OPTIMA

Accelerating real-world applications on novel multi-FPGA platforms

www.optima-hpc.eu

@optima_hpc /optima-hpc

Team:
Dr. Iakovos Mavroidis, TSI
Pavlos Malakonakis, TSI
Dr. Konstantinos Georgopoulos, TSI
Dr. Olivier Michel, Cyberbotics
Yannick Goumaz, Cyberbotics
Dr. Valeria Bartsch, Fraunhofer ITWM
Dr. Elisa Thiel, Fraunhofer ITWM
Dr. Ioannis Papaefstathiou, EXAPSYS
Dr. Aggelos Ioannou, EXAPSYS
Kostas Harteros, EXAPSYS
Prof. Dionisios N. Pnevmatikatos, ICCS
Dimitris Theodoropoulos, ICCS
Giovanni Isotton, M3E
Dr. Tobias Becker, Maxeler
Dr. Andreas Herten, FZJ
Dr. Albert Njoroge Kahira, FZJ
Dr. Eng. Gino Perna, EnginSoft
Marisa Zanotti, EnginSoft
Dr. Manuel Arenaz, Appentra Solutions
Estefanía García, Appentra Solutions

Runtime of the project ***March 2021 – November 2023***

PROJECT SUMMARY

OPTIMA is an SME-driven project that aims to port and optimize industrial applications and a set of open-source libraries into two novel FPGA-populated HPC systems. Target applications are from the domain of robotics simulation, underground analysis and computational fluid dynamics (CFD), where data processing is based on differential equations, matrix-matrix and matrix-vector operations. Moreover, the OPTIMA OPen Source (OOPS) library will support basic linear algebraic operations, sparse matrix-vector arithmetic, as well as computer-aided engineering (CAE) solvers. The OPTIMA target platforms are JUMAX, an HPC system that couples an AMD Epyc Server with Maxeler FPGA-based Dataflow Engines (DFEs), and server-class machines with Alveo FPGA cards installed.

