MPI on the Cray XC30

Aaron Vose
4/15/2014

Many thanks to Cray’s Nick Radcliffe and Nathan Wichmann for slide ideas.
MPI on XC30 - Overview

• Cray MPI.

• MPI Message Pathways.

• MPI Environment Variables.
  – Environment variables to change MPI defaults.
  – Consequences of changing defaults.

• MPI Progress Engine.
MPI on XC30 - Cray MPI

• Cray MPI uses MPICH2 from Argonne.

• Great start for robust, feature-rich MPI.

• Cray provides enhancements:
  – Low-level communication libraries.
  – Point-to-point performance tuning.
  – Collective performance tuning.
  – Shared memory built on top of XPMEM.
MPI on XC30 - Cray MPI Stack

XE6/XK7 Software Stack

- MPI Apps
- PGAS/SHMEM Apps
- Gemini-aware optimized PE components
  - MPI
  - SHMEM
  - UPC & CAF PGAS

- User-level GNI
- Kernel
  - CLE Linux Core
  - Kernel-level GNI
  - Gemini Hardware Abstraction Layer (HAL)
- Gemini Network

XC30 Software Stack

- MPI Apps
- PGAS/SHMEM Apps
- Aries-aware optimized PE components
  - MPI
  - SHMEM
  - UPC & CAF PGAS

- User-level GNI
- Kernel
  - CLE Linux Core
  - Kernel-level GNI
  - Aries Hardware Abstraction Layer (HAL)
- Aries Network
MPI Pathways - Overview

• Gemini / Ares NIC Resources.

• MPI Message Pathways:
  – Internode Messages:
    • Eager Message Paths: E0, E1
    • Rendezvous Message Paths: R0, R1
  – Intranode Messages:
    • memcpy / xpmem

• How to control which paths are used/when.
MPI Pathways - NIC Resources

• FMA (Fast Memory Access):
  – Used for small messages.
  – Called directly from user mode.
  – Low overhead: low latency.

• BTE (Block Transfer Engine):
  – Used for large messages.
  – All ranks on a node share BTE resources (4 V.C.).
  – Once started, BTE transfers progress without CPU.
**MPI Pathways - E0,E1,R0,R1**

- Four main pathways:
  - Two Eager: E0, E1.
  - Two Rendezvous: R0, R1.

- Pathways generally based on message size:

<table>
<thead>
<tr>
<th>E0</th>
<th>E1</th>
<th>R0</th>
<th>R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 512</td>
<td>1k, 2k, 4k, 8k</td>
<td>16k, 32k, 64k, 128k, 256k</td>
<td>512k, 1M, 2M, 4M, ...</td>
</tr>
</tbody>
</table>

**MPICH_GNI_MAX_VSHORT_MSG_SIZE**

**MPICH_GNI_MAX_EAGER_MSG_SIZE**

**MPICH_GNI_NDREG_MAXSIZE**
MPI Pathways - Eager

- **Eager Pathway:**
  - Designed for small messages.
  - Data is transferred on encountering send call.
  - **E0:**
    - Small messages that fit into GNI SMSG Mailbox.
  - **E1:**
    - Too big for SMSG Mailbox, but small enough for pre-allocated MPI buffers.
**MPI Pathways – E0**

- **Sender**

1. GNI SMSG Send (MPI header + user data)

- **SMSG Mailboxes**

- **Receiver**

2. Memcpy

- **Eager messages fit in GNI SMSG Mailbox.**
MPI Pathways – E1

1. Memcpy to pre-allocated MPI buffers
2. GNI SMSG Send (MPI header)
3. RDMA GET
4. GNI SMSG Send (Recv done)
5. Memcpy

• Eager messages don’t fit in SMSG Mailbox.
MPI Pathways - Rendezvous

• Rendezvous Pathway
  – Designed for large messages.
  – Data is transferred after receiver has posted matching receive for a previously initiated send.
  – R0:
    • RDMA “GET”.
    • Can overlap comm/comp in this path: issue send first.
  – R1:
    • Pipelined RDMA “PUTs”.
    • Comm/comp overlap with progress engine/threads.
MPI Pathways - R0

1. Register App Send Buffer
2. GNI SMSG Send (MPI header)
3. Register Recv Buffer
4. RDMA GET
5. GNI SMSG Send (Recv done)

- No extra copies. Best comm/comp overlap.
- Repeat 2-6; chunks: MPI_GNI_MAX_NDREG_SIZE.
MPI Progress Engine

• Async Progress Threads:
  – Each MPI rank spawns a “helper thread”.
  – Threads progress MPI engine during app compute.
  – Progresses only inter-node, rendezvous messages.

• To enable (app is 1 stream per core; HT off):
  – MPICH_NEMESIS_ASYNC_PROGRESS=1
  – MPICH_MAX_THREAD_SAFETY=multiple
  – MPICH_GNI_USE_UNASSIGNED_CPUS=enabled
MPI Environment Variables 1

• Eager messages require buffers on receiver:
  – Can increase buffer size: MPICH_GNI_NUM_BUFS

• Enable/Disable Cray collective optimizations:
  – MPICH_COLL_OPT_OFF=mpi_allgather

• Allocate mailbox resources on demand / once:
  – MPICH_GNI_DYNAMIC_CONN=disabled
Controlling which memory is used for SMSG:

- Default is on the memory of faulting process.
- For optimal MPI performance, place on die0 (die0 is near the Aries NIC):
  - MPICH_GNI_MBOX_PLACEMENT=nic
  - Only applies to first 4096 mailboxes of each rank.

Mixed MPI/SHMEM/UPC/CAF code w/ errors?

- Try: MPICH_GNI_DMAPP_INTEROP=disabled
Conclusion

• Four main pathways:
  – Two Eager (E0, E1) and two Rendezvous (R0, R1):
    • MPICH_GNI_MAX_VSHORT_MSG_SIZE (E0->E1)
    • MPICH_GNI_MAX_EAGER_MSG_SIZE (E1->R0)
    • MPICH_GNI_NDREG_MAXSIZE (R0->R1)

• MPI progress engine:
  – MPICH_NEMESIS_ASYNC_PROGRESS=1
  – MPICH_MAX_THREAD_SAFETY=multiple
  – MPICH_GNI_USE_UNASSIGNED_CPUS=enabled