Charm++/AMPI BoF

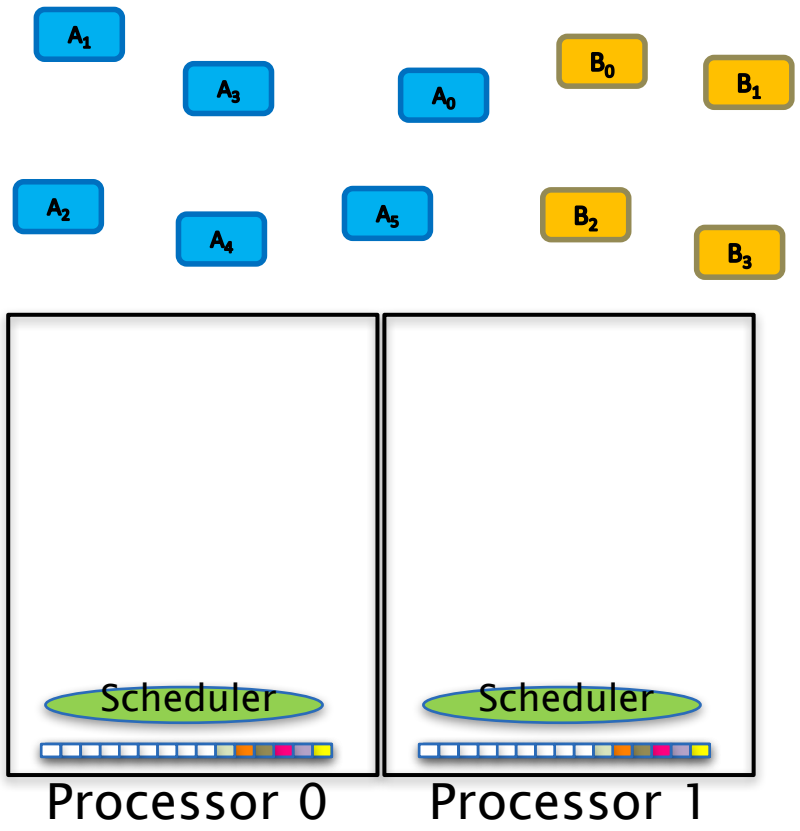
Laxmikant (Sanjay) Kale
Parallel Programming Laboratory
University of Illinois at Urbana-Champaign
http://charm.cs.Illinois.edu
http://charmplusplus.com
http://charmplusplus.org

Challenges in Parallel Programming

- Applications are getting more sophisticated
  - Adaptive refinement
  - Multi-scale, multi-module, multi-physics
  - E.g. load imbalance emerges as a huge problem for some apps
- Exacerbated by strong scaling needs from apps
  - Strong scaling: run an application with same input data on more processors, and get better speedups
  - Weak scaling: larger datasets on more processors in the same time
- Hardware variability
  - Static/dynamic
  - Heterogeneity: processor types, process variation, etc.
  - Power/Temperature/Energy
  - Component failure

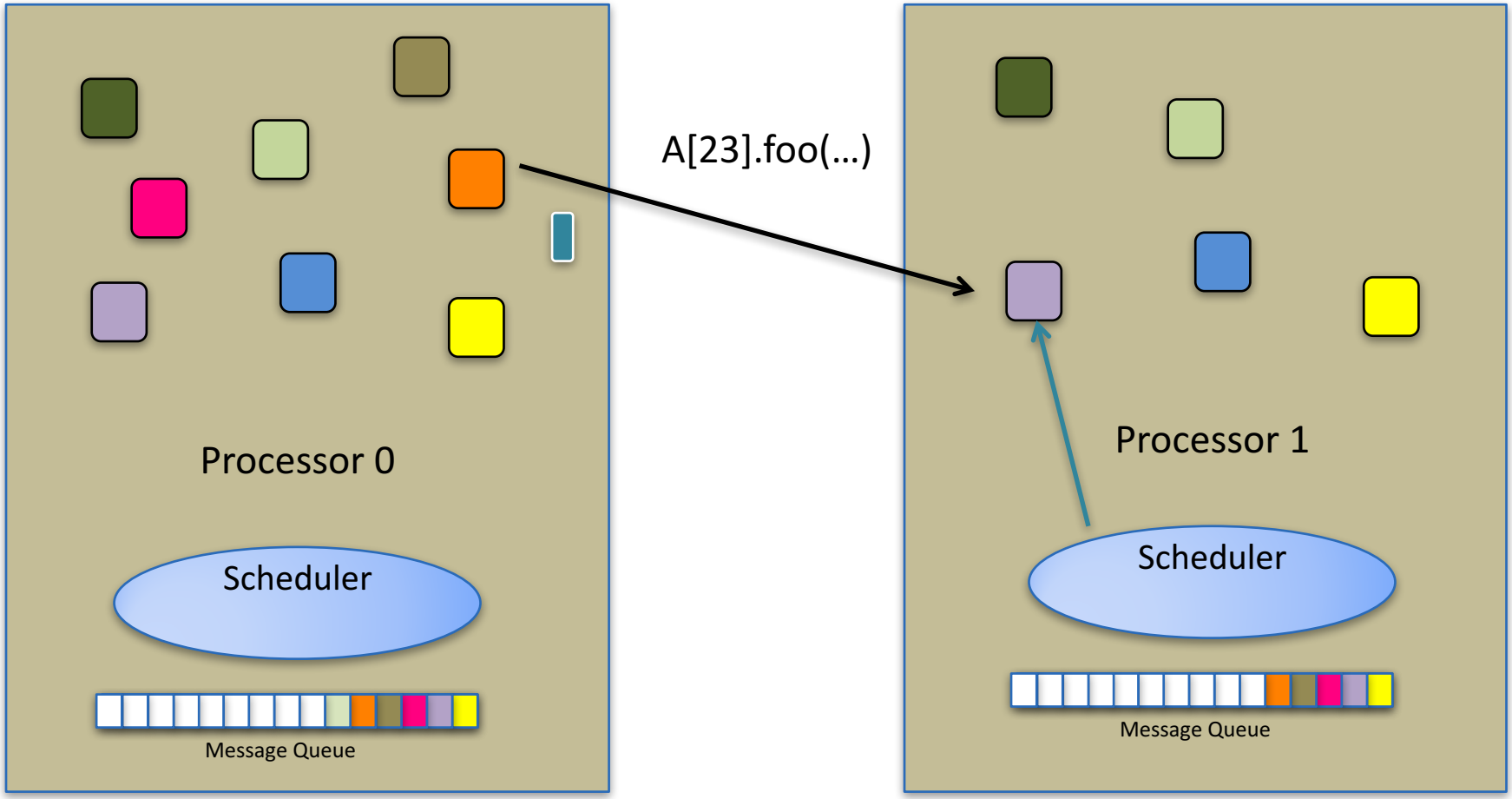
Charm++ Programming Model and Capabilities

