Introduction to OpenCL

Open Computing Language

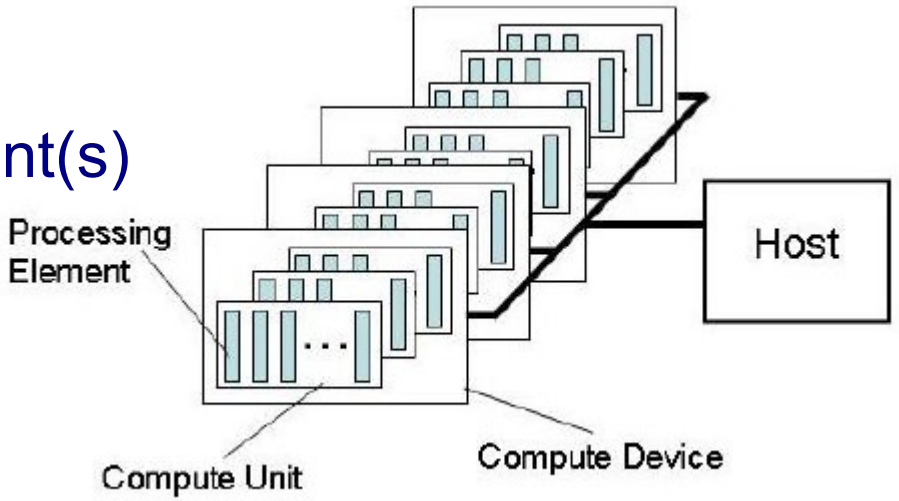
- OpenCL is an open royalty-free standard for general purpose parallel programming across CPUs, GPUs and other processors, giving software developersportable and efficient access to the power of these heterogeneous processing platforms [khronos]
  - Platform model
  - Memory model
  - Execution model
  - Programming model

Terminology

- Framework
  - set of components supporting SW development and execution
  - libraries, APIs, runtime systems, compilers, …
- Application
  - Program running on host and (OpenCL) devices

OpenCL platform model

- One Host + Compute Devices
  - Managed by OpenCL framework
    - Sharing of resources
    - Execution of kernels on devices
- Compute Device
  - Compute Unit(s)
    - Processing Element(s)



OpenCL execution model

- Device program - Kernels
- Host program
  - Context and kernel execution management
- kernel execution → index space → work item
- Work item
  - Instance of a kernel for a particular index
  - Identified by point in index space
    - Global ID

OpenCL execution model

- kernel execution → index space → work item
- Work group – block of work-items
  - ID – same dimension as work items
  - Work item – local ID within group
  - All work items execute concurrently on single compute unit
- Index space = NDRange
  - N - 1D,2D,3D

Execution model

Context and command queues

- Context defined by host
  - devices
  - kernels
  - program objects
  - memory objects

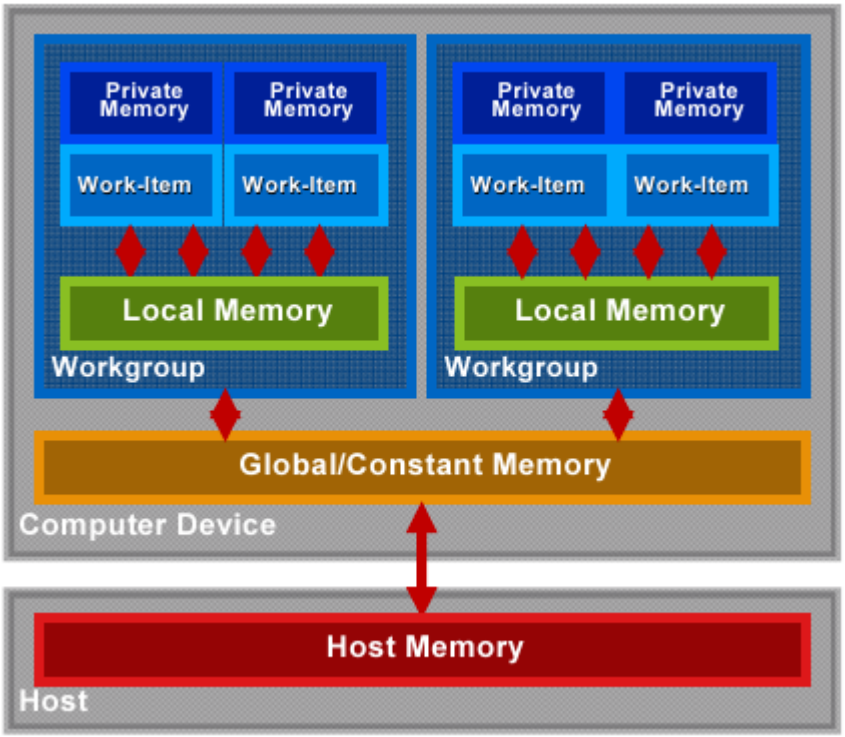
Execution model

Context and command queues

- Commands placed into command queues
  - Kernel execution commands
  - Memory commands
  - Synchronization commands
  - Queue execution order
    - in-order
    - out-of-order
- Categories of kernels
  - OpenCL kernels
  - Native kernels

