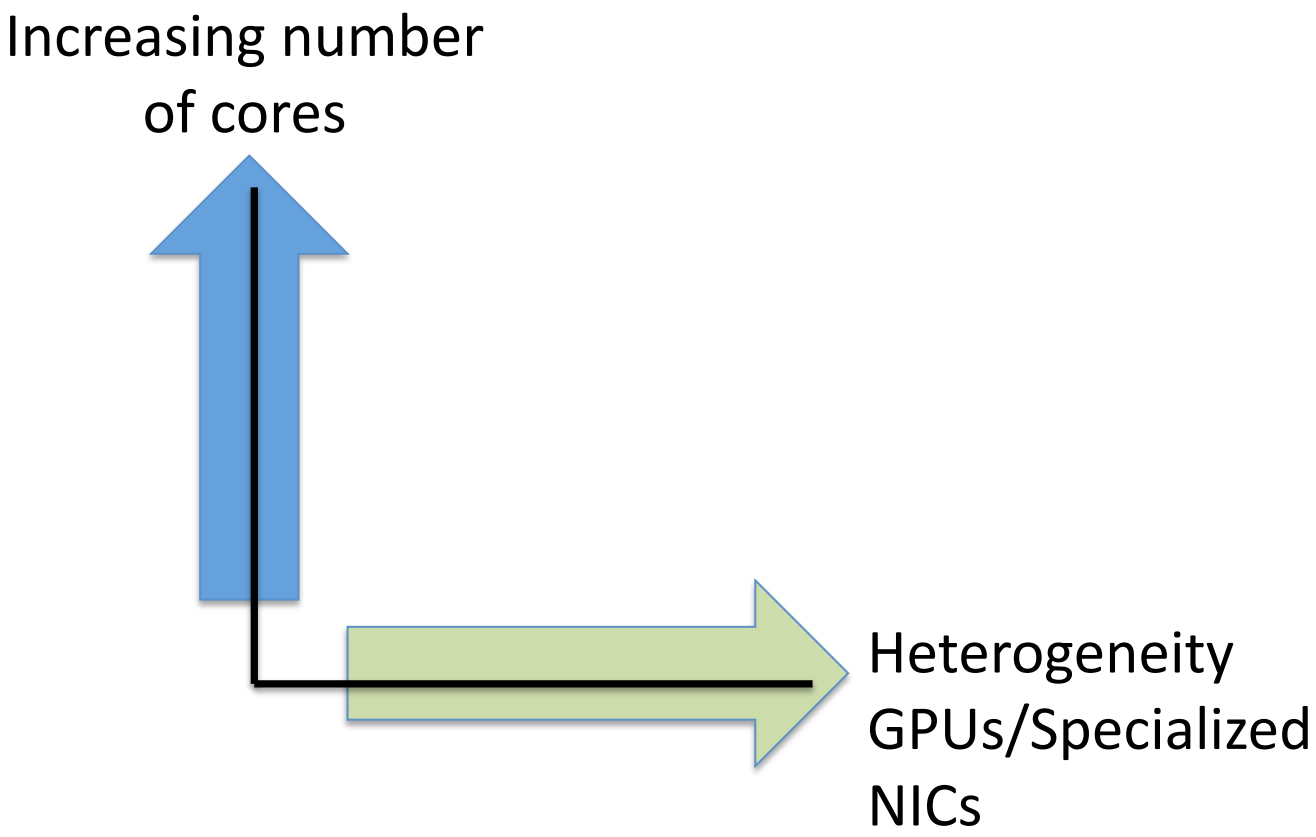
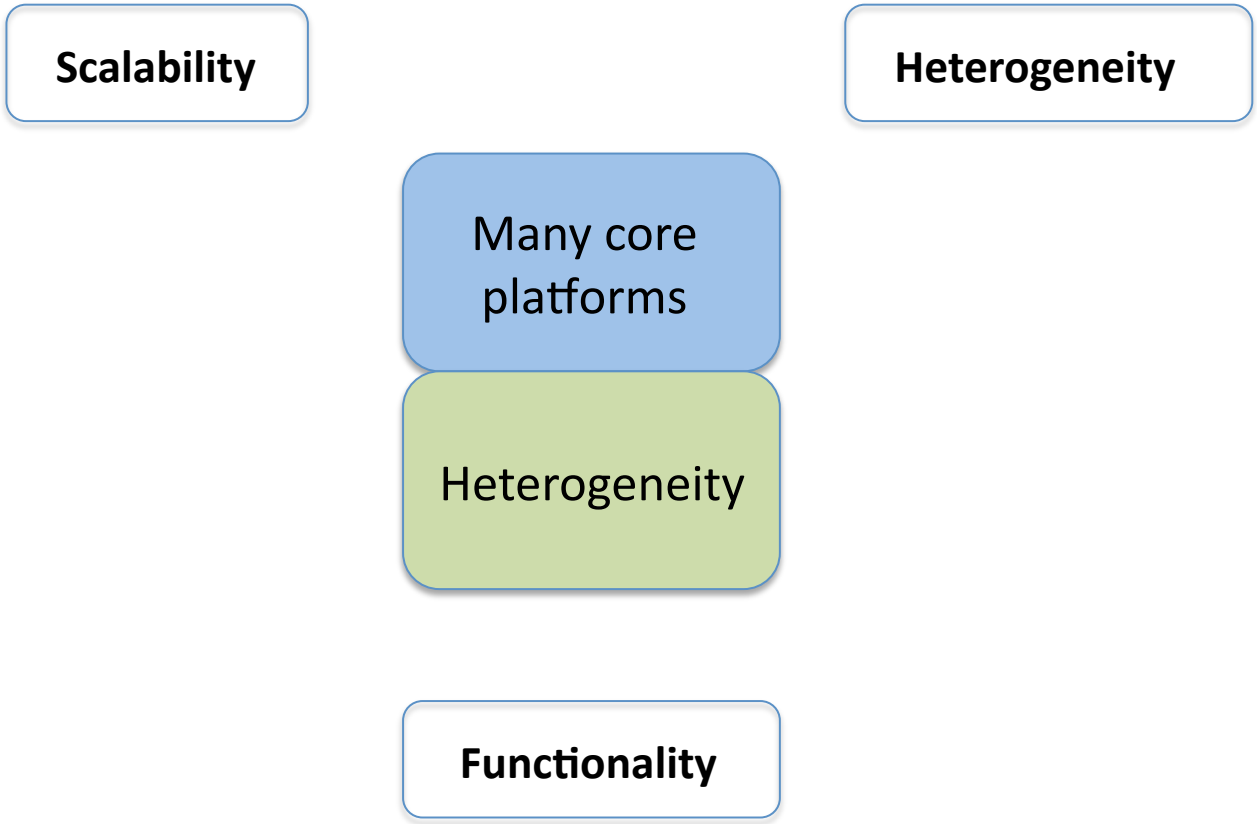
Islands of Cores
Coordinated Resource Management for Manycore platforms

Priyanka Tembey, Ada Gavrilovska, Karsten Schwan
(Georgia Tech)

Platform Trends



System Software Challenges



System Software Challenges

Scalability

- Due to shared state and cache coherence protocols
- "Fix-it" patches to existing kernels when number of cores increase

