Brown Deer Technology

Release Notes for the COPRTHR SDK version 1.5.0 (Marathon)

The CO-PRocessing THReads (COPRTHR) SDK provides libraries and tools for application developers targeting multi-core and many-core parallel co-processing computer platforms.

---

1 New In Version 1.5 (Marathon) release

- *OpenCL Remote Procedure Call (RPC) implementation (CLRPC)*
  - Stand-alone CLRPC server (clrpcd) for exporting OpenCL platforms over a network connection
  - Client-side OpenCL RPC implementation (`libclrpc`)
- *New OpenCL loader (`libocl`) with advanced functionlity*
  - Backward-compatible with Khronos ICD loader
  - Supports precise platform configuration with ocl.conf files
  - Integrated CLRPC support for access to remote CLRPC servers
  - Suppors process accounting and extensible host call hooks
- *New error reporting and debugging infrastructure*
  - OpenCL and STDCL error reporting via `oclerrno` and `clerrno`
  - Run-time selection of reporting level from all tools and libraries
- *New processor support for Adapteva Epiphany/Parallella*
  - Full SDK support for the Adapteva Epiphany processor and Parallella platform
  - Includes OpenCL and STDCL support for Parallella Epiphany and ARM processors
  - Includes extensions for the Epiphany architecture
- *STDCL CONTEXT `stdnpu` for networked compute devices (preview feature)*
  - Provides a single default compute context for accessing all networked devices available via CLRPC
- *New tools and commands*
  - `cltop` for monitoring co-processing similar to UNIX top command
  - `cldebug` front-end for application debugging

---

2 Support and Requirements

Full SDK support is provided for Linux, Ubuntu, and FreeBSD operating systems. Additionally, the STDCL API is supported for Windows 7.
Supported hardware includes AMD and Nvidia GPUs, Intel and AMD x86 multicore CPUs, ARM multicore CPUs, and Epiphany multicore processors. The COPRTHR SDK leverages vendor OpenCL GPU implementations and also provides OpenCL implementions for multicore processors to provide truly cross-vendor/cross-device support for heterogeneous computing platforms.

Package dependencies (O=Optional, R=Required)

1. libelf-0.8.13 www.mr511.de/software/libelf-0.8.13.tar.gz [Required]
2. libconfig-1.4.8 or later www.hyperrealm.com/libconfig/libconfig-1.4.9.tar.gz [Required]
3. libevent-2.0.18 or later github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz [Required]
4. GCC 4.6 or later [Required]

Additionally, vendor support is needed for certain specific OpenCL devices

5. AMD APP developer.amd.com/sdks/AMDAPPSDK/downloads [Optional]
6. Nvidia CUDA 4 developer.nvidia.com/cuda-toolkit-40 [Optional]
7. Intel OCL 1.5 software.intel.com/en-us/articles/vcsource-tools-opencl-sdk [Optional]
8. Adapteva Epiphany SDK for Epiphany and Parallella support [Optional]

Please take note that libelf 1.x branch found on most Linux distributions is *not a valid substitute* for libelf-0.8.13 since they lack the required features and exhibit undocumented broken behavior.

---

3 Installation

3.1 Installation Overview

The COPRTHR SDK may be installed from pre-compiled binaries for selected platforms or built from source. It may be necessary to install one or more 3rd-party packages as part of the installation process as described above in the previous section [Support and Requirements)(#support_and_requirements).

For Linux and FreeBSD installation a configure script is provided to customize the package. Detailed descriptions of the various options are described at the end of this section.

3.2 Download the Release

COPRTHR SDK is available from the github project page located at http://www.github.com/browndeer/coprthr .

3.3 Linux and FreeBSD Installation

The following generic steps can be used to install the COPRTHR SDK on Linux and FreeBSD systems. For a full description of configure options and suggested system-specific configurations, see section 2.5 and 2.6.

1) From the root coprthr directory, type,

./configure [options]

2) Build the package, type,

gmake

3) Install the package, type,

gmake install

4) Set the appropriate paths to use the headers and libraries when building your own applications.

3.4 Windows 7 Installation

Run the Windows installer (libstdcl-1.4.0-win7-install.msi) and set the appropriate paths to use the headers and library from your application.

