Enabling HPC and Deep Learning Applications using MVAPICH2-GDR on SDSC Systems

Mahidhar Tatineni
MVAPICH2 User Group (MUG) Meeting
August 21, 2019
SDSC is one of the original NSF-funded Supercomputer Centers

- Established 1985 as one of original NSF-funded supercomputer centers
- Transitioned from General Atomics to UCSD in 1997
- 225 staff with diverse computational, HPC, operational and applications backgrounds.
- 12,000 sq. ft. main data center, 5,000 sq. ft. secure data center (FISMA Moderate security level)
- 4 major supercomputers (Comet, GordonS, Popeye, Triton Cluster)
- ~15 petabytes capacity high performance storage systems
Overview

- Comet Hardware
- Applications Overview
- Singularity containerization on Comet
- MVAPICH2-GDR via containers
- Benchmark results: OSU, TensorFlow w/ Horovod
- Future system: Expanse

NSF Award# 1341698, Gateways to Discovery: Cyberinfrastructure for the Long Tail of Science
PI: Michael Norman
Co-PIs: Shawn Strande, Amit Majumdar, Robert Sinkovits, Mahidhar Tatineni
SDSC Project in Collaboration with Indiana University (led by Geoffrey Fox)
Comet: System Characteristics

- Total peak flops ~2.76 PF
- Dell primary integrator
  - Intel Haswell processors w/ AVX2
  - Mellanox FDR InfiniBand
- 1,944 standard compute nodes (46,656 cores)
  - Dual CPUs, each 12-core, 2.5 GHz
  - 128 GB DDR4 2133 MHz DRAM
  - 2*160GB GB SSDs (local disk)
- 72 GPU nodes
  - 36 nodes with two NVIDIA K80 cards, each with dual Kepler3 GPUs
  - 36 nodes with 4 P100 GPUs each
- 4 large-memory nodes
  - 1.5 TB DDR4 1866 MHz DRAM
  - Four Haswell processors/node
  - 64 cores/node

- Hybrid fat-tree topology
  - FDR (56 Gbps) InfiniBand
  - Rack-level (72 nodes, 1,728 cores) full bisection bandwidth
  - 4:1 oversubscription cross-rack

- Performance Storage (Aeon)
  - 7.6 PB, 200 GB/s; Lustre
  - Scratch & Persistent Storage segments

- Durable Storage (Aeon)
  - 6 PB, 100 GB/s; Lustre
  - Automatic backups of critical data

- Home directory storage
- Gateway hosting nodes
- Virtual image repository
- 100 Gbps external connectivity to Internet2 & ESNet
Comet K80 node architecture

<table>
<thead>
<tr>
<th></th>
<th>GPU0</th>
<th>GPU1</th>
<th>GPU2</th>
<th>GPU3</th>
<th>mlx4_0</th>
<th>CPU Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU0</td>
<td>X</td>
<td>PIX</td>
<td>SOC</td>
<td>SOC</td>
<td>SOC</td>
<td>0-0,2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20,22-22</td>
</tr>
<tr>
<td>GPU1</td>
<td>PIX</td>
<td>X</td>
<td>SOC</td>
<td>SOC</td>
<td>SOC</td>
<td>0-0,2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20,22-22</td>
</tr>
<tr>
<td>GPU2</td>
<td>SOC</td>
<td>SOC</td>
<td>X</td>
<td>PIX</td>
<td>PHB</td>
<td>1-1,3-3,5-5,7-7,9-9,11-11,13-13,15-15,17-17,19-19,21-21,23-23</td>
</tr>
<tr>
<td>GPU3</td>
<td>SOC</td>
<td>SOC</td>
<td>PIX</td>
<td>X</td>
<td>PHB</td>
<td>1-1,3-3,5-5,7-7,9-9,11-11,13-13,15-15,17-17,19-19,21-21,23-23</td>
</tr>
<tr>
<td>mlx4_0</td>
<td>SOC</td>
<td>SOC</td>
<td>PHB</td>
<td>PHB</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- X = Self
- SOC = Connection traversing PCIe as well as the SMP link between CPU sockets (e.g. QPI)
- PHB = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
- PXB = Connection traversing multiple PCIe switches (without traversing the PCIe Host Bridge)
- PIX = Connection traversing a single PCIe switch
- NV# = Connection traversing a bonded set of # NVLinks

