Assignment Nbr. 1

Given two processors $P0$ and $P1$ the time needed to send a message from $P0$ to $P1$ is a function of the *message length* $T_{\rm comm} = T_{\rm comm}(n)$, where n is the number of bits in the message. If we approximate this relation by a *linear relation*, then

$$T_{\rm comm}(n) = l + n/b \tag{1}$$

where l is the *latency* and b is *bandwidth*. Both parameters depend on the hardware and software of the network (TCP/IP layer in Linux), and the message passing library (MPI).

The task consists in launching a parallel run in the cluster with 3 compute nodes and find the latency and bandwidth between each pair of the processes in the parallel run.

Notes:

- **Each node of the cluster has 8 cores, so you have to launch the parallel run with `NSLOTS=24`.**
- **Beware that some processes may be in the same computing node. For instance if each node has 8 cores, then MPI processes 0-7 will reside in**

the first computing node, 8-15 in the second, and 16-23 in the third. The latency and bandwidth will be higher between processes in that reside in the same node that between processes that reside in differente nodes.

- **You can start from the code *bw.cpp*, in */clonetroop/master1/mstorti/bw*. The program only sends a message roundtrip and computes the bandwidth (not the latency). You have to modify this program so as to compute the latency by using different include the more relevant formulas here. If you have a set of measurements n_j and $T_j = T(n_j)$ ($j = 1, \ldots, N$, where N is the nuber of measurements) then you can compute the bandwidth and latency as the parameters of (1) with the following expressions**

$$1/b = \frac{\langle Tn \rangle - \langle T \rangle \langle n \rangle}{\langle n^2 \rangle - \langle n \rangle^2},$$

$$l = \langle T \rangle - \langle n \rangle / b,$$

$$\langle n \rangle = 1/N \sum_j n_j, \quad \langle T \rangle = 1/N \sum_j T_j, \quad \langle Tn \rangle = 1/N \sum_j T_j n_j,$$

$$\langle n^2 \rangle = 1/N \sum_j n_j^2.$$

- **Don't take too large n, I suggest something like $n = 1000$ to 10000 in steps of 1000.**
- **It's better to do some *statistics*, i.e. if you want to to measure $T(n)$ for $n = 1000$ you send and receive the buffer 100 times, and then you take an average of this time.**

```
1  // j,l is a pair of processes (j!=l)
2  int ntimes=100;
3  start = MPI_Wtime(); // Start chrono
4  for (int k=0; k<ntimes; k++) {
5    if (myrank==j) {
6      MPI_Send(buff1,...,l);
7      MPI_Recv(buff2,...,l);
8    } else if (myrank==l) {
9      MPI_Recv(buff2,...,j);
10     MPI_Send(buff1,...,j);
11   }
12 }
13 // Stop chrono, compute average
14 elapsed = (MPI_Wtime()-start)/double(ntimes);
```

- **In order to compute the bandwidth and latencies between all the 24 processes you need to do two nested loops like this**

```
1  // Loop over DISTINCT pairs (j,l) in [0,size)
2  for (int j=0; j<size-1; j++) {
3    for (int l=j+1; l<size; l++) {
4      start = MPI_Wtime(); // Start chrono
5      if (myrank==j) {
```

```
6         MPI_Send(buff1,...,1);
7         MPI_Recv(buff2,...,1);
8       } else if (myrank==1) {
9         MPI_Recv(buff2,...,j);
10        MPI_Send(buff1,...,j);
11      }
12      elapsed = MPI_Wtime()-start; // Stop chrono
13    }
14  }
```


(docver "texstuff-1.2.1-137-gb5b81fb") (docdate "Mon Feb 24 11:00:24 2014 +0100") (procdate "Wed Feb 26 19:23:37 2014 +0100")