

CS F422 Parallel Computing

Project – Exercises

Exercise I

Consider the problem of computing the following operation in parallel:

$$A_1 \text{ OR } A_2 \text{ OR } \dots \text{ OR } A_k$$

where each A_i is a task computing a Boolean value. Since only one result is required all tasks can be terminated as soon as one A_j results in TRUE for some j unless all tasks results in FALSE.

Design and implement a template parallel algorithm to run on a message passing system where each node is in turn a shared memory (multi-core) system. Your design should handle the following three cases:

- a) the computation is SPMD: i.e. all A_i are doing the same computation – only data are different
- b) the computation is MPMD: i.e. each A_i may be different from the others
- For each case, schedule computations given p nodes, with c cores per node.
- For case (a), provide an interface such that data partitioning can be skewed:

e.g. if each A_i is a search operation on a range of elements in a list then the size of the range can be uniform across the tasks; but

e.g. if each A_i is trying to find a prime number in a range of numbers by testing each number for primality, then the size of the range should depend on the size of the numbers because primality testing will take time that is a function of the size of the numbers;

- For case (b), provide a design such that load balancing can be done at run-time. Process migration need not be pre-emptive.

Demonstrate your template with full solution for at least two examples for each of the two cases. Design your own interface for assigning computation and data to your tasks. Your interface has to be simpler and at a higher level than that of MPI.

You may use any existing sequential solutions for example tasks such as searching or primality testing.

Your implementation must use MPI for message passing and either PThreads or OpenMP for shared memory computations.