Improving Application Performance and Predictability using Multiple Virtual Lanes in Modern Multi-Core InfiniBand Clusters

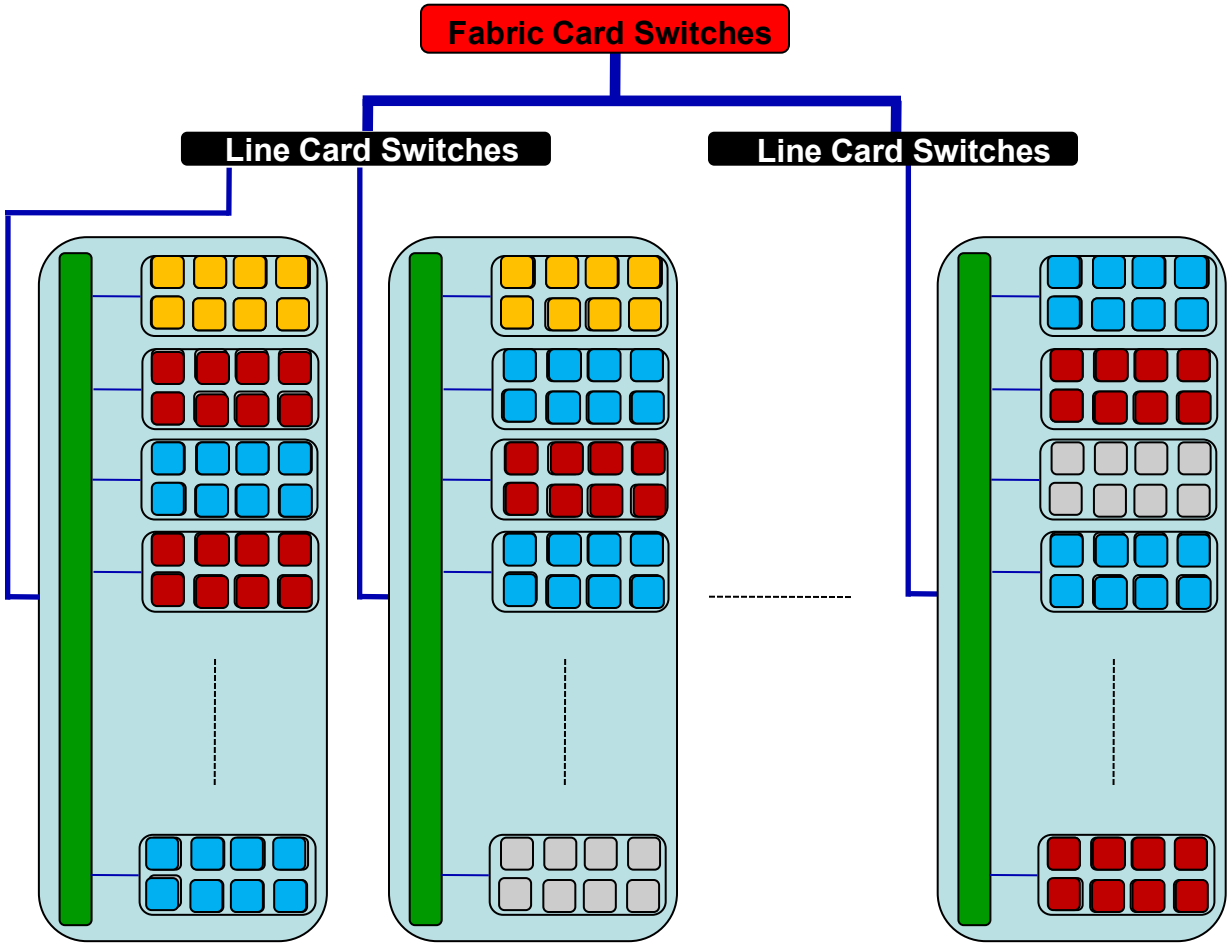
Hari Subramoni, Ping Lai, Sayantan Sur and Dhabhaleswar. K. Panda

Department of Computer Science & Engineering
The Ohio State University

Outline

- Introduction & Motivation
- Problem Statement
- Design
- Performance Evaluation and Results
- Conclusions and Future Work