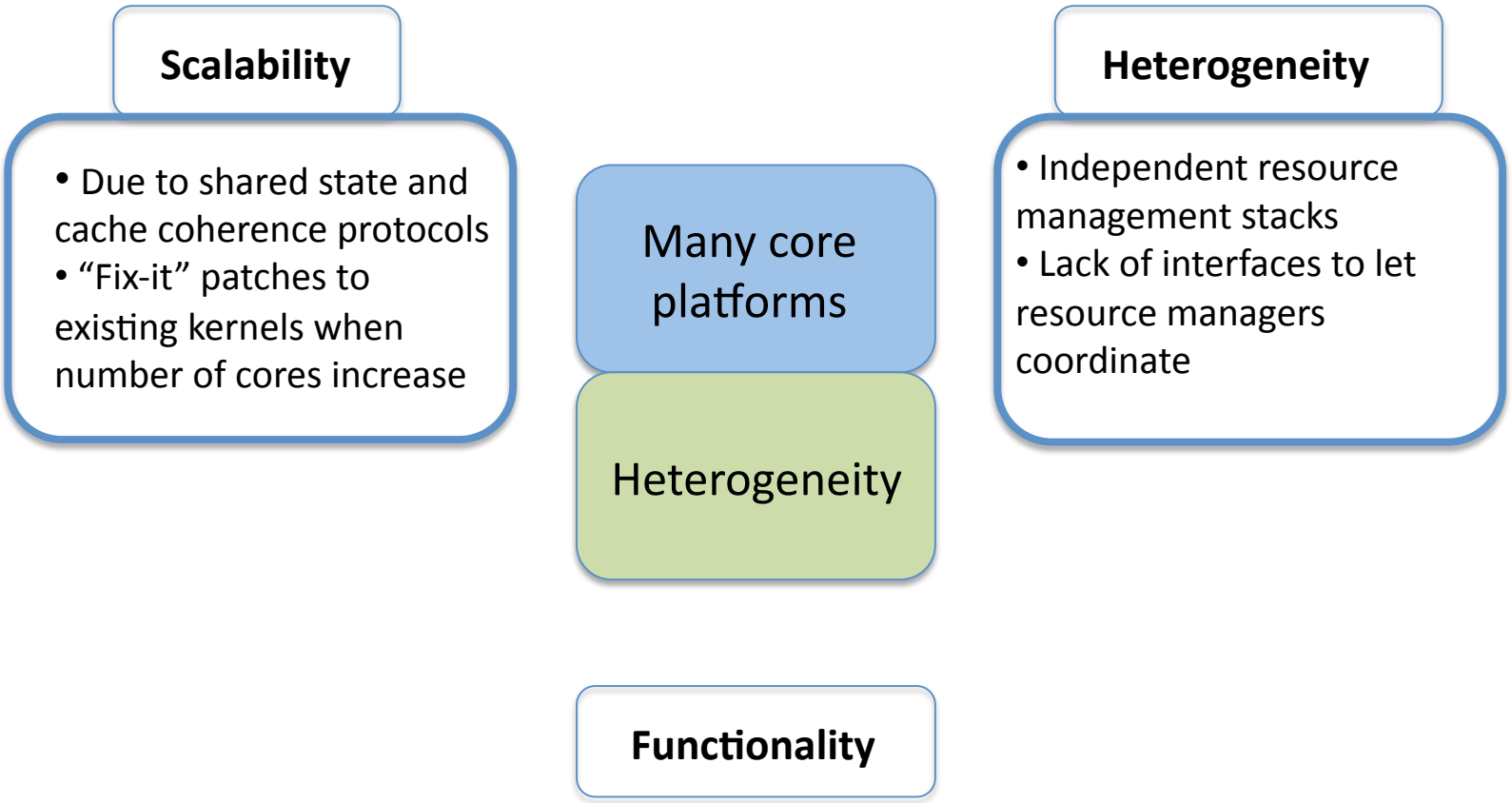
Many core platforms

Heterogeneity

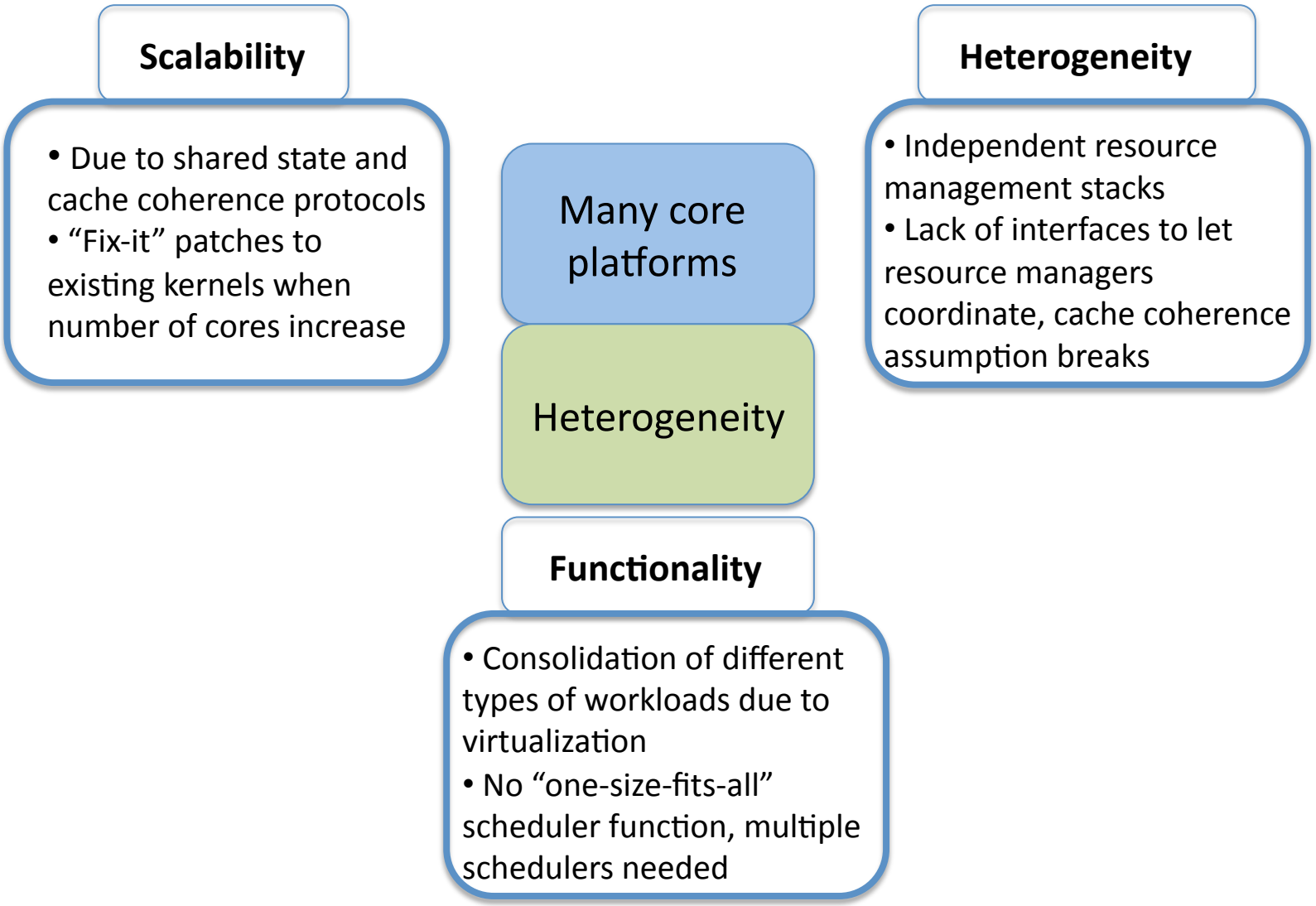
Heterogeneity

Functionality

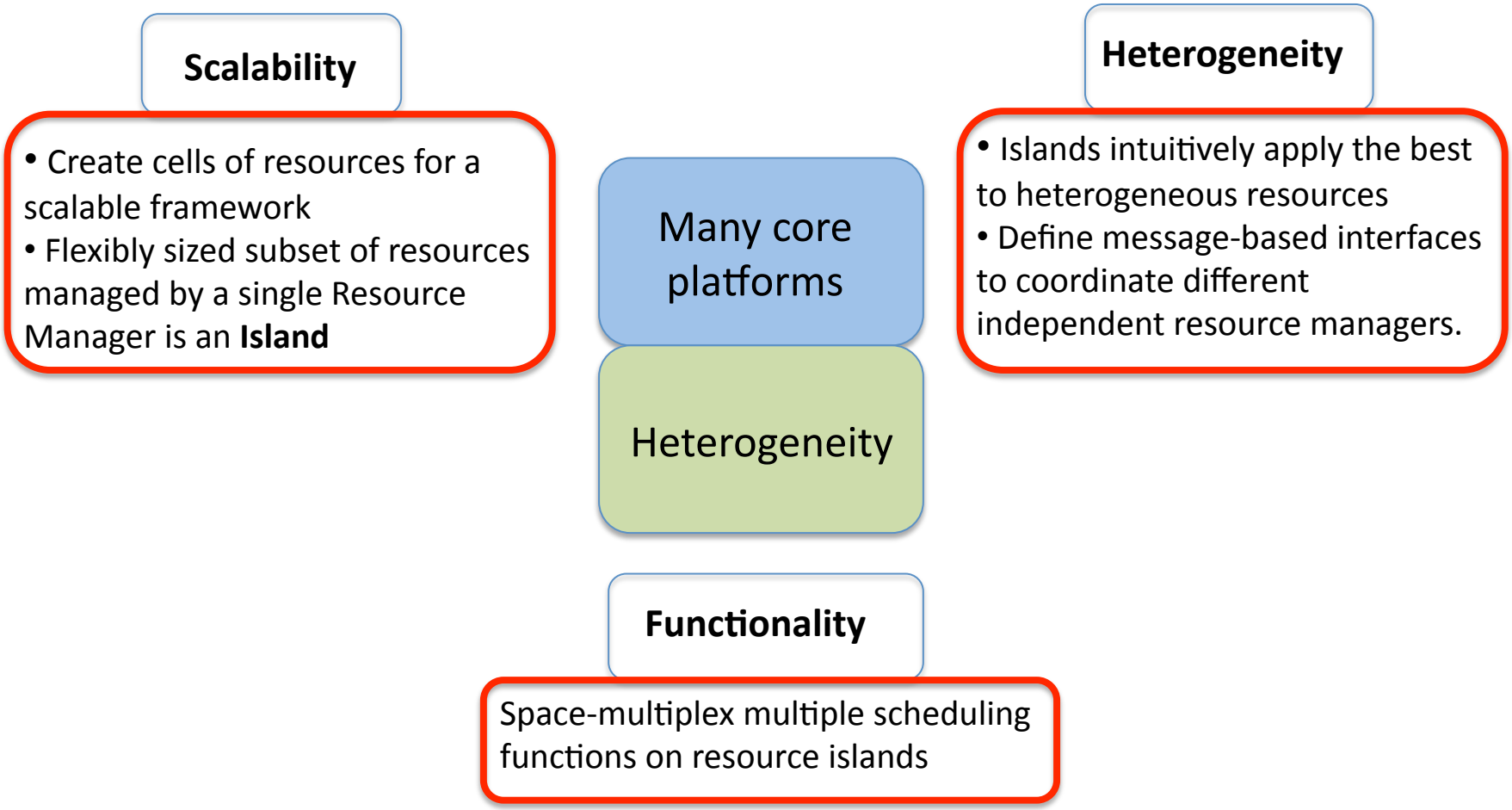
System Software Challenges



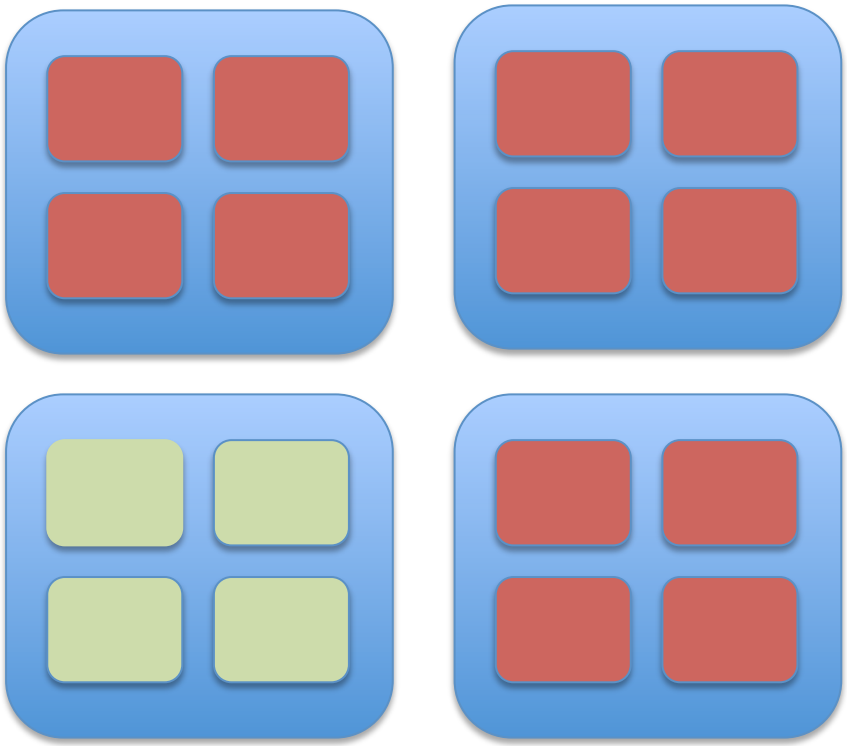
System Software Challenges



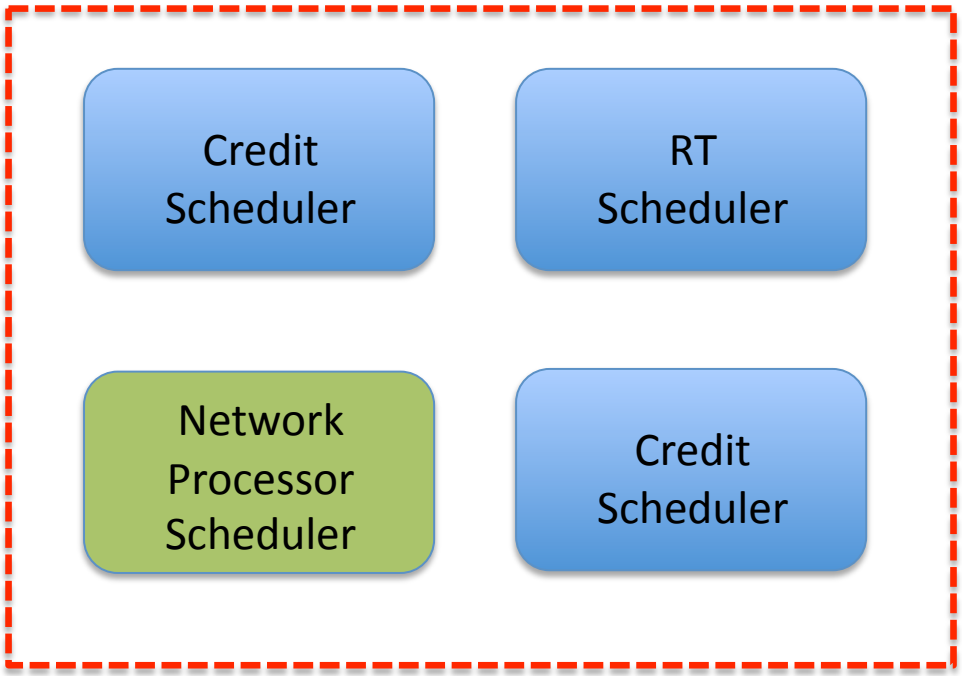
Islands of Cores: New System software abstraction