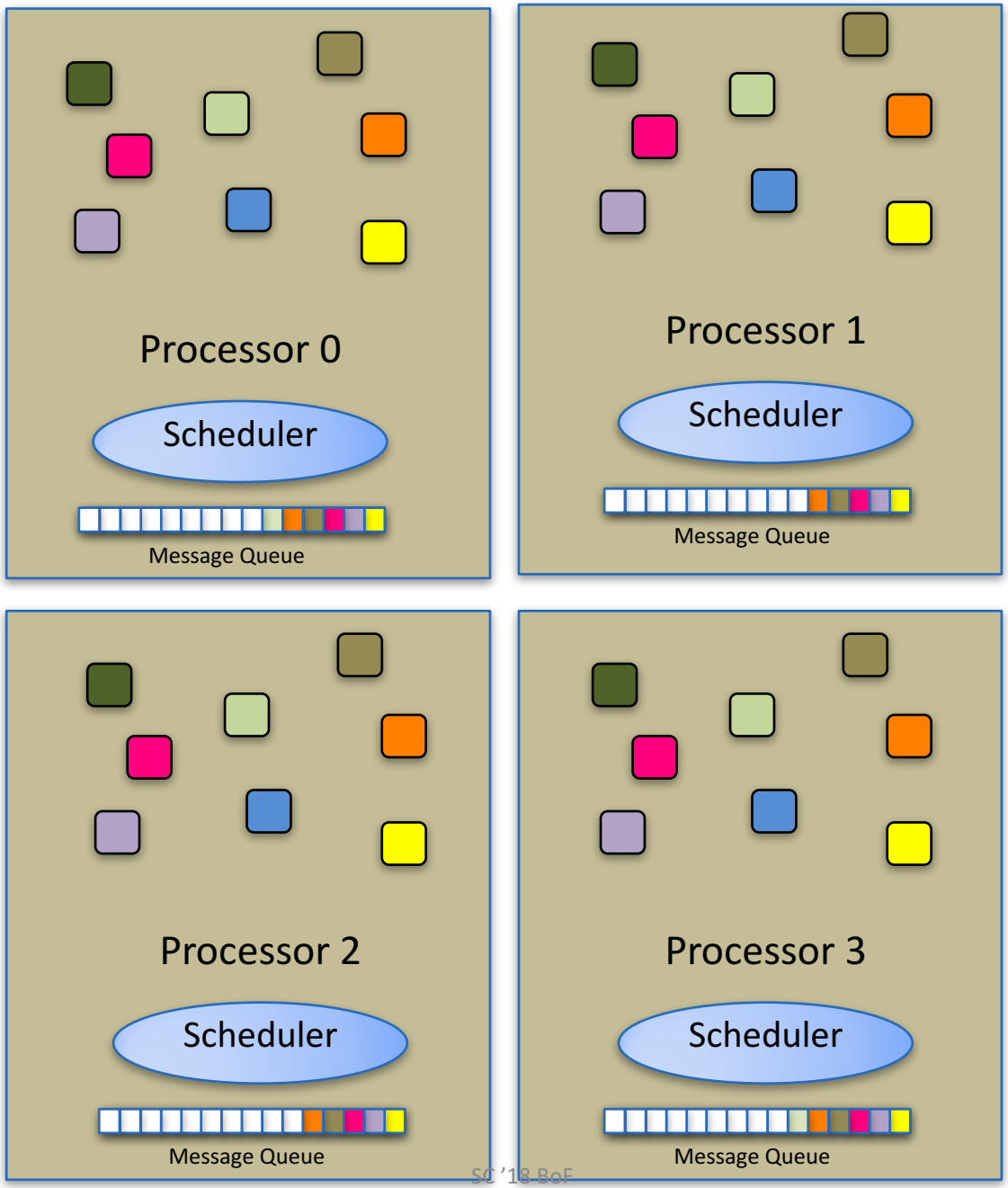
- Object-based processor virtualization:
  - Computation is expressed as a large number of objects
  - Interacting via asynchronous method invocations
  - Number of objects is independent of processors
  - Communication: using global names, not procs
- *Migratability:* The RTS assigns Objects to processors, and may reassign them
- *Asynchrony:* A message Driven Scheduler

