MONASH University

FIT3143 Lab Week 4

Lecturers: ABM Russel (MU Australia) and Vishnu Monn (MU Malaysia)

OPENMP

OBJECTIVES

- The purpose of this lab is to introduce you to OpenMP
- Practice OpenMP

INSTRUCTIONS

- Download and set up the Linux VM [Refer to Lab Week 1]
- Setup eFolio (including Git) and share with tutor and partner [Refer to Lab Week 1]

TASK

DESCRIPTION:

- Introduction to OpenMP

WHAT TO SUBMIT:

1. Screenshot of the running programs and git repository URL in the eFolio. Screenshot of the running programs and git repository URL in the eFolio.
2. Code in the Git.

EVALUATION CRITERIA

- This Lab-work is part of grading
- Code compile without errors (2), well commented (2), lab-work questions fully answered (4), analysis/report is well formatted (2) = 10 marks

LAB ACTIVITIES

1. OpenMP

(BASED ON TUTORIAL BY ASAD KHAN)

The aim of this task is to develop a gaming machine using OpenMP *workshare* programming model. The machine has a 10-digit display. The following rule applies to register a win with the machine.

At least two of the displayed digits must represent the same value to register a win.

The machine may generate more than one win after a play. It will depend on the number of (different) repeated digits appearing on the display, Figure 1.

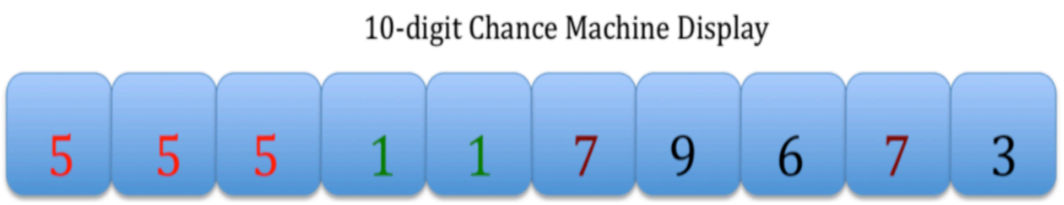

Figure 1: Number '5' appears thrice (a win), number '1' appears twice (a win), and number '7' appears twice as well (a win). The total number of wins counted by the machine in this case will be 3.

The machine is to be controlled by a parallel software code. The code specifications are as follows.

- The code forks four concurrent threads. With *chunksize* = 2.
- Each thread executes an independent pseudorandom number generator function that produces random numbers in the range of 1 to 25.
- The threads work on a shared integer array of size 10 to fill each of the array elements with a random number.
- The threads join to form the master thread.
- The array elements are checked for matching entries. A win is counted if two or more occurrences of a number are found in the array. The total number of such occurrences is displayed on the terminal screen.

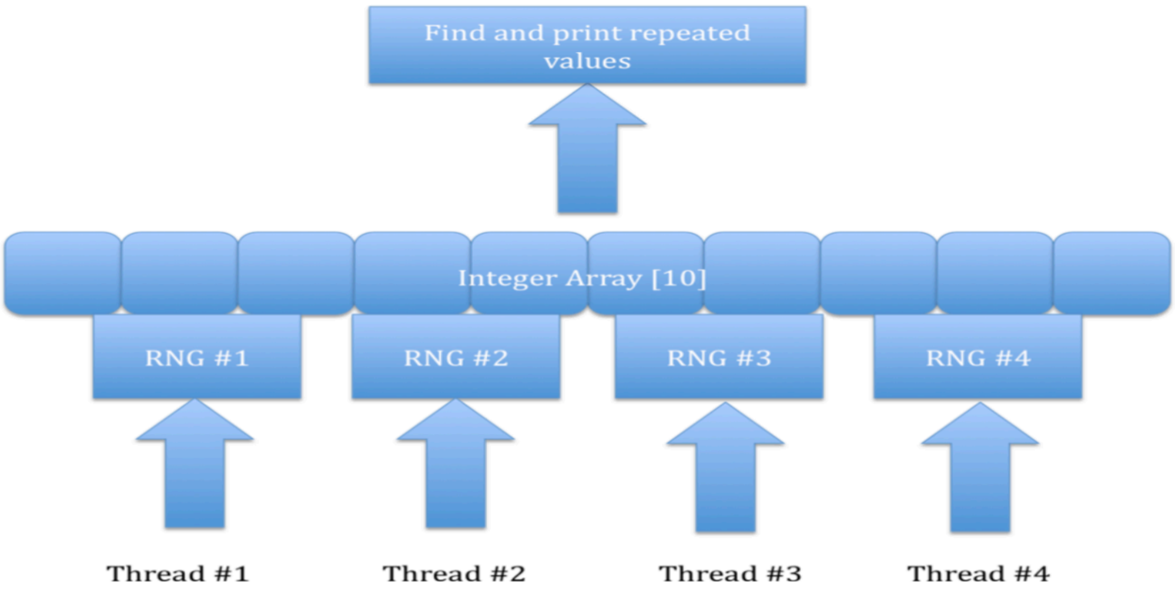

Figure 2:The code creates four concurrent threads to workshare. Each thread independently generates random number over the prescribed range. The threads work collaboratively and in parallel to populated the shared integer array. Once the array is fully populated, the threads join to form a single (sequential) master thread. The master thread finds the repeated values in the array and displays the result.

Discuss your solution in relation to parallel code to count the number of wins.