Université Grenoble Alpes, Grenoble INP, UFR IM^{2}AG

Master 1 Informatique and Master 1 MOSIG

UE Parallel Algorithms and Programming
TD # 3

F. Desprez (Inria), J-F. Méhaut (UGA/Polytech), T. Ropars (UGA/IM^{2}AG), E. Saillard (Inria)

March 14, 2017

Exercise 1: Adding n integers

In this exercise, we want people to collaborate to add n integers. One person should get the final result, and does not need to make the others aware of it.

1. If one person can add two integers in a time t_c, how much time will it take her to add n integers?
2. How much time will it take at best if their are 8 persons sitting in a circle? Here we assume that one can only communicate with her neighbors. Sending a message to a neighbor takes time t_s.
3. How much time will it take at best if the 8 persons are sitting in two rows of 4 persons? Here we assume that one can only communicate with her left and right neighbors, plus the person in front or behind her.
4. Propose another configuration to make the people sit so as to minimize the total time to run the computation.

Exercise 2: Broadcast in a ring

1. We consider a ring of n processes with a single one-way link between neighbors in the ring. Describe the algorithm to broadcast a message from process k to all processes in the ring. To describe the algorithm you can use the following functions:
   - `MY_NUM()` returns the *id* of the local process
   - `NUM_PROCS()` returns the total number of processes in the ring.
   - `SEND(m, id)` sends message m to process *id*.
   - `RECV(&m, id)` receives a message from *id* and stores it into m.
2. Compute the time required to run the algorithm, assuming store-and-forward communication. We note M the size of the message; L the latency on a link; B the bandwidth of a link.
3. We consider now two-ways links. Give the new version of the algorithm and compute its execution time.

4. Finally, we consider a single row of n processes. We want to compute the time required to send a message from the first to the last process in the row. To optimize performance, we are going to apply pipelining by dividing the message into r packets. To better understand how this may work, we suggest you to try reasoning on an example with, for instance, $n = 4$ and $r = 5$.

Exercise 3: Network topology

We consider the network topology described in Figure 1.

1. How many nodes and links are included in the topology?
2. What is the degree of the nodes and the diameter of the network?
3. What is the bisection width of this network? We recall that the bisection width can be defined as the minimum number of links to cut to partition the network into parts of equal size.

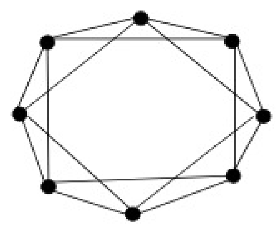

Figure 1: Network topology

We now consider another topology, namely a grid with $m \times n$ processors ($n \leq m$).

1. What is the bisection width is m is even?
2. What is the bisection width is m is odd?