Parallel Scientific Computing

Rationale

Computationally complex problems cannot be solved on a single computer. They need to be run in an environment of 100 to 1000 processors or more. Designing algorithms to efficiently execute in such a parallel computation environment requires a different thinking and mindset than designing algorithms for single processor computers. This course is designed to give the students the parallel computation perspective using the MPI framework.

Description

Computationally complex problems cannot be solved in a single computer either because they are combinatorially complex (NP-Hard) or because they are large involving much data such as very large matrices or much computation. The framework we use to solve these kinds of problems in parallel is called MPI, short for Message Passing Interface. We examine combinatorial problems such as Boolean Satisfiablility, Set Partitioning, Traveling Salesman and large problems such as might be in matrix multiplication or simulated annealing.

Topic List

- MPI Tutorial
- Amdahl's and Gustfason's Laws
- Matrix Multiplication
- Boolean Satisfiability
- Set Partitioning
- Simulated Annealing
- Graph Coloring

- Graph Betweenness
- Large Optimization Problems
- Student Presentations of Papers and their Programming Results

Learning Goals

- Learn how to design algorithms in parallel environments
- Learn how to use MPI in a parallel program
- Learn how to use MPI in solving
  - Clustering Problems
  - The Traveling Salesman Problem
  - The Set Partitioning Problem
  - Matrix Multiplication
  - Simulated Annealing
  - Optimization Problems
  - Graph Coloring
  - Graph Betweenness

Assessment

Every student will work on two different MPI programs to solve computationally complex problems of their own choosing. In the second half of the course they will report on their algorithms and the results of their programs. Grades will be based entirely on their programs and presentations.