Adaptive MPI

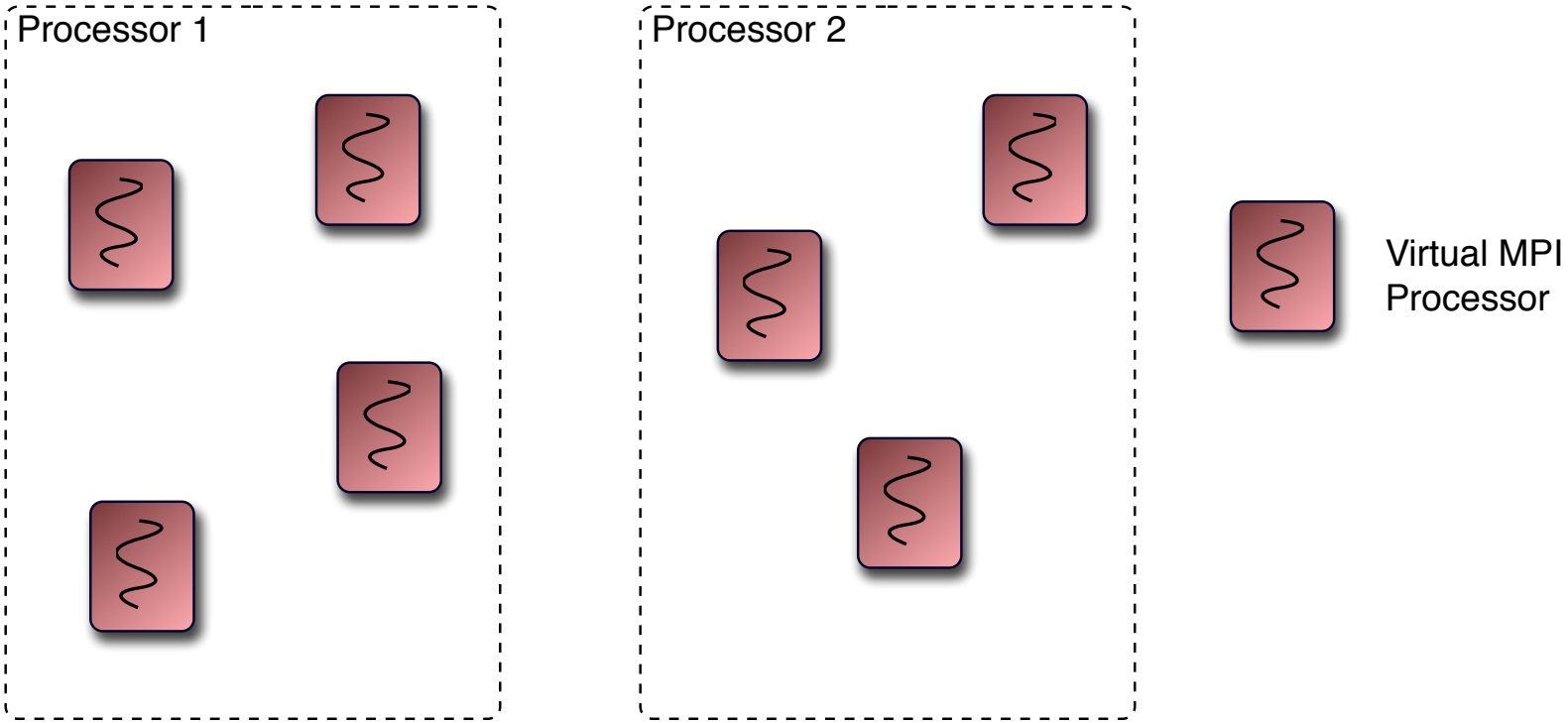
Isaac Dooley

Parallel Programming Laboratory

University of Illinois at Urbana-Champaign

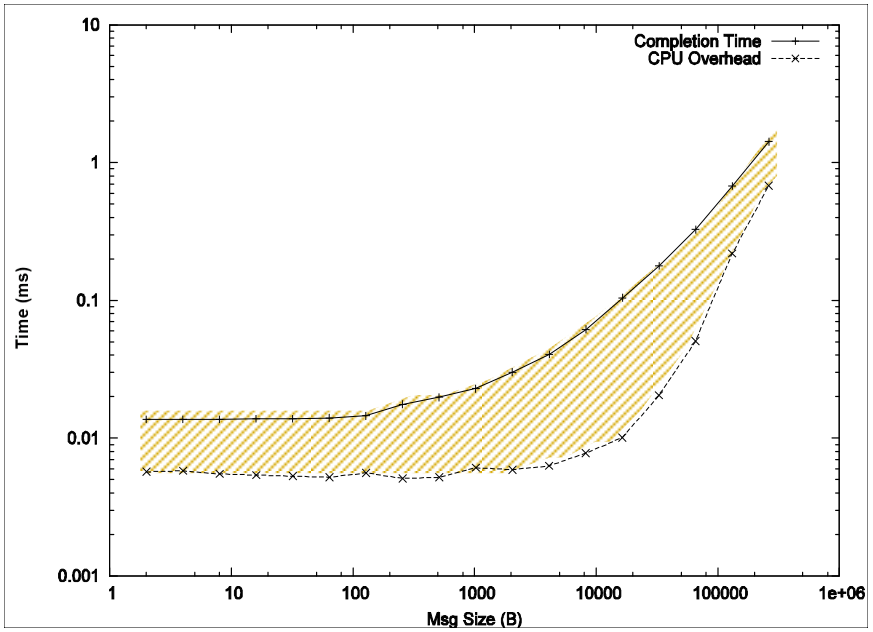
AMPI: MPI with Virtualization

- Each AMPI virtual process is implemented by a user-level *thread* embedded in a migratable *object*



