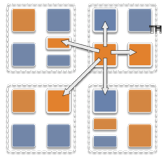
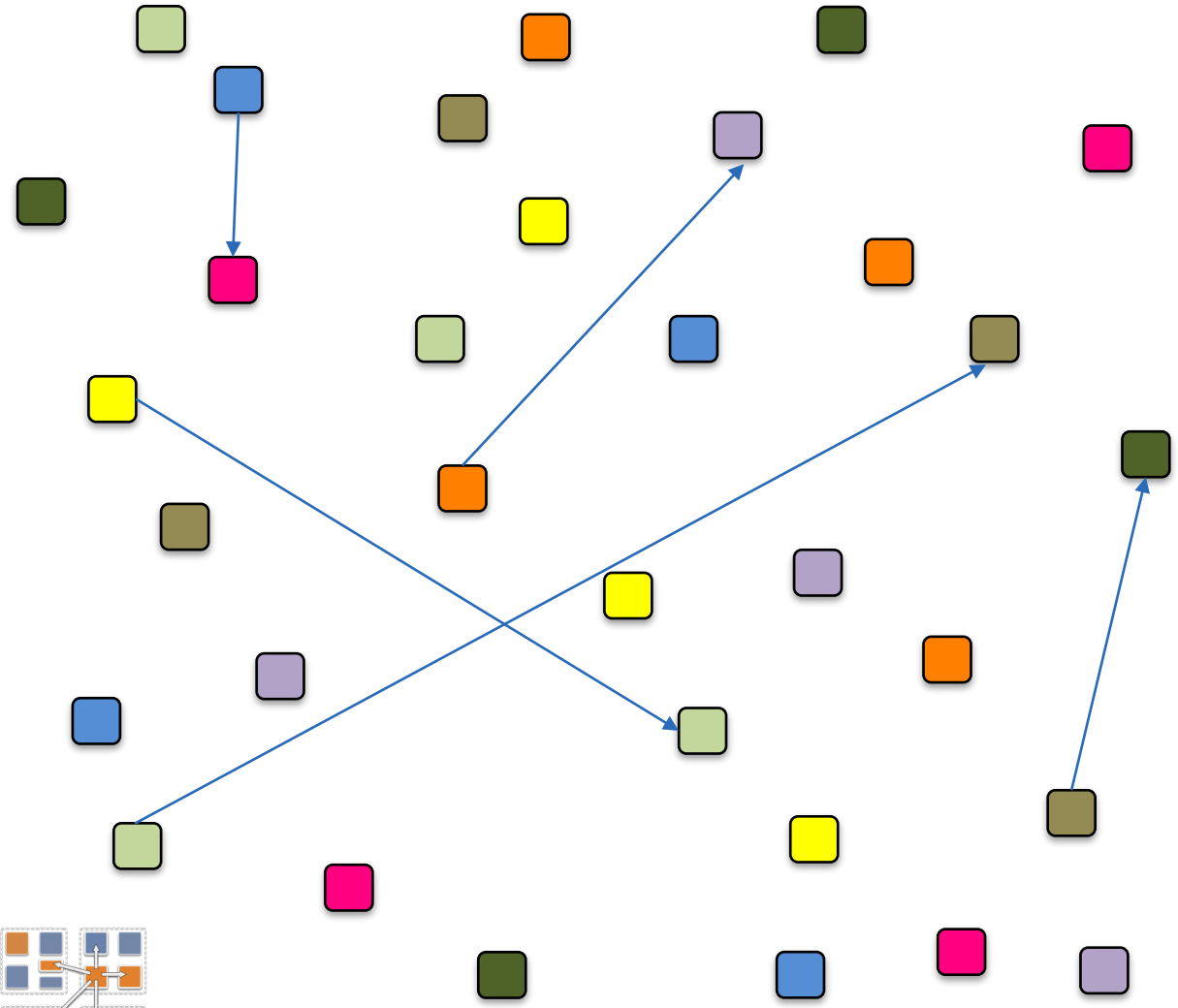
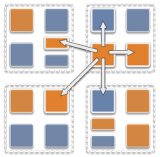
Task Based Algorithms and Applications: Charm++

