

CS 140 Assignment 2: Monte Carlo integration

Assigned January 17, 2008

Due by 11:59 pm Thursday, January 24

The goals of this assignment are, first, to make sure you're able to log onto DataStar and compile and run MPI programs and the VAMPIR tool; and, second, to use MPI's broadcast and reduce functions in an embarrassingly parallel code. You will do this by running and then modifying the Monte Carlo pi program that Stefan demoed in class on January 15. (Note, this is *not* the same as the pi program I used as in example in class on January 17.)

This is a pretty simple assignment, just to get you going on DataStar, but please start on it right away so there's time to fix any difficulties with your DataStar directories and accounts.

1 What experiments to do

You should start with Stefan's `parpi.c` program, which is at

<http://www.engineering.ucsb.edu/~stefan/examples/parpi.c>

Do the following:

1. Compile and run the program on four processors. How far is the answer (that is, the overall average answer) from the real value of $\pi \approx 3.1415926535897932$? Turn in the output (not the source code, which you shouldn't have to change).
2. Compile and run the program on various different numbers of processors, including at least $p = 1, 4, 8, 12,$ and 16 (try 32 if you can). For each run, record the running time and the error in your approximation to π . Make a table (and a plot if you want) that shows, for each p , (1) the run time; (2) the speedup (the time for $p = 1$ divided by the time for this p); (3) the error in the answer. What do you observe about the speedup and the error? Turn in your table (and plots if any) and your observations.
3. Recompile the program with VAMPIR and use VAMPIR to make plots of the global timeline display as Stefan showed, rather like the one at

<http://www.nersc.gov/nusers/resources/software/tools/vampir.gif>

for $p = 4$ and $p = 16$. Turn in the VAMPIR plots (in any standard format so we can see it).

4. Modify the source code so that it uses `MPI_Bcast` and `MPI_Reduce` instead of `MPI_Send` and `MPI_Recv`. (In the modified code you don't need to print every process's estimate of π , just the overall average estimate.) Compile your new code with VAMPIR and run it for $p = 4$ and $p = 16$, and make the same VAMPIR plots as in part (3). How do the runtimes and errors compare with the original version? How do the VAMPIR plots compare? Turn in the modified source code and the VAMPIR plots.

2 What to turn in

Turn in the output file from part (1), the table (and maybe plots) from part (2), the VAMPIR plots from part (3), and the modified source code and VAMPIR plots from part (4). Also turn in a README file containing a writeup that answers the questions above, and that lists and clearly identifies the other files you are turning in.

Use `ssh` to copy the files from DataStar to CSIL, and then use the `turnin` command from CSIL to turn the files in.

I recommend that you do this first programming assignment alone, so that you can verify that your own DataStar account is set up correctly; but, following the course policy, you may do it in groups of two if you wish.