ROCm Support in UCX

Sourav Chakraborty
Sourav.Chakraborty@amd.com
UCF Virtual Workshop 2020

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.” AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Features added/in-progress

- ROCm memory support for perf tools
- Remove dependency on GDRCopy module
- SSE based memcpy for small messages
- Improved agent selection for IPC transfers
- Staged D2D transfers for inter-node
- Added Hardware Tag Matching support

ROCm memory support for perf tools

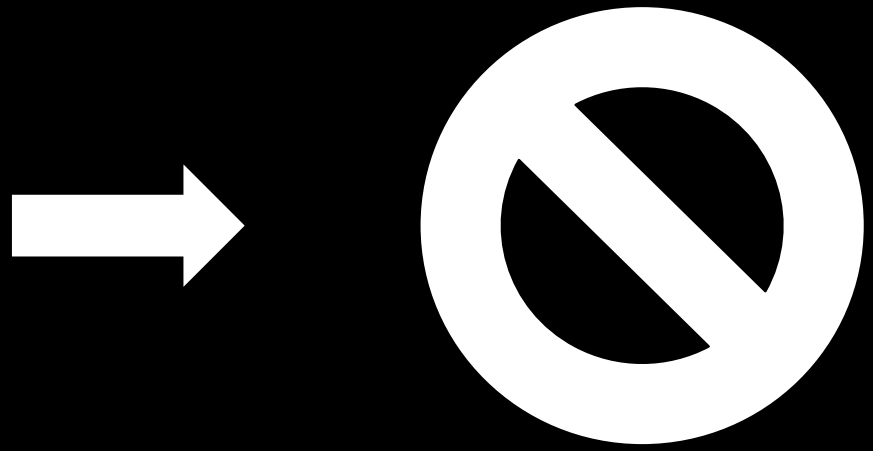
```
$ ./bin/ucx_perftest localhost -f -c 2 -t tag_lat -m rocm
+--------------+-----------------------------+---------------------+-----------------------+
|              |       latency (usec)        |   bandwidth (MB/s)  |  message rate (msg/s) |
+--------------+---------+---------+---------+----------+----------+-----------+-----------+
| # iterations | typical | average | overall |  average |  overall |   average |   overall |
+--------------+---------+---------+---------+----------+----------+-----------+-----------+
       1000000     1.433     7.848    11.025       0.97       0.69      127413       90702
$ ./bin/ucx_perftest localhost -f -c 2 -t tag_lat -m host,rocm
+--------------+-----------------------------+---------------------+-----------------------+
|              |       latency (usec)        |   bandwidth (MB/s)  |  message rate (msg/s) |
+--------------+---------+---------+---------+----------+----------+-----------+-----------+
| # iterations | typical | average | overall |  average |  overall |   average |   overall |
+--------------+---------+---------+---------+----------+----------+-----------+-----------+
       1000000     0.792     2.302     2.338       3.31       3.26      434408       427652
$ ./bin/ucx_perftest localhost -f -c 2 -t tag_lat -m rocm,host
+--------------+-----------------------------+---------------------+-----------------------+
|              |       latency (usec)        |   bandwidth (MB/s)  |  message rate (msg/s) |
+--------------+---------+---------+---------+----------+----------+-----------+-----------+
| # iterations | typical | average | overall |  average |  overall |   average |   overall |
+--------------+---------+---------+---------+----------+----------+-----------+-----------+
       1000000     0.775     2.284     2.292       3.34       3.33      437768       436257
```

- UCX perftest now supports ROCm memory (#4587)
- Added support for measuring D2H/H2D transfers (#4607)