Introduction & Motivation



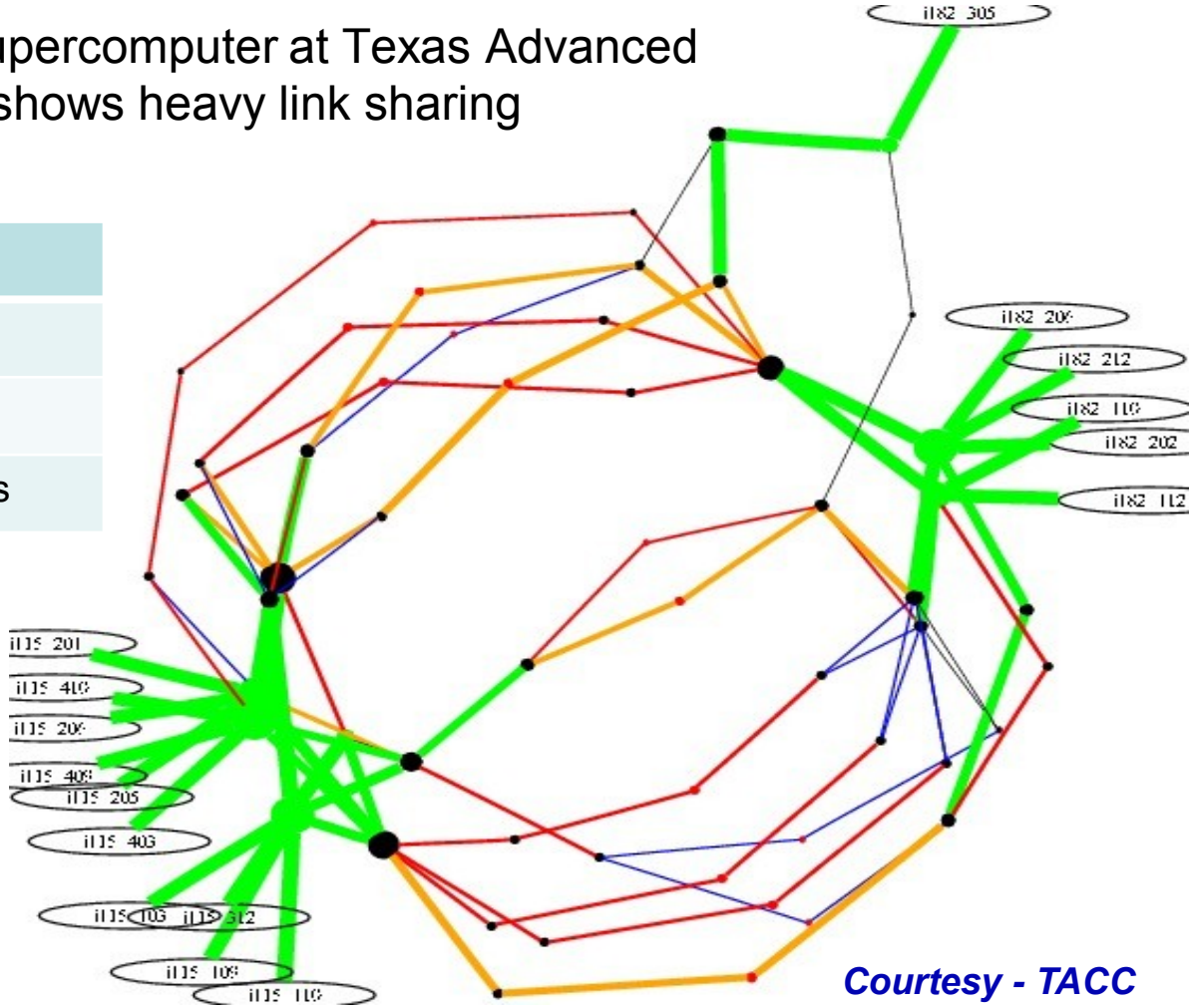
- Supercomputing clusters growing in size and scale
- MPI – predominant programming model for HPC
- High performance interconnects like InfiniBand increased network capacity
- Compute capacity outstrips network capacity with advent of multi/many core processors
- Gets aggravated as jobs get assigned to random nodes and share links

Analysis of Traffic Pattern in a Supercomputer

- Traffic flow in the Ranger supercomputer at Texas Advanced Computing Center (TACC) shows heavy link sharing
  - *http://www.tacc.utexas.edu*

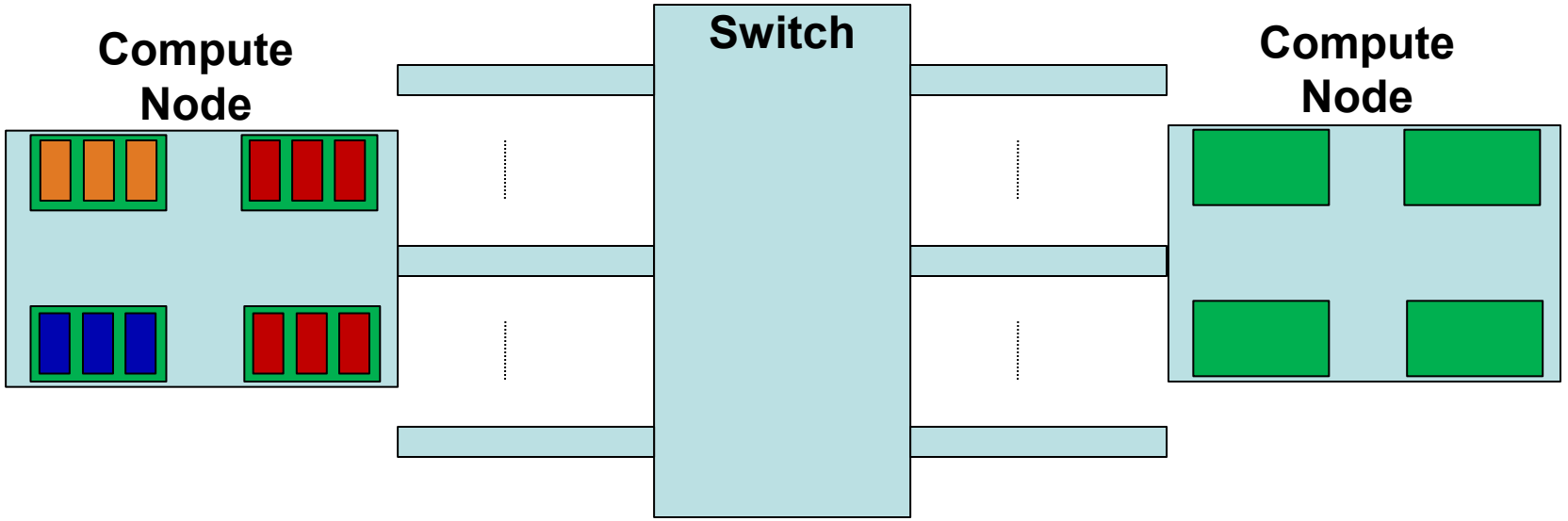
Color of Dot	Description
Green	Network Elements
Black	Line Card Switches
Red	Fabric Card Switches

