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

Master 1 Informatique and Master 1 MOSIG

UE Parallel Algorithms and Programming

TD # 3

Exercise 1

Below an improved solution for question 1.2.

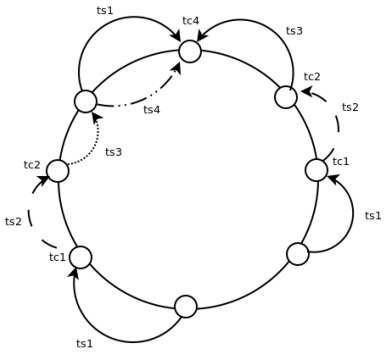

Figure 1: Solution to exercise 1.2

- at ts1: 3 sends to 2; 4 sends to 5 and 7 sends to 0
- at tc1: 2 computes its local sum; 5 computes and 0 computes
- at ts2: 5 sends to 6; 2 sends to 1
- at tc2: 6 computes; 1 compute
- at ts3: 1 sends to 0; 6 sends to 7
- at tc3: 0 computes
- at ts4: 7 sends to 0
- at tc4: 0 computes

We notice that tc3 and ts4 can be run in parallel !!

$$T = 3(ts + tc) + max(ts, tc)$$

Exercise 2

Bellow a possible answer for Exercise 2, question 3: Algorithm to broadcast a message in a ring with two-way links.

```
1  /* source is k*/
2  Bcast(M, k)
3  {
4    my_id = MY_NUM();
5    nb_procs = NUM_PROCS();
6
7    /* now root of bcast is 0 */
8    id_shift = (my_id - k)% nb_procs;
9
10   /* if I'm in the first half of the processes, I recv from left and I
11   send to the right*/
12   if(id_shift < nb_procs/2){
13     from = (my_id -1) % nb_procs;
14     to = (my_id + 1) % nb_procs;
15   }
16   else{
17     /* recv from right and send to left */
18     from = (my_id + 1) % nb_procs;
19     to = (my_id - 1) % nb_procs;
20   }
21
22   if(my_id = k){
23     /* send an initial message in both directions */
24     SEND(M, (k-1)% nb_procs));
25     SEND(M, (k+1)% nb_procs));
26   }
27   else{
28     /* if I'm the last process on one subpart, I only recv */
29     if(id_shift == (nb_procs/2)-1 || id_shift == nb_procs/2){
30       RECV(&M, from);
31     }
32     else{
33       RECV(&M, from);
34       SEND(M, to);
35     }
36   }
37 }
```