Brown Deer
Technology

COPRTHR version 1.2 (middletown) Release Notes

The CO-PRocessing THReads (COPRTHR) SDK provides several OpenCL™ related libraries and tools for developers targeting GPU compute technology and hybrid CPU/GPU computing architectures.

1 New In Version 1.2 (middletown release)

- Expanded Operating System support:
  - Full SDK support for FreeBSD-8 including an open-source OpenCL implementation (libocl) for amd64
  - STDCL beta support for Windows 7 / MSVS 2010
- C++ container classes with OpenCL device-sharable memory:
  - OpenCL extension of STL vector and BOOST multi_array
  - Containers allow conventional data management on the host side with memory synchronization for OpenCL devices
- Transparent/automatic GPU acceleration of vector operations for C++ containers
  - Automatic kernel generation and host interfacing uses CLETE expression template engine
- Improvements to the SDTCL interface including

  - Support for image2D memory allocation using clmalloc
  - CLGL buffer sharing support
  - Run-time device management including exclusive device locks for MPI support
- Improvements to the open-source OpenCL run-time (libocl) including
  - Partial support for images
  - Many enhancements for performance and functionality
- Updated examples and documentation
- STDCL now tested against AMD SDK v2.4, Nvidia CUDA 4.0, Intel OpenCL SDK v1.1

---

2 Support and Requirements

With this release, support has been expanded to included FreeBSD-8 and Windows 7 operating systems, with Windows support limited to the basic functionality provided by the STDCL interface to OpenCL. Support continues for most modern Linux distributions including RHEL 5.4/5.5, CentOS 5.4/5.5, OpenSuSE 11.2/11.3 and Ubuntu 10.4. Specific feature support by operating system is shown in the table below.

COPRTHR Feature	Description	Linux	FreeBSD	Windows
libstdcl	STDCL interface	x	x	x
cltrace	tracing tool	x	x	
clld	link tool	x	x	
libocl	OpenCL x86_64 runtime	x	x	

This release is compatible with the OpenCL implementations provided with AMD APP v2.4, Nvidia CUDA-4 and Intel OpenCL SDK v1.1. In addition, an open-source OpenCL run-time implementation for x86_64 multi-core processors is provided as part of the COPRTHR SDK, which may be used on platforms for which no vendor support is available. The COPRTHR OpenCL implementation may also be of interest since it exhibits better performance than vendor implementations on some real-world benchmarks.

This release supports x86_64 CPUs from AMD and Intel as well as GPUs from AMD and Nvidia, and has been tested successfully on the following graphics cards: AMD Radeon HD 4850, 4870, 4870X2, 5870, 5970, 6970, AMD FirePro V8800, Nvidia Tesla S1070, C2050, and C2070.

If you are only interested in the basic functionality of the STDCL interface for OpenCL, no additional packages are required beyond the standard vendor implementation of OpenCL for your platform. (If none is available, try the open-source implementation provided by with COPRTHR SDK.)

If you wish to use the `clld` tool for embedding OpenCL kernel code into ELF objects to create single executables, you will need to install libelf-0.8.13. Note that versions of libelf typically found on Linux distributions (designated 1.x) are *not compatible* and not very useful since they often contain many undocumented behaviors, and should not be used.

If you wish to use the open-source OpenCL run-time implementation provided by COPRTHR SDK you will need to install libelf-0.8.13 along with a few additional packages. Specifically, you will need LLVM and CLANG v2.6 and the ATI Stream SDK v2.1 . (Newer versions of the AMD SDK are *not* valid substitutes.)

The table below provides a comprehensive matrix of required packages matched to a specific platform and feature set.

COPRTHR Feature				**Package**	**Download**
libstdcl	cltrace	clld	libocl		
Linux Red Hat 5.4/5.5, CentOS 5.4/5.5, OpenSuSE 11.2					
O				AMD APP v2.4	http://developer.amd.com/sdks/AMDAPPSDK/downloads
O				Nvidia CUDA 4	http://developer.nvidia.com/cuda-toolkit-40
O				Intel OpenCL SDK-1.1	http://software.intel.com/en-us/articles/download-intel-opencl-sdk
		R	R	libelf 0.8.13	http://www.mr511.de/software/libelf-0.8.13.tar.gz
			R	LLVM-2.6	http://llvm.org/releases/2.6/llvm-2.6.tar.gz
			R	CLANG-2.6	http://llvm.org/releases/2.6/clang-2.6.tar.gz
			R	ATI Stream SDK-v2.1	http://developer.amd.com/Downloads/ati-stream-sdk-v2.1-lnx64.tgz
FreeBSD-8					
		R	R	libelf 0.8.13	http://www.mr511.de/software/libelf-0.8.13.tar.gz
			R	LLVM-2.6	http://llvm.org/releases/2.6/llvm-2.6.tar.gz
			R	CLANG-2.6	http://llvm.org/releases/2.6/clang-2.6.tar.gz
			R	ATI Stream SDK-v2.1	http://developer.amd.com/Downloads/ati-stream-sdk-v2.1-lnx32.tgz
Windows 7					
O				AMD APP v2.4	http://developer.amd.com/sdks/AMDAPPSDK/downloads
O				Nvidia CUDA 4	http://developer.nvidia.com/cuda-toolkit-40
O				Intel OpenCL SDK-1.1	http://software.intel.com/en-us/articles/download-intel-opencl-sdk

O=Optional, R=Required

3 Important Notes

- The libraries `libstdcl` and `libocl` are provided with debug versions `libstdcl_d` and `libocl_d`, respectively. Linking against these libraries can be very useful for debugging as well as understanding how each library operates.

- If you install a binary release, it may not have been compiled with your preferred OpenCL implementation as a default. The most reliable way to ensure the correct implementation is used is to set the appropriate environment variable. The following are examples:

  export STDGPU_PLATFORM_NAME=nvidia

  Use the Nvidia OpenCL implementation for the stdgpu context

  export STDCPU_PLATFORM_NAME=intel

  Use the Intel OpenCL implementation for the stdcpu context

- The flag CL_EVENT_RELEASE has been removed and the flag CL_EVENT_NORELEASE has been added; the default behavior of event management calls has been changed so as to always release events unless the latter flag is used.

- The use of environment variable to control certain aspects of each default context has been changed. As an example, the environment variable STDGPU is now only checked for a 0 or 1 to determine whether the context is enabled. For more information see the revised STDCL reference manual (revision 1.2).

- The type of STDCL context pointers has been changed from CONTEXT* to CLCONTEXT* to avoid namespace collisions on Windows 7. The use of CONTEXT* is still acceptable but should be considered deprecated since its use will eventually be removed.

4 Frequently Asked Questions

Below are answers to frequently asked questions regarding COPRTHR SDK and STDCL.

Does STDCL require the BDT OpenCL run-time?

No. The basic installation of libstdcl.so will work with any compliant OpenCL installation including the latest implementations from AMD, Nvidia and Intel.

Will using STDCL reduce performance or limit access to OpenCL functionality?

No. STDCL is implemented as a very light-weight interface and does not restrict access to direct OpenCL and fully supports asynchronous operations across multiple devices.

Are STDCL calls simply wrappers for OpenCL calls?

No. There is a bit more to the interface than wrapping OpenCL calls. For the curious, take a look at the source code.

5 More Information

Additional information including installation instructions and examples may be found in **The COPRTHR Primer** revision 1.2 along with more detailed documentation and examples.

revised 18 June 2011 by DAR