Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

WP6 Applications and Use cases

D6.1 Evaluation plan

Revised version

http://textarossa.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme, EuroHPC JU, grant agreement No 956831

TEXTAROSSA

Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

Grant Agreement No.: 956831

Deliverable: D6.1 Evaluation plan

Project Start Date: 01/04/2021 **Duration**: 36 months
Coordinator: *AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE - ENEA, Italy.*

Deliverable No	D6.1 (revised)
WP No:	WP6
WP Leader:	PSNC
Due date:	M18 (September 30, 2022)
Delivery date:	M27 (revised)

Dissemination Level:

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020 research and innovation programme, EuroHPC JU, grant agreement No 956831

DOCUMENT SUMMARY INFORMATION

Project title:	Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale
Short project name:	TEXTAROSSA
Project No:	956831
Call Identifier:	H2020-JTI-EuroHPC-2019-1
Unit:	EuroHPC
Type of Action:	EuroHPC - Research and Innovation Action (RIA)
Start date of the project:	01/04/2021
Duration of the project:	36 months
Project website:	textarossa.eu

WP6 Applications and Use cases

Deliverable number:	D6.1					
Deliverable title:	Evaluation plan					
Due date:	M18					
Actual submission date:	M27 (revised)					
Editor:	Michał Kulczewski					
Authors:	Carlos Alvarez , Olivier Beaumont, Berenger Bramas, Laura Cappelli, Daniele Cattaneo, Sebastian Ciesielski, Pasqua D'Ambra, Daniel Jaschke, Jens Krueger, Martin Kuehn, Michał Kulczewski, Alessandro Lonardo, Ariel Oleksiak, Alice Pagano, Paolo Palazzari ,Cristian Rossi, Federico Rossi, Sergio Saponara, Francesco Simula, Federico Terraneo, Giuseppe Zummo					
Work package:	WP6					
Dissemination Level:	Public					
No. pages:	48					
Authorized (date):	30/05/2022					
Responsible person:	Michał Kulczewski					
Status:	Plan	Draft	Working	Final	Submitted	Approved

Revision history:

Version	Date	Author	Comment
0.1	2022-07-05	Michał Kulczewski	Draft structure

0.2	2022-08-24	Michał Kulczewski	Executive summary, Future work, ready for partner's contribution
0.3	2022-08-29	Martin Kuehn, Pasqua D'Ambra, Michał Kulczewski	RTM, CNR contributions and tables update
0.4	2022-09-10	Sergio Saponara	CINI-UNIPI contribution
0.5	2022-09-12	WP6 partners	First version for WP1
0.6	2022-09-15	INFN	Update
0.7	2022-09-21	Alessandro Lonardo	3.1 update
0.8	2022-09-21	Michał Kulczewski	Ready for QC
0.9	2022-09-23	Pasqua D'Ambra	Internal review
1.0	2022-09-29	WP6 partners	Addressing internal review comments, final update
1.1	2023-04-17	WP6 partners	Addressing rejections comments
1.2	2023-05-09	All partners	Addressing rejections comments
1.3	2023-05-12	Michał Kulczewski	Ready for internal review
1.4	2023-05-30	All partners	Addressing reviewers comments
2.0	2023-05-31	Michał Kulczewski	Final version

Quality Control:

Checking process	Who	Date
Checked by internal reviewer	Carlos Alvarez	22/05/23
Checked by internal reviewer	Elisabetta Boella	23/05/23
Checked by Task Leader	Pasqua D'Ambra	15/05/2023
Checked by WP Leader	Michał Kulczewski	31/05/2023
Checked by Project Coordinator	Massimo Celino	03/06/2023

COPYRIGHT

ACKNOWLEDGEMENTS

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement no 956831. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Italy, Germany, France, Spain, Poland.
Please see http://textarossa.eu for more information on the TEXTAROSSA project.
The partners in the project are AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE (ENEA), FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. (FHG), CONSORZIO INTERUNIVERSITARIO NAZIONALE PER L'INFORMATICA (CINI), INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), BULL SAS (BULL), E4 COMPUTER ENGINEERING SPA (E4), BARCELONA SUPERCOMPUTING CENTER-CENTRO NACIONAL DE SUPERCOMPUTACION (BSC), INSTYTUT CHEMII BIOORGANICZNEJ POLSKIEJ AKADEMII NAUK (PSNC), ISTITUTO NAZIONALE DI FISICA NUCLEARE (INFN), CONSIGLIO NAZIONALE DELLE RICERCHE (CNR), IN QUATTRO SRL (in4). Linked third parties of CINI are POLITECNICO DI MILANO (CINI-POLIMI), Università di Torino (CINI-UNITO) and Università di Pisa (CINI-UNIPI); linked third party of INRIA is Université de Bordeaux; in-kind third party of ENEA is Consorzio CINECA (CINECA); in-kind third party of BSC is Universitat Politècnica de Catalunya (UPC).
The content of this document is the result of extensive discussions within the TEXTAROSSA © Consortium as a whole.

DISCLAIMER

Table of content

List of Tables

List of Figures

Executive Summary

This deliverable provides initial results of task T6.4 to orchestrate the evaluation of Textarossa solutions. The document presents an evaluation plan using a top-down approach. First, it gives a rough overview of Textarossa applications and their connection with other tasks using different approaches. Heterogenous resources will be used by eight applications; mixed-precision will be applied to three use cases; dynamic runtime systems will be evaluated by one library. Important to say, these use cases are coming from many different scientific domains, representing examples of AI, HDPA and HPC codes. As a result, this brings a broad vision of the needs to the project. As of M24, all applications reached MS2, which means that the codes are well tested and ready to apply Textarossa solutions. Going bottom, a comprehensive list of Textarossa features (hardware, software and programming models) is presented to i) demonstrate which application is going to use these features and to ii) give an overview on how developed technology will be evaluated. Going into more details, there is a summary of Key Peformance Indicators (KPIs) for each application, followed by their explanation and an overall evaluation plan for the aforementioned three different application approaches. At the bottom, a detailed evaluation plan for each application is presented, and contribution to the project objectives is discussed.

The work carried out in WP6 corresponds directly to the following overall project objectives:

- Energy efficiency, by the application developments;
- Sustained application performance, by the application developments;
- Seamless integration of reconfigurable accelerators, by using the APEIRON framework;
- Development of new IPs, by using INFN intra/inter-FPGA communication IP behind the APEIRON framework;
- Integrated Development Platform, by using existing IDV-A and IDV-E;
- Opening of new usage domain, by the application developments.

