Assignment 2- Messaging Passing Interface
CSCI 596: Scientific Computing & Visualization

Anup V Kanale

September 18, 2017

This assignment deals with implementing a global summation program using `MPI_Send` & `MPI_receive` commands. Functionally, this is the same as using `MPI_AllReduce`. Since this job has to be run on the USC HPC clusters, I will first provide a minimal overview of how to run any program on the HPC cluster.

1 Using the HPC cluster

With respect to this assignment, the following files are needed to run a job on the HPC cluster–

1. C code
2. pbs file

Firstly, compile the program using `mpicc` compiler and generate the executable. For example, `mpicc -o global global.c` complies the program `global.c` and generates the executable `global`. Once this is done, we have to queue this job to run on a given number of processors, say `np` using a `.pbs` (**P**ortable **B**atch **S**ystem). To this end, the `.pbs` file contains information regarding the path to the executable and the wall time request.

The slightly tricky part here is that everytime I log into the hpc account, the `mpicc` compiler has to be setup using a shell script (`source /usc/usc/openmpi/default/setup.sh`), and the pbs file creates a new terminal session for the same user account. Therefore, it is important to include even the line to setup the mpi compiler. This can be done either in the pbs script, or in the `bashrc` file. The latter is recommended as mpicc will be used regularly.

Finally, submit the job using the command `qsub <filename>.pbs`. To query the status, use `qstat -u <user-name>`

2 Global Summation using `MPI_Send` & `MPI_Recv`

This program uses the butterfly communication structure, refer to the question sheet and lecture notes on MPI for details. Each process contributes a partial value, and at the end, all the processes will have the globally summed value of these partial contributions.

The source code, and the results are shown below.

```c
double global_sum(double partial) {
  /* Implement your own global summation here */
  double hisdone, mydone, partner;
  int bitValue;
  MPI_Status status;

  mydone = partial;
  for (bitValue=1; bitValue<nprocs; bitValue*=2){
    //partner := myid XOR 2-to-the-power-l;
    partner=myid^bitValue;
    //send mydone to partner;
    MPI_Send(&mydone,1,MPI_DOUBLE,partner,bitValue,MPI_COMM_WORLD);
    //receive hisdone from partner;
    MPI_Recv(&hisdone,1,MPI_DOUBLE,partner,bitValue,MPI_COMM_WORLD, &status);
    mydone = mydone + hisdone;
  }
  return mydone;
}
```

Figure 1: C program

```
 GNU nano 2.3.1                                              File: global.out

----------------------------------------
Begin PBS Prologue Mon Sep 18 19:36:37 PDT 2017
Job ID:           24275856.hpc-pbs1.hpcc.usc.edu
Username:         kanale
Accountname:      lc_an2
Group:            me-ar
Project:          lc_an2
Name:             global
Queue:            quick
Shared Access:    no
All Cores:        no
Has MIC:          no
Is hsda:          false
Nodes:            hpc1015 hpc1120
Scratch is:        /scratch
TMPDIR:           /tmp/24275856.hpc-pbs1.hpcc.usc.edu
End PBS Prologue Mon Sep 18 19:37:11 PDT 2017
----------------------------------------
Node 1 has 1.000000e+00
Node 3 has 3.000000e+00
Node 0 has 0.000000e+00
Node 2 has 2.000000e+00
Global average = 1.500000e+00
Node 3 has 3.000000e+00
Node 2 has 2.000000e+00
Node 0 has 0.000000e+00
Node 1 has 1.000000e+00
Node 6 has 6.000000e+00
Node 7 has 7.000000e+00
Node 5 has 5.000000e+00
Node 4 has 4.000000e+00
Global average = 3.500000e+00
----------------------------------------
Begin PBS Epilogue Mon Sep 18 19:37:15 PDT 2017
Job ID:          24275856.hpc-pbs1.hpcc.usc.edu
Username:        kanale
Accountname:     lc_an2
Group:           me-ar
Job Name:        global
Session:         27105
Limits:          neednodes=2:ppn=4,nodes=2:ppn=4,walltime=00:00:59
Resources:       cput=00:00:07,energy_used=0,mem=33888kb,vmem=517740kb,walltime=00:00:02
Queue:           quick
Shared Access:   no
Has MIC:         no
Is hsda:         false
Account:         lc_an2
End PBS Epilogue Mon Sep 18 19:37:41 PDT 2017
----------------------------------------
```

Figure 2: Results for running the global summation program on 4 and 8 processors