AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

SVM Binary Tree Search

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 `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86\` for 32-bit builds, and at `$<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. `SVMBinaryTreeSearch`
   This runs the program with the default options.
2. `SVMBinaryTreeSearch -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`	Devices on which the program is to be run. Acceptable values are cpu or gpu.
`-q`	`--quiet`	Quiet mode. Suppresses most 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.
	`--flags`	Specify compiler flags to build the kernel.
`-p`	`--platformId`	Select platformId to be used (0 to N-1, where N is the number of available platforms).
`-d`	`--deviceId`	Select deviceId to be used (0 to N-1, where N is the number of available devices).
`-v`	`--version`	AMD APP SDK version string.
`-i`	`--iterations`	Number of iterations for kernel execution.

Short Form	Long Form	Description
-n	--nodes	Number of nodes in the binary tree.
-k	--keys	Number of keys to be searched.
-r	--randMax	Maximum random number value.
-s	--seed	Seed to the random number generator.

2 Introduction

This sample demonstrates the use of the coarse grain Shared Virtual Memory (SVM) feature of OpenCL.2.0. SVM is an address space exposed to both the host and the device. This address space could be effectively used to share virtual pointers created in this space. Thus, data structures based on memory pointers, such as linked lists and binary trees could be shared between the host and the device using SVM. For coarse grain SVM, granularity is at the level of OpenCL memory objects, and consistency is guaranteed at synchronization points.

3 Implementation Details

This implementation uses coarse grain SVM to share a binary search tree between the host and the OpenCL device. The host allocates an SVM buffer and creates a binary search tree (BST) in this buffer. It then passes this BST and an array of search keys to be searched in this BST. The device takes this array and performs a parallel key search with one work-item performing one key search.

4 References

1. The OpenCL Specification, version 2.0, rev 22 document (page 38).
2. The OpenCL C Programming Language (ver 2.0, rev 22) document.

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

The contents of this document are provided in connection with Advanced Micro Devices, Inc. (“AMD”) products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD’s Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD’s products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD’s product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Copyright and Trademarks

© 2014 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, Radeon, FireStream, and combinations thereof are trademarks of Advanced Micro Devices, Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos. Other names are for informational purposes only and may be trademarks of their respective owners.