CSCI596 Assignment 3—Parallel Computation of $\pi$
Due: September 25 (Mon), 2017

The purpose of this assignment is to acquire hands-on experience on the scalability analysis of a parallel program — one of the key skills you learn in this class. We use a simple application that utilizes the function you have written for assignment 2 (where the purpose was to obtain a confidence that $\text{MPI\_Send()}$ and $\text{MPI\_Recv()}$ are sufficient to build any parallel programs, using a concrete example of global reduction.)

Part I: Programming

Write a message passing interface (MPI) program, $\text{global\_pi.c}$, to compute the value of $\pi$ based on the lecture note on “Parallel Computation of Pi” and using the $\text{global\_sum()}$ function you have implemented in assignment 2.

(Assignment)
1. Submit the source code of $\text{global\_pi.c}$.

(Note)
- Insert $\text{MPI\_Wtime()}$ function (which takes no argument and returns the wall-clock time in seconds as double) to measure the running time of the program.

Part II: Scalability

In this assignment, we measure the scalability of $\text{global\_pi.c}$.

(Assignment)
2. (Fixed problem-size scaling) Run your $\text{global\_pi.c}$ with the fixed number of quadrature points, $N_{\text{BIN}} = 10^7$, varying the number of compute nodes = 1, 2, 4 and 8 with processor per node 1 (i.e., the number of processors $P = 1, 2, 4$ and 8). Plot the fixed problem-size parallel efficiency as a function of $P$.

3. (Isogranular scaling) Run $\text{global\_pi.c}$ with the constant number of quadrature points per processor, $N_{\text{BIN}}/P = 10^7$, per processor for $P = 1, 2, 4$ and 8. Plot the isogranular parallel efficiency as a function of $P$. 