NETWORK-BASED COMPUTING LABORATORY

Reducing Network Contention with Mixed Workloads on Modern Multicore Clusters

Matthew Koop[1] Miao Luo D. K. Panda

matthew.koop@nasa.gov

{luom, panda}@cse.ohio-state.edu

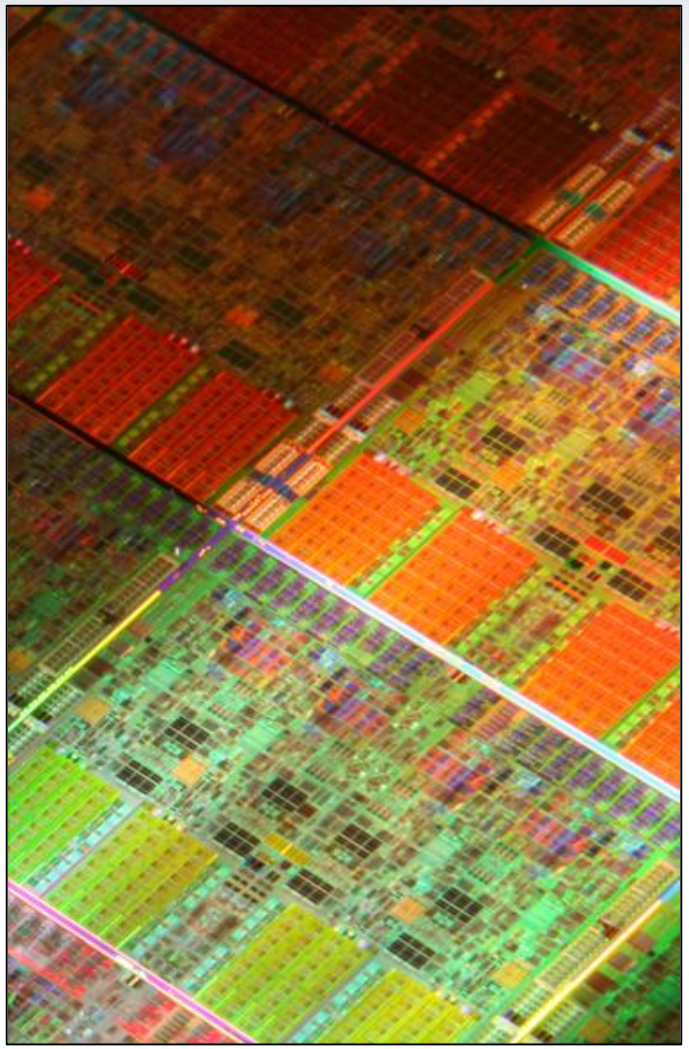
[1]NASA Center for Computational Sciences
Goddard Space Flight Center / CSC
Greenbelt, MD USA