1 Introduction

Work performed in WP6 is essential to demonstrate the Textarossa outcomes in both, hardware and software perspectives. Moreover it is important to provide conclusions if and how they provided solutions can improve energy efficiency and sustainable performance of application representing different domains.

In Textarossa, we focus on applications related to AI (Artificial Intelligence), HDPA (High Performance Data Analytics) and HPC (High Performance Computing). Provided software represents quite a comprehensive set of different hardware used (CPU, GPU, FPGA), programming models (MPI for distributed computing/data exchange, CUDA, Intel One API, etc.) and problems to be solved (sparse and dense linear algebra, iterative and direct solvers, etc.). Because of that, there is a different set of computational and energy efficiency metrics defined (KPI – Key Performance Indicator) for each of the applications, though some naturally overlaps. To provide a high-quality evaluation plan for these many different applications, we applied a top-down approach to describe it. First, we start with a general overview of applications. Then, we discuss in details technologies developed in the Textarossa project – how they will be evaluated, which applications are using them. Next, we discuss applications in detail. Starting from explaining the reasons to improve and how the project objectives are targeted, we draw KPIs and an overall evaluation plan to finalise with details of evaluation of each of the use cases.

The work carried out in WP6 corresponds directly to the following overall project objectives:

- Energy efficiency, by the application developments;
- Sustained application performance, by the application developments;
- Seamless integration of reconfigurable accelerators, by using the APEIRON framework;
- Development of new IPs, by using INFN intra/inter-FPGA communication IP behind the APEIRON framework;
- Integrated Development Platform, by using existing IDV-A and IDV-E;
- Opening of new usage domain, by the application developments.

This document is organised as follows. Section 2 provides overview of the applications. Section 3 discusses technologies developed in Textarossa to be exploited by the applications. Section 4 details overall evaluation plan for applications. Section 5 describes individual evaluation plan. In Section 6 we discuss how KPIs can be extended to take the outcomes of the WP1 and how we will update the plan during the scope of the project if necessary.

2 Applications

In WP6 there are 10 main applications representing AI, HDPA and HPC classes. The mathematical libraries developed by CNR and INRIA can be divided into separate modules, however they are referred as a single mathlib to keep things simpler. A high-level overview is given in Table 1. Nine out of ten applications will benefit from heterogeneous hardware resources, three of them plan to introduce mixed-precision, and one of them will benefit from dynamic runtime systems. However, some heterogeneous applications will consider applying mixed-precision and/or dynamic runtime systems, thus the number of use cases using different approaches may change during the scope of the project. All applications accomplished MS2 – prototype applications are ready, albeit not yet integrated. The most important feature is that there is at least one application for each functionality (task) already available.

App name	Partner	Heterogeneous (T6.1)	Mixed-precision (T6.2)	Dynamic runtime (T6.3)	MS2
Smart cities	CINI	CPU+GPU	Mixed-precision formats supported by GPU	No	Yes
Mathlib-CNR	CNR	CPU+GPU	Planned, work not started	No	Yes
RTM (benchmark kernel only)	Fraunhofer	No	Yes (Posit)	No	Yes
HEP	INFN	GPU, possibly FPGA	No	No	No
NestGPU	INFN	GPU	No	No	Yes
RAIDER	INFN	FPGA	No	Yes (OmpSs)	Yes
TNM	INFN	GPU	No	Yes	Yes
DNN Inference	INRIA	GPU	No	Yes (StarPU)	Yes
Mathlib-INRIA	INRIA	GPU, FPGA	No	Yes (StarPU)	Yes
UrbanAir	PSNC	GPU	Planned, worked not started	No	Yes

Table 1 Overview of WP6 applications

In order to evaluate all technologies developed in the project, additionally two mini applications have been introduced: Deflate (heterogeneous resources) and HPC-Benchmark (heterogeneous resources, dynamic runtime systems).

3 Technologies and evaluation

Table 2 presents an overview of technologies developed in the project, including short description, responsible partners, targeted hardware, applications or technologies used to evaluate KPIs. In the next subsections each technology is discussed in detail.

Technology	Partner	Description	IDVs	Processor	App or technologies used as benchmark	KPI to test the technology (more related functionalities)
StarPU	INRIA	Task-based process management	IDV-A	CPU/GPU	MathLib (INRIA)	Power efficient scheduler
StarPU	INRIA	Task-based process management	IDV-E	ARM	MathLib (INRIA)	Seamless addressing of FPGA, efficient model management
FastFlow	UNIPI	Streaming process management	IDV-E/IDV-A	ARM	MiniApp ImageProce + deflate compression	Execute some fastflow node on FPGA accelerators
APEIRON	INFN	Streaming process management	IDV-E	FPGA	RAIDER (INFN)	Enable deployment of communicating Vitis HLS kernels on multi-FPGA systems for dataflow applications. It addresses project objectives: 1) Seamless integration of reconfigurable accelerators, 2) Energy efficiency, and 3) Sustained Performance. Events/s and Events/s (RAIDER application). It leverages INFN Communication IP
OmpSs	BSC	Task-based process management	IDV-A	ARM	RAIDER (INFN), BSC-HPC	Gflops and GigaPAIRS per second (BSC-HPC). Events/s, Events/J (RAIDER)
Mixed-precision	UNIPI	Yolo-based people detection + deepSort people tracking for AI-based video surveillance	IDV-A (IDV-E) ARM or X86 GPP plus Nvidia GPU	ARM or x86	Yolo-based people detection + deepSort people tracking for AI-based video surveillance in different platforms (ARM Neoverse or ARM Cortex or x86 plus T4 or A100) using FP16 or FP32. Smart Cities	Execution time and power consumption

