Homework 2 – Dependencies, Patterns, Map Pattern, and Collectives Pattern

Due Date: 4/28 @ 5pm

Dependencies

Consider the following nested loop:

```
for $i = 5$ to $100$ do
    for $j = i$-$2$ to $i$ do
        $a[i][j] = a[i-4][j];$
    end
end
```

Suppose the loop is executed on a shared-memory machine using multi-threading.

1. Draw the data dependency graph for the nested loop.
2. What level of parallelism is available with this nested loop?
3. How you would parallelize this nested loop for three threads? Write your answer in CilkPlus, TBB, or OpenMP.

Patterns

1. One algorithm used in image blurring is to loop through each pixel in an image, gather the values of the current pixel's neighbors and average them into the current pixel. If we wanted to do this in parallel, what pattern might we choose? Explain.
   (a) Reduce
   (b) Fork-Join
   (c) Stencil
   (d) Recursion
2. Given an array of data, what parallel pattern might one use to copy a subset of that data with another array of indices where the indices correspond to our input array? Explain.
   (a) Pack
   (b) Gather
   (c) Stack Allocation
   (d) Scatter

Map Pattern

1. What makes map such a desirable target in terms of parallel computing?
2. Given matrix A:

$$A_{m,n} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

   And vector b:

$$b_n = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

   It is possible to multiply matrix A by vector b such that:

$$Ab = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_{1,1}b_1 + a_{1,2}b_2 + \cdots + a_{1,n}b_n \\ a_{2,1}b_1 + a_{2,2}b_2 + \cdots + a_{2,n}b_n \\ \vdots \\ a_{m,1}b_1 + a_{m,2}b_2 + \cdots + a_{m,n}b_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = c$$

   So, each row of the matrix is multiplied element by element of our vector, and then summed into one output element, just like a dot product of two vectors. The resulting elements are then gathered into an output vector. Write a pseudo-code algorithm to do this in parallel using a map operation.
3. Why might one want to "fuse" a series of maps on the same data, as opposed to performing multiple maps one after the other?

Collectives Pattern

1. Given a list of words, write a pseudo-code function to sort them in parallel in dictionary order.
2. Why would we want to "tile" our data when performing a reduce or a scan? Doesn't this lead to less parallelization than non-tiled data?
3. Only associativity of a combiner function is required to parallelize reduce and scan. Addition and multiplication of real numbers both satisfy associativity. What then is the danger of adding and multiplying floating point numbers in parallel?