Optimizing Industrial Applications for Heterogeneous HPC Systems: The OPTIMA Project

Intermediate stage

Dimitris Theodoropoulos[†] Olivier Michel[‖] Pavlos Malakonakis[**] Konstantinos Georgopoulos[**]
Giovanni Isotton[¶] Dionisios Pnevmatikatos[†] Ioannis Papaefstathiou[*] Gino Perna [††] Marisa Zanotti [††]
Panagiotis Miliadis[†] Panagiotis Mpakos[†] Chloe Alverti[†] Aggelos Ioannou[*] Max Engelen[‡]
Albert Njoroge Kahira[‡‡] Iakovos Mavroidis[**]

[*]EXAPSYS plc., Thessaloniki, Greece
[†]Institute of Communications and Computation Systems, National Technical University of Athens, Athens, Greece
[‡]Maxeler IoT Labs, Delft, Netherlands
[¶]M3E S.r.l., Padua, Italy
[‖]Cyberbotics, Ltd., Lausanne, Switzerland
[**]Telecommunication Systems Institute, Chania, Greece
[††]EnginSoft SpA, Trento, Italy
[‡‡] Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Jülich, Germany

***Abstract*—OPTIMA is an SME-driven project (intermediate stage) 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. 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.**

I. Introduction

Accelerators are devices that can provide very high performance and energy efficiency when executing certain applications. Towards this end, and for certain HPC applications, Field-Programmable Gate Arrays (FPGAs) can significantly outperform GPUs, which in turn significantly outperform CPUs [1], [2]. Therefore, it is highly desirable to optimize HPC applications for FPGAs so as to take the most advantage possible from such reconfigurable accelerators.

As such, OPTIMA is a heavily SME-driven project that aims to help programmers easily deploy applications to FPGA-supported HPC systems. Its goal is to optimize real-world industrial applications to two novel FPGA-populated HPC systems; JUMAX, an HPC system that combines an AMD Epyc CPU and Maxeler FPGA-based DFEs, and server-class machines with Xilinx Alveo FPGA cards. Target applications include robotics simulation, underground analysis and computational fluid dynamics (CFD), where data processing is based on differential equations, matrix-matrix and matrix-vector operations. In addition, the project will also publish the OPTIMA OPen Source (OOPS) library, a set of basic linear algebraic operations, sparse matrix-vector arithmetic, as well as computer-aided engineering (CAE) solvers.

Having reached its intermediate state, OPTIMA's main achievements are the following so far:

- Webots, a robotics simulation software, has been ported to JUMAX, with a performance improvement close to 1.2x, when comparing the multi-threaded CPU to the FPGA performance;
- The open-source solver Lattice-Boltzmann computational fluid dynamics (LBM-CFD) has been ported to JUMAX, with a performance improvement of up to 4.7x;
- A sparse matrix-vector kernel has been successfully ported to a server machine with two U55C Alveo FPGA cards, providing marginal performance improvement for the underground analysis application;
- Minimum-resource BLAS L1 routines have been ported to a U280 Alveo card, where performance is improved by up to 7x, compared to optimized software OpenBLAS [3] implementations.

The rest of the paper is structured as follows: Section II provides an overview of the OPTIMA project. Section III presents the project's application domains. Section IV describes the project hardware platforms. Section V provides the project's intermediate results. Finally, Section VI concludes the paper and presents the project's final steps.

This project has received funding from the European High-Performance Computing Joint Undertaking 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.

978-3-9819263-7-8/DATE23/© 2023 EDAA

II. Concept

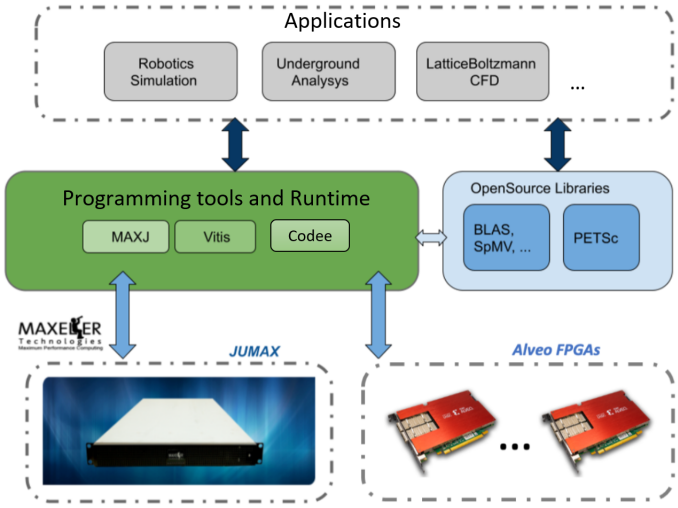

Fig. 1. High level concept of the OPTIMA framework

The OPTIMA concept is depicted in Figure 1, and consists of three main layers: The Optimized Applications and Libraries (based on the supported Programming Environment), the Programming tools and Runtime System, and the Hardware HPC Prototypes, each one of them comprising a set of novel modules. As mentioned, target applications are from the domain of robotics simulation, underground analysis, as well as computational fluid dynamics (CFD) based on the Lattice Boltzmann (LBM) algorithm.