- 4 GPUs per node
- GPUs (0,1) and (2,3) can do P2P communication
- Mellanox InfiniBand adapter associated with second socket (GPUs 2, 3)
Comet P100 node architecture

<table>
<thead>
<tr>
<th>GPU0</th>
<th>GPU1</th>
<th>GPU2</th>
<th>GPU3</th>
<th>mlx4_0</th>
<th>CPU Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>PIX</td>
<td>SOC</td>
<td>SOC</td>
<td>PHB</td>
<td>0-2-4-6-8-10-12-14-16-18-20-22-24-26</td>
</tr>
<tr>
<td>PIX</td>
<td>X</td>
<td>SOC</td>
<td>SOC</td>
<td>PHB</td>
<td>0-2-4-6-8-10-12-14-16-18-20-22-24-26</td>
</tr>
<tr>
<td>SOC</td>
<td>SOC</td>
<td>X</td>
<td>PIX</td>
<td>SOC</td>
<td>1-3-5-7-9-11-13-15-17-19-21-23-25-27</td>
</tr>
<tr>
<td>SOC</td>
<td>SOC</td>
<td>PIX</td>
<td>X</td>
<td>SOC</td>
<td>1-3-5-7-9-11-13-15-17-19-21-23-25-27</td>
</tr>
<tr>
<td>PHB</td>
<td>PHB</td>
<td>SOC</td>
<td>SOC</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Legend:

- **X** = Self
- **SOC** = Connection traversing PCIe as well as the SMP link between CPU sockets (e.g. QPI)
- **PHB** = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
- **PXB** = Connection traversing multiple PCIe switches (without traversing the PCIe Host Bridge)
- **PIX** = Connection traversing a single PCIe switch
- **NV#** = Connection traversing a bonded set of # NVLinks

- 4 GPUs per node
- GPUs (0,1) and (2,3) can do P2P communication
- Mellanox InfiniBand adapter associated with first socket (GPUs 0, 1)
Applications on Comet GPU Nodes

- Comet GPU partitions are typically the most busy ones on Comet, with a higher expansion factor than the regular compute nodes.

- The bulk of the usage is from the phylogenetic analysis code BEAST, a wide range of molecular dynamics, chemistry codes (AMBER, NAMD, GROMACS, LAMMPS, HOOMD-Blue), and photon tracking simulation code (IceCube project).

- Most of the MD codes can run multi-node with MPI or ibverbs based implementations.

- Rapid growth in machine/learning deep learning applications. Large number of allocated projects (87 allocations, 50 institutions) across many disciplines. In aggregate they consume a small amount of the overall resource (~10-15% of the GPU time on Comet).
Singularity on Comet
Singularity: Provides Flexibility and Portability

- Singularity has been available on Comet since 2016 and its become very popular on Comet.
- Singularity runs in user space, and requires very little user support – in fact it actually reduces the support load in most cases.
- Singularity allows groups to easily migrate complex software stacks and workflows to Comet.

Glotzer Lab
Univ. Michigan
HOOMD-Blue and other packages
Singularity Use Cases

• Applications with newer library OS requirements than available on the HPC system – e.g. TensorFlow, PyTorch, Caffe2 (SDSC staff maintain optimal versions for Comet).

• Commercial application binaries with specific OS requirements.

• Importing singularity and docker images to enable use in a shared HPC environment. Usually this is entire workflows with a large set of tools bundled in one image.

• Training – encapsulate all the requirements in an image for workshops and SDSC summer institute. Also makes it easy for users to try out outside of the training accounts on Comet.
Machine Learning/Deep Learning on Comet via Singularity

• Quite a few machine learning/deep learning applications on Comet are enabled via Singularity.

• Lot of these packages are constantly upgraded and the dependency list is difficult to update in the standard Comet environment.

Install options
- Singularity image provides dependencies and user can compile actual application from source. e.g. PyTorch
- Entire dependency stack and the application is in the image. e.g Keras with backend to TensorFlow and some additional python libraries

Run options
- Most cases are run on single GPU nodes (4 GPUs at most)
- Multi-node runs are not common but we are starting to see requests => look at MVAPICH2-GDR based options.
Typical usage on Comet

• Several frameworks available for developing, training, testing machine learning/deep learning models.