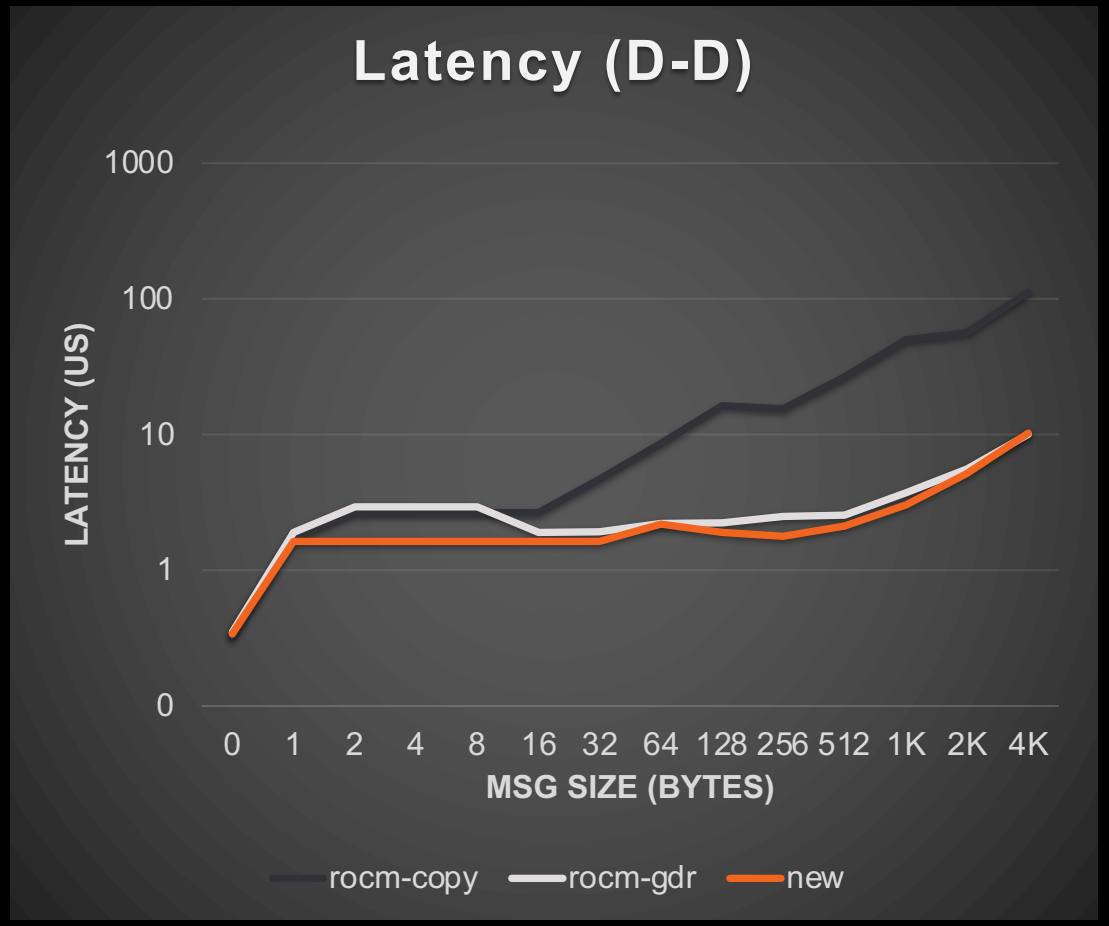
Remove dependency on GDRCopy

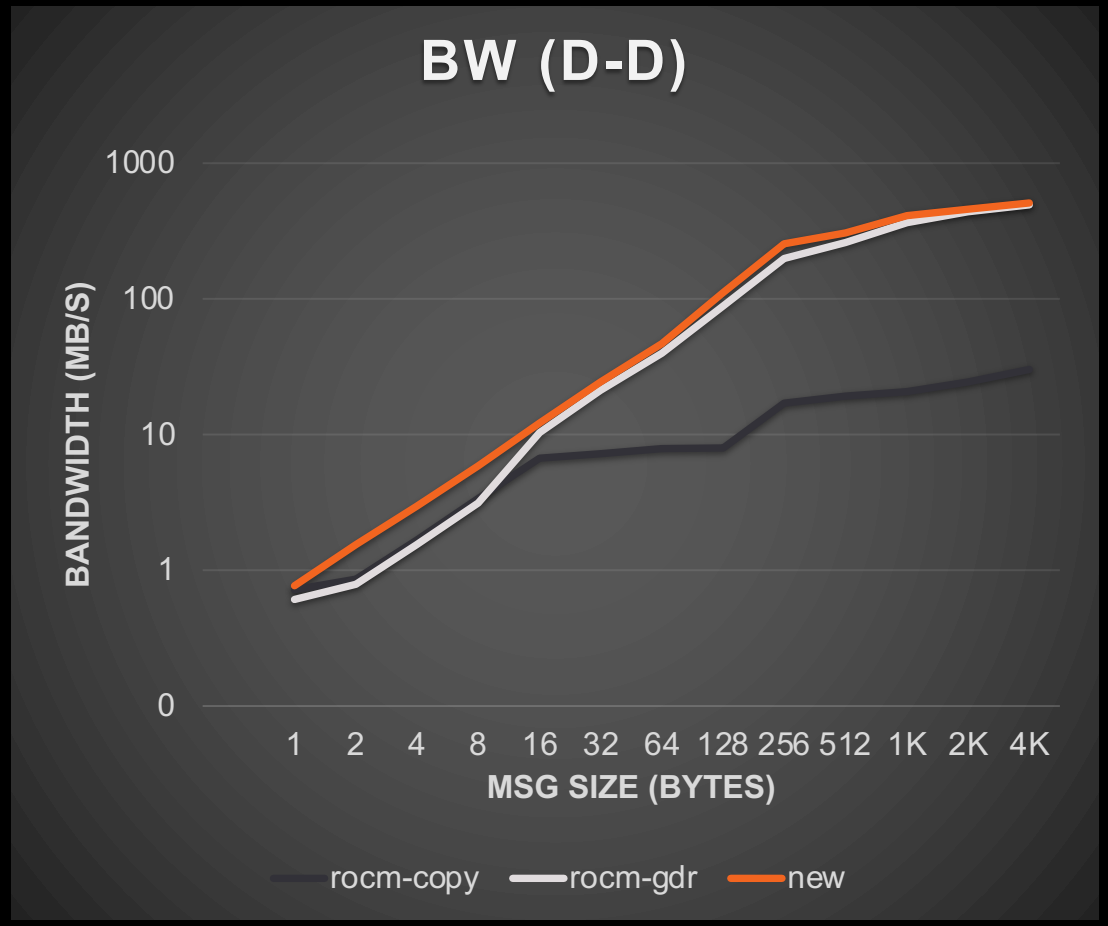
- No need for GDRCopy module to be installed
- Integrated into rocm_copy module (#4532)

```
export GDR_DIR=$INSTALL_DIR/gdr export
LD_LIBRARY_PATH=$GDR_DIR/lib64:$LD_LIBRARY_PATH
git clone https://github.com/NVIDIA/gdrcopy.git
cd gdrcopy
git checkout -b v1.3 tags/v1.3
mkdir -p $GDR_DIR/lib64 $GDR_DIR/include
make PREFIX=$GDR_DIR lib install
configure --with-gdrcopy=$GDR_DIR …
```



