Heterogeneous computing is energy efficient computing

Short Presentation for EEC SIG Meeting

Paul Keir
Head of Research
Codeplay Software Ltd.
www.codeplay.com

Mon 28th July, 2014

Codeplay Software Ltd.

- Incorporated in 1999
- Based in Edinburgh, Scotland
- 34 full-time employees
- Compilers, optimisation and language development
  - GPU and Heterogeneous Architectures
  - Increasingly Mobile and Embedded CPU/GPU SoCs
- Commercial partners include:
  - Qualcomm, Movidius, AGEIA, Fixstars
- Member of three 3-year EU FP7 research projects:
  - Peppher (Call 4), CARP (Call 7) and LPGPU (Call 7)
- Sony-licensed PlayStation®3 middleware provider
- Contributing member of Khronos group since 2006
- A member of the HSA Foundation since 2013

Correct and Efficient Accelerator Programming (CARP)

"The CARP European research project aims at improving the programmability of accelerated systems, particularly systems accelerated with GPUs, at all levels."

- Industrial and Academic Partners:
  - Imperial College London, UK
  - ENS Paris, France
  - ARM Ltd., UK
  - Realeyes OU, Estonia
  - RWTHA Aachen, Germany
  - Universiteit Twente, Netherlands
  - Rightware OY, Finland
- `carpproject.eu`

Low-power GPU (LPGPU)

"The goal of the LPGPU project is to analyze real-world graphics and GPGPU workloads on graphics processor architectures, by means of measurement and simulation, and propose advances in both software and hardware design to reduce power consumption and increase performance."

- Industrial and Academic Partners:
  - TU Berlin, Germany
  - Geomerics Ltd., UK
  - AiGameDev.com KG, Austria
  - Think Silicon EPE, Greece
  - Uppsala University, Sweden
- `lpgpu.org`

Khronos SYCL for OpenCL

- Simplified software porting for existing parallel applications
- Code reuse, through sharing of host and device code
- Generic algorithms through C++ template meta-programming
- A foundation for higher-level programming models
- n.b. Host execution fallback if OpenCL device is unavailable
- Prototyped under the EU FP7 project LPGPU
- `www.khronos.org/opencl/sycl`

A Basic SYCL Code Example

```
void example()
{
  queue q;
  float f[64];
  buffer<float> data(f, 64);
  command_group(&q, [&]()
  {
    auto kdata = data.get_access<access::write>();

    parallel_for({4,4,4},
      kernel_lambda<class Bdp>([=](item id) {
        kdata[id] = id.get_global(0);
      })
    );
  });
}
```

Future Collaboration

- R&D in heterogeneous parallel programming models
- Targeting novel energy efficient architectures
- Professional compilers, optimisation and language development
- Interested in TSB and H2020 collaboration
- Email `paul@codeplay.com`