AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

BufferBandwidth

1 Overview

1.1 Location

`$<APPSDKSamplesInstallPath>\samples\opencl\cl\`

1.2 How to Run

See the *Getting Started* guide for how to build samples. You first must compile the sample.

Use the command line to change to the directory where the executable is located. The pre-compiled sample executable is at `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86\` for 32-bit builds, and `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

`BufferBandwidth`
This runs the program with the default options: `-t 0 -d 0 -nwk 1 -nl 20 -nr 1 -nk 20 -nb 33554432` (32MB) `-nw 7 -s 2 -if 0 -of 1 -cf 5 -cf 2`.

1. `BufferBandwidth -h`
   This prints the help file.

1.3 Command Line Options

Table 1 lists, and briefly describes, the command line options.

Table 1 Command Line Options

Short Form	Long Form	Description
`-h`	`--help`	Shows all command options and their respective meanings.
	`--device [cpu|gpu]`	Devices on which the OpenCL kernel is to be run. Acceptable values are `cpu` or `gpu`.
`-q`	`--quiet`	Quiet mode. Suppresses all text output.
`-e`	`--verify`	Verify results against reference implementation.
`-v`	`--version`	AMD APP SDK version string.
	`--dump [filename]`	Dump the binary image for all devices.
	`--load [filename]`	Load the binary image and execute on the device.
	`--flags [filename]`	Specify the filename containing the compiler flags for building the kernel.
`-i`	`--iterations`	Number of iterations.
`-p`	`--platformId`	Select the platformId to be used[0 to N-1 where N is number platform s available].
`-d`	`--deviceId`	Select deviceId to be used[0 to N-1 where N is number devices available].
`-nwk`	`--numcpuwrk`	Number of CPU workers (max: 32 (Linux: 1))

Short Form	Long Form	Description
-nl	--numLoops <n>	Number of iterations to repeat overall bandwidth measurement
-lp	--numIterBfr <n>	When -noblock is active, set the number of iterations to run read/write buffer calls
-nb	--numBytes <n>	Buffer size in bytes (min: 2048*CPU Workers)
-nr	--numRepeats <n>	Repeat each timing <n> times (can't be 0)
-nw	--numWavefronts <n>	# of wave fronts per SIMD (can't be 0) (default: 7)
-ty,	--testType <n>	Type of test: 0 clEnqueue[Map,Unmap] 1 clEnqueue[Read,Write] 2 clEnqueueCopy 3 clEnqueue[Read,Write], prepinned
-dma	--pcie	Measure PCIe/interconnect bandwidth
-nob	--noblock	When -pcie is activ, measure PCIe/interconnect bandwidth using multiple back-to-back asynchronous buffer copies
-s,	--nSkip <n>	Skip first <n> timings for average (default: 1)
-l	--printLog	Print complete timing log
-[if, of, cf]	--[inputflag, outputFlag, coyflag] <n>	Input, output, copy flags. OK to use multiple. 0 CL_MEM_READ_ONLY 1 CL_MEM_WRITE_ONLY 2 CL_MEM_READ_WRITE 3 CL_MEM_USE_HOST_PTR 4 CL_MEM_COPY_HOST_PTR 5 CL_MEM_ALLOC_HOST_PTR 6 CL_MEM_USE_PERSISTENT_MEM_AMD 7 CL_MEM_HOST_WRITE_ONLY 8 CL_MEM_HOST_READ_ONLY 9 CL_MEM_HOST_NO_ACCESS
-hbw	--enablehostbw	Enable/disable host mem B/W baseline. 0 or 1.
-m	--mapRW	Always map as MAP_READ | MAP_WRITE
-t	--timings	Print all timings including setup-time

2 Introduction

This sample measures a complete round trip loop of data transfer steps to, and from, an OpenCL device. It also assesses the bandwidth characteristics of a given system, including GPU memory and interconnect (for example: PCIe) bandwidth, achievable in OpenCL.

This sample can run the following tests:

- Create a simple baseline for host memory read and write performance. This can be used to ensure sanity of device buffer access performance numbers created by the other tests.
- Benchmark a round-trip chain of synchronous, serialized transfer steps between the host and the device.
- The sample can create a log over many iterations to locate one-time effects or variations over time.

The following transfer paths can be tested via command line option:

clEnqueueMap/UnmapBuffer
clEnqueueRead/WriteBuffer
clEnqueueCopyBuffer
clEnqueueRead/WriteBuffer, prepinned

This sample allows selection of any of the various CL buffer creation attributes for the source and destination buffers of the transfer chain.

On an OpenCL 2.x compliant device, the sample also measures the SVM buffer bandwidth when run with Map/Unmap test (command line option: -t 0 (default)).

3 Implementation Details

The bandwidth reported by -pcie or -dma is obtained by measuring the clEnqueueRead/Write performance on a prepinned buffer. This typically corresponds to the maximum interconnect rate achievable at application level for the explicit copy path (usually by DMA engine).

Details on the various buffer types and recommended transfer paths are provided in Chapter 4 of the *AMD APP OpenCL Programming Guide*. The combined -pcie -noblock option set measures the transfer rate of submitting back-to-back multiple asynchronous buffer copies. A higher transfer rate could be achieved using this model due to reduced overhead. The application code should follow this model by submitting as many commands to a CL queue as possible before forcing the queue to drain.

4 Notes and Caveats

- Do not run graphics applications while benchmarking compute or transfer operations.
- The read and write GPU kernels are written for clarity, and should achieve around 85% of HW peak with the right number of threads.
- The data verification used is basic.
- The smallest supported buffer size in this sample is 2048 bytes, corresponding to a single work-group of 128 work-items. Buffer sizes supplied by -nb are adjusted to a multiple of a block size that is known to perform well across all measurement stages.

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For AMD Accelerated Parallel Processing:

URL: developer.amd.com/appsdk
Developing: developer.amd.com/
Forum: developer.amd.com/openclforum