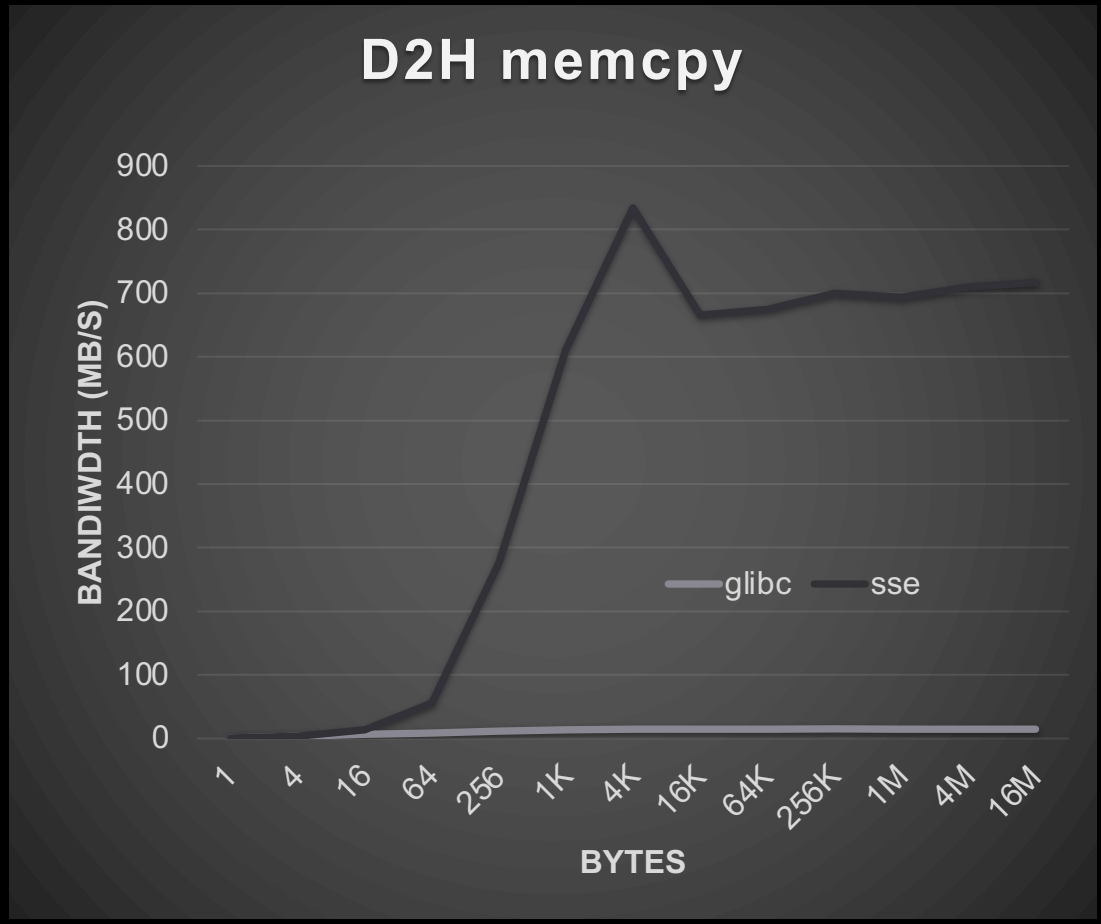
With no degradation in performance!



