

Crea sito

TechCloud

See opportunities behind difficulties

OpenCL

SHARE

0

G+1

2

Tweet

Mi piace 0

Condividi



Introduction

With some years of experience on multithreading programming on many systems (Windows, Linux, Android) I would like to start developing using the emerging High Performance Computing (HPC) paradigm, addressing specifically HCS and openCL:

(from wikipedia) [Heterogeneous computing systems](#) refer to electronic systems that use a variety of different types of computational units. A computational unit could be:

- a [general-purpose processor \(GPP\)](#)
- a special-purpose processor (i.e. [digital signal processor \(DSP\)](#) or [graphics processing unit \(GPU\)](#)),
- a [co-processor](#), or custom acceleration logic ([application-specific integrated circuit \(ASIC\)](#)
- [field-programmable gate array \(FPGA\)](#)).

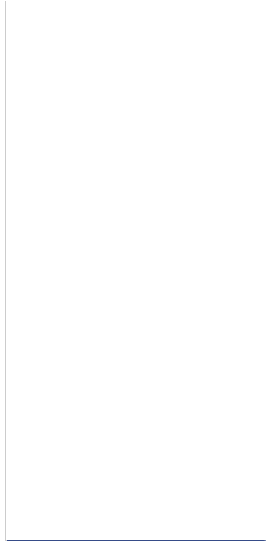
In general, a [heterogeneous](#) computing platform consists of processors with different [instruction set architectures \(ISAs\)](#).

With the emerging bottleneck of performance boost scaled by frequency increasing, and to respect the [Low of Moore](#), the new HCS are spreading: in our all-day computers we can take advantages from the new GPU/CPU capabilities to execute together the same code to speed-up elaborations.

This tutorial will address the parallel programming on current system architectures using openCL which

use CPUs and GPUs, and their integration in the accelerated processing unit (for example [AMD APU](#)).

*Please note: this page is about GPGPU and OpenCL, and is (always) under construction. Feel free to leave a comment to let me know about errors/news and keep this page updated.*



## OpenCL update sites

- wikipedia page, with history and version descriptions and releases:  
<http://en.wikipedia.org/wiki/OpenCL>
- <https://www.khronos.org/openc1/> OpenCL Home Page of Khronos
- <http://hgpu.org/> info and updated about HPC in general using GPU
- <http://www.openc1blog.com/> Blog about openCL
- <http://www.reddit.com/r/OpenCL/> News about openCL
- <http://mahout.apache.org/> Scalable machine learning library
  - <http://blog.sematext.com/tag/mahout/>

## GPU Vendors supporting openCL & SDK

- **ARM's Mali 6xx**  
wiki Mali GPU [http://en.wikipedia.org/wiki/Mali\\_\(GPU\)](http://en.wikipedia.org/wiki/Mali_(GPU))  
Mali developer Center for openCL  
<http://malideveloper.arm.com/develop-for-mali/sdk/mali-openc1-sdk/>
- Qualcomm's adreno 3xx
- Imaginations PowerVR 5xx/6xx  
wiki PowerVR: <http://en.wikipedia.org/wiki/PowerVR>
- Vivante support OpenCL
- Intel SDK <http://software.intel.com/en-us/vcsource/tools/openc1-sdk>
- NVIDIA's Tegra K1 OpenCL 1.2 ([http://www.nvidia.com/content/PDF/tegra\\_white\\_papers/Tegra-K1-whitepaper-v1.0.pdf](http://www.nvidia.com/content/PDF/tegra_white_papers/Tegra-K1-whitepaper-v1.0.pdf))

## OpenCL and Android

### Status of openCL on Android

- Google introduces support to openCL in Android 4.0, by adding openCL runtimes and driver in terms of shared libraries that could be found on devices.
- From 4.3, Google remove the openCL support in favor of its proprietary accelerated API RenderScript. This language is implemented on top of ARM Mali GPU, so the openCL is still there, but cannot be accessed by "normal means" by developer.
- This doesn't mean openCL cannot be used, simply we have to use older Android tools/versions or Linux toolchains.
- Even if the device supports OpenCL because the manufacturer has included the relative driver, it is necessary to have libOpenCL.so library on the development computer in order to link and use OpenCL. This shared library is device-specific, so cannot be use a library pulled from other device.