Color	Number of Streams
Black	1
Blue	2
Red	3 – 4
Orange	5 - 8
Green	> 8



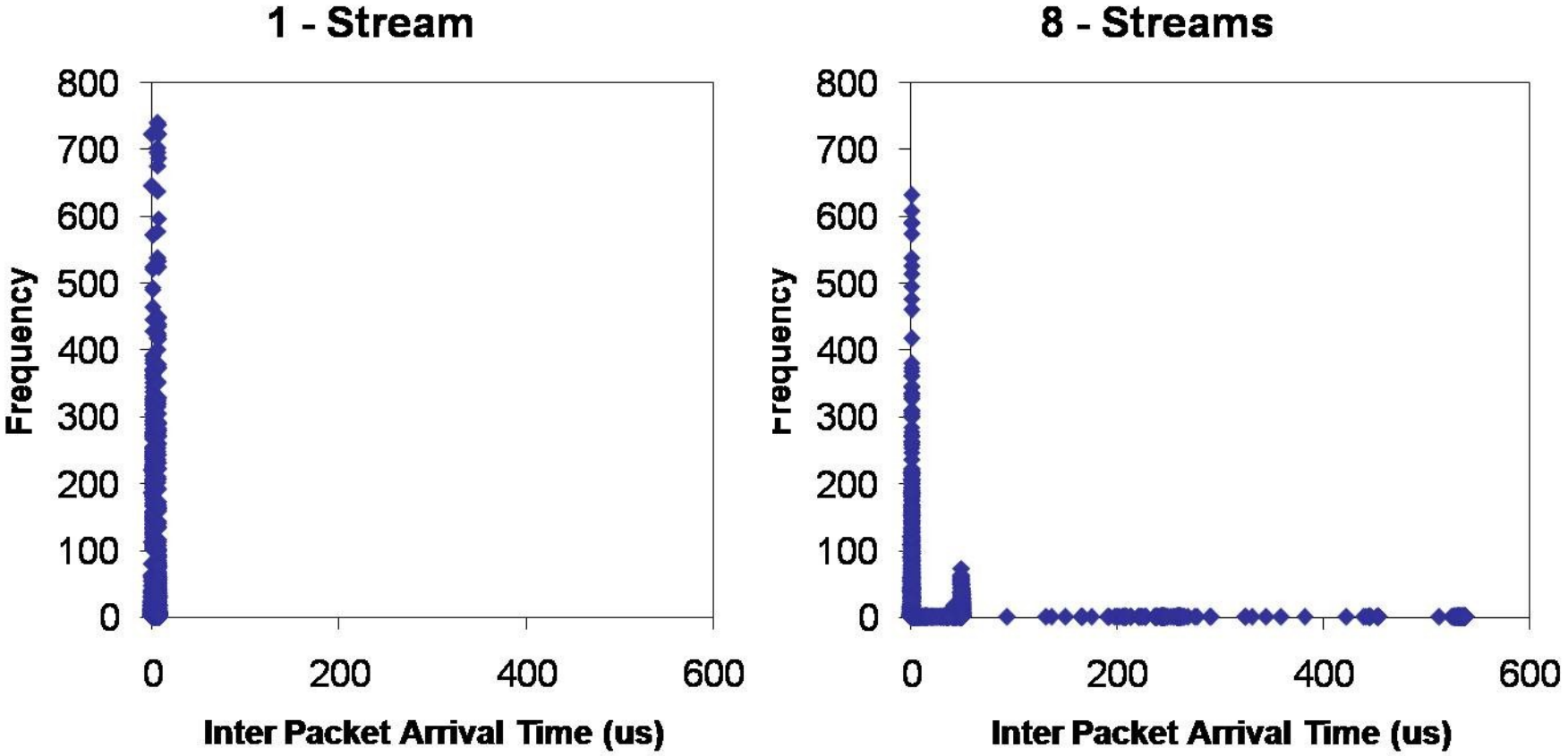
Courtesy - TACC

Possible Issue with Link Sharing



- Few communicating peers – No Problem
- Packets get backed up as number of communicating peers increases
- Results in delayed arrival of packet at destination

Frequency Distribution of Inter Arrival Times



- Packet size – 2 KB (results same for 1 KB to 16 KB)
- Arrival time is directly proportional to the load on the links

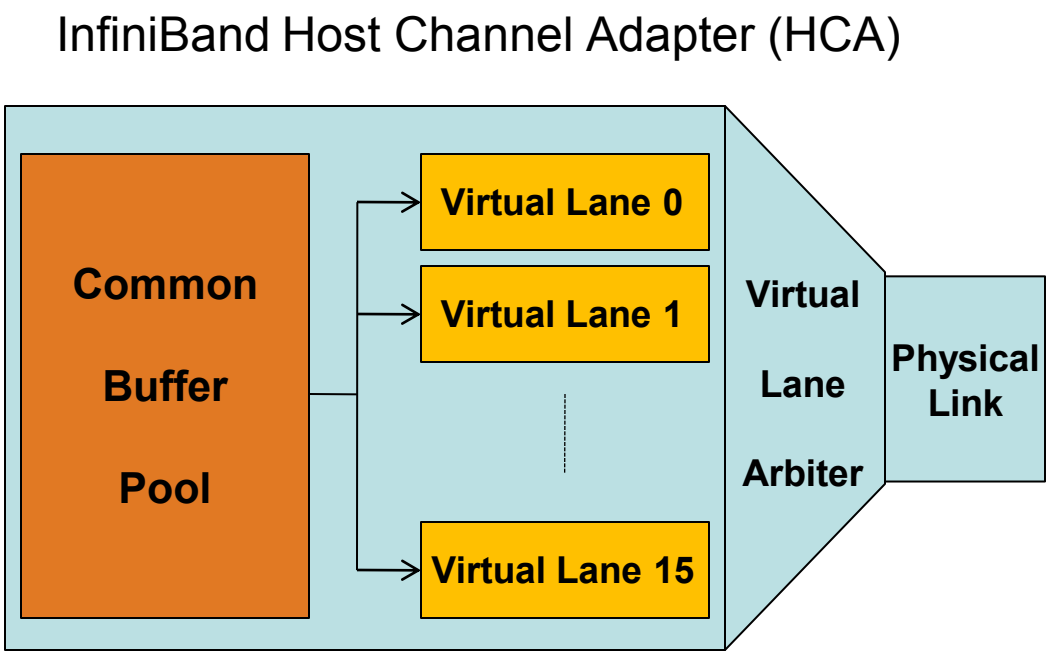
Introduction & Motivation (Cont)

