

## Algorithm Design



If the algorithm was not designed for parallel execution, it just can't be directly ported to OpenCL. We have the expertise in both algorithm design and GPU-programming to walk the shortest path to optimal performance.

Designing parallel version of algorithms is an intensive but important task of the speed-up process. By focusing on programmability-aspects such as caching, parallelisability and data-locality while redesigning the algorithms from the ground up, the performance can be maximised. For example a recursive algorithm is very understandable, but it uses much more memory than the stack-based version. We have experience in converting algorithms to be used in various parallel programming languages.

The algorithm-document will help you to fully understand what has changed, how it works and how you can continue from where we delivered.

In short:

- Redesign algorithms for optimal performance on parallel architectures
- Implementation in OpenCL, CUDA, OpenMP, MPI and more.
- Full documentation of the redesign process.

Check out the blog-series on [Programming Theories](#) to learn more about what we do to make your software more scalable!