Network-Based Computing Lab
The Ohio State University
Columbus, OH USA

OHIO STATE

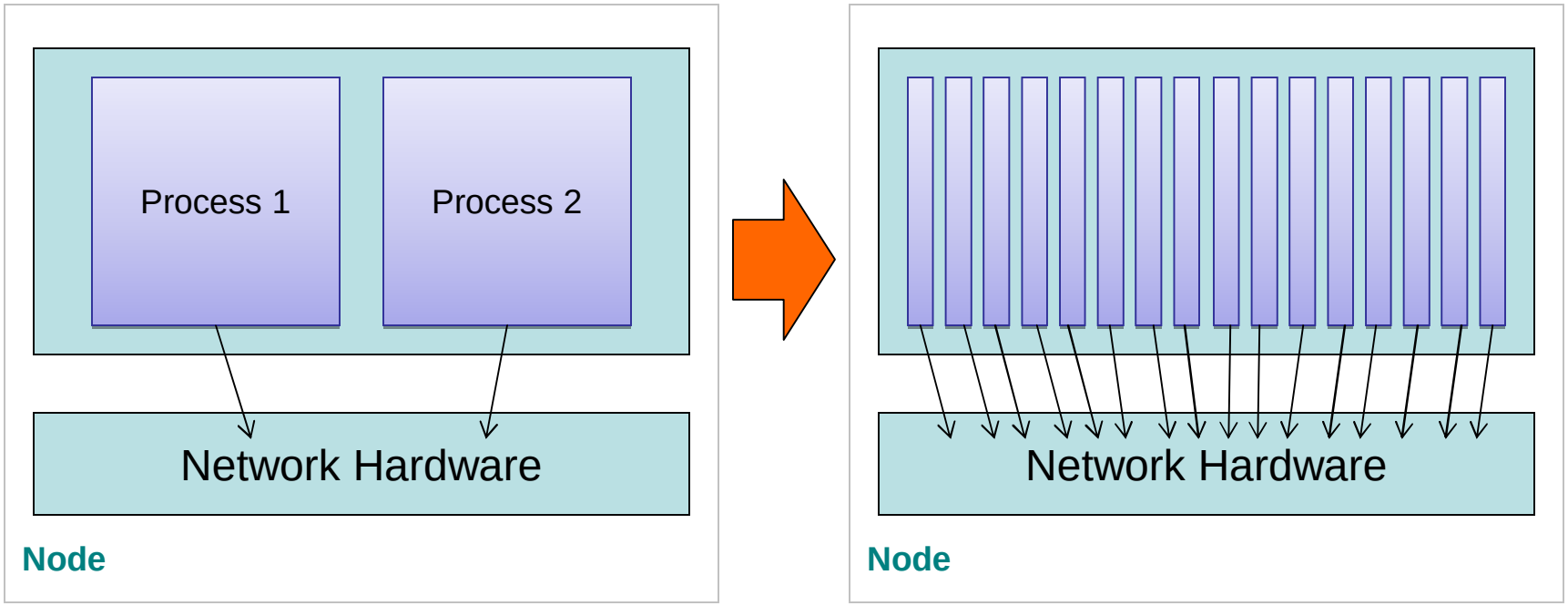
Introduction

- Multi-core systems are now extremely common
- There can be problems with *contention* in multicore systems unless other components are scaled appropriately
  - Memory Speed / Capacity
  - Network?
- This type of contention needs to be evaluated and addressed



Intel Core i7 (Nehalem) - courtesy Intel

Network Contention



- We are specifically interested in looking at network contention on a single node
- More cores, but usually only one network device per node

Introduction:

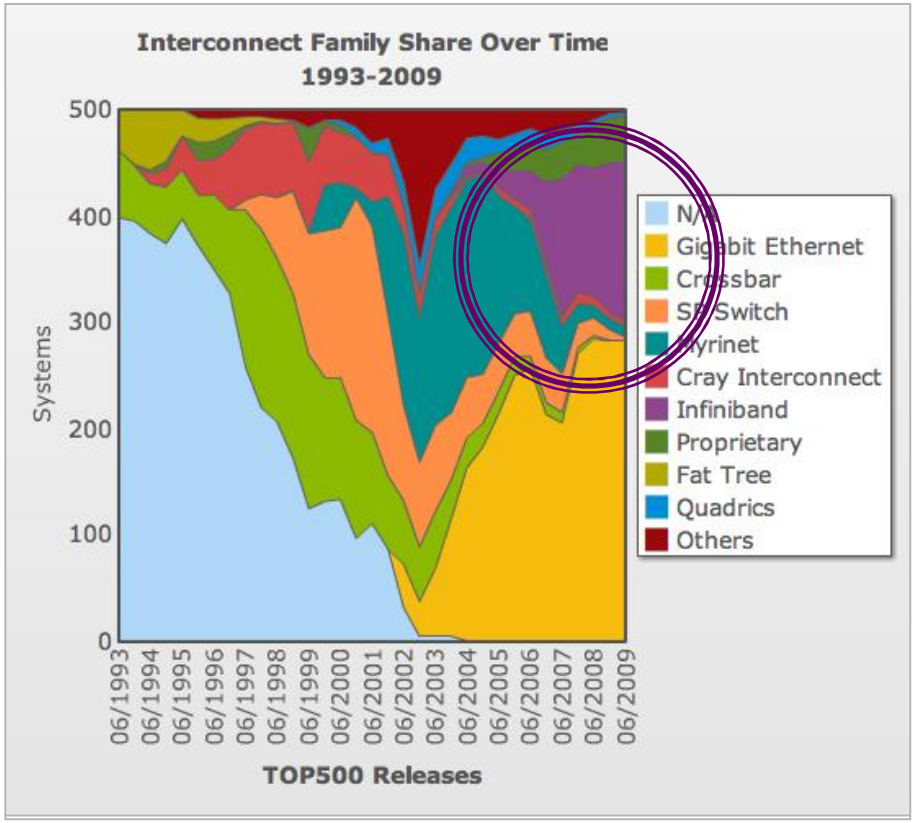
Network Contention

- How can we determine if there is network contention?
- MPI is the most popular programming model for scientific computing
  - How can we determine how much traffic is being contributed by each core?
  - Particularly more important for increasing numbers of cores per node

Introduction:

