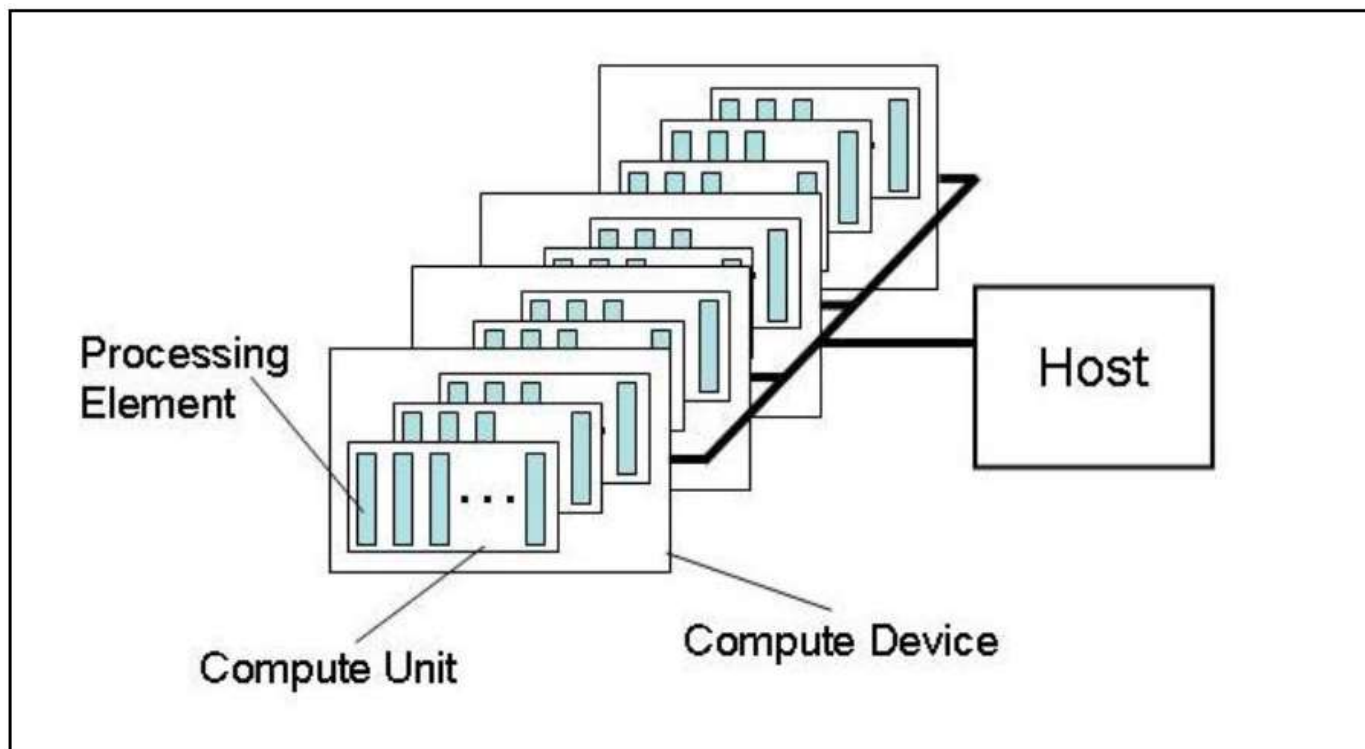


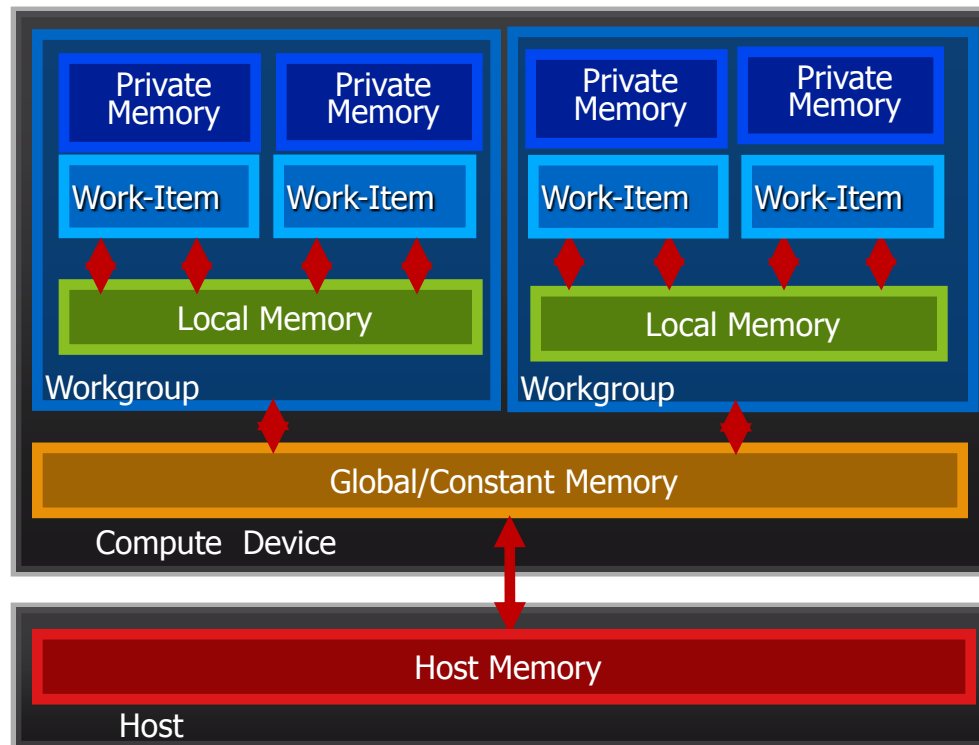
OpenCL Platform Model

- **One Host + one or more Compute Devices**
 - Each Compute Device is composed of one or more Compute Units
 - Each Compute Unit is further divided into one or more Processing Elements



OpenCL Memory Model

- **Private Memory**
 - Per work-item
- **Local Memory**
 - Shared within a workgroup
- **Global/Constant Memory**
 - Visible to all workgroups
- **Host Memory**
 - On the CPU

