Parallel Scientific Computation
Fall 2018
Room: TBA
Wednesday 9:30-11:30

Dr. Alexander Tzanov
Office Hours: after class, and by appointment

Course Description

This course will introduce the students to the concepts, methods, models and technologies used in development of parallel scientific applications for modern high performance computational architectures.It focuses onto entire range of issues that a developer of parallel scientific applications should consider when designing a parallel algorithm.The course consists of 3 parts. First, the students are introduced to fundamentals of modern computational architectures. That includes overview of several processor architectures (Intel and IBM),and study of memory organization and memory hierarchy of parallel computers, characteristics and topologies of interconnection networks applied in high performance systems and routing and switching algorithms. In that part of the course, the students will be introduced to abstract models of parallel computation PRAM, DMM and BSP. In the second part of the course the students will be introduced to various levels of parallelism and to fundamental parallel programming paradigms and main programming models for HPC. The students will learn how to design and write effective parallel numerical code for typical distributed manycore systems. Here the MPI (OpenMPI flavour) will be introduced as a framework for solving large problems. The third part of the course the students will study parallelization of linear system solvers, Adaptive Mesh Refinement, Set Partitioning problem, Simulated Annealing,Graph Coloring, Graph Betweenness, 3D Fast Fourier Transform (FFT) and Matrix Multiplication. If time permits other topics may also be included.

Course Materials:

◇ *Parallel Programming for Multicore and Cluster Systems* by Thomas Tauber and Gudula Runger, Springer, 2013

◇ *Introduction to Parallel Computing* by Grama, A. Gupta, A. Karypis,G.,Kumar V., Pierson Education, 2004.

◇ *Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers (2nd ed.) by B. Wilkinson and M. Allen, Prentice Hall. ISBN-10: 0131405632*

◇ *Stochastic numerics for mathematical physics by Milstein, A., Grigori, N., Springer Science and Business Media, 2013 (optional)*

◇ *Software concepts and numerical algorithms for a scalable adaptive parallel finite element method in Adv Comput Math (2015) 41: 1145. (optional)*

◇ *An implicit parallel multigrid computing scheme to solve coupled thermal-solute phase-field equations for dendrite evolution in Journal of Computational Physics, Volume 231, Issue 4, 2012, pp. 1781-1796 (optional)*

◇ *Other Resources and references:*

- *Using MPI - 2nd Edition: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation) by W. Gropp, E. Lusk, and A. Skjellum, MIT Press. ISBN: 0262571323. (Reference)*
- *Rudin Walter (1976). Principles of Mathematical Analysis, 3-rd ed. McGraw-Hill.(Reference)*

Additional papers and materials will be provided in class.

Learning Goals and Objectives: *This course will introduce students to the main concepts in designing scalable parallel algorithms. Specific goals: There will be an introduction to the concepts and techniques which are critical to develop scalable parallel scientific codes, listed below.*

1. *Learn about abstract models of parallel computation and real HPC architectures.*
2. *Learn how to design algorithm in distributed environments.*
3. *MPI tutorial and optimal domain decomposition*
4. *Use MPI for parallelization of Linear System Solvers, Adaptive Mesh Refinement, Set Partitioning problem, Simulated Annealing,Graph Coloring, Graph Betweenness, 3D Fast Fourier Transform (FFT) and Matrix Multiplication.*

Prerequisites:

The class requires working knowledge of undergraduate level discrete mathematics (advanced numerical methods preferred), good programming skills in C/C++ or Fortran (95 90 or 08) plus knowledge of advanced data structures and algorithms. Students is expected to have good grip of advanced linear algebra, finite element, finite volume numerical methods, adaptive mesh, multigrid and some knowledge in hierarchical methods for particle simulations. IMPORTANT: Working knowledge of Linux OS (user level) is a must. For lab students will login remotely to high performance sever so working knowledge of secure communication (ssh) and file transfer protocols (scp,sftp, globus) is also required.

Academic Integrity:
Students are responsible for understanding the CUNY academic integrity

---

Teaching Method:
The course will be taught using class lectures. Class-related material (lecture notes, messages, etc.) will be posted. Additional material will be distributes in some classes. *Each student gets individual project and will write a report about used parallelization strategy, models and obtained results including benchmarks. Most of this work will take place outside the classroom, as will connect remotely to HPC resource, and apply the material and techniques to which students were introduced in class.*

Grading: The grade[1] will be based on project (40%), midterm exam (50%) and quiz (10The exam and quiz will be closed-book. Please write your name, CUNY ID and email on every page of your work.

Tentative Course Outline.

Sections labeled R refer to Rauber, sections labeled G refer to Gupta, sections labeled M refer to M. Alen

[1]Your final grade is not subject to negotiation. If you feel I have made an error, submit your written argument to me within one week of receiving your final grade. Clarify the precise error I made and provide all due supporting documentation. If I have made an error, I will gladly correct it. But I will adjust grades only if I have made an error.

◇ Weeks 1-2: HPC Fundamentals
Sections C-1.1 through C-3.3 of chapter 3, and JP-8 to JP-10

HPC architectures overview

Scientific problems and their characteristics.

Flynn Taxonomy, Amdahl and Gustafson Laws R-2.2

Types of CPU CISC, RISC, single core, multicore, architectural characteristics of multicore CPU in terms of HPC. R-3-4, G 2.1 to G2.8, R3-8

Memory organization and memory hierarchy of parallel computers distributed, shared and distributed-shared, caches, cache coherency and ccNUMA

Interconnection networks for HPC - direct interconnect networks, dynamic networks, crossbar, omega etc

Routing algorithms and flow control.

Memory organization and memory hierarchy of parallel computers distributed, shared and distributed-shared, caches, cache coherency and ccNUMA. R3-4,R3-5

Interconnects R2-7

◇ Week 3-8: Programming paradigms and programming models
Sections R3-1 through R3-6, Sections R3-3

Programming models PRAM, BSP, LogP

Levels of parallelism instructional, data, loop, functional, explicit and implicit, etc. Considerations.

Programming paradigms processes, treads, data placement, affinity

Intro to OpenMPI Sections G4-1 to R4-7m notes in class.R5-1 to R5-5

Intro to OpenMP Sections R6-1 and R6-3

Intro to SHMEM - notes in class

◇ Weeks 9-14 Parallel numerical algorithms
Sections G8-1 through G9-9 & R8-1 to R8-6 M2-2 to M3-5, D1-2 to D3-3 Section G10-1 to G10-8, sections G12-1 to G12-5 Sections G13-1 to G13-3

Linear system solvers G8-1 to G8-3 R8-1 to R8-3

Adaptive mesh refinement Comp. Physics Paper

Jacobi iteration Adv. Comp Math paper

Gauss-Seidel on sparse systems R8-3

Cholesky factorization notes in class

Graph partitioning G10-1 to G10-8

Fast Fourier Transform G13-1 to G13-3

Stochastic PDE Milstein scheme M3-1 to M3-5

Dynamic programming iteration scheme G-12-1 to G12-5