AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

FineGrainSVMSCAS

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. Ensure that the OpenCL 2.0 environment is installed.

Type the following command(s).

1. `FineGrainSVMCAS`
   This command runs the program with the default options.
2. `FineGrainSVMCAS -h`
   This command 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.
`-t`	`--timing`	Print timing-related statistics.
`-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].

2 Introduction

This sample demonstrates the usage of the atomic operation "CompareAndSwap" (CAS) call called `atomic_compare_exchange`, which is introduced in OpenCL 2.0 (adopted from C11 standards). This sample illustrates the creation of a linked list using the "lock free" atomics programming approach.

3 Implementation

This sample demonstrates the usage of the `atomic_compare_exchange` call which is introduced in OpenCL 2.0. The `atomic_compare_exchange` semantics is to update a given object atomically if object has an expected value. This can be used to create a linked list by inserting global ids of each work item into an array. The second kernel unlinks each of these again atomically using CAS operation again.

The sample workflow is as follows:

1. The host creates a fine grain array which represents the linked list to hold IDs of all work items.
2. There are two kernels. The first kernel inserts in lock-free manner their respective IDs into the linked list using CAS operation.
3. The second kernel unlinks or deletes them one-by-one atomically using CAS.
4. The host checks for the final result and exits.

This sample must be run in the OpenCL 2.0 environment. For details, see Chapter 6 of the *AMD OpenCL User Guide* document for details. For details about fine grain and atomics calls in OpenCL 2.0, see the OpenCL 2.0 spec on the Khronos web site.

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
Forum: developer.amd.com/openclforum