HPC Runtime Software

SC’11
Agenda

• Intro (10 minutes)
• Runtime System Experiences
  ‣ 10-12 minutes: Charm++ (Dr. Kale)
  ‣ 10-12 minutes: Parallex/HPX (Dr. Sterling)
  ‣ 10-12 minutes: SWARM (Dr. Khan)
  ‣ 10-12 minutes: OCR (Dr. Sarkar)
• Questions to the audience (20 minutes)
• Comments from the audience (10 minutes)
Future Challenges

• Many-core is coming
  ‣ Single-threaded programs with OpenMP acceleration will no longer fully exploit hardware
  ‣ New programming paradigms are necessary

• Hardware is getting more heterogeneous
  ‣ Irregular mix of Architecture, Processor Speeds, Memory Layout, etc.
  ‣ Current hybrid programming techniques (MPI+ OpenMP + OpenCL) are not maintainable with growing complexity

• Application Computation Irregularity
  ‣ Elastic RTM
  ‣ Full Wave Inversion
  ‣ Static scheduling can no longer properly load balance
Runtime System Comparisons

**MPI, OpenMP, OpenCL**
- Communicating Turing Machines
- Bulk Synchronous
- Message Passing

**New Runtime Systems**
- Asynchronous Event-Driven Tasks
- Dependencies
- Constraints
- Resources
- Active Messages

Diagram comparing active threads and waiting in time.
Properties of Future Runtime Systems

• Expose and exploit concurrency of application
  ‣ Hide latency
  ‣ Prioritize critical path
  ‣ … but don’t expose too much concurrency!

• Expose and exploit data locality
  ‣ Minimize data movement between memory hierarchies
  ‣ Maximize data reuse

• Low overhead object oriented type system for runtime introspection

• High performance lock-free data structures

• Abstraction and unification of user/runtime interactions for productive heterogeneous programming

• Dynamic mapping of application needs and available resources
Questions

• In what contexts does your runtime improve performance over existing mainstream HPC runtimes (OpenMP/MPI)?
• What are the benefits of your runtime?
• What are the key runtime concepts you would like the audience to walk away with?