PARALLEL AND DISTRIBUTED COMPUTING

2014/2015 — 1st Semester

1st Exam — January 6th, 2015 — Duration: 2h00

- No extra material allowed. This includes notes, scratch paper, calculator, etc.
- Give your answers in the available space after each question. You can use either Portuguese or English.
- Be sure to write your name and number on all pages, **non-identified pages will not be graded!**
- Justify all your answers.
- Do not hurry, you should have plenty of time to finish this exam. Skip questions that you find less comfortable with and come back to them later on.

I. (1,5 + 1 + 2,5 = 5 val.)

1. Write the code to parallelize the outer loop of the following program using OpenMP commands. Assume that function `random1000()` returns a random number between 0 and 999.

```
for(i = 0; i < N; i++){
    M = random1000();
    for(j = 0; j < M; j++)
        a[i][j] *= 2.0 + a[i][j];
}
```

2. In the OpenMP directive `omp parallel for` what is the purpose of the, optional, `chunk` parameter? When should a larger or smaller value of `chunk` be used?

3. Compare from a programmer's point of view a UMA machine (uniform memory access) versus a DSM machine (distributed shared memory). Make sure you explain the mechanisms that support these systems.

II. (1,5 + 0,8 + 0,6 + 0,6 + 1,5 = 5 val.)

1. Consider that we want to parallelize the following code:

```
for(i = 0; i < N; i++)
    b[i] = DoComputation(a[i]);
```

Write an MPI program that implements this code using a workpool (aka master/slave) architecture. To simplify, assume that there are only 2 slave processes. (you don't need to worry about the detailed syntax or order of the parameters of the MPI functions, but be sure to indicate all the relevant parameters)

2. a) What is the purpose of the tag parameter in MPI's `MPI_Send` function?

b) Is there a way to circumvent this "restriction"? Indicate a situation when this may be helpful.

c) Discuss why you should not avoid using this tag parameter.

3. In a good MPI implementation, what is the complexity of the `MPI_AllGather` function? Justify.

III. (2 + 1,5 + 0,5 + 1 = 5 val.)

1. Consider a problem with a sequential algorithm that runs in $\Theta(n \log^2 n)$ and with a parallel implementation that runs in $\Theta(\frac{n}{p} \log^2 n)$ with p processors and whose overhead (communication + redundant computation) per processor is given by $\Theta(n \log n)$. If the required memory grows with n^2, compute the scalability function for this parallel algorithm. Discuss the result obtained.

2. Although the Amdahl's Law defines a ceiling for the maximum speedup of a parallel program, the same speedup can be computed using the Gustafson-Barsis' Law which presents no such limitation. Discuss the main underlying difference between these two expressions that explains this contradiction.

3. a) What is the value of the Experimentally Determined Serial Fraction for a system with ideal speedup?

   b) In general, why does the value of the Experimentally Determined Serial Fraction tend to increase?

IV. (1 + 1 + 1,5 + 1,5 = 5 val.)

1. a) Why are Monte Carlo methods amenable to parallel programming?

   b) What is the main difficulty when using Monte Carlo in parallel programming?

2. Combinatorial search is usually performed through a tree. In a parallel implementation, typically each processor is assigned a subtree to search. Discuss what is the main issue that needs to be addressed, and how is this generally handled.

3. What is the main reason why the Foster's design methodology is popular? Give a brief description of each step, indicating its main objective and how to achieve it.