Pre-compile the OpenCL Kernel Program - Part 1

Fri 21 November 2014
Tags opencl
Posted by Logan

In the previous post, we have written a simple vector addition OpenCL program. We were compiling the OpenCL kernel program from source code at run-time, thus we have to distribute the OpenCL source code to our users.

However, in some cases, we may prefer to **pre-compile the OpenCL kernel program**. For example:

1. It might take too long to compile the kernel functions.
2. Debug the kernel function much earlier.
3. We would like to implement the OpenCL program compilation cache.
4. There might be some trade secret in the kernel functions.

Fortunately, there are some OpenCL APIs which can make this possible.

First, we have to compile our OpenCL kernel function:

```
$ ioc64 -cmd=build -input=vec_add.cl -ir=vec_add.bin
```

Now, we have converted `vec_add.cl` into binary executable for Intel OpenCL SDK.

Second, we have to load the binaries from our host program. Instead of loading `vec_add.cl` with `clCreateProgramWithSource()`, we should use `clCreateProgramWithBinary()` instead. Here's the listing of the changed lines:

```
// Create program
unsigned char* program_file = NULL;
size_t program_size = 0;
read_file(&program_file, &program_size, "vec_add.bin");

cl_program program =
    clCreateProgramWithBinary(ctx, 1, &device, &program_size,
                              (const unsigned char **)&program_file
                              NULL, &err);

err = clBuildProgram(program, 1, &device, NULL, NULL, NULL);

free(program_file);
```

Please notice that there is two subtle differences between `clCreateProgramWithSource()` and `clCreateProgramWithBinar()`:

1. We have to pass the devices list.
2. For each device, we have to specify the corresponding binary.

Although this is troblesome, this is necessary because the pre-compiled binaries are inherently not portable.

Last, I have uploaded the source code to opencl-examples. Check vec_add_binary.c for complete source code.