

We ported GROMACS from CUDA to OpenCL

Gromacs

fast
flexible
free



Gromacs

fast
flexible
free



GROMACS does soft matter simulations on molecular scale. Let it fly.

GROMACS is an important molecular simulation kit, which can do all kinds of "soft matter" simulations like nanotubes, polymer chemistry, zeolites, adsorption studies, proteins, etc. It is being used by researches worldwide and is one of the bigger bio-informatics softwares around.

To speed up the computations, GPUs can be used. The big problem is that only NVIDIA GPU could be used, as CUDA was used. To make it possible to use other accelerators, we ported it to OpenCL. It took several months with a small team to get to the alpha-release, and now I'm happy to present it to you.

For who knows us from consultancy (and training) only, might have noticed. This is our first product!

We promised to keep it under the same open source license and that effectively means we are giving it away for free. Below I'll explain how to obtain the sources and how to build it, but first I'd like to explain why we did it pro bono.

Why we did it

Indeed, we did not get any money (income or funds) for this. There have been several reasons, of which the below four are the most important.

The first reason is that we want to **show what we can**. Each project was under NDA and we could not demo anything we made for a customer. We chose for a CUDA package to port to OpenCL, as we notice that there is a trend to port CUDA-software to OpenCL (i.e. Adobe software).

The second reason is that **bio-informatics is an interesting industry**, where we would like to do more work.

Third reason is that we can **find new employees**. Joining the project is a way to get noticed and could end up in a job-proposal. The GROMACS project is big and needs unique background knowledge, so it can easily overwhelm people. This makes it perfect software to test out who is smart enough to handle such complexity.

Fourth is gaining **experience** with handling open source projects and distributed teams.

Therefore I think it's a very good investment, while giving something (back) to the community.

Presentation of lessons learned during SC14

We just jumped in and went for it. We learned a lot, because it did not go as we expected. All this experience, we would like to share on SuperComputing 2014.

During SC14 I will give a presentation on the OpenCL port of GROMACS and the lessons learned. As AMD was quite happy with this port, they provided me a place to talk about the project:

"Porting GROMACS to OpenCL. Lessons learned"

SC14, New Orleans, AMD's mini-theatre.

19 November, 15:00 (3:00 pm), 25 minutes

The SC14 demo will be available on the AMD booth the whole week, so if you're curious and want to see it live with explanation.

If you'd like to **talk in person**, please send an [email](#) to make an appointment for SC14.

Getting the sources and build

It still has rough edges, so a better description would be "we are currently porting GROMACS to OpenCL", but we're very close.

As it is work in progress, no binaries are available. So besides knowledge of C, C++ and Cmake, you also need to know how to work with GIT. It builds on both Windows and Linux, and NVIDIA and AMD GPUs are the target platforms for the current phase.

The project is waiting for you on <https://github.com/StreamHPC/gromacs>.

The [wiki](#) has lots of information, from how to build, supported devices to the project planning. Please RTFM, before starting! If something is missing on the wiki, please let us know by simply [reporting a new issue](#).

Help us with the GROMACS OpenCL port

We would like to invite you to join, so we can make the port better than the original. There are several reasons to join:

Improve your OpenCL skills. What really applies to the project is this quote:

Tell me and I forget.

Teach me and I remember.

Involve me and I learn.

- Make the OpenCL ecosphere better. Every product that has OpenCL support, gives choice to the user what GPU to use (NVIDIA, AMD or Intel)
- Make GROMACS better. It is already a large community and OpenCL-knowledge is needed now.
- Get hired by StreamHPC. You'll be working with us directly, so you'll get to know our team.

What can you do? There is much you can do. Once you managed to build and run it, look at the [bug reports](#). First focus is to get the failing kernels working - this is top priority to finalise phase 1. After that, the real fun begins in phase 2: add features and optimise for speed on specific devices. Since AMD FirePro is much better in double precision than Nvidia Tesla, it would be interesting to add support for double precision. Also certain parts of the code is done on the CPU, which have real potential to be ported to the GPU.

If things are not clear and obstruct you from starting, don't get stressed and send an [email](#) with any question you have. We're awaiting your merge request or issue report!

Special thanks

This project wasn't possible without the help of many people. I'd like to thank them now.

The GROMACS team in Sweden, from the KTH Royal Institute of Technology.

Szilárd Páll. A highly skilled GPU engineer and PhD student, who pro-actively keeps helping us.

- Mark Abraham. The GROMACS development manager, always quickly answering our various questions and helping us where he could.

Berk Hess. Who helped answering the harder questions and feeding the discussions.

- Anca Hamuraru, the team lead. Works at StreamHPC since June, and helped structure the project with much enthusiasm.
- Dimitrios Karkoulis. Has been volunteering on the project since the start in his free time. So special thanks to Dimitrios!
- Teemu Virolainen. Works at StreamHPC since October and has shown to be an expert on low-level optimisations.
- Our contacts at AMD, for helping us tackle several obstacles. Special thanks go to Benjamin Coquelle, who checked out the project to reproduce problems.
- Michael Papili, for helping us with designing a demo for SC14.
- Octavian Fulger from Romanian gaming-site [wasd.ro](#), for providing us with hardware for evaluation.]

Without these people, the OpenCL port would never been here. Thank you.