Dynamic Time Variant Connection Management for PGAS Models on InfiniBand

Abhinav Vishnu\textsuperscript{1}, Manoj Krishnan\textsuperscript{1}
and Pavan Balaji\textsuperscript{2}

\textsuperscript{1}Pacific Northwest National Laboratory
Richland, WA
\textsuperscript{2}Argonne National Laboratory
Argonne, IL
Outline

- Introduction
- Background and Motivation
  - InfiniBand Connection Semantics
  - Global Arrays and ARMCI
- Overall Design
  - Efficient Connection Teardown
  - Connection Cache Management
- Performance Evaluation
  - Computational Chemistry
  - Sub-surface modeling
- Conclusions and Future Work
Introduction

- For runtime systems, scalable communication data structures is critical

- Communication data structures
  - Buffers (data, control messages ..)
  - Connections
    - End-points (Gemini, Seastar, BG ..)
    - One-to-one mapping (IB))
  - Registration data structures (Local for MPI, Local + Remote for PGAS)
  - ..... 

- Efficient connection management is important
  - 213 InfiniBand systems in TOP500

- PGAS Models are becoming popular
InfiniBand Connection Management

- On-demand pair-wise process creation
  - Cluster’02 (VIA), IPDPS’06, Cluster’08 (IB-MPI), CCGrid’10 (IB-PGAS)
  - Persistent through the application lifetime

- Unreliable datagram based approaches (ICS’07)
  - Natural fit for two-sided communication (send/receive model)
  - Designing get and bulk data transfer is prohibitive
  - Software maintained reliability

- eXtended Reliable Connection (XRC)
  - Connection memory increases with nodes and not processes (ICS’07, Cluster’08)
Use Cases for PGAS Models

- Frequently combined with non-SPMD execution models

- Task Based Computations
  - Dynamic load balancing and work stealing

- Linear communication over the application execution lifetime
  - Time-Variance in execution
  - Little temporal reuse (SC’09)
  - Connection persistence is not useful
Problem Statement

Given low temporal locality for PGAS models and non-SPMD executions

- What are the design choices for a disconnection protocol?
- What are the memory benefits and possible performance degradations?
Outline

- Introduction
- Background and Motivation
  - InfiniBand Connection Semantics
  - Global Arrays and ARMCI
- Overall Design
  - Efficient Connection Teardown
  - Connection Cache Management
- Performance Evaluation
  - Computational Chemistry
  - Sub-surface modeling
- Conclusions and Future Work
InfiniBand Transport Semantics

- **Reliable Connection**
  - Most frequently used
  - Supports In-order delivery, RDMA, QoS ..

- **Reliable Datagram**
  - Most RC features, but ..

- **Unreliable Connection**
  - RDMA, requires dedicated QP
  - No ordering

- **Unreliable Datagram**
  - Connectionless
  - limited message size to MTU
  - No ordering or reliability guarantees
Global Arrays is a PGAS programming model

- GA presents a shared view
- Provides one-sided communication model
- Used in wide variety of applications
  - Computational Chemistry
    - NWChem, Molcas, Molpro …
  - Bioinformatics
    - ScalaBLAST
  - Ground Water Modeling
    - STOMP

Physically distributed data

Global Address Space
Communication Runtime Systems for Global Arrays
- Used in Global Trees, and Chapel
- Provides one-sided communication runtime primitives

Currently Supported Platforms
- Commodity Networks
  - InfiniBand, Ethernet ..
- Leadership Class Machines
  - Cray XE6, Cray XTs
  - IBM BG’s
  - On-going -> BG/Q and BlueWaters

Upcoming features
- Fault tolerant continued execution (5.1)
- Energy Efficiency modes (5.2)
Connection Structure in ARMCI

Data Server thread

Client Process

Master Process
Connection Cache Management

- Number of active connections
  - Model?
  - Dynamic behavior for task-based computations
- Finding a victim connection
  - LRU
- LRU insufficient with communication cliques
  - Multi-phase applications (use-case: Flow + Chemistry)
  - Modified-LRU (LRU-M)
  - Temporal locality of connections
Overlap Disconnection Protocol

![Diagram showing the Overlap Disconnection Protocol with steps including WaitProc, Flush, Teardown Req, and Acknowledgement.]
Outline

- Introduction
- Background and Motivation
  - InfiniBand Connection Semantics
  - Global Arrays and ARMCI
- Overall Design
  - Efficient Connection Teardown
  - Connection Cache Management
- Performance Evaluation
  - Computational Chemistry and Sub-surface Modeling
- Conclusions and Future Work
Performance Evaluation

- **Evaluation Test Bed**
  - 160 Tflop system with 2310 Dual socket quad core Barcelona processor
  - InfiniBand DDR with PCI Express using DDR Voltaire switches

- **Original implementation is Global Arrays (GA) version 4.3**
  - The presented design is available with GA-5.0

- **Methodologies**
  - LRU, and LRU-M
  - Varying the number of connection entries in connection cache

- **Applications**
  - Northwest Chemistry (NWChem)
  - Sub-surface Transport on Multiple Phases
Performance Evaluation with NWChem

- Evaluation with pentane input deck on 6144 processes
- The connection cache has a total of 128, 32, and 4 entries
- Negligible performance degradation for 128 and 32 cache size
- Total connections created – 91-117
  - 3-4 times for 32 cache size
  - ~32 times for 4 cache size
Performance Evaluation : NWChem (Contd)

- Evaluation with siosi7 input deck on 4096 processes
- The connection cache has a total of 128, 32, and 4 entries
- Negligible performance degradation for 128 and 32 connection size
- Total connections created – 93-121
  - 3-4 times for 32 cache size
  - ~32 times for 4 cache size
Performance Evaluation: STOMP

- Evaluation on 8192 processes
- The connection cache has a total of 128, 32, and 4 entries
- LRU-M reduces the overall connection establishment and break time in comparison to LRU
Conclusions and Future Work

 Persistent on-demand connection approaches are insufficient

 Presented a design for connection management
  - Efficient connection cache management
  - A conducive protocol for PGAS Models

 Memory benefits for two class of applications

 Future Work:
  - Solve the problem for two-sided (pair-wise) connections
  - Apply the problem to other communication data structures (remote registration caches)
Questions

Global Arrays
- http://www.emsl.pnl.gov/docs/global/

ARMCI
- http://www.emsl.pnl.gov/docs/parsoft/armci/

HPC-PNL
- http://hpc.pnl.gov