Instructor:  Aiichiro Nakano;  office: VHE 610;  phone: (213) 821-2657;  email: anakano@usc.edu
TA:  John Tran;  email: jtran@isi.edu;  homework submission: usc.csci596@gmail.com
Classes:  3:30-4:50pm M W, SGM 601—lecture; 4:30-5:30 pm F, KAP 156—discussion
Office Hour:  2:00-3:00pm M, VHE 611
Course Page:  http://cacs.usc.edu/education/cs596.html
Prerequisites:  Basic knowledge of programming, data structures, linear algebra, and calculus.
Textbooks:  W. D. Gropp, E. Lusk, and A. Skjellum, Using MPI, 2nd Ed. (MIT Press, 1999)—recommended
A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to Parallel Computing, 2nd Ed. (Addison-Wesley, 2003)—recommended
Course Description
Particle and continuum simulations are used as a vehicle to learn basic elements of scientific computing and visualization. Students will obtain hands-on experience in: 1) formulating a mathematical model to describe a physical phenomenon; 2) discretizing the model, which often consists of continuous differential or integral equations, into algebraic forms in order to allow numerical solution on computers; 3) designing/analyzing numerical algorithms to solve the algebraic equations efficiently on parallel computers; 4) translating the algorithms into a program; 5) performing a computer experiment by executing the program; 6) visualizing simulation data in an immersive and interactive virtual environment; and 7) managing/mining large datasets.
Syllabus
1. Basic molecular dynamics (MD) algorithms
   • Integration of ordinary differential equations; periodic boundary condition; linked-list cells
2. Parallel MD
   • Spatial decomposition (interprocessor caching and migration); load balancing; scalability analysis; asynchronous MD
   • Message passing interface (MPI) vs. shared memory (OpenMP) programming
   • Hybrid MPI+OpenMP programming
   • Multicore parallel programming (e.g., GPU—CUDA, Phi, Cell)
3. Grid/cloud scientific computing
   • Computation steering on the Grid/cloud (e.g., Globus, Grid RPC, MapReduce)
   • Grid/cloud enabling parallel applications
4. Scientific visualization
   • OpenGL programming
   • Scientific visualization software—VMD, VisIt, ParaView
   • Virtual-reality programming—CAVE Library, ImmersaDesk, tiled display
5. Scientific big data management/mining
   • Data compression for scalable I/O
   • Graph-based knowledge discovery
   • In situ data analysis
6. Object-oriented scientific programming
   • Parallel software tools for irregular data structures; object-oriented MD; scripting wrappers
7. Other simulation methods
   • Stochastic simulations: Monte Carlo method
   • Continuum simulations: Schrödinger equation in quantum mechanics
Grading Scheme
Assignments (5-6 programming projects), 80%; final project, 20%
A (100-90%); A− (90-85%); B+ (85-80%); B (80-75%); B− (75-70%); C (70-60%); D (60-50%)
Schedule
Final presentation (Nov. 30, Dec. 2 & 4)