Posit	UNIPI	Accelerator with Posit computing arithmetic and data compression for AI efficient processing	IDV-E (ARM-based GPP as in EPI2 Rhea plus accelerator emulated in FPGA)	ARM	Mini-app (CNN with test benches like GTRSB, MNIST, CIFAR), in collaboration with Fraunhofer to test posits in Reverse Time Migration (wave equation). Common contributions UNIPI-Fraunhofer at ISC2023 e UNIPI-POLIMI at PARMA-DITAM 2023	Accuracy for AI-based classification e data width di Posit vs Float ; power consumption evaluated with POLIMI (see POLIMI activity Automatic instrumentation of RTL to create power monitors of HW accelerators)
Secure HPC service	UNIPI	Secure over the air firmware update using in a HW-SW implementation Shake as hashing function for digital signature plus PQC Crystals Dilithium (Use case developed by UNIPI in EPI2)	IDV-E (ARM-based GPP as in EPI2 Rhea plus accelerator emulated in FPGA)	ARM	Use case developed by UNIPI in EPI2	Use case developed by UNIPI in EPI2
INFN Communication IP	INFN	IP for low-latency intra-node e inter-node communication	IDV-E	FPGA	RAIDER (INFN), miniapp (video processing)	1) One way Communication latency 2) Communication Bandwidth
HW fast scheduler IP	BSC	HW fast scheduler IP	IDV-E	FPGA	BSC-HPC, RAIDER	Num. of tasks scheduled per second
Memory hierarchy optimization and runtime systems	BSC	Memory interleave	IDV-E	FPGA	BSC-HPC, RAIDER	Memory bandwidth

TAFFO extended to support GPU/FPGA	POLIMI	Compiler Technology for Mixed-Precision Support	IDV-E/IDV-A	GPU/FPGA	UrbanAir kernels for IDV-A (PSNC), Mathlib (INRIA), Mathlib kernels for IDVA-A (CNR) under investigation	Speedup with given error bound
Automatic instrumentation of RTL to create power monitors of hardware accelerators	POLIMI	Automatic instrumentation of RTL to create power monitors of hardware accelerators	IDV-E	FPGA	PPU (UNIPI)	Availability, accuracy vs area overhead
Multi-level thermal management	POLIMI	Thermal and power control of 2-phase cooling and actuators	IDV-A (IDV-E)		IDV-E only if DVFS control is available. Any application running on the platform	Temperature under the set point with a given accuracy
Two-phase cooling	Inquattro	Board/Node cooling	IDV-E	Board/Node		The KPI is for assessing the cooling performance of a High-Performance Computing (HPC) single server. Electrical power consumption of the two-phase cooling per cooled thermal power (KW) at 40°C, 45°C and 50°C. These are the temperatures of the external heat rejection (with free cooling) into atmosphere, during summer season
	Inquattro	Board/Node cooling	IDV-A	Board/Node	Nvidia stress-test	

Table 2 Technologies and KPIs

3.1 StarPU

StarPU is a task-based runtime system that enables the programmer to insert tasks and define their dependencies, thereby allowing StarPU to manage their execution: the movement of data across processing units, the parallel execution of tasks while ensuring coherency, and the selection of processing units for task computation.
In the Textarossa project, our objective is to enhance StarPU's functionalities by introducing a new scheduler that optimizes task distribution and having StarPU to support FPGAs. The primary goal of the new scheduler is to minimize makespan and/or reduce energy consumption during execution.
As part of WP6, we intend to evaluate the new features of StarPU on two applications, namely Chameleon and ScalFMM. The KPIs we will study include makespan, assessed by analyzing the execution time as a whole or against a workload coefficient (GFlop or the number of particle interactions), and energy consumption, analyzed by measuring Watt instead of time.

3.2 FastFlow

FastFlow is a structured parallel programming environment based on the key concept of parallel design patterns provided to application programmers to seamlessly model parallelism exploitation. FastFlow targets standalone multicore shared memory nodes as well as clusters of shared memory multicore nodes. In the Textarossa project, FastFlow is being extended in such a way that its patterns can now be used to properly orchestrate offloading of computations to FPGAs. We assume that FPGA kernels exist, compiled through the Vitis toolchain, that can be used to offload computations to the available Xilinx ALVEO FPGA boards. We extend FastFlow run time such that the offloading of computations to FPGA kernels may be orchestrated in parallel by using specialized versions of the parallel patterns provided by the original FastFlow. Specialization consists in the possibility to use the FPGA kernels as “business logic code” of the FastFlow parallel pattern components in the very same way standard C/C++ code may be used to implement the business logic code of classic, host side FastFlow patterns.

As part of WP6, we will use mini applications to demonstrate the efficiency of the approach. We are considering currently video stream processing and compression mini applications where significant parts of video frame processing and compression algorithms are executed through FastFlow parallel patterns offloading computations to FPGA kernels. We will evaluate the results achieved through two kinds of KPIs, namely i) the number of different FastFlow patterns supporting FPGA offloading and ii) the performance achieved in the parallel mini apps with respect to more classical parallel application codes written with the explicit usage of OpenCL host side code orchestrating the parallel offloading. We expect that at the end of the project FPGA i) offloading will be supported in pipeline and farm patterns (the most used patterns in stream parallel applications) and ii) we achieve performances similar to the ones achieved using OpenCL specific host side code to orchestrate the same computations with the same kind of parallelism.

3.3 APEIRON

APEIRON is a framework encompassing the general architecture of a distributed heterogeneous stream processing platform and the corresponding software stack, from the low-level device drivers up to the high-level programming model. The framework is designed to be efficiently used for studying, prototyping and deploying high performance distributed dataflow applications, such as the RAIDER real-time AI-based data analytics application that represents the main use case for APEIRON.
Using APEIRON developers can define scalable applications using a dataflow programming model (inspired by Kahn Process Networks) that can be efficiently deployed on a multi-FPGAs system: the INFN Communication IPs allows low-latency communication between processing tasks deployed on FPGAs, even if hosted on different computing nodes.

Thanks to the use of High-Level Synthesis tools in the workflow, namely Xilinx Vitis, tasks are described in a high-level language (C++) while communication between tasks is expressed through a lightweight C++ API based on non-blocking send() and blocking receive() operations.
The mapping between the computational dataflow graph and the underlying network of FPGAs is defined by the designer with a configuration tool, by which the framework will produce all project files required for the FPGAs bitstream generation.
The interconnection logic is therefore automatically built according to the application needs (in terms of input/output data channels), allowing the designer to focus on the processing tasks expressed in the C++ programming language.

The development of the framework addresses the project objectives:

- Objective 1 - Energy efficiency. APEIRON addresses this objective enabling the complete offload of the dataflow processing to FPGA devices. Furthermore, avoiding the involvement of the CPUs and system bus resources in data transfers and the usage of bounce buffers, it improves the energy efficiency of the multi-FPGA execution platform. The achievement of this objective will be assessed by measuring the processed Events/J by the RAIDER application, and comparing it with what will be obtained running the same processing pipeline both on CPU only and CPU+GPU systems.

