MVAPICH2 at NERSC

Shahzeb Siddiqui (LBNL)
Sameer Shende (University of Oregon)
Prathmesh Sambrekar (LBNL/Arizona State University)

10th Annual MVAPICH User Group (MUG)
Aug 24th, 2022
National Energy Research Scientific Computing Center (NERSC) is scientific computing facility as part of the Office of Science in the U.S Department of Energy.

At NERSC we support up to 8000+ users that span a wide range of disciplines including climate modeling, simulation of universe, high energy physics experiments, protein simulation, and many more.

NERSC most recent supercomputer called Perlmutter is named after astrophysicist Saul Perlmutter for his Nobel Prize in Physics for discovering the rate of expansion of the universe.
● **Perlmutter** is a HPE Cray Shasta platform composed of CPU-only and GPU accelerated nodes.

● The system will be delivered in two phases

● The system contains 1536 GPU accelerated nodes with NVIDIA A100 GPUs and 3072 CPU-only nodes powered with AMD EPYC Milan CPUs

● Perlmutter is currently ranked #7 on the [Top500 list](https://www.top500.org/list/) with a peak performance of 93.75 PFlop/s
We recently published a technical report Software Deployment Process at NERSC: Deploying the Extreme-scale Scientific Software Stack (E4S) Using Spack at the National Energy Research Scientific Computing Center (NERSC) that outlines our software deployment process.

We are working with MVAPICH2 team to enable MPI support for Perlmutter when building the spack stack.

Currently we are building the spack stack with cray-mpich which works for building most of the stack, however some spack packages don’t build hence we will try to build remainder of packages with mvapich2-gdr.
We have modulefiles for mvapich2 on shared filesystem, one must add the directory to MODULEPATH since we don’t expose these modules to all users.

```
$ module use /global/cfs/cdirs/m3896/shared/modulefiles
$ module av

mvapich2/3.0a
```
Common Errors during spack builds

● We have noticed a few spack packages fail to build with cray-mpich due to proper support with cray compilers.

● Some packages expect MPI wrappers such as mpicc but on Cray we expect to use cray wrappers `cc, CC, ftn` when invoking MPI programs

```bash
# configure: error: Failed to find C MPI Wrapper. see https://cdash.spack.io/buildSummary.php?buildid=104940
# - rempi@1.1.0

#- mpifileutils@0.11.1 ~xattr # failed to install libcircle Unable to find suitable MPI Compiler. Try setting MPICC.
```
MVAPICH2 installation on Perlmutter

- MVAPICH2 3.0a supports SLURM and Slingshot 11 network interfaces on both CPU and GPU nodes of Perlmutter
- To use MVAPICH2:
  - module use /global/cfs/cdirs/m3896/shared/modulefiles
  - module load mvapich2/3.0a
  - Use mpicc, mpicxx, mpif90 as compilers
    - mpicc foo.c -o foo
  - Use srun to launch the binary
    - srun -n 4 ./foo
The Extreme-scale Scientific Software Stack (E4S) is installed on Perlmutter using the mvapich2/3.0a module.

For more details regarding this deployment, please refer to E4S documentation page https://e4s.readthedocs.io/en/latest/deployment.html#perlmutter

- module use /global/cfs/cdirs/m3896/shared/modulefiles
- module avail e4s
  e4s/22.05/mvapich2-3.0a ...
- module load e4s/22.05/mvapich2-3.0a
- module avail
sameer@perlmuter:login24:~$ module avail

------ /global/cfs/cdirs/m3896/shared/ParaTools/E4S/22.05/mvapich2-3.0a-slurm/spack/share/spack/lmod/cray-sles15-x86_64/mvapich2/3.0a-es35auw/Core ------
adios/1.13.1 fortlilios/2.0.0 nccmp/1.9.0.1 py-warpx/22.05-dims3
adios/2.8.0-cuda80 globarrays/5.8 ncc/5.0.1 py-warpx/22.05-dims8RZ (D)
adios/2.8.0 (D) hdf5/1.10.7 omega-h/9.34.1 scr/3.0rc2
amrex/22.05 heffe/2.2.0-cuda80 openpmd-api/0.14.4 sles/2021.05.02-cuda80-openmp
arborx/1.2 heffe/2.2.0 (D) papyrus/1.0.2 sles/2021.05.02-openmp (D)
axom/0.6.1-openmp hpx/1.7.1-cuda80 parsec/3.6.2012 slepc/3.17.1-cuda80
butterflypack/2.1.1 hpx/1.7.1 petsc/3.17.1-cuda80 (D)
cabana/0.4.0 hypre/2.24.0 petsc/3.17.1 (D)
caliper/2.7.0-cuda80 kokkos-kernels/3.6.00-cuda80 precice/2.4.0 strumpack/6.3.1-openmp
     (D) lammps/20220107-openmp sundials/6.2.0
caliper/2.7.0 (D) pumi/2.2.7 tasmanian/7.7-openmp
darshan-runtime/3.3.1 liquio/1.3.1 tau/2.31.1-cuda
datatransfertkit/3.1-rc3 mercury/2.1.0 py-cinemasci/1.7.0

dyninst/12.1.0-openmp metall/0.20 py-libensemble/0.9.1 veloci/1.5
faodel/1.2108.1 mfm/4.4.0 py-petsc4py/3.17.1
odus/22.05-dims2

