

ROCm Bandwidth Test

The application `rocm_bandwidth_test` was developed to allow users discover the performance characteristics of Host-To-Device, Device-To-Host and Device-To-Device copy operations on a Rocm platform. The application can be run on any compliant Rocm platform. The application provides various options for users to experiment the cost of various copy operations in both unidirectional and bidirectional modes. Users can query the various options that are supported by giving the "-h" option.

Building RocmBandwidthTest:

ROCm Repo Forest:

- `cd ~/git/compute/tests/rocm_bandwidth_test`
- Execute `build_rocm_bandwidth_test.sh` script

Clone of Github Repo:

- `cd rocm_bandwidth_test`
- `mkdir build`
- `cd build`
- `export BUILD_TYPE=Debug | Release`
- `export ROCR_INC_DIR=<Path To ROCr Header Files>`
- `export ROCR_LIB_DIR=<Path To ROCr Library Files>`
- `cmake -DCMAKE_BUILD_TYPE=$BUILD_TYPE -DROCR_LIB_DIR=$ROCR_LIB_DIR -DROCR_INC_DIR=$ROCR_INC_DIR ..`

The following sections show how users can use the test to get performance data for various scenarios:

@note: The test will filter out certain operations that are either considered not supported or don't make sense. These include the following:

- No copy requests when both Src and Dst devices are Cpu.
- No copy requests when both Src and Dst devices are Same Gpu device and the request is either a partial or a full bidirectional copy operation

Where To Get Test From: Users can build it from [source](#) available at github. Currently access to source is limited to approved users. To request permission please contact the author of this page.

Help Screen: Run the test to print the help screen

```
<shell_prompt> ./rocm_bandwidth_test -h
```

Print Rocm Topology: Run the test to print topology of the various devices, their allocatable memory and access paths

```
<shell_prompt> ./rocm_bandwidth_test -t
```

The above command will print four things: Version of RBT, List of devices and their allocatable memory, Access matrix and Numa Distance among the various devices.

Copy Overhead Determination: Run the test to determine overhead of copy path - copy sizes from ONE byte to hundreds of bytes in increments of power of 2

```
<shell_prompt> ./rocm_bandwidth_test -s <gpu_dev_IdM> -d <gpu_dev_IdN> -l
```

The above command will print Version of RBT and time taken to perform copy for given device list

Data Path Validation: Run the test to validate data path from a source device to destination device by copying data

```
<shell_prompt> ./rocm_bandwidth_test -v
```

The above command will print four things: Version of RBT, List of devices, Access matrix and Data Path Validation among the various devices.

Default Unidirectional & Bidirectional All Devices Bandwidth: Run the test to collect performance characteristics of **unidirectional** and **bidirectional** copy operations involving **ALL** devices of a given Rocm platform.

```
<shell_prompt> ./rocm_bandwidth_test
```

The above command will issue **unidirectional** and **bidirectional** copy operations among all the devices of the platform.

Host-To-Device (H2D) Bandwidth: Run the test to collect performance characteristics of H2D copy operations of a given Rocm platform

```
<shell_prompt> ./rocm_bandwidth_test -s <cpu_dev_IdX>,<cpu_dev_IdY>,- - - -d <gpu_dev_IdM>,  
<gpu_dev_IdN>, - - -
```

The above command will issue **unidirectional** copy operations between Src and Dst devices. Specifically it will pair each device of Src List it with each device of Dst List i.e. it will launch `sizeof(SrcList) x sizeof(DstList)` number of copy operations. It is assumed that user has determined access from Src device to Dst device exists by consulting device access matrix.

Device-To-Host (D2H) Bandwidth: Run the test to collect performance characteristics of D2H copy operations of a given Rocm platform

```
<shell_prompt> ./rocm_bandwidth_test -s <gpu_dev_IdX>,<gpu_dev_IdY>,- - - -d <cpu_dev_IdM>,  
<cpu_dev_IdN>, - - -
```

The above command will issue **unidirectional** copy operations between Src and Dst devices. Specifically it will pair each device of Src List it with each device of Dst List i.e. it will launch `sizeof(SrcList) x sizeof(DstList)` number of copy operations. It is assumed that user has determined access from Src device to Dst device exists by consulting device access matrix.

Device-To-Device (D2D) Bandwidth: Run the test to collect performance characteristics of D2D copy operations of a given Rocm platform

```
<shell_prompt> ./rocm_bandwidth_test -s <gpu_dev_IdX>,<gpu_dev_IdY>,- - - -d <gpu_dev_IdM>,<gpu_  
dev_IdN>, - - -
```

The above command will issue copy **unidirectional** operations between Src and Dst devices. Specifically it will pair each device of Src List it with each device of Dst List i.e. it will launch `sizeof(SrcList) x sizeof(DstList)` number of copy operations. It is assumed that user has determined access from Src device to Dst device exists by consulting device access matrix.

Bidirectional Bandwidth: Run the test to collect performance characteristics of bidirectional copy operations of a given Rocm platform

```
<shell_prompt> ./rocm_bandwidth_test -b <device_IdX>,<device_IdY>,<device_IdZ>,- - -
```

The above command will issue **bidirectional** copy operations among all the devices of the list. In the example given it will issue `copy(x,x)`, `copy(x,y)`, `copy(x,z)`, `copy(y,x)`, `copy(y,y)`, `copy(y,z)`, `copy(z,x)`, `copy(z,y)` and `copy(z,z)` operations. The devices can be either be all Gpu's or Gpu/Cpu combination.

Unidirectional All Devices Bandwidth: Run the test to collect performance characteristics of **unidirectional** copy operations involving **ALL** devices of a given Rocm platform.

```
<shell_prompt> ./rocm_bandwidth_test -a
```

The above command will issue **unidirectional** copy operations among all the devices of the platform.

Bidirectional All Devices Bandwidth: Run the test to collect performance characteristics of **bidirectional** copy operations involving **ALL** devices of a given Rocm platform.

```
<shell_prompt> ./rocm_bandwidth_test -A
```

The above command will issue **bidirectional** copy operations among all the devices of the platform.