Laxmikant (Sanjay) Kale
http://charm.cs.illinois.edu
Parallel Programming Laboratory
Department of Computer Science
University of Illinois at Urbana Champaign



- A Charm++ computation consists of multiple collections of globally visible objects
- Each collection is individually indexed





Processor 0

Processor 1

Processor 2

Processor 3

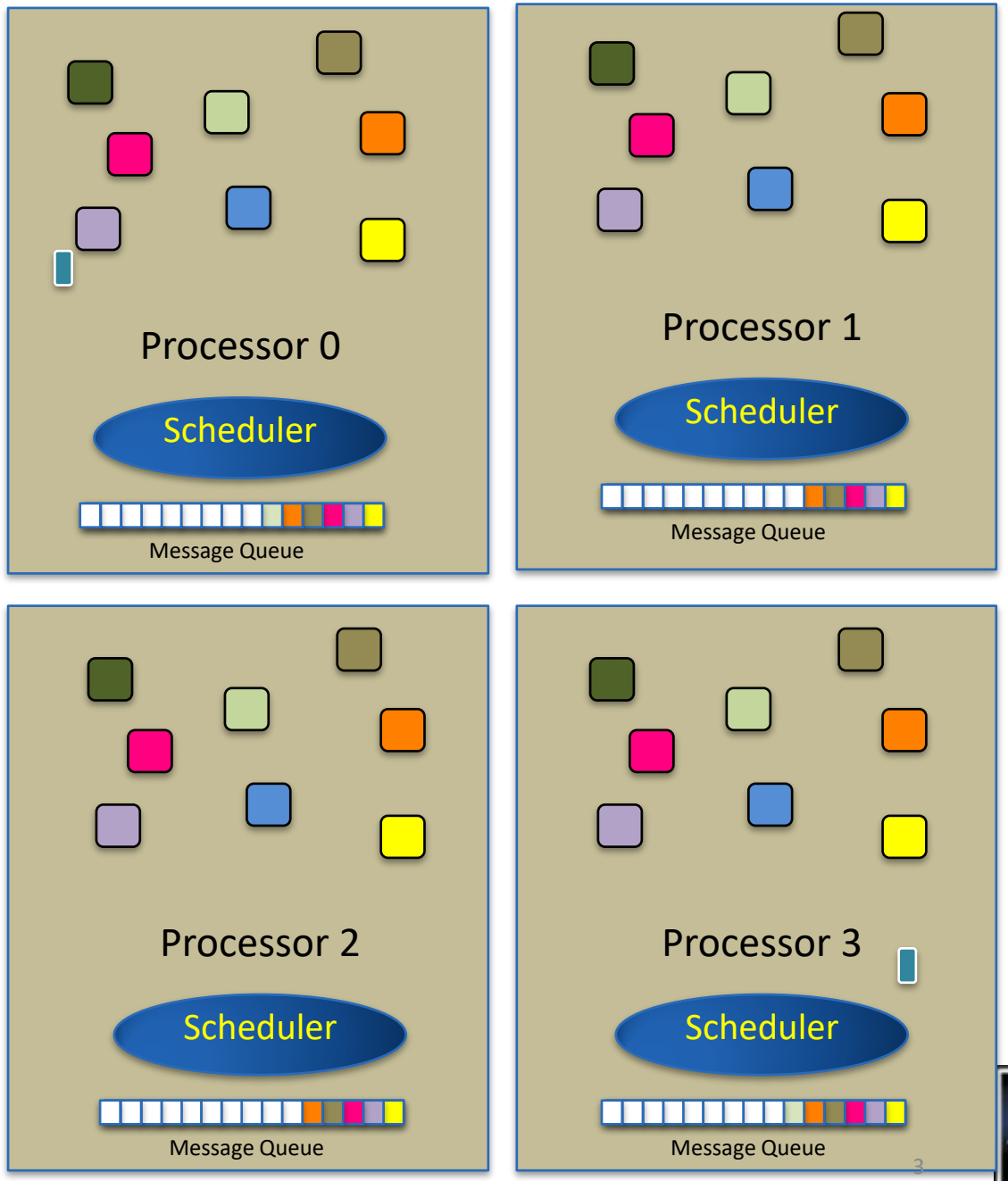
PPL UIUC

- A Charm++ computation consists of multiple collections of globally visible objects
- Each collection is individually indexed

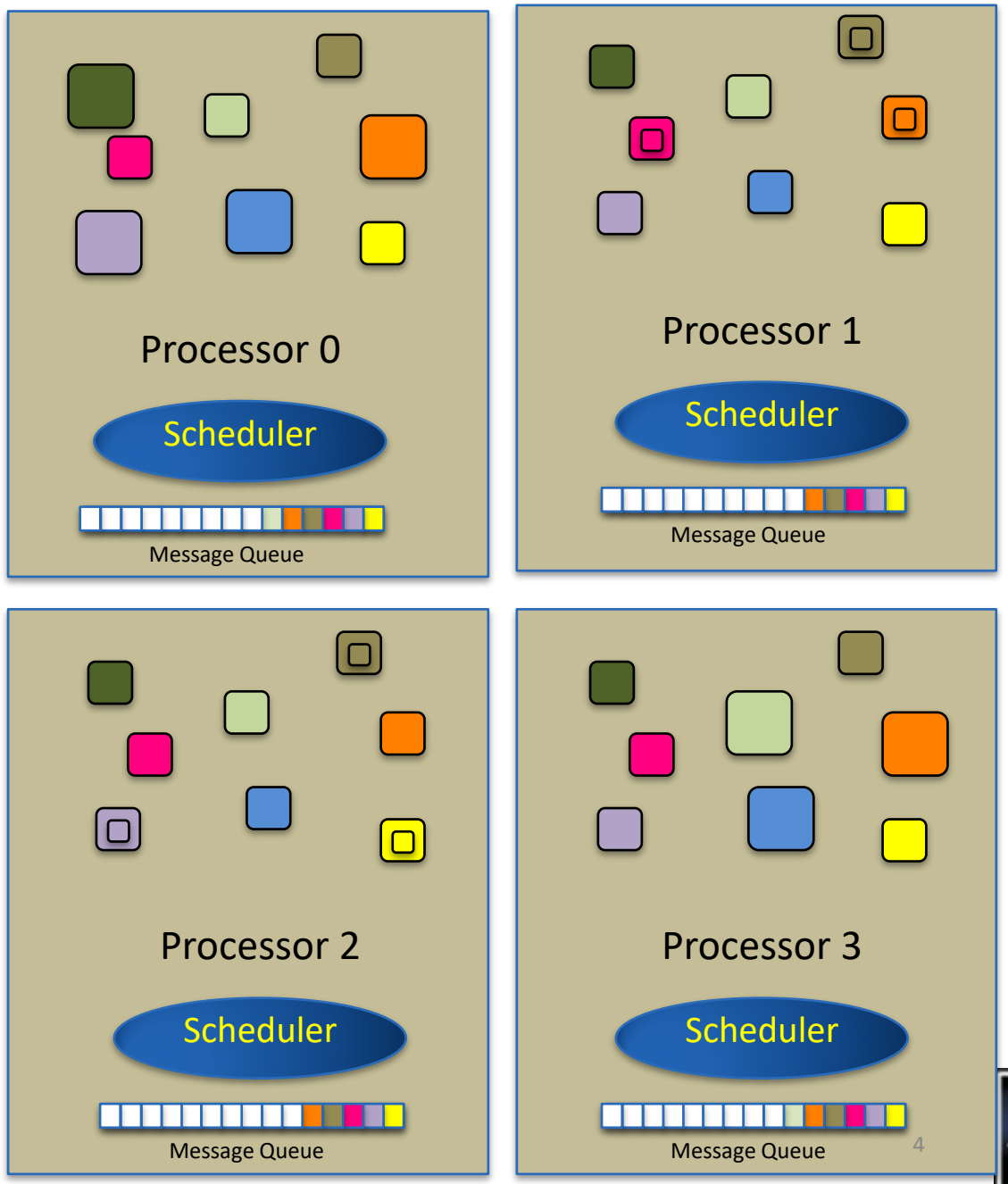
- Objects are assigned to processors by the runtime system
  - Programmer does not need to know where an object is located