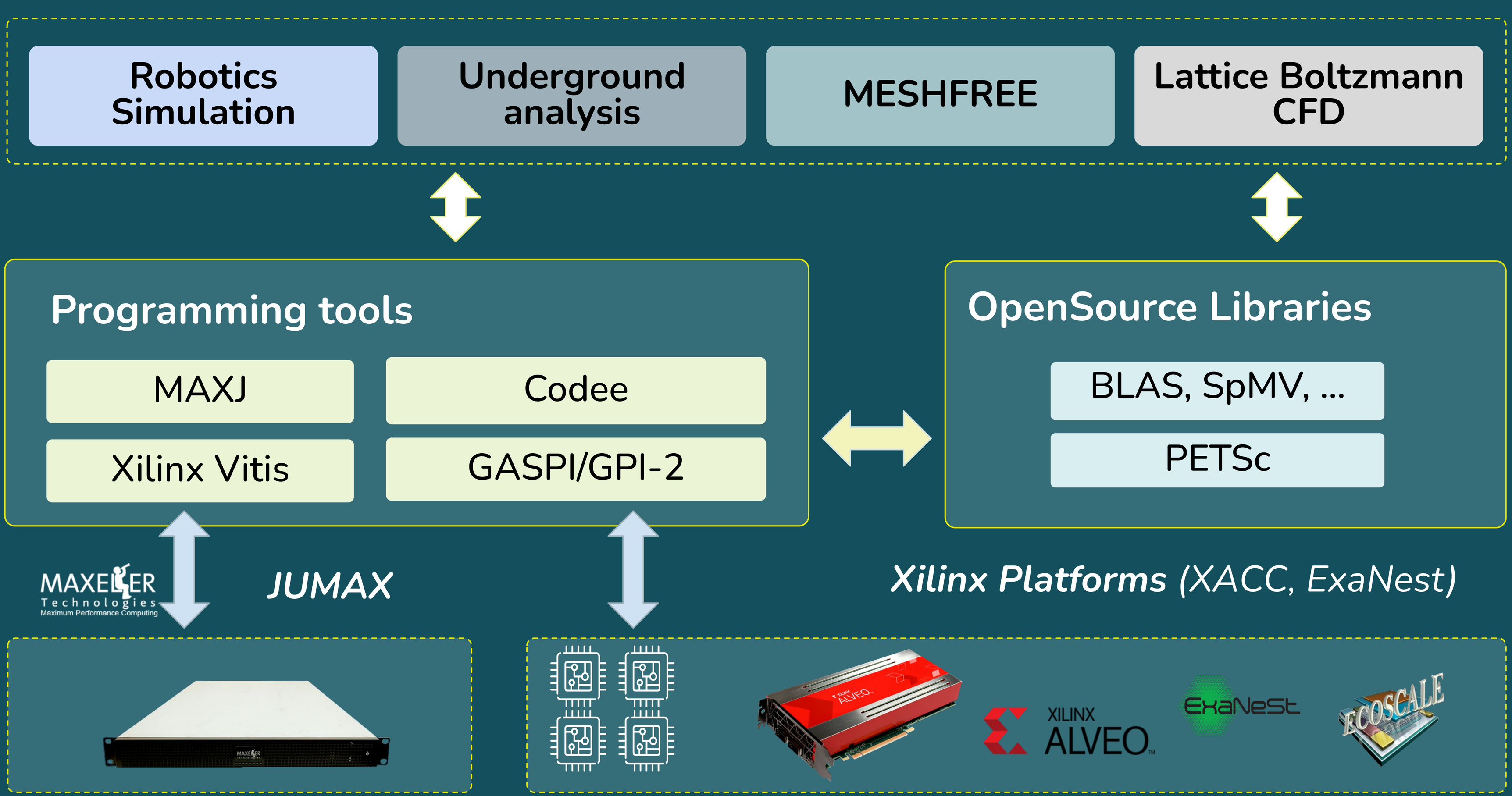
PURSUED APPROACH AND METHODOLOGY

The main activity of the project is the optimization and porting of four industrial applications and an open source library to two FPGA-populated HPC systems, so that they can be executed faster and more energy efficiently while producing more accurate results if required (for example in the case of Machine/Deep learning applications) than in current CPU and GPU-based HPC systems.

CURRENT RESULTS

OPTIMA has successfully mapped its target applications on its hardware prototypes.

Experimental results on applications up to now, show that performance on robotic simulation can be enhanced up to 1.2x, CFD calculations up to 4.7x, and BLAS routines up to 7x compared to optimized software implementations from OpenBLAS.



The JuMaX system is an FPGA-based prototype system to demonstrate high-performance and energy efficient HPC on accelerators. It is a high-performance compute system that consists of an AMD EPYC CPU server and a Maxeler MPC-X dataflow node with 8 MAX5 Dataflow Engines (DFEs). Additionally, the system contains a third node which serves as a login / head node.

The OPTIMA project is implementing a new Xilinx ALVEO-based prototype. It consists of two state-of-the-art servers, each containing two Alveo U55C accelerator cards. The U55C Alveo card is currently Xilinx's most powerful Alveo card designed for HPC and Big Data applications. The card is supported by Xilinx's Vitis Unified Software Platform, which provides an efficient software platform for developing and deploying HPC workloads on the reconfigurable hardware of the accelerator cards. The host CPUs are connected via a 1 Gbps link, and each host CPU connects two Alveo accelerator cards via a PCIex8 Gen4 that provides 16 GBytes/s throughput.

ROADMAP, NEXT STEPS & KPIS

The project has successfully completed its second year of activities, which included porting and optimizing the performance-critical parts of the applications to two HPC platforms. In terms of the PCG algorithm, experiments revealed that hardware-based versions are capable of completing up to 16 linear systems with a 1.3x speed-up compared to the software version. The D2Q9 lattice solver has also been updated to minimize memory transfers between the host CPU and FPGA. Recent experiments demonstrate that the FPGA-assisted approach is comparable to a software version that utilizes 10 host CPU cores. Finally, the BLAS L1 routines from the OOPS library showed very promising results with a speedup and performance-per-Watt ratios of up to 7x and 45x respectively, compared to Intel's MKL highly-optimized library suite, when executed on server-class host CPUs.