## Device actually support openCL (or howto)

- Samsung Galaxy S4 from:  
[http://www.reddit.com/r/OpenCL/comments/2063ci/442\\_kitkat\\_update\\_brings\\_openc1\\_on\\_samsung\\_galaxy/](http://www.reddit.com/r/OpenCL/comments/2063ci/442_kitkat_update_brings_openc1_on_samsung_galaxy/)

a site hosting openCL library:

[http://chomikuj.pl/ruras/Samsung+Galaxy+S4+I9505/I9505\\_Omega\\_v16\\_XXUEMK9\\_md5\\_F3CC0951A656670E169BFCB5EBF33D25/system/vendor/lib/libOpenCL,3311496998.so](http://chomikuj.pl/ruras/Samsung+Galaxy+S4+I9505/I9505_Omega_v16_XXUEMK9_md5_F3CC0951A656670E169BFCB5EBF33D25/system/vendor/lib/libOpenCL,3311496998.so)

- Sony Xperia Z has support for it:  
<https://clbenchmark.com/device-info.jsp?config=14950526&test=CLB10101>
- Nexus 10 & 4 (manufactured by Samsung)
  - Nexus 10 is shipped with openCL drivers installed and libOpenCL.so available
  - Update 4.3 remove drivers:  
<http://www.reddit.com/comments/1jz2th>
  - Nexus 10 a workaround (for not rooted phone) can be found here: <http://beyond3d.com/showthread.php?t=63071>
  - Active discussion of openCL on Nexus 10:  
<https://groups.google.com/forum/#!topic/android-ndk/jQLs5fMRRGY>

## Tutorial, examples and demos

- OpenCL/Nexus 4 blur image example  
<http://finnzan.blogspot.it/2013/05/opengl-on-android.html>
- Tutorial on use openCL on Android with NDK r9c/SDK 22.3 – Ubuntu 13.10 (this tutorial is specific for **Odroid**, but could be easily adapted for any device-opencl enabled device):  
<http://aplacetogeek.wordpress.com/android-with-opengl-tutorial/>
- An example of image manipulation with Sony Xperia Z1:  
[http://developer.sonymobile.com/knowledge-base/tutorials/android\\_tutorial/boost-the-performance-of-your-android-app-with-opengl/](http://developer.sonymobile.com/knowledge-base/tutorials/android_tutorial/boost-the-performance-of-your-android-app-with-opengl/)
- <http://software.intel.com/en-us/articles/opengl-basic-sample-for-android-os>
- <https://code.google.com/p/aopengl/>
- Google Play App to check if your device support openCL:  
<https://play.google.com/store/apps/details?id=idv.android.openglinfo&hl=lv>
- OpenCL on a Nexus 10:  
<http://www.openglblog.com/2013/02/opengl-on-nexus-10-part-1.html>

### Some resources:

- Future ARM snapdragon 610-615 8-core 64bit full openCL:  
<http://www.qualcomm.com/media/releases/2014/02/24/qualcomm-technologies-announces-worlds-first-commercial-64-bit-octa-core>
- Benchmark on Android **RenderScript** devices:  
<https://compubench.com/result.jsp?benchmark=compu20>
- GPUVerify: check kernel tool (CUDA/openCL)
  - <http://multicore.doc.ic.ac.uk/tools/GPUVerify/>
- The GPU Battle: Google inhibit OpenCL on its smartphones (Nexus)
  - Google stop supporting openCL on its Nexus devices  
<http://streamcomputing.eu/blog/2013-08-01/google-blocked-opengl-on-android-4-3/>  
<http://www.openglblog.com/2013/08/opengl-disabled-on-android.html>

## OpenCL Programming Resources

- AMD Developer Central, Programming in OpenCL  
<http://developer.amd.com/resources/heterogeneous-computing/opengl-zone/programming-in-opengl/>
- QUALCOMM Snapdragon (SoC) openCL Developer Guide  
<https://developer.qualcomm.com/forum/qdn-forums/mobile-technologies/mobile-gaming-graphics-optimization-adreno/27201>
- Intel openCL SDK 2014 (beta)  
<http://software.intel.com/en-us/vcsources/tools/opengl-sdk-beta>
- A list of SDK from StreamComputing:  
<http://streamcomputing.eu/knowledge/sdks/>
- OpenCL Studio  
<http://www.opengldev.com/>

## OpenGL & OpenCL interoperability

- Stackoverflow discussion:  
<http://stackoverflow.com/questions/9919650/advantage-of-opengl-interoperability-with-opengl>
- Apple site:

[https://developer.apple.com/library/mac/documentation/Performance/Conceptual/OpenCL\\_MacProgGuide/shareGroups/shareGroups.html](https://developer.apple.com/library/mac/documentation/Performance/Conceptual/OpenCL_MacProgGuide/shareGroups/shareGroups.html)

- Intel site:  
<http://software.intel.com/en-us/articles/opengl-and-opengl-sharing>

Learning Materials

- AMD OpenCL Entry-level tutorial  
<http://developer.amd.com/resources/heterogeneous-computing/opengl-zone/programming-in-opengl/>
- AMD openCL curse  
<http://developer.amd.com/resources/heterogeneous-computing/opengl-zone/training-events/opengl-course-introduction-to-opengl-programming/>

Releases

OpenCL 2.0 (18 Nov 2013)

Khronos received significant and thoughtful developer feedback from the provisional release of OpenCL 2.0, much of which has been adopted, or will be merged with emerging hardware capabilities as this state-of-the-art parallel programming platform continues to evolve,” said Neil Trevett, chair of the OpenCL working group, president of the Khronos Group and vice president of mobile content at NVIDIA. “OpenCL continues to gather momentum on desktop, mobile and embedded devices, including providing a unified programming environment for dynamically balancing diverse CPU, GPU, DSP and hardware resources in mobile SOCs for advanced use cases ranging from vision processing for Augmented Reality to physics simulation for mobile gaming.”

