Сравнение CUDA и OpenCL

Дмитрий Микушин, Александр Харламов

OpenCL

- Кроссплатформенный стандарт (CPU, GPU, Cell, ...)
- Фокус на portability, в меньшей степени – на производительности
- Для разных платформ в стандартном коде требуются различные исправления и оптимизации

CUDA и OpenCL: терминология

Поток (thread)	Рабочий элемент (work-item)
Блок потоков (thread block)	Рабочая группа (work-group)
Сеть (grid)	N-мерное пространство индексов (ND-range index space)
Ядро (kernel)	Ядро (kernel)

CUDA и OpenCL: спецификаторы

__global__	__kernel
__host__	N/A
__device__	N/A

CUDA и OpenCL: память

__device__	__global
__shared__	__local
__constant__	__constant
local	__private

OpenCL – пример

```
cl_context ctx;
cl_command_queue cmd_q;
cl_program program;
cl_kernel kernel;
cl_device_id * pDevId = NULL;


ctx = clCreateContextFromType(0,CL_DEVICE_TYPE_GPU,0,0,0);

clGetContextInfo(ctx,CL_CONTEXT_DEVICES,0,0,&dev_cnt);
clGetContextInfo(ctx,CL_CONTEXT_DEVICES,dev_cnt,pDevId,0);

cmd_q= clCreateCommandQueue(ctx,pDevId[0],0,0);

program = clCreateProgramWithSource(ctx,1,pText,0,0);
clBuildProgram(program, 0,0,0,0,0);

kernel = clCreateKernel(program, "simple", 0);
```

OpenCL – пример

```
cl_mem mem = clCreateBuffer(ctx,CL_MEM_WRITE_ONLY,
                            N*sizeof(float),0,0);

clSetKernelArg(kernel, 0, sizeof(cl_mem),(void*) &mem);
clSetKernelArg(kernel, 1, sizeof(int),   (void*) &N);

clEnqueueNDRangeKernel(cmd_q,kernel,1,0,&N,&N,0,0,0);

clEnqueueReadBuffer(cmd_q, mem, CL_TRUE, 0,
                    N*sizeof(float), pData,0,0,0);

clReleaseMemObject(mem);
clReleaseKernel(kernel);
clReleaseProgram(program);
clReleaseCommandQueue(cmd_q);
clReleaseContext(ctx);
```