• TensorFlow, PyTorch most commonly used on Comet system. Others include scikit-learn, Caffe2, Theano, MXNet.

• Both TensorFlow and PyTorch are available via Singularity on Comet (for optimized builds from source). Also have conda based installs on the host (w/o Singularity).

• Currently most users run with single GPU node with 4 GPUs. However, there are a large number of ML/DL projects spinning up in the past year or two, with some requiring scale up.
MVAPICH2-GDR via Singularity Containers

- **Installed in Singularity Container**
  - NVIDIA driver, CUDA 9.2 (this can alternately be pulled in via the --nv flag)
  - Mellanox OFED stack
  - gdrcopy library - *kernel module is on the host system.*
  - MVAPICH2-GDR (w/o slurm)
  - TensorFlow (conda install)
  - Horovod (pip installed)

- **Other modifications:**
  - Wrap ssh binary in Singularity container to run remote commands via image environment (more on this next slide)
MVAPICH2-GDR Job Launch w/ Singularity Containers

- Use `mpirun/mpirun_rsh` on the host (external to the image) and wrap the executable/script in Singularity "exec" command.

- Launch using `mpirun_rsh` within the Singularity image.
  - Needs `ssh` to be wrapped so that the remote command is launching in `ssh` environment
  - `ssh` binary was moved in container, and then wrapped `ssh` is used (to point to `ssh + singularity command`).
Benchmark Results

- OSU Micro-Benchmark results (both native and via Singularity):
  - osu_latency
  - osu_bw
  - osu_allreduce

- HOOMD-Blue Benchmark

- TensorFlow (tf_cnn_benchmarks) with MVAPICH2-GDR
  - Run using Singularity based install.
  - Horovod
OSU Latency (osu_latency) Benchmark
Intra-node, K80 nodes

- Latency between GPU 2, GPU 3: 2.82 µs
- Latency between GPU 1, GPU 2: 3.18 µs
OSU Latency (osu_latency) Benchmark Inter-node, K80 nodes

- Latency between GPU 2, process bound to CPU 1 on both nodes: 2.27 µs
- Latency between GPU 2, process bound to CPU 0 on both nodes: 2.47 µs
- Latency between GPU 0, process bound to CPU 0 on both nodes: 2.43 µs
OSU Bandwidth (osu_bw) Benchmark
Intra-node, P100 nodes
OSU Latency (osu_latency) Benchmark Inter-node, P100 nodes

- Latency between GPU 0, process bound to CPU 0 on both nodes: 2.17 µs
- Latency between GPU 2, process bound to CPU 1 on both nodes: 2.35 µs
MVAPICH2-GDR (v2.3.2) Results Using Containerized Approach

![Graph showing osu_latency, GPU 0 on both nodes]
MVAPICH2-GDR (v2.3.2) Results Using Containerized Approach

osu_bw, GPU 0 on both nodes

Bandwidth (MB/s)

Message Size (Bytes)

Container
MVAPICH2-GDR (v2.3.2) Results Using Containerized Approach

![Graph showing performance results for osu_allreduce, 2 GPUs per node (0,1)]
HOOMD-blue Benchmarks using MVAPICH2-GDR

- HOOMD-blue is a *general-purpose* particle simulation toolkit
- Results for Hexagon benchmark.

References:
- HOOMD-blue web page: [http://glotzerlab.engin.umich.edu/hoomd-blue/](http://glotzerlab.engin.umich.edu/hoomd-blue/)
- HOOMD-blue Benchmarks page: [http://glotzerlab.engin.umich.edu/hoomd-blue/benchmarks.html](http://glotzerlab.engin.umich.edu/hoomd-blue/benchmarks.html)
HOOMD-Blue: Hexagon Benchmark
Strong scaling on K80 nodes
TensorFlow Benchmark (tf_cnn_benchmarks)

- Interactive access to resources using "srun"
- Get an interactive shell in Singularity image environment
  