------ /global/cfs/cdirs/m3896/shared/ParaTools/E4S/22.05/mvapich2-3.0a-slurm/spack/share/spack/lmod/cray-sles15-x86_64/openmpi/4.1.3-gw3a4bv/Core ------
gptune/3.0.0

------ /global/cfs/cdirs/m3896/shared/ParaTools/E4S/22.05/mvapich2-3.0a-slurm/spack/share/spack/lmod/cray-sles15-x86_64/Core ------

aml/0.1.0 flux-core/0.38.0-cuda legion/21.03.0-cuda80-cuda plasma/21.8.29
archer/2.0.0 flux-core/0.38.0 (D) legion/21.03.0 (D) py-jupyterhub/1.4.1
argbots/1.1.0 gasnet/2022.3.0 magma/2.6.2-cuda80 qthreads/1.16
bolt/2.0 ginkgo/1.4.0-cuda80-openmp mpark/variant/1.4.0 raja/0.14.0-cuda80-openmp
chai/2.4.0 ginkgo/1.4.0-openmp (D) mvapich2/3.0a (D) superlu/5.3.0
charliecloud/5.26 gmp/6.2.1 nrm/0.1.0 swig/4.0.2-fortran
cmake/3.23.1 (D) gotcha/1.0.3 nvhpc/22.3 umap/2.1.0
darshan-util/3.3.1 kokkos-kernels/3.6.00-openmp (D) papi/6.6.0.1-cuda zfp/0.5.5-cuda80
flit/2.1.0 kokkos/3.6.00-openmp pdt/3.25.1
E4S products built using MVAPICH2
E4S products built using E4S (contd.)
● We have developed a test to run the OSU benchmark for the mvapich2 installation.

● Test are run via **buildtest**, an HPC testing framework that will automate build and execution of test.

● Test results are pushed to CDASH via **buildtest**.

```bash
buildspec:
  evapich2_mpi_memb_tests:
    type: script
    executor: perlmutter.slurm.debug
    description: Run OSU microbenchmarks
tags: [gpu, mpi]
    script: |-
      cd Installations/mvapich2/ &&
      export MVAPICH2_MEMB=1 &&
      mpirun -n 8 -t OpenMpiPthreaded -mca btl vader-via-socket -mca orte_recursive_envs 1 -x MVAPICH2_MEMB=1 -x MVAPICH2_MEMB_DIR= -x MVAPICH2_MEMB_MPART= -x MVAPICH2_MEMB_MPI implementations/mvapich2/pthreads/lenormand
```

<table>
<thead>
<tr>
<th>Size</th>
<th>Bandwidth (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.37</td>
</tr>
<tr>
<td>2</td>
<td>4.80</td>
</tr>
<tr>
<td>4</td>
<td>9.05</td>
</tr>
<tr>
<td>8</td>
<td>19.13</td>
</tr>
<tr>
<td>16</td>
<td>38.54</td>
</tr>
<tr>
<td>32</td>
<td>74.10</td>
</tr>
<tr>
<td>64</td>
<td>154.04</td>
</tr>
<tr>
<td>128</td>
<td>386.16</td>
</tr>
<tr>
<td>256</td>
<td>556.54</td>
</tr>
<tr>
<td>512</td>
<td>1002.87</td>
</tr>
<tr>
<td>1024</td>
<td>2189.58</td>
</tr>
<tr>
<td>2048</td>
<td>4368.47</td>
</tr>
<tr>
<td>4096</td>
<td>8648.64</td>
</tr>
<tr>
<td>8192</td>
<td>15522.25</td>
</tr>
<tr>
<td>16384</td>
<td>30838.97</td>
</tr>
<tr>
<td>32768</td>
<td>61686.61</td>
</tr>
<tr>
<td>65536</td>
<td>123373.22</td>
</tr>
<tr>
<td>131072</td>
<td>246746.20</td>
</tr>
<tr>
<td>524288</td>
<td>390681.84</td>
</tr>
<tr>
<td>1048576</td>
<td>140963.33</td>
</tr>
<tr>
<td>2097152</td>
<td>281925.20</td>
</tr>
<tr>
<td>4194304</td>
<td>563850.83</td>
</tr>
</tbody>
</table>
Closing Remarks

- NERSC will provide support for the vendor MPI (cray-mpich), on the contrary NERSC users will experiment with any MPI provider they have access to including OpenMPI, MVAPICH2, MPICH which can be troubling since we don’t have proper support for any other MPI
- MVAPICH2-GDR is promising, considering it provides support with Slurm and Slingshot 11
- With sufficient testing, we can provide user with documentation on how to use MVAPICH2-GDR on Perlmutter
  - Performance comparison between cray-mpich and MVAPICH2-GDR
- MVAPICH2-GDR can provide user to experiment with spack build instead of being locked into cray-mpich, we have seen several spack packages fail to recognize MPI wrappers.