Memory model

- Global
- Constant
- Local
  - shared within work-group
- Private
  - private to work-item



Memory model

- Host and device memory independent
  - Data copying
  - Mapping/Unmapping regions of memory object

Programming model

- Data parallel
  - Sequence of instructions on multiple elements of memory object
- Task parallel
- Synchronization
  - Work-item in single work-group
  - Commands in command queues in single context

OpenCL framework

- Platform Layer
  - Allows host program to discover devices, their capabilities and create contexts
- Runtime
  - Allows to manipulate contexts
- Compiler
  - Creates program executables with OpenCL kernels
  - OpenCL C – subset of ISO C99 + extensions for parallelism

Memory objects

- Buffer objects
  - 1D collection of elements
    - Arbitrary elements
    - Stored sequential
    - Accessible via pointers
    - Kernel element format same as buffer element
- Image objects
  - 2D or 3D texture, framebuffer, image
    - Pre-defined types of element
    - Accessible via built-in functions
    - Kernel element format is 4D float/integer vector

Sampler objects

- Description how to sample image object read by kernel
  - Out-of-range sampling
  - Filtering
  - Use of normalized coordinates
- Create sampler for context
  - clCreateSampler

Getting started

- Initialization
- Creating of memory objects
- Transfering (input) data
- Execution
- Synchronization
- Transfering (output) data
- Cleanup

Getting started
initialization

- Get platform
  - clGetPlatformIDs
- Get devices for platform
  - clGetDeviceIDs
- Create context for devices
  - clCreateContext
- Create command queue on a device within context
  - clCreateCommandQueue

Getting started
create memory objects

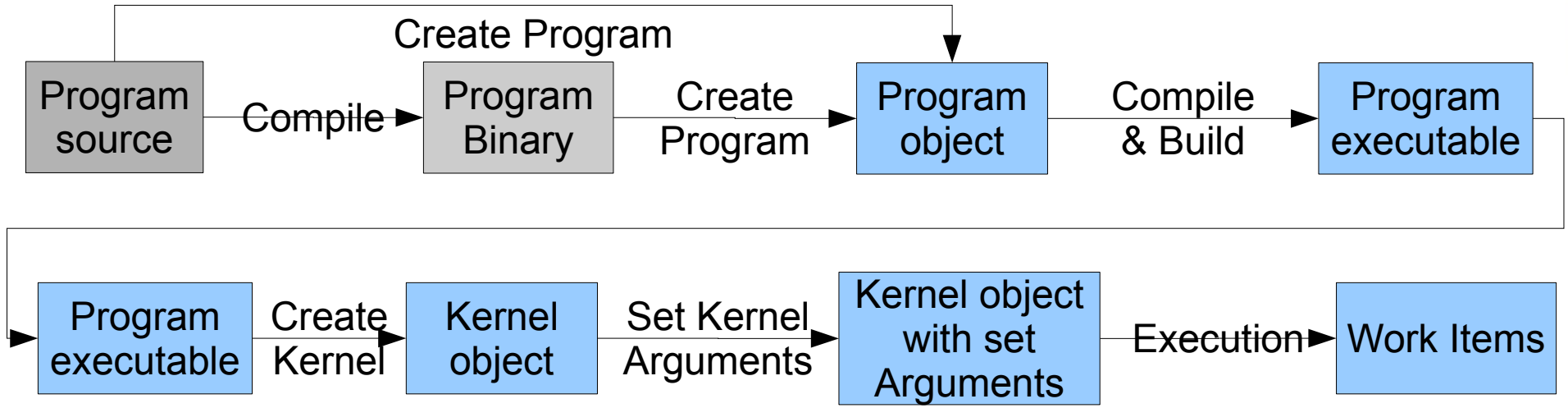
- Create Buffer object for context
  - clCreateBuffer
- Create Image object for context
  - clCreateImage2D
  - clCreateImage3D

Getting started
transfer data

- Read/Write/Copy Buffer/Image
  - clEnqueueRead/Write/Copy Buffer/Image
  - Copy between buffer and image
    - clEnqueueCopyBufferToImage
    - clEnqueueCopyImageToBuffer
- Map/Unmap Buffer/Image
  - clEnqueueMapBuffer/Image
  - clEnqueueUnmapMemObject