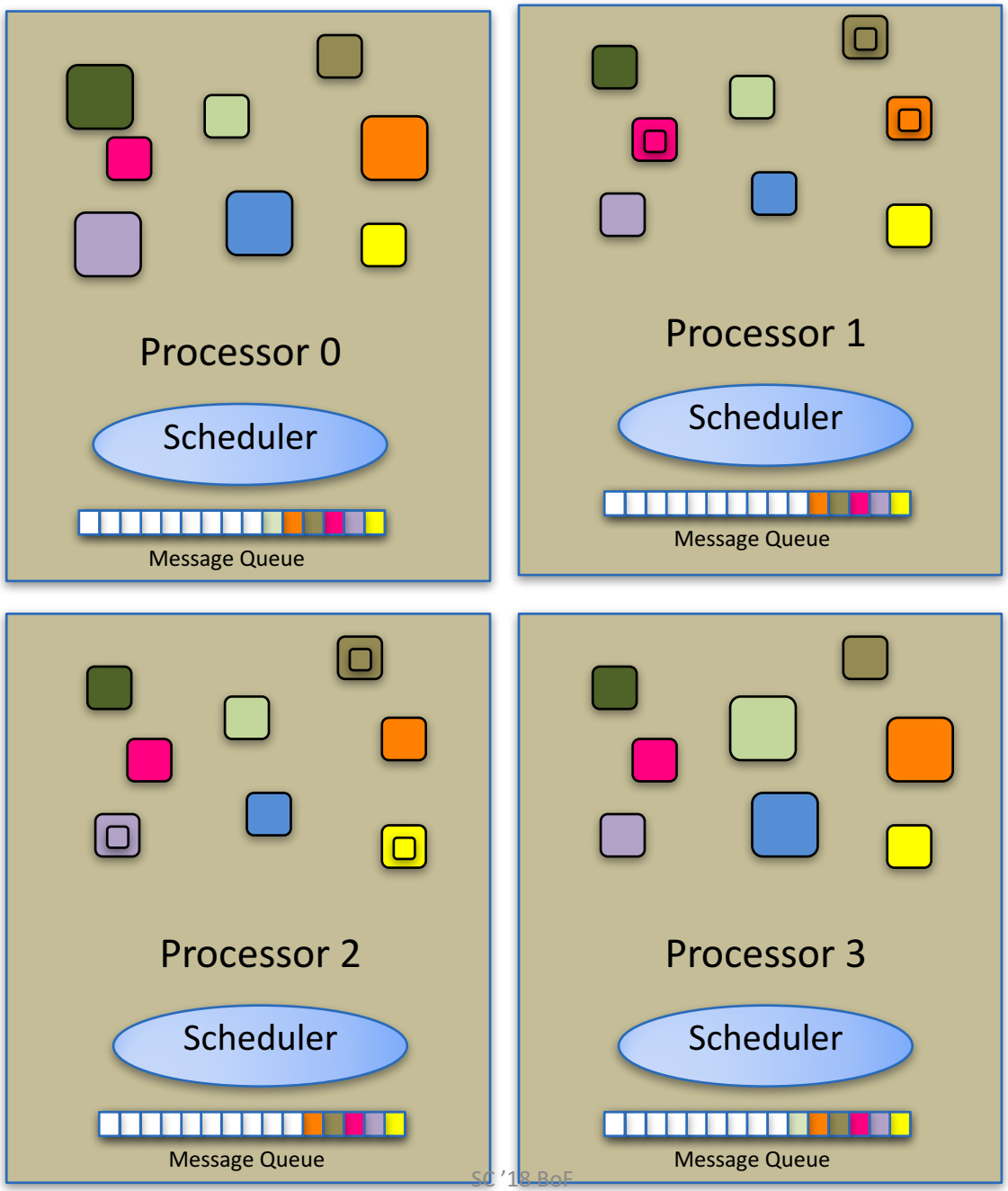


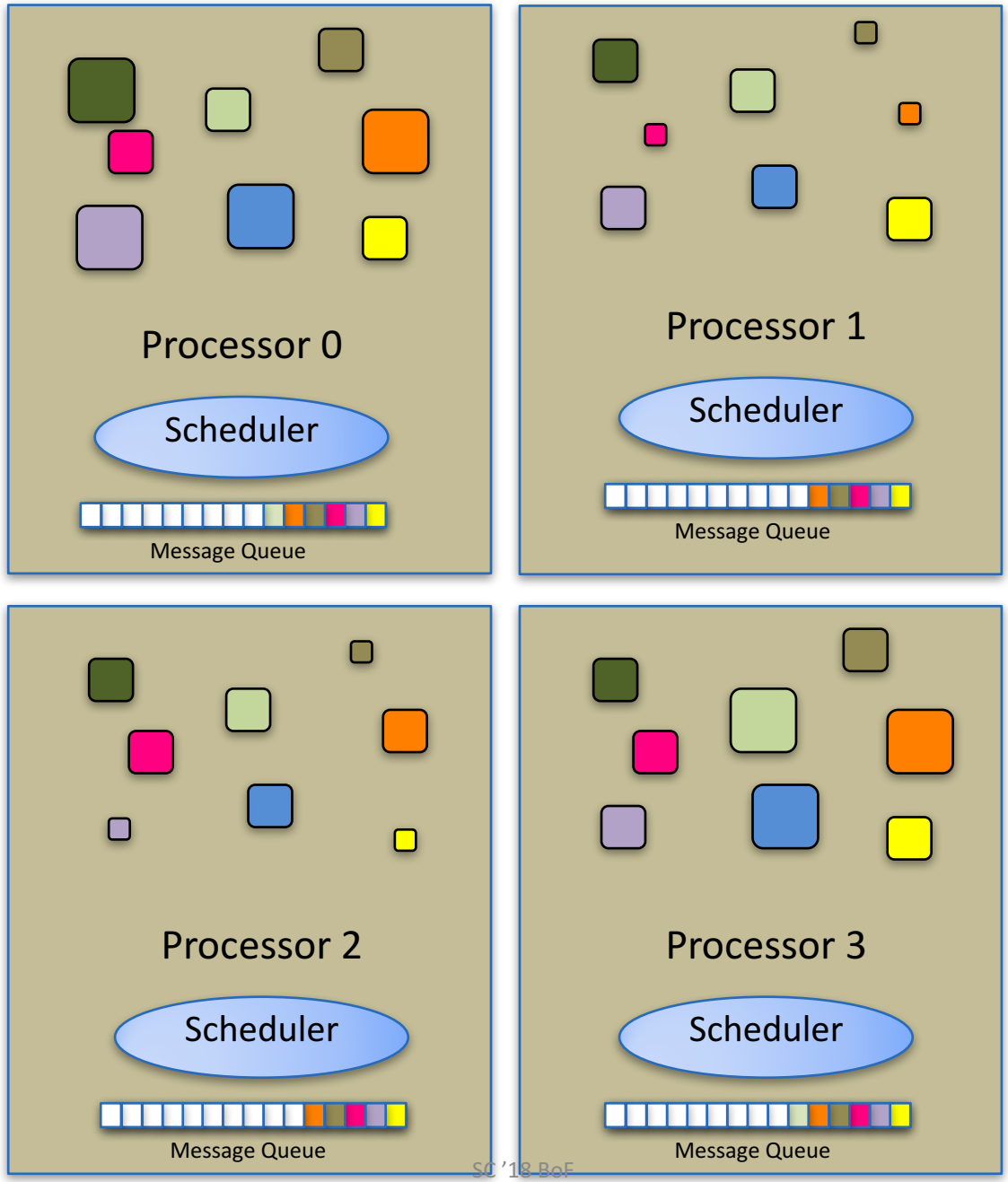
- Dynamic load balancing via object migration
- Compositionality: interleaves execution of modules
- Resilience support
- Power management

