CS 6230: High Performance Computing and Parallelization

Professor Mike Kirby
School of Computing and Scientific Computing and Imaging Institute
University of Utah

Administrative Issues

- Hand out course syllabus
- Discuss research areas of interest
- Discuss what motivates parallel computing
- **Goal of the course**: This course is structured to train students to reason about the design and implementation of efficient parallel programs. The focus areas of the class will be on the modeling, implementation and evaluation of distributed, message-passing interface (MPI) based programs, shared-memory thread-based OpenMP programs, and hybrid (MPI/OpenMP) programs. Almost all examples will be aligned with numerical scientific computing. This course is appropriate for those that want transform serial algorithms into parallel algorithms, want to modify currently existing parallel algorithms, or want to write parallel algorithms from scratch.

SCI INSTITUTE

THE UNIVERSITY OF UTAH

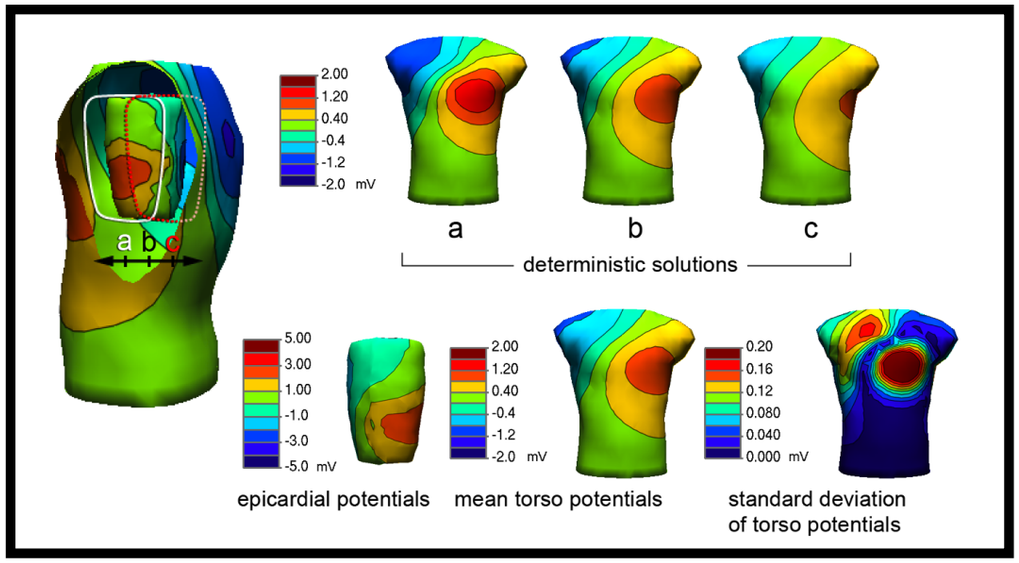
References Used Throughout Course

- Textbook (*Parallel Computing* by Wilkinson and Allen)
- [Lin] Calvin Lin and Lawrence Snyder, *Principles of Parallel Programming,* Addison-Wesley, 2009.
- [Pacheco] Peter S. Pacheco, *Parallel Programming with MPI,* Morgan-Kaufmann, 1997.
- [Mattson] Timothy G. Mattson, Beverly A. Sanders, and Berna L. Massingill, *Patterns for Parallel Programming,* Addison-Wesley, 2005.
- [Grama] Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Introduction to Parallel Computing, Second Edition, Addison-Wesley, 2003.
- [Chandra] Rohit Chandra, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon, *Parallel Programming in OpenMP,* Morgan-Kaufmann, 2001.