Adaptive Overlap

- Problem: Gap between completion time and CPU overhead
- Solution: Overlap between communication and computation



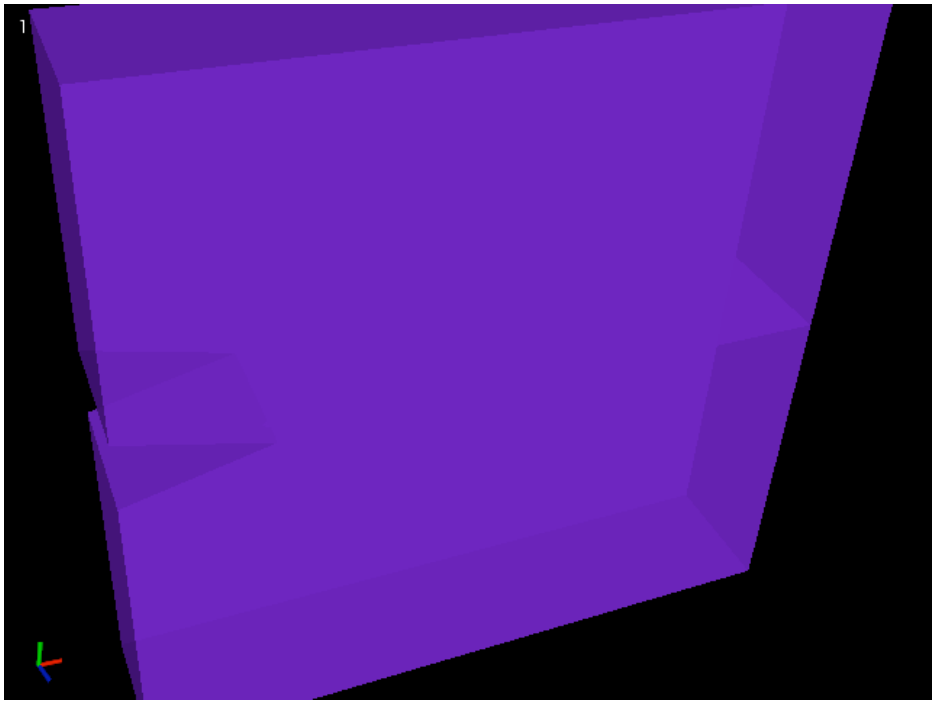
Completion time and CPU overhead of 2-way ping-pong program on Turing (Apple G5) Cluster

