OPENFABRICS ALLIANCE

2021 OFA Virtual Workshop

DIRECT PSM2 SUPPORT FOR NCCL

Brendan Cunningham, Software Engineer

CORNELIS™ NETWORKS

MOTIVATION

- **NCCL is a library from NVIDIA for collectives-heavy GPU programs**
  - E.g., Artificial Intelligence (AI) / Machine Language (ML)
  - Programmers can work in GPU code; NCCL does the internode communication.
  - This means NCCL has to have transport methods available to it.
  - NCCL 2.8.4 has sockets, shared memory, and Verbs transports.
  - If a transport supports GPUDirect (device-device DMA), NCCL will pass DMA-able pointer to transport instead of bounce buffer.
- **Customer wanted to use NCCL with Cornelis™ Omni-Path™ Architecture (OPA).**
  - OPA can do GPUDirect via PSM2.
  - OPA cannot do GPUDirect via Verbs or OFI-PSM2.
    - NCCL will still run over OPA Verbs but performance won't be as good as it could be.
  - GPUDirect benefits OPA performance going from/to GPU buffers, especially for smaller messages.
  - From OSU benchmarks, we have a good idea of how much performance to expect for point-to-point messages on OPA with GPUDirect.
- **Given all this, making NCCL PSM2-aware was the logical choice for our situation.**

MAPPING NCCL NET PLUGIN API ONTO PSM2

NCCL NET PLUGIN API – SOURCE AND LOADING

- **NCCL transport API**
  - `nccl/src/include/nccl_net.h`
    - Defined in C
  - Internal to NCCL; need a nccl repo clone
  - NCCL will load plugin from `libnccl-net.so` if `.so` can be found at runtime.
  - Plugin can provide point-to-point (`ncclNet_t`) and/or collectives (`ncclCollNet_t`) implementation.
    - NCCL favors collectives implementation if available and initializes successfully but can implement collectives ops using point-to-point.
  - NCCL calls plugin `.getProperties()` to get info about network transport like maximum number of communicators, if transport supports GPUDirect, speed.
- **Since PSM2-NCCL provides point-to-point implementation, the following slides will discuss point-to-point operations.**

NCCL NET PLUGIN API – KEY OBJECTS AND METHODS

Important objects are communicator objects and request objects.

- A communicator object is not to be confused with the formal concept of NCCL Communicator (https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/usage/communicators.html#communicator-label). It is analogous to MPI communicator.
- This presentation will use the term “comm objects” or “comm” to avoid confusion with the NCCL Communicator proper.
- The plugin returns comm and request type-objects to NCCL as ‘void *’; types are opaque to NCCL.
- Implementation is responsible for the lifetime/cleanup of these objects.

▪ Comm objects

- Two flavors: send and receive
- Represent endpoints for sending to or receiving from remote ranks
- Each NCCL rank needs both send and receive per remote rank for bidirectional communication.

▪ Request objects

- Handles for in-progress send and receive

▪ Key methods for plugin to implement

- `.isend(), .irecv()` – Non-blocking send and receive; return request object used to test for completion
- `.test()` – Test request for completion
- `.listen(), .connect(), .accept()` – Connection establishment

TESTING AND RUNNING

- **nccl-tests repo from NVIDIA**
  - Location: https://github.com/NVIDIA/nccl-tests
  - A collection of six collectives programs that provide basic benchmarking and correctness checks
- **Use OpenMPI to start the host ranks on each node**
- **Can run as many NCCL ranks per node as there are GPUs**
  - Can run 1:1 or 1:* host:NCCL ranks
- **We wrote a test module to run all of the nccl-tests test-programs with different PSM2 and PSM2 NCCL settings.**
  - Also extracts performance data from test cases for comparison

PSM2-NCCL IMPLEMENTATION

- **Straightforward to map NCCL Net API onto PSM2**
  - `.isend()` → `psm2_mq_isend()`
  - `.irecv()` → `psm2_mq_irecv()`
  - `.test()` → `psm2_mq_test()`
- **Since PSM2 is messaging, not RDMA-oriented, no need to implement `.regMr()`, `.deregMr()`.**
- **Comm object type stores PSM2 endpoint (EP), matched queue (MQ), tag to use when sending message to remote endpoint.**

