NVSHMEM

Release Notes

RN-08515-001_v2.0.2 | November 2020

Table of Contents

Chapter 1. NVSHMEM Release 2.0.2 EA

This is the NVIDIA® NVSHMEM™ 2.0.2 EA release notes.

Key Features And Enhancements

This NVSHMEM release includes the following key features and enhancements:

- Added the teams and team-based collectives APIs from OpenSHMEM 1.5.
- Added support to use the NVIDIA® Collective Communication Library (NCCL) for optimized NVSHMEM host and on-stream collectives.

  Note: This feature is not yet supported on Power 9 systems.

- Added support for RDMA over Converged Ethernet (RoCE) networks.
- Added support for PMI-2 to enable an NVSHMEM job launch with srun/SLURM.
- Added support for PMIx to enable an NVSHMEM job launch with PMIx-compatible launchers, such as Slurm and Open MPI.
- Uniformly reformatted the perftest benchmark output.
- Added support for the `putmem_signal` and `signal_wait_until` APIs.
- Improved support for single-node environments without InfiniBand.
- Fixed a bug that occurred when large numbers of fetch atomic operations were performed on InfiniBand.
- Improved topology awareness in NIC-to-GPU assignments for DGX A100 systems.

Compatibility

NVSHMEM 2.0.2 EA has been tested with the following:

- The following version of CUDA:
  - 10.2
  - 11.0
  - 11.1
- x86 and Power 9

Limitations

- NVSHMEM with NCCL is not yet supported on Power 9 systems.

Fixed Issues

- Concurrent NVSHMEM collective operations with active sets are not supported.
- Concurrent NVSHMEM memory allocation operations and collective operations are not supported.

  The OpenSHMEM specification has clarified that only memory management routines that operate on `NVSHMEM_TEAM_WORLD`, and no other collectives on that team, are permitted concurrently.

Breaking Changes

- Removed support for active set-based collectives interface in OpenSHMEM.

Known Issues

- NVSHMEM and libraries that use NVSHMEM can only be built as static libraries and not as shared libraries.

  This is because the linking of CUDA device symbols does not work across shared libraries.
- `nvshmem_barrier*`, `nvshmem_quiet`, and `nvshmem_wait_until` only ensure PE-PE ordering and visibility on systems with NVLink and InfiniBand.

  They do not ensure global ordering and visibility.
- Complex types, which are enabled by setting `NVSHMEM_COMPLEX_SUPPORT` at compile time, are not currently supported.
- In some cases, `nvshmem_<typename>_g` over InfiniBand and RoCE has been reported to return stale data.

  We are continuing to investigate this issue. In the meantime, you can use `nvshmem_<typename>_atomic_fetch` as a workaround for `nvshmem_<typename>_g`, but the performance of these options is different.

Chapter 2. NVSHMEM Release 1.1.3

This is the NVIDIA® NVSHMEM™ 1.1.3 release notes.

Key Features And Enhancements

This NVSHMEM release includes the following key features and enhancements:

- Implemented the `nvshmem_<type>_put_signal` API from OpenSHMEM 1.5.
- Added the `nvshmemx_signal_op` API.
- Optimized the implementation of a signal set operation over P2P connected GPUs.
- Optimized the performance of the `nvshmem_fence()` function.
- Optimized the latency of the NVSHMEM atomics API.
- Fixed a bug in the `nvshmem_ptr` API.
- Fixed a bug in the implementation of the host-side strided transfer (iput, iget,and so on) API.
- Fixed a bug in the on-stream reduction for the `long long` datatype.
- Fixed a hang during the nvshmem barrier collective operation.
- Fixed `__device__ nvshmem_quiet()` to also do quiet on IB ops to self.

Compatibility

NVSHMEM 1.1.3 has been tested with the following:

- CUDA 10.1, 10.2, and 11.0
- x86 and PowerPC

Known Issues

- NVSHMEM and libraries that use NVSHMEM can only be built as static libraries, not as shared libraries.

  This is because linking of CUDA device symbols does not work across shared libraries.
- NVSHMEM collective operations with active sets are not supported.
- Concurrent NVSHMEM memory allocation operations and collective operations are not supported.

- `nvshmem_barrier*`, `nvshmem_quiet`, and `nvshmem_wait_until` only ensure PE-PE ordering and visibility on systems with NVLink and InfiniBand.

  They do not ensure global ordering and visibility.

Chapter 3. NVSHMEM Release 1.0.1

This is the NVIDIA® NVSHMEM™ 1.0.1 release notes. This is the first official release of NVSHMEM.

Key Features And Enhancements

This NVSHMEM release includes the following key features and enhancements.

- Combines the memory of multiple GPUs into a partitioned global address space that's accessed through NVSHMEM APIs.
- Includes a low-overhead, in-kernel communication API for use by GPU threads.
- Includes stream-based and CPU-initiated communication APIs.
- Supports peer-to-peer communication using NVIDIA® NVLink® and PCI Express and for GPU clusters using NVIDIA Mellanox® InfiniBand.
- Supports x86 and POWER9 processors.
- Is interoperable with MPI and other OpenSHMEM implementations.

Compatibility

NVSHMEM 1.0.1 has been tested with the following:

- CUDA 10.1, 10.2, and 11.0 RC
- x86 and PowerPC

Known Issues

- NVSHMEM and libraries that use NVSHMEM can only be built as static libraries, not as shared libraries. This is because linking of CUDA device symbols does not work across shared libraries.
- NVSHMEM collective operations with overlapping active sets are known not to work in some scenarios.
- `nvshmem_quiet` only ensures PE-PE visibility and not global visibility of data.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
www.nvidia.com

Trademarks

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
www.nvidia.com