Programming Massively Parallel Systems

1 Rationale

Computationally complex problems such as graphical representations of movement cannot be processed in a reasonable amount of time on a single CPU. Currently, most graphical computations and many scientific calculations involving large datasets and complex systems are run in a massively parallel environment. Designing algorithms to efficiently execute in both time and memory usage in such environments requires an understanding of concurrency and the hardware requirements of massively parallel systems, for example, Graphical Processing Units (GPUs). This course is designed to give the students an introduction to the concepts and usage of GPUs and the CUDA extensions to the C/C++ languages.

2 Description

A survey of the approaches to massively parallel computer applications with emphasis on using graphical programming units (GPUs) and the CUDA extensions to the C/C++ programming languages. Comparisons between multicore CPUs and multi-processor GPUs will be given. Issues such as organization of large data sets, memory usage, and communication concerns will be addressed. Different levels of concurrency will also be discussed with most the focus on thread level-concurrency. Also multiple data streams on a single GPU and multiple GPUs will be covered with quick reviews of OpenMP and OpenMPI usage. Standard problems will be discussed.

3 Topic List

Topics may include but are not limited to:

- The Graphical Processing (GPU) Capabilities
- CUDA
- Threading Concurrency

- Open MP
- Open MPI

4 Learning Goals

1. The student will understand the concept of concurrency in an environment involving many parallel processors. 2. The student will understand the relationship between programming using a traditional multicore-CPU versus using massively parallel GPUs. 3. The student will acquire an understanding of the importance of the memory model needed for massively parallel programming. 4. The student will gain experience programming with CUDA on a GPU. 5. The student will be introduced to how to use multiple GPUs connected to a single CPU and using multiple GPUs over a network.

5 Assessment

- At least five weekly or biweekly assignments 25%
- Each student will be expected to carry out a project transforming an existing serial code or writing a new parallel code to use CUDA on a GPU. The student will write a paper in a research format describing their project 25%
- One mid term exam 25%
- A final presentation of the project in class 25%