Can modern networks like InfiniBand alleviate this?

InfiniBand Architecture

- An industry standard for low latency, high bandwidth System Area Networks
- Multiple features
  - Two communication types
    - Channel Semantics
    - Memory Semantics (RDMA mechanism)
  - Queue Pair (QP) based communication
  - Quality of Service (QoS) support
  - Multiple Virtual Lanes (VL)
  - QPs associated to VLs by means of pre-specified Service Levels
- Multiple communication speeds available for Host Channel Adapters (HCA) – 10 Gbps (SDR) / 20 Gbps (DDR) / 40 Gbps (QDR)

InfiniBand Network Buffer Architecture



- Buffers in most IB HCAs and switches grouped into two
  - Common Buffer Pool and,
  - Private VL buffers
- Most current generation MPIs only use one VL
- Inefficient use of available network resources
- *Why not use more VLs?*
- Possible con
  - Would it take more time to poll all the VLs

Outline

- Introduction & Motivation
- Problem Statement
- Design
- Performance Evaluation and Results
- Conclusions and Future Work

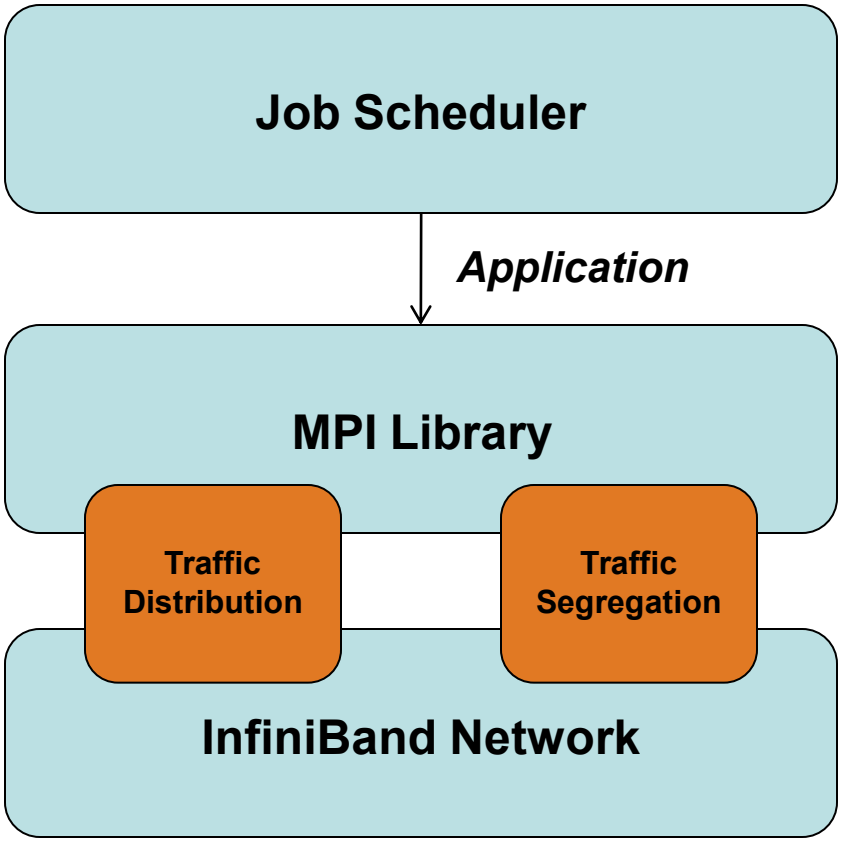
Problem Statement

- Can multiple virtual lanes be used to improve performance of HPC applications
- How can we integrate this design into an MPI library so that end applications will be benefited

Outline

- Introduction & Motivation
- Problem Statement
- Design
- Performance Evaluation and Results
- Conclusions and Future Work

