Parallelizing Serial Programs

Parallelizing serial programs is often a complex task that requires rethinking and reformulating tried and tested algorithms. The goal of the project is parallelize two existing programs to increase their performance and ability to solve large problems. Various considerations will have to be examined, such as the effect of I/O and object-oriented programming, during the parallelization process.

The first program solves the advection equation in a straightforward serial fashion. A 2D partitioning algorithm to divide work among processors is required to achieve parallelization in a way that is efficient and scalable.

The second program, an implementation of the Navier-Stokes equations, is a large and complex piece of software that uses object oriented programming to achieve better design goals. Parallelizing the application will require carefully thinking about the most efficient form of processing among multiple nodes. This is in contrast to most parallel programs where the flow of program, although in parallel, is procedural in fashion. With objects and methods that may be arbitrarily re-used and called, the structure of the parallelization will be profoundly different from most parallel programs. I/O will also be a factor, as the program needs to generate and store a large quantity of data. Reforming this data into one file, after having been created on different processors, will be challenging, especially in terms of minimizing communication costs.

Outline
Week 4: Complete parallelization of advection program
Week 6: Spend time with Navier-Stokes code to understand how it works
Week 7: Identify pieces for parallelization and begin process
Week 9: Complete parallelization of code.
Week 10: Complete performance and scalability measurements on DataStar.