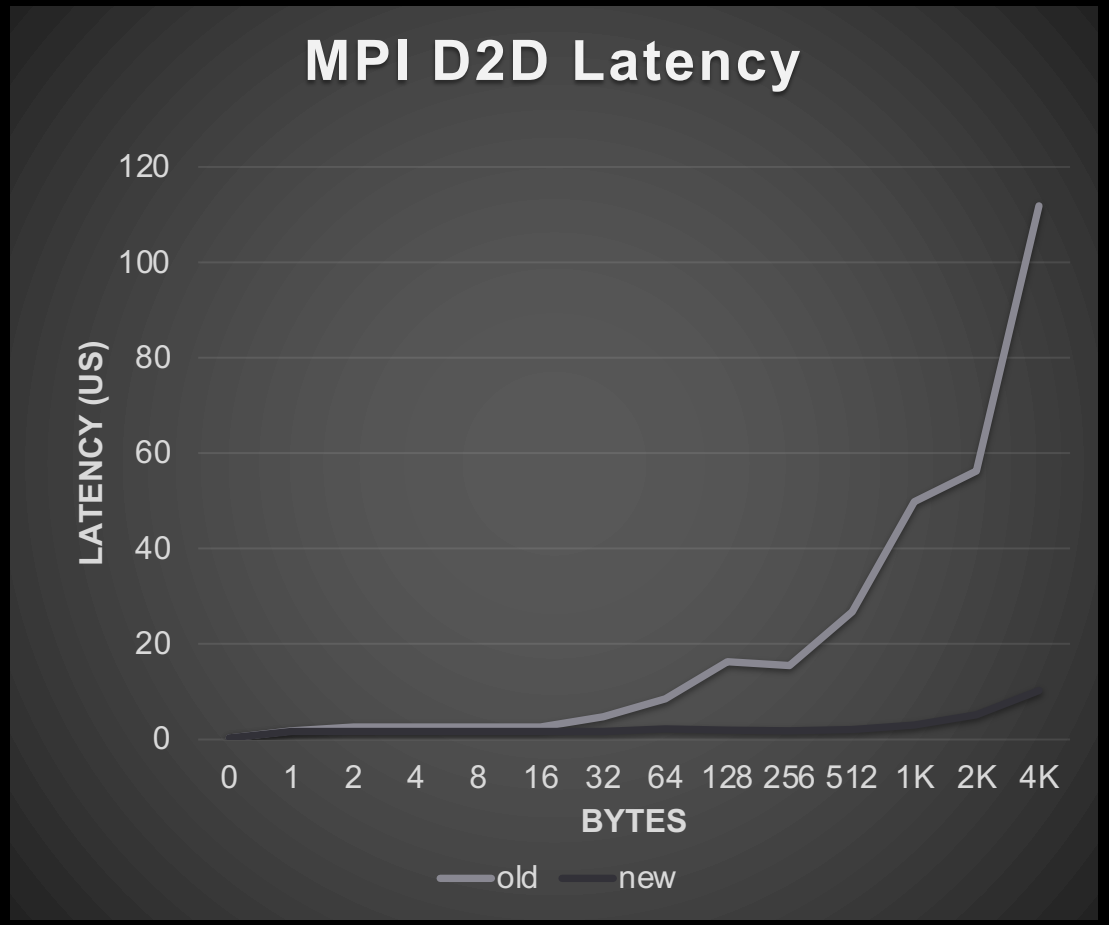


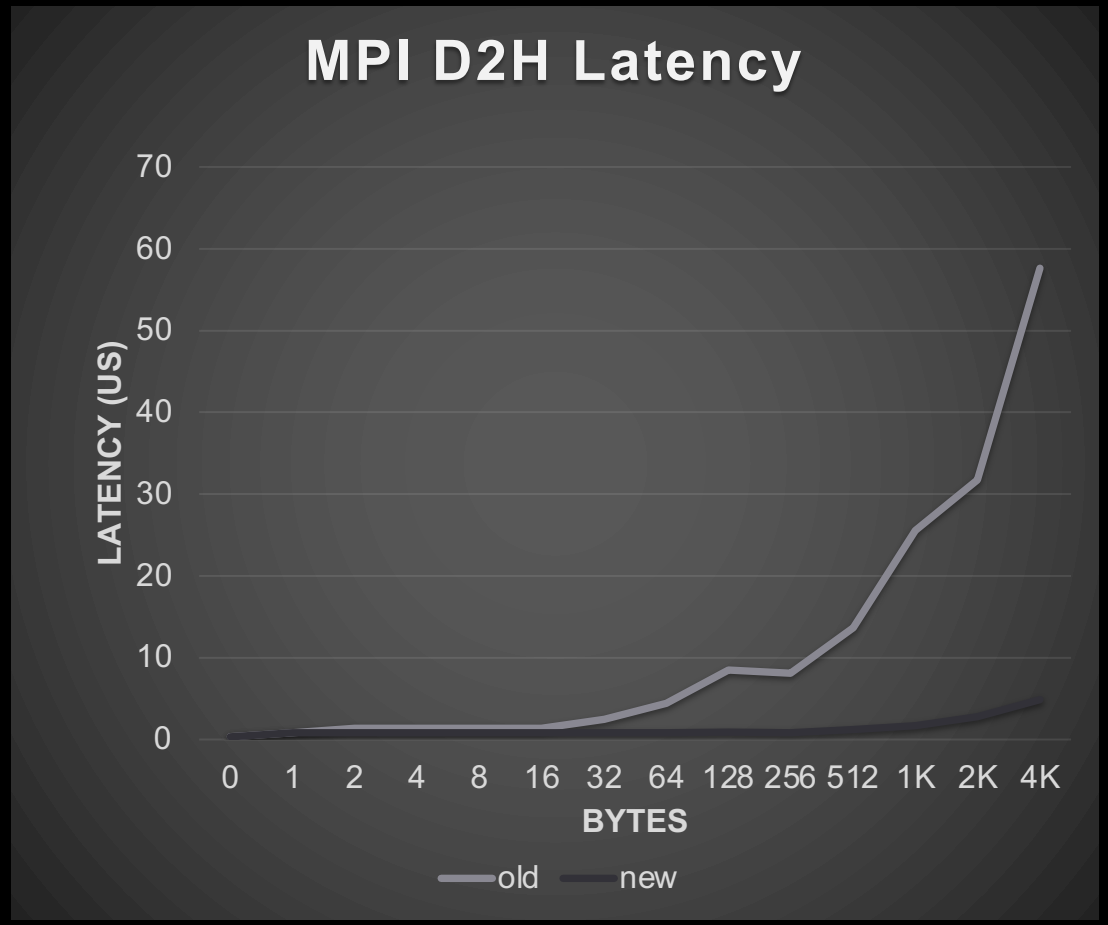
SSE based memcpy for small messages

- glibc memcpy is slow for D2H
- Did exhaustive study of the performance of vectorized copy intrinsics
- Added Non-temporal instruction based memcpy (#4532)
- Up to 50x faster than baseline