OpenCL 2.0 updates and additions include:

Shared Virtual Memory

Host and device kernels can directly share complex, pointer-containing data structures such as trees and linked lists, providing significant programming flexibility and eliminating costly data transfers between host and devices.

Nested Parallelism

Device kernels can enqueue kernels to the same device with no host interaction, enabling flexible work scheduling paradigms and avoiding the need to transfer execution control and data between the device and host, often significantly offloading host processor bottlenecks.

Generic Address Space

Functions can be written without specifying a named address space for arguments, especially useful for those arguments that are declared to be a pointer to a type, eliminating the need for multiple functions to be written for each named address space used in an application.

Images

Improved image support including sRGB images and 3D image writes, the ability for kernels to read from and write to the same image, and the creation of OpenCL images from a mip-mapped or a multi-sampled OpenGL texture for improved OpenGL interop.

C11 Atomics

A subset of C11 atomics and synchronization operations to enable assignments in one work-item to be visible to other work-items in a work-group, across work-groups executing on a device or for sharing data between the OpenCL device and host.

Pipes

Pipes are memory objects that store data organized as a FIFO and OpenCL 2.0 provides built-in functions for kernels to read from or write to a pipe, providing straightforward programming of pipe data structures that can be highly optimized by OpenCL implementers.

Android Installable Client Driver Extension

Enables OpenCL implementations to be discovered and loaded as a shared object on Android systems.

Public Demo available

Many projects involving openCL are private and not of public domain. I would like to list all public projects I'm aware of. Please feel free to point me to news projects, I will update this sections

1. Intel openCL demo on low-depth fluid simulation, with openGL rendering  
<http://software.intel.com/en-us/vcsample/samples/3d-fluid-simulation>

## Tools

- <http://technet.microsoft.com/en-us/sysinternals/default>

## Publications

1. SURVEY OF ACCELERATING APPLICATIONS ON MOBILE DEVICE BY GPU PARALLEL PROGRAMMING (18/3/2014)  
[http://web.engr.oregonstate.edu/~sunr/ece570\\_project/Fin\\_report.pdf](http://web.engr.oregonstate.edu/~sunr/ece570_project/Fin_report.pdf)
2. Efficient Mapping of Irregular C++ Applications to Integrated GPUs  
[http://cgo.org/cgo2014/wp-content/uploads/2013/05/Mapping\\_Irregular\\_GPUs.pdf](http://cgo.org/cgo2014/wp-content/uploads/2013/05/Mapping_Irregular_GPUs.pdf)

## Library

- ViennaCL (openCL/openMP/CUDA), dense and sparse linear algebra library, with some benchmarks: <http://viennacl.sourceforge.net/viennacl-about.html>
- AMD's Accelerated Parallel Processing Math Libraries (APPML):<http://developer.amd.com/tools/heterogeneous-computing/amd-accelerated-parallel-processing-math-libraries/>
- Jack Dongarra's MAGMA linear algebra library: <http://icl.cs.utk.edu/magma/software/>
- AccelerEyes ArrayFire math library: [http://www.accelereyes.com/arrayfire\\_tour](http://www.accelereyes.com/arrayfire_tour)
- VexCL library for OpenCL C++ vector arithmetic, reduction, and sparse matrix-vector multiplication:<http://ddemidov.github.com/vexcl/index.html>
- clpp, an OpenCL data parallel primitives library (reduction, parallel prefix sum etc):<http://code.google.com/p/clpp/>
- The Boost.Compute library provides a C++ STL-like API for OpenCL.  
Source code :<https://github.com/kylelutz/compute>  
documentation : <http://kylelutz.github.io/compute/>.
- **VirtualCL (VCL) cluster platform is a wrapper for OpenCL™ that allows most unmodified applications to transparently utilize multiple OpenCL devices in a cluster as if all the devices are on the local computer** [http://www.mosix.org/txt\\_vcl.html](http://www.mosix.org/txt_vcl.html)
- Simply openCL <https://code.google.com/p/simple-openc1/>

## List of openCL Projects

- A very comprehensive list of openCL projects  
<https://masterbranch.com/openc1-projects>

## Others

Computer vision papers     <http://www.cvpapers.com/index.html>

## General Multi-Threading Resources

1. Parallel STL (ISO C++ proposal) C++ Extensions for Parallelism
  - A. "Democratizing Parallelism in C++"  
<http://blogs.msdn.com/b/vcblog/archive/2014/04/16/parallel-stl-democratizing-parallelism-in-c.aspx>
  - B. CodePlex Parallel STL project is the Microsoft prototype of the Parallel STL proposal  
<https://parallelstl.codeplex.com/>
  - C. Working Draft, Technical Specification for C++ Extensions for Parallelism, Revision 1  
<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3960.pdf>

SHARE

0



2

Tweet



 Mi piace 0

Condividi