AMD Accelerated Parallel Processing TECHNOLOGY

SAMPLE

Sort

1 Overview

1.1 Location

$<*APPSDKSamplesInstallPath*>\samples\C++Amp\

1.2 How to Run

See the *Getting Started* guide for how to build samples. You must first compile the sample. Use the command line to change to the directory where the executable is located. The pre-compiled sample executable is at `$<APPSDKSamplesInstallPath>\samples\C++Amp\bin\x86\` for 32-bit builds, and `$<APPSDKSamplesInstallPath>\samples\C++Amp\bin\x86_64\` for 64-bit builds.

Ensure you have installed Microsoft® Visual Studio® 2012 or higher.

Type the following command(s).

1. `Sort`
   This runs the program with the default option s = 8.
2. `Sort -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 meaning.
`-q`	`--quiet`	Quiet mode. Suppresses all text output.
`-e`	`--verify`	Verify results against reference implementation.
`-t`	`--timing`	Print timing.
`-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).
`-s`	`--size`	The number of M data to be sorted.
`-i`	`--iterations`	Number of iterations for kernel execution.

2 Introduction

This sample implements a radix sort by using C++ Amp. The radix sort algorithm is divided into three phases:

1. Calculate the histogram of an unsorted array.

2. Scan the histogram bins.
3. Rank and permute to keys to get a sorted array.

3 Implementation Details

The implemented radix sort breaks keys (32 integers) into 16-bit digits and sorts one 16-bit digit at a time, starting with the least significant one. It loops eight times to complete sorting. In each *i*th loop, the following three phases sort the input array using ith 16-bit digit.

1. Calculate histogram bins.

   There are cNTiles * cTileSize threads to calculate histogram bins. cTileSize is the number of threads per AMP tile. cNTiles is the number of AMP tiles. The input array is divided into blocks of regionSize * cNTiles elements. In this case, cTileSize = 256, cNTiles = 64. Each work-item calculates its histogram bin from the allotted 512 elements and passes this histogram to next phase.

2. Scan histogram bins.

   In this phase the histogram bins are scanned column-wise, where histogram bins are arranged in the following way.

   There are B * N histogram bins, where B is the number of blocks, and N is the number of work-items in a block. Histogram bins are arranged such that the *0*th block bin comes first, and the (B - 1)th block comes last. Each block's histogram bins are arranged so that the *0*th work-item bin comes first, and (N - 1)th work-item bin comes last.

   The scanned histogram is passed to next phase.

3. Rank and permute keys to get the sorted array.

   Each work-item permutes the allotted 128 elements by using its scanned histogram bins.

4 References

1. Marcho Zagha and Guy E. Blelloch. “Radix Sort For Vector Multiprocessor.” in: Conference on High Performance Networking and Computing, pp. 712-721, 1991.

2. Guy E. Blelloch, Prefix Sums and Their Applications, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1990.

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**