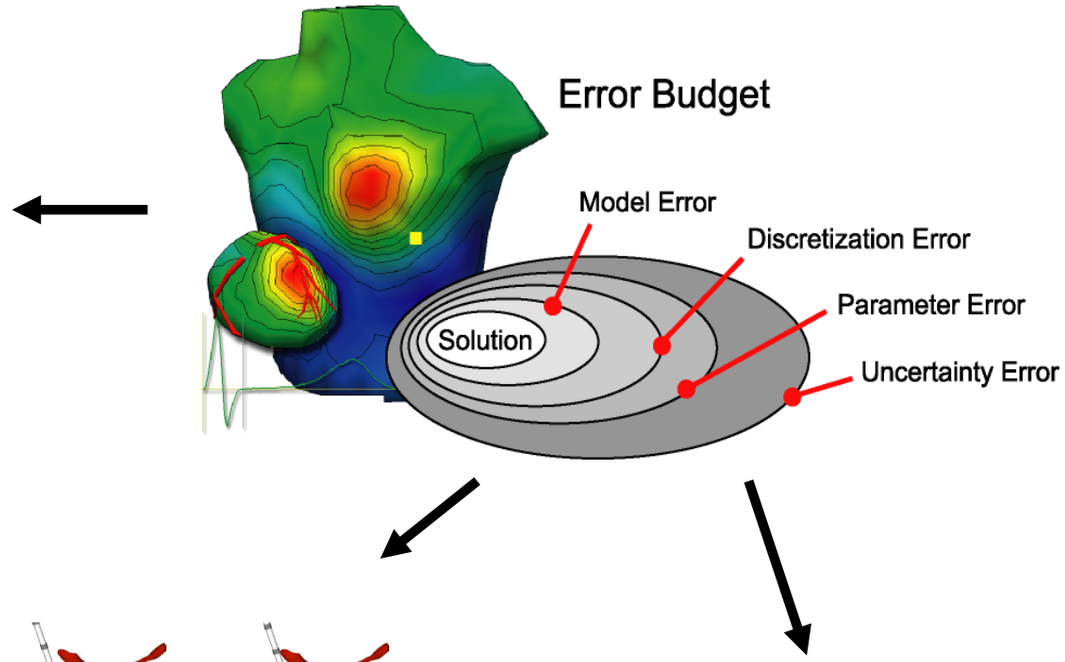
SCI INSTITUTE

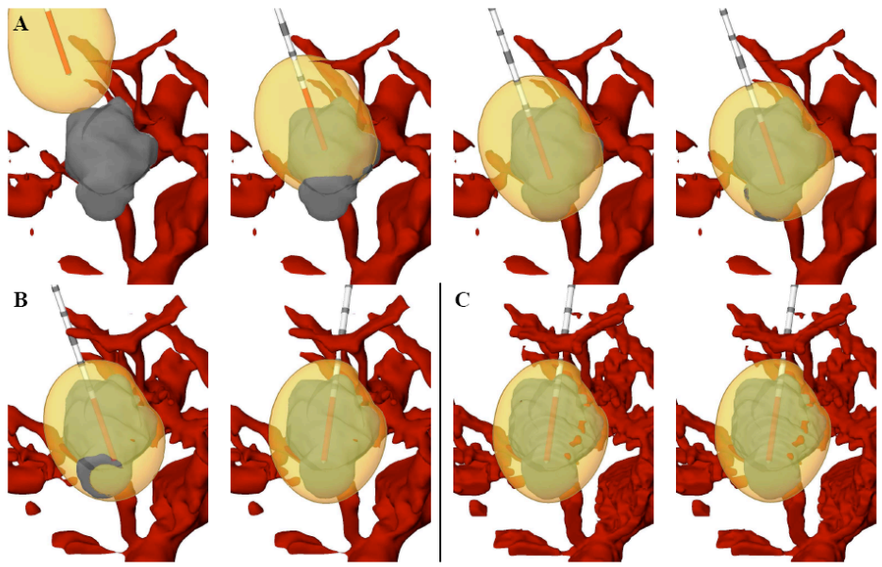
THE UNIVERSITY OF UTAH

Simulation Science: Quantification and Control of Modeling, Discretization and Uncertainty Errors

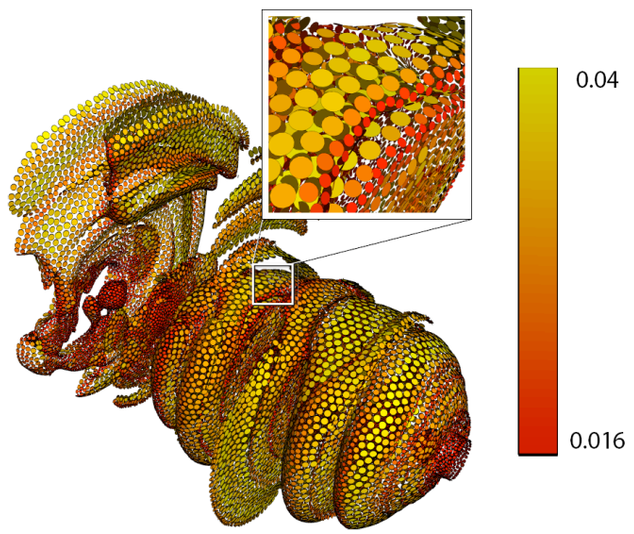


Quantification of Position Uncertainty in ECG Simulations





Control of Uncertainty Errors In Material Parameters in Liver Ablation Simulations



SCI INSTITUTE

THE UNIVERSITY OF UTAH

Current: Formal Methods for HPC

Modeling of the MPI Library

MPI : Widely used HPC library with COMPLEX and EVOLVING semantics

In situ Model Checking

Large MPI programs are "MPI Calls Hanging off a Program Scaffolding." Hence Finite State Machine model extraction + Model Checking is ineffective in many cases

Verifying One-Sided MPI Constructs

Some of the new MPI Extensions are Extremely Poorly Understood

Parallel Model Checking

Parallelism can benefit even the verification process !!

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Goal of Parallel Computing

The goal of **parallel computing** has traditionally been to provide performance -- either in terms of processor power or memory -- that a single processor cannot provide; thus, the goal is to use multiple processors to solve a single problem. The goal of **distributed computing** is to provide convenience, where convenience includes availability, reliability, and physical distribution (being able to access the distributed system from many different locations). [Lee]

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Concurrency versus Parallelism

Concurrency and Parallelism: Though these terms are closely related, history influences how we use them. **Concurrency** is widely used in the operating system and database communities to describe executions that are logically simultaneous, while **parallelism** is typically used by the architecture and supercomputing communities to describe executions that physically execute simultaneously. [Lee]

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Four important parts of parallelism

- Correctness
- Performance
- Scalability
- Portability

[Lee]

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Possible Software "Solutions" Employed

- OpenMP (www.openmp.org)
- Message-Passing Interface (MPI)
- Charm++ (charm.cs.uiuc.edu)
- Unified Parallel C (upc.lbl.gov)
- High-Performance Fortran (hpff.rice.edu)

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Sources of Performance Loss

- Overhead, which the sequential computation does not need to pay

- Non-parallelizable computation

- Idle processors

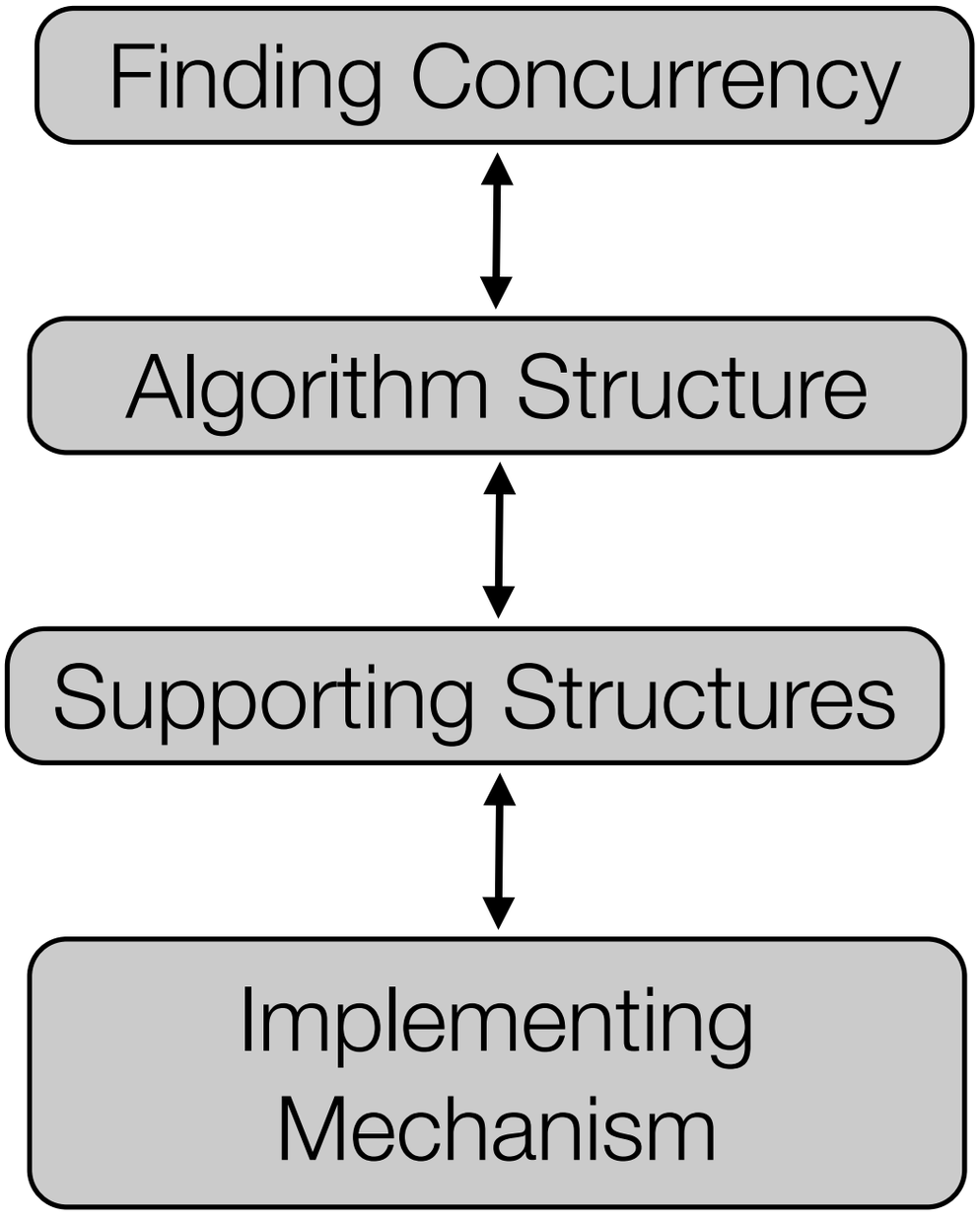
- Contention for resources

[Lee]

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Overview of the Pattern Language



Where will we run our tests?

Raven Cluster
- head node: raven-srv.cs.utah.edu (2x AMD Opteron 240)
- compute nodes: raven1-32 (1x AMD Athlon 64 3500+)
- Memory: 3GB on srv, 2GB per node
- GCC v4.5.2
- MPICH2 v1.3.1

CADE Lab 1 (40) and Lab 3 (30) (e.g. lab1-1, lab3-10)
- Intel core i7-860 2.8 GHz Quad Core (2-way)
- Memory: 4GB per node
- GCC 4.4.4
- OpenMPI 1.2.8

SCI INSTITUTE

THE UNIVERSITY OF UTAH

Next Time

- Read Chapter 1 (note that material in 1.2 will be discussed in future lectures in detail)

- Discussion of the current “state of the art” in parallel computing (a guest lecture by Professor Martin Berzins)

SCI INSTITUTE

THE UNIVERSITY OF UTAH