AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Stencil2D

1 Overview

1.1 Location

`$(AMDAPPSDKSAMPLESROOT)\samples\C++Amp\examples`

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 default executables are placed in `$(AMDAPPSDKSAMPLESROOT)\samples\C++Amp\bin\x86\` for 32-bit builds, and `$(AMDAPPSDKSAMPLESROOT)\samples\C++Amp\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `SPMV`

   This runs the program with the default options `-i 10`

2. `SPMV -h`

   This prints the help file.

Ensure Microsoft® Visual Studio® 2012 or higher is installed.

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`	Show all command options and their respective meaning.
`-e`	`--verify`	Verify results against reference implementation.
`-t`	`--timing`	Print timing-related statistics.
`-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 times to repeat each algorithm. The default is 10.
`-V`	`--array_view`	Use array_view instead of array.

1.4 Array and Array_view

To use array, you must copy the data explicitly to the accelerator; with array_view, this is not necessary.

2 Introduction

This sample uses the mathematically simple model of a finite difference stencil [1] to compute the two dimensional matrix (the size is 4096 * 4096) by following method.

1	2	3
4	5	6
7	8	9

The value of each point is:

*centerWeight *centerVal + cardinalWeight *cardinalSum + diagonalWeight * diagonalSum*.

The *centerWeight*, *cardinalWeight*, and *diagonalWeight* are three random weights.

For example, the 5 in the table for example, the *centerVal* is 5. The *cardinalSum* is 2+4+8+6 and the *diagonalSum* is 1+3+7+9. Compute each point (except the boundary) by this equation.

The input is a 2D grid (4096*4096). It is subdivided into multiple tiles of 16x16. Each tile is processed by a single thread on the GPU. Each thread computes a value according to the previously described algorithm. Edge positions are handled with special weight values.

3 References

[1] Xie D. “A new block parallel SOR method and its analysis.” *SIAM Journal on Scientific Computing*, 2006, 27:1513-1533.

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