- Objective 2 - Sustained application performance. The sustained application performance of distributed dataflow applications, such as the RAIDER use case, is strongly affected by the performance of the network system. Implementing a direct FPGA to FPGA interconnect and bypassing the host network stack allows to keep the communication latency in the sub-microsecond range and increase the bandwidth for small messages. The achievement of this objective will be assessed measuring the processed Events/s obtained by the RAIDER application, and comparing it with what will be obtained running the same processing pipeline both on CPU only and CPU+GPU systems.

- Objective 4 - Seamless integration of reconfigurable accelerators. The APEIRON framework leverages the Vitis HLS workflow, extending it to a multi-FPGA execution platform through a lightweight communication library (HAPECOM) at programming level, and through a simple configuration system for the deployment of the distributed application to the multi-FPGA execution platform.

- Objective 5 - Development of new IPs. The INFN Communication IP is the key enabling technology behind the APEIRON framework, allowing direct low-latency intra/inter FPGA communications between HLS kernels.

- Objective 6 - Integrated Development Platform. The ARMv8 based IDV-E represents the target execution platform for the APEIRON runtime. Porting its software stack to this architecture, besides the X86_64 that will be used for the framework development, will demonstrate the host-agnosticism feature of the framework.

3.4 OmpSs

OmpSs is a task-based programming model. OmpSs has been designed to be non-invasive so that minimal changes have to be made in order to port and parallelize an application. In particular, pragma directives annotations are used in order to allow productive parallel programming, even for heterogeneous architectures, such as GPUs and FPGA. In Textarossa, the OmpSs@FPGA framework is going to be used in order to facilitate the programming of the IDV-E platform. In addition, we expect to extend the OmpSs@FPGA platform to use the Fast Task Scheduling IP (also developed in Textarossa) and other new researched features to improve the performance of both the framework and the platform.

Task based programming models such as OmpSs provide a good opportunity to abstract the underlying hardware complexity, so implementation effort is kept low while maintaining good levels of performance. The objectives of the OmpSs programming model in the Textarossa project can be summarized as follows:

- Provide support to the execution of task-based parallelized programs (with the OmpSs programming model) on the Textarossa IDV-E platform.
- Improve performance of such task-based parallelized programs so it can be competitive with alternative current state-of-the-art programming models.
- Explore how task-based and stream programming models could be mixed to obtain systems that get the best of both worlds in terms of programmability, performance and energy efficiency.
- Support the Fast Task Scheduling Hardware developed in Task 2.5 and leverage it to accomplish previous objectives.

The last objective is accomplished by fulfilling the three previous objectives. These objectives are related to the project objectives:

- Objective 1 - Energy efficiency. FPGAs demonstrated to be competitive with other computing platforms in terms of energy efficiency. Besides providing the support to executing applications on the IDV-E platform, the OmpSs task-based model is going to be integrated with power measurement tools in order to acquire further control and improve the energy spent when executing on the platform.
- Objective 2 - Sustained application performance. As explained in the next sections, we aim to improve the performance obtained when executing applications over the IDV-E platform both by improving the framework and also by improving the task scheduling through the use of the Fast Task Scheduler developed in Task 2.5.
- Objective 3 - Fine-tuned thermal policies integrated with an innovative cooling technology. As explained in Objective 1, the power measurement tools to be integrated in the OmpSs framework for IDV-E are expected to provide the basis for integrating fine-tuned thermal policies in Task 4.5.
- Objective 4 - Seamless integration of reconfigurable accelerators. OmpSs@FPGA runtime allows for seamless integration of reconfigurable accelerators.
- Objective 5 - Development of new IPs. The Fast Task Scheduler IP is a key part of the OmpSs@FPGA framework. OmpSs@FPGA contributes to the IP development as a primary tool to test the IP functionality. It also provides design requisites that must be incorporated in the IP for the whole framework to work as expected.
- Objective 6 - Integrated Development Platform. The OmpSs@FPGA runtime will be used in applications executing on the IDV-E platform. It is important to highlight that IDV-E features a host CPU (ARM based) that has never before been used to drive computation in a PCIe attached FPGA. Developing the system in a way that is compatible with new different CPUs helps ensuring new host CPUs (like EPI CPUs) will be able to drive this kind of computations in the future.

3.5 Posit Arithmetic

The main objective of designing and developing a hardware IP that handles Posit computations is to deliver a mean to compress standard IEEE 32-bit floating point (FP32) arithmetic by a factor from 2 to 4, while maintaining similar accuracy in different applications and tasks such as machine learning. The compression factor of bit width for similar accuracy, i.e. use of Posit8 or Posit 16 as replacement of FP32, depends also on the considered algorithm and application scenario.

The aim is focused on two paths:

1. Designing and developing an hardware IP for numerical compression, exploiting the compression factor of posits from 32-bit floating point numbers; this IP macrocell is called Light PPU (Posit Processing Unit)
2. Designing and developing a complete Posit unit capable of performing all real arithmetic operations between posit numbers; this IP macrocell is called Full PPU (Posit Processing Unit).

The above 2 IP macrocells, light PPU and Full PPU, have been designed in SystemVerilog at RTL level and implemented in FPGA in WP2, while in WP6 the Posit IPs have been tested via both:

- mini-apps (classification CNN applied to benchmark applications like MNIST, CIFAR10, GTRSB-German Traffic Road Side Benchmark)
- collaboration with Fraunhofer to test Posit in the Oild&Gas RTM application by using a Posit Test Array

In both cases the two hardware IPs are expected to be integrated in larger hardware platforms, such as complete RISC-V processors (Ariane CVA6, Pulpino+Ibex) or in other accelerator complexes such as a Posit Test Array that is also carried out in EPI2 SGA2 together with Fraunhofer. The hardware IP integrations in existing systems are the following:

1. HW IP for posit compression as an execution unit inside the Ariane CVA6 RISC-V platform.
2. HW IP for posit arithmetic as a full real number execution unit for the Ibex 32-bit core, inserted in the larger Pulpino SoC ecosystem.
3. HW IP for posit arithmetic as an accelerator co-processor in the Posit Test Array, paired to an ARM main host processor for posit computation offloading.
4. HLS blackbox for the Apeiron framework to provide a high-level C++ interface for high-level synthesis, exploiting RTL co-simulation.