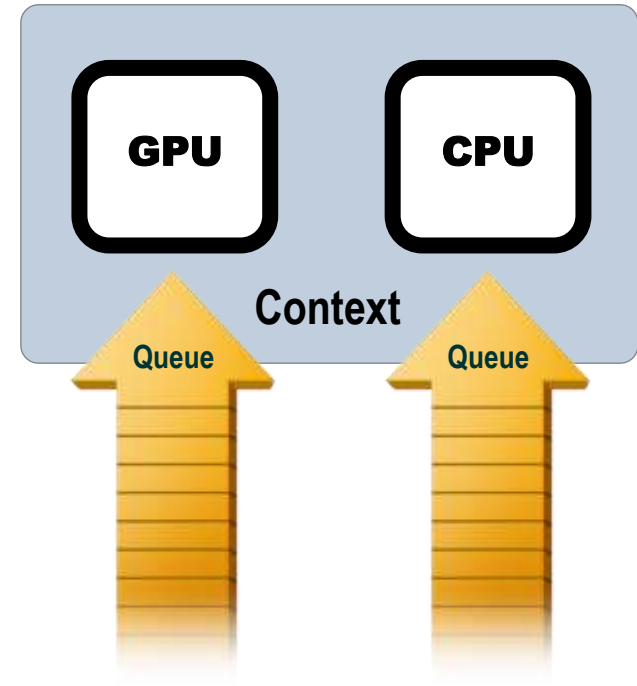


Memory management is Explicit

You must move data from host -> global -> local ... *and* back

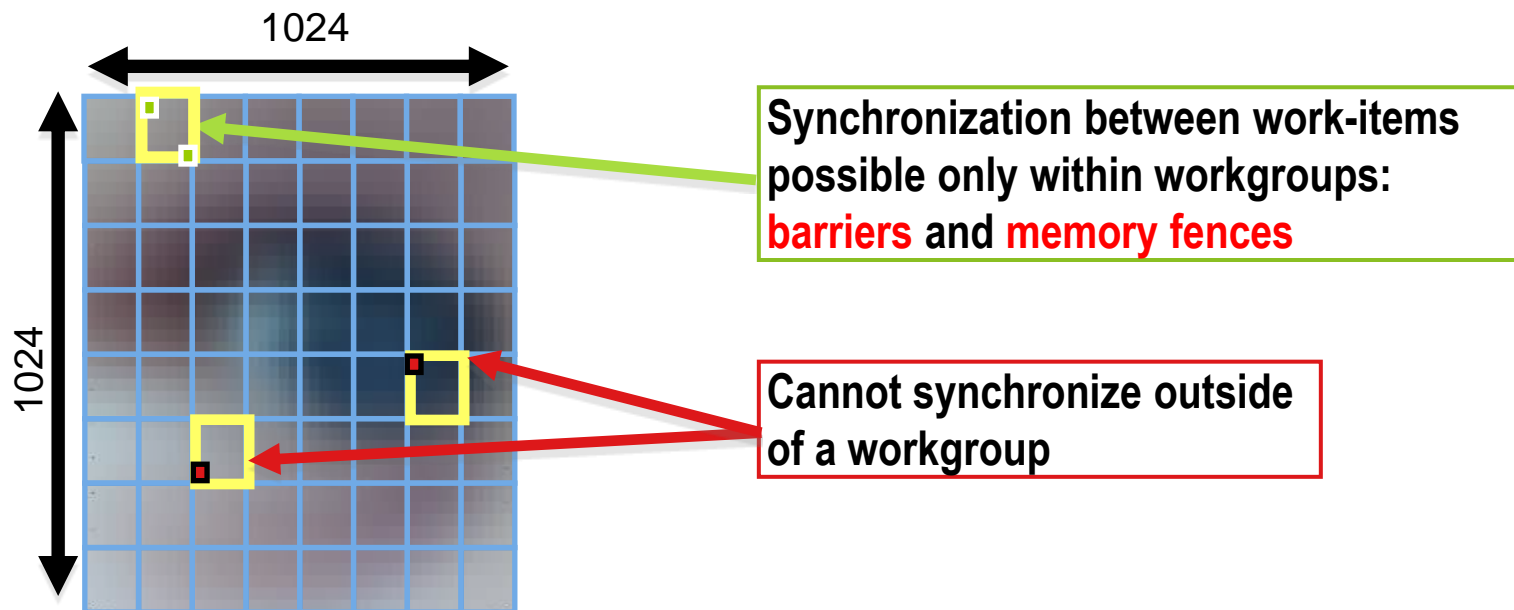
OpenCL Execution Model

- **OpenCL application runs on a host which submits work to the compute devices**
 - **Context:** The environment within which work-items executes ... includes devices and their memories and command queues
 - **Program:** Collection of kernels and other functions (Analogous to a dynamic library)
 - **Kernel:** the code for a work item.
