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.

On Linux systems, this sample dynamically links to the `libVectorAddition.so` shared library. This library is created and placed in the same directory in which the executable for `DynamicOpenCLDetection` is placed. In an independent build, this library is placed in `$<APPSDKSamplesInstallPath>/samples/opencl/cl/DynamicOpenCLDetection/bin/x86_64/Release`, while the pre-built binary is placed in `$<APPSDKSamplesInstallPath>/samples/opencl/bin/x86_64/Release`. In order to make this sample work properly, the `LD_LIBRARY_PATH` environment variable must be updated to include the directory in which the `libVectorAddition.so` library is placed.

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