ViennaData 1.0.0

User Manual

Institute for Microelectronics
Gußhausstraße 27-29 / E360
A-1040 Vienna, Austria

Main Contributors:

Karl Rupp

Current Maintainers:

Karl Rupp

Institute for Microelectronics
Vienna University of Technology
Gußhausstraße 27-29 / E360
A-1040 Vienna, Austria/Europe

Phone	+43-1-58801-36001
FAX	+43-1-58801-36099
Web	http://www.iue.tuwien.ac.at

Contents

Introduction

ViennaData is nice!

Chapter 1

Installation

This chapter shows how ViennaData can be integrated into a project and how the examples are built. The necessary steps are outlined for several different platforms, but we could not check every possible combination of hardware, operating system and compiler. If you experience any trouble, please write to the maining list at

viennacl-support@lists.sourceforge.net

1.1 Dependencies

ViennaData uses the CMake build system for multi-platform support. Thus, before you proceed with the installation of ViennaData, make sure you have a recent version of CMake installed.

- A recent C++ compiler (e.g. GCC version 4.2.x or above and Visual C++ 2008 are known to work)
- CMake [1] as build system (optional, but highly recommended for building the examples)

1.2 Generic Installation of ViennaData

Since ViennaData is a header-only library, it is sufficient to copy the folder viennadata / either into your project folder or to your global system include path. On Unix based systems, this is often /usr/include/ or /usr/local/include/.

On Windows, the situation strongly depends on your development environment. We advise users to consult the documentation of their compiler on how to set the include path correctly. With Visual Studio this is usually something like C:\Program Files\Microsoft Visual Studio 9.0\VC\include and can be set in Tools -> Options -> Projects and Solutions -> VC++-Directories.

Tutorial No.	Dependencies
tutorial/tut1.cpp	OpenCL
tutorial/tut2.cpp	OpenCL, ublas
tutorial/tut3.cpp	OpenCL, ublas
tutorial/tut4.cpp	ublas
tutorial/tut5.cpp	OpenCL
benchmarks/vector.cpp	OpenCL
benchmarks/sparse.cpp	OpenCL, ublas
benchmarks/solver.cpp	OpenCL, ublas

Table 1.1: Dependencies for the examples in the examples/ folder

1.3 Building the Examples and Tutorials

For building the examples, we suppose that CMake is properly set up on your system. The other dependencies are listed in Tab. 1.1.

1.3.1 Linux

To build the examples, open a terminal and change to:

```
$> cd /your-ViennaData-path/build/
```

Execute

```
$> cmake ..
```

to obtain a Makefile. Executing

```
$> make
```

builds the examples. If some of the dependencies in Tab. 1.1 are not fulfilled, you can build each example separately:

```
$> make tut1              #builds tutorial 1
$> make vectorbench       #builds vector benchmarks
```

Speed up the building process by using jobs, e.g. make -j4.

1.3.2 Mac OS X

The tools mentioned in Section 1.1 are available on macintosh platforms too. For the GCC compiler the Xcode [2] package has to be installed. To install CMake and Boost external portation tools have to be used, for example, Fink [3], DarwinPorts [4] or MacPorts [5]. Such portation tools provide the aforementioned packages, CMake and Boost, for macintosh platforms.

If the CMake build system has problems detecting your Boost libraries, determine the location of your Boost folder. Open the CMakeLists.txt file in the root directory of ViennaData and add your Boost path after the following entry:
`IF($CMAKE_SYSTEM_NAME MATCHES "Darwin")`

The build process of ViennaData is similar to Linux.

1.3.3 Windows

In the following the procedure is outlined for Visual Studio: Assuming that an OpenCL SDK and CMake is already installed, Visual Studio solution and project files can be created using CMake:

- Open the CMake GUI.
- Set the ViennaData base directory as source directory.
- Set the build/ directory as build directory.
- Click on 'Configure' and select the appropriate generator (e.g. Visual Studio 9 2008)
- Click on 'Generate' (you may need to click on 'Configure' one more time before you can click on 'Generate')
- The project files can now be found in the ViennaData build directory, where they can be opened and compiled with Visual Studio (provided that the include and library paths are set correctly, see Sec. 1.2).

The examples and tutorials should be executed from within the build/ directory of ViennaData, otherwise the sample data files cannot be found.

Change Logs

Version 1.0.0

First release

License

Bibliography

[1] “CMake.” [Online]. Available: http://www.cmake.org/

[2] “Xcode Developer Tools.” [Online]. Available: http://developer.apple.com/technologies/tools/xcode.html

[3] “Fink.” [Online]. Available: http://www.finkproject.org/

[4] “DarwinPorts.” [Online]. Available: http://darwinports.com/

[5] “MacPorts.” [Online]. Available: http://www.macports.org/