OPTIMA will provide the efficiency of FPGA-based technologies in several industrial applications; thus the European industry shall benefit from a new class of HPC resources strongly characterised by advancing State-of-the-Art and delivering truly innovative solutions. These solutions shall take advantage of the novel heterogeneous HPC systems and commoditise the access and utilisation of such resources transforming them into a service that can be accessed by everyone, from SMEs to large organisations.

III. Applications and Libraries

A. Robotics simulation

Cyberbotics, OPTIMA's partner, has developed Webots [4], an open-source software that can simulate robots of different types, such as drones, industrial robots or autonomous cars. Webots allows developers to create a wide variety of environments and to add robots on which several different sensors can be installed, such as cameras, distance sensors, etc. Robot controllers can be programmed in many languages like C, C++, Python, Java or even using the Robot Operating System (ROS) [5]. Typical robot simulation experiments involve deep learning algorithms, such as Multi-Layered Perceptions (MLP) or Convolutional Neural Networks (CNN). They are used to process sensor input and produce actuator output. Such artificial neural networks usually rely on massively parallel computations. They usually run on multi-threaded CPU or GPU nodes in parallel with the simulation process. Since these calculations can severely affect the overall simulation time, we accelerated them on FPGA-based HPC systems.

B. Underground analysis

M3E, another partner of OPTIMA, works on underground analysis that mainly consists of simulations of the withdrawal/injection of subsurface resources (water, CO2, oil, gas, etc.) to understand and evaluate the impact on the environment. OPTIMA focuses on Chronos [6], a proprietary collection of sparse linear algebra kernels designed for modern HPC systems with thousands of interconnected nodes equipped with CPUs and GPUs accelerators. The software is mainly written in C++. Inter-processor data communication is performed using MPI, while computation on shared memory is accelerated through OpenMP directives or through the use of GPUs. Chronos has a strongly object-oriented design to be readily linked to other software and to be easily modified to support FPGAs using OpenCL APIs for host/device communications, while MPI communication between nodes is preserved.

C. LBM-CFD

Fluid dynamics is a notoriously difficult part in simulation analysis. Only a few problems can be solved analytically. Computations often are approximated at a first degree of the turbulence models and do not catch the real physics in all space but only as a mean value. Especially when coupling more physics into the same Computational Fluid Dynamics (CFD) problem (for example sound waves, radiation etc.) the behavior can run into secondary order effects (due to oscillating waves that often form vortex). These effects need second degree approximations which require a lot of computational resources. This behavior is inherently nonlinear and difficult to understand quantitatively and it needs a huge quantity of equations and iterations in order to be solved.

D. OOPS library kernels

Although matrix and/or vector based calculations dominate most computations in scientific and industrial software, there are few options available for mapping on Xilinx FPGAs. Xilinx provides an open-source kernel subset of the BLAS library that supports only 13 primitives [7] out of the total 170 that are available. Moreover, its available sparse matrix-vector (SpMV) kernel utilizes a large amount of resources, and requires matrix pre-processing (that can take hours), before making the user matrix compatible with the kernel [8]. Other approaches propose custom matrix formats [9] that, although they increase memory bandwidth utilization, require application code changes, increasing substantially development efforts.

On the other hand, 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 OOPS library will expose a C-based API to users for easy integration with existing development environments.

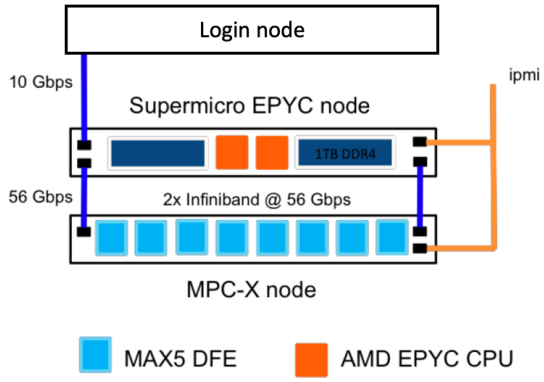

Fig. 2. The JUMAX system with 8 Maxeler Dataflow Engines (DFEs).