An Islands Example

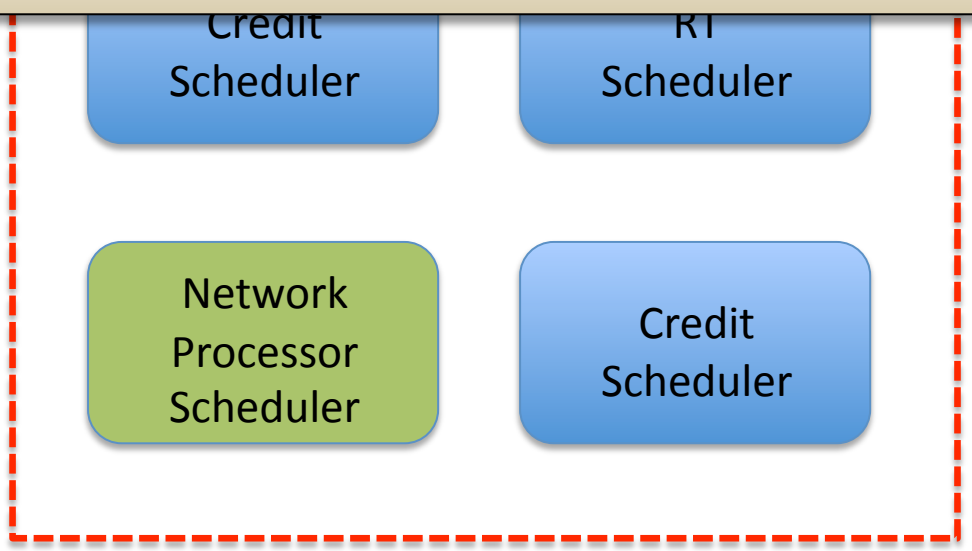


An Islands Example

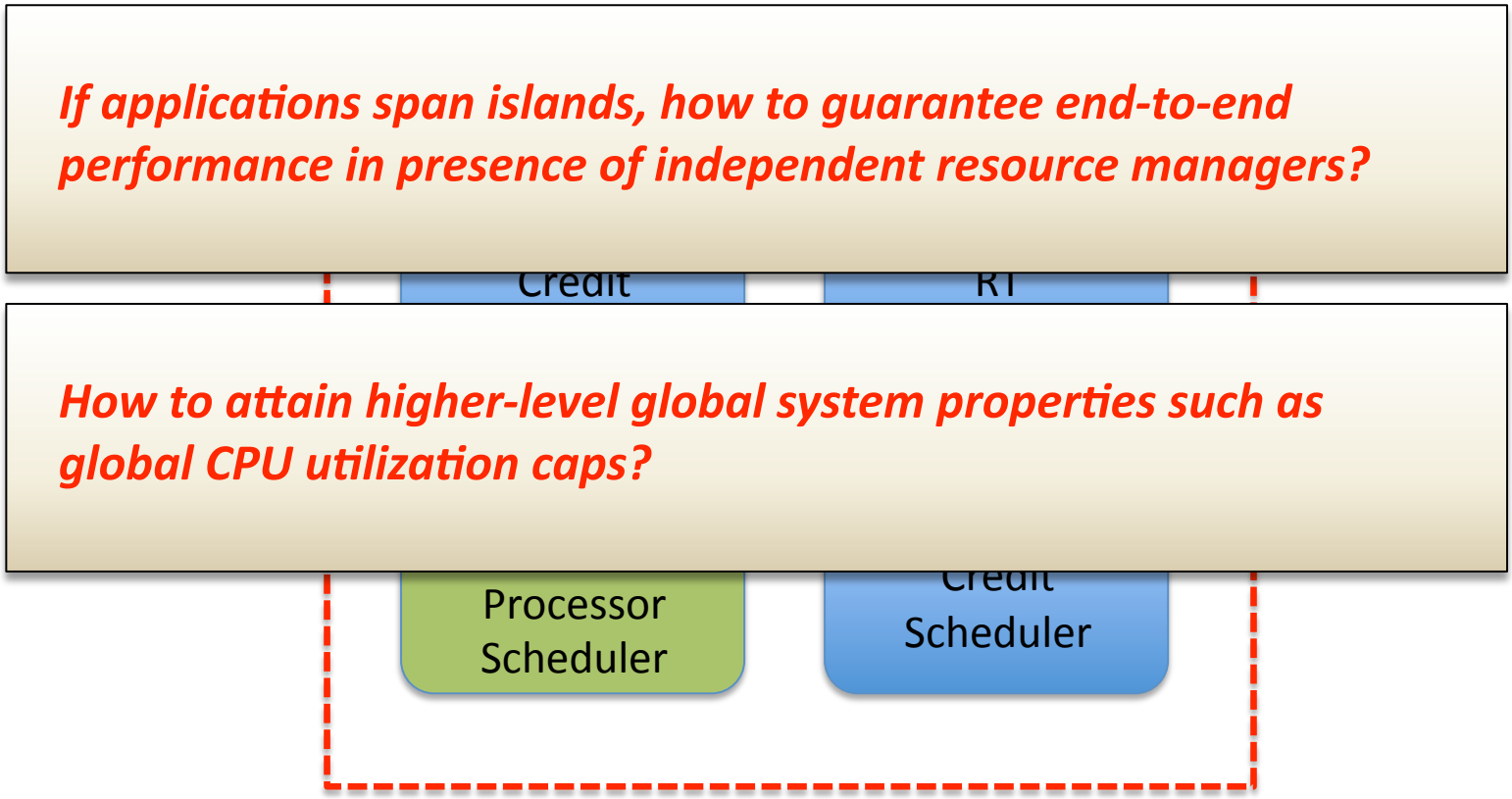


Coordinated Islands are Important

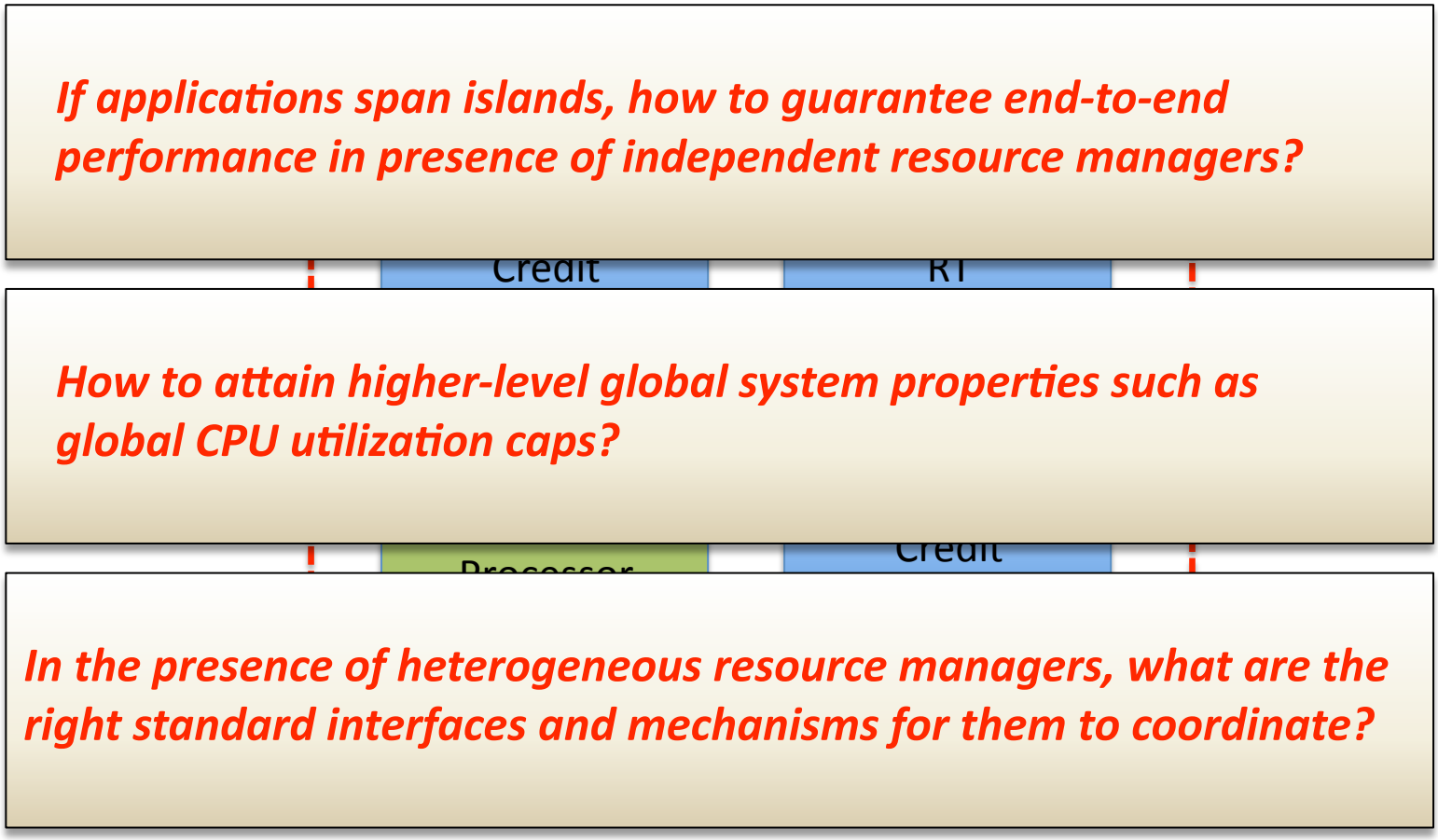
If applications span islands, how to guarantee end-to-end performance in presence of independent resource managers?