InfiniBand Overview

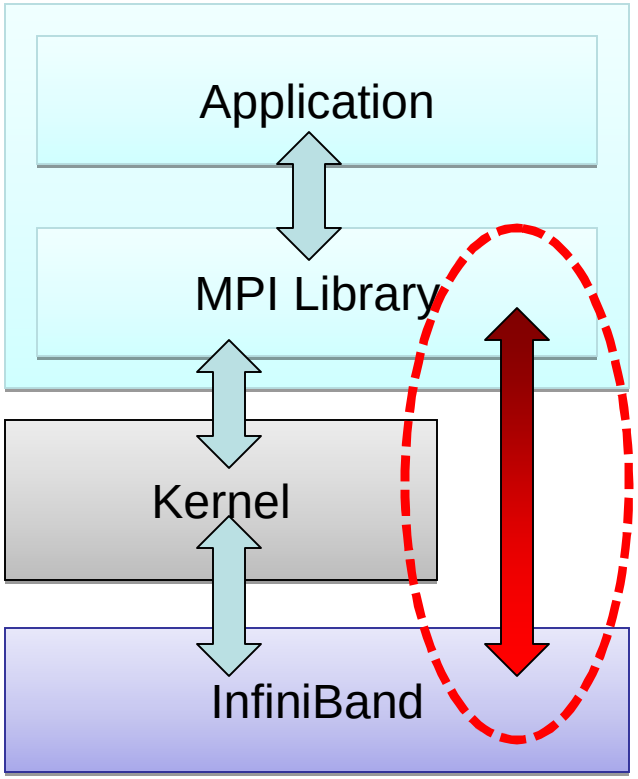
- InfiniBand is a popular high-speed interconnect
  - Minimum Latency: ~1-2us
  - Peak Bandwidth: ~1500-2500MB/s
- ~30% of *Top500* now uses InfiniBand as the primary interconnect
- We will use it as our testbed



Introduction:

Userspace Interconnects

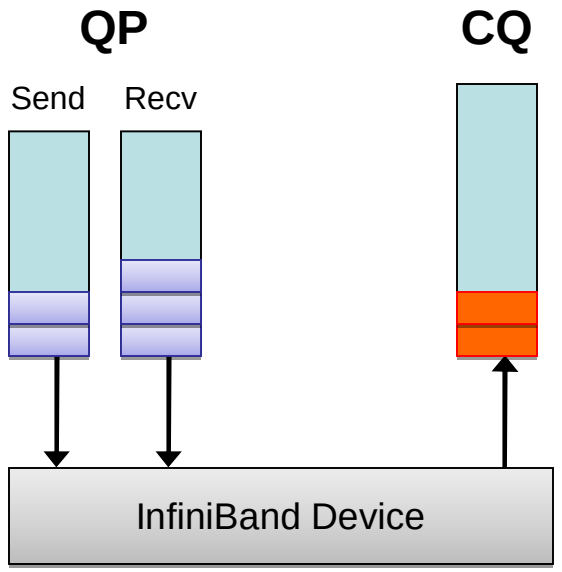
- If MPI is run over typical implementations of TCP/IP then the kernel knows the traffic
- InfiniBand is different….
  - The Host Channel Adapter (HCA) can be directly accessed from userspace applications
  - *Cannot have kernel capture statistics!*



Introduction: InfiniBand

InfiniBand Communication

- **Queue Pair (QP) Model**
  - Each QP consists of two queues:
    - Send Queue (SQ)
    - Receive Queue (RQ)
  - A QP must be linked to a Completion Queue (CQ)
    - Gives notification of operation completion from QPs



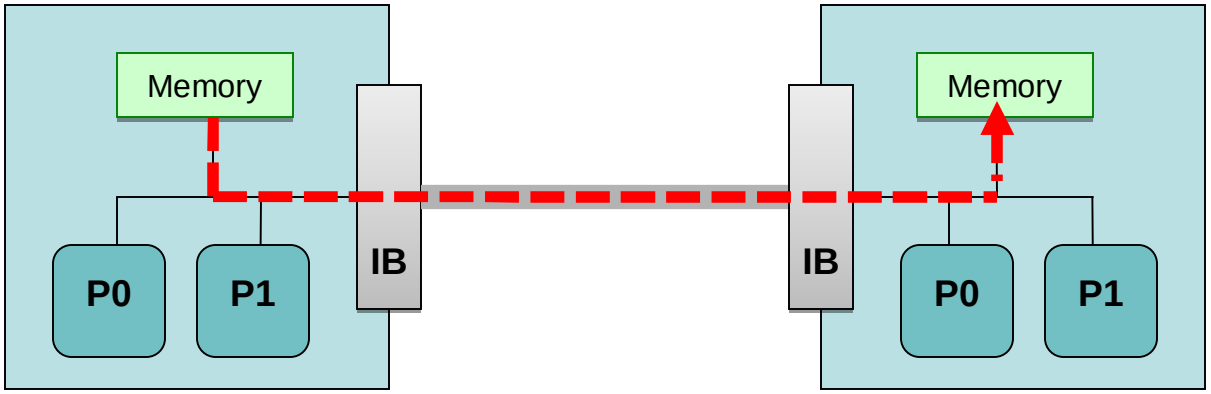
Introduction: InfiniBand

InfiniBand Communication (cont.)