To be noted a pure software library called CppPosit is available from University of Pisa. The results of its implementation on multiple platforms have demonstrated thatthe use of Posit via a pure software library can be useful to verify the functionalities of Posits (for example the accuracy of Posit8 and Posit 16 when used to implement Convolutional Neural Networks in training and inference instead of using FP32), but if the computing platform does not support in hardware a Posit Processing Unit then the computation speed using Posit is lower than using FP32 (since the computation of Posit will be reversed to a computation using the FPU).
This is why for the Smart Cities benchmark developed on heterogeneous platforms made of a general purpose multi-core CPU (ARM Neoverse, ARM Cortex-A or x86) plus a NVIDIA GPU (T4 or A100), the mixed-precision processing has been implemented not using Posit (NVIDIA GPUs have no Posit hardware support) but mixing the formats (e.g. FP16, FP32) supported by the NVIDIA GPU used.

3.6 Secure HPC service

As in the description of work in the grant agreement the project Textarossa foresees the development in WP2 of secure hardware IP macrocells to be implemented in FPGA technology that aim at extending the secure capabilities of the IP developed in EPI SGA1.
From a security point of view, in EPI SGA1 a set of cryptographic accelerators called crypto tile was developed by UNIPI (and released to SiPearl in order to be integrated within Rhea1). They support in hardware the implementation[1] of:

- symmetric cryptography: AES 128bits/256bits with 9 cipher modes: ECB, CBC, CFB, OFB, CTR, CMAC, CCM, GCM, XTS
- hashing: SHA2, SHA3 for computation of digests on 224, 256, 384 and 512 bits and supporting SW aided high-level hash schemes (HMAC)
- asymmetric cryptography (public-private keys) based on Elliptic Curve Cryptographic schemes such as Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic Curve Diffie Hellman (ECDH), ECIES (Elliptic Curve Integrated Encryption Scheme)
- TRNG/CSPRNG (True Random Number Generator/Cryptographycally Secure Pseudo Random Number Generator)

[1] https://www.european-processor-initiative.eu/wp-content/uploads/2020/07/EPI-Technology-FS-CryptoTile.pdf

The aim of the secure IP development in WP2 of Textarossa, implemented in FPGA technology, is to extend the security capability in EPI SGA1 and provide inputs for the further security work in EPI SGA2.

Particularly, the activities in WP2 of Textarossa for security aim at supporting new schemes for hashing functions such as SHAKE, an evolution of SHA3 that is adopted in digital signature generation and verification with the new post quantum crypto algorithms, recently approved by NIST such as Crystals-Dilithium. Differently from SHA2 and SHA3, SHAKE128 and SHAKE256 allow an arbitrary output length, which is useful in applications such as optimal asymmetric encryption padding.

Moreover, WP2 of Textarossa aims also at implementing an IP to support in hardware some operations for lattice RLWE (Ring Learning with Errors) algorithms. RLWE algorithms are both at the base of post quantum crypto, recently approved by NIST, such as Crystals (Cryp-tographic Suite for Algebraic Lattices) -Kyber and Dilithium, and of SW libraries recently developed for homomorphic encryption, such as Microsoft SEAL embedded library.
As of today, algorithms for advanced security services such as post quantum crypto or homomorphic encryption are still not frozen, not yet standardized.
Therefore, in WP2 of Textarossa the IP for security, written in SystemVerilog at RTL level is verified vs. code written in C and released by developers of Crystals algorithms and of the SEAL embedded Microsoft library.

Unfortunately, within the time frame of Textarossa, ending in H1 2024, it will not be possible to develop system level applications using security algorithms that are still not frozen. This is why in the description of the work to be performed detailed in the grant agreement of Textarossa the development of a system level application using the security IP developed in WP2 has been not foreseen.
The output of Textarossa activities related to IP security development in WP2 will be used as starting point for security development in EPI2, where system level applications will be developed. This contribution is possible thanks to the fact that some key partners of Textarossa such as UNIPI are also key partners in EPI2 SGA2.

Particularly, EPI SGA2 foresees the development of an application for secure over the air update of SW from a cloud server to distributed ECUs, using an algorithm for signature generation and signature verification based on SHAKE. For this application, in EPI SGA2, a SHAKE accelerator derived from the IP in Textarossa will be used.
In Textarossa, the security IP has a generic AXI4 interface and a FPGA implementation target while in EPI SGA2 the SHAKE crypto accelerator will be optimized to be integrated within the multi-core RHEA processor.
EPI SGA2 foresees also the development of a server-edge secure connectivity exploiting Microsoft SEAL library, using a secure hardware accelerator for lattice algorithms developed starting from the output of Textarossa WP2.
Also for this IP, in Textarossa, the security IP for lattice algorithms has a generic AXI4 interface and a FPGA implementation target, while in EPI SGA2 the crypto accelerator will be optimized to be integrated within the multi-core RHEA processor.

The aim of the IP cell for security in development in WP2 of Textarossa is trying to add hardware security features missing in EPI SGA1, and that can be an input for the hardware security activity in the project EPI SGA2 for both ARM-based GPP like that developed by SiPearl in EPI1/EPI2 and RISC-V-based accelerator like EPAC. Indeed, the interface of what is proposed in WP2 Textarossa for security is standard, AXI4 as mentioned above, and hence can be integrated with both ARM and RISC-V based computing systems.

3.7 Communication IP

The INFN Communication IP implements a n-D Torus direct network for FPGA accelerators, allowing low-latency data transfer between processing tasks deployed on the same FPGA (intra-node communication)

and on different FPGAs (inter-node communication). Being the key enabling technology behind the distributed implementation of dataflow applications in the APEIRON framework, the set of project objectives covered by its development, and the way to assess their achievement, are largely overlapped with those already described in section 3.3 - APEIRON:

- Objective 1 – Energy Efficiency. See section 3.3. Furthermore, depending on the availability of the power modelling tools developed in Task 4.5, a more detailed assessment of the energy cost for data movements through the communication IP will be performed.
- Objective 2 – Objective Sustained application performance. The performance of a distributed dataflow application depends heavily on those of the underlying network infrastructure. For this reason we identify one-way latency and communication bandwidth as KPI for the Communication IP for this objective. See also section 3.3.
- Objective 4 - Seamless integration of reconfigurable accelerators. The IP enables the deployment of distributed dataflow applications over a multi-FPGA execution platform.
- Objective 5 - Development of new IPs. See section 3.3.