Coordinated Islands are Important

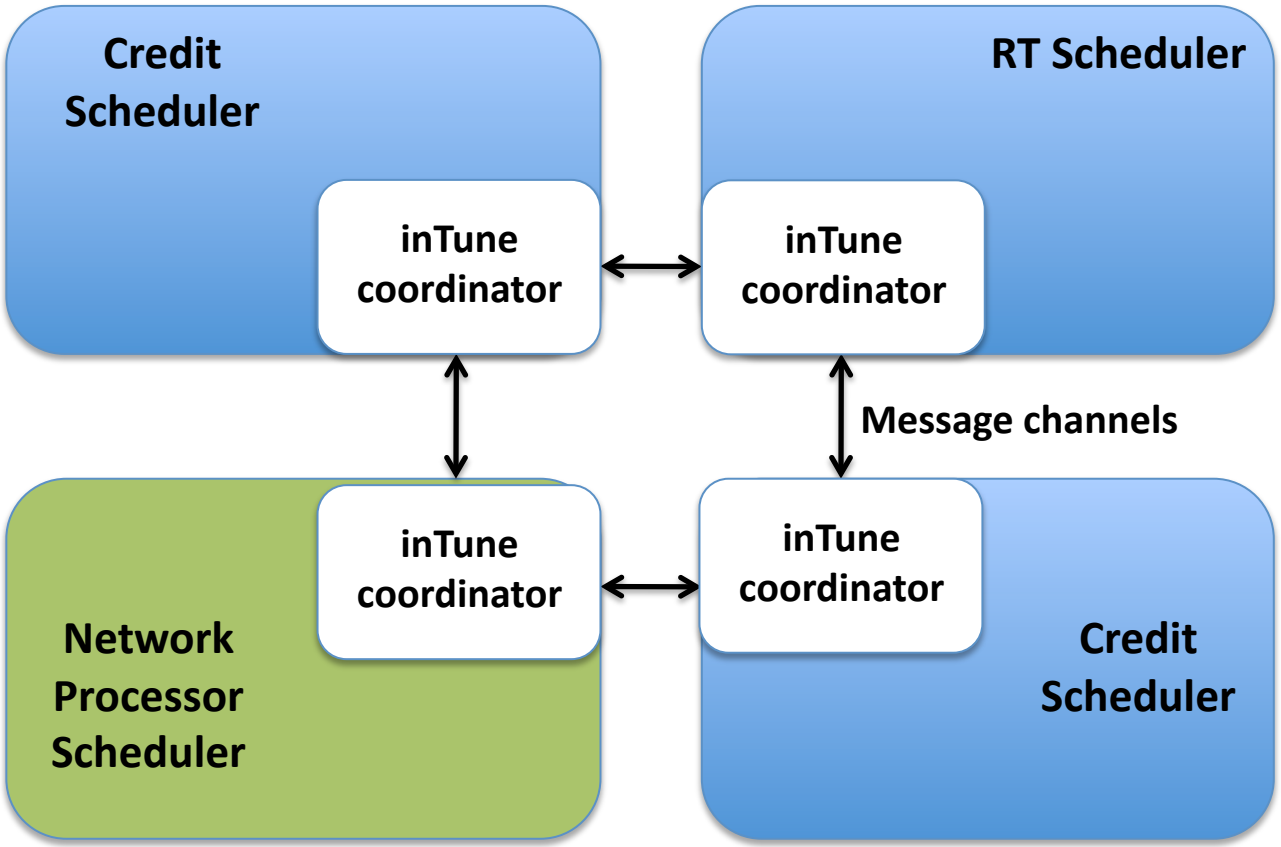


Coordinated Islands are important



inTune

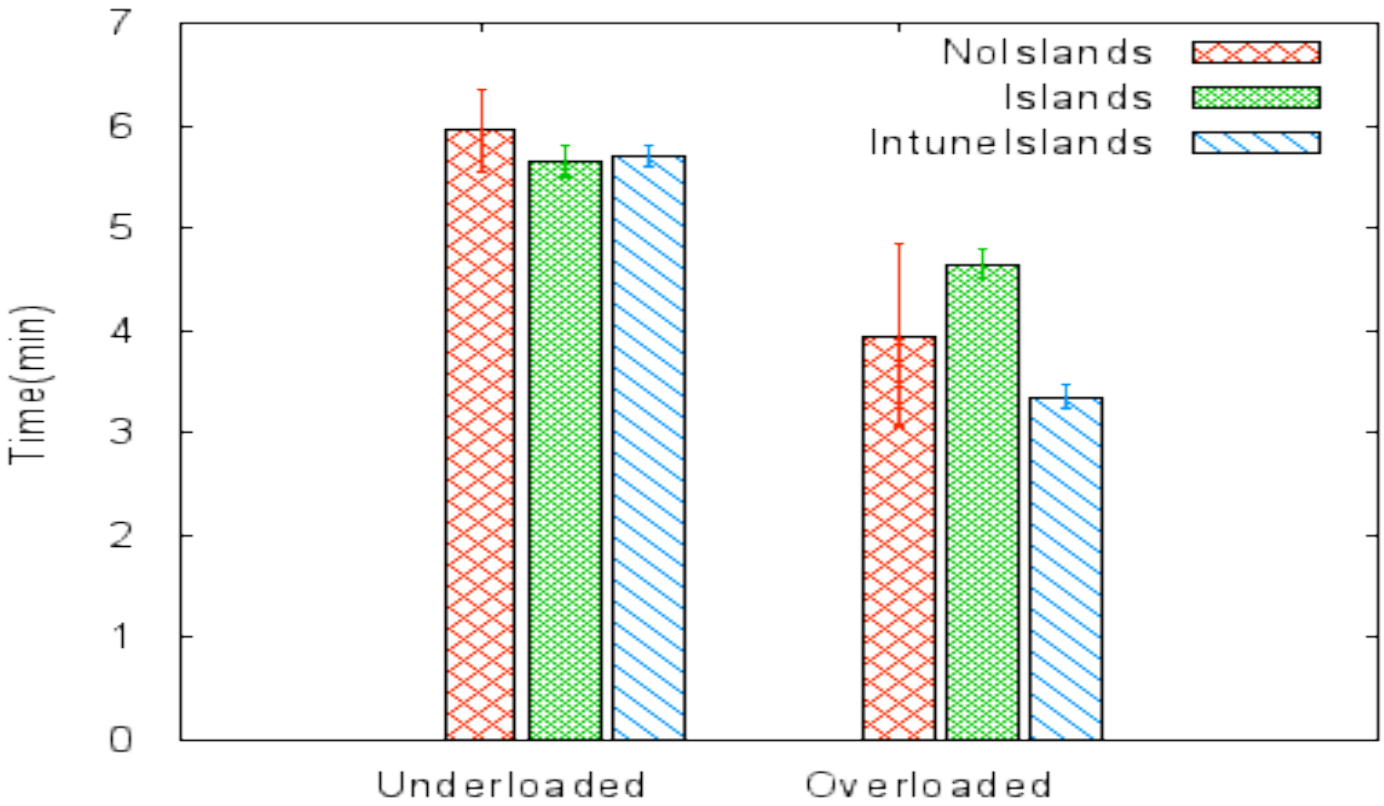
Island Coordination Framework



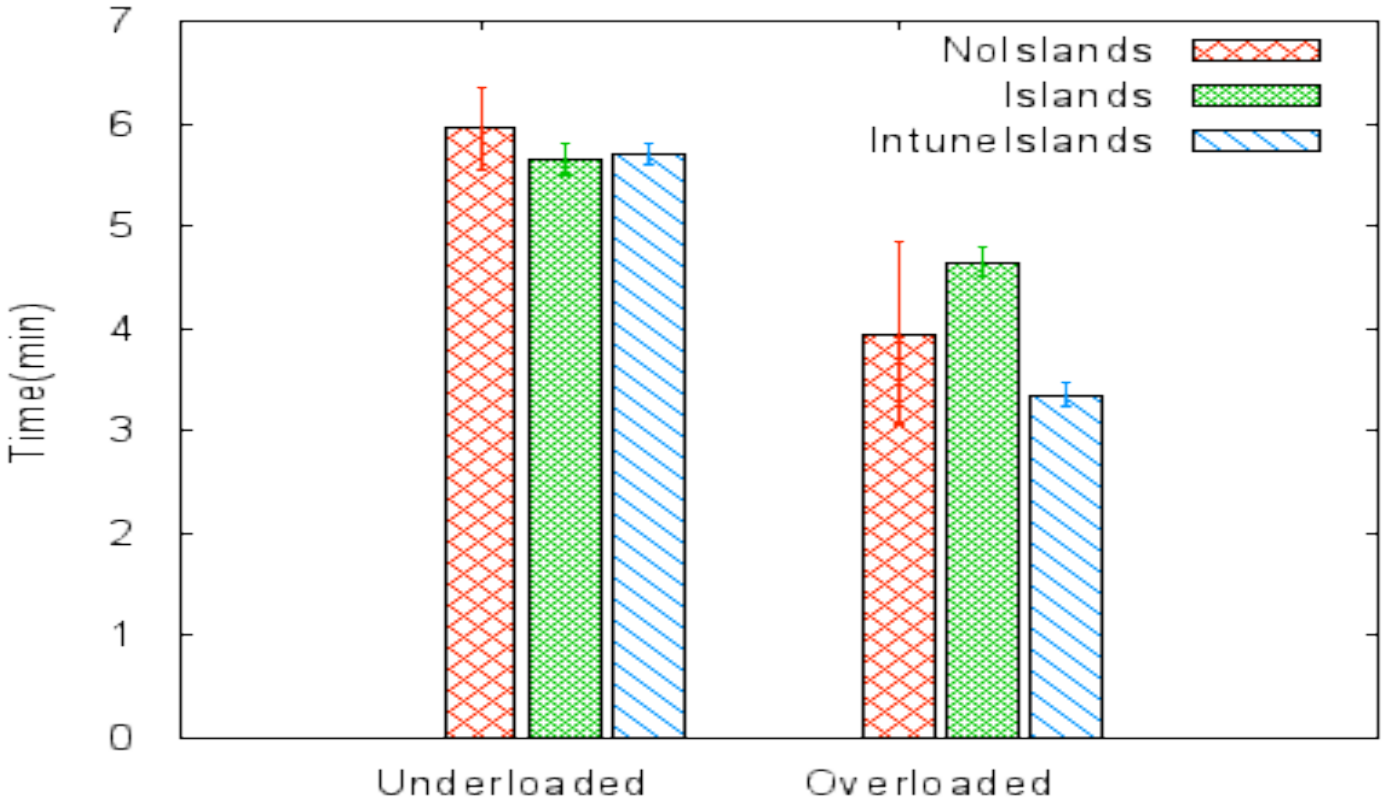
inTune Coordination Messages

- **Tune**: Increase/Decrease resource allocation to a consumer in a remote island
- **Trigger**: "Urgent" Tune
- **Borrow**: Increase size of island resources (limited by hardware cache coherence)
  - **Borrow_any**
  - **Borrow_preferred**
- **Release**: Release resources to lender island

Benefits of inTune for CPU Islands



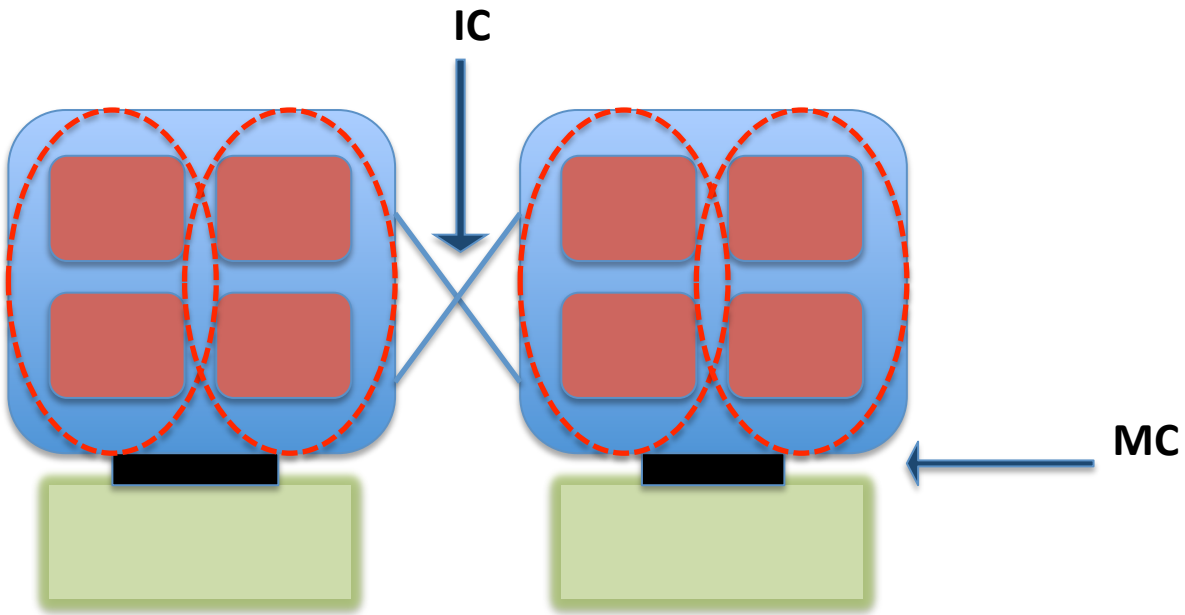
Benefits of inTune for CPU Islands