Message-driven Execution




Processor 0
Scheduler
Message Queue
Processor 1
Scheduler
Message Queue
Processor 2
Scheduler
Message Queue
Processor 3
Scheduler
Message Queue


Processor 0
Scheduler
Message Queue
Processor 1
Scheduler
Message Queue
Processor 2
Scheduler
Message Queue
Processor 3
Scheduler
Message Queue

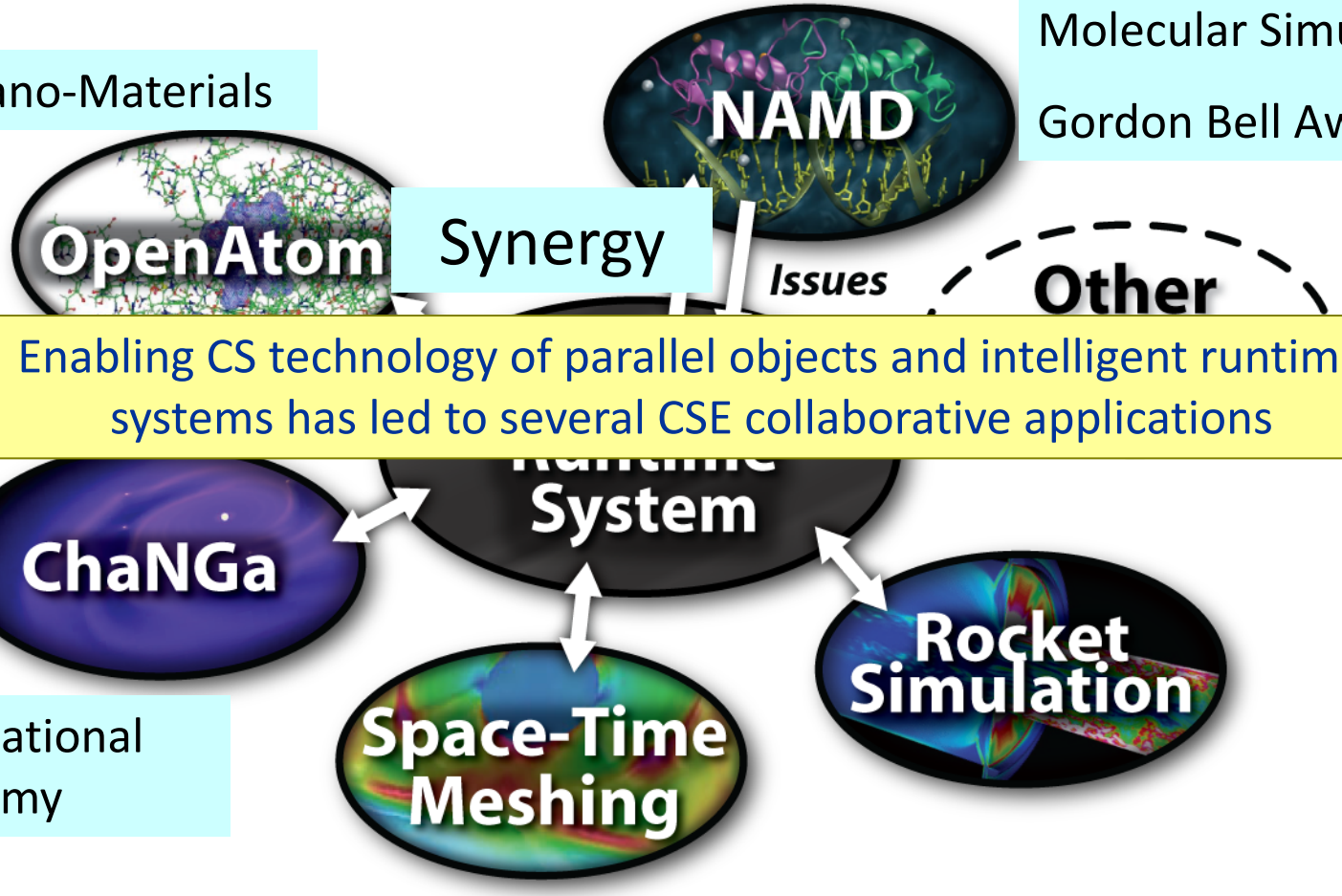

Processor 0
Scheduler
Message Queue
Processor 1
Scheduler
Message Queue
Processor 2
Scheduler
Message Queue
Processor 3
Scheduler
Message Queue

