# More multithreading

9/25/24

Administrivia

- Reading for Friday: Section 14.1

- HW 3 (multithreading) due Tuesday 10/1

- Exam 1 out Wednesday
  - Multi-day takehome
  - Open notes and book, closed internet and friends
  - No class next Thursday (10/3)
  - Due early the next week (probably Monday night)
  - Everything thru multithreaded (induction, asymptotic ordering, AVL trees, D&C, multithreaded)

Multithreading so far

Allow different parts to run at the same time; program with spawn and sync

Metrics: Work T_1 = Total amount to do

Span T_∞ = Length of longest path of dependencies
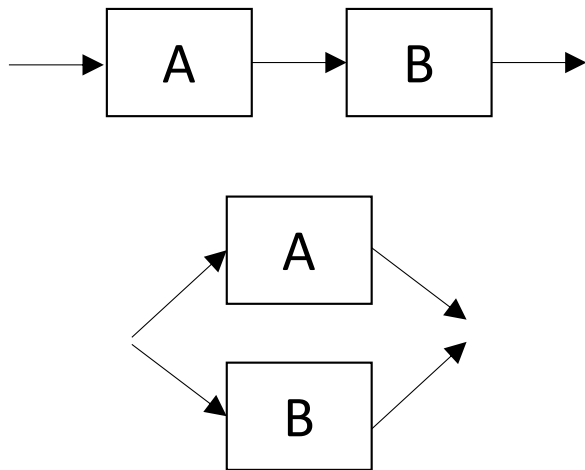

```
void do_it(int s, int e) {
        if(s == e)
                loop body
        else {
                spawn do_it(s, (s+e)/2)
                do_it((s+e)/2+1, e)
                sync
        }
}
```

What are T_1 and T_∞ for the following?

Transpose(A) {
  n = A.rows
   for j = 2 to n
       for i = 1 to j-1
          exchange a_{ij} with a_{ji}
}

A. θ(n log n) and θ(n)
B. θ(n log n) and θ(n log n)
C. $\theta(n^2)$ and θ(n log n)
D. $\theta(n^2)$ and $\theta(n^2)$
E. None of the above

What are T_1 and T_∞ for the following?

Transpose(A) {

  n = A.rows

  for j = 2 to n

      for i = 1 to j-1

         exchange a_{ij} with a_{ji}

}

A. Θ(n log n) and Θ(n)

B. Θ(n log n) and Θ(n log n)

C. $\Theta(n^2)$ and Θ(n log n)

D. $\Theta(n^2)$ and $\Theta(n^2)$

E. None of the above

What are T_1 and T_∞ for the following?

Transpose(A) {
  n = A.rows
  for j = 2 to n
        parallel for i = 1 to j-1
            exchange a_{ij} with a_{ji}
}

A. Θ(n log n) and Θ(n)
B. $\Theta(n^2)$ and Θ(n)
C. $\Theta(n^2)$ and Θ(n log n)
D. $\Theta(n^2)$ and $\Theta(n^2)$
E. None of the above

What are T_1 and T_∞ for the following?

Transpose(A) {

  n = A.rows

  for j = 2 to n

      parallel for i = 1 to j-1

         exchange a_{ij} with a_{ji}

}

A. $\Theta(n \log n)$ and $\Theta(n)$

B. $\Theta(n^2)$ and $\Theta(n)$

C. $\Theta(n^2)$ and $\Theta(n \log n)$

D. $\Theta(n^2)$ and $\Theta(n^2)$

E. None of the above

What about this version?

```
Transpose(A) {
  n = A.rows
  parallel for j = 2 to n
        parallel for i = 1 to j-1
              exchange a_ij with a_ji
}
```

Practice problem: Summing an array

based on 26-4a

Give multithreaded code to sum the values in an array A using O(n) work and having O(log n) span.

Practice problem: Matrix-vector multiplication

Another matrix operation is matrix-vector multiplication, in which a matrix M is multiplied by an array x to produce another array y. Here is a serial implementation:

```
for(int i=0; i < n; i++) {
    y[i] = 0;
    for(int j=0; j < n; j++)
        y[i] = y[i] + M[i][j] * x[j];
}
```

a) Parallelize the iterations of the outer loop (controlled by i). The inner loop should be unchanged. Analyze the work and span of your code.

Practice problem: Matrix-vector multiplication

Another matrix operation is matrix-vector multiplication, in which a matrix M is multiplied by an array x to produce another array y. Here is a serial implementation:

```
for(int i=0; i < n; i++) {
    y[i] = 0;
    for(int j=0; j < n; j++)
        y[i] = y[i] + M[i][j] * x[j];
}
```

b) Parallelize the iterations of the inner loop (controlled by j). The outer loop should be unchanged. Analyze the work and span of your code.