- **For underloaded platforms, coordination helps in reducing performance variation**
- **As consolidation levels increase, performance variation and absolute performance gains increase close to 20%**
- **Islands must not be statically sized, and should adapt to application's elastic needs**

Islands amongst Shared Resources

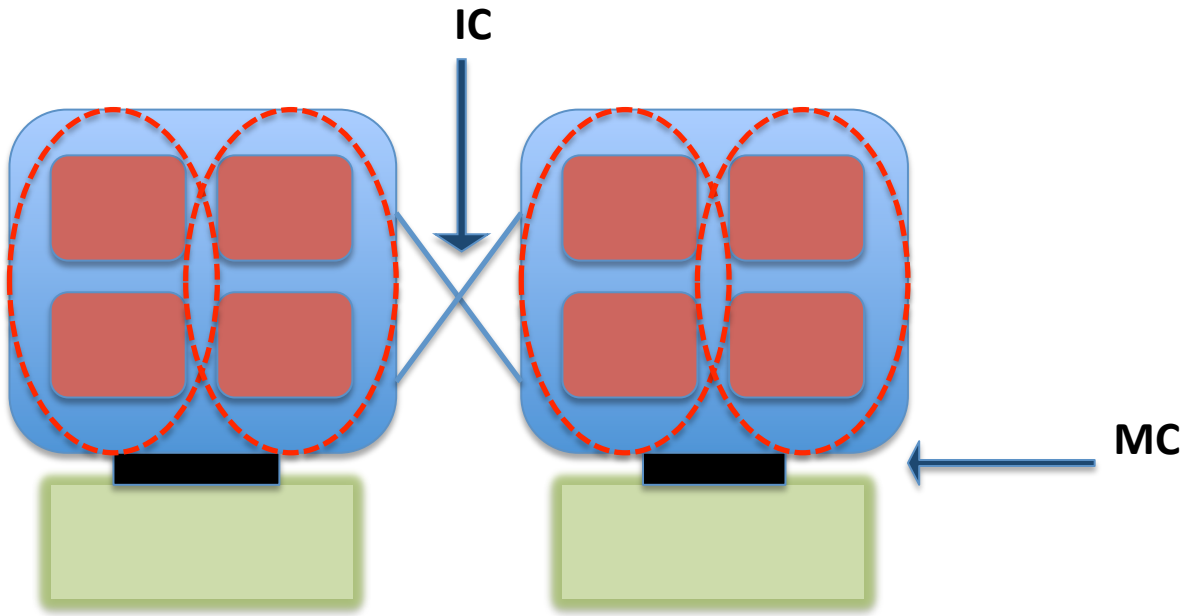
CPUs are not independent entities. Due to platform architecture, they share resources such as Memory controllers and Shared Interconnect Bandwidth



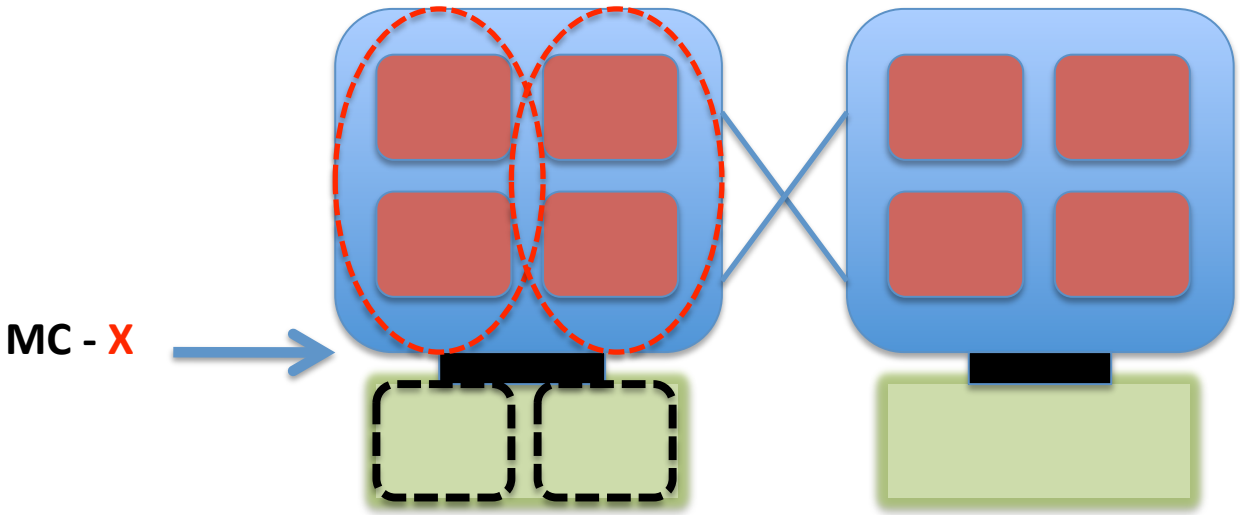
Islands amongst Shared Resources

CPUs are not independent entities. Due to platform architecture, they share resources such as Memory controllers and Shared Interconnect Bandwidth