  ```
  singularity shell ./centos7mv2gdr.img
  ```
- Run benchmark using hosts (get list from Slurm)
  ```
  export MV2_PATH=/opt/mvapich2/gdr/2.3.2/mcast/no-openacc/cuda9.2/mofed4.5/mpirun/gnu4.8.5
  export MV2_USE_CUDA=1
  export MV2_USE_MCAST=0
  export MV2_GPUDIRECT_GDRCOPY_LIB=/opt/gdrcopy/lib64/libgdrapi.so
  export CUDA_VISIBLE_DEVICES=0,1
  export MV2_SUPPORT_TENSOR_FLOW=1
  
  $MV2_PATH/bin/mpirun_rsh -export -np 4 comet-34-16 comet-34-16 comet-34-17 comet-34-17 python tf_cnn_benchmarks.py --model=resnet50 --variable_update=horovod > TF_2NODE_4GPU.txt
  ```
Sample Output

```
/usr/bin/ssh_orig -q comet-34-17 $(which singularity) exec -B /etc/ssh /oasis/scratch/comet/mahidhar/temp_project/RPMS/centos7mv2gdr.img /bin/bash -c "cd /oasis/scratch/comet/mahidhar/temp_project/RPMS/benchmarks/scripts/tf_cnn_benchmarks;
/bin/env LD_LIBRARY_PATH=/opt/mvapich2/gdr/2.3.2/mcast/no-openacc/cuda9.2/mofed4.5/mpirun/gnu4.8.5/lib64::/.singularity.d/libs MPISPAWN_MPIRUN_MPD=0 USE_LINEAR_SSH=1 MPISPAWN_MPIRUN_HOST=comet-34-13.sdsc.edu MPISPAWN_MPIRUN_HOSTIP=198.202.120.189 MPIRUN_RSH_LAUNCH=1 MPISPAWN_CHECKIN_PORT=35542 MPISPAWN_MPIRUN_PORT=35542 MPISPAWN_NNODES=4 MPISPAWN_GLOBAL_NPROCS=16 MPISPAWN_MPIRUN_ID=137874 MPISPAWN_ARGC=4 MPDMAN_KVS_TEMPLATE=kvs_814_comet-34-13.sdsc.edu_137874 MPISPAWN_LOCAL_NPROCS=4 MPISPAWN_ARGV_0='python' MPISPAWN_ARGV_1='tf_cnn_benchmarks.py' MPISPAWN_ARGV_2='--model=resnet50' MPISPAWN_ARGV_3='--variable_update=horovod' MPISPAWN_ARGC=4 MPISPAWN_GENERIC_ENV_COUNT=0 MPISPAWN_ID=3 MPISPAWN_WORKING_DIR=/oasis/scratch/comet/mahidhar/temp_project/RPMS/benchmarks/scripts/tf_cnn_benchmarks MPISPAWN_MPIRUN_RANK_0=12 MPISPAWN_MPIRUN_RANK_1=13 MPISPAWN_MPIRUN_RANK_2=14 MPISPAWN_MPIRUN_RANK_3=15 /opt/mvapich2/gdr/2.3.2/mcast/no-openacc/cuda9.2/mofed4.5/mpirun/gnu4.8.5/bin/mpispawn 0"
/usr/bin/ssh_orig -q comet-34-14 $(which singularity) exec -B /etc/ssh /oasis/scratch/comet/mahidhar/temp_project/RPMS/centos7mv2gdr.img /bin/bash -c "cd /oasis/scratch/comet/mahidhar/temp_project/RPMS/benchmarks/scripts/tf_cnn_benchmarks;
/bin/env LD_LIBRARY_PATH=/opt/mvapich2/gdr/2.3.2/mcast/no-openacc/cuda9.2/mofed4.5__/--More--(5%)```

Sample Output

Mode: training
SingleSess: False
Batch size: 1024 global
            64 per device
Num batches: 100
Num epochs: 0.88
Devices: ['horovod/gpu:0', 'horovod/gpu:1', 'horovod/gpu:2', 'horovod/gpu:3',
          'horovod/gpu:14', 'horovod/gpu:15']
NUMA bind: False
Data format: NCHW
Optimizer: sgd
Variables: horovod

Generating training model
Initializing graph
Running warm up
Done warm up
Step     Img/sec total_loss
1        images/sec: 145.0 +/- 0.0 (jitter = 0.0)    7.715
10       images/sec: 144.7 +/- 0.4 (jitter = 0.6)    7.719
20       images/sec: 144.4 +/- 0.4 (jitter = 1.0)    7.568
--More--(21%)
TensorFlow Benchmark (GPU 0,1,2,3 on each node)
TensorFlow Benchmark
(GPU 0,1 on each node)
Computing Without Boundaries: Cyberinfrastructure for the Long Tail of Science

Upcoming system

EXpanse