Basically a C function
 - **Work item:** the basic unit of work on an OpenCL device
- **Applications queue kernel execution**
 - Executed in-order or out-of-order



An N-dimension domain of work-items

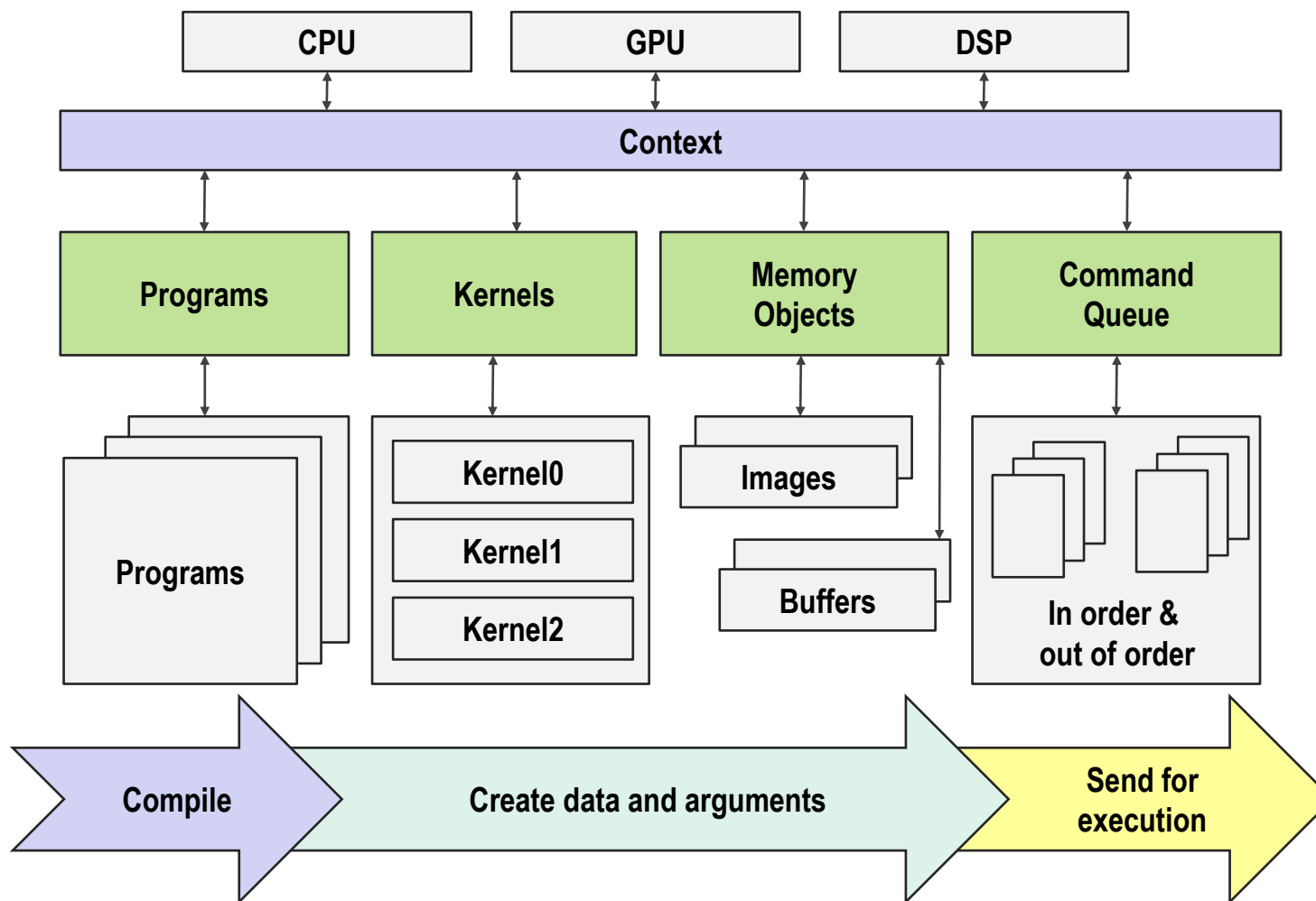
- Kernels executed across a global domain of *work-items*
- Work-items grouped into local *workgroups*
- Define the “best” N-dimensioned index space for your algorithm
 - Global Dimensions: 1024 x 1024 (whole problem space)
 - Local Dimensions: 128 x 128 (work group ... executes together)



