

Student	Tobias Stauber
Examiner	Thomas Corbat
Subject Area	Software and Systems

ReSYCLator

A CUDA to SYCL converter



Cevelop, the extensible simple C++ IDE

```

1 __global__ void matrixMultiplicationKernel(float* A,
2                                           float* B,
3                                           float* C,
4                                           int N) {
5
6     int ROW = blockIdx.y * blockDim.y + threadIdx.y;
7     int COL = blockIdx.x * blockDim.x + threadIdx.x;
8
9     float tmpSum = 0;
10
11     if (ROW < N && COL < N) {
12         for (int i {0}; i < N; i++) {
13             tmpSum += A[ROW * N + i] * B[i * N + COL];
14         }
15     }
16     C[ROW * N + COL] = tmpSum;
17 }

```

An unconverted CUDA kernel

Introduction: With the receding improvements in clock-speed in the newer CPU generations, the demand for highly parallelized computation increased. A conventional CPU is typically not optimized for this kind of tasks. There are things like Advanced Vector Extensions (AVX) for Intel- and AMD processors. However, even using those extensions, the potential for parallel computation falls a long way short of that provided by a GPU or a Field Programmable Gate Array (FPGA). Specialized parallel computing platforms were designed facilitating code execution on GPU and FPGA. One of the most known examples is NVIDIA's CUDA. CUDA has the significant advantage over its main alternative, OpenCL, to be less elaborate and less complicated to learn. This simplicity drew many developers to CUDA. Today much code is written using CUDA; this leads to the most significant disadvantage - CUDA only works on NVIDIA GPUs. For usage on alternative hardware, the applications need to be ported to a computing platform. One possible platform is SYCL developed by Khronos (the consortium also providing OpenCL). SYCL is a domain specific language embedded into C++.

Result: The objective is to create an Eclipse plug-in providing a conversion from CUDA to SYCL. The conversion plug-in allows transformation of simple CUDA programs into their SYCL equivalent. The Nvidia NSightEE plug-in provides the underlying support for CUDA and is used to recognize language constructs specific to CUDA. Using NSight poses some problems: The license prohibits reverse-engineering their plug-in; making it particularly hard to interact with it. Additionally, it depends on an outdated version of Eclipse CDT. Future work will require to create an own CUDA implementation based on up-to-date Eclipse. Currently, the ReSYCLator plug-in can convert most single-file applications. The plug-in's next iteration should be able to handle shared memory and kernels whose dependency-graph spans over multiple files.

```

1 template<int dimensions>
2 void matrixMultiplicationKernel(cl::sycl::nd_item<dimensions> item,
3                                cl::sycl::global_ptr<float> A,
4                                cl::sycl::global_ptr<float> B,
5                                cl::sycl::global_ptr<float> C,
6                                int N) {
7
8     int ROW = item.get_group(1) * item.get_local_range(1) + item.get_local_id(1);
9     int COL = item.get_group(0) * item.get_local_range(0) + item.get_local_id(0);
10
11     float tmpSum = 0;
12
13     if (ROW < N && COL < N) {
14         for (int i {0}; i < N; i++) {
15             tmpSum += *(A + ROW * N + i) * *(B + i * N + COL);
16         }
17     }
18     *(C + ROW * N + COL) = tmpSum;
19 }

```

The resulting SYCL kernel