
1 What's New

New OpenCL samples.

- UnsharpMask – An Image Processing sample that implements an Unsharp mask filter.
- StringSearch – An optimized string search sample that demonstrates how to reduce divergence through workload re-balancing with a queue.

Updated OpenCL samples.

The following samples have been updated to use the `map` and `unmap` buffer calls: `clEnqueueMapBuffer()` and `clEnqueueUnmapMemObject()`.

- GlobalMemoryBandwidth
- BinarySearch
- BitonicSort
- RadixSort
- ScanLargeArrays
- Histogram
- HistogramAtomics
- QuasiRandomSequences
- MonteCarloAsian

The following samples have been updated to calculate and display performance measurement in a more standardized manner.

- BufferBandwidth
- GlobalMemoryBandwidth
- BinarySearch
- BitonicSort
- RadixSort
- ScanLargeArrays
- Histogram
- HistogramAtomics
- QuasiRandomSequences
- MonteCarloAsian

The following samples have an additional scalar version of the OpenCL kernel. The scalar version is more suitable for executing on the latest AMD GCN architecture GPUs.

- GlobalMemoryBandwidth
- Histogram
- HistogramAtomics
- QuasiRandomSequences
- MonteCarloAsian

The following sample now works for larger sample sizes.

- PrefixSum

New Bolt samples.

- BoltAMPIntro – A novice sample that shows how to sort an array of elements using the BOLT C++ AMP APIs.
- BlackScholesAMP – A sample that demonstrates the usage of the BOLT APIs for Black-Scholes Option Pricing calculation.
- BoltSort – A sample that demonstrates the usage of different Sort routines in the BOLT library.
- BoxFilterSAT – A sample that demonstrates the BoxFiltering method using the BOLT APIs and shows BOLT interop with raw OpenCL kernels.
- MonteCarloPI – A sample that demonstrates the Monte Carlo method of PI calculation using the BOLT APIs.
- StocksDataAnalysis – A sample that demonstrates the usage of the BOLT APIs for performing analytics computation on stocks data.

All of these BOLT samples include support to use the new execution paths available in BOLT 1.0 for Serial CPU and Multicore CPU in addition to OpenCL/C++ AMP.

Updated BOLT Samples.

- BlackScholes and RGBToYUV – The updated samples use the newly supported execution paths in Bolt for Serial CPU and Multicore CPU execution in addition to OpenCL
- All BOLT samples now include 32-bit support.

New OpenCV samples.

- OpenCV-CLIntro – An introductory sample that demonstrates how to use the OpenCV and OpenCV-CL functions in an OpenCV program
- CartoonFilter – A sample that uses Cartoon Filter as an example to show the different forms of interoperability between OpenCV, OpenCV-CL, and raw OpenCL kernels.

2 Samples

- AMD APP SDK samples are located in the `$(AMDAPPSDKSAMPLESROOT)\samples` folder. See the *Installation Notes* and the *Getting Started Guide* for more information. These documents also provide information about using OpenCL, BOLT, C++ AMP, Aparapi, and OpenCV samples.

3 Important Notes

- For a successful build and correct operation of individually downloaded samples, update to Catalyst 13.6 Beta2 driver or newer.
- When building on Linux systems, the GL samples require the development files for OpenGL and for the OpenGL utility library. If these are not already installed on your system, install them on your system in order for the sample to build. If required, create a symlink `libGLU.so` to `libGLU.so.1`.
- Support for Microsoft® Visual Studio® 2008 is deprecated.

4 Resolved Issues and Improvements

- Nbody – Optimized for performance.
- SDKUtil – Removed reference to `readme.txt` in the VS project file.
- BinomialOption and BinomialOptionGPU – Fixed the performance measurement output to display the correct sample size. Also, the samples now use the correct sample size as provided by the `-x` option.
- LDSBandwidth – Fixed a bug in the verification with CPU implementation
- DX11AMPInterop – Now supports the `-q` option.
- BlackScholes Aparapi – Now uses the "for" loop instead of the "range" parameter for kernel iterations.
- BinomialOptionMultiGPU and MonteCarloAsianMultiGPU – Removed improper error messages printed when the `--cpu` option is selected.
- OpenCLService – Now returns the proper exit codes and prints the appropriate error messages on failure.
- Updated all the samples to return with an appropriate exit code.
- TemplateC – Excluded this sample from the 2.8.1 release, as the sample is a duplicate of the `Template` sample.

