Project 1

Notes

- You do any one or more of the problems assigned.
- Only one problem will be graded if you do more.
- Each problem will earn you 20 points for this project.
- In your project report, please organize it in a packed PDF or a Zipped folder of files. Although you can be freely creative, please include the following sections in your report:
    1) A project description.
    2) A description of the main structure of your programs in algorithms or pseudo-code. Source code with compilation and execution instructions to enable grading.
    3) Results (numbers, tables, and figures, etc).
    4) Analysis of program performance and your other answers to specific questions given by the problem set.
- Due: September 29 (Friday) at 11:59 pm.
- No late projects will be accepted.

Problem 1.1

When running a parallel program on P processors to write out "Hello from Processor X" to you (who has one series way to receive the write out), the order is non-deterministic when you run the same program on the same system, multiple times.

For example, when running on a system of $P = 10$, you get the following print out:

Hello from Processor 3
Hello from Processor 9
Hello from Processor 8
…
Hello from Processor 0

Next time, repeating the same program again on the same computer, you get

Hello from Processor 0
Hello from Processor 8
Hello from Processor 7
…
Hello from Processor 4

This non-deterministic feature is sometimes undesirable and, too, rectifiable. Write a parallel program to rectify, i.e., whenever running your program, you always get the system to write out in a desired/fixed order, e.g.,

Hello from Processor 0
Hello from Processor 1
Hello from Processor 2
…
Hello from Processor 9

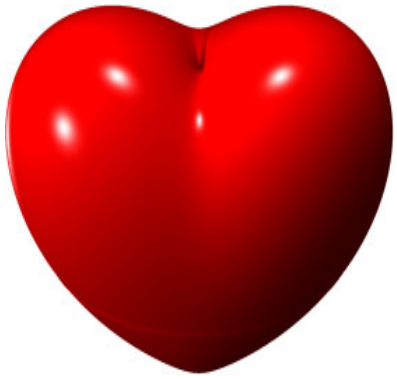
You may test it for a system with at least 13 processors.

One example of revealing many differences of parallel programs from the sequential ones.

Problem 1.2

Write a parallel program using any method for integration (preferably, the Monte Carlo method) to compute the volume of a heart expressed in equation

$$\left(x^2+\left(\frac{3}{2}y\right)^2+z^2-1\right)^3-\left(x^2+\frac{1}{50}\left(\frac{3}{2}y\right)^2\right)z^3=0$$



You perform minimum necessary calculations to achieve four digits of accuracy. How would you know you got this accuracy?

Now, please measure the number of floating-point operations in each core, for achieving similar accuracy as a whole, you need to obtain the results using the following cases of processor counts: $P = 1, 4, 16, 48$.

Some **additional thoughts** about this problem (and, unless you have the time/interests, you do not need to bother with this addition):

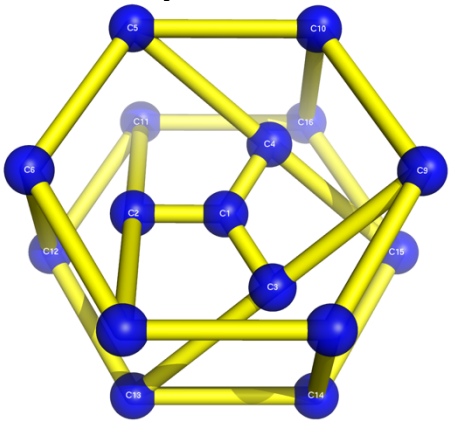
Obviously, what's given is the 3D surface equation of the heart and the assumption that the heart has no "cavities" and, thus, its density is uniform, etc. However, the heart pumps blood in and out and its volume, surface area, mass (including blood in it) is expected to vary with time. With such in mind, "crazy" problems can be hacked to make this problem "more" fun and harder, at least, for parallel computing.

1. Compute the surface area of the heart with the given surface eq.
2. Compute the heart's mass at time t if the heart mass density is
$$\rho(x,y,z;t)=\rho_0(x,y,z)\left(\frac{1+\sin\omega t}{2}\right)$$
where ω is the heart rate. Of course, I made up the formula and still do not know what $\rho_0(x,y,z)$ is. You may create a PDF to find it. The trivial case is $\rho_0(x,y,z)=1$.

You might even create a Ph.D. project out of this ☺. Did I have a heavy heart?!

Problem 1.3

In the following 3D graph, the 16 blue balls are considered as identical vertices and the lines are considered as bi-directional edges. Now, you need to broadcast N floating-point numbers from the central vertex (C1) to the entire system of 16 vertices including the source.



Please complete the following:

(1) Design algorithm(s) to quickly complete the broadcast of $N = 10^6$;
(2) Implement your algorithm(s), collect timing results, on a supercomputer by imitating the above given network topology with the basic non-collective light-weight basic single-sided MPI functions: MPI_Graph_create(), MPI_Isend(), MPI_Irecv() and a few other supporting functions.
(3) Call the MPI-provided MPI_Bcast() to carry out the broadcast and collect timing results.
(4) Repeat the above two steps for $N = 10^7$ and $N = 10^8$;
(5) Construct a table as follows and make comments:

N	My_BCast() Time (T1)	MPI_Bcast() Time (T2)	T1/T2
10^6			
10^7			
10^8			

Problem 1.4

Using online information, e.g., the latest Top500 list at www.top500.org, to analyze the #1 supercomputer.

Please follow the lecture, Hardware Introduction, to list all the features of the #1 supercomputer, with detailed definition of the terms you use. If possible, please find the data for its Linpack efficiency defined as Rmax/Rpeak, power efficiency, and its Graph500 score using BFS.

Please list at least three applications developed for the #1 supercomputer and describe briefly (~100 words) each application and why (briefly) it is a good choice.

Finally, construct the following table for the latest Graph500 computers.

Sys	BFS GTEPS	Rpeak	Rmax	Linpack Efficiency	CPU	Accelerator	Network	Power
#1								
#2								
#3								
…								
#10								