Computing Without Boundaries

Category 1: Capacity System, NSF Award # 1928224
PI: Mike Norman, CoPIs: Ilkay Altintas, Amit Majumdar, Mahidhar Tatineni, Shawn Strande
Thank you to our collaborators, partners, and the SDSC team!

Ilkay Altintas
Trevor Cooper
Jerry Greenberg
Eva Hocks
Tom Hutton
Christopher Irving
Marty Kandes
Amit Majumdar
Dima Mishin
Mike Norman
Wayne Pfeiffer
Scott Sakai
Fernando Silva
Bob Sinkovits
Subha
Sivagnanam
Shawn Strande
Mahidhar Tatineni
Mary Thomas
Nicole Wolter

San Diego Supercomputer Center

EXpanse
Computing without Boundaries

In Production October 2020

XSEDE
Extreme Science and Engineering Discovery Environment
EXPA NSE
COMPUTING WITHOUT BOUNDARIES
5 PETAFLOP/S HPC and DATA RESOURCE

HPC RESOURCE
13 Scalable Compute Units
728 Standard Compute Nodes
52 GPU Nodes: 208 GPUs
4 Large Memory Nodes

DATA CENTRIC ARCHITECTURE
12PB Perf. Storage: 140GB/s, 200k IOPS
Fast I/O Node-Local NVMe Storage
7PB Ceph Object Storage
High-Performance R&E Networking

REMOTE CI INTEGRATION

LONG-TAIL SCIENCE
Multi-Messenger Astronomy
Genomics
Earth Science
Social Science

INNOVATIVE OPERATIONS
Composable Systems
High-Throughput Computing
Science Gateways
Interactive Computing
Containerized Computing
Cloud Bursting
Expanse is a heterogeneous architecture designed for high performance, reliability, flexibility, and productivity.
Expanse is a heterogeneous architecture designed for high performance, reliability, flexibility, and productivity.

SDSC Scalable Compute Units (SSCU) (13)
- 56 Standard compute nodes
- 4 GPU nodes

Compute Nodes
- Two-socket, Intel Cooper Lake
- 256 GB DDR4
- 1 TB NVMe

GPU Nodes
- 4x V100/follow-on
- SMX2
- 32 GB GDDR
- 1.6 TB NV

Object Storage
- 7 PB
- CEPH

Performance Storage
- 12 PB
- 140 GB/s
- Lustre
- Native IB

Interconnect
- HDR 100
- Full bisection in an SSCU
- 3:1 oversubscription between SSCUs

Large Memory Nodes
- 4 @ 2TB each
Summary

• MVAPICH2-GDR can support both HPC and ML/DL workloads on Comet.
• Can be run in a containerized environment providing software stack and OS flexibility. No significant performance changes.
• Expect continued increase in adoption on upcoming system (Expanse) which features a GPU partition.

• NSF Award# 1341698, Gateways to Discovery: Cyberinfrastructure for the Long Tail of Science
  PI: Michael Norman Co-PIs: Shawn Strande, Amit Majumdar, Robert Sinkovits, Mahidhar Tatineni
  SDSC Project in Collaboration with Indiana University (led by Geoffrey Fox)
• NSF Award #1565336, SHF: Large: Collaborative Research: Next Generation Communication Mechanisms exploiting Heterogeneity, Hierarchy and Concurrency for Emerging HPC Systems
  Collaborative project with OSU (Lead Institution, PI: DK Panda), OSC, SDSC, TACC
• NSF Award # 1928224, Category 1: Capacity System: Computing Without Boundaries: Cyberinfrastructure for the Long Tail of Science
  PI: Mike Norman, CoPIs: Ilkay Altintas, Amit Majumdar, Mahidhar Tatineni, Shawn Strande
Thanks!

Questions: Email mahidhar@sdsc.edu