Programming Kernels: OpenCL C

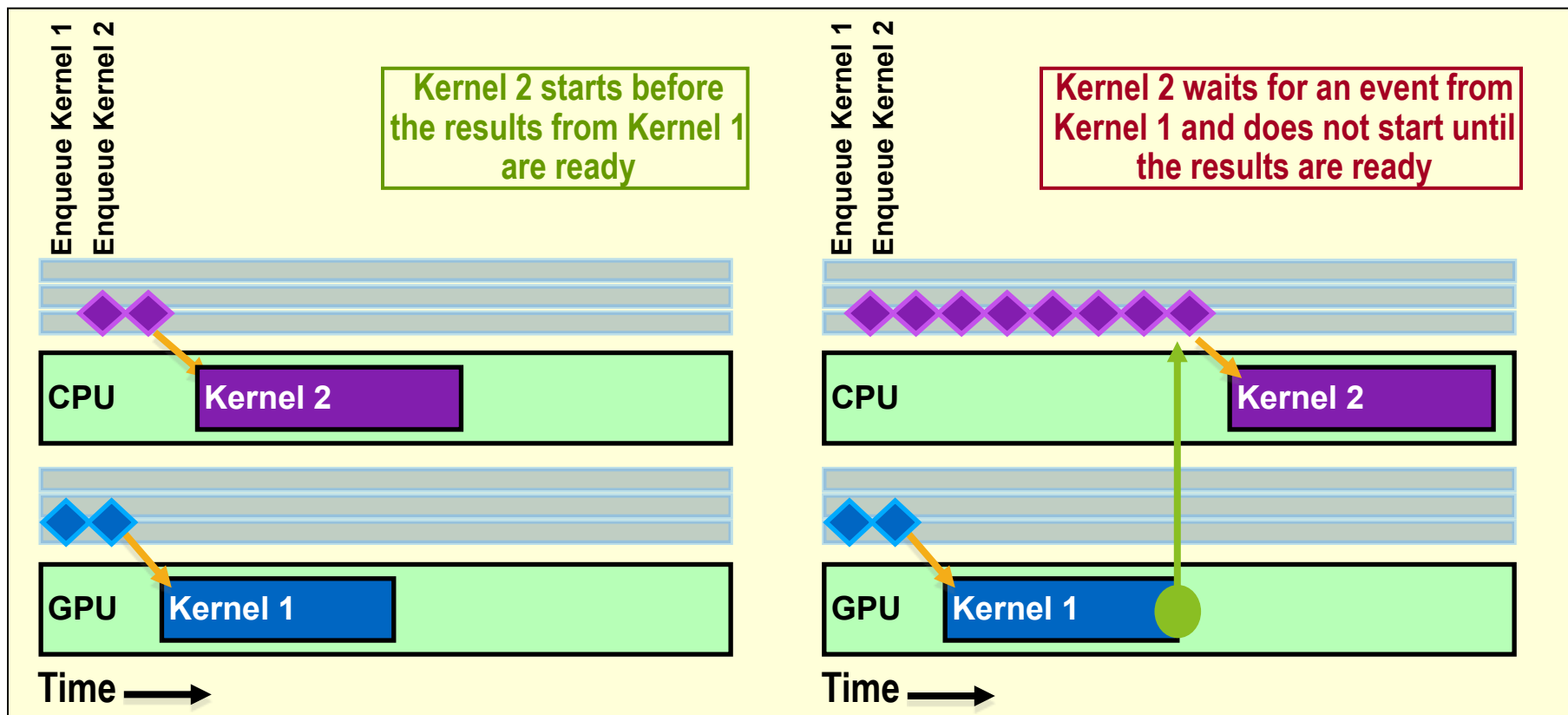
- **Derived from ISO C99**
 - But without some C99 features such as standard C99 headers, function pointers, recursion, variable length arrays, and bit fields
- **Language Features Added**
 - Work-items and workgroups
 - Vector types
 - Synchronization
 - Address space qualifiers
- **Also includes a large set of built-in functions**
 - Image manipulation
 - Work-item manipulation,
 - Math functions, etc.