Proposed Framework and Goals



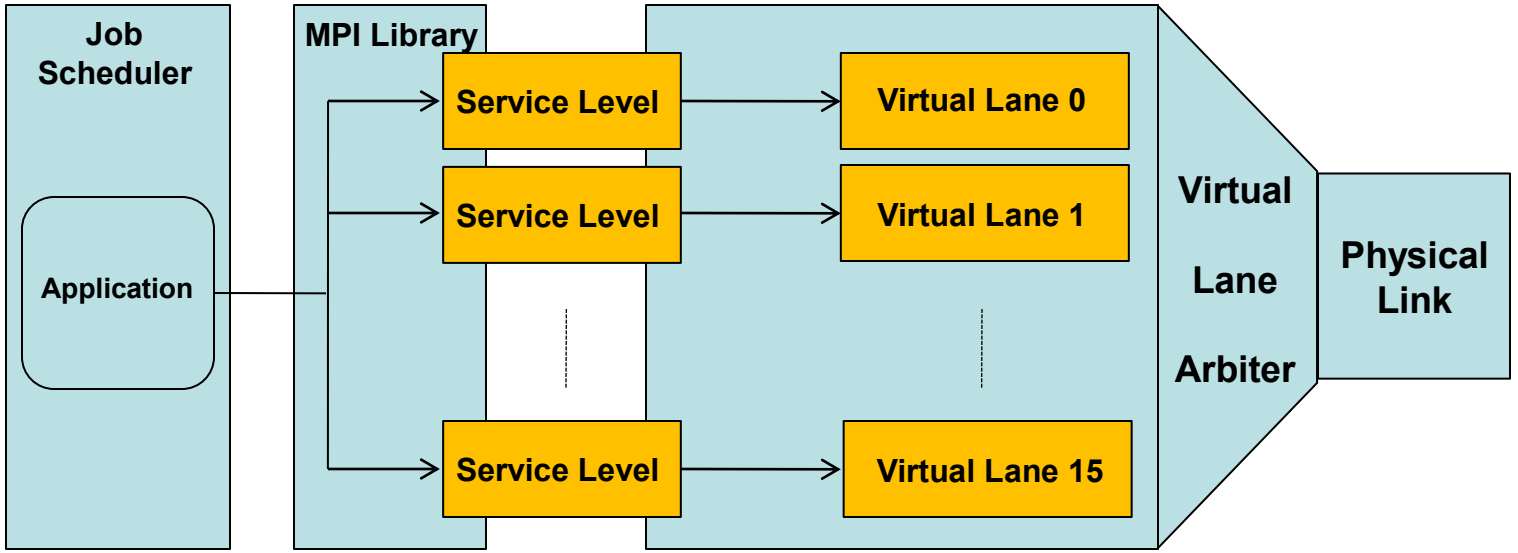
- No change to application
- Re-design MPI library to use multiple VLs
- Need new methods to take advantage of multiple VLs
  - Traffic Distribution
    - Load balance traffic across multiple VLs
  - Traffic Segregation
    - Ensure one kind of traffic does not disturb other
    - Distinguish between
      - Low & High priority traffic
      - Small & Large messages

Proposed Design

- Re-design MPI library to use multiple VLs
  - Multiple Virtual Lanes configured with different characteristics
    - Transmit less packets at high priority
    - Transmit more packets at lower priority etc
  - Multiple Service Levels (SL) defined to match VLs
  - Queue Pairs (QPs) assigned proper SLs at QP creation time

- Multiple ways to assign Service Levels to applications
  - Assign SLs with similar characteristics in a round robin fashion
    - *Traffic Distribution*
  - Assign SLs with desired characteristic based on type of application
    - *Traffic Segregation*
  - *Other designs being explored*

Proposed Design (Cont)



Outline

- Introduction & Motivation
- Problem Statement
- Design
- Performance Evaluation & Results
- Conclusions and Future Work

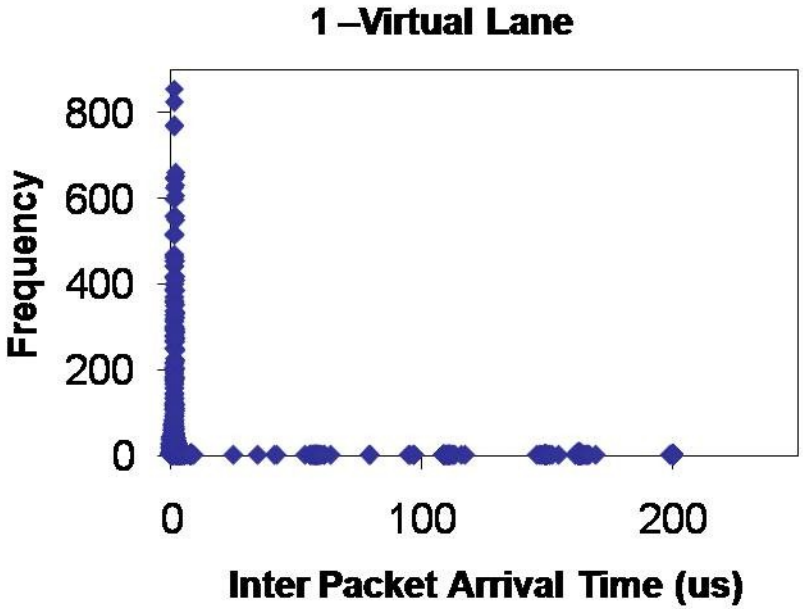
Experimental Testbed

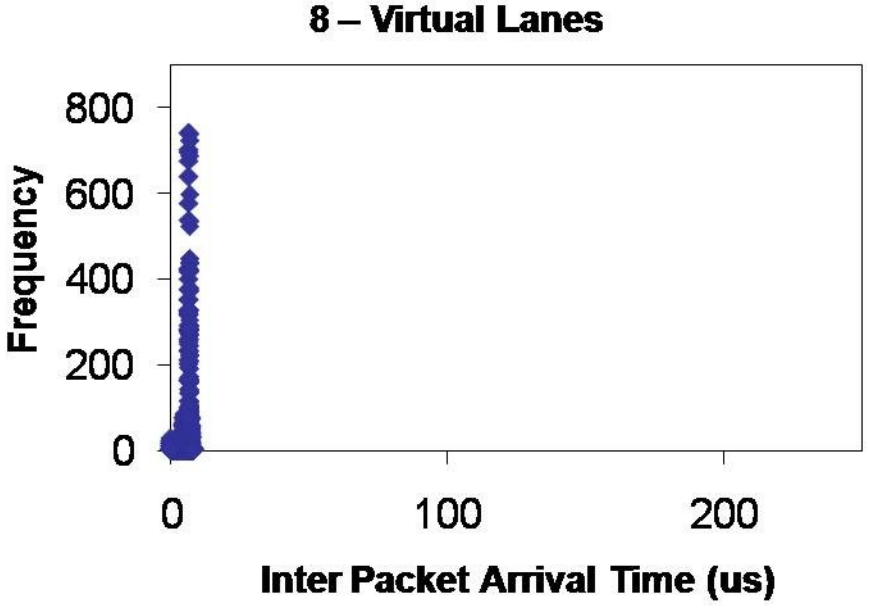
- Compute platforms
  - Intel Nehalem
    - Intel Xeon E5530 Dual quad-core processors operating at 2.40 GHz
    - 12GB RAM, 8MB cache
    - PCIe 2.0 interface
- Network Equipments
  - MT26428 QDR ConnectX HCAs
  - 36-port Mellanox QDR switch used to connect all the nodes
- Red Hat Enterprise Linux Server release 5.3 (Tikanga)
- OFED-1.4.2
- OpenSM version 3.1.6
- Benchmarks
  - Modified version of OFED perftest for verbs level tests
  - MPIBench collective benchmark
  - CPMD used for application level evaluation