- **Memory and Channel Semantics**
  - Memory: Remote Data Memory Access (RDMA)
    - No CPU interaction in copy (below)
  - Channel: Receive buffers are posted to the QP Receive Queue (or SRQ) and consumed in order

---

RDMA Transfer with no CPU interaction



Problem Statement

- *Is the network a bottleneck for systems of increasing numbers of cores? How can we see the traffic?*

- *If so, what can be done to reduce this contention and increase overall cluster throughput?*

Presentation Outline

- Introduction
- Problem Statement
- **Measuring Network Access**
- Mixed Workloads & Evaluation
- Conclusions and Future Work

Measuring Network Traffic

- We cannot measure network traffic per process directly on the host without additional hardware assistance
- Instead, can we instrument or time MPI calls to give network access information?

Measuring Network Traffic:

MPI vs. Network

- MPI has functionality through PMPI interface to allow tools to intercept MPI calls and get timing information
  - Used in many MPI profiling tools – Vampir, mpiP, etc.
- Message completion in MPI in general does not imply network access has started or completed at that time
  - Only means that buffer is available
  - e.g. `MPI_Send` finishing only means that the send buffer can now be reused

Measuring Network Traffic:

MPI vs. Network (contd.)

MPI Time:

Network Usage Time:

Timing for MPI and the network need not match

Measuring Network Traffic:

Track Directly inside MPI

- The MPI library can know when message sends are initiated
- Unfortunately due to asynchronous nature of InfiniBand communication we must *estimate* duration of network access
- Use performance characteristics based on number of concurrent accesses by number of processes on the same node
  - e.g. If there are ‘n’ processes all sending large messages each will take roughly ‘n’ times longer to finish than if it were only one process sending.
  - Use specially designed benchmarks to determine characteristics

Presentation Outline

- Introduction
- Problem Statement
- Measuring Network Access
- **Mixed Workloads & Evaluation**
- Conclusions and Future Work

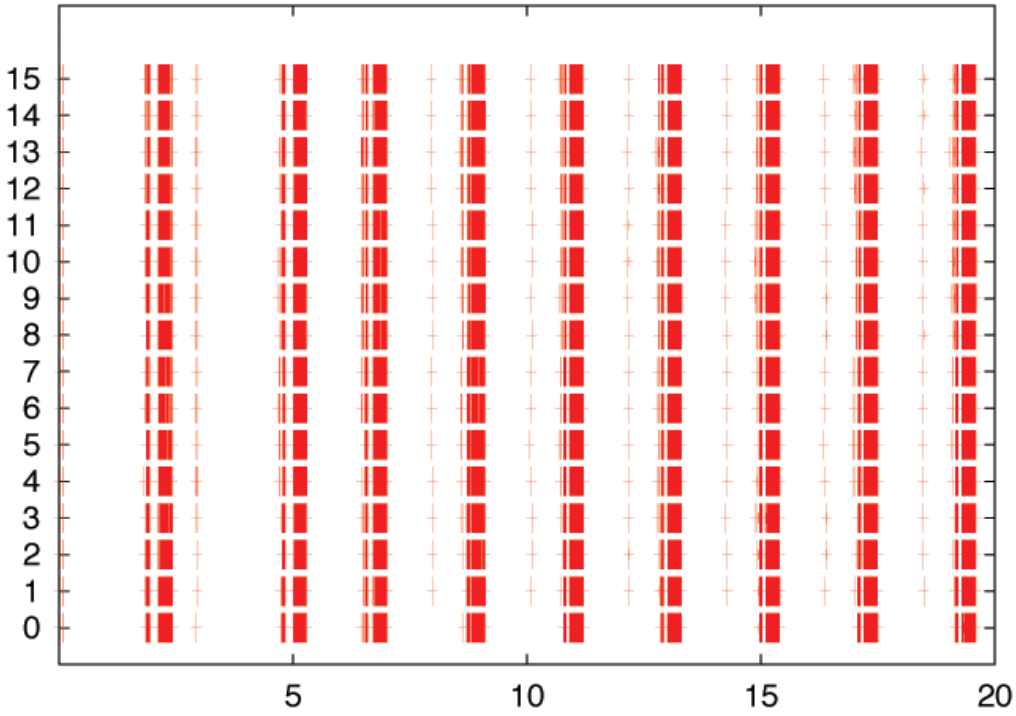
Mixed Workloads & Evaluation:

Experimental Setup

- Cluster Configuration:
  - 128-core InfiniBand Cluster
  - Quad Socket, Quad-Core Opteron 2GHz
  - Mellanox DDR ConnectX HCA
  - OpenFabrics Enterprise Edition (OFED) 1.3
- Implementation
  - Based off of MVAPICH2 1.2 (http://mvapich.cse.ohio-state.edu)
  - MPI library used by over 960 organizations worldwide
  - Offline analysis of performance data stored per node
- Benchmarks
  - We use the NAS Parallel Benchmarks
  - Fluid dynamics kernels / mini apps

