SAMPLE

Concurrent Kernels

1 Overview

1.1 Location

`$<AMDAPPSDKSamplesInstallPath>\samples\opencl\cpp_cl\1.x`

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
`$<AMDAPPSDKSamplesInstallPath>\samples\opencl\bin\x86\` for 32-bit builds, and
`$<AMDAPPSDKSamplesInstallPath>\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `ConcurrentKernels`
   This command runs the program with the default options.
2. `ConcurrentKernels -h`
   This command prints the help file.

1.3 Command Line Options

Table lists, and briefly describes, the command line options.
Command Line Options:

Short Form	Long Form	Description
`-h`	`--help`	Shows all command options and their respective meanings.
	`--device`	Devices on which the program 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.
`-t`	`--timing`	Print timing-related statistics.
	`--dump`	Dump binary image for all devices.
	`--load`	Load binary image and execute on device.
`-p`	`--platformId`	Select platformId to be used (0 to N-1, where N is the number of available platforms).
`-v`	`--version`	AMD APP SDK version string.
`-d`	`--deviceId`	Select deviceId to be used (0 to N-1, where N is the number of available devices).
`-i`	`--iterations`	Number of iterations.
`-x`	`--size`	Size of the input buffer per kernel (in Bytes).

Short Form	Long Form	Description
`-w`	`--workgroups`	Number of workgroups per kernel execution.
`-l`	`--localsize`	Number of work items (2^N) per workgroup.
`-k`	`--kernels`	Number of kernels for concurrent executions.

2 Introduction

This sample shows how to execute more than one kernel concurrently on a GPU. Currently, concurrent kernel execution is supported on AMD's GCN architecture GPUs.

To run multiple kernels concurrently, the kernels must be submitted to different command queues. There must be no data or buffer dependency between the kernels. The GPU must possess enough compute resources to execute the kernels simultaneously.

The sample includes a sequential implementation in which kernel executions happen sequentially as only one command queue is used. Not all compute resources are fully utilized in this case. The concurrent version shows an improvement in the resource utilization of the GPU by way of executing kernels concurrently using separate command queues for each kernel.

3 Implementation

The sample implements two versions of kernel executions:

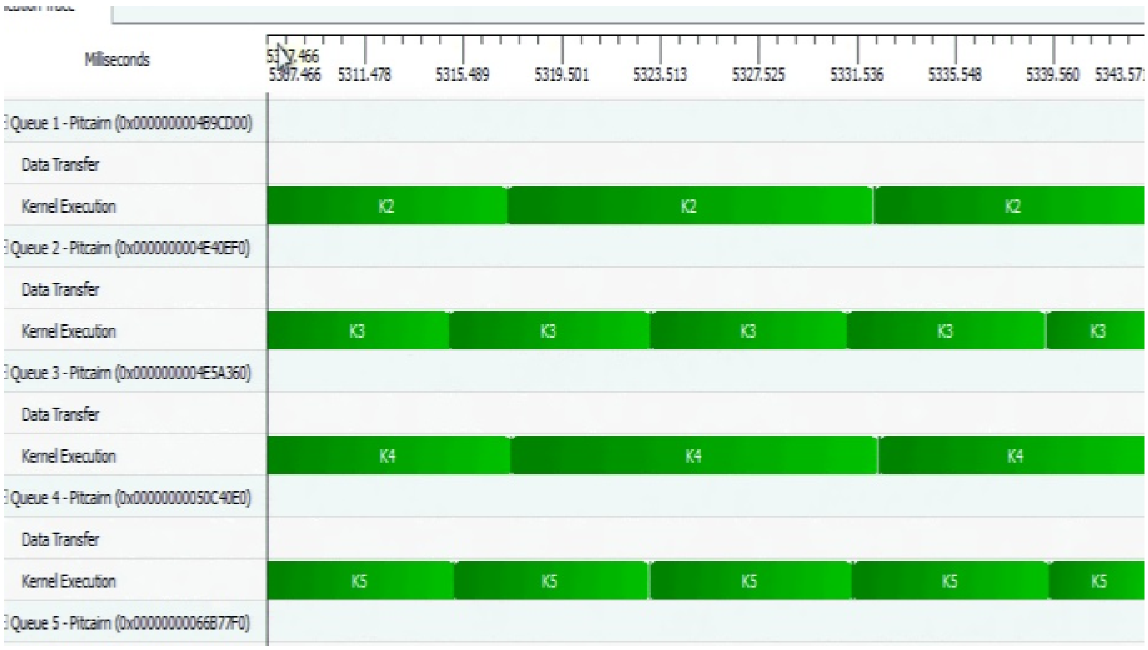
1. Sequential
2. Concurrent

3.1 Sequential execution implementation

As the name says, the kernels run sequentially one after the other. The kernels are queued in a single queue and are executed sequentially. The execution of a kernel starts after the previous kernel completes execution. The following CodeXL snapshot illustrates the sequential operation:

3.2 Concurrent execution implementation

In this version, the kernels are executed concurrently by using separate command queues for each kernel. The following CodeXL snapshot illustrates the concurrent operation:



As seen, the kernels are queued in separate queues (1, 2, 3, and 4) and execute concurrently.

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/
Support: developer.amd.com/appsdksupport