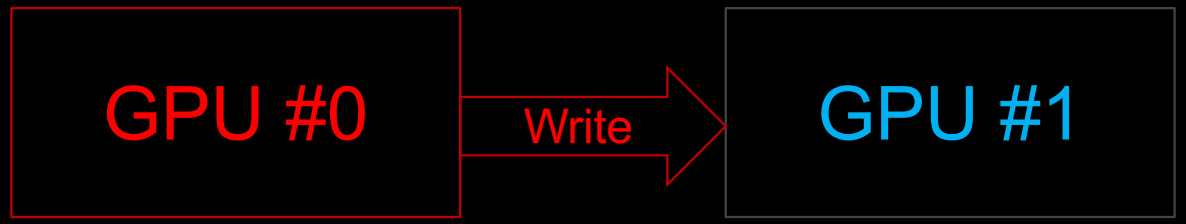


SSE based memcpy for small messages

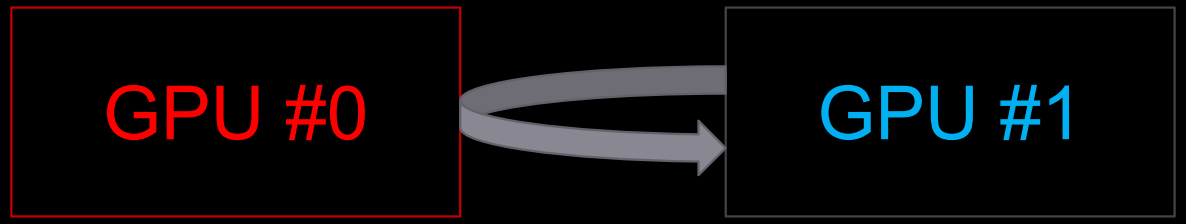




Improved agent selection for IPC transfers

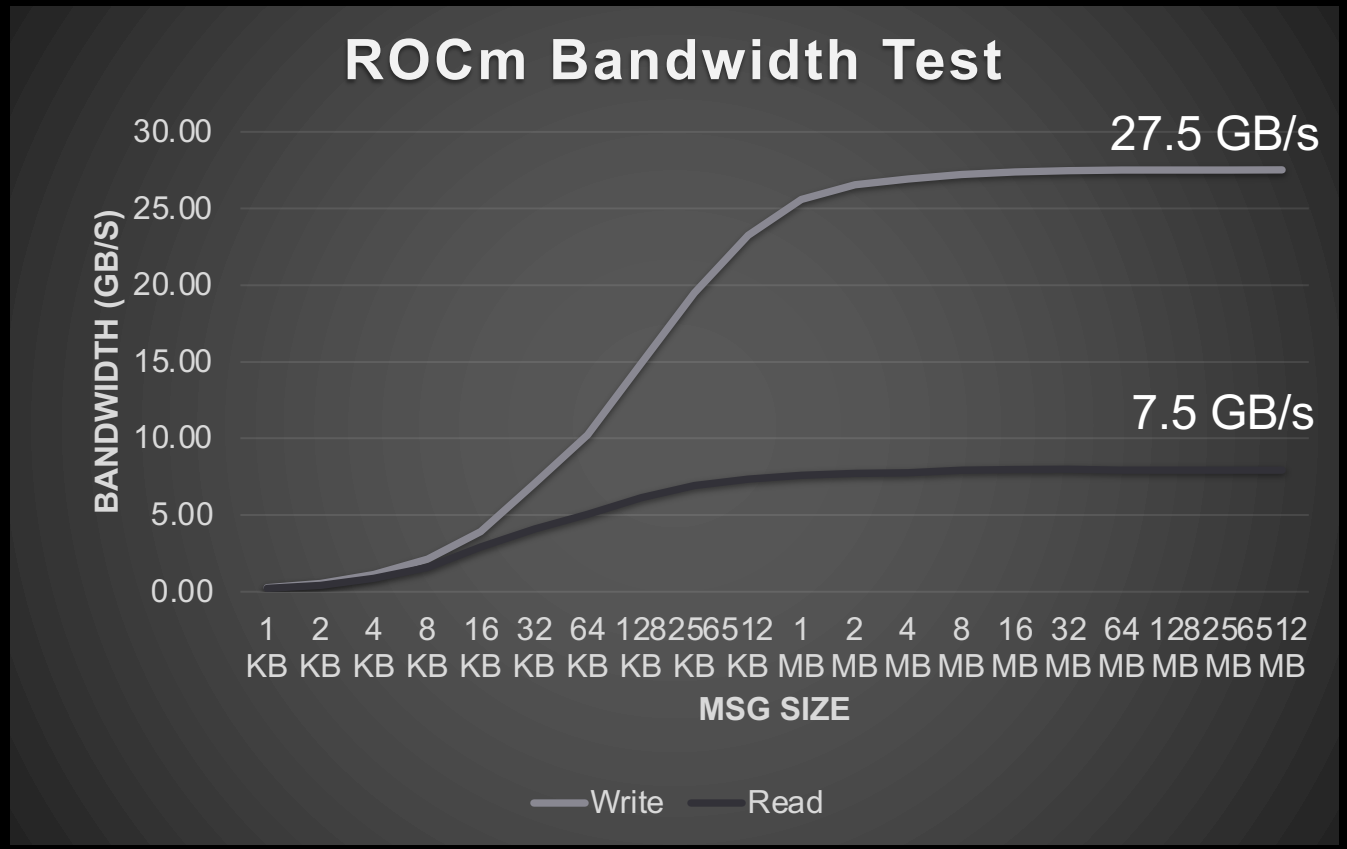


Local Read Remote Write (LRRW)



Remote Read Local Write (LRRW)