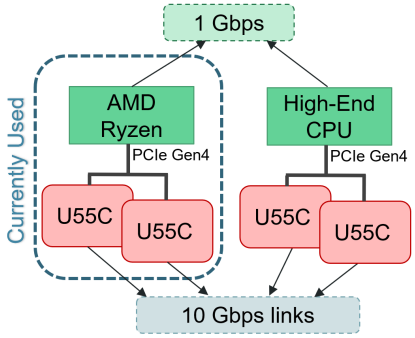

Fig. 3. Target OPTIMA prototype with four U55C accelerator cards, and two host CPUs, an AMD Ryzen and a high-end CPU; until now the project has used the AMD Ryzen CPU with two Alveo U55C cards.

Exposing a standard C-based interface serves the key target of releasing an Open-Source library set to users, which can be easily integrated and combined with existing frameworks from varying application domains. The full OOPS API will be available at the OPTIMA code repository, and will be compatible with Xilinx's latest Alveo U55C FPGAs.

IV. OPTIMA Hardware and Runtime

A. The JUMAX Platform

JUMAX[1] is an FPGA-based prototype system to demonstrate high-performance and energy efficient HPC on accelerators. The system has been developed and deployed as part of PRACE PCP. 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 DFEs. Additionally, the system contains a third node which serves as a login / head node.

The overall structure is illustrated in Figure 2. The high-performance compute node (jumax-cpu) serves as the main host system for application processing and directly invokes DFEs via an Infiniband connection to the MPC-X node. The machine uses two top-of-the line CPUs from the AMP EPYC server-grade processor generation. This is combined with a total of 1TB of DDR4 memory.

The Maxeler MPC-X 3000 node integrates 8 FPGA-based MAX5 DFEs into a dense 1U dataflow system. MAX5 DFEs are PCIe accelerator cards that combine a Xilinx UltraScale+ VU9P FPGA with 48GB of DDR4 ECC memory. MAX5 DFEs provide a 16x PCIe Gen3 interface as well as additional high-speed links directly between neighbouring DFEs (MaxRing). The card also contains a 100GB QSFP Ethernet port; however, this is not accessible in the MPC-X configuration. The MPC-X node does not contain any application processors. The DFEs are accessible via RDMA over Infiniband from the jumax-cpu host server. Finally, a login node is also provided that is directly accessible to external users. This node is connected to the JUMAX-CPU node over a 10G network.

B. The OPTIMA Alveo-based HPC Platform

As shown in Figure 3, the second target prototype is based on a dual-server machine setup, where each host CPU is connected to two Xilinx U55C Alveo accelerator cards. The U55C Alveo card is currently the Xilinx's most powerful Alveo card buit 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 a 12-core AMD Ryzen 9 and an Intel processor, connected over a 1 Gbps link. Each host CPU can interface two Alveo accelerator cards through a PCIe x8 Gen4 offering 16 GBytes/sec throughput between the two accelerators and the host memory. Moreover, all Alveo cards will be interconnected directly through 10 Gbps custom QSFP links. It should be noted that until the current intermediate state, OPTIMA uses the AMD host CPU and two Alveo cards for development and kernel validation / evaluation.

V. Experimental Results

A. Robotics simulation

Webots was developed using various configurations; first a multi-threaded CPU version of the Convolutional Neural Network (CNN) inference (784 inputs, 64 hidden layers, and 10 outputs) was parallelized in software, and executed using 128 parallel threads on the JUMAX host CPUs. The same CNN was ported in hardware and accelerated on the JUMAX dataflow engines (DFEs). Results showed that there was no significant performance increase, because the bottleneck of the overall simulation was rendering of the camera images on the simulator side, due to not using a dedicated GPU on JUMAX. As a result, the overall Webots performance gain was approximately 1.2x when comparing the performance of the FPGA implementation to the CPU implementation. The FPGA kernel facilitates a number of optimization techniques, such as loop-tiling, fixed-point streams, stream replication and multiple data-flow engines in order to enhance performance.

When a GPU was installed in the JUMAX platform, Webots was configured to use the CPU/GPU nodes for image rendering, and the CPU/FPGA JUMAX nodes for CNN processing. It should be noted that the CPU/GPU-CPU/FPGA JUMAX nodes communicate through high bandwidth network connections. Under this configuration, processing bottleneck was shifted to

[1]located at the Jülich Supercomputing Center (FZJ)

TABLE I
SPMV EXECUTION TIME ON SOFTWARE AND FPGA.

CPU	FPGA-8CUs	FPGA-16CUs
28 msec	46 msec	27 msec

the FPGA, and thus resulted in a performance boost of about 5.3x, compared to the purely CPU implementation.

B. Underground analysis

The Preconditioned Conjugate Gradient implemented in Chronos is an iterative method for solving the linear algebraic system whose algorithm is a sequence of LEVEL-1 BLAS kernels and Sparse Matrix-Vector (SpMV) products. The computation time of the PCG, as for all iterative methods, is dominated by SpMV products, so special care was given to the porting of this kernel to preserve the computation efficiency of the linear solver.