Creating an OpenCL Program



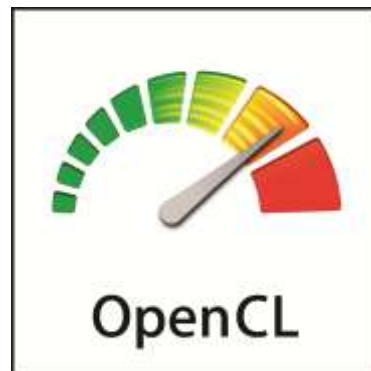
Synchronization: Queues & Events

- Events can be used to synchronize kernel executions between queues
- Example: 2 queues with 2 devices



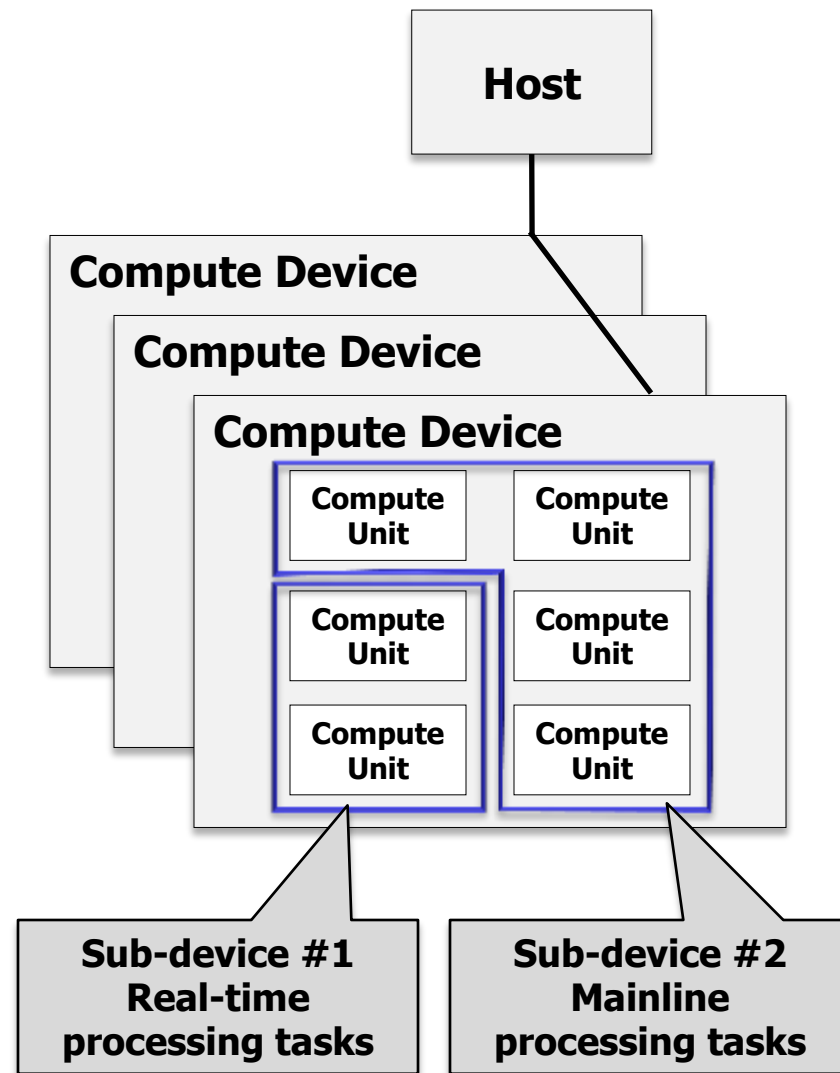
OpenCL 1.2 Announced in December

- **Significant updates - Khronos being responsive to developer requests**
 - Updated OpenCL 1.2 conformance tests available
 - Multiple implementations underway
- **Backward compatible upgrade to OpenCL 1.1**
 - OpenCL 1.2 will run any OpenCL 1.0 and OpenCL 1.1 programs
 - OpenCL 1.2 platform can contain 1.0, 1.1 and 1.2 devices
 - Maintains embedded profile for mobile and embedded devices