3.8 HW fast scheduler IP

The main objective of developing a hardware IP for fast task scheduling is to provide an effective and efficient way to send tasks to FPGA accelerators . The scheduler IP allows to offload the process of scheduling tasks into individual accelerators and keep track of accelerator status and finished tasks. This reduces the communications and synchronizations between host and FPGA accelerators, increasing overall performance.

We envision two main fields of application for the IP developed in this deliverable. The first one is as part of the OmpSs@FPGA framework as explained in section 3.4. The fast task scheduling IP will be an integral piece of the hardware runtime (in fact it will be the hardware runtime itself). In the OmpSs@FPGA framework application the Fast Task Scheduler is expected to communicate with the host CPU fast enough that it doesn't represent a bottleneck to the scheduling of small tasks to FPGA accelerators as it happens with software schedulers. This will allow task-based designs to be competitive and even outperform streaming applications by improving the shared use of FPGA resources.

The second application scenario of the Fast Task Scheduling is the interconnection of different CPU cores and/or accelerators improving the performance of task-based programming models (like OpenMP or OmpSs). It has been demonstrated that the runtime overhead can be the main bottleneck in the performance of manycore systems as the number of tasks should increase with the number of cores to take advantage of large systems. For this kind of problems, we aim to integrate the Fast Task Scheduling IP with a RISC-V manycore system and demonstrate significant performance improvements.

The HW fast task scheduler is related to the following project objectives and strategic goals as stated in the DoA:

- Objective 1 - Energy efficiency. The IP reported in this deliverable is designed to be integrated in an FPGA or attached as a runtime accelerator to a manycore system. It provides two ways of increasing energy efficiency: a first-order effect by improving the energy efficiency of the task-based runtime and, a second-order effect that is achieved by improving the efficiency of the application being executed using the runtime.
- Objective 2 - Sustained application performance. As with Objective 1, the IP reported in this deliverable contributes to sustained application performance: by improving the performance of the task-based runtime and also, by improving the performance of the application being executed using the runtime. As a fast task scheduling effectively increases application available parallelism, this second effect improvement is expected to be significant.
- Objective 3 - Fine-tuned thermal policies integrated with an innovative cooling technology. The Fast Task Scheduling IP is expected to be able to work with the software part of the runtime by

either, providing it with information about the power consumption of the tasks and/or enabling the thermal control system (in software) to actuate over the accelerators if necessary.

- Objective 4 - Seamless integration of reconfigurable accelerators. OmpSs@FPGA runtime allows for seamless integration of reconfigurable accelerators (as detailed in Deliverable 4.1 and Deliverable 1.4). As an integral part of the framework the IP should allow for scheduling of tasks that are either specific to an accelerator or destined to be executed in a general-purpose unit.
- Objective 5 - Development of new IPs. This deliverable reports the development of a new IP dedicate to scheduling tasks, so it directly tackles objective 5.
- Objective 6 - Integrated Development Platform. As part of the OmpSs@FPGA runtime the IP reported in this deliverable will be used in applications executing on the IDV-E platform. It is important to highlight that IDV-E features a host CPU (ARM based) that has never before been used to drive computation in a PCIe attached FPGA. Developing the system in a way that is compatible with new different CPUs helps ensuring new host CPUs (like EPI CPUs) will be able to drive this kind of computations in the future.

3.9 Memory hierarchy optimization and runtime systems

Memory hierarchy optimization is a critical subject when extracting performance out of FPGA heterogenous systems like the IDV-E platform. Also, it usually is a time-consuming problem that requires significant development time to programmers. As part of our OmpSs@FPGA runtime development explained in section 3.4, we aim to integrate and evaluate different memory optimization features that we expect will help programmers to obtain more performance in a nearly transparent way. These features should be integrated for compatibility inside the Fast Task Scheduler IP explained in section 3.9. This effort is expected to address the following objectives of the DoA:

- Objective 1 - Energy efficiency. Better memory management is expected to help with a more efficient computation and consequently, with more energy efficiency in the IDV-E platform.
- Objective 2 - Sustained application performance. As with Objective 1, a better memory management by the runtime will help with performance by improving the performance of the task-based runtime and also, by improving the performance of the application being executed using the runtime.
- Objective 4 - Seamless integration of reconfigurable accelerators. The objective of the memory optimization by the runtime is that accelerators profit from it without programmers' intervention contributing to the performance of the seamless integrated reconfigurable accelerators.
- Objective 5 - Development of new IPs. The memory management in the IDV-E platform will be done by the hardware runtime that is partly integrated in the Fast Task Scheduler IP. Other parts of the memory management will not be integrated in the FTS IP but in the hardware part of the freely released OmpSs@FPGA framework so they will also be freely available to the IP development community.
- Objective 6 - Integrated Development Platform. As part of the OmpSs@FPGA runtime the memory management will be used in applications executing on the IDV-E platform contributing to the performance and usefulness of the platform.

3.10 TAFFO

TAFFO is a precision tuning framework capable of automatically converting application codes to mixed precision, given appropriate contextual information provided by the programmer in the form of annotations of the source code. The annotations that must be provided only affect a small part of the code to be transformed, saving significant developer time and effort. Additionally, TAFFO guarantees the absence of overflow errors caused by the reduced precision, and minimizes the error bound as much as possible exploiting static analyses. TAFFO supports C and C++ applications, including those exploiting OpenMP, CUDA and OpenCL. This broad support for industry-standard parallel and heterogeneous

computing programming interfaces allows TAFFO to be readily applied to concrete use-cases, included other components of the overall Textarossa framework.

In particular, TAFFO has already been employed for optimizing General Matrix-Matrix products (GEMM) kernels such as those used by INRIA-MathLib. The utilization of TAFFO for implementing mixed precision on a GPGPU platform supporting CUDA (specifically an Nvidia GeForce RTX 3070 GPU) allowed to achieve a 40% speedup over the single-precision floating-point-based baseline with a relative error in the output below 0.1%. These positive results make it straightforward to propose the integration of TAFFO with MathLib. This integration task requires an extension of MathLib to provide TAFFO with the appropriate information for the compile-time mixed-precision data type allocation to ensure that application requirements for precision are fulfilled.