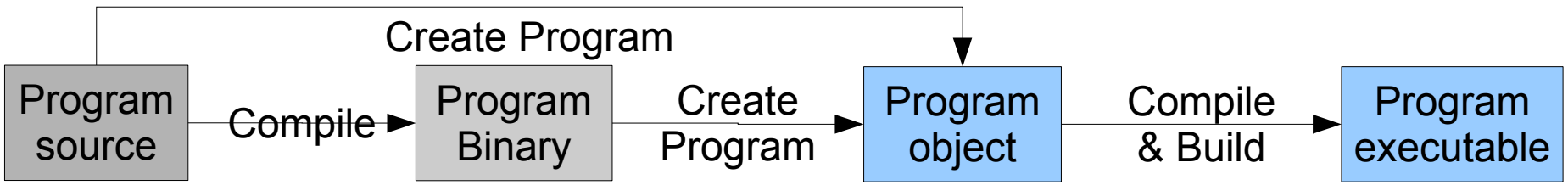
Execution overview

- Program source/binary, object, executable
- Kernel object
  - Create, Set arguments, Execute



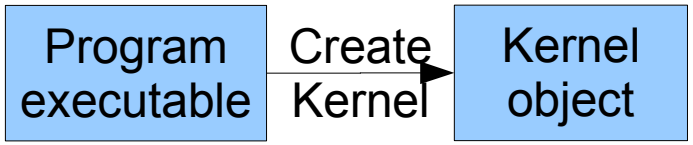
Program objects

- Create program for context and load source code/binary
  - clCreateProgramWithSource/Binary
- Compile and link program executable from source or binary for specified devices
  - clBuildProgram



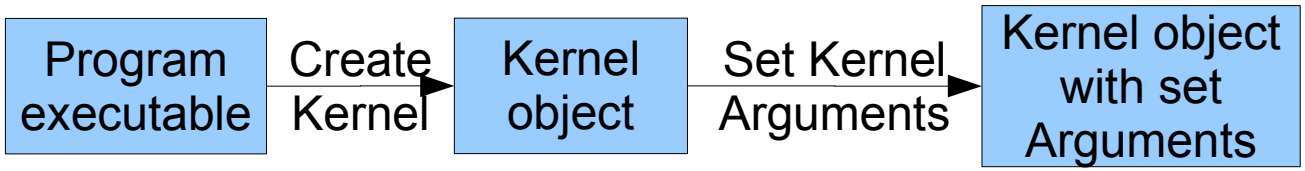
Kernel objects

- Create kernel object for a kernel within program
  - clCreateKernel
- Create kernel objects for all kernels of a program
  - clCreateKernelsInProgram



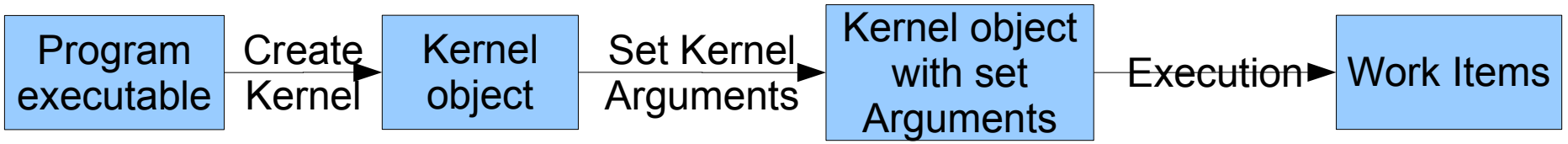
Kernel arguments

- Set kernel argument by index
  - clSetKernelArg



Kernel execution

- Enqueue execution of a kernel on a NDRange
  - ClEnqueueNDRangeKernel
- Enqueue execution of a single instance kernel
  - clEnqueueTask
- Enqueue execution of a native C/C++ function
  - clEnqueueNativeKernel



OpenCL C Language

- Data types
  - Scalar/Vector (2,4,8,16)
  - image2d_t/3d_t, sampler_t, event_t
- Adress space qualifiers
  - __global, __local, __constant, __private
- Image access qualifiers
  - __read_only, __write_only
- Function qualifiers
  - __kernel

OpenCL C Language

- get_work_dim
- get_global_size
- get_global_id
- get_local_size
- gel_local_id
- get_num_groups
- get_group_id
- barrier

Summary

- Open standard supported by various vendors
  - Apple, Nvidia, ATI,
- Independent of HW

References

- The OpenCL Specification, khronos
- OpenCL parallel computing for heterogeneous devices, khronos
- OpenCL API 1.0 quick reference card, khronos
- ATI stream SDK, ati