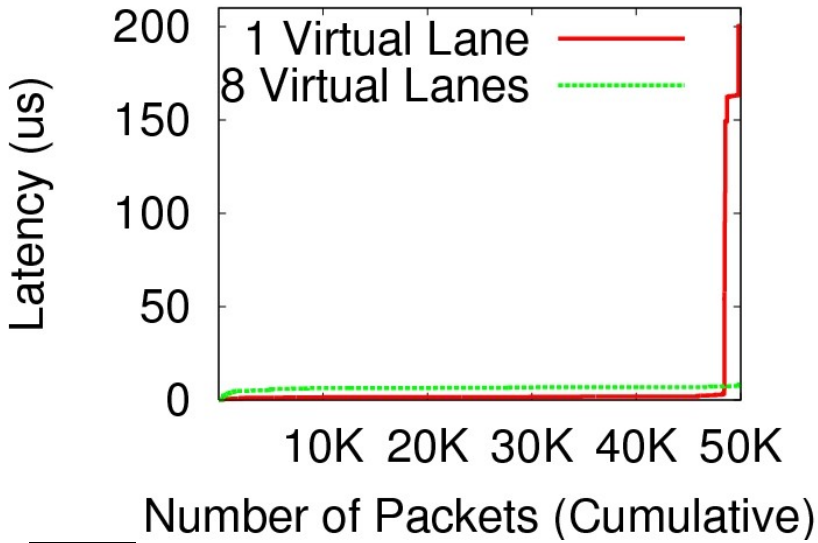
MVAPICH / MVAPICH2 Software

- High Performance MPI Library for IB and 10GE
  - MVAPICH (MPI-1) and MVAPICH2 (MPI-2)
  - Used by more than 1255 organizations in 59 countries
  - More than 44,500 downloads from OSU site directly
  - Empowering many TOP500 clusters
    - 11th ranked 62,976-core cluster (Ranger) at TACC
  - Available with software stacks of many IB, 10GE and server vendors including Open Fabrics Enterprise Distribution (OFED)
  - Also supports uDAPL device to work with any network supporting uDAPL
  - http://mvapich.cse.ohio-state.edu/

