AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Dynamic OpenCL Detection

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).

1. `DynamicOpenCLDetection`
   This command tests the kernel execution on multi-devices asynchronously.
2. `DynamicOpenCLDetection -h`
   This command prints the help file.

2 Introduction

This sample demonstrates how to write an OpenCL application that will run even in the absence of `OpenCL.dll`. The program dynamically executes an OpenCL version of the program when the OpenCL runtime is present and executes the sequential or CPU version of the program when the OpenCL runtime is not present. This approach is useful in scenarios in which applications are to be ported across homogeneous and heterogeneous systems.

3 Implementation

The OpenCL runtime is detected by checking for `libOpenCL.so` in Linux and `OpenCL.dll` in Windows. The `dlOpen()` call in Linux and the `LoadLibrary()` call in Windows are used to locate the runtimes.

The OpenCL version of the application is pre-compiled and stored as a shared object (.so) in Linux and as a dynamically linked library (.dll) in Windows. Once the OpenCL runtime is found, `dlOpen()` or `LoadLibrary()` are used again to load the OpenCL version of the application. If the OpenCL version of the application is also found, then the address of the function to start with is obtained by using the `dlsym()` call in Linux or the `GetProcAddress()` call in Windows. The function address is used for calling it with the relevant parameters.

If the OpenCL runtime is not found and/or the OpenCL application is not found, then the sequential version is executed.

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

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

© 2013 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.