Charm++ and CSE Applications

Well-known Biophysics Molecular Simulation App

Gordon Bell Award, 2002

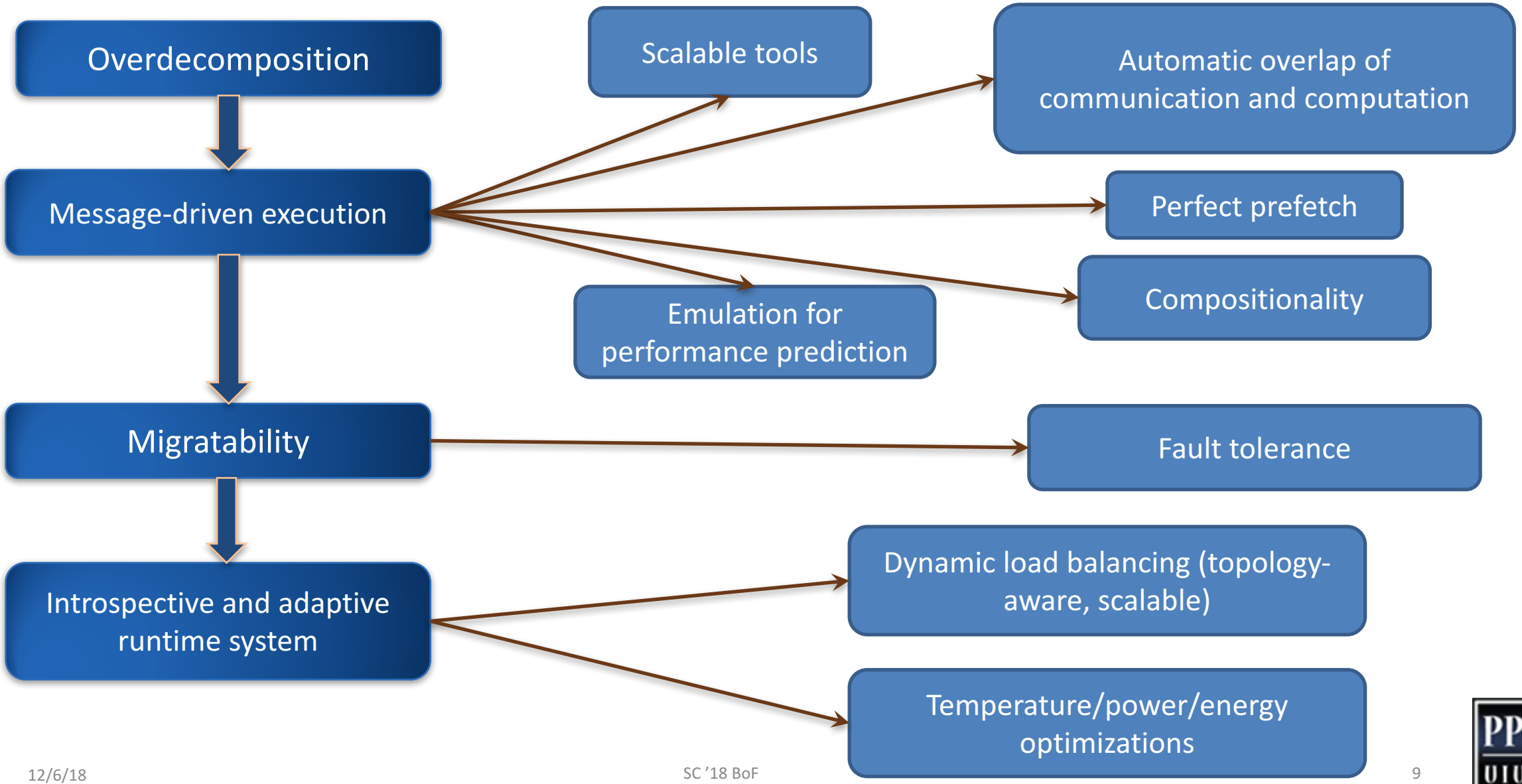
Nano-Materials



Enabling CS technology of parallel objects and intelligent runtime systems has led to several CSE collaborative applications

