Unifying UPC and MPI Runtimes: Experience with MVAPICH

Jithin Jose   Miao Luo   Sayantan Sur   D. K. Panda

Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University, USA
Introduction

- UPC and PGAS concepts are gaining interest
- Exascale programming model roadmap: MPI + “X”
- Is “X” == UPC? Maybe!
- MPI has been around for many years
  - Hundreds of man years invested in scientific software
  - Cannot afford to re-implement all this in PGAS
- InfiniBand – open standard, fast, scalable
  - MPI (MVAPICH, MVAPICH2) optimized to the hilt
  - Not productive to re-implement it for PGAS
- Must allow incrementally optimizing apps with UPC
- An unified runtime will be a first step in this direction
The Need for a Unified Runtime

- Deadlock when a message is sitting in one runtime, but application calls the other runtime
- Current prescription to avoid this is to barrier in one mode (either UPC or MPI) before entering the other runtime
- Bad performance!!
Coercing UPC over MPI not Optimal

• MPI does not provide Active Messages
  – AMs critical to UPC compilation and performance
  – Simulating AMs over MPI leads to performance loss
  – Not going to be included in MPI-3

• MPI RMA model for non cache-coherent machines
  – Penalizes vast majority of cache coherent machines
  – MPI-3 considering a proposal to support both cache-coherent and non cache-coherent machines (will take time)

• MPI will not support instant teams
  – Communicators in MPI require group communication

• Path forward: unify runtimes, not programming models
Outline

• Introduction

• Problem Statement

• Proposed Design

• Experimental Results & Analysis

• Conclusions & Future Work
Problem Statement

• Can we design a communication library for UPC?
  - Scalable on large InfiniBand clusters
  - Provides equal or better performance than existing runtime

• Can this library support both MPI and UPC?
  - Individually, both with great performance
  - Simultaneously, with great performance and less memory
Outline

• Introduction
• Problem Statement
• Proposed Design
• Experimental Results & Analysis
• Conclusions & Future Work
Overall Approach

- Unified runtime provides APIs for MPI and GASNet
- **INCR** (Integrated Communication Runtime)
The INCR Interface

• Different AM APIs based on size for optimization
  – Send short AM without arguments
  – Short AM (no data payload)
  – Medium AM (bounce buffer using RDMA FP)
  – Large AM (RDMA Put, on-demand connections)

• GASNet Extended interface for efficient RMA
  – Inline put
  – Put (may be internally buffered)
  – Put bulk (send buffer will not be touched, no buffering)
  – Get (RDMA Read)
Unified Implementation

- All resources are shared between MPI and UPC
  - Connections, buffers, memory registrations
  - Schemes for establishing connections (fixed, on-demand)
  - RDMA for large AMs and for PUT, GET
Various Configurations for running UPC and MPI Applications

**Pure MPI Applications**
- MVAPICH - MPI Standard Interface
- MVAPICH-Aptus Runtime
- InfiniBand Network

**UPC Compiler**
- Pure UPC Applications
- GASNet Interface and UPC Runtime
- InfiniBand Network

**GASNet Interface and UPC Runtime**
- GASNet IBVerbs Runtime
- InfiniBand Network

**GASNet-MPI**
- GASNet Interface and UPC Runtime
- InfiniBand Network

**GASNet-IBV**
- Pure UPC Compiler
- GASNet Interface and UPC Runtime
- InfiniBand Network

**GASNet Interface and UPC Runtime**
- Pure UPC Compiler
- GASNet-MPI Runtime
- InfiniBand Network

**GASNet-MPI**
- GASNet Interface and UPC Runtime
- InfiniBand Network

**GASNet-INCR**
- GASNet Interface and UPC Runtime
- InfiniBand Network

**Our Contribution**
- MVAPICH-INCR Implementation
- MVAPICH-Aptus Runtime
- InfiniBand Network

**Conclusion**
Outline

• Introduction
• Problem Statement
• Proposed Design
• Experimental Results & Analysis
• Conclusions & Future Work
MVAPICH and MVAPICH2 Software

• MVAPICH and MVAPICH2
  - High-performance, scalable, and fault-tolerant MPI library for InfiniBand/10GigE/iWARP and RDMA over Converged Enhanced Ethernet (RoCE)
  - Developed by Network-Based Computing Laboratory, OSU
  - 45,000 direct downloads from OSU site
  - Included in InfiniBand OFED, RedHat, SuSE etc.
  - Being used by more than 1,275 organizations worldwide, including many of the top 500 supercomputers (Jun’10 ranking)
    • 6th ranked 81,920 core (Pleiades) at NASA
    • 7th ranked 71,680 core (Tianhe-1) at NUDT, China
    • 11th ranked 62,976 core (Ranger) at TACC
    • 34th ranked 18,224 core (Juno) at LLNL
  - Proposed design will be incorporated in MVAPICH2 for public release

• MVAPICH Aptus runtime
  - Designed as a hybrid of Unreliable Datagram, Shared Receive Queues, Extended Reliable Connection (XRC), RDMA Fast Path
  - Designs will be integrated into MVAPICH2
Experimental Setup

• MVAPICH version 1.1 extended to support INCR
• Berkeley GASNet version 2.10.2 (--enable-pshm)
• Experimental Testbed
  - Type 1
    • Intel Nehalem (dual socket quad core Xeon 5500 2.4GHz)
    • ConnectX QDR InfiniBand
  - Type 2
    • Intel Clovertown (dual socket quad core Xeon 2.33GHz)
    • ConnectX DDR InfiniBand
  - Type 3
    • AMD Barcelona
    • Quad-socket quad-core Opteron 8530 processors
    • ConnectX DDR InfiniBand
Microbenchmark: upc-memput

- Cluster #1 used for these experiments
- GASNet-INCR performs identically with GASNet-IBV
- Comparatively GASNet-MPI performs much worse
- Mismatch of Active Message semantics
  - Message queue processing overheads
Microbenchmark: upc_memget

- GASNet-INCR performs identically with GASNet-IBV
- Due to mismatch of AM semantics with MPI leads to worse performance
• UPC “hello world” program
• GASNet-IBV establishes all-to-all reliable connections
  • Not scalable (may be improved in future release)
• GASNet-INCR best scalability due to inherent Aptus design
• Cluster #2 used for this experiment
Evaluation using UPC NAS Benchmarks

- GASNet-INCR performs equal or better than GASNet-IBV
- 10% improvement for CG [B, 128]
- 23% improvement for MG [B, 128]
- Cluster #3 used for these experiments
Evaluation using Hybrid NAS-FT

- Modified NAS FT UPC all-to-all pattern using MPI_Alltoall
- Truly hybrid program
- 34% improvement for FT (C, 128)
- Cluster #3 used for this experiment
Conclusions and Future Work

- Integrated Communication Runtime (INCR): supports MPI and UPC simultaneously
- Promising: MPI communication not harmed and UPC communication not penalized
- No need for programmer to barrier between UPC and MPI modes, as is current practice
- Pure UPC NAS: 10% improvement CG (B, 128), 23% improvement MG (B, 128)
- MPI+UPC FT: 34% improvement for FT (C, 128)
- Public release with MVAPICH2 coming soon
Thank You!

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

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

MVAPICH Web Page
http://mvapich.cse.ohio-state.edu/