Table I shows the results in terms of the execution time of a single SpMV product. In particular, there is a comparison between the time obtained by the CPU version using 1 core of an AMD Ryzen 9 5900X at 4.5 GHz and the accelerated version on 1 and 2 U55C accelerator cards, with 8 and 16 Compute Units (CUs) respectively, since we could have 8 CUs per U55C. The square sparse matrix used as a test case arises from a real-world geomechanical application and has a size of about 700,000 with an average of 44 non-zeroes per row. The evaluation 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. Given the promising results, in the next phase of the OPTIMA project, this kernel will be the basis for extending PCG for multi-node platforms through MPI directives.

C. LBM-CFD

In the first part of the project the main work was focused on implementing a hybrid CPU-FPGA version of the solver, mainly based on an SMP (openMP) code and a specific lattice pattern: D2Q9. By instrumenting the kernel with run-time performance collecting libraries and running a number of different simulations using D2Q9 lattice, the main computational kernel has been isolated. The kernel has been then re-programmed in order to execute in a single FPGA CU the entire domain of the analysis. The performance results have shown that the speed of the computational kernel was boosted by a factor of 10.0, while the overall time by a 4.7x. All computations were run on the JUMAX host and one FPGA.

D. OOPS library kernels

As mentioned, the OOPS library set exposes high-level function prototypes that applications executed on the host processor can call to enable data processing onto hardware. On the host side, each kernel uses a set of Xilinx Runtime (XRT) system calls to allocate space on the memory located to the FPGA card (either DDR or HBM). Next, the software copies input data to the allocated memory space on the FPGA side. On the FPGA side, the hardware compute unit (CU) leverages (when possible) burst data accesses. Using dedicated HLS directives, the CU processes input data and / or updates intermediates status (e.g., accumulators), and writes back the results to the device memory. On the host side, when done, the software copies back results from the device to the host memory.

TABLE II
SPEEDUP VS OPENBLAS FOR THE OOPS LIBRARY BLAS L1 KERNELS.

iamax	asum	axpy	copy	dot	sddot
1.2x	1.05x	1.7x	7x	2.3x	2.1x
nrm2	**rot**	**rotm**	**scal**	**swap**	**iamin**
4.8x	1.6x	2.1x	0.8x	1.3x	1.2x

Resource utilization for 12 BLAS L1 kernels (that are ready up to this project state), when implemented on an Alveo U280 FPGA, is less than 2%, 1.6% and 3.5% for register, LUTs and BRAMs respectively. Moreover, Table II compares all kernels vs their counterpart single-threaded optimized software implementation from the OpenBLAS library, using vectors with 60M elements each. It should be noted that hardware execution includes accessing the FPGA High-Bandwith Memory (HBM), as well as data processing. As shown, all kernels (except scal) enhance performance, with copy speeding up data transfers between locations in the device memory up to 7 times.

VI. CONCLUSIONS

Up to this stage, OPTIMA has successfully mapped many target applications on its hardware prototypes. The Webots robotics simulation software and LBM-CFD execution times were improved by a factor of 1.2x and 4.7x respectively, whereas a first version of the sparse matrix-vector kernel has been successfully mapped to a server machine with two U55C Alveo FPGA cards, providing marginal performance improvements. Finally, a set of minimum-resource BLAS L1 routines have also been mapped successfully to U280 Alveo cards, with performance improvements of up to 7x compared to optimized software implementations from OpenBLAS.

REFERENCES

[1] Joshua Lant, Javier Navaridas, Mikel Lujan, John Goodacre, "Making the case for FPGA based HPC," in *IEEE Micro*, October 2019.

[2] Eriko Nurvitadhi, "Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC," in *FPT*, 2016.

[3] Wang Qian, Zhang Xianyi, Zhang Yunquan, Qing Yi, "Augem: Automatically generate high performance dense linear algebra kernels on x86 cpus," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2013.

[4] Oliver Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004. [Online]. Available: http://www.ars-journal.com/International-Journal-of-Advanced-Robotic-Systems/Volume-1/39-42.pdf

[5] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, William Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, 05 2022.

[6] https://www.m3eweb.it/chronos., [Online; accessed November 2021].

[7] https://xilinx.github.io/Vitis_Libraries/blas/2022.1/user_guide/L1/L1_compute_api.html, [Online; accessed November 2022].

[8] https://xilinx.github.io/Vitis_Libraries/sparse/2022.1/user_guide/L2_spmv_double_intro.html, [Online; accessed November 2022].

[9] Yixiao Du, Yuwei Hu, Zhongchun Zhou, Zhiru Zhang, "High-performance sparse linear algebra on hbm-equipped fpgas using hls: A case study on spmv," in *Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2022.