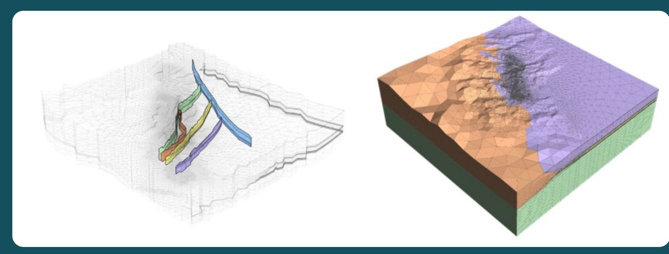
PARTNERS

APPLICATIONS

Finite Element Method (FEM) for Underground Analysis

FEM software simulators for geomechanical and groundwater analysis accelerated by Chronos, a fast and robust linear solver designed for High Performance Computing.

In the first part of the OPTIMA project, the Preconditioned Conjugate Gradient implemented in Chronos was initially ported to various FPGAs, including a Xilinx ZU9EG, a U280 and a U55C. Since the computation time of the PCG, as for all iterative methods, is dominated by SpMV products, special care was given to the porting of this kernel to preserve the computation efficiency of the linear solver. Figure on the right shows the results in terms of the execution time of a single SpMV product on a square sparse matrix arising from a real-world geomechanical application with a size of about 700,000 and an average of 44 non-zeroes per row. The results show that the computation time on 2 U55C is comparable with the CPU computation time, while on a single U55C the time does not degrade significantly In the next phase of the OPTIMA project this kernel will be the basis for extending PCG for multi-node platforms through MPI directives.



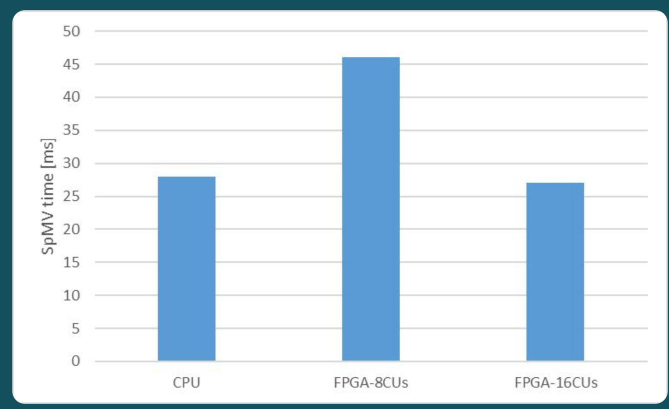


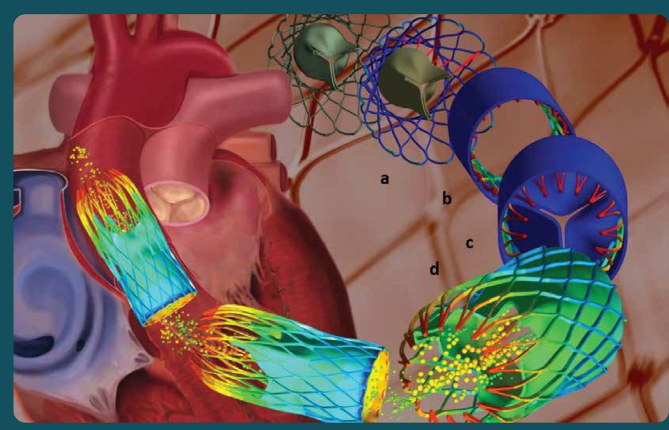
Figure 1. SpMV execution time [ms] using: 1 core of an AMD Ryzen 9 5900X at 4.5 GHz (CPU); 1 U55C with 8CUs (FPGA-8CUs); 2 U55C with 16CUs, 8 each (FPGA-16CUs).

Lattice-Boltzmann CFD code

Enginsoft (ES) is using a modified version of LBM-CFD code in order to develop custom approaches tailored to vertical solutions for its business. The code is based on an open-source solver for LBM equations for Fluid-Dynamics. The code allows to very efficiently solve a wide variety of problems that ES normally parameterized in order to let users perform sensitivity and optimization design and take advantage of HPC systems.