Verbs Level Performance

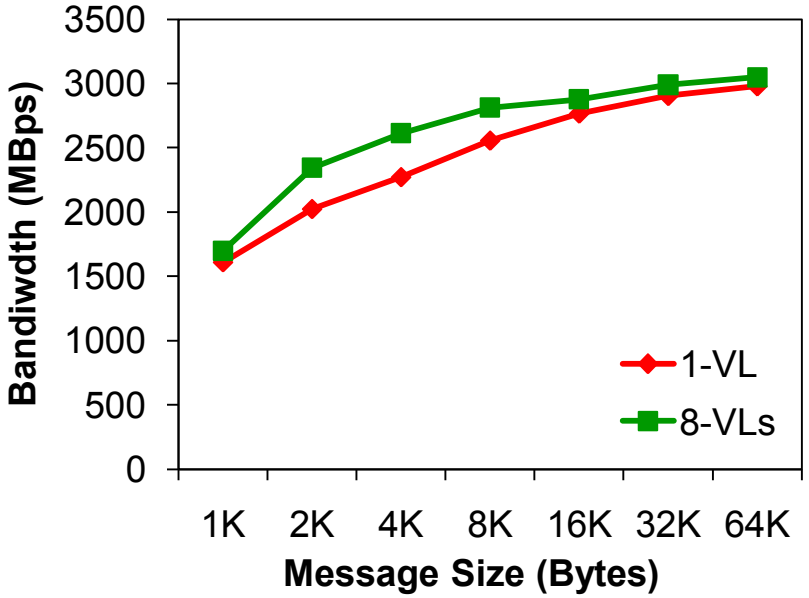


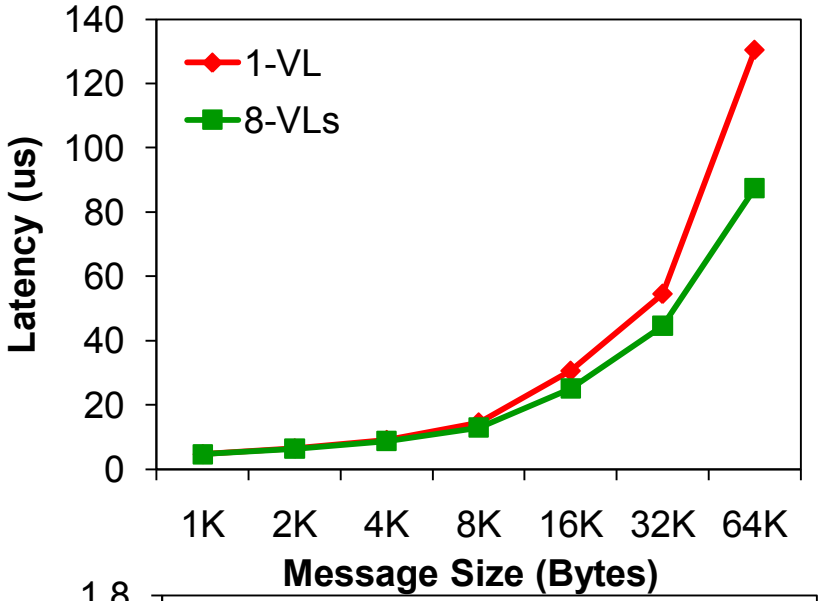




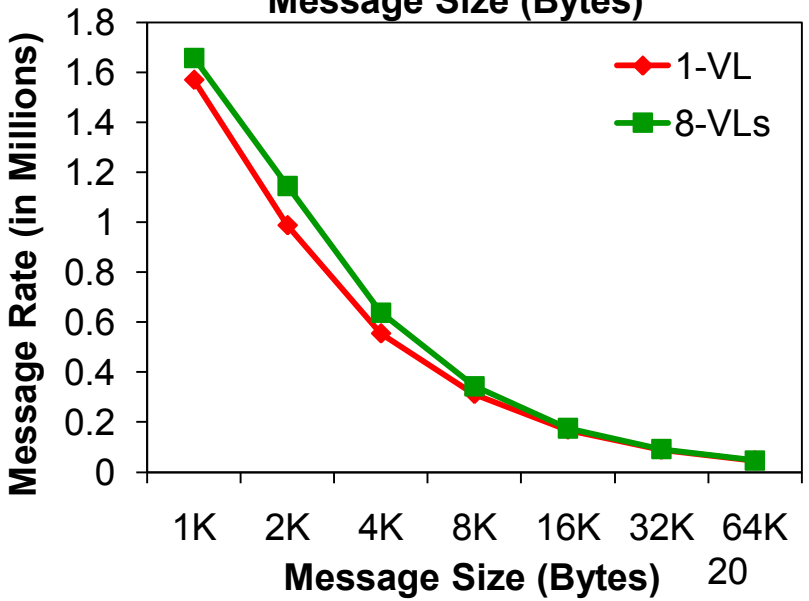
- Tests use 8 communicating pairs
  - One QP per pair
  - Packet size – 2 KB
    - Results same for 1 KB to 16 KB
- Traffic distribution using multiple VLs results in more predictable Inter arrival time
- Slight increase in average latency

