King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No xx – Quarter 3 - 2023

Question 1

Let's define a Quadtree as a tree:

- that is empty, or
- that is composed of a root and 4 possible sub-Quadtrees.

Let's consider that the data of a Quadtree is stored in a N by N matrix called ***data***. Let's consider that we would like to process this Quadtree (data) in a parallel way. Let's consider the following kernel:

__global__ void **Quadtree_Kernel(**int *** data,** int **R,** int **C,** int **W,** int **level);**

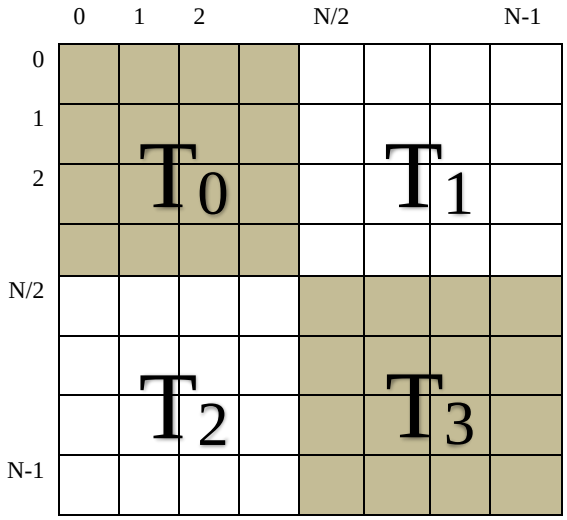
- This kernel will process the sub-Quatree that is represented by a sub-Matrix of size ***W * W*** starting from ***data[R, C]***.
- ***level*** is the level of the sub-Quadtree.

The parallel processing of a Quadtree is launched by the main program using the following call:

Quadtree_Kernel<<<1,4>>>(data, 0, 0, N, 1);

This will launch a grid composed of 1 block of 4 threads. Every thread will process a sub-Quadtree as follows:

- Thread T_0: will process the sub-Quadtree S_0, that corresponds to the data starting from ***data*** [0, 0] with width = N/2
- Thread T_1: will process the sub-Quadtree S_1 that corresponds to the data starting from ***data*** [0, N/2] with width = N/2
- Thread T_2: will process the sub-Quadtree S_2 that corresponds to the data starting from ***data*** [N/2, 0] with width = N/2
- Thread T_3: will process the sub-Quadtree S_3 that corresponds to the data starting from ***data*** [N/2, N/2] with width = N/2



Every sub-Quadtree will be decomposed recursively into 4 sub-Quatrees until no more decomposition is possible. So, every thread T_i will process a sub-Qaudtree S_i. Every thread T_i will launch 4 threads to decompose recursively its corresponding sub-Qautree as explained above.

1. In order to identify its corresponding sub-Quadtree $\mathbf{S_i}$, every thread $\mathbf{T_i}$ will calculate 2 values $\mathbf{X_i}$ and $\mathbf{Y_i}$ which will be used to calculate the starting address of its corresponding sub-Quadtree $\mathbf{S_i}$. Values of $\mathbf{X_i}$ and $\mathbf{Y_i}$ for every thread are as follows:

$(\mathbf{X_0}, \mathbf{Y_0}) = (0, 0)$ for thread $\mathbf{T_0}$.	$(\mathbf{X_1}, \mathbf{Y_1}) = (0, 1)$ for thread $\mathbf{T_1}$.
$(\mathbf{X_2}, \mathbf{Y_2}) = (1, 0)$ for thread $\mathbf{T_2}$.	$(\mathbf{X_3}, \mathbf{Y_3}) = (1, 1)$ for thread $\mathbf{T_3}$.

   As such, a thread $\mathbf{T_i}$ will consider that the data that correspond to its sub-Quadtree $\mathbf{S_i}$ starts from the address:

$$\boldsymbol{data[R + (X_i*N/2), C + (Y_i*N/2)]}$$

   a. Give the code that will calculate $\mathbf{X_i}$ and $\mathbf{Y_i}$ for every thread $\mathbf{T_i}$.

2. Give an implementation of the kernel. We assume that the process will stop at level 10.