

Success stories



Customer using our accelerated code

We help our customers get faster, more responsive and/or more precise software. But what does that mean? What is it what we do here in Amsterdam?

Unfortunately we cannot share what we have done for most of our customers since 2010, as we are their secret weapon to make them more competitive. Below is an selection of blog posts discussing our demos, open source software and even some work for customers.

Building rocRAND. The world's fastest random number generator is built for AMD GPUs, and it's open source. With random numbers generated at several hundreds of gigabytes per second, the library makes it possible to speed up existing code numerous times. The code is faster than Nvidia's cuRAND and is therefore the preferred library to be used on any high-end GPU.

[Porting Gromacs from CUDA to OpenCL](#). Till we ported the simulation software end of 2014, it has been CUDA-only. Porting took several man-months to manually port all code. You can now download the source, build it and run it on AMD/Intel hardware - see [here](#) for more info. All is open source, so you can see our code.

Porting Manchester's UNIFAC to OpenCL@XeonPhi. Even though XeonPhi Knights Corner is not a very performant accelerator, we managed to get a 160x speedup from single threaded code. Most of the speedup is due to clever code-optimisations.

Android video filter demo [to be published]. Real-time Android-app, where the webcam stream has several OpenGL filters applied to make the it look like an old movie.

Speeding up Excel. A heavy financial algorithm is offloaded to a GPU, resulting in a big speedup.

Flooding simulation. Software that simulates flooding of land, which we ported to multi-GPU on OpenCL and got a 35x speedup over MPI.

Cartoonizer. The webcam or video stream is "cartoonized" using several image filters on an FPGA.

We work together with hardware partners to show hardware-optimised demos, so many of the demos are visual. An important part work is speeding up large-scale simulations, but these have no nice-looking visual output.