

OpenCL Events



**Oregon State
University**

Mike Bailey

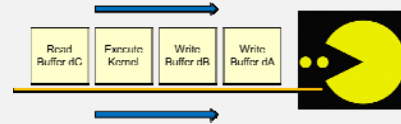
mjb@cs.oregonstate.edu



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



**Oregon State
University**
Computer Graphics

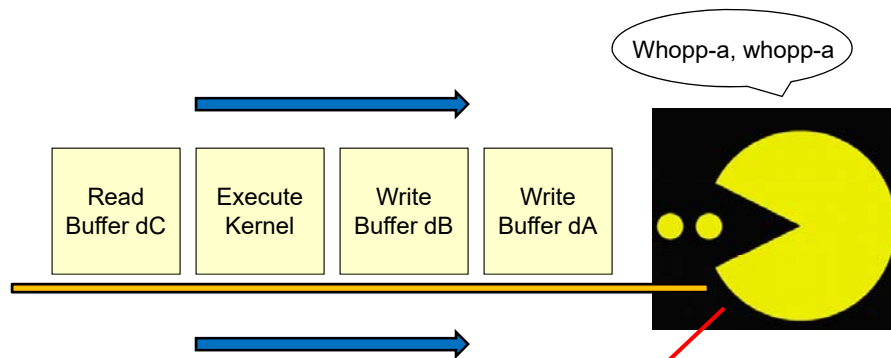


opencl.events.pptx

mjb - April 7, 2020

OpenCL Events

An event is an object that communicates the status of OpenCL commands



**Oregon State
University**
Computer Graphics

mjb - April 7, 2020

From the OpenCL Notes: 11. Enqueue the Kernel Object for Execution

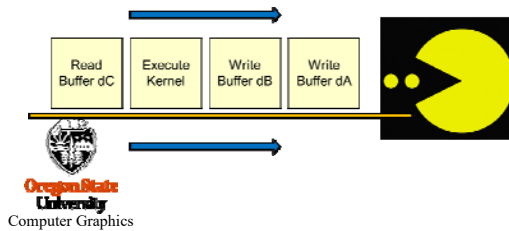
3

```
size_t globalWorkSize[ 3 ] = { NUM_ELEMENT, 1, 1 };
size_t localWorkSize[ 3 ] = { LOCAL_SIZE, 1, 1 };

status = clEnqueueNDRangeKernel( cmdQueue, kernel, 1, NULL, globalWorkSize, localWorkSize, 0, NULL, NULL );
```

```
status = clEnqueueNDRangeKernel( cmdQueue, kernel, 1, NULL, globalWorkSize, localWorkSize, 0, NULL, NULL );
```

events event object
 ↓ ↓
 event wait list



mjb - April 7, 2020

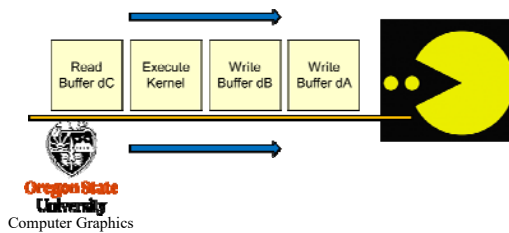
Creating an Event

4

```
cl_event waitKernelC;
```

```
status = clEnqueueNDRangeKernel( cmdQueue, kernel, 1, NULL, globalWorkSize, localWorkSize, 0, NULL, &waitKernelC );
```

event being created
 ↓
 event(s) to wait for



mjb - April 7, 2020

Waiting for Events

5

```
cl_event waitKernelA, waitKernel B.  
  
...  
  
cl_event dependencies[ 2 ];  
dependencies[ 0 ] = waitKernelA;  
dependencies[ 1 ] = waitKernelB;  
  
status = clEnqueueNDRangeKernel( cmdQueue, kernelC, 1, NULL, globalWorkSize, localWorkSize, 2, dependencies, NULL );
```

