HPC Laboratory Assignment

Lab 5: OpenMP (part II)

J.R Herrero and M.A. Senar

Spring 2023/24

Hybrid execution of MPI-OpenMP applications

Given the *hellohybrid.c* program (a parallel application that uses threads and processes combining calls to OpenMP and OpenMPI libraries) execute it using the following combinations of nodes and processes; in all cases you should use all the cores available in a node; each node from the cluster has two CPUs (two sockets) and each CPU has 6 cores. That means that, in terms of SLURM resources, each node has 12 CPUs.

Combinations:

- 1 node running a single MPI process with 12 threads.
- 2 nodes running a single process in each node and each process is divided into 12 threads.
- 2 nodes running two processes each, with 6 threads per process.
- 2 nodes running 24 single-threaded MPI processes (12 per node).

Compilation of hellohybrid.c:

```
module load gcc/12.1.1

module load openmpi/4.1.1

mpicc -o hellohybrid -fopenmp hellohybrid.c
```

A submit file sample (*hellohybrid.slurm*) is also provided. You should modify it according to each execution case. All files are available at Escritorio/alumnos/OpenMP_2.

Parallelization of a program to compute Subarrays with 0 Sum

Given an array of positive and negative numbers, the task is to find if there is a subarray (of size at least one) with **0** sum.

```
Input: {4, 2, -3, 1, 6}
Output: true (there is a subarray with zero sum from index 1 to
3).

Input: {4, 2, 0, 1, 6}
Output: true (the third element is zero. A single element is also
a sub-array).

Input: {-3, 2, 3, 1, 6}
Output: false
```

A sample C program (*sum0.c*) is provided (it is located et Escritorio/alumnos/OpenMP_2) that generates a random list of integers and checks whether a 0-sum subarray exists or not.

Starting from this program, the following changes must be made:

1. modify the program to count **ALL** 0-sum subarrays that exist in a given list of integers.

   ```
   Input: { -9, -6, -1, 9, -2, 0, 0, -1 }
   Output: 6  ({0},{0},{0,0},{-6,-1,9,-2},{-6,-1,9,-2,0},
               {-6,-1,9,-2,0,0})
   ```

2. parallelize the function that counts the number of 0-sum subarrays using the OpenMP library. The program must have two versions of the same function: the sequential and the parallel, to verify that the program provides the same correct results in both cases.

`In your tests,` you have to run the program with larger arrays and different number of threads to ensure its correct behavior.

```c
// C program to find if there is a zero sum subarray

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define N 2000
int sum[N];

bool subArrayExists(int arr[], int n)
{
    for (int i = 0; i < n; i++) {
        int sum = arr[i];
        if (sum == 0)
            return true;
        for (int j = i + 1; j < n; j++) {
// we are finding the sum till jth index starting from ith index
            sum += arr[j];
            if (sum == 0)
                return true;
        }
    }
    return false;
}

// Driver's code
int main()
{
  int seed=2;
  int x, i, upper=10, lower=-10;

  srand(seed);
  printf ("\n");
  for (i=0; i < N; i++){
     x = (rand() % (upper - lower + 1)) + lower;
     sum[i] = x;
     printf ("%d ", x);
  }
    // Function call
  if (subArrayExists(sum, N))
      printf ("\n \n Found a subarray with 0 sum\n");
  else
      printf ("\n \n No Such Sub Array Exists!\n)";
  return 0;
}
```

Deliverable

After the session for this laboratory assignment you will have to deliver a compressed file that includes a report in PDF format (other formats will not be accepted) containing a brief description of all the solutions achieved as well as any other relevant issue related to it. Your compressed file should include also the source code of your program. In your report document, provide screenshots or any equivalent method that shows the correct outputs generated by the program.

Only one file has to be submitted per group through the course website. Late submissions will be accepted but a deduction between 10%-30% of the maximum mark available from the actual mark achieved by the student shall be imposed. Important:

- Please, follow the same recommendations that we made for the previous deliverables.
- In the front cover of the document, please clearly state the name of all components of the group, the identifier of the group (username bio-hpcXX), title of the assignment, date, academic course/semester,... and any other information you consider necessary.