Another application where TAFFO can be exploited for achieving a better precision-performance trade-off is the optimization of Sparse Matrix-Vector product (SpMV) kernels for GPGPUs. Previous works[2] already demonstrated that great benefits can be achieved by performing loop-splitting optimizations on such kernels, in order to employ lower precision for only the least precision-sensitive parts of the computation. However, it would be of great benefit to the users of SpMV kernels if this kind of tasks could be performed automatically, as the development effort required for adopting this specific kind of mixed precision in an application with current state-of-the-art tools is significant. TAFFO is the ideal foundation on which to build optimizations requiring data-dependent precision selection, as it already provides the foundational analyses required for such task. The analysis of this potential use-case is currently in progress as a joint effort between CINI (PoliMI) and CNR.

3.11 Automatic instrumentation of RTL to create power monitors of hardware accelerators

In nowadays FPGAs for HPC applications, the reference solution consists of energy management policies based on dynamic voltage and frequency scaling (DVFS) and offload computation performed by GPU-based servers. However, the HPC evolution process highlighted the possibility to implement the most computing-expensive parts of the application on FPGAs.

FPGAs used in this field can expose up to 200W of thermal design power. This level of power requires dedicated run-time power optimization techniques, since such contribution needs to be considered carefully when designing such devices. The power monitor addresses this specific problem.

The automatic RTL instrumentation avoids the intervention of the designer to inject power monitors inside the design and implement the related glue logic. Moreover, the power monitoring infrastructure used to estimate the power monitors allows us to monitor the power consumption of any hardware accelerator, in a precise way, within large FPGAs for HPC. These power monitors can be reconfigured dynamically for any change in the set of hardware accelerators implemented into the FPGA.

This methodology will be validated on a Full Posit Processing Unit (FPPU) developed by University of Pisa as first step. The FPPU is a computing module implementing a Posit format which, in contrast to standard IEEE 754 represents numbers with a much bigger and more accurate range. The final plan includes the possible validation of the methodology to generate the runtime power monitoring infrastructure for an IP Router Switch implemented on FPGA developed by INFN (University of Rome).

3.12 Multi-level thermal management

The purpose of the multi-level thermal management policy is to ensure adequate cooling of computing devices. Compared to standard approaches to thermal management, it takes advantage of the evaporative cooling solution to limit reducing the operating frequency, thereby improving computational performance.

[2] https://ieeexplore.ieee.org/abstract/document/9980904

At the same time, the policy adapts to the computational workload to avoid over-provisioning of the available cooling capacity.

Temperature rise in integrated circuits is governed by two timescales. The first timescale is induced by the thermal capacitance of the silicon die, which due to its small physical size results in fast temperature transients, which in modern HPC chips can vary from milliseconds to tens of milliseconds. The second timescale is due to the thermal capacitance of the heat dissipation solution, which is significantly bulkier than the silicon chip, resulting in considerably longer timescales varying from seconds to minutes.

As such, when an increase in power dissipation caused by computational load transients occurs, temperature must first be kept under control using fast actuators such as dynamic voltage and frequency scaling (DVFS). Simply increasing the coolant flow rate would not be fast enough. However, reducing operational frequencies reduces the dissipated power at the expense of a performance degradation. In the absence of a controllable evaporative cooling solution this performance degradation will persist as long as the required power consumption of the computational devices exceeds the cooling capacity. This is what happens in commercial thermal policies such as Intel Turbo Boost, where the boost frequency can only be kept for a limited period of time of high CPU activity, after which the frequency is reduced to the base value.
To overcome this limitation, the proposed multi-level thermal management policy is of hierarchical nature. It adds to the system a second control loop acting on the evaporative coolant flowrate, with the aim of taking advantage of the increasing dissipation heat flux caused by two-phase evaporative cooling to partially restore peak operating frequency and provide sustained high performance operation while keeping the operational temperature under the specified threshold.
This multi-level thermal management policy will be implemented in at least one of the two IDVs (access to DVFS actuators and temperature sensors is required to implement the policy) and will be tested using Textarossa applications.

3.13 Two-phase cooling

The increasing demand for data processing in recent years along with advances in processor technology produced a rapid growth in power capacity of server electronics. In this context, thermal management of data centres in terms of cooling high thermal power densities of new processors and using the so called “free cooling” has become a significant challenge for thermal engineers. The traditional air-cooling technique has approached its heat dissipation limit, therefore it is crucial to provide new thermal control solutions for upcoming demanding datacentres. In the present project, InQuattro proposes an innovative cooling solution based on two-phase flow mechanically pumped loop. The innovative feature of this system is the use of latent heat transfer for cooling electronic devices. Compared to traditional cooling systems, significantly higher heat transfer coefficients can be achieved at significantly low flow rates and pumping power.

The main objective of the project is to install the new two-phase cooling system in the two node platforms provided by ATOS (IDV-A) and E4 (IDV-E). Two cooling prototypes will be designed, manufactured and installed in the two nodes. These platforms will spread from FPGAs to high-power GPUs. These prototypes have the possibility to regulate the cooling temperature of the thermal sink, where the waste heat is rejected. This will be crucial to verify the capability of the cooling technology to work with the high ambient temperatures of the summer season (up to 40°C - 45°C). This is an important objective towards high efficient data centres to avoid the use of energy-intensive chillers and cooling towers with the use of considerable amount of water per day.

After the installation, the two platforms will be tested with specific stress tests to verify the reliability of the new cooling solution. All the parameters of the cooling loops will be measured and recorded. These data will be used to optimize the control system of the cooling loop. Tests will be performed at different simulated external temperatures in the range 20°C - 45°C.

4 General evaluation methodology for applications

This section discusses a common strategy for benchmarking and evaluating heterogeneous, mixed-precision. Dynamic runtime systems applications are also discussed.

4.1 Heterogeneous applications

The complexity of a heterogeneous computing platform such as the Textarossa project requires the use of a common methodology to perform power measurements, in order to manage a trade between computational power and energy consumption. For this purpose, a dedicated working group has been created within the project. The complete results of their activities are summarized on the technical document “Methodology for Power Measurement in the TEXTAROSSA Project”, which will be the part of deliverable D1.4. In the following we summarise the most relevant topics of the document.

4.1.1 CPU

Textarossa project will deal with two kinds of CPU architectures: x86_64 (AMD Milan/Rome, Intel Sapphire Rapids) and ARM V8.2 64 bit (AMPERE Altra Max).