PSM2-NCCL IMPLEMENTATION

- **Problems encountered**
  - PSM2 assumed that CUDA context was always set before `psm2_init()` was called.
    - Solved by lazy initialization pull-request from hanjo (https://github.com/cornelisnetworks/opa-psm2/pull/46).
    - Special build of libpsm2 is required to run PSM2-NCCL.
  - PSM2 requires the user to call one of the PSM2 progress functions to ensure message progress.
    - Solution was to put `psm2_poll()` in `.test()` implementation.
  - The initial release uses one PSM2 EP per comm object. This made the code simple but limited job scaling.
    - Tried sharing one EP for all comms in a host process. This solved the scaling problem but did not perform as well.
    - Contention between PSM2 receive thread and application main thread hurt performance.
    - Disabling the receive thread (`PSM2_RCVTHREAD=0`) solved this issue.
    - At time of presentation, shared-EP code is still in development but should be out soon.
  - GPU-page-pinning failure when `PSM2_GDRCOPY=1` causes job to fail.
    - Works with `PSM2_GDRCOPY=0`.

PSM2-NCCL PERFORMANCE

all_reduce_perf, OPA Verbs

```
# nThread 1 nGpus 1 minBytes 8192 maxBytes 134217728 step: 2(factor) warmup iters: 5 iters: 20 validation: 1
#
# Using devices
#   Rank  0 Pid  14829 on hds1fnaf211 device  0 [0x04] Tesla P100-PCIE-16GB
#   Rank  1 Pid  14352 on hds1fnaf251 device  0 [0x04] Tesla P100-PCIE-16GB
#
#                                                    out-of-place                       in-place
#       size         count    type   redop     time   algbw   busbw  error     time   algbw   busbw  error
#        (B)    (elements)                     (us)  (GB/s)  (GB/s)            (us)  (GB/s)  (GB/s)
        8192          2048   float     sum    36.11    0.23    0.23  0e+00    35.42    0.23    0.23  0e+00
       16384          4096   float     sum    42.53    0.39    0.39  0e+00    39.41    0.42    0.42  0e+00
       32768          8192   float     sum    63.51    0.52    0.52  0e+00    64.54    0.51    0.51  0e+00
       65536         16384   float     sum    88.85    0.74    0.74  0e+00    90.39    0.73    0.73  0e+00
      131072         32768   float     sum    127.1    1.03    1.03  0e+00    128.4    1.02    1.02  0e+00
      262144         65536   float     sum    148.1    1.77    1.77  0e+00    152.2    1.72    1.72  0e+00
      524288        131072   float     sum    180.2    2.91    2.91  0e+00    177.9    2.95    2.95  0e+00
     1048576        262144   float     sum    213.9    4.90    4.90  0e+00    212.8    4.93    4.93  0e+00
     2097152        524288   float     sum    353.7    5.93    5.93  0e+00    344.8    6.08    6.08  0e+00
     4194304       1048576   float     sum    624.6    6.72    6.72  0e+00    642.7    6.53    6.53  0e+00
     8388608       2097152   float     sum   1252.0    6.70    6.70  0e+00   1254.9    6.68    6.68  0e+00
    16777216       4194304   float     sum   2474.7    6.78    6.78  0e+00   2522.2    6.65    6.65  0e+00
    33554432       8388608   float     sum   4829.2    6.95    6.95  0e+00   4785.7    7.01    7.01  0e+00
    67108864      16777216   float     sum   9524.8    7.05    7.05  0e+00   9740.1    6.89    6.89  0e+00
   134217728      33554432   float     sum    18727    7.17    7.17  0e+00    18905    7.10    7.10  0e+00
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 3.97361
#
```

