Iowa High Performance Computing Summer School 2013

Parallel Performance and Optimization Exercises

Wednesday, May 22, 2013

NOTE: For many of these exercises, it is probably better to use Moffett instead of Helium, because Moffett has more than 4500 cores and is usually very lightly loaded.

1. Strong Scaling of `HYDRO`
   Perform a strong scaling test using the example program `HYDRO`. An example input file, `ss1a.in`, can be found on the website under "Exercises"
   `http://www.physics.uiowa.edu/~ghowes/teach/ihpc13/exercises.html`
   in the tar file `hydro_input_scaling_exercise.tar`.
   (a) Run the strong scaling for 2, 4, 8, 16, 32, 64, 128, and 256 processors.
   (b) Present your strong scaling test results in graphical form using `gnuplot` (see the `HYDRO` documentation for instructions on using `gnuplot` for plotting data and producing a postscript figure of the result).

2. Weak Scaling of `HYDRO`
   Perform a strong scaling test using the example program `HYDRO`. An example input file for `nproc=2`, `ws1a.in`, can be found on the website under under "Exercises"
   `http://www.physics.uiowa.edu/~ghowes/teach/ihpc13/exercises.html`
   in the tar file `hydro_input_scaling_exercise.tar`.
   (a) Run the weak scaling for 2, 4, 8, 16, 32, 64, 128, and 256 processors. Be sure to determine a reasonable way to double the problem size for each case.
   (b) Present your weak scaling test results in graphical form using `gnuplot` (see the `HYDRO` documentation for instructions on using `gnuplot` for plotting data and producing a postscript figure of the result).
   (c) For a single doubling, do the results differ if the grid is doubled in the x direction rather than the y direction? Can you think of any reason why this may be? To see if there is a difference, try running a series of scaling runs from 2 to 256 processors doubling the size only in x, and then repeat the series doubling only in y.

3. Compute the load balance in `HYDRO` by adding some communication between processors at the end of the code to pass the timing statistics. Consider the following issues:
   - Each processor computes its own timing statistics independently.
   - Which of the timing categories do you want to use to determine the load balance?

4. Using the code you wrote in the MPI Programming Exercises to determine numerically the value of π, run the profiling tools `PAPI` and `MpiP` to determine a general overview of your code's performance and the MPI time and load balance. Run at least 16 processors. Note that you must use Moffett to do this exercise since Helium does not have the profiling tools `PAPI` and `MpiP` installed.

5. Advanced: Modify `HYDRO` to enable domain decomposition in two dimensions. Consider the issue of how to achieve good load balancing.