Heterogeneous Parallel Programming

Lecture 7.3

Related Programming Models
OpenCL Host Code Part 1

Wen-mei Hwu – University of Illinois at Urbana-Champaign
Objective

• To learn to write OpenCL host code
  • Create OpenCL context
  • Create work queues for task parallelism
  • Device memory Allocation
  • Kernel compilation
  • Kernel launch
  • Host-device data copy
OpenCL Context

- Contains one or more devices
- OpenCL memory objects are associated with a context, not a specific device
- `clCreateBuffer()` is the main data object allocation function
  - Error if an allocation is too large for any device in the context
- Each device needs its own work queue(s)
- Memory copy transfers are associated with a command queue (thus a specific device)
OpenCL Context Setup Code (simple)

```c
cl_int clerr = CL_SUCCESS;
cl_context clctx = clCreateContextFromType(0,
                                          CL_DEVICE_TYPE_ALL, NULL, NULL, &clerr);

size_t parmsz;
clerr = clGetContextInfo(clctx, CL_CONTEXT_DEVICES, 0,
                         NULL, &parmsz);

cl_device_id* cldevs = (cl_device_id *) malloc(parmsz);
clerr = clGetContextInfo(clctx, CL_CONTEXT_DEVICES, parmsz,
                          cldevs, NULL);

cl_command_queue clcmdq = clCreateCommandQueue(clctx,
                                             cldevs[0], 0, &clerr);
```
OpenCL Kernel Compilation: vadd

const char* vaddsrc =

"__kernel void vadd(__global float *d_A, __global float *d_B, __global float *d_C, int N) {
[…etc and so forth…]

cl_program clpgm;
clpgm = clCreateProgramWithSource(clctx, 1, &vaddsrc, NULL, &clerr);

char clcompileflags[4096];
sprintf(clcompileflags, "-cl-mad-enable");
clerr = clBuildProgram(clpgm, 0, NULL, clcompileflags, NULL, NULL);
cl_kernel clkern = clCreateKernel(clpgm, "vadd", &clerr);
OpenCL Device Memory Allocation

- **clCreateBuffer();**
  - Allocates object in the device **Global Memory**
  - Returns a pointer to the object
  - Requires five parameters
    - OpenCL context pointer
    - Flags for access type by device (read/write, etc.)
    - Size of allocated object
    - Host memory pointer, if used in copy-from-host mode
    - Error code

- **clReleaseMemObject()**
  - Frees object
    - Pointer to freed object
OpenCL Device Memory Allocation (cont.)

- Code example:
  - Allocate a 1024 single precision float array
  - Attach the allocated storage to d_a
  - “d_” is often used to indicate a device data structure

VECTOR_SIZE = 1024;
cl_mem d_a;
int size = VECTOR_SIZE* sizeof(float);

    d_a = clCreateBuffer(clctx, CL_MEM_READ_ONLY, size, NULL, NULL);

    ...

    clReleaseMemObject(d_a);
OpenCL Device Command Execution

Application → Command → Cmd Queue

Cmd Queue

OpenCL Device

OpenCL Context
TO LEARN MORE, READ SECTION 14.4-14.5