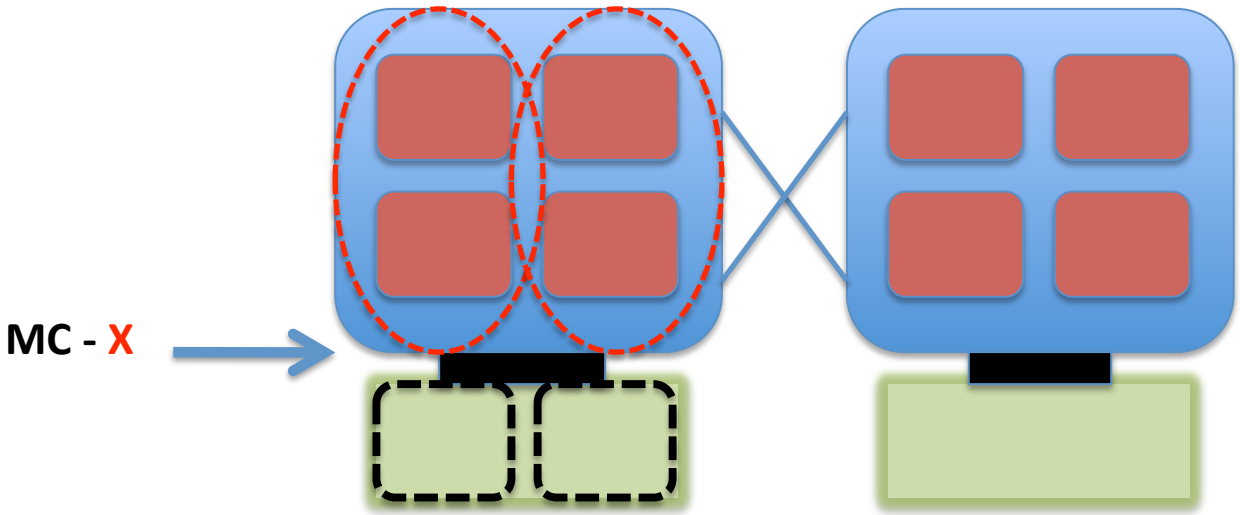
How should islands be created in presence of shared resources? Which inTune mechanisms can be used to provide performance isolation to applications?



2 VM Example

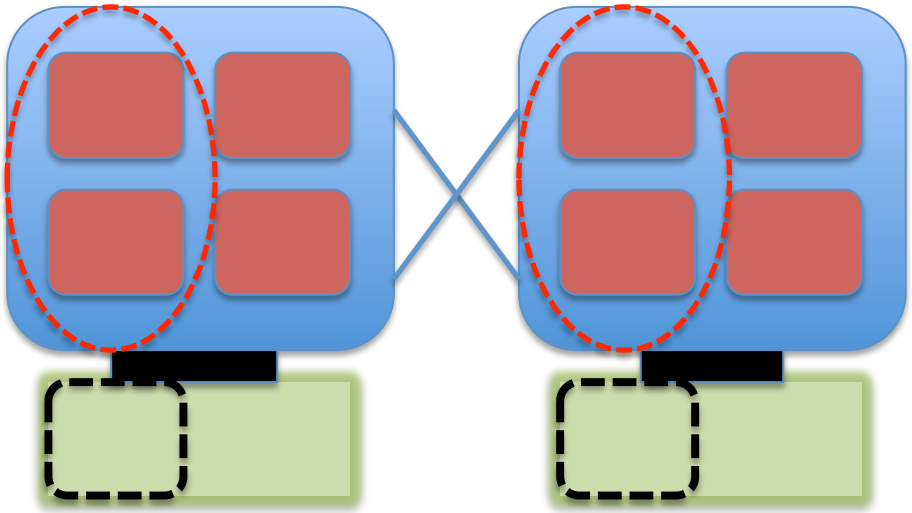


2 VM Example



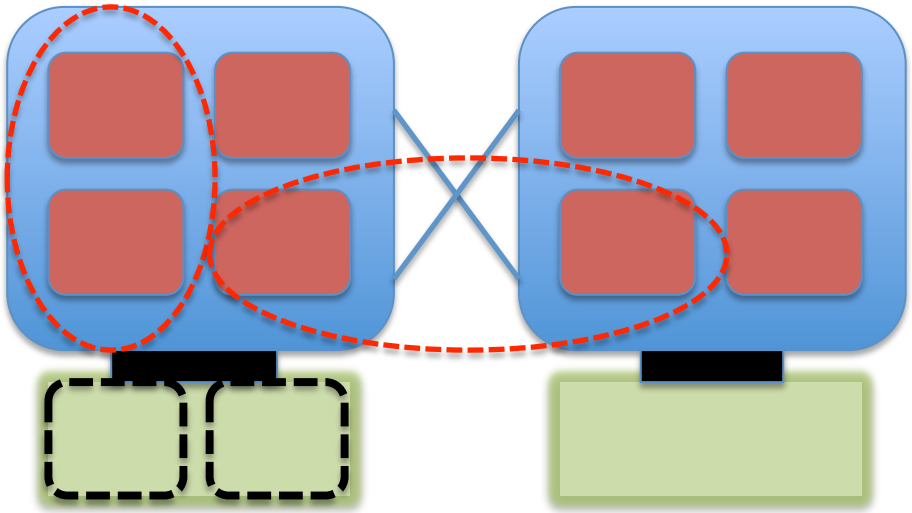
VM-1 memory bandwidth drops by 35-40% in presence of VM-2 sharing MC

2 VM Example: Performance Isolation for VM-1



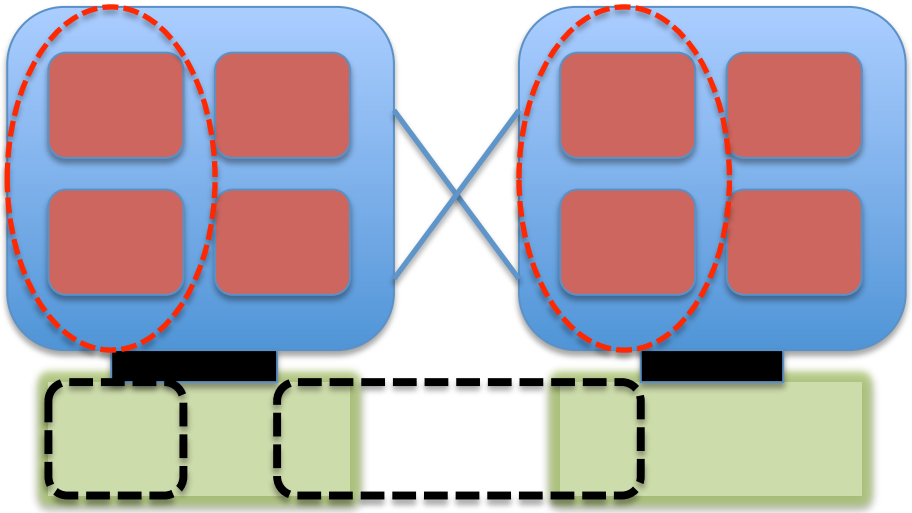
Using hardware isolation, VM-2 can be pinned to Node-2, but can islands add more flexibility?

2 VM Example



- "Borrow" CPU from neighbor island, so that some accesses directed to on-chip IC.
- VM-1 bandwidth lower by 10-15%.

2 VM Example

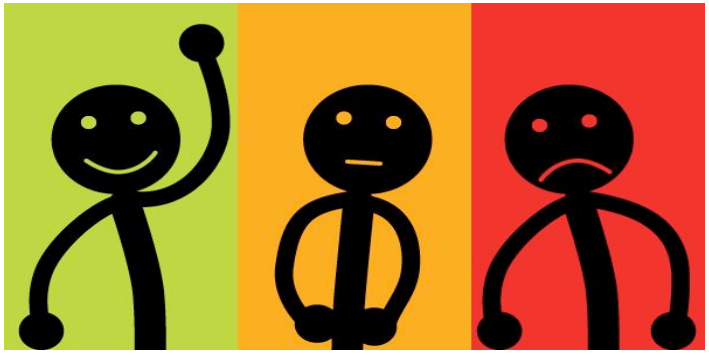


- "Borrow" Memory and CPU from neighbor island to see added benefits
- VM-1 bandwidth low by 5-10%

Summary

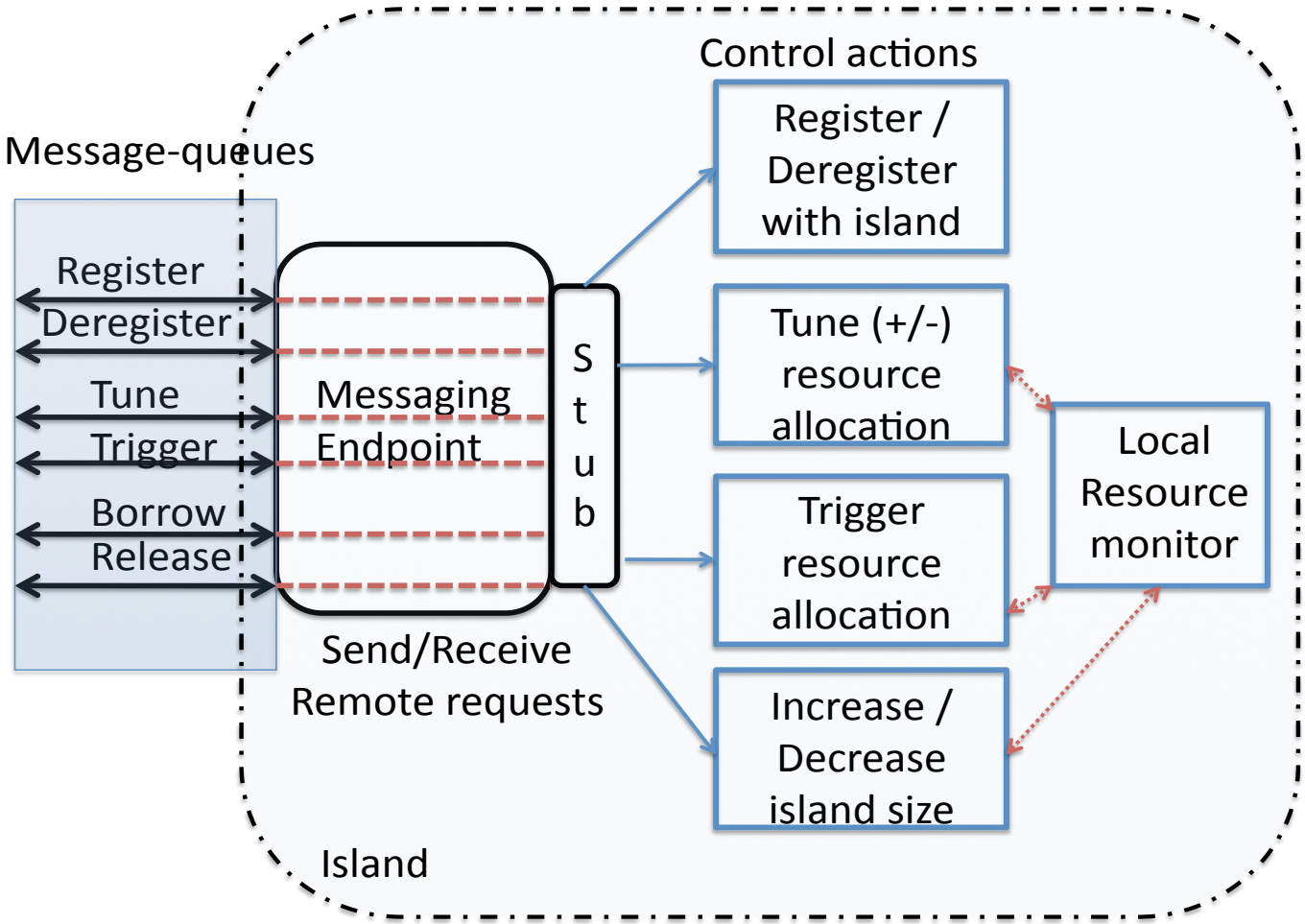
- Islands and inTune **coordination is important** to achieve higher level policies like CPU caps
- Coordinated Islands show **lesser performance variation**
- Due to platform architecture and shared resources, **CPU and Memory islands need to coordinate** as well
- Islands important for flexibility and performance isolation

Thank you.
Questions?