The LBM approach permits to efficiently solve problems without having to deal with differential equations, allowing to speed up the solution when this method is applicable. The core of the solver is coded in C. In the first phase of the OPTIMA project, the D2Q9 lattice solver has been considered and ported to FPGA. The porting process did go through a deep analysis of the calling tree of functions and a specialized full solution algorithm was implemented on FPGA using the JuMAX system. The results were promising on serial solver comparison but did suffer memory bottleneck when compared to corresponding OpenMP version on CPU. The speedup of the pure kernel has been in the order of 8 for the FPGA version. In the second part of the project a different approach has been adopted, by rewriting the solver in C++ and focusing on limiting memory transfers.

The Figure on the right shows the computation time for solving the BGK part of the code up to 10 CPU cores with the initial implementation on just 1 FPGA compute unit on JuMAX. In this new implementation scalability on FPGA is still affected by memory transfers, which will be addressed in the last part of the project.



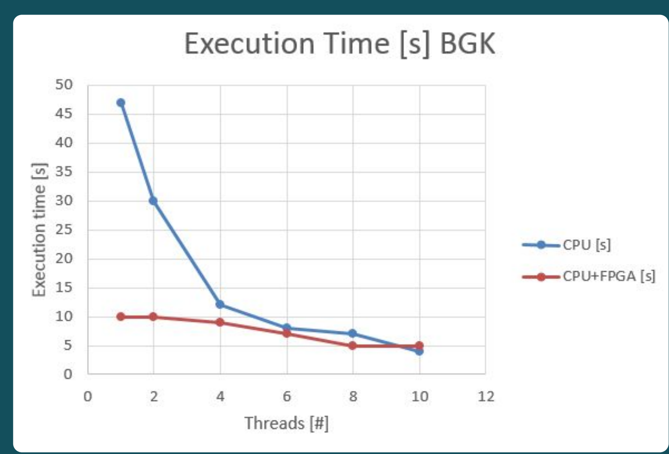
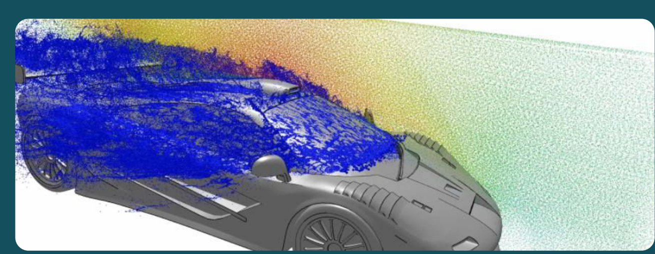


Figure 2: BGK computation time for CPU and FPGA+CPU version of the code for a 2D 1000000 cell problem

MESHFREE

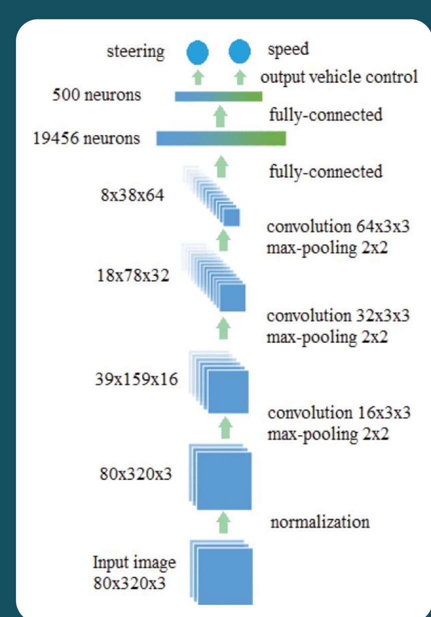
Generalized Finite Difference Method for simulations of fluid dynamics, continuum mechanics or multiphase scenarios with complex geometries.



Robotic Simulations

The robotics simulation explores FPGA-accelerated machine learning in simulated autonomous vehicles. The robot application consists of a self-driving car using a front camera to follow a given track. The objective is to accelerate robotics simulations with complex control algorithms involving Deep Learning, which slows down computation considerably. We developed a robotics simulation application based on Webots which accelerates the inference CNN in JUMAX. We implemented a car controller with trained CNN on CPU and FPGA. The simulation speed was significantly increased by Jumax when CNN inference is done on FPGAs (see Figure 3).