Mixed Workloads & Evaluation:

Network Profile (NAS.FT)



- NAS FT (and many other applications) have a very structured pattern where network communication is done concurrently

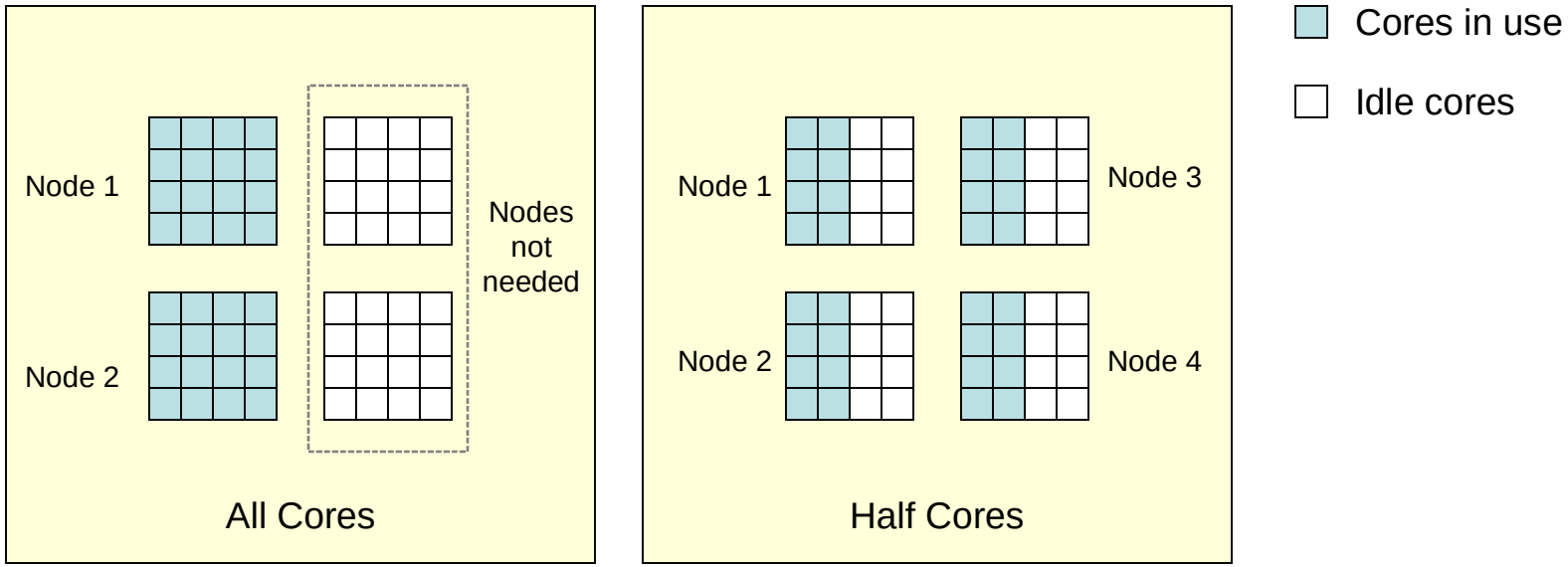
Mixed Workloads & Evaluation

Network Concurrency

- Concurrent network access is very inefficient
- The network is idle most of the time and then under significant contention
- *What if we could reduce this contention?*
  - *How much would the speedup be?*
  - What if half the number of cores were using the same network?

Mixed Workloads & Evaluation

All vs. Half Cores



- By running on half the cores we can see the benefit of having ½ as many cores per network adapter
- Using 'Half Cores' means twice as many nodes are needed

Mixed Workloads & Evaluation

Network Concurrency

§ We ran each NAS benchmark with '*All Cores*' and '*Half Cores*' configurations

	BT	CG	EP	FT	IS	LU	MG	SP
All Cores	198.22	34.82	28.53	44.19	3.10	180.46	156.14	196.29
Half Cores (on double number of nodes)	196.71	26.17	29.07	38.89	2.29	179.86	14.78	188.84
Half/All	99%	75.2%	99%	88.0%	73.9%	99.6%	97.6%	96.2%