3.5 Configure Options

When building from source the configure script supports the following options:

–**prefix=/path/to/target-install-dir** set the root directory for

installation

–**with-opencl-platform=NAMELIST** specify default platform selection as a priority-ordered comma seperated list

–**with-opencl-include=DIR** specify OpenCL include path

–**with-opencl-lib=DIR** specify OpenCL lib path

–**with-fortran=PROG** specify Fortran compiler PROG

–**with-libelf=DIR** specify path to libelf

–**with-libcoprthr-cc=PROG** specify the C compiler PROG to be used for the libcoprthr OCL implementation

–**with-libcoprthr-cxx=PROG** specify the C++ compiler to be used for the libcoprthr OCL implementation

–**enable-clgl** enable clgl support (enabled by default)

–**enable-cltrace** enable building cltrace tool (obsolete)

–**enable-clete** enable CLETE support (enabled by default)

–**enable-libocl** enable building the experimental libocl loader (enabled by default)

–**enable-libcoprthr** enable building the libcoprthr OCL implementation for multi-core CPUs (disabled by default)

–**enable-libcoprthr-ngpu** enable multiple CPU device support in libocl (disabled by default)

–**enable-fortran** enable fortran binding (disabled by default)

3.6 Suggested System-Specific Configurations

The following are just a few examples of common system-specific configurations to be used as a guide for installing COPRTHR SDK on your specific platform. In very many cases, simply using ./configure will work if you have installed 3rd-party software in standard locations.

Linux using only AMD APP SDK for CPU/GPU

```
./configure --with-opencl-platform=amdapp
```

Linux using AMD APP SDK for CPU/GPU and COPRTHR OpenCL for CPU also

```
./configure --enable-libcoprthr --with-opencl-platform=amdapp,coprthr
```

Linux using only Nvidia OCL for GPU

```
./configure --enable-libcoprthr --with-opencl-platform=nvidia --disable-libocl
```

(Note that libocl must be disabled to use the Nvidia OCL implementation at this time.)

Linux using only Intel OCL for CPU/GPU

```
./configure --with-opencl-platform=intel
```

Linux or FreeBSD standalone installation with COPRTHR OCL for CPU (x86)

```
./configure --enable-libcoprthr
```

Linux standalone installation with COPRTHR OCL for CPU (ARM)

```
./configure --enable-libcoprthr --disable-cltrace
```

Parallella/Epiphany support

```
./configure --enable-epiphany
```

---

4 Important Notes

- Release 1.5.0 must be used with the Epiphany SDK version 4.13.01.04 and is not compatible with the API used in version 4.13.03.30; support for the newer API will be provided in the next COPRTHR release (1.5.1).
- On Windows 7 platforms the function init_stdcl() must be called to initialize the STDCL API. This call is unnecessary for Linux and FreeBSD, but there is no harm in including it since it will default to an empty macro.
- The "debug libraries" have been eliminated with the new debug and reporting structure.
- The version of the boost library (1.47.0) used here for development unfortunately has a very minor bug that prevents it from working with MSVS 2010. A fix to the boost package is provided in this release. Just follow the instructions under the `msvs2010/boost_1_47_0-multi_array-iterator-fix/` directory. (All you need to do is copy over a replacement for iterator.hpp.)
- Some example kernels include the stdcl.h header. Including this header for OpenCL kernels allows the use of alternate syntax that avoids breaking C. Programmers are encouraged to begin this practice now. A future release will formalize syntax corrections to OpenCL to produce a C compliant language for programming thread functions (kernels). The alternate syntax will be completely backward compatable with OpenCL.

- When installing COPRTHR SDK on Linux Angstrom for ARM, be aware of the following required changes to the standard Linux Angstrom distribution. All busybox substitutes for UNIX commands that are encountered should be replaced with real implementations that actually work. Specifically, GNU coreutils and binutils should be installed and used to provide relevant commands. The command /usr/bin/time should be replaced with a command that works properly to execute the standard tests that come with COPRTHR. The standard shell /bin/sh should be replaced with a real implementation of bash which actually works. Finally, you must also re-install bison-2.5, flex-2.5.35, and m4-1.4.16 using the GNU distributions of these packages since the installed version of bison is broken.

---

5 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, 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.

---

6 More Information

Additional information including installation instructions and examples may be found in The COPRTHR Primer version 1.5 along with more detailed documentation and examples.

Revised 9 June 2013 by DAR