EE382V Multicore Computing: Assignment 1

Wenwen Zhang [wz3585]
Chunheng Luo [cl38532]

September 11, 2016

Question 0

TACC User IDs:

- Wenwen Zhang: wenwen22

- Chunheng Luo: chluo

Question 1

Part (a)

- Assuming other parts of the program can be sped up by the factor of n, then the overall speedup is

$$Speedup = \frac{1}{0.4 + \dfrac{0.6}{n}}$$

- Assuming the method M accounts for x of the program's execution time on a single-core processor, M can be sped up by 2^3 and other parts of the program can be sped up by the factor of n, then the overall speedup is

$$Speedup = \frac{1}{\dfrac{x}{8} + \dfrac{(1-x)}{n}}$$

So in order to double the speedup, we require

$$\frac{1}{\dfrac{x}{8} + \dfrac{(1-x)}{n}} = 2 \times \frac{1}{0.4 + \dfrac{0.6}{n}}$$

which leads to

$$x = \frac{0.2n - 0.7}{0.125n - 1}$$

Therefore, M must account for $(0.2n - 0.7)/(0.125n - 1)$ of the total execution time **on a single-core processor** in order to double the overall speedup of the program.

Part (b)

Assuming the parts of the program that can be totally parallelized account for P of the total execution time on a single-core processor, and all of the other parts of the program, which accounts for $(1 - P)$ of the total execution time on a single-core processor, are not able to gain any speedup from the multicore architecture, then we have

$$S_2 = \frac{1}{(1-P) + \dfrac{P}{2}}$$

and

$$S_n = \frac{1}{(1-P) + \dfrac{P}{n}}$$

Solving the equations, we get

$$S_n = \frac{nS_2}{(2-n)S_2 + 2(n-1)}$$

Question 2

Question 3

In order make Filter Algorithm able to solve the l-exclusion problem, we can simply reduce the number gates from N to $(N - l)$.

```
const int   N;
int[N]      gate init 0;
int[N-l+1]  last init 0;

/* For P_i */
request CS;
for (k = 1 : N - l) {
  gate[i] = k;              // P_i is at gate k
  last[k] = i;              // P_i updates last for that gate

  int numAhead = l + 1;
  while (numAhead > l && last[k] == i) {
    for (j = 1 : N - l) {
          if (j != i && gate[j] >= k)
            numAhead += 1;
    }
  }
}
CS;
release CS;
gate[i] = 0;
```

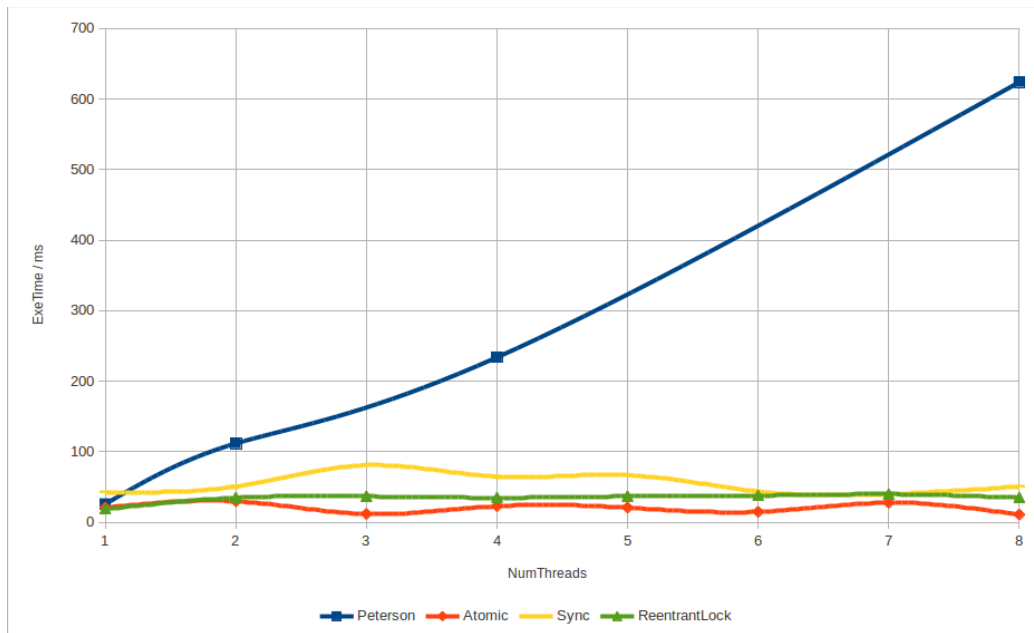# Question 4

* Code submitted on Canvas.

Figure 1: Execution time using the four methods

Question 5

* Code submitted on Canvas. The execution time plot is shown in Fig 1.