1 Biography

I am a first year graduate student in Bioengineering working in Shadden Research Group for Biomedical Computation. I graduated from Rensselaer Polytechnic Institute in May of 2013 with a B.S. in Mechanical Engineering. For my current research, I am helping make improvements and advancements to the software our research group currently uses to analyze blood flow in the human body; this software is called Simvascular. This software is a pipeline to take image data of patient-specific vasculature and output the solved flow for this geometry. The current solver uses the finite element method to solve for the complex flow found in the human body.

My motivation for taking this course is twofold. Number one, I hope to build on my knowledge of computational science. I took CS294-73 last semester, and I was able to gain a deeper understanding of computer programming. CS267 seems like the next logical course to help me build on this knowledge. In a heavily computational lab, and with the growing gap between memory and processing developments, knowledge of this subject seems necessary for the near future. Number two, the finite element solver in Simvascular is written for parallel architectures. In order to make improvements to the code, it will help significantly to understand the foundation of parallel computing.

2 Parallel Application Problem

2.1 Electrophysiological Model of the Heart. Accurately modeling processes is an extremely difficult task when it comes to Biomedical Computation for many reasons. First, there are very complex processes taking place that occur on many different scales. For example, for the electrophysiology of the heart, there are processes occurring on a very micro scale including cellular interactions. Also, on a larger scale, there is a reaction-diffusion system that governs the propagation of movement of the heart. With this massive difference in scale and very different processes, it is very difficult to combine them in such a way to accurately model the very complicated mechanisms occurring.

Also, with the very complex domain of the heart, time dependencies, and complex differential equations, solving a problem such as this takes a heavy computational cost. This is where the use of parallel processors in parallel computing comes into play.
2.2 The Alya System and the use of Parallelism. The Alya System is the code developed for high-performance computation mechanics (HPCM) at the Barcelona Computing Center, and the tools utilized to develop this electrophysiological model of the heart. The model makes use of the more macro scale reaction-diffusion mechanisms mentioned above.

The parallelization of the Alya System uses multiple levels of parallel methods. The outer layer uses the MPI tools for distributed memory between multicores. The inner layer makes use of shared memory within each multicores.

To solve the problem using the finite element method, two types of communication are necessary for parallel computation. Number one, communication between neighboring nodes to share array information is needed. Second, global information needs to be shared for things such as critical time step and convergence. The mesh for this problem is created using the automatic mesh partition tool, METIS. This tool works using a master-slave strategy for parallel processing.

The resulting model is a mesh of 17 million elements, and has a resolution of about .2 mm. For this mesh size, it took around 15 minutes clock time to solve this complex problem using 500 cores of the Marenostrum supercomputer in Barcelona. This supercomputer is currently ranked 465th in the Top 500.

Figure 1: Domain mesh performed by METIS (left) on an unstructured mesh (right) of the left ventricle

There is very little experimental work done to compare this computational work to, so no comparison to past work has been done. In terms of computation, the model runs efficiently using the tools described above. This work is noteworthy because parallelism has been used to help solve a very complex problem in a reasonable time. Previous computational work done that was similar took three weeks of computation time. In addition, by using the Alya System tools, the problem is set up well for expansion and addition of even more complex models.
References