all_reduce_perf, PSM2-NCCL, non-shared-EP, GPUDirect

```
# nThread 1 nGpus 1 minBytes 8192 maxBytes 134217728 step: 2(factor) warmup iters: 5 iters: 20 validation: 1
#
# Using devices
#   Rank  0 Pid  12476 on hds1fnaf211 device  0 [0x04] Tesla P100-PCIE-16GB
#   Rank  1 Pid  11303 on hds1fnaf251 device  0 [0x04] Tesla P100-PCIE-16GB
#
#                                                    out-of-place                       in-place
#       size         count    type   redop     time   algbw   busbw  error     time   algbw   busbw  error
#        (B)    (elements)                     (us)  (GB/s)  (GB/s)            (us)  (GB/s)  (GB/s)
        8192          2048   float     sum    95.65    0.09    0.09  0e+00    76.54    0.11    0.11  0e+00
       16384          4096   float     sum    134.6    0.12    0.12  0e+00    133.5    0.12    0.12  0e+00
       32768          8192   float     sum    200.4    0.16    0.16  0e+00    207.4    0.16    0.16  0e+00
       65536         16384   float     sum    454.4    0.14    0.14  0e+00    432.4    0.15    0.15  0e+00
      131072         32768   float     sum    110.1    1.19    1.19  0e+00    107.2    1.22    1.22  0e+00
      262144         65536   float     sum    181.0    1.45    1.45  0e+00    177.6    1.48    1.48  0e+00
      524288        131072   float     sum    256.0    2.05    2.05  0e+00    255.1    2.06    2.06  0e+00
     1048576        262144   float     sum    288.6    3.63    3.63  0e+00    283.4    3.70    3.70  0e+00
     2097152        524288   float     sum    364.6    5.75    5.75  0e+00    370.0    5.67    5.67  0e+00
     4194304       1048576   float     sum    657.7    6.38    6.38  0e+00    596.3    7.03    7.03  0e+00
     8388608       2097152   float     sum   1252.2    6.70    6.70  0e+00   1141.1    7.35    7.35  0e+00
    16777216       4194304   float     sum   2089.2    8.03    8.03  0e+00   2074.4    8.09    8.09  0e+00
    33554432       8388608   float     sum   3973.8    8.44    8.44  0e+00   3962.3    8.47    8.47  0e+00
    67108864      16777216   float     sum   7745.3    8.66    8.66  0e+00   7755.1    8.65    8.65  0e+00
   134217728      33554432   float     sum    15303    8.77    8.77  0e+00    15381    8.73    8.73  0e+00
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 4.1518
#
```

