

OpenCL



StreamHPC is best-known for its OpenCL services, including development. We have hit records of over 250'000 times in optimising code using OpenCL. During those projects several techniques have been used to get to these high numbers - well-designed projects we can still speed-up 2 to 8 times.

Advantages

OpenCL works on most types of hardware of any language. Compare it to C and C++, which is used to program all kinds of software on very different hardware.

The basic OpenCL is used to make portable software that performs. The advanced OpenCL is used to optimise for maximum performance for specific accelerators.

Hardware

Projects can be targeting one or more operating systems, focusing on one or several processors:

- CPUs, by Intel, AMD and/or ARM
- NVidia GPUs
- AMD GPUs

Embedded GPUs

- Vivante
- ARM MALI
- Imagination
- Qualcomm

- Altera FPGAs
- Xilinx FPGAs
- Several special focus processors, mostly implementing a subset of OpenCL.

We use modern coding techniques to make code for maximum flexibility and maximum performance.

How OpenCL works

OpenCL is an extension to existing languages. It makes it possible to specify a piece of code that is executed multiple times independently from each other. This code can run on various processors ? not only the main one. Also there is an extension for vectors (float2, short4, int8, long16, etc), because modern processors have support for that.

So for example you need to calculate $\sin(x)$ of a large array of one million numbers. OpenCL detects which devices could compute this for you and gives some statistics of each device. You can pick the best device, or even several devices, and send the data to the device(s). Normally you would loop over the million numbers, but now you say something like: ?Get me $\sin(x)$ of each x in array A?. When finished, you take the data back from the device(s) and you are finished.

As the compute-devices can do more in parallel and OpenCL is better in describing independent functions, the total execution time is much lower than conventional methods.

4 common questions on OpenCL

Q: Why is it so fast?

A: Because a lot of extra hands make less work, the hundreds of little processors on a graphics card being the extra hands. But cooperation with the main processor keeps being important to achieve maximum output.

Q: Does it work on any type of hardware?

A: As it is an open standard, it can work on any type of hardware that targets parallel execution. This can be a CPU, GPU, DSP or FPGA.

Q: How does it compare to OpenMP/MPI?

A: Where OpenMP and MPI try to split loops over threads/servers and is CPU-oriented, OpenCL focuses on getting threads being data-position aware and making use of processor-capabilities. There are several efforts to combine the two worlds.

Q: Does it replace C or C++?

A: No, it is an extension which integrates well with C, C++, Python, Java and more.