Adding a GPU to JUMAX system leads to real-world robotics simulation that is more than five times faster than the equivalent optimised CPU version (see Figure 4).



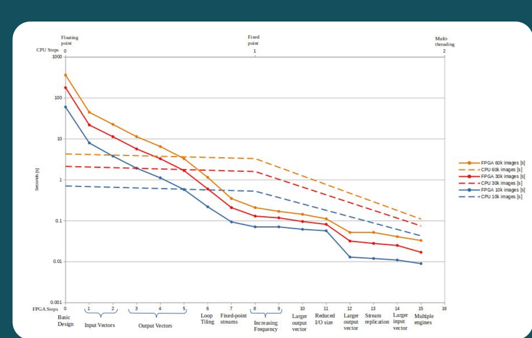


Figure 3: Successive FPGA optimizations improving the CNN inference

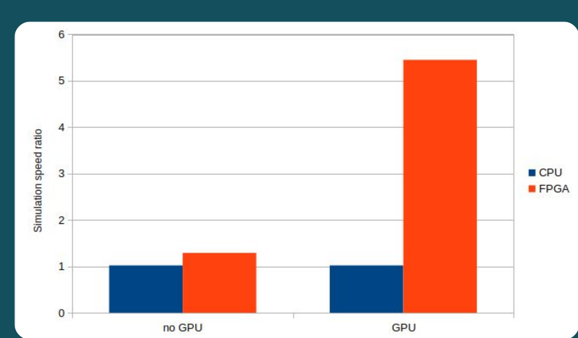
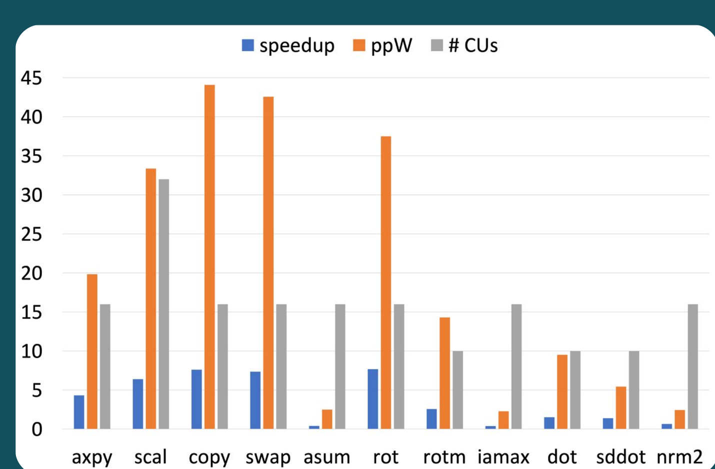


Figure 4: Comparison of simulation speed ratios between CPU and FPGA with and without GPU.

OPTIMA Open Source Library (OOPS)

The OPTIMA OPen-Source (OOPS) library set will support (by the end of the project) 27 BLAS L1, L2, and L3 subroutines, a configurable sparse matrix-vector multiplication (SpMV) kernel, and a set of computer-aided engineering (CAE) solvers, namely a Jacobi preconditioner, LU factorization, and the Krylov Conjugate Gradient (CG) algorithm. The current work focuses on the development and evaluation of all BLAS L1 routines when mapped onto the FPGA. The chart on the right shows the speedup, performance-per-Watt (ppW) and number of compute units that fit within a single U55c for all BLAS L1 kernels, when compared against Intel's MKL library functions executed on a server-class Intel Gold Xeon CPU at 2.2GHz with 56 threads. As shown, performance improvement can be up to 7x better for kernels copy, swap and rot. Scal, axpy and rotm provide 6x, 4x and 2.5x speedup respectively, whereas the rest of the kernel achieve slightly slower performance compared to the CPU. However, in all cases, the OOPS BLAS L1 kernels provide more energy-efficient implementations starting from 2.2x (nrm2) and up to 44x (copy) compared to the CPU.



This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955739. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Greece, Germany, Italy, Netherlands, Spain, Switzerland.