PSM2-NCCL PERFORMANCE

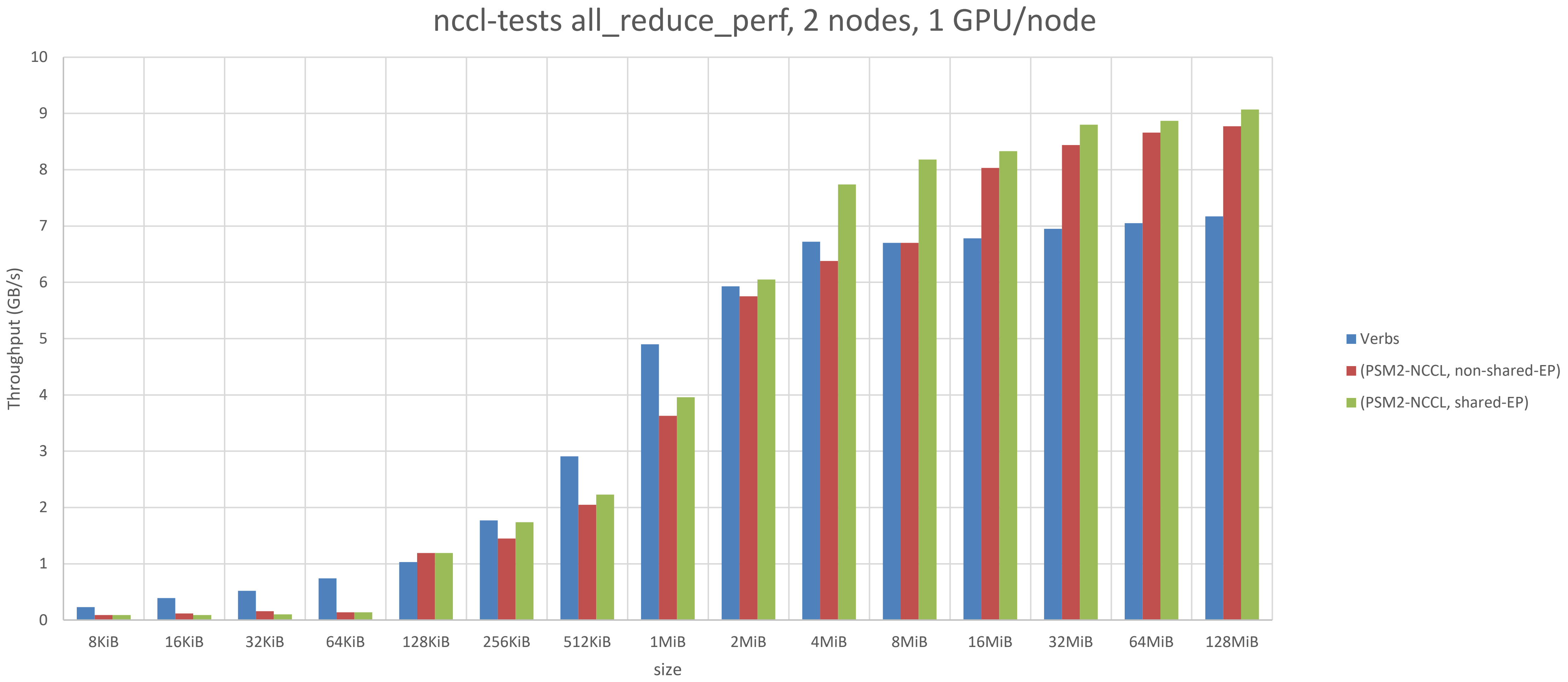
all_reduce_perf, OPA Verbs

```
# nThread 1 nGpus 1 minBytes 8192 maxBytes 134217728 step: 2(factor) warmup iters: 5 iters: 20 validation: 1
#
# Using devices
#   Rank  0 Pid  14829 on hds1fnaf211 device  0 [0x04] Tesla P100-PCIE-16GB
#   Rank  1 Pid  14352 on hds1fnaf251 device  0 [0x04] Tesla P100-PCIE-16GB
#
#                                                    out-of-place                       in-place
#       size         count    type   redop     time   algbw   busbw  error     time   algbw   busbw  error
#        (B)    (elements)                     (us)  (GB/s)  (GB/s)            (us)  (GB/s)  (GB/s)
        8192          2048   float     sum    36.11    0.23    0.23  0e+00    35.42    0.23    0.23  0e+00
       16384          4096   float     sum    42.53    0.39    0.39  0e+00    39.41    0.42    0.42  0e+00
       32768          8192   float     sum    63.51    0.52    0.52  0e+00    64.54    0.51    0.51  0e+00
       65536         16384   float     sum    88.85    0.74    0.74  0e+00    90.39    0.73    0.73  0e+00
      131072         32768   float     sum    127.1    1.03    1.03  0e+00    128.4    1.02    1.02  0e+00
      262144         65536   float     sum    148.1    1.77    1.77  0e+00    152.2    1.72    1.72  0e+00
      524288        131072   float     sum    180.2    2.91    2.91  0e+00    177.9    2.95    2.95  0e+00
     1048576        262144   float     sum    213.9    4.90    4.90  0e+00    212.8    4.93    4.93  0e+00
     2097152        524288   float     sum    353.7    5.93    5.93  0e+00    344.8    6.08    6.08  0e+00
     4194304       1048576   float     sum    624.6    6.72    6.72  0e+00    642.7    6.53    6.53  0e+00
     8388608       2097152   float     sum   1252.0    6.70    6.70  0e+00   1254.9    6.68    6.68  0e+00
    16777216       4194304   float     sum   2474.7    6.78    6.78  0e+00   2522.2    6.65    6.65  0e+00
    33554432       8388608   float     sum   4829.2    6.95    6.95  0e+00   4785.7    7.01    7.01  0e+00
    67108864      16777216   float     sum   9524.8    7.05    7.05  0e+00   9740.1    6.89    6.89  0e+00
   134217728      33554432   float     sum    18727    7.17    7.17  0e+00    18905    7.10    7.10  0e+00
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 3.97361
#
```