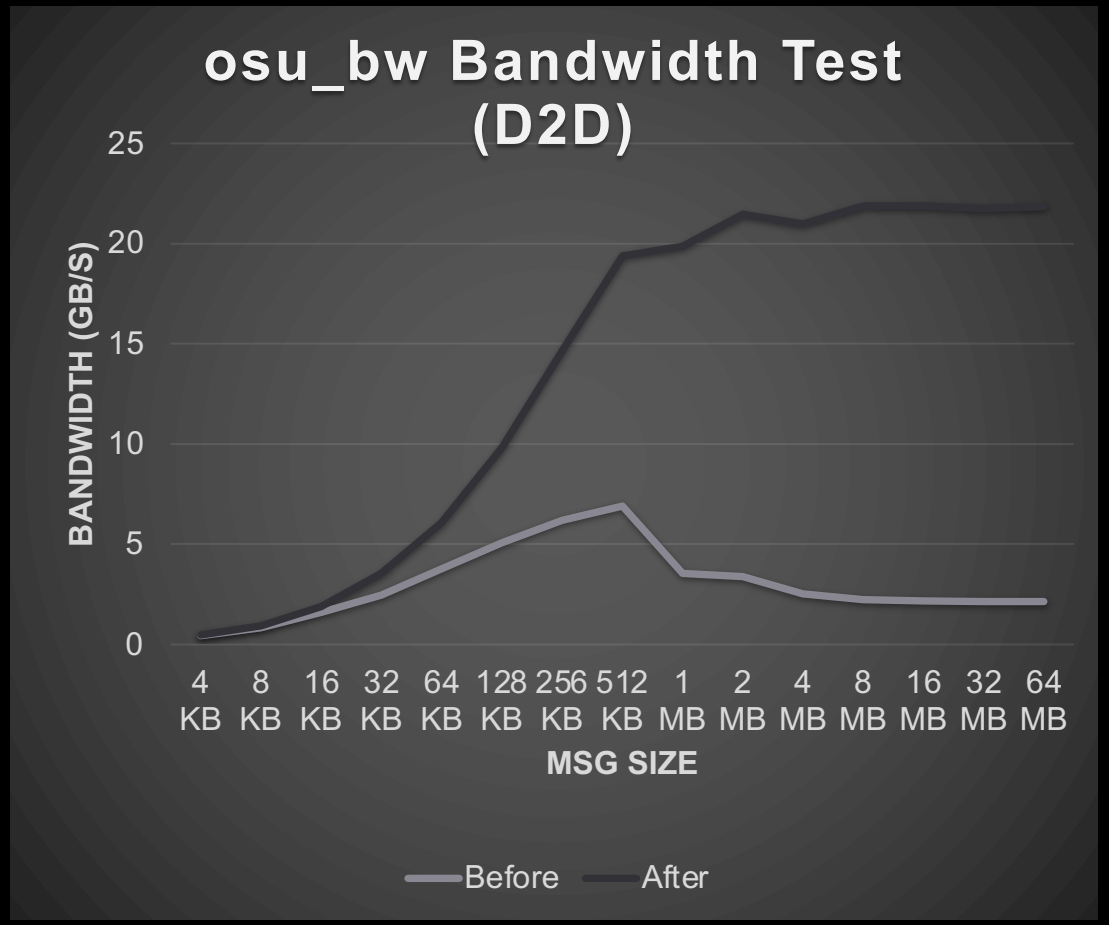
Read / Write bandwidth over PCIe 4.0



- Read performs worse than Write
- get_zcopy performance is worse than put_zcopy
- How to make get_zcopy better?