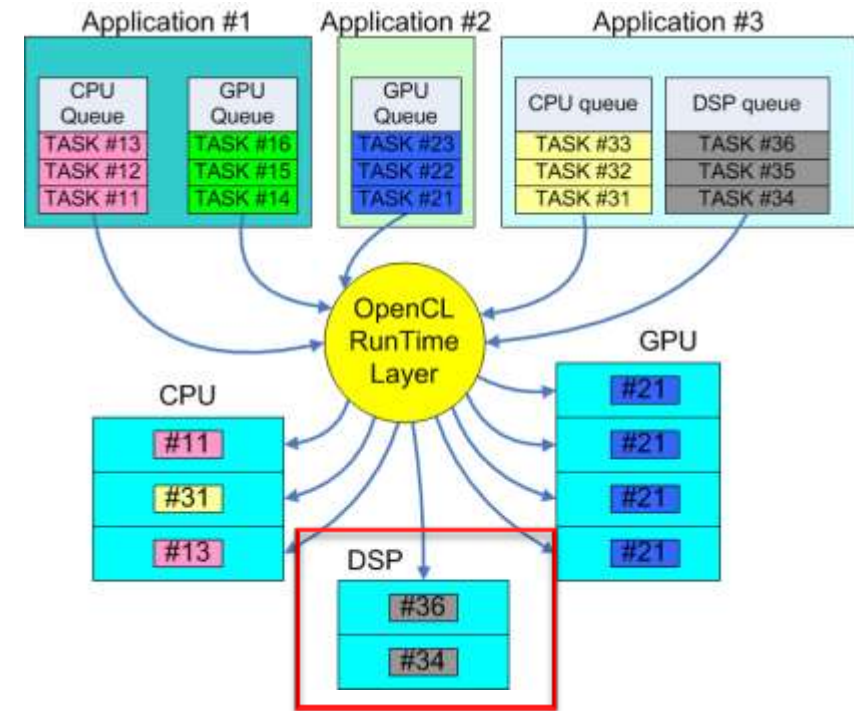
Partitioning Devices

- **Devices can be partitioned into sub-devices**
 - More control over how computation is assigned to compute units
- **Sub-devices may be used just like a normal device**
 - Create contexts, building programs, further partitioning and creating command-queues
- **Three ways to partition a device**
 - Split into equal-size groups
 - Provide list of group sizes
 - Group devices sharing a part of a cache hierarchy



Custom Devices and Built-in Kernels

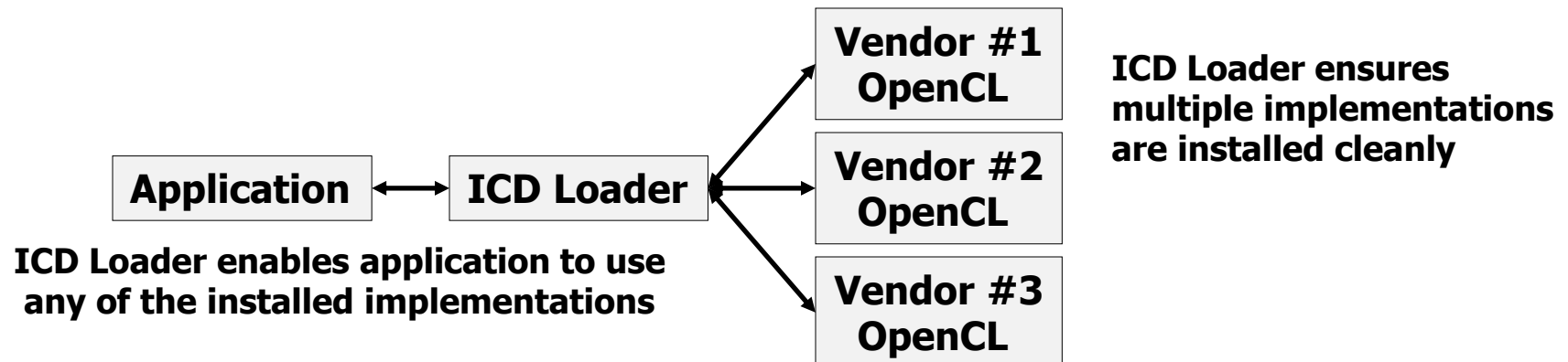
- **Embedded platforms often contain specialized hardware and firmware**
 - That cannot support OpenCL C
- **Built-in kernels can represent these hardware and firmware capabilities**
 - Such as video encode/decode
- **Hardware can be integrated and controlled from the OpenCL framework**
 - Can enqueue built-in kernels to custom devices alongside OpenCL kernels
- **FPGAs are one example of device that can expose built-in kernels**
 - Latest FPGAs can support full OpenCL C as well
- **OpenCL becomes a powerful coordinating framework for diverse resources**
 - Programmable and non-programmable devices controlled by one run-time



Built-in kernels enable control of specialized processors and hardware from OpenCL run-time

Installable Client Driver

- **Analogous to OpenGL ICDs in use for many years**
 - Used to handle multiple OpenGL implementations installed on a system
- **Optional extension**
 - Platform vendor will choose whether to use ICD mechanisms
- **Khronos OpenCL installable client driver loader**
 - Exposes multiple separate vendor installable client drivers (Vendor ICDs)
- **Application can access all vendor implementations**
 - The ICD Loader acts as a de-multiplexor