4.1.1.1 x86_64 Architectures

Most modern processors, including Intel processors, provide Running Average Power Limit (RAPL) interfaces for reporting the accumulated energy consumption of various power domains of the CPU chip, attached DRAM and on-chip GPU. The update interval of the RAPL energy counters is approximately one millisecond. The RAPL energy reporting feature has been available for many generations on Intel SoC products, and energy reporting is standard practice for the industry. Intel processors utilise this energy information for internal SoC management purposes, such as control of SoC power limits in association with Intel® Turbo Boost Technology power limit settings within the SoC. This RAPL energy data is exposed to the platform via the host-software-accessible model specific registers (MSRs) such as *MSR_PKG_Energy_Status* and *MSR_PP0_Energy_Status*. This allows software to use the RAPL energy data for observation, telemetry, and/or inputs to platform-level power or thermal control algorithms [1]. The RAPL features described above are also available for AMD processors from family 17h on.

RAPL readings are highly correlated with plug power, promisingly accurate enough and have negligible performance overhead. Experimental results suggest that RAPL can be a very useful tool to measure and monitor the energy consumption of servers without deploying any complex power meters [2].

RAPL supports multiple power domains. The RAPL power domain is a physically meaningful domain (e.g., Processor Package, DRAM etc) for power management.

Figure 1 illustrates the hierarchy of the power domains graphically.

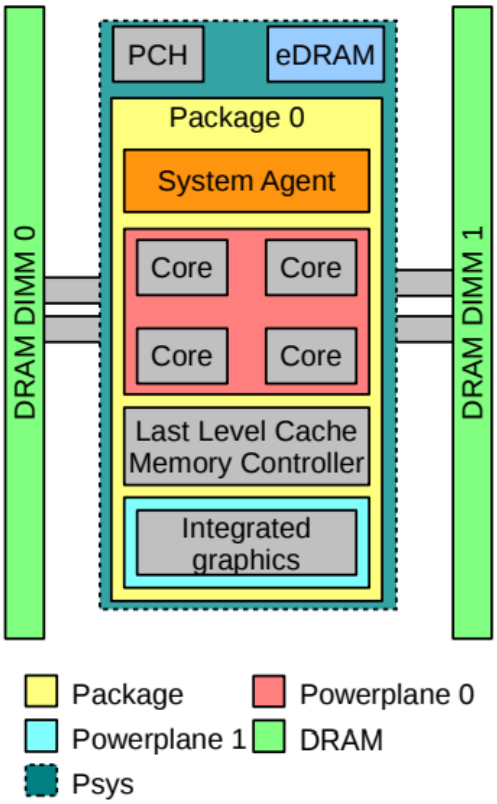

Figure 1 Power domains supported by RAPL [3]

Each power domain informs the energy consumption of the domain, allows to limit the power consumption of that domain over a specified time window, monitors the performance impact of the power limit and provides other useful information, that is, energy measurement units, minimum or maximum power supported by the domain [3].

RAPL provides the following power domains for both measuring and limiting energy consumption:

- Package: Package (PKG) domain measures the energy consumption of the entire socket. It includes the consumption of all the cores, integrated graphics and also the uncore components (last level caches, memory controller).
- Power Plane 0: Power Plane 0 (PP0) domain measures the energy consumption of all processor cores on the socket. RAPL does not support measuring the power consumption of individual CPU cores.
- Power Plane 1: Power Plane 1 (PP1) domain measures the energy consumption of processor graphics (GPU) on the socket (desktop models only).
- DRAM: DRAM domain measures the energy consumption of random-access memory (RAM) attached to the integrated memory controller. Deviations up to 20% from actual measurements have been reported for this particular domain, with a strong dependence on the specific processor architecture [4].
- PSys: Intel Skylake has introduced a new RAPL Domain named PSys. It monitors and controls the thermal and power specifications of the entire SoC and it is useful especially when the source of the power consumption is neither the CPU nor the GPU.

As Figure 1 suggests, PSys includes the power consumption of the package domain, System Agent, PCH, eDRAM and a few more domains on a single socket SoC. For multi-socket server systems, each socket

reports its own RAPL values (for example a 2-socket computing system has two separate PKG readings for both the packages, two separate PP0 readings, etc). The support for different power domains varies according to the processor model, as energy unit used: the Sandy Bridge uses energy units of 15.3 microjoules, whereas Haswell and Skylake uses units of 61 microjoules.

RAPL measurements are accurate, the correlation coefficient between RAPL and plug AC power values has been measured using the Stream benchmark on a Haswell processor, resulting in a value of 0.99 [3].

Linux supports RAPL since from kernel 3.14, access to RAPL data is possible through several mechanisms, such as reading files under /sys/class/powercap/intel-rapl/intel-rapl:0, using the perf_event interface (e.g. sudo perf stat -a -e "power/energy-cores/" executable) or using raw-access to the underlying MSR registers provided by the msr kernel module.

Several energy profiling tools using the RAPL infrastructure are currently available. We selected likwid-powermeter as reference power measuring tool for CPU tasks. likwid-powermeter is part of the Likwid toolsuite [5] of command line applications and a library for performance-oriented programmers. It works for Intel, AMD, ARMv8 and POWER9 processors on the Linux operating system. There is additional support for Nvidia GPUs.

4.1.1.2 Ampere Altra Max

According to the technical documentation provided by the manufacturer, this implementation of the ARM V8.2 64-bit architecture does not provide any RAPL-like facility for fine-grained power measurement. There are four high power domains for Altra / AltraMax processors:

- PCP power domain for CPU cores and mesh interconnects
- SoC power domain for SoC blocks, memory and PCIe controllers
- RCA power domain for PCIe/CCIX controllers
- DDR4 power domain for memory IOs and DIMMs

The Altra Max Processor Complex (PCP) features include:

- 128 Arm v8.2+ 64-bit CPU cores at up to 3.00 GHz maximum
- 64 KB L1 I-cache, 64 KB L1 D-cache per core
- 1 MB L2 cache per core
- 16 MB System Level Cache (SLC
- 2x full-width (128b) SIMD
- Coherent Mesh Interconnect (CMI):

Only PCP and SoC power domains are accessible using the Linux HWMON infrastructure, either reading the corresponding /sys/class/hwmon/hwmon0/* entries of the filesystem, or using the sