Alex Gerbessiotis
Spring 1999

# CIS 485: Introduction to Parallel Computing

## 1. General Information

The course is intended for computer science, mathematics and engineering students who are interested in parallel computing and parallel programming. The purpose of the course is to introduce what parallel computing involves and explain why efficient programming of a parallel machine can be a complex but rewarding task. At the completion of the course students will understand different parallel architectures, know how to think "parallel" and how to solve problems in parallel, and gain hand-on experience in parallel programming. As a parallel machine can be either an expensive massively parallel computer or a distributed cluster of PC workstations connected by a high-speed network, the techniques that will be introduced in this course would help students program both types of platforms.

## 2. Course Requirements

A student taking this course should have elementary ANSI-C programming experience (desirably, but not necessarily, under a UNIX environment) and some knowledge of introductory algorithms and data-structures (such as matrix multiplication and sorting).

Two quizzes (Q1 and Q2) will be handed out. They will contribute 200 points (100 points each) towards the final grade (1000 points maximum). Two or three programming assignments (PA1, PA2, and/or PA3) will be handed out. They will contribute 300 points towards the final grade. A midterm exam will cover the first part of the course (approximately first seven weeks of classes). The final exam will cover the second part of the course only (last seven weeks). Each of the two exams will contribute 250 points towards the final grade. The two exams will be closed book and closed notes (this may change, however). The duration of each of the two exams will be approximately 90 minutes.

## 3. Recommended Books

1. B. Wilkinson and M. Allen "Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers", Prentice Hall, 1999.
2. M. J. Quinn "Parallel Computing. Theory and Practice", McGraw Hill, 1994.

## 4. Contact Information

**Office:** CIS Department, GITC 4213.

**E-mail:** `alexg@cis.njit.edu`

**Tel:** (973)-596-3244

**Office Hours:** Mon and Wed 12:15-12:55, Mon 16:00-17:30

**Class Hours:** Mon/Wed 13:00-14:30, University Hall 108.

**Web-page:** `http://www.cis.njit.edu/~alexg/courses/cis485`

The most effective way of communication is by e-mail. Please, try not to use attachments in your e-mail; plain text is preferable.

**Course mailing-list setup**

Some time after the first lecture, each student is strongly advised to send the instructor an e-mail indicating that he/she attends the class (Subject: CIS 485 student, E-mail body: Full name). Using this information a mailing list will be created that will be used to inform students on various course developments.

## 5. Course outline- Projected Weekly Schedule

**W1** : Limitations of uniprocessor machines. Need for parallel computers: Grand Challenge Problems. What is a parallel computer? A taxonomy of parallel models and architectures. Amdahl's Law and Gustafson's Law. Interconnection networks and interprocessor communication. Parallel Computing Terminology.

**W2** : Parallel algorithm design and implementation: Past, Present and Future. A simple shared-memory model: The PRAM. PRAM Variants. Fundamental PRAM techniques (broadcasting, parallel-prefix, matrix multiplication, list-ranking). (**Quiz 1 out**)

**W3** : PRAM algorithms (continued). Interconnection networks for parallel computers. Linear Arrays. Sorting on a linear array.

**W4** : 2-d Arrays (meshes) and sorting on meshes. Matrix and arithmetic algorithms on arrays. (**Quiz 1 in/Quiz 2 out**)

**W5** : Trees. Parallel prefix and carry-lookahead addition on trees. Hypercubes and Hypercubic Networks.

**W6** : Embeddings. Odd-even mergesort. (**Quiz 2 in**)

**W7** : FFT on a butterfly. (**Midterm Exam**)

**W8** : Routing on arrays. Routing on hypercubic networks.

**W9** : Realistic Parallel Computer Models: The Bulk-Synchronous Parallel (BSP) model. The BSP programming paradigm and a library that implements it. (**PA1 out**)

**W10** : Algorithm design on the BSP model. Primitive operations (broadcasting and parallel prefix) on the BSP. Performance prediction under the BSP model. Case study: Matrix vector multiplication.

**W11** : Communication and synchronization primitives. Data distribution and locality. Case Study: Matrix multiplication on the BSP model. (**PA1 in/PA2 out**)

**W12** : Parallel linear system solution under various data distributions. (**PA3 out**)

**W13** : Parallel Sorting. Bitonic Sorting. General and integer sorting. (**PA2 in**)

**W14** : Sorting (continued). Special topics (Image processing, searching).

**Final** : By noon on the day before: **PA3 in**. **Final Exam**.