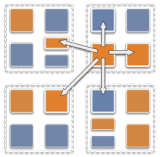
- Scheduling on each processors is under the control of a user-space message-driven scheduler

- Example: an object on 0 wants to invoke a method on object A[23]
  - The Runtime System packages the method invocation into a message
  - Locates where the target object is
  - Sends the message to the queue on destination processor
  - Scheduler invokes the method on the target object

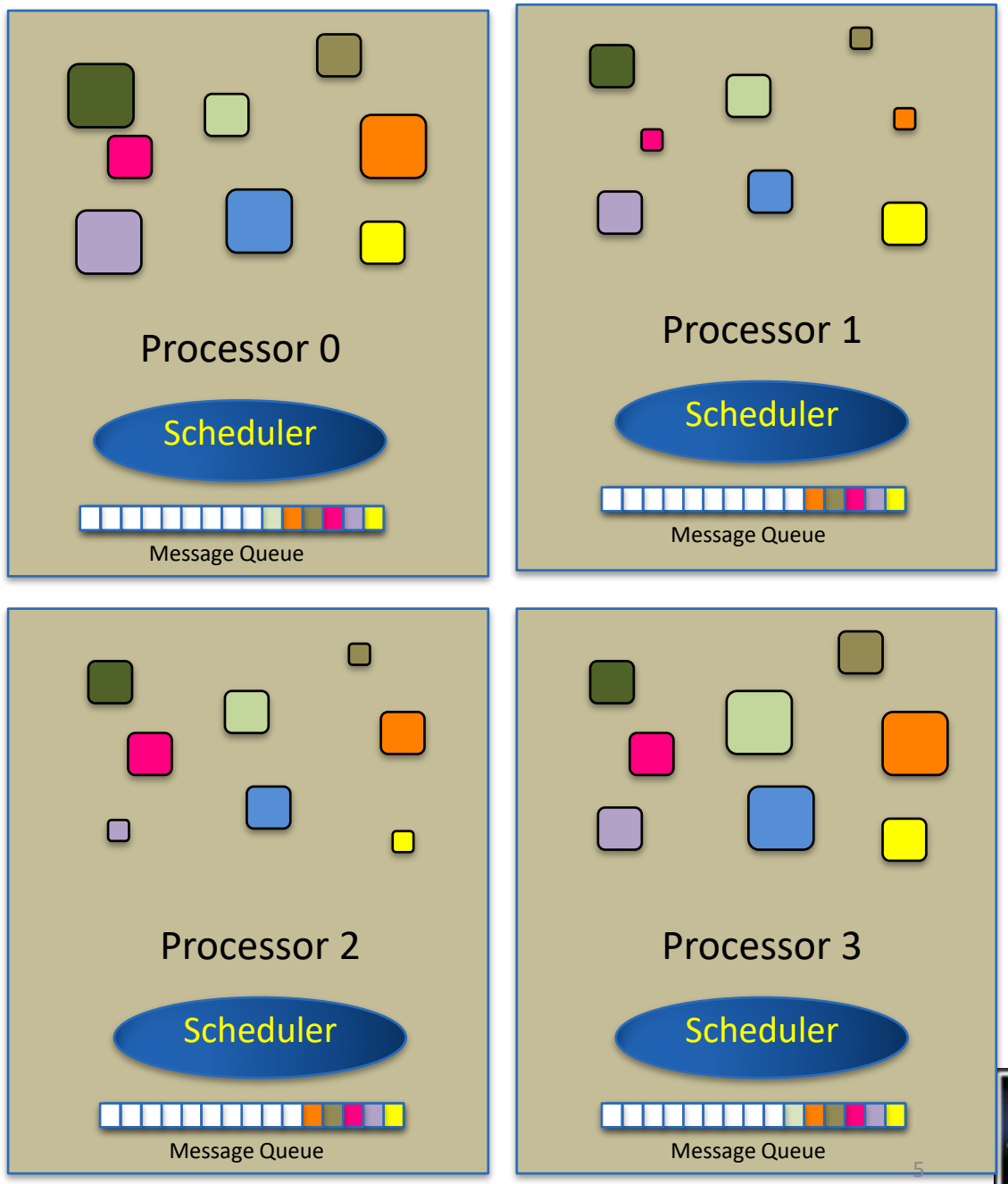


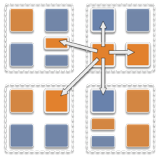
The runtime system knows which processors are overloaded, which objects are computationally heavy, which objects talk to which





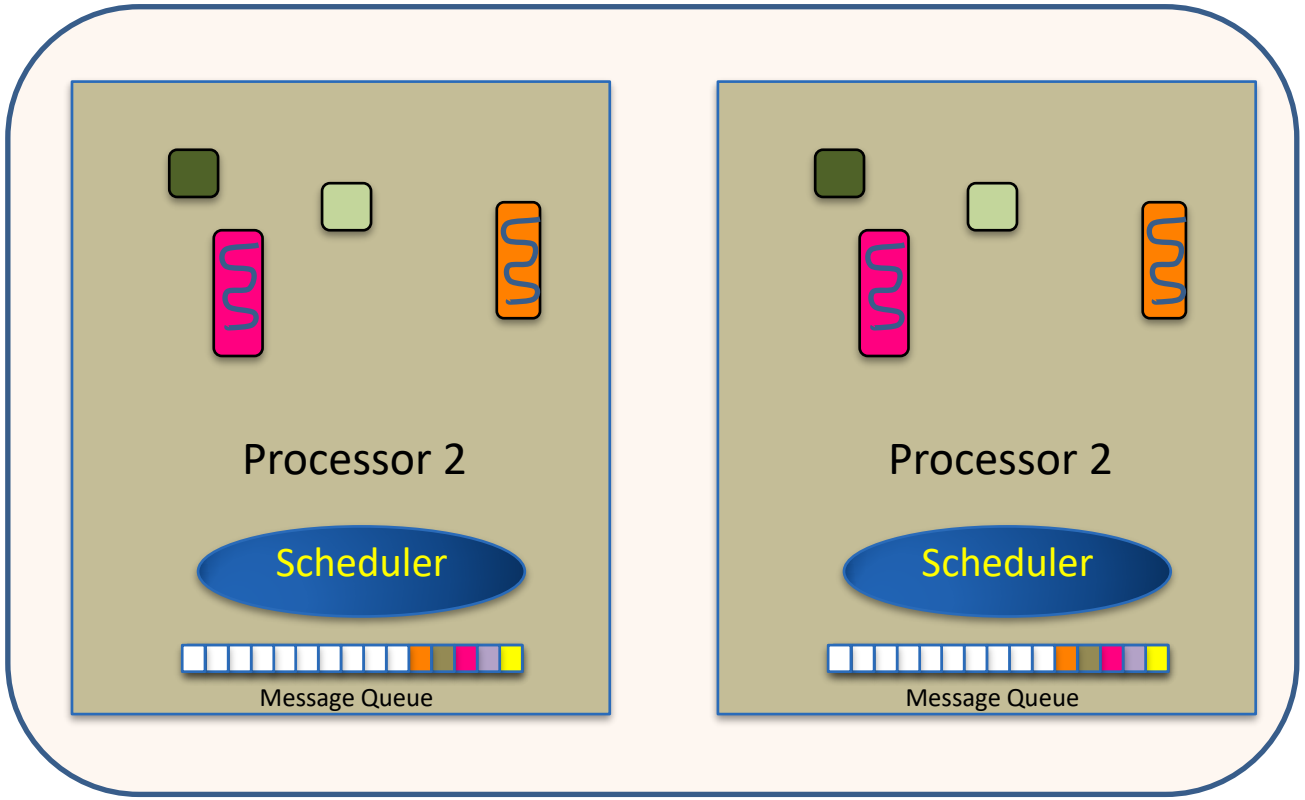
Using this information, it migrates objects to rebalance load and optimize communication