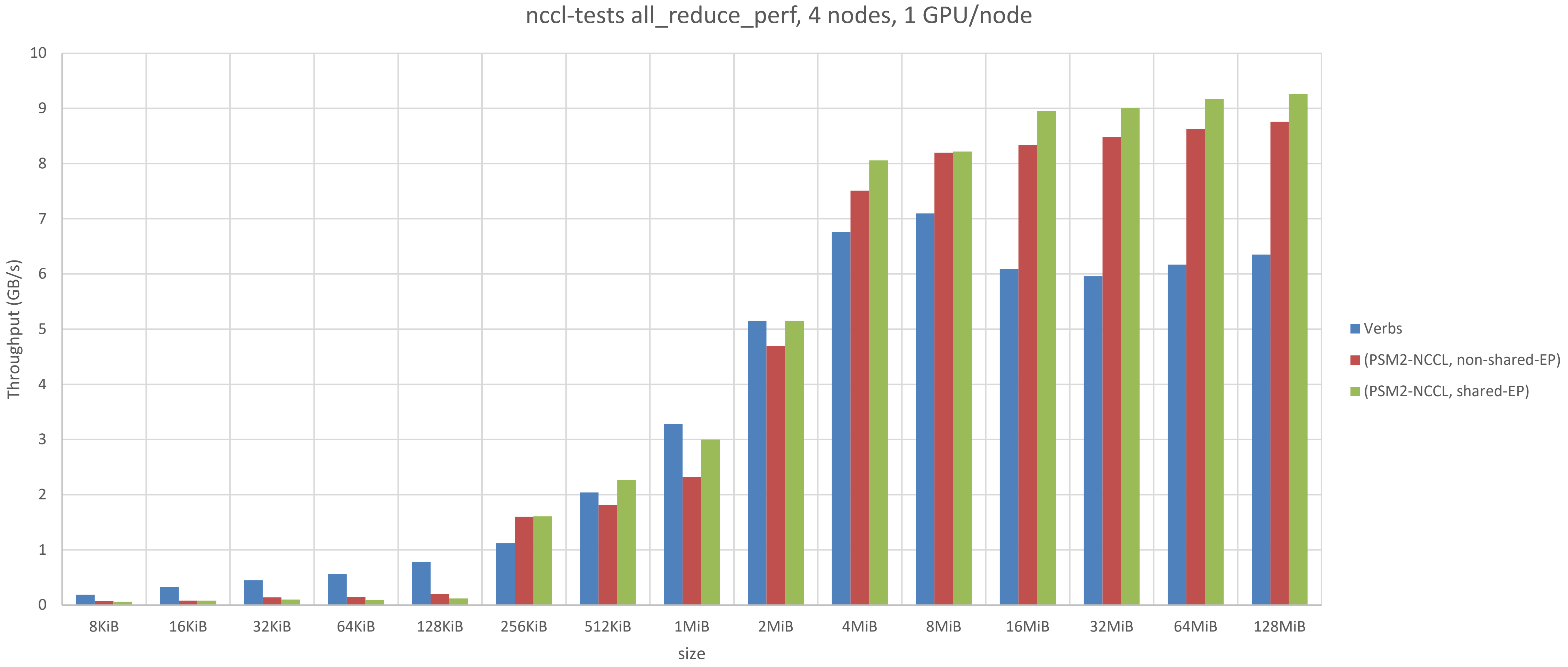
all_reduce_perf, PSM2-NCCL shared-EP, GPUDirect

```
# nThread 1 nGpus 1 minBytes 8192 maxBytes 134217728 step: 2(factor) warmup iters: 5 iters: 20 validation: 1
#
# Using devices
#   Rank  0 Pid  11100 on hds1fnaf211 device  0 [0x04] Tesla P100-PCIE-16GB
#   Rank  1 Pid   9584 on hds1fnaf251 device  0 [0x04] Tesla P100-PCIE-16GB
#
#                                                    out-of-place                       in-place
#       size         count    type   redop     time   algbw   busbw  error     time   algbw   busbw  error
#        (B)    (elements)                     (us)  (GB/s)  (GB/s)            (us)  (GB/s)  (GB/s)
        8192          2048   float     sum    95.32    0.09    0.09  0e+00    90.30    0.09    0.09  0e+00
       16384          4096   float     sum    176.1    0.09    0.09  0e+00    174.6    0.09    0.09  0e+00
       32768          8192   float     sum    326.4    0.10    0.10  0e+00    306.1    0.11    0.11  0e+00
       65536         16384   float     sum    479.9    0.14    0.14  0e+00    477.6    0.14    0.14  0e+00
      131072         32768   float     sum    109.8    1.19    1.19  0e+00    120.1    1.09    1.09  0e+00
      262144         65536   float     sum    150.9    1.74    1.74  0e+00    145.8    1.80    1.80  0e+00
      524288        131072   float     sum    234.6    2.23    2.23  0e+00    230.5    2.27    2.27  0e+00
     1048576        262144   float     sum    264.9    3.96    3.96  0e+00    265.7    3.95    3.95  0e+00
     2097152        524288   float     sum    346.6    6.05    6.05  0e+00    342.7    6.12    6.12  0e+00
     4194304       1048576   float     sum    542.1    7.74    7.74  0e+00    550.4    7.62    7.62  0e+00
     8388608       2097152   float     sum   1025.1    8.18    8.18  0e+00   1032.9    8.12    8.12  0e+00
    16777216       4194304   float     sum   2012.9    8.33    8.33  0e+00   2009.1    8.35    8.35  0e+00
    33554432       8388608   float     sum   3815.1    8.80    8.80  0e+00   3811.3    8.80    8.80  0e+00
    67108864      16777216   float     sum   7565.5    8.87    8.87  0e+00   7457.3    9.00    9.00  0e+00
   134217728      33554432   float     sum    14805    9.07    9.07  0e+00    14732    9.11    9.11  0e+00
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 4.44138
#
```