Other Major New Features in OpenCL 1.2

- **Separate compilation and linking of objects**
 - Provides the capabilities and flexibility of traditional compilers
 - Create a library of OpenCL programs that other programs can link to
- **Enhanced Image Support**
 - Added support for 1D images, 1D & 2D image arrays
 - OpenGL sharing extension now enables an OpenCL image to be created from an OpenGL 1D texture, 1D and 2D texture arrays
- **DX9 Media Surface Sharing**
 - Efficient sharing between OpenCL and DirectX 9 or DXVA media surfaces
- **DX11 surface sharing**
 - Efficient sharing between OpenCL and DirectX 11 surfaces
- **And many other updates and additions..**

OpenCL Desktop Implementations

- <http://developer.amd.com/zones/OpenCLZone/>
- <http://software.intel.com/en-us/articles/openccl-sdk/>
- <http://developer.nvidia.com/openccl>

OpenCL™ Zone

Home > Zones > OpenCL™ Zone

OpenCL™ Zone

OpenCL™ (Open Computing Language) is the first truly open and royalty-free programming standard for general-purpose computations on heterogeneous systems. OpenCL™ allows programmers to preserve their expensive source code investment and easily target multi-core CPUs, GPUs, and the new APUs.

ATI Stream-enabled Software Applications

ATI Stream Processors

ATI Stream Accelerators

Developed in an open standards committee with representatives from major industry vendors, OpenCL™ gives users what they have been demanding: a cross-vendor, non-proprietary solution for accelerating their applications on CPU/GPU/Server GPUs.

VISUAL COMPUTING DEVELOPER COMMUNITY

Intel® OpenCL SDK

Download Now

OpenCL

About Intel® OpenCL SDK

About OpenCL™

OpenCL™ (Open Computing Language) is the first open, royalty-free standard for general-purpose parallel programming of heterogeneous systems. OpenCL provides a uniform programming environment for software developers to write efficient, portable code for client computer systems, high-performance computing servers, and handheld devices using a diverse mix of multi-core CPUs and other parallel processors.

About Intel® OpenCL SDK 1.1

Intel® OpenCL SDK 1.1 is Intel's implementation of the OpenCL standard optimized for Intel processors, running on Microsoft® Windows® and Linux® operating systems. This SDK implementation is fully conformant with the OpenCL 1.1 specification for the CPU, and with Microsoft® Windows® 7 operating systems.

Developers are now able to use the Intel® OpenCL SDK to create and distribute OpenCL based applications optimized for Intel® Core™ and Intel® Xeon® processors.

Technical Content

Getting Started

- [Announce Intel® OpenCL SDK 1.1 News!](#)
- [Release Notes](#)
- [Installation notes](#)
- [Intel® OpenCL SDK 1.1 FAQ](#)
- [Intel® OpenCL SDK User Guide](#)

Support and Feedback

Intel® OpenCL SDK FAQ (Frequently Asked Questions)

Answers to the most common questions asked by OpenCL developers.

Forums - Get answers to your questions about Intel® OpenCL SDK from Intel engineers and other OpenCL developers.

NVIDIA DEVELOPER ZONE

DEVELOPER CENTERS TECHNOLOGIES TOOLS RESOURCES COMMUNITY

OpenCL

OpenCL™ (Open Computing Language) is a low-level API for heterogeneous computing that runs on CUDA architecture GPUs. Using OpenCL, developers can write compact kernels using a C-like programming language to harness the massive parallel computing power of NVIDIA GPUs to create compelling computing applications. As the OpenCL standard matures and is supported on processors from other vendors, NVIDIA will continue to provide the drivers, tools and training resources developers need to create GPU accelerated applications.

In partnership with NVIDIA, OpenCL was submitted to the Khronos Group by Apple in the summer of 2008 with the goal of forging a cross platform environment for general purpose computing on GPUs. NVIDIA has chaired the industry working group that defines the OpenCL standard since its inception and shipped the world's first conformant GPU implementation of OpenCL for both Windows and Linux in June 2009.

NVIDIA has been delivering OpenCL support in end-user production drivers since October 2009, supporting OpenCL on all 300,000,000+ CUDA architecture GPUs shipped since 2006. OpenCL v1.1 support is included in publicly available NVIDIA driver version 280.63 or later on the [driver download page](#).