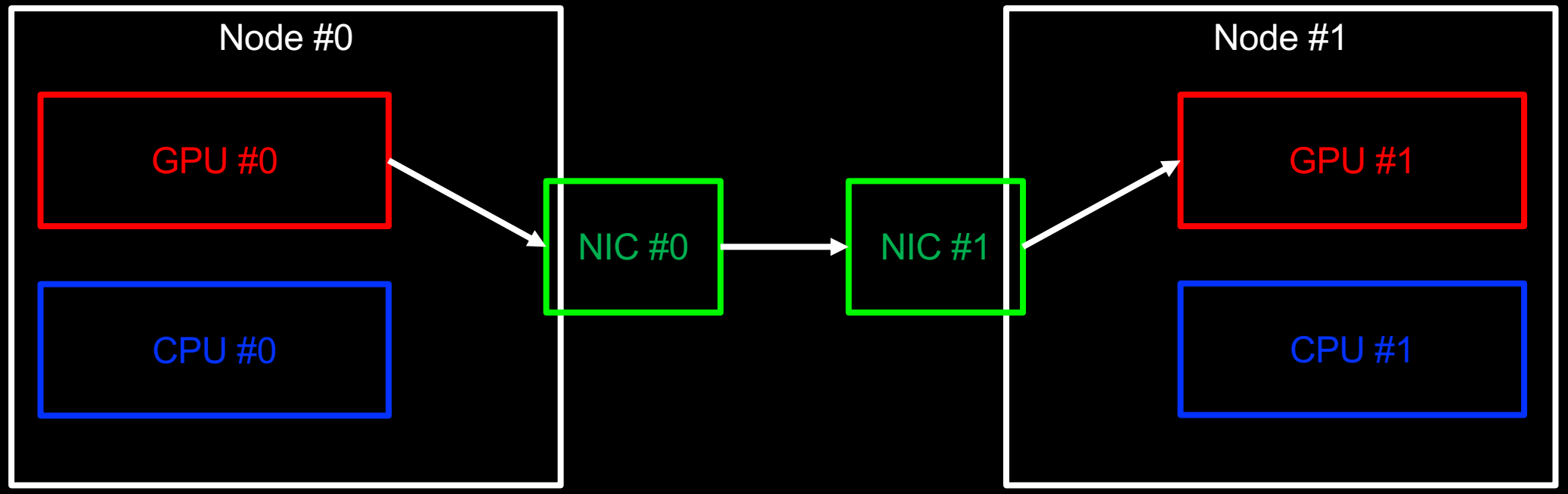
- Only PCIe systems are affected

Improved agent selection for IPC transfers

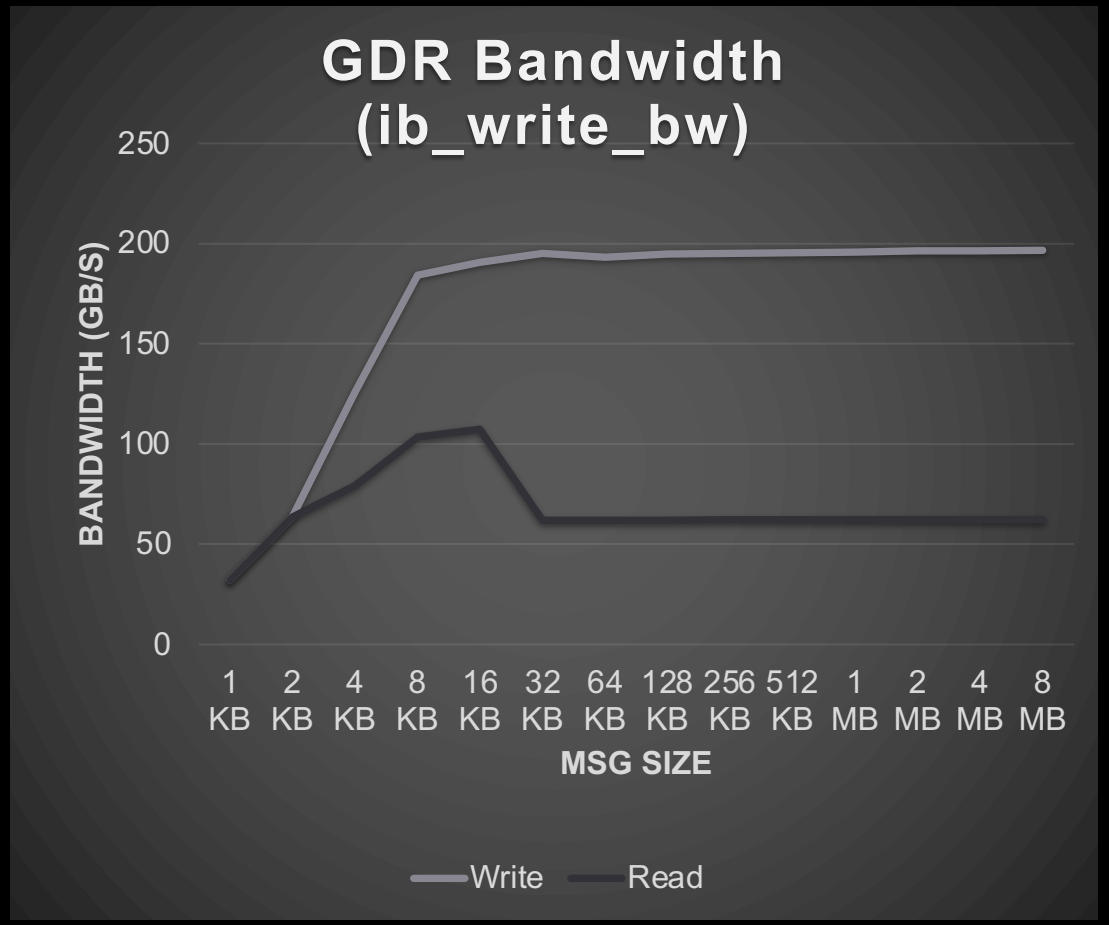


- Solution: issue write from the source GPU using ROCR/HSA
- Limitation: Doesn’t work if both devices are not visible to both processes