Automatic Load Balancing

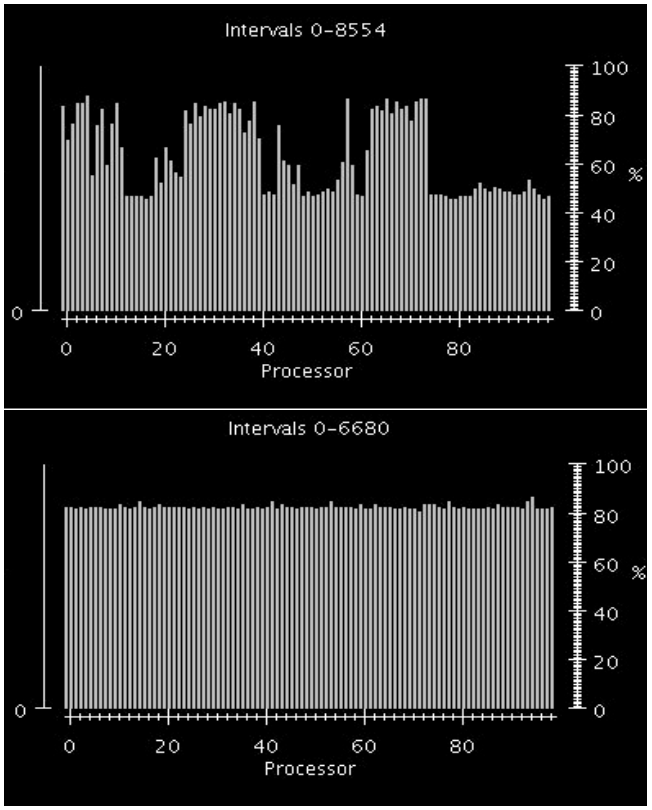
- Challenge
  - Dynamically varying workloads
  - Load imbalance impacts overall performance
- Solution
  - Measurement-based load balancing
    - Scientific applications are typically iteration-based
    - RTS collects CPU and network usage of VPs
  - Load balancing by migrating threads (VPs)
    - Threads can be packed and shipped as needed

