Some Ideas for Course Project

Please feel free to pick any of the following suggested ideas, or come up with variations of them, or any other idea that you want to work. The choice of the parallel paradigm will be up to you, either you can implement the algorithm that you pick in OpenMP, MPI or in OpenACC or in a combination of any of them or using all of them. Depending on the difficulty level and the flexibility of the algorithm for parallelization you can chose the respective parallelization paradigm.

1. Solution of a system of linear equations, denoted by, $Ax = b$, using multigrid method. You can work with full matrices or with sparse matrix representation as the case may be.

2. Implementation of Monte Carlo method with applications to either of (a) neutron transport in a homogeneous/heterogeneous medium (b) solution of partial differential equations (c) modeling of cell populations (d) sharpening of images produced by satellites etc.

3. Develop a parallel program for discrete Fourier transform with applications to either of (a) voice recognition (b) image processing (c) turbulence energy spectra etc.

4. Develop a parallel version of the classical lattice Boltzmann method (LBM) that solves the discrete Boltzmann equation on a specified lattice in a two-dimensional plane with applications to either of (a) steady state heat conduction equation (b) unsteady heat conduction with source terms (c) fluid flow in a lid-driven cavity (d) laminar jet flow etc.

5. Develop a parallel programs for (a) steepest descent method (b) conjugate gradient method (c) Bi-conjugate gradient method etc. You can showcase your results with applications to solution of two-dimensional or three-dimensional Poisson's equation.

6. Implement parallel cyclic reduction (PCR) using OpenACC and benchmark it with the existing in-built PCR routines provided by NVIDIA. Showcase your results with applications to calculation of derivatives using compact-schemes, and solution of other tri-diagonal systems that are encountered. You can use applications in both 1D and 2D.

7. Parallelize any machine learning algorithm with applications to big data. Showcase how your parallelization speeds up the training and prediction parts of the algorithm.

8. Implement a parallel molecular dynamics code with applications to interaction of particles/cells at micro scale. Consider scaling up your code to several hundreds of thousands of particles.

9. Develop a hybrid parallel program that combines both MPI and OpenMP to solve for unsteady heat conduction in 2D and 3D domains. Compare how does this new hybrid code perform in comparison to solely MPI or OpenMP code? What are the advantages of such an implementation. Provide limits on how many processes can be used in each of the paradigms.

10. Develop a parallel ray-tracing algorithm to display graphics on screen. Use this algorithm to display a chosen object that can be defined analytically. Extend your code to include classification of a given point (in the three-dimensional space) whether it lies inside/on/outside a given object.