D2D transfers for inter-node using GDR

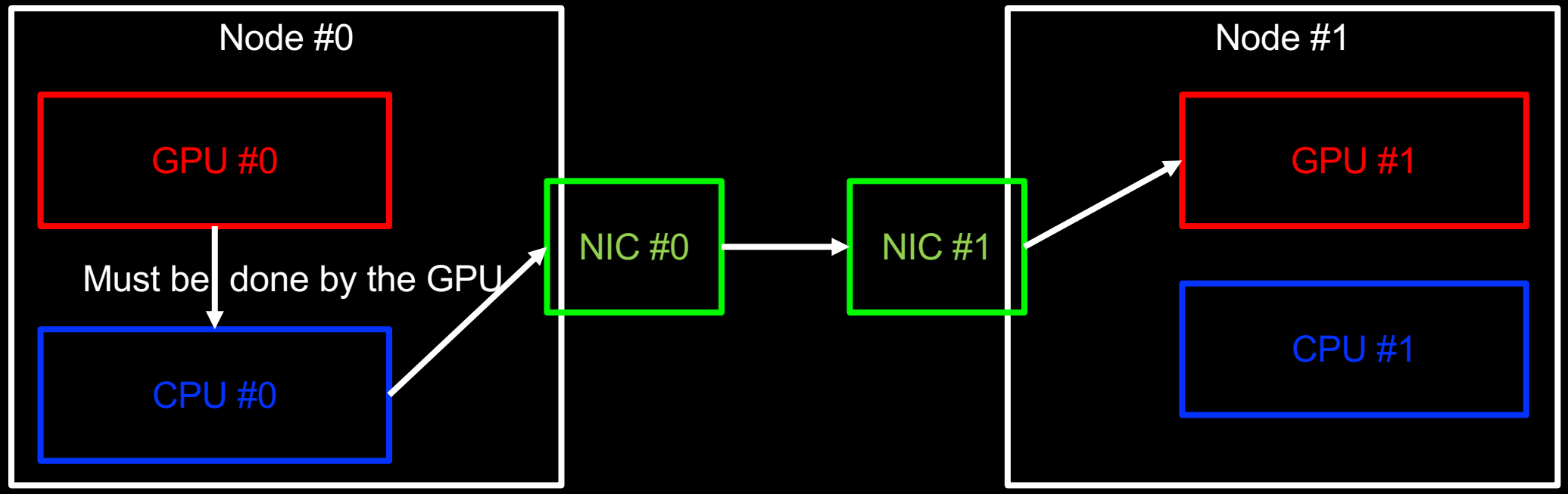


GDR Read/Write performance



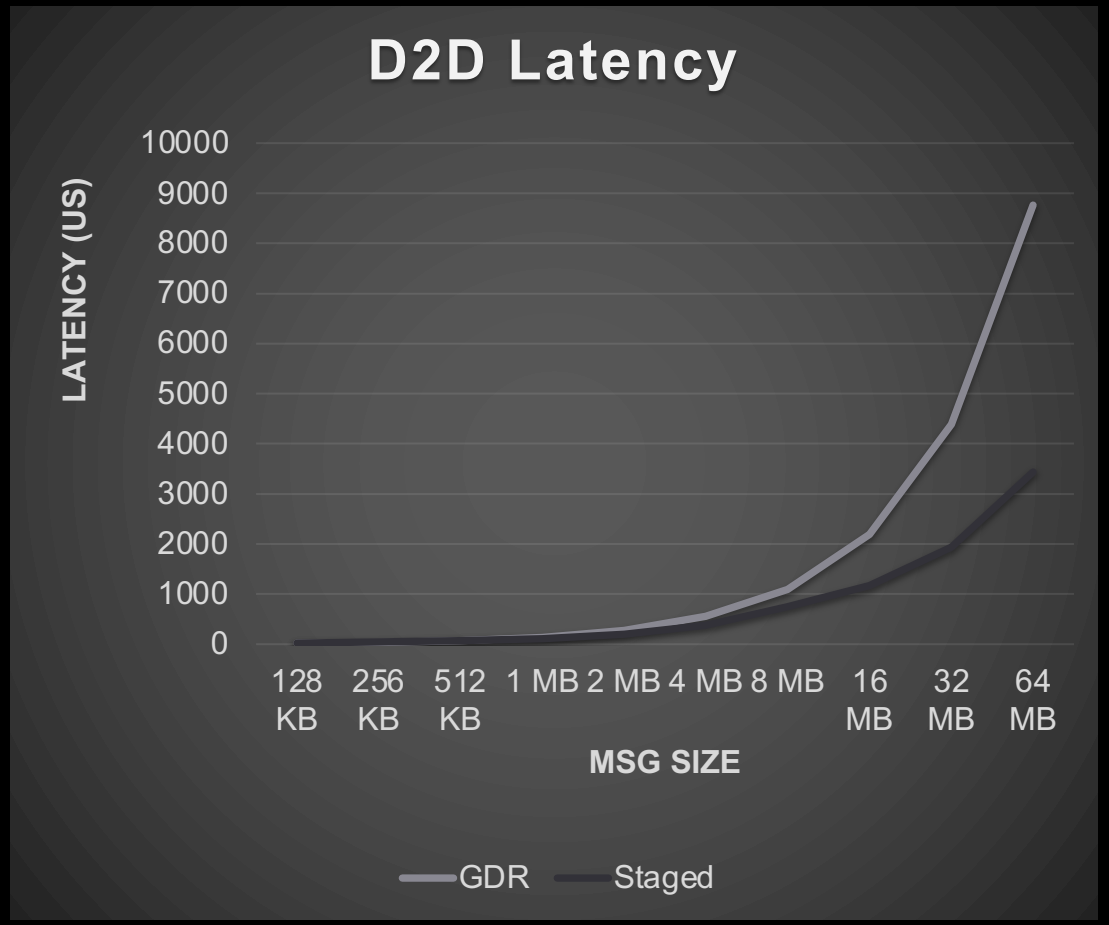
- GDR Writes can saturate line rate (~200 Gb/s)
- GDR Reads are slow due to PCIe limitations (~60/120 Gb/s)
- Performance depends on PCIe root complex sharing between GPU and NIC
- put_zcopy doesn't help here because the NIC still needs to read from the source GPU#0

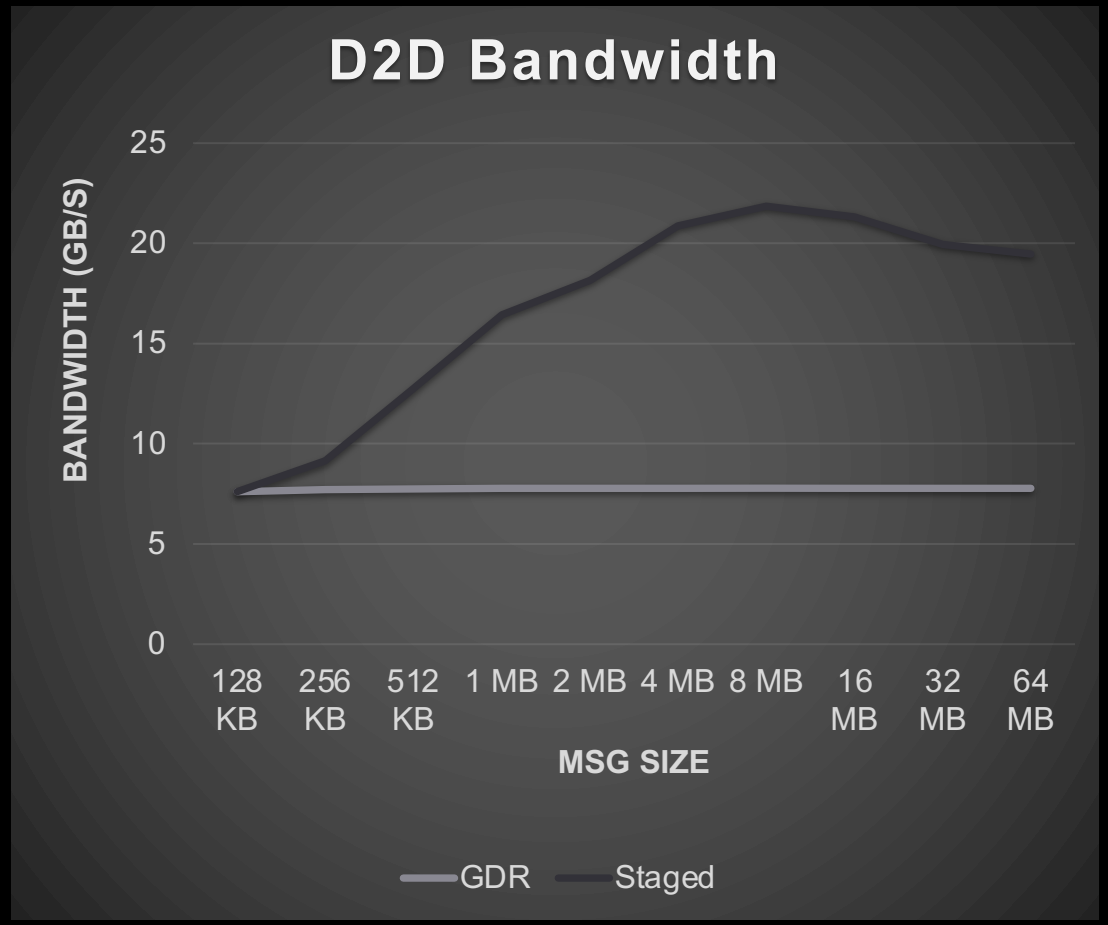
Staged D2D transfers for inter-node



Initial Support: @bureddy et al
Add ROCm support: #5742

Staged D2D transfers for inter-node





What's next?

- Support for MPICH

What we are looking forward to?

- Heterogenous topology support
  - How to detect and encode link type? (PCIe/Infinity Fabric/NVLink)
  - How to encode distance between PCIe devices?


AMD