

Porting CUDA to OpenCL



OpenCL speed-meter in 1970 Plymouth Cuda car

Why port your CUDA-acelerated software to OpenCL? Simply, to make your software also run on AMD CPU/APU/GPU, Intel CPU/GPU, Altera FPGA, Xilinx FPGA, Imagination PowerVR, ARM MALI, Qualcomm Snapdragon and upcoming architectures.

And as OpenCL is an open standard, supported by many vendors, it has much more security that it will keep existing in the future than any proprietary language.

If you look at the history of GPU-programming you'll find many frameworks, such as BrookGPU, Close-to-Metal, Brook+, Rapidmind, CUDA and OpenCL. CUDA was the best choice from 2008 to 2013, as OpenCL had to catch up. Now that OpenCL is gaining serious market traction, the demand for porting legacy CUDA-code to OpenCL rises - as we clearly notice here.

We are very experienced in porting legacy CUDA-code to all flavours of OpenCL (CPU, GPU, FPGA, embedded). Ofcourse porting from OpenCL to CUDA is also possible, as well as updating legacy CUDA-code to the latest standards of CUDA 7.0 and later. We can also add several improvements to the architecture; we have made many customers happy with giving them more structured and documented code, while working on the port. Want to see some work we did? [We ported molecular dynamics software Gromacs from CUDA to OpenCL.](#)

Contact us today, to discuss your project in more detail. We have porting services for each budget.