Runtimes in seconds for different benchmarks

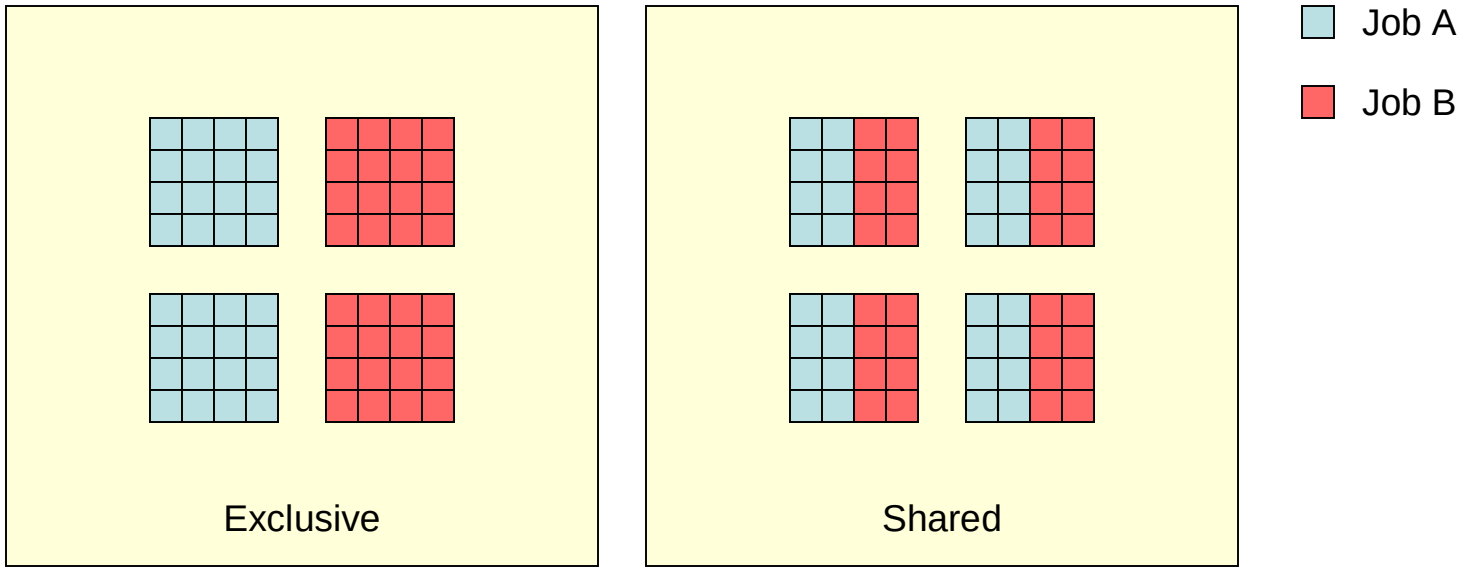
Mixed Workloads & Evaluation

Mixed Workloads

- Others have previously suggested running multiple jobs on the same node since they have different requirements
  - Generally this has been done in the context of file I/O access, memory access, or cache usage
- *We propose adding the network as a key component of deciding job co-location*
  - NUMA already significantly segments nodes
  - We believe network is the main shared resource for contention

