9 رابعة حاسبات ٢٠١٩

Faculty of Electronic Engineering Fourth Year Dept. of Computer Science & Engineering Time (3 hrs.): 10 am – 1 pm **Instructor: Prof. Nabil Ismail**	جامعة المنوفية	Subject: **Elective 5** 1st Semester 2018/2019 Sunday 6/1/2019 No. of pages: 2 Total Marks: 70 Marks

Answer as much as you can:

[1] **[15 Marks]**

i) Multiple choice questions (you must select only one choice):

1- To take advantages of the multicore CPUs/manycore GPUs you have to use:
a) Shared programs **b)** distributed programs **c)** multithreaded programs

2- A parallel program runs on a single multicore node uses ....
a) distributed memory **b)** loosely coupled cores **c)** shared memory

3- CUDA is well suited for implementing the ..... parallel design model.
a) SISD **b)** SIMD **c)** SPMD

ii) Fill in the spaces:

1- Parallel programmers use a ...., or .... to write shared memory parallel programs

2- Parallel application dimensions taxonomy are ...., and ....

3- PCAM parallel solutions methodology stands for, ..., ..., ..., and ....

4- CNNs are the most representative supervised deep learning model. They consist of mainly 4 types of layers ...., ..., ..., and .....

iii) True/False you should state why?

1- A shared address space facilitates migration from a sequential programming model to a parallel one.

2- GPU cores best performance model is MIMD parallelism

3- OpenCL distinguishes between the devices (usually GPUs or CPUs) and the host (CPU).

4- Data-dependent branching will perform well on a GPU.

[2] **[20 Marks]**

i) A sequential application with a 20% part that must be executed sequentially, is required to be accelerated three-fold. How many CPUs are required for this task? If the required speedup was 5, what would be the number of CPUs required?

ii) A parallel application running on 10 CPUs, spends 15% of its total time, in sequential execution. What kind of CPU (how much faster) would we need to run this application completely sequentially, while keeping the same total time?

iii) Why is multithreading needed? How can multiple threads run simultaneously in a single-processor system? and how can they run on multicore system? Write a program that launches 1,000 threads. Each thread adds 1 to a variable sum that initially is 0.

[3] [20 Marks]

i) Recall that a number is prime if it is divisible only by itself and 1.

a) Write an algorithm uses a trial division to test the primality of a number x, i.e. whether the number x is prime or not.

b) Write the parallel program steps to test a big number x if prime or not. You can use any parallel programming model (language) to implement the proposed algorithm in (i-a). In your parallel program steps, you must use the multicore multithreaded.

ii) Why use the parallel reduction pattern?

Write a parallel program to implement the Monto Carlo algorithm to approximate the value of π.

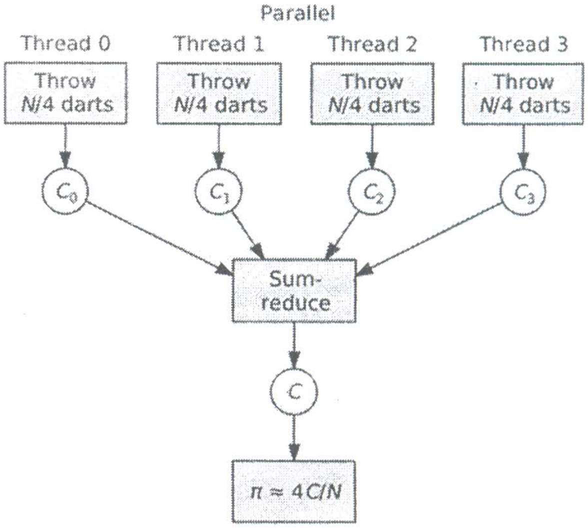
The algorithm works by taking a random sample of points (C) from a uniform distribution (N) with 0 < x,y < 1 and determining if that point lies within the first quadrant of the unit circle centered at (0,0). If it does, that is considered a success, then

$\boldsymbol{success/total \approx \pi/4}$ **or** $\boldsymbol{C/N \approx \pi/4}$

or $\boldsymbol{\pi = 4C/N}$

Since the area of the first quadrant of the unit circle is $\pi/4$.

Note: In your program, you must implement the reduction operation, ***Sum-reduce***, as shown in the Figure.



[4] **[20 Marks]**

i) In the parallel deep Convolutional Neural Network (CNN) if the input image is 255 x 255 x 3, two filters are used of sizes 11 x 11 at stride 4 and 9 x 9 at stride 3 respectively.

a) How many output tensors are generated from the 1st convolution layer? What are its sizes?

b) What are the total number of parameters?

ii) Consider the problem of parallelizing the application of a 3x3 convolution kernel for sharpening an image. The kernel is a square matrix (3x3) with weights that are used in the calculation of the new pixel data. Convolution between a kernel K of odd size n and an image f is defined by the formula:

$$g(x.y) = \sum_{i=-n2}^{n2} \sum_{j=-n2}^{n2} k(n2+i, n2+j) f(x-i, y-j)$$

Where $n2 = \left\lfloor \frac{n}{2} \right\rfloor$, floor function.

If, for example, a 3x3 kernel is used:

$$K = \begin{vmatrix} k_{0.0} & k_{0.1} & k_{0.2} \\ k_{1.0} & k_{1.1} & k_{1.2} \\ k_{2.0} & k_{2.1} & k_{2.2} \end{vmatrix}$$

then for each pixel at row *i* and column *j*, the new pixel value $v'_{i.j}$ resulting from the convolution is determined by the values of the pixel and its eight neighbors according to the formula:

$$\begin{aligned} v'_{i.j} &= v_{i-1.j-1} \cdot k_{2.2} + v_{i-1.j} \cdot k_{2.1} + v_{i-1.j+1} \cdot k_{2.0} \\ &+ v_{i.j-1} \cdot k_{1.2} + v_{i.j} \cdot k_{1.1} + v_{i.j+1} \cdot k_{1.0} \\ &+ v_{i+1.j-1} \cdot k_{0.2} + v_{i+1.j} \cdot k_{0.1} + v_{i+1.j+1} \cdot k_{0.0} \end{aligned}$$

Where v are original pixel values.

1- State the steps how to implement the application (ii) in parallel using GPU

2- Write the pseudocode for applying the convolution kernel to an image as stated in (ii).