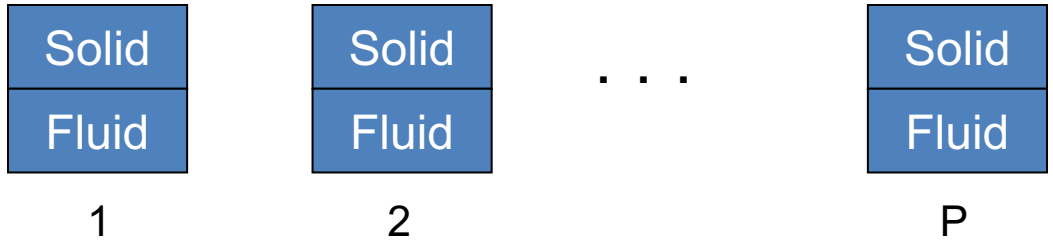
Computational Astronomy

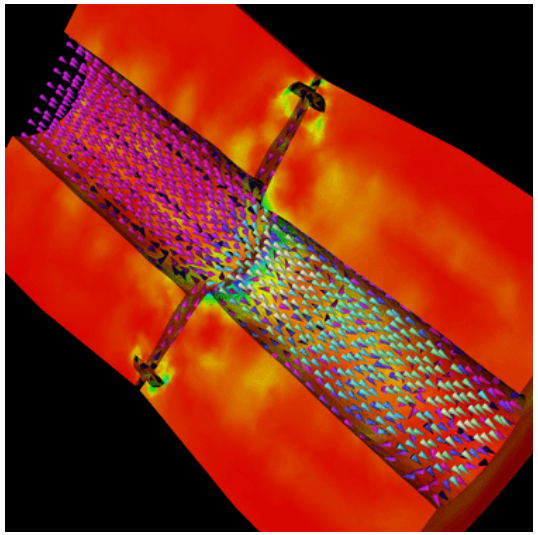
Charm++ Benefits



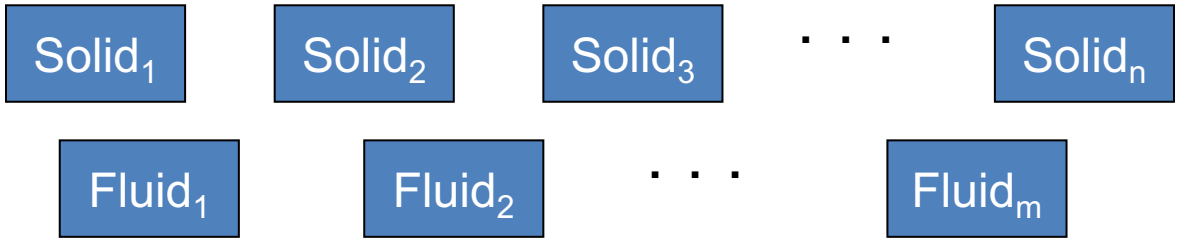
Decomposition Independent of numCores

- Rocket simulation example under traditional MPI





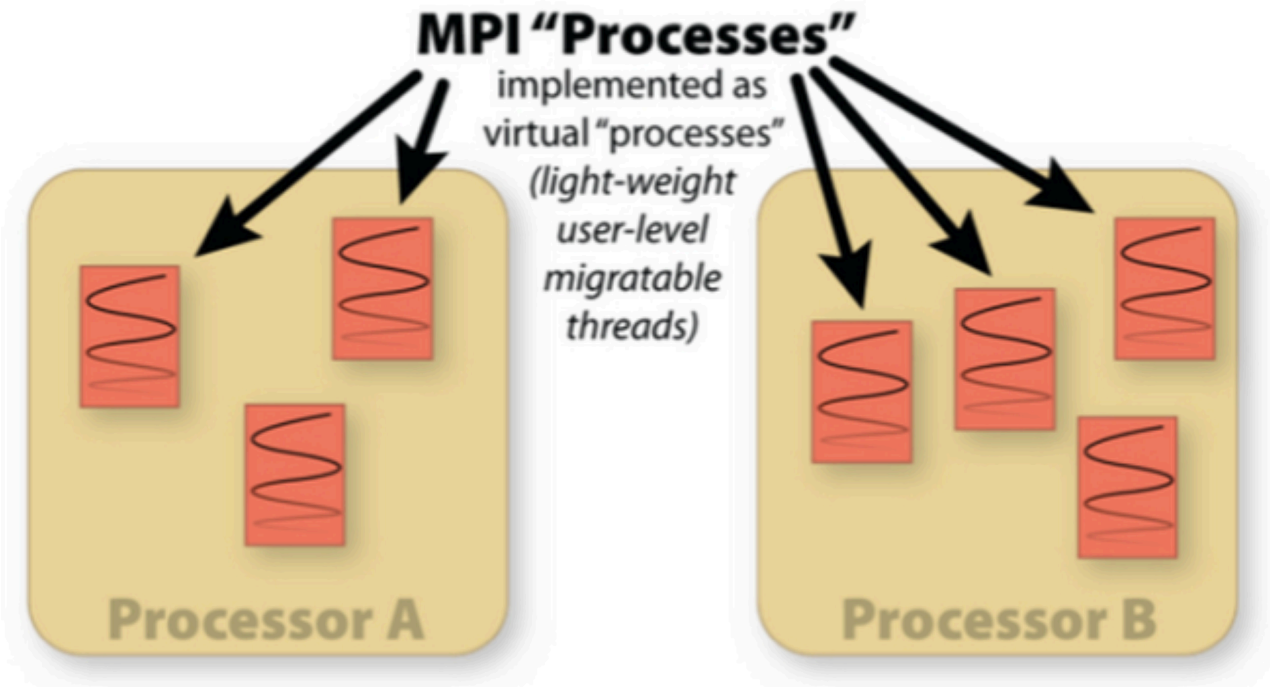
- With migratable-objects:



  - Benefit: load balance, communication optimizations, modularity

What is Adaptive MPI?

- AMPI is an MPI implementation on top of Charm++'s runtime system
  - Enables Charm++'s dynamic features for pre-existing MPI codes



Recent Work

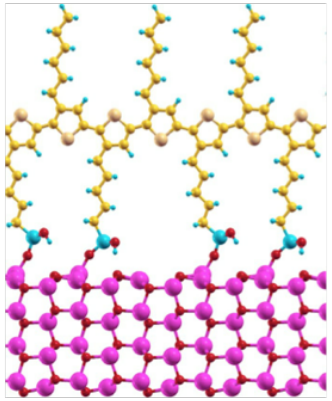
Recent Research Highlights

- LLVM OpenMP runtime library ported to use Converse threads and scheduler
  - Enables co-scheduling of Chare entry methods with OpenMP parallel regions
- Charades framework for Parallel Discrete Event Simulation
  - New asynchronous synchronization algorithms combined with dynamic load balancing improve speculative execution accuracy with low synchronization costs
- SpECTRE relativistic astrophysics framework developed at Cornell
  - Has effectively hidden .ci files behind C++ template metaprogramming for its users
  - Many fixes and new features added to charmc to enable more generic code
- OpenAtom/GW: released 1.0 version

