ACM40640/PH504 Practical 2

ICHEC

2022/23 Spring

1 Scalability: strong and weak scaling

When using HPC clusters, it is almost always worthwhile to measure the parallel scaling of your jobs. The measurement of strong scaling is done by testing how the overall computational time of the job scales with the number of processing elements (being either threads or MPI processes), while the test for weak scaling is done by increasing both the job size and the number of processing elements. The results from the parallel scaling tests will provide a good indication of the amount of resources to request for the size of the particular job.

1. Copy the C or Fortran codes under Practical folder of Week 2 from Brightspace/Blackboard: inc_serial.c, inc_omp.c and inc_mpi.c or inc_serial.f90, inc_omp.f90 and inc_mpi.f90.

2. Compile and run these codes using length=10^6, 10^7, 10^8 and 10^9 in a number of times (i.e. 1000) for $1, 2, 4, 8, 12$ and 20 processing elements to get the average execution time. The instructions on how to compile and run are available at each file.

   - Study the scalability of the openmp/mpi codes. Observe the change in computation time and communication time with the number of proccesses in mpi codes.
   - Is there a good scalability? What factors can be limiting the scalability?
   - Why do we need to run the computation in a number of times?

3. Calculate relative speedup and efficiency in openmp/mpi codes for given length and threads/processes above. Plot the curves with your favorite charting tool. Observe Amdahl's law.

4. Calculate absolute speedup at a few different problem sizes and processing elements. Observe Gustafson-Barsis' law.

Note: sciprog is a 2-core machine so you may not observe a speedup. In this case, you can use the results obtained on ICHEC cluster. See files inc_serial_out.txt, inc_omp_out.txt and inc_mpi_out.txt in the Practical folder.