MPI Level Point to Point Performance

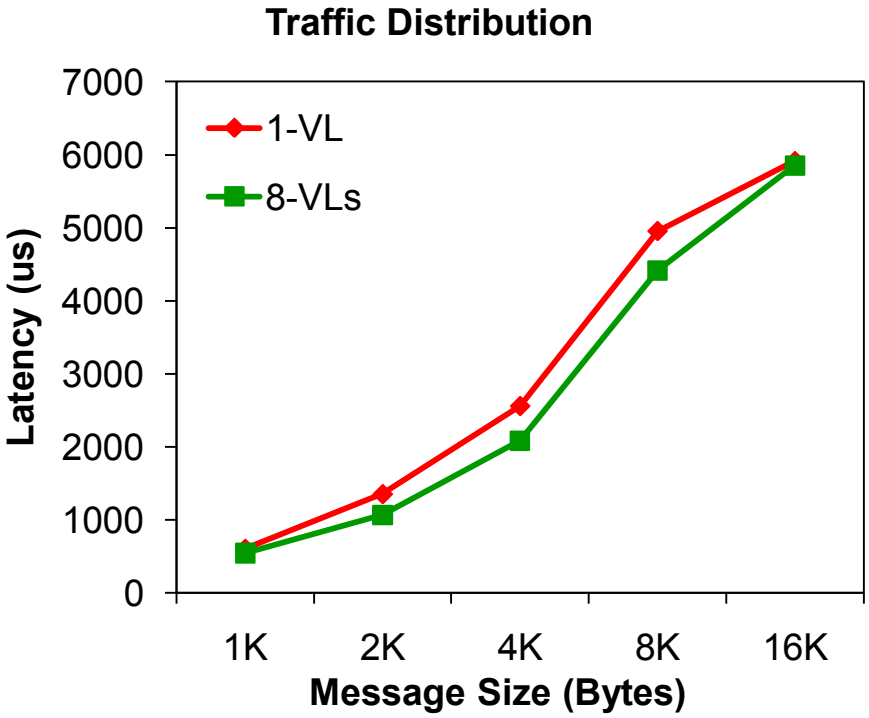


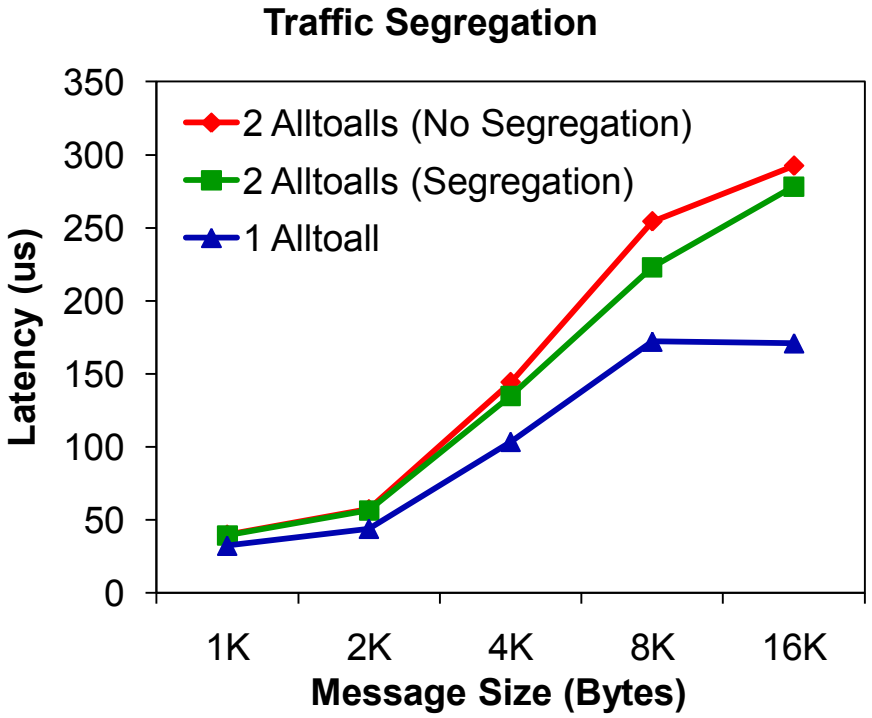


- Tests use 8 communicating pairs
  - One QP per pair
- Traffic distribution using multiple VLs result in better overall performance
- 13% performance improvement over case with just one VL



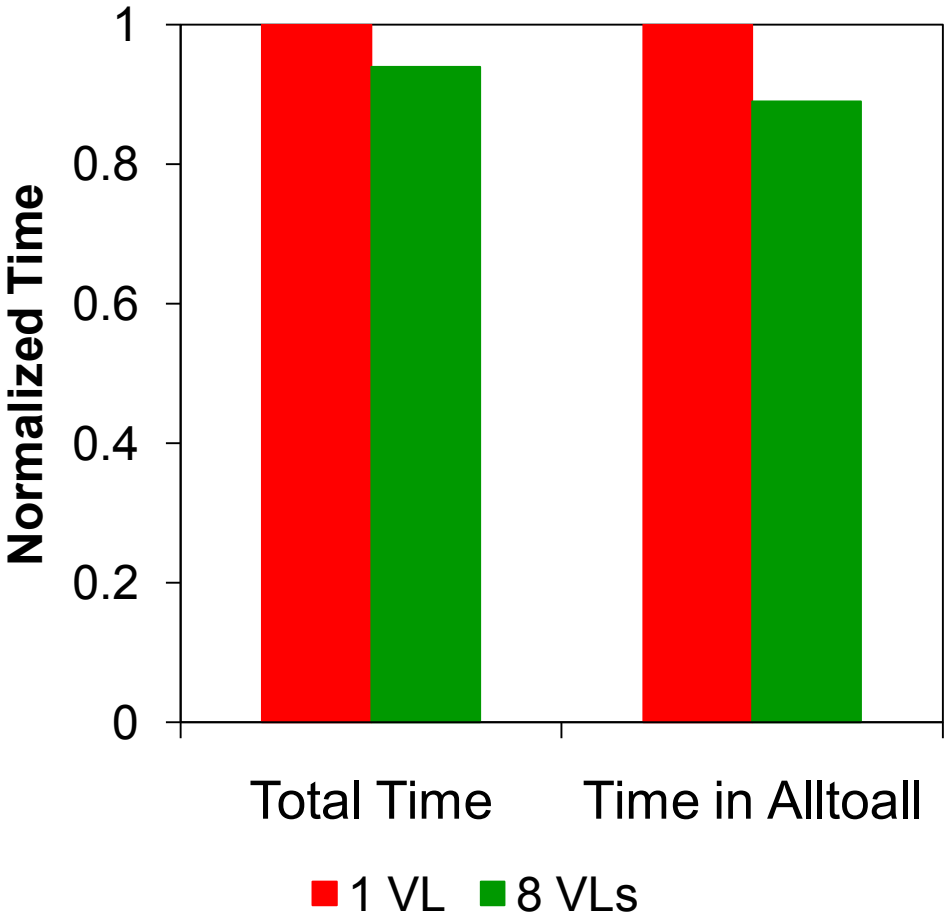
MPI Level Collective Performance





- For 64 process Alltoall, Traffic Distribution and Traffic Segregation through use of multiple VLs results in better performance
- 20% performance improvement seen with Traffic Distribution
- 12% performance improvement seen with Traffic Segregation

Application Level Performance



- CPMD application
  - 64 processes
- Traffic Distribution through use of multiple VLs results in better performance
- 11% performance improvement in Alltoall performance
- 6% improvement in overall performance

Outline

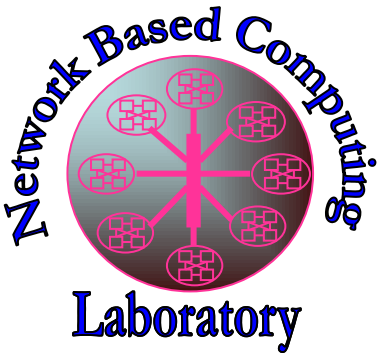
- Introduction & Motivation
- Problem Statement
- Design
- Performance Evaluation & Results
- Conclusions & Future Work

Conclusions & Future Work

- Explore use of Virtual Lanes to improve predictability and performance of HPC applications
- Integrate our scheme into MVAPICH2 MPI library and conduct performance evaluations at various levels
- Consistent increase in performance at verbs, MPI and application level evaluations
- Explore advanced schemes to improve performance using multiple virtual lanes
- Proposed solution will be available in future MVAPICH2 releases

Thank you!

{subramon, laipi, surs, panda}@cse.ohio-state.edu



Network-Based Computing Laboratory

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