OpenAtom GW-BSE

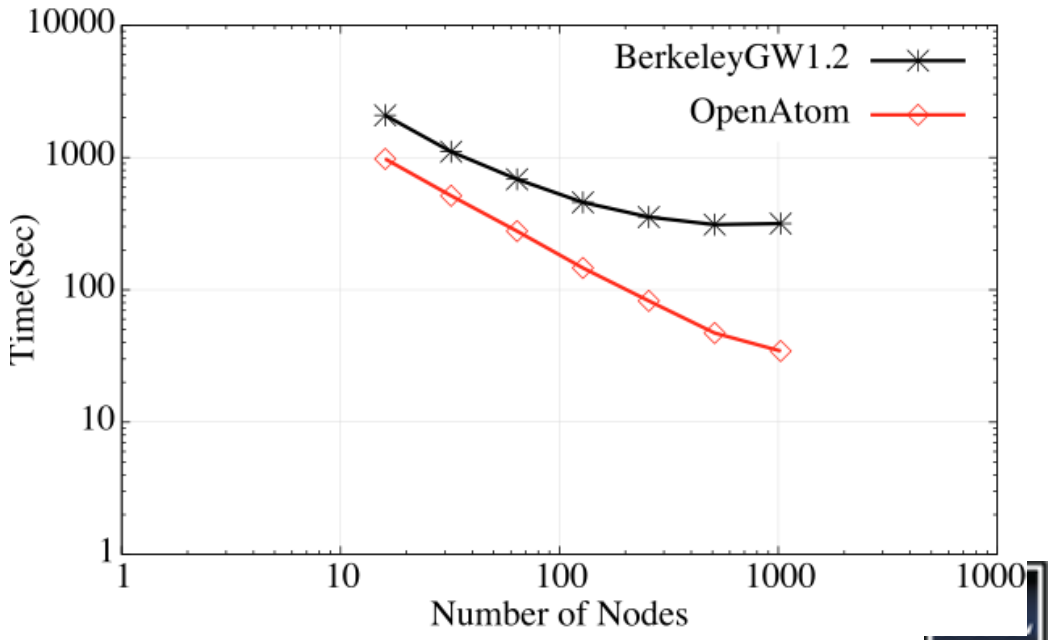
- GW approximation to study processes that involve excited electron states – large matrix computations N^4
- Study transport of electrons in a material or across an interface
  - Eg: Photovoltaic systems
- We have released version 1.0 of GW software
- Demonstrated improved scaling compared to Berkeley GW
- Working on faster N^3 method to improve performance