Mixed Workloads & Evaluation

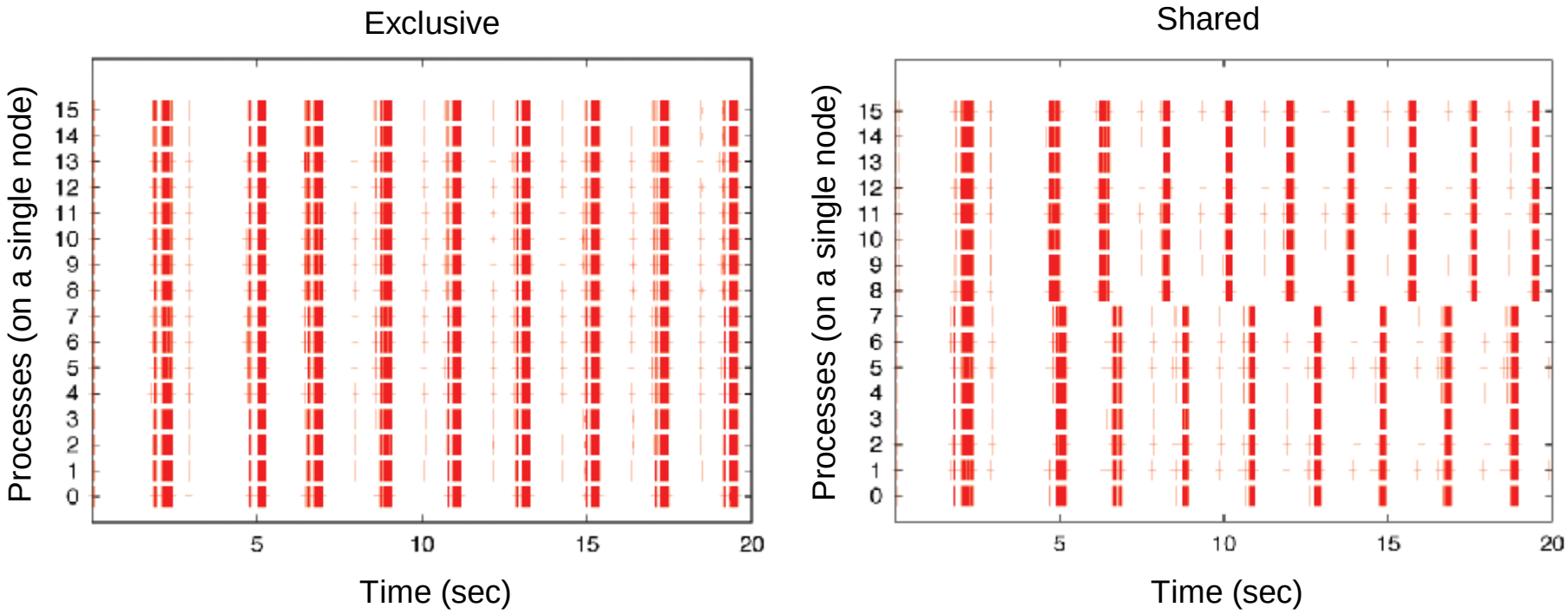
Exclusive vs. Shared



- We evaluate each of the NAS Parallel Benchmarks with each other to determine what patterns can be scheduled together and achieve higher throughput

Mixed Workloads & Evaluation

Exclusive vs. Shared



- Running FT together with itself shows how communication becomes offset
- Overall runtime is **89%** of *Exclusive* run

