | 0.62 | 5.85 |
| 32 | 0.30 | 3.87 |

D) The speedup is higher for the improved algorithms because the number array is reduced in half and the execution time is reduced by that ratio.

Q4. For one person, it would take t\_c(n-1).

1. In a situation where all 8 people are sitting in a circle, they all need to pass the sums forward to be added again. Initial sums happen, then 1 pass, sum execution, then two passes, then sum execution and then 4 passes, then sum again and 1 pass to load new numbers or to communicate the results. Thus, we could assume the total execution and communication time can be obtained by, n/p + p t\_c.
2. For an all-to-all configuration, we would have a total time of n/p+2logp.

Q5. Bulk Synchronous Parallel (BSP) models are similar to Parallel Random Access Machine (PRAM) models but with few major differences. In BSP models, each processor has its own local memory and the data stored in them is synchronized through a communication network and the model features a barrier function that forces all processors to synchronize at certain points in the algorithm. PRAM algorithms do not utilize distributed memory and bulk synchronization via barrier function.