P3HT polymer



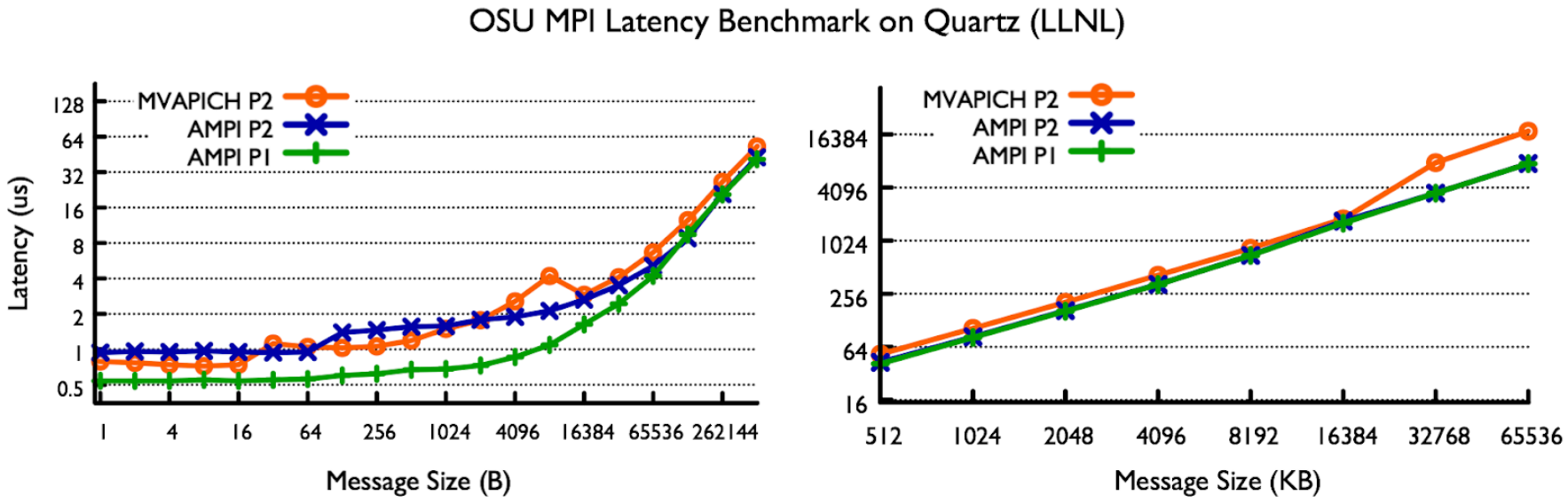
Zinc oxide nanowire

Scaling on Mira



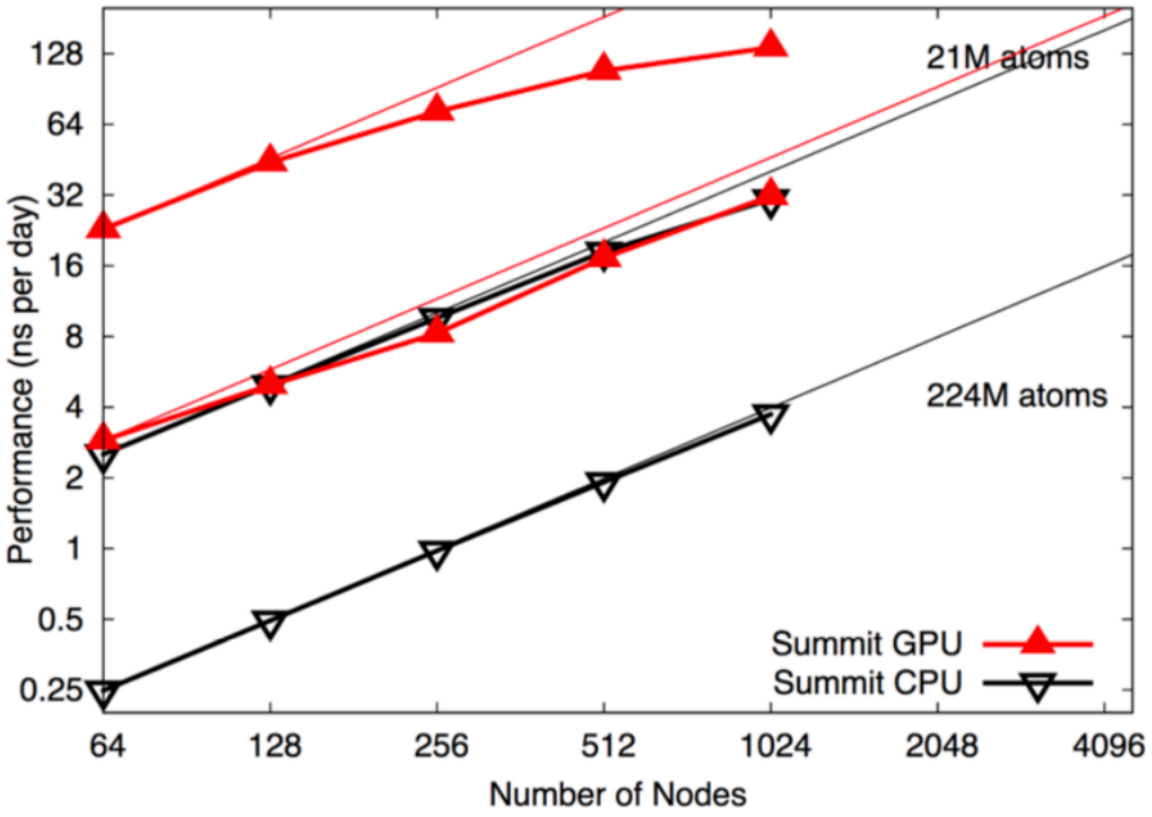
Adaptive MPI

- AMPI optimizes for Shared Memory systems
  - Idea: threading is hoisted within the MPI implementation
  - Key: utilize *user-space* shared memory for messaging between the many ranks co-located on the same node
    - Up to 2.33x lower latency and 4x higher bi-directional bandwidth compared to all process-based MPIs for large messages within a node



NAMD on Summit

- NAMD is one of the CAAR (Center for Accelerated Application Readiness) Applications to use Summit
- Results obtained with NAMD 2.13 and Charm 6.8.2
- Builds on PAMI-smp layer
- ~6x speedup over Titan



Charm++ v6.9.0 released on Monday

- We now require C++11 compiler support, and provide much better support for modern C++ usage in Charm++ applications
  - PUP support added for various std types and containers
  - Enhanced support for metaprogramming: see SpECTRE
- Hwloc integrated for more intuitive process and thread launching
- New Zero Copy API for communication:
  - Optimized for RDMA-enabled network layers and shared memory
- AMPI now implements most of MPI-3.1, has faster ULTs via Boost, lower memory footprint, communication optimizations
- LLVM OpenMP runtime integration
- Support for IBM POWER and ARM8 architectures
- Charm4Py and new GPU interface will be covered in talks today

Charm++ v7.0 Proposals

- Zero Copy communication API
  - "In-place" migration via new PUP interface
  - Collective communication routines built on top of the Zero Copy API
- Elimination of .ci files with a new C++ template-based interface
  - See Nils Deppe's talk from the 2018 Charm++ Workshop for one proposed interface
  - Will continue to support .ci files for backwards compatibility and for users that prefer it
- GPU interfaces
  - Ability to send/recv messages to/from GPUs via GPUdirect integration with the zero copy API
  - Kernel agglomeration
- Load balancing infrastructure refactoring to reduce memory and overhead, improve modularity, and simplify custom strategy development
- New optimized UCX layer in collaboration with Mellanox for Infiniband/RoCE
- Transition to GitHub for development, code review, and issue tracking

17th Annual Charm++ Workshop (2019)

- Two-day workshop at UIUC
  - Followed by an optional half-day tutorial on Charm++/AMPI
- Dates will be announced in the coming weeks
  - Will be held in April or May
  - Talk proposals due in March
- Topics:
  - Parallel algorithms and applications
  - Adaptive runtime system techniques
  - Novel parallel programming models, languages, and libraries
  - Runtime management of energy, power, and heat
  - Parallel performance analysis
- All talks (slides and video) from past workshops are posted online: http://charm.cs.illinois.edu/workshops/

Charmworks Inc

- Charmworks provides commercial support for Charm++, AMPI, and their users
  - Charm++/AMPI continue to be free for academic use, and the source code is freely available
  - All Charmworks development is contributed back into the freely available University version of the source code
- Charmworks offers tutorials, consulting, and development of Charm++/AMPI and applications

- Visit the Charmworks booth #3058 in the Exhibit Hall

Next: Application & Developer Talks

Followed by Open Discussion / Q & A