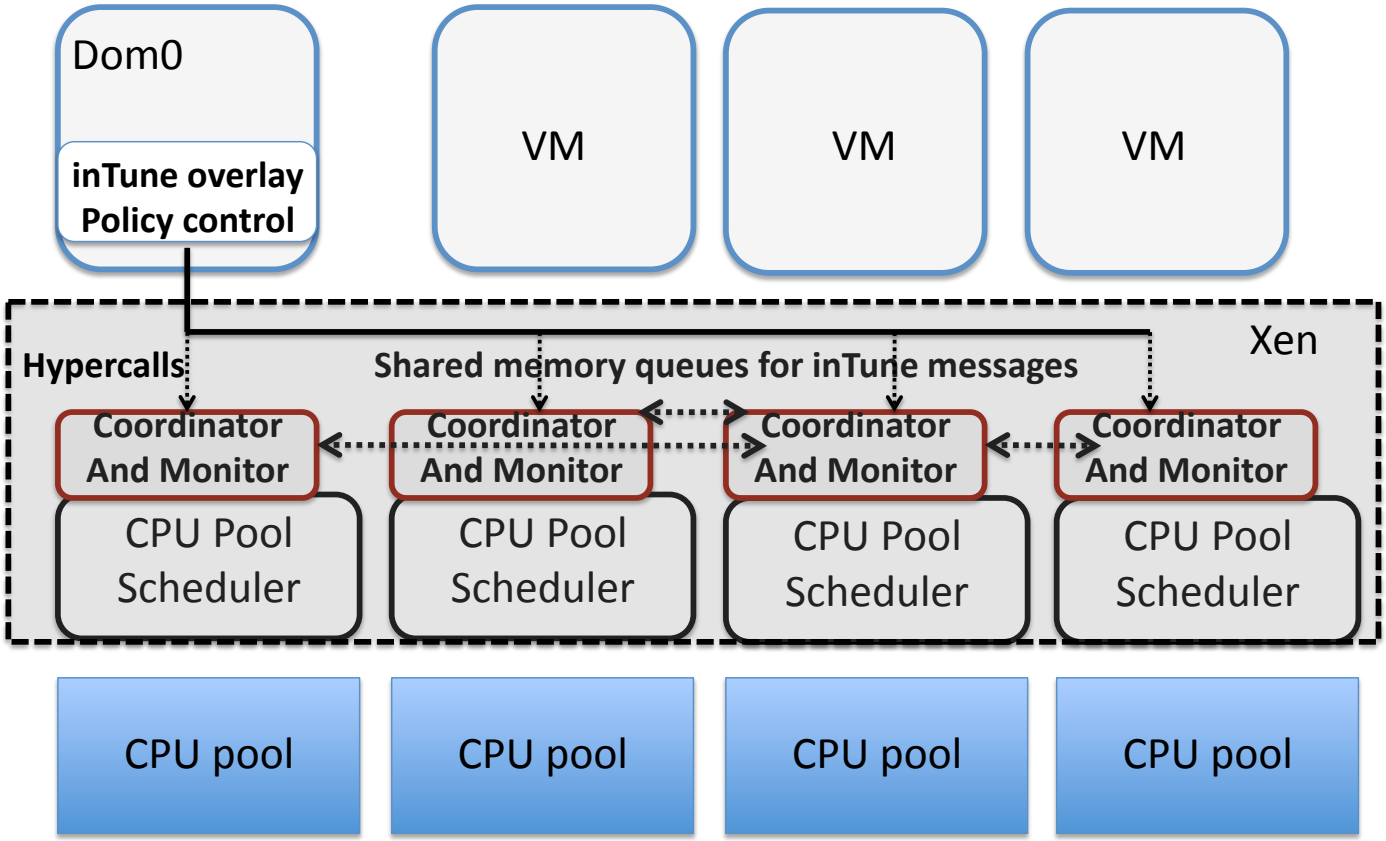


Backup slides

inTune Coordinator



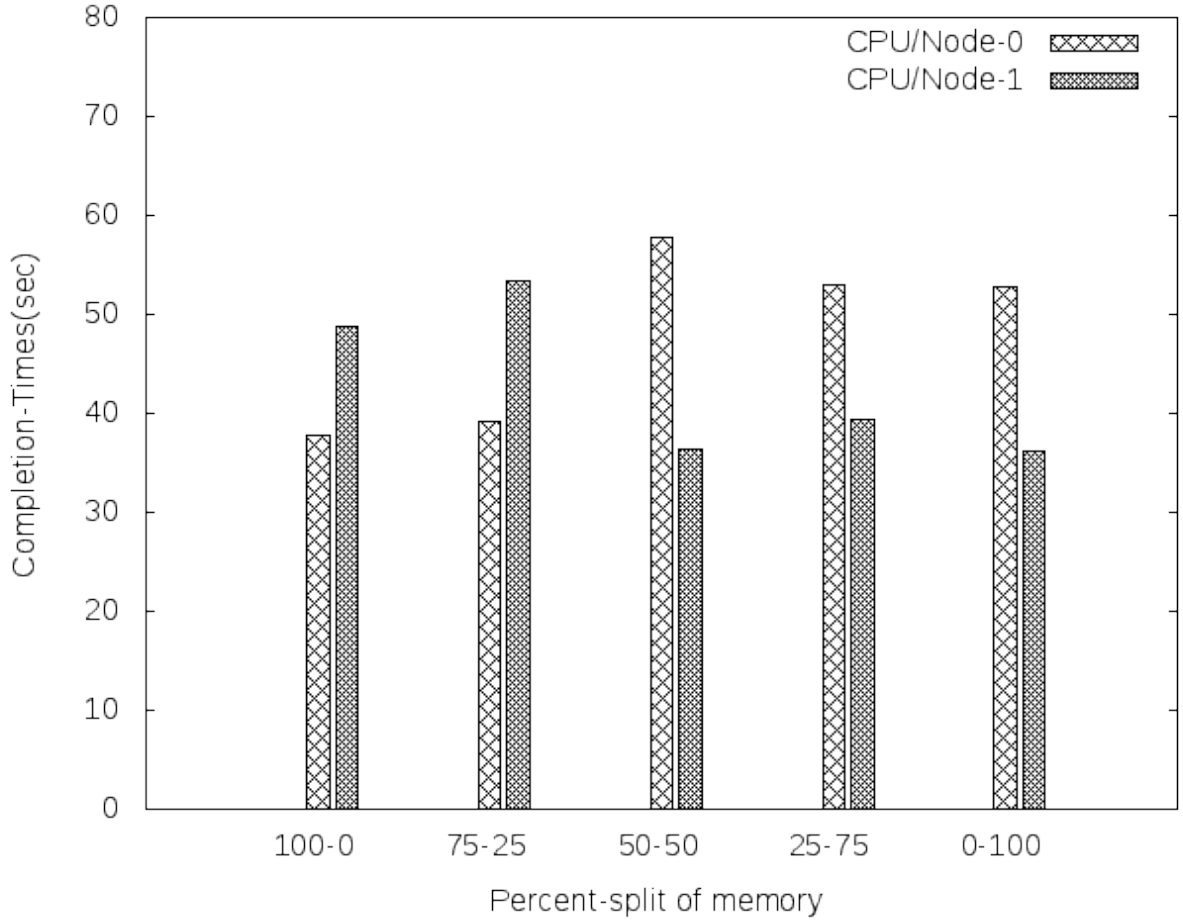
Implementation



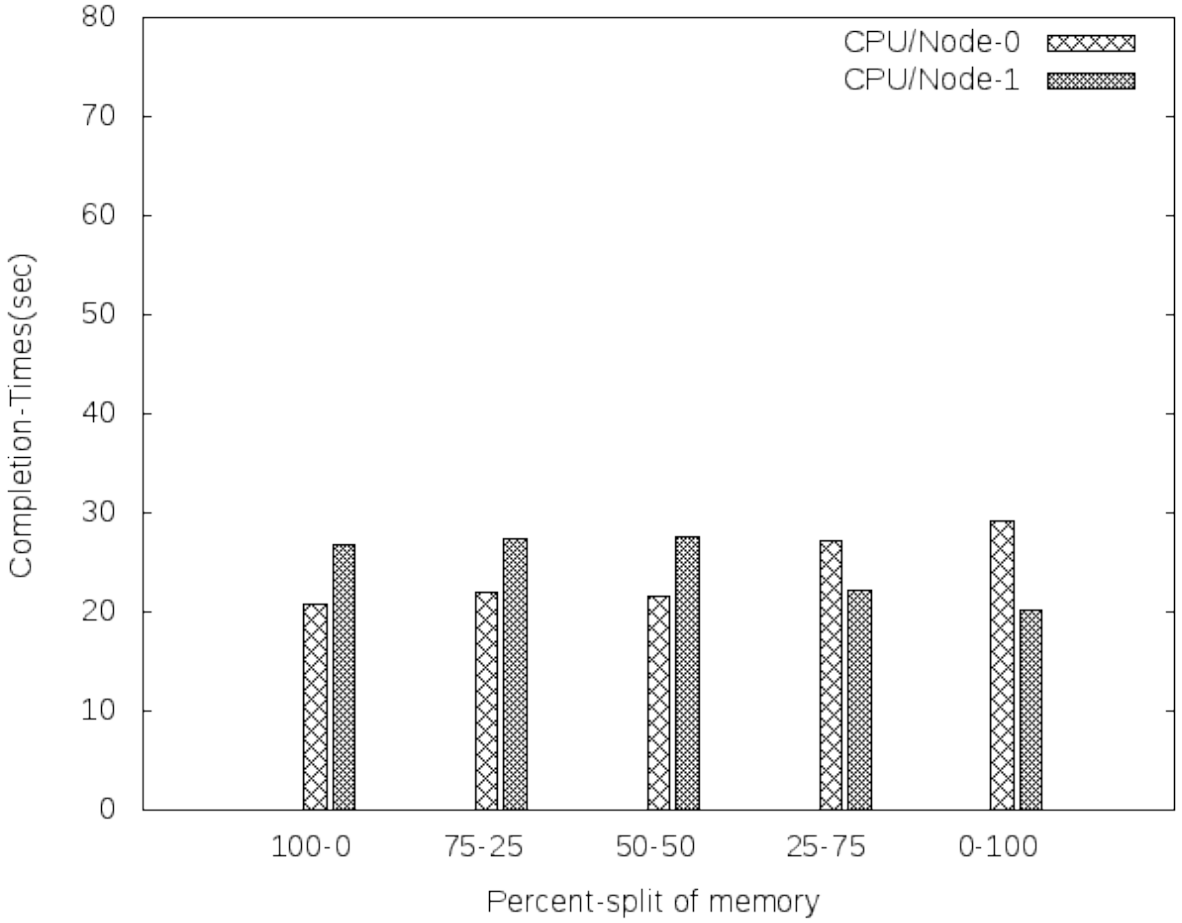
Cross Resource coordination: CPU & Memory

- Can coordinated islands be used to manage shared resource contention?
- Shared Resource Management
  - Contention in local memory island
    - Coordinate with local CPU island to tune CPU caps
    - Need to distribute load to other memory islands (E.g., When remote memory access costs < local memory access)
  - Related work in hardware/simulated systems, none in system software

Spec-swim

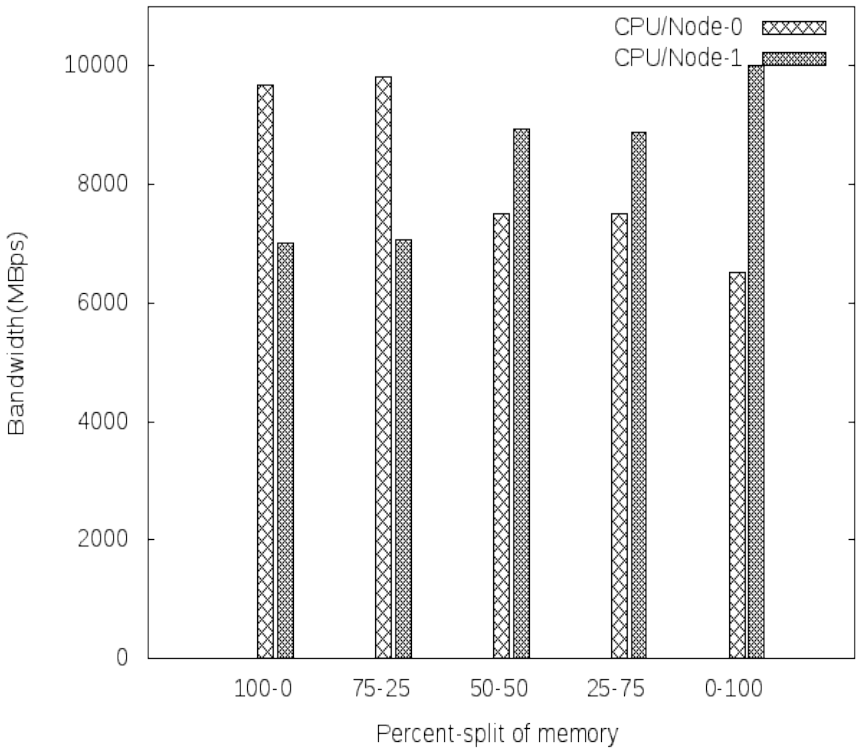

CPU/Node-0
CPU/Node-1
Completion-Times(sec)
0
10
20
30
40
50
60
70
80
100-0
75-25
50-50
25-75
0-100
Percent-split of memory

Spec-equake

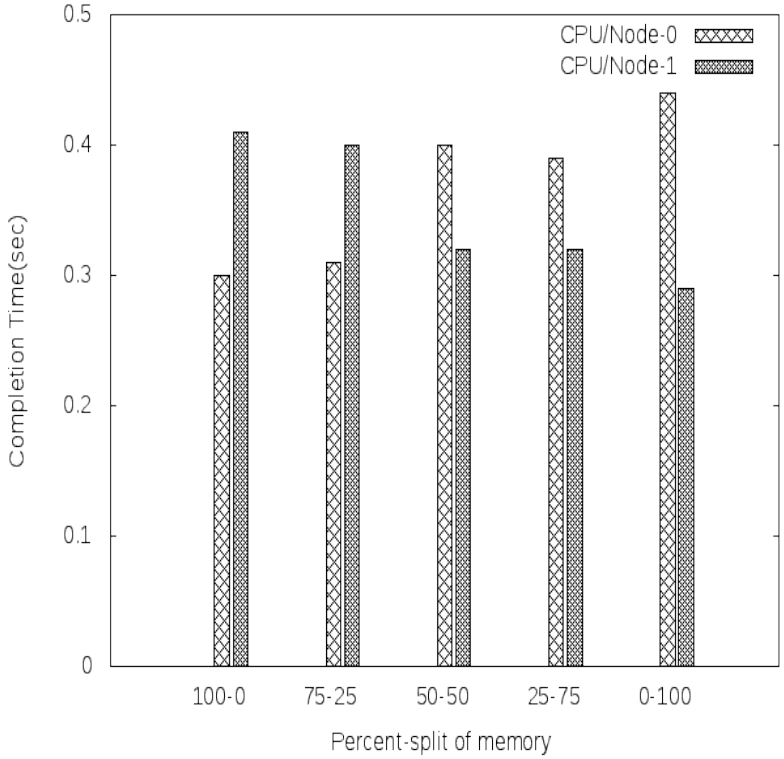

CPU/Node-0
CPU/Node-1
Completion-Times(sec)
0
10
20
30
40
50
60
70
80
100-0
75-25
50-50
25-75
0-100
Percent-split of memory

Stream-add

Bandwidth

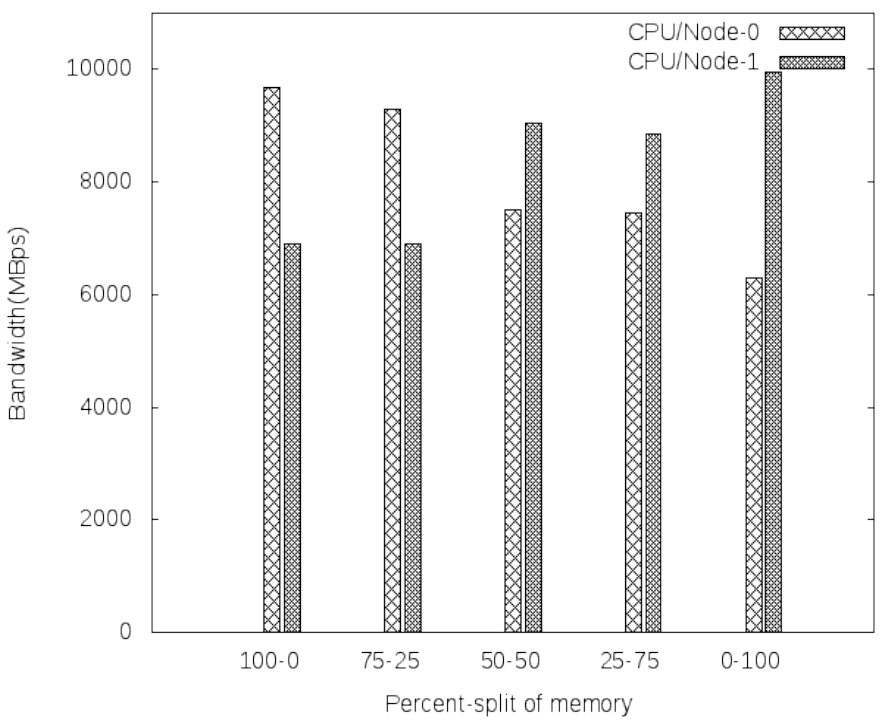


Latency



Stream - triad

Bandwidth



Latency