PSM2-NCCL PERFORMANCE



PSM2-NCCL PERFORMANCE



PSM2-NCCL PERFORMANCE

- **In our 2-node tests at size=128 MiB, PSM2-NCCL outperformed OPA Verbs by 22% for non-shared-EP code and 26% for shared-EP code.**
- **In our 4-node tests at size=128 MiB, PSM2-NCCL outperformed OPA Verbs by 38% for non-shared-EP code and 46% for shared-EP code.**
- **But, Verbs generally performed better below 1 MiB.**
  - PSM2_GDRCOPY code is meant to benefit small GPUDirect sends and receives. PSM2_GDRCOPY workaround may hurt small data set performance.

CONCLUSIONS AND LINKS

- **Conclusions**
  - The PSM2-NCCL plugin is a simple way for NCCL to take advantage of GPUDirect on OPA.
  - However, in doing so, NCCL presented new use cases for us to consider.
  - Initial performance is good but room for improvement with small data set sizes.
- **Future plans**
  - Fix bugs.
  - Improve small data set performance.
  - Test larger jobs.
- **Thanks**
  - To my colleague Marisa Roman for taking on the shared-EP performance problem.
  - To Jonas Hahnfeld (hanjo) for opa-psm2 PR #46.
- **Links**
  - PSM2-NCCL plugin source - https://github.com/cornelisnetworks/psm2-nccl
  - PSM2 for PSM2-NCCL source - https://github.com/cornelisnetworks/opa-psm2/tree/PSM2_11.2.NCCL

OPENFABRICS ALLIANCE

2021 OFA Virtual Workshop

THANK YOU

CORNELIS™ NETWORKS