AES Embedded Systems Architecture

BACHELOR AND MASTER THESIS
MULTICORE ARCHITECTURES

Technische Universität Berlin

Programming Models for Parallel Architectures

From High-Performance Computing (HPC) platforms to embedded systems, the advent of heterogenous and massively-parallel architectures has made such systems difficult to program. Existing parallel programming models are not suited for such systems because of the increasing sophistication of application domains, the need to mix very different programming models, the need to address new challenges such as latency, contention, starvation, and energy/power limitations. Technological trends drive towards new programming models based on runtimes systems to assure scalability and power efficiency, and to tackle the diversity of supported architectures (heterogenous computing).

```
queue myQueue;
buffer<int,1> d_a(h_a, N);
buffer<int,1> d_b(h_b, N);
buffer<int,1> d_c(h_c, N);
myQueue.submit([&](handler& cmdgroup){
 auto a =d_a.get_access<access::read>();
 auto b =d_b.get_access<access::read>();
 auto c =d_c.get_access<access::write>();
 cmdgroup.parallel_for<class test>(
  nd_range<1>(N),[=](id<1> index){
    if (index < N) {
      c[index] = a[index] + b[index];
    }
  }
 );
}
```

EXAMPLE OF SYCL CODE

Goals

The **SYCL** programming model is a cross-platform abstraction layer that builds on the underlying concepts, portability and efficiency of OpenCL and enables code for heterogeneous processors to be written in a "single-source" style using completely standard C++. The objective of the proposed theses is to extend SYCL in order to easily tackle heterogeneity on different target architectures and to enable context-specific optimizations. Two theses are available:

Available theses

1. **SYCL extensions for cyber-physical systems**. Cyber–Physical Systems (CPS) are typically embedded systems interacting with physical devices such as cameras or sensors. They usually have an additional requirement: real-time constraints. The goal is the extension of SYCL with a library/runtime system support that implements soft-real-time optimization of SYCL applications. The final code will run on a prototype heterogenous CPS system.
2. **SYCL extensions for distributed systems.** Large-scale HPC clusters are equipped with heterogenous hardware, typically a multi-core CPU and (one or more) GPUs. The goal is to extend SYCL with a distributed runtime system that supports the execution of the kernels on non-local compute nodes. The final code will run on large-scale HPC infrastructure, e.g., JSC in Julich.

Desired Skills

- Excellent C++
- Knowledge of parallel programming model such as OpenCL, CUDA, OpenMP or MPI is a plus.

Contact Person

Dr. Biagio Cosenza, cosenza@tu-berlin.de

References

- OpenCL https://www.khronos.org/opencl
- SYCL https://www.khronos.org/sycl

- Existing runtime systems (for HPC): Ompss, Starpu, OCR, HPX, Charm++, libWater, SnuCL, Darma, Legion