Mixed Workloads & Evaluation

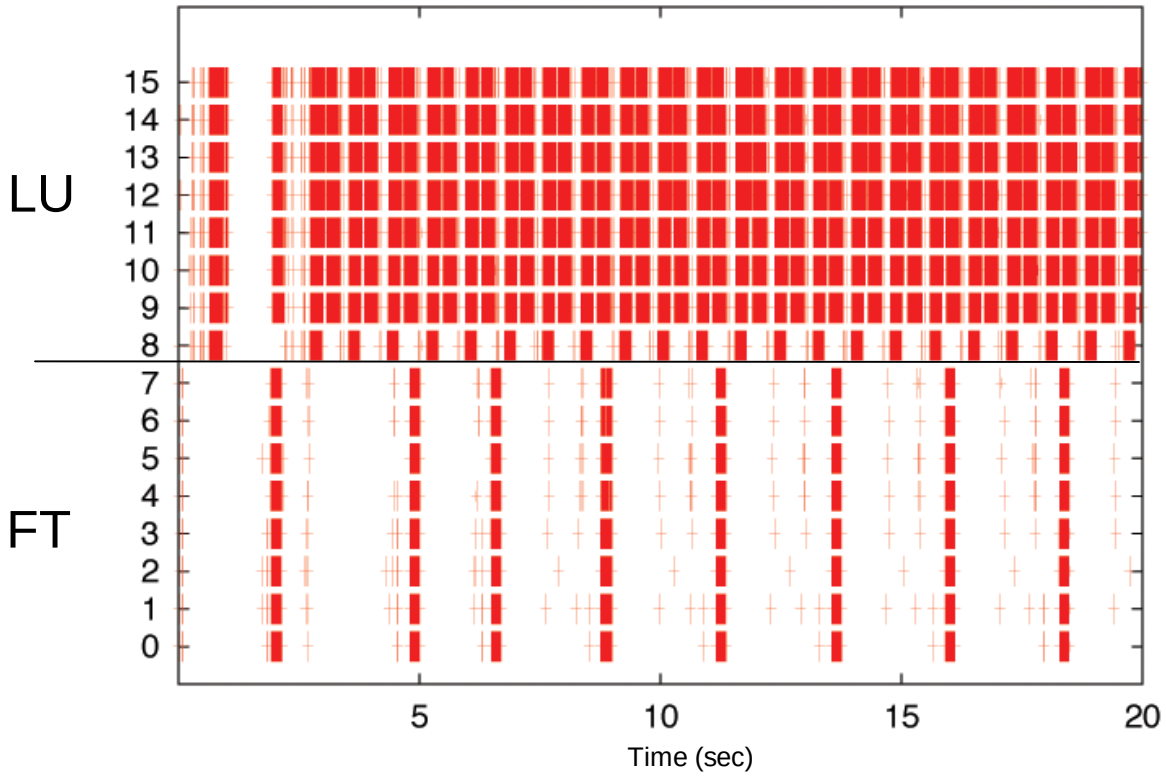
Shared Mode Evaluation

Measured Application

Background Application	BT	CG	EP	FT	IS	LU	MG	SP
BT	99.1%	77.3%	100.2%	91.9%	81.6%	99.7%	98.4%	96.6%
CG	100.5%	101.8%	100.5%	96.0%	90.2%	100.9%	100.8%	102.0%
EP	98.8%	75.2%	99.6%	93.8%	80.1%	100.1%	97.9%	97.2%
FT	99.4%	84.3%	99.9%	89.6%	87.6%	100.5%	99.5%	98.9%
IS	100.2%	79.0%	99.1%	91.0%	84.4%	99.6%	98.8%	96.2%
LU	99.2%	76.2%	100.0%	88.0%	80.7%	100.4%	98.9%	97.0%
MG	99.0%	77.3%	100.4%	89.4%	73.9%	99.6%	98.1%	100.5%
SP	99.6%	79.2%	100.4%	93.2%	86.7%	100.3%	97.6%	99.5%
Max	99%	75.2%	99%	88.0%	73.9%	99.6%	97.6%	96.2%

Mixed Workloads & Evaluation

Shared Mode: FT & LU

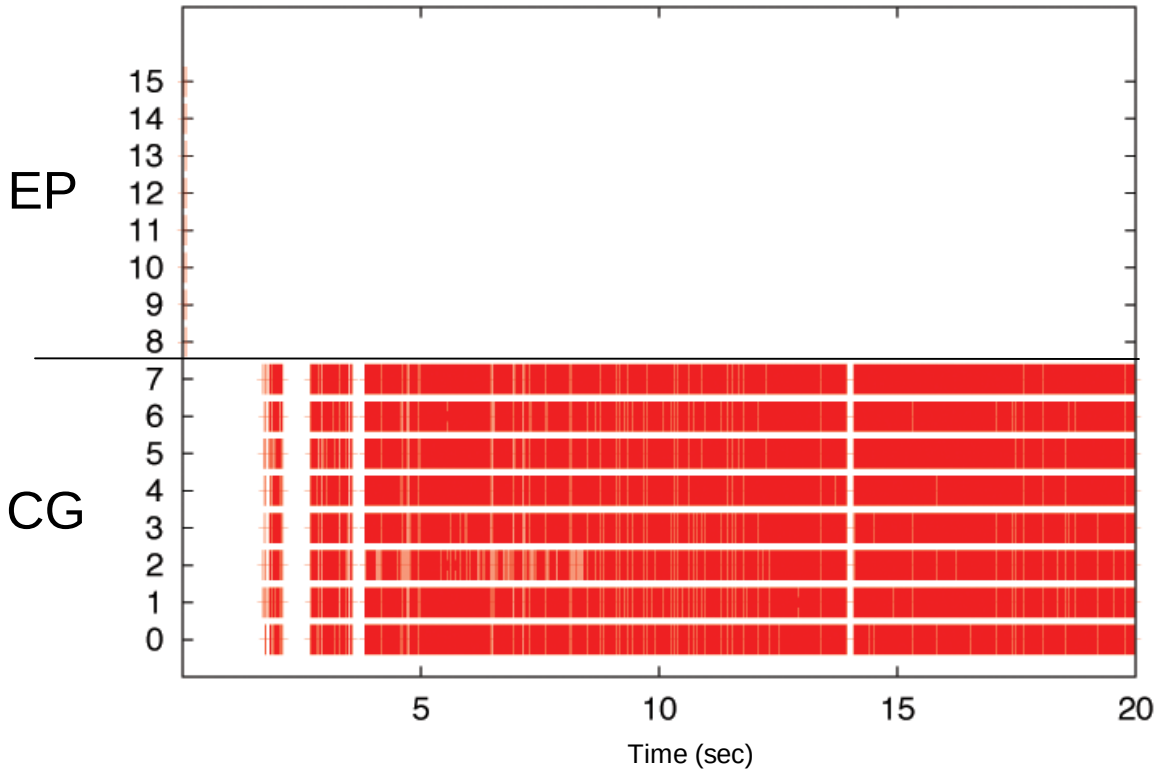


	FT	LU
FT	**89.6**	100.5
LU	**88.0**	100.4

- FT is extremely bandwidth sensitive
- Less communication-intensive applications, such as LU don't improve, but offer a perfect companion
- ***Why does CG/CG not work if FT/FT shows improvement?***

Mixed Workloads & Evaluation

Shared Mode: CG & EP



	CG	EP
CG	101.8	100.5
EP	**75.2**	99.6

- CG has a near-constant communication profile
- Few gaps – less possibility for improvement in other applications

Conclusion and Future Work

- Network contention is an increasing concern as additional cores are added
  - Many applications exhibit very synchronized communication
- Developed a method to profile network access for MPI applications on InfiniBand
- Showed 20% improvement when using a shared method of scheduling
- *Future*:
  - Better determination of symbiotic job scheduling / automated
  - Studying the effect of QDR on this study

http://mvapich.cse.ohio-state.edu

Questions?

matthew.koop@nasa.gov

{luom, panda}@cse.ohio-state.edu