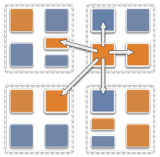




Converse: flexible scheduling enables sophisticated task management

- Converse is the underlying scheduling system
- Supports
  - User level threads (ULTs), as well as stackless tasks (active messages), or object method invocations
  - Once you have per-core scheduling control, all kinds of tasks, loops, runtime strategies can be implemented
- Converse is from 1996, still being used in Charm++
  - Argobots is a modern equivalent

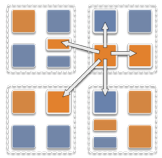




Capabilities and Applications

- Capabilities
  - Dynamic load balancing
  - Fault Tolerance
  - Elasticity:
    - change the set of nodes allocated to a job
  - Communication optimizations
  - Out-of-core execution
  - Energy optimizations
- Programming Systems
  - Adaptive MPI
  - Charm4Py
  - Charades
  - Experimental: MSA, Charisma, CharJ, …

- Some example Applications
  - NAMD (Biophysics)
  - ChaNGa (Astronomy/Cosmology)
  - OpenAtom (Elect. Structure)
  - Enzo-P and Cello (Astronomy/Cosmology)
  - SPeCTRE (Black hole mergers, etc.)
  - Many more
- Interesting application papers
  - BRAMS (weather simulation)
  - Cloth Simulation
  - Crack propagation,
  - MiniApps
  - Many more

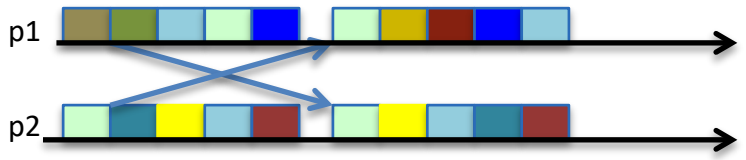


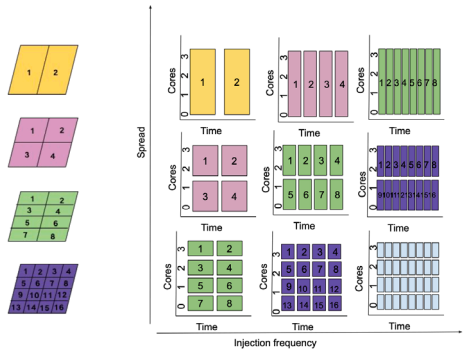
PPL UIUC

Current State and Recent Results

- Energy control by turning cores on/off dynamically
- Adaptive MPI advances at Charmworks
  - Standard compliant MPI,
  - More automatic ways of running MPI codes virtualized via Charm++

- Communication optimizations
  - Spreading communication injection over time
  - Dynamically controlled ratio of "drone" cores to cores that anchor objects, and spread of OpenMP loops
  - Similar efforts for GPGPUs





More info: https://charm.cs.Illinois.edu
https://hpccharm.com