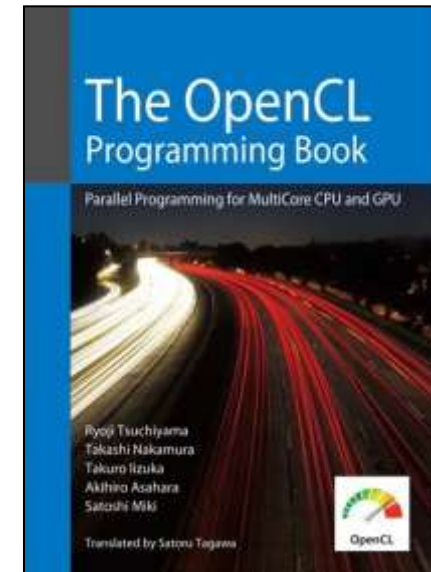
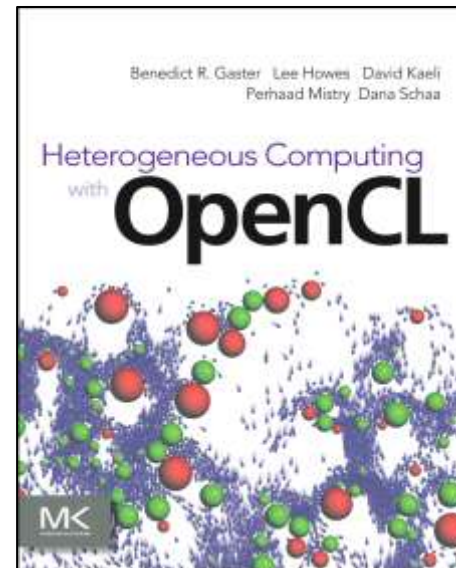
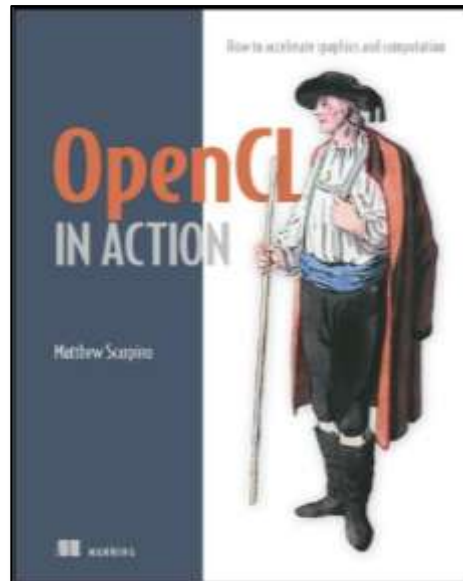
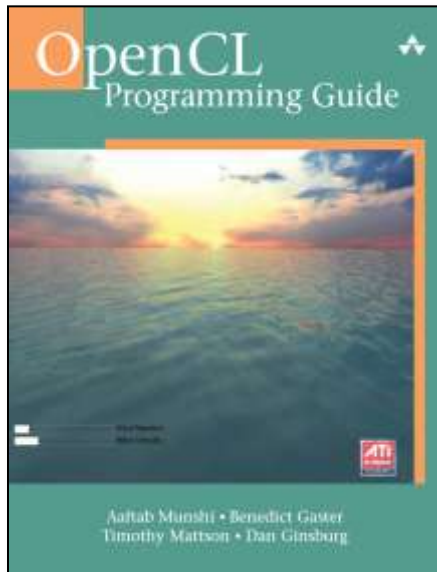
- For OpenCL v1.1 support on Windows Server, use the Windows 7 drivers
- Windows XP drivers with OpenCL v1.1 support are available for GeForce desktop products only

NVIDIA also provides powerful performance analysis tools for OpenCL developers, including NVIDIA [ParallelView](#) for Visual Studio and NVIDIA [Nvprof](#) for Linux and Mac OS.

On the same day Khronos Group announced the new OpenCL v1.1 specification update (June 14th, 2010), NVIDIA released OpenCL v1.1 pre-release drivers and SDK code samples to all GPU Computing registered developers. [Log in](#) or [apply for an account](#) to download the NVIDIA OpenCL v1.1 drivers and SDK.

OpenCL Books – Available Now!

- **OpenCL Programming Guide - The “Red Book” of OpenCL**
 - <http://www.amazon.com/OpenCL-Programming-Guide-Aaftab-Munshi/dp/0321749642>
- **OpenCL in Action**
 - <http://www.amazon.com/OpenCL-Action-Accelerate-Graphics-Computations/dp/1617290173/>
- **Heterogeneous Computing with OpenCL**
 - <http://www.amazon.com/Heterogeneous-Computing-with-OpenCL-ebook/dp/B005JRHYUS>
- **The OpenCL Programming Book**
 - <http://www.fixstars.com/en/opencil/book/>



Spec Translations

- **Japanese OpenCL 1.1 spec translation available today**
 - <http://www.cutt.co.jp/book/978-4-87783-256-8.html>
 - Valued partnership between Khronos and CUTT in Japan
- **Working on OpenCL 1.2 specification translations**
 - Japanese, Korean and Chinese



Khronos OpenCL Resources

- **OpenCL is 100% free for developers**
 - Download drivers from your silicon vendor

- **OpenCL Registry**

- www.khronos.org/registry/cl/

- **OpenCL 1.2 Reference Card**

- PDF version
 - <http://www.khronos.org/files/ocl-1-2-quick-reference-card.pdf>

- **Online Man pages**

- <http://www.khronos.org/registry/cl/sdk/1.2/docs/man/xhtml/>

- **OpenCL Developer Forums**

- Give us your feedback!
 - www.khronos.org/message_boards/