5 Known Issues

- For correct operation of the samples when using RHEL 5.5 or RHEL 5.8, it may be necessary to build the samples using that OS.
- Mandelbrot: When using the vector version of the kernel, `LocalThread` is divided by four to ensure correct operation.
- SimpleDX10: This sample does not operate correctly when using the MinGW compiler.
- SimpleDX11: This sample does not operate correctly when using the MinGW compiler.
- When building on Linux systems, some samples require the development files for OpenGL and for the OpenGL utility library. This must be installed on your system for the sample to build. In some cases you may also have to create a symlink `libGLU.so` to `libGLU.so.1`.
- If GL samples fail on Linux, ensure that the `libGL.so.1` file is linked to `fglrx-libGL.so.1.2`, which for Ubuntu is found in `/usr/lib/fglrx/` and `/usr/lib32/fglrx/`, or for RHEL is found in `/usr/lib64/fglrx/`. The sample fails when linked to `/usr/lib/x86_64-linux-gnu/mesa/libGL.so.1`, `/usr/lib/i386-linux-gnu/mesa/libGL.so.1` in Ubuntu or

`/usr/lib/libGL.so.1`, `/usr/lib64/libGL.so.1` in RHEL. For more details, see:
<http://phoronix.com/forums/showthread.php?7351-Does-fglrx-s-libGL-so-1-2-have-wrong-soname>
<https://bugs.launchpad.net/ubuntu/+source/mesa/+bug/943162>

- The SDK samples provided with this release of the AMD APP SDK are not necessarily tuned for optimal performance. AMD is improving the samples continually; check <http://developer.amd.com/tools/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/samples-demos/> for new and updated samples.
- If you intend to use OpenCL on supported AMD GPUs, ensure that a supported display driver is installed on your system before running the SDK installer. You also can run the individual developer and samples `.msi` files from the location to which they are unzipped by the SDK installer in order to manually install any missing components. When running 32-bit samples executables on 64-bit Ubuntu systems, ensure that the `ia32-libs` package is installed. To do this, log in as root and type:

```
sudo apt-get install ia32-libs
```

If the `ia32-libs` package is not installed, running 32-bit samples executables on 64-bit Ubuntu systems can result in the following error message:

```
'clGetPlatformIDs() failed'
```

For more information about this issue, see: <http://www.debian-administration.org/articles/534>

- Running samples can result in an error if the `TEMP` environment variable contains multi-byte characters.
- The OpenCL NBody may report differences when run with the `--verify` option and a higher number of iterations due to precision differences between the OpenCL and C functions.
- The FluidSimulation2D and NBody samples are the only ones still using glut; thus, they may crash if the “X” button is used to close the application window. This is a known issue in glut64 for all Windows 64-bit operating systems.
- For MinGW64 issues, please read KnowledgeBase article KB125:
<http://developer.amd.com/resources/documentation-articles/knowledge-base/>
- Increased error tolerance due to reduced accuracy when using native versions of SIN and COS functions on the AMD Radeon™ HD 6970 and AMD Radeon™ HD 6950 relative to other AMD GPUs.
- Bolt samples, C++AMP samples, as well as SimpleDX10 and FluidSimulation2D cannot be built using the Intel C Compiler.
- Samples that include `cl.hpp` do not build on MinGW. These samples are in the `cpp_cl` folder.
- All samples in `$(AMDAPPSDKSAMPLESROOT)/samples/opengl/benchmark`, as well as GL-based, DirectX, Bolt, and MultiGPU samples do not work on CPU-only machines
- On a machine with [Intel/AMD CPU + integrated Intel/AMD GPU] + discrete AMD GPU, C++AMP samples do not run on the discrete GPU when the Power-Express(PX) mode is enabled. This limitation holds true for any DirectX samples as well. On a desktop, to enable running on discrete GPU, connect 2 monitors to both iGPU and dGPU ports. This way DirectX will detect both the cards in power-express mode. In case of a mobile device, such as a laptop, disable iGPU using the BIOS settings. This way DirectX will be use the dGPU.

- Executing samples on Linux using the CPU runtime reports the following message, but continues to execute as expected:

```
FATAL: Module fglrx not found.
Error! Fail to load fglrx kernel module! Maybe you can switch to root user
to load kernel module directly
```
- The execution of `GlobalMemoryBandwidth` might fail when the sample is run with vector width ≥ 4 (for example, using the command-line option `-c 4`) on a device on which the Maximum global memory allocation size is less than 132 MB. Executing with vector width `-c 1` or `-c 2` should work fine on such devices. `clinfo.exe` can be used to run and check the Max Memory Allocation limit on the target machine.
- The pre-built 32-bit `BoxFilterGL` sample under `samples\opencl\bin\x86` might fail when the sample is run with the `-sep` command-line option. Recompiling the sample and running it should resolve the issue.
- The `BinarySearch` sample gives incorrect results for data sizes larger than the default size of 512.
- The `QuasiRandomSequence` might give incorrect results on an OpenCL device whose preferred float vector width is greater than 4.
- The `UnsharpMask` sample might report differences between the C-implementation values and the OpenCL values when run on Linux with the `--verify` option.
- The `IntroStaticCPPKernel` sample fails on machines that do not have a GPU.
- The `MonteCarloAsianDP` sample might fail when run for a "maturity" value greater than 2.5 (specified using the `-m` command line option).
- The `Histogram` sample might report differences between the C-implementation values and the OpenCL values when run on Linux with the `--verify` option and the `-scalar` option.

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/opencforum



The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Copyright and Trademarks

© 2013 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, Radeon, FireStream, and combinations thereof are trademarks of Advanced Micro Devices, Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos. Other names are for informational purposes only and may be trademarks of their respective owners.