event being
created

event(s) to wait for



mjb - April 7, 2020

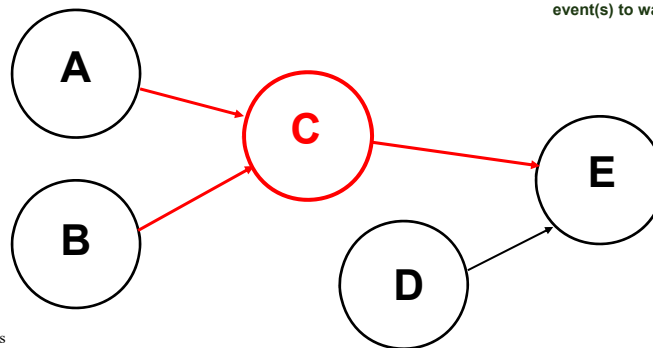
Creating an Execution Graph Structure

6

```
cl_event waitKernelC.  
  
...  
  
cl_event dependencies[ 2 ];  
dependencies[ 0 ] = waitKernelA;  
dependencies[ 1 ] = waitKernelB;  
  
status = clEnqueueNDRangeKernel( cmdQueue, kernelC, 1, NULL, globalWorkSize, localWorkSize, 2, dependencies, &waitKernelC );
```

event being
created

event(s) to wait for



mjb - April 7, 2020

Waiting for One Event

7

```
cl_event waitKernelA, waitKernel B.  
  
...  
  
status = clEnqueueNDRangeKernel( cmdQueue, kernelC, 1, NULL, globalWorkSize, localWorkSize, 1, &waitKernelA, NULL );
```

event(s) to wait for



mjb - April 7, 2020

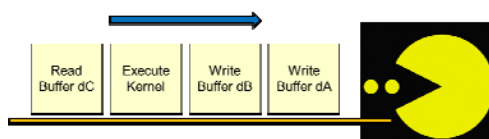
Placing a Barrier in the Command Queue

8

```
status = clEnqueueBarrier( cmdQueue );
```

Note: this *cannot* throw its own event

This does not complete until all commands enqueued before it have completed.



mjb - April 7, 2020

Placing an Event Marker in the Command Queue

9

```
cl_event waitMarker;  
status = clEnqueueMarker( cmdQueue, &waitMarker );
```

Note: **this can throw its own event**

This does not complete until all commands enqueued before it have completed.

This is just like a barrier, but it can throw an event to be waited for.



mjb - April 7, 2020

Waiting for Events Without Enqueuing Another Command

10

```
status = clWaitForEvents( 2, dependencies );
```


event(s) to wait for

This **blocks** until the specified events are thrown, so use it carefully!



mjb - April 7, 2020

I Like Synchronizing Things This Way

11

// wait until all queued tasks have taken place:

```
void
Wait( cl_command_queue queue )
{
    cl_event wait;
    cl_int status;

    status = clEnqueueMarker( queue, &wait );
    if( status != CL_SUCCESS )
        fprintf( stderr, "Wait: clEnqueueMarker failed\n" );

    status = clWaitForEvents( 1, &wait );    // blocks until everything is done!
    if( status != CL_SUCCESS )
        fprintf( stderr, "Wait: clWaitForEvents failed\n" );
}
```



Call this before starting the timer, before ending the timer, and before retrieving data from an array computed in an OpenCL program.

Computer Graphics

mjb - April 7, 2020

Getting Event Statuses Without Blocking

12

CL_EVENT_COMMAND_QUEUE
CL_EVENT_CONTEXT
CL_EVENT_COMMAND_TYPE
CL_EVENT_COMMAND_EXECUTION_STATUS

Specify one of these

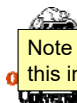
```
cl_int eventStatus;

status = clGetEventInfo( waitKernelC, CL_EVENT_COMMAND_EXECUTION_STATUS, sizeof(cl_int),
    &eventStatus, NULL );
```

CL_EVENT_COMMAND_EXECUTION_STATUS
returns one of these

CL_QUEUED
CL_SUBMITTED
CL_RUNNING
CL_COMPLETE

cl_int is what type
CL_EVENT_COMMAND_EXECUTION_STATUS
returns



Note that this is a nice way to check on event statuses without blocking. Thus, you could put this in a loop and go get some other work done in between calls.

Computer Graphics

mjb - April 7, 2020