AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Gaussian LU Decomposition

1 Overview

1.1 Location

`$(AMDAPPSDKSAMPLESROOT)\samples\opencl\cl\app`

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 `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86` for 32-bit builds and `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `LUDecomposition`
   This uses the Gaussian LU decomposition algorithm to determine the factorization of a random square matrix, which is the product of a lower triangular matrix and and upper triangular matrix. Default option is x = 16.
2. `LUDecomposition -h`
   This prints the help message.

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 meaning.
	`--device`	Devices on which the program 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.
	`--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 deviceId to be used (0 to N-1, where N is the number of available devices).
`-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.
`-x`	`--dimension`	Actual dimension of input matrix.
`-i`	`--iterations`	Number of iterations for kernel execution.

2 Description

The sample calculates LU Decomposition of a random square matrix using basic Gaussian LU Decomposition Algorithm.

Let *A* be a square matrix. An **LU decomposition** is a decomposition of the form

A = ***L U***

where ***L*** and ***U*** are the lower and upper triangular matrices (of the same size), respectively. This means that *L* has only zeros above the diagonal and *U* has only zeros below the diagonal. For a 3X3 matrix it becomes

$$[A]=[L][U]=\begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

3 Algorithm

The algorithm used is a naïve Gaussian elimination algorithm. It is a recursive algorithm that treats a matrix n-1 times recursively. The algorithm makes all elements in a column below diagonal zero in each successive step.

The operation performed in the k^{th} step makes all elements of the k^{th} column below the diagonal zero using basic matrix algebra.

At the end of n-1 steps, an upper triangular matrix (U) is obtained from the original matrix. The multipliers used in each step are grouped into a lower triangular matrix (L).

4 Implementation Details

The basic requirement of the algorithm is that one step must be completed before starting the next step. To enforce this global behavior each step of the n-1 steps are performed one after other by different calls to the kernel. The matrix need not be transferred each time because of the persistent model of the global memory. So all the kernels modify the same buffer, one after other without the overhead of bringing the data in global buffers again & again. Double is used for better precision and vectorized reads are writes performed for efficiency. In addition LDS is used to store the multipliers for each step which reduced the memory fetches and eliminate redundant calculation.

5 References

- www.cse.illinois.edu/courses/cs554/notes/06_lu.pdf
- http://mathworld.wolfram.com/GaussianElimination.html
- http://www.math-linux.com/spip.php?article51

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

© 2011 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.