Parallel Programming Languages and Accelerations

HPC@mellanox.com
Mellanox ScalableHPC
Mellanox ScalableHPC

- Offer high performing and scalable parallel programming libraries for HPC

- Support a comprehensive set of MPIs and PGAS languages
  - Integration of Mellanox acceleration technology into broad list of languages
  - Provide our own language library package when there is no open source alternative

- Integrates Mellanox acceleration components into MPIs/PGAS languages
  - MXM – MellanoX Messaging Accelerator
  - FCA – Mellanox Fabric Collective Accelerator
The I/O Bottleneck Paradigm

Server/Storage → Application → Communication Libraries → Network

Bottleneck

Server/Storage → Application → Communication Libraries → Network
The I/O Bottleneck Paradigm

- **Network**
- **Server/Storage**
  - **Application**
  - **Communication Libraries**

**Bottleneck**

- Highest throughput
- Lowest latency, Message rate
- Low CPU overhead, Hardware accelerations

Network
The I/O Bottleneck Paradigm – Scaling Issues

Network

Server/Storage

Application

Communication Libraries

Bottleneck

Network

Server/Storage

Application

Communication Libraries

Bottleneck

Highest throughput
Lowest latency, Message rate
Low CPU overhead,
Hardware accelerations

© 2012 MELLANOX TECHNOLOGIES
The I/O Bottleneck Paradigm – Co-Design Architecture

Network

Server/Storage

Application

Communication Libraries

Bottleneck

Network

Server/Storage

Application

Communication Libraries

Bottleneck

Extension of I/O communications (RDMA, collectives, synchronization etc)

Highest throughput
Lowest latency, Message rate
Low CPU overhead, Hardware accelerations
MPI/SHMEM/PGAS Architecture

Application

MPI/SHMEM/PGAS

InfiniBand Verbs

InfiniBand Network
Mellanox ScalableHPC Architecture

Application

MPI/SHMEM/PGAS

Mellanox Messaging (MXM)
- One-sided/Two-sided communication
- Intra-Node Shared Memory

Mellanox Collectives
- Collectives accelerations (FCA with CORE-Direct)

InfiniBand Verbs

InfiniBand Network (with Hardware Offloading)
High performing and scalable accelerations for collective operations

- Topology aware collectives take advantage of optimized message coalescing

- Makes use of powerful multicast capabilities in network for one-to-many communications

- Run collectives on separate service level so no interference with other communications

- Utilizes Mellanox CoreDirect collective hardware offload to minimize system noise
MellanoX Messaging (MXM)

- High performance and scalability for send/receive (or put/get) messages
  - Proper management of HCA resources and memory structures
  - Optimized intra-node communication
  - Hybrid transport technology for large scale deployments
  - Efficient memory registration
  - Connection management
  - Receive Side tag matching
  - Fully utilizes hardware offloads and capabilities
  - Incorporated in MLNX_OFED-1.5.3-300 and later
    - Also provided as a stand-alone package
ScalableHPC Communication Libraries
HPC Parallel Programming Models

- **MPI - Message Passing Interface**
  - Based on Send/Receive and collectives communication semantics

- **SHMEM - Shared Memory**
  - Provides logically shared memory model and one-way put/get communications

- **PGAS - Partitioned Global Address Space**
  - Message passing abstracted into a partitioned global address space
  - UPC (Unified Parallel C) is one example of a PGAS language
SHMEM Details

- SHared MEMory library
  - Library of functions somewhat similar to MPI (e.g. `shmem_get()`)
  - ….but SHMEM supports one-sided communication (puts/gets vs. MPI’s send/receive)

- SHMEM and PGAS both allow for a unique combination of using a ‘Distributed Memory Model’ (like MPI), and a ‘Shared Memory Model’ (like SMP machines)

- Cray first introduced SHMEM in 1993

- OpenSHMEM consortium formed to consolidate the various SHMEM versions into a widely accepted standard

- Mellanox ScalableSHMEM based on OpenSHMEM-1.0 specification with FCA/MXM integration
**UPC Details**

- UPC, or ‘Unified Parallel C’ is another PGAS language
- Higher level abstraction than MPI or SHMEM
- Allows programmers to directly represent and manipulate distributed data structures
- Commercial compilers are available for Cray, SGI and HP machines
- Open source compiler from LBNL/UCB (Berkeley UPC) available on InfiniBand
- Mellanox ScalableUPC based on BUPC with FCA/MXM integration
FCA Details
What are Collective Operations?

- Collective Operations are Group Communications involving all processes in job

- Synchronous operations
  - By nature consume many ‘Wait’ cycles on large clusters

- Popular examples
  - Barrier
  - Reduce
  - Allreduce
  - Gather
  - Allgather
  - Bcast
Collective Operation Challenges at Large Scale

- Collective algorithms are not topology aware and can be inefficient

- Congestion due to many-to-many communications

- Slow nodes and OS jitter affect scalability and increase variability
Mellanox Fabric Collectives Accelerations (FCA)

**Mellanox InfiniBand Switches**
High performance IB multicast for result distribution

**FCA Manager**
Topology-based collective tree
Separate Virtual network
IB multicast for result distribution

**FCA Agents**
Library integrated with MPI
Intra-node optimizations
CoreDirect integration
Collective Example – Allreduce using Recursive Doubling

- Collective Operations are Group Communications involving all processes in job

- A 4000 process Allreduce using recursive doubling is 12 stages
Scalable Collectives with FCA

1st tier coalescing

Intra-node processing

2nd tier coalescing (result at root)

Multicast Result
Performance Results
FCA collective performance with OpenMPI

IMB Barrier - FDR

- With FCA
- Without FCA
FCA collective scalability for SHMEM

**Barrier Collective**

- Processes (PPN=8)
  - Without FCA
  - With FCA

**Reduce Collective**

- Processes (PPN=8)
  - Without FCA
  - With FCA

**8-Byte Broadcast**

- Processes (PPN=8)
  - Without FCA
  - With FCA
Mellanox MXM – HPCC Random Ring Latency

FDR - HPCC Random Ring (16n, 8ppn)

- OpenMPI with MXM
- OpenMPI w/XRC
- OpenMPI w/o XRC
Thank You
HPC@mellanox.com