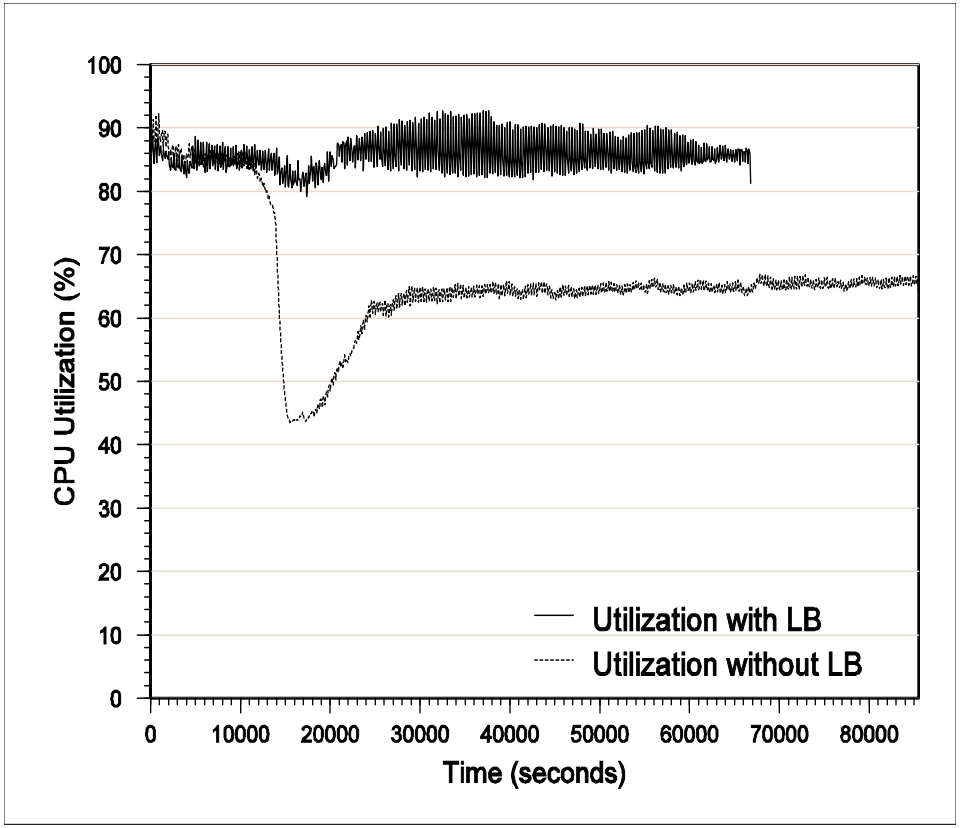
Automatic Load Balancing

- Application: Fractography3D
  - Models fracture propagation in material



Automatic Load Balancing

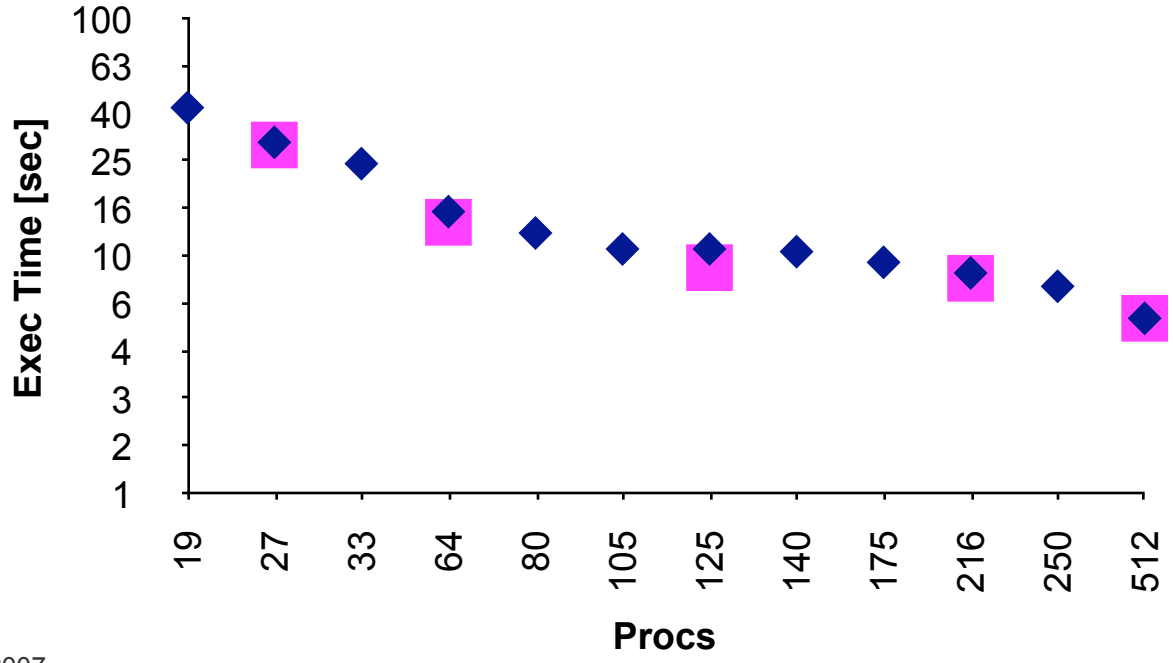




CPU utilization of Fractography3D without vs. with load balancing

Flexibility

- Running on arbitrary number of processors
  - Runs with a specific number of MPI processes
  - Big runs on a few processors



3D stencil calculation of size 240^3 run on Lemieux.

Communication Optimizations

- AMPI run-time has capability of
  - Observing communication patterns
  - Applying communication optimizations accordingly
- Examples
  - Streaming strategy for point-to-point communication
  - Collectives optimizations

Thank You!

Download of AMPI is available at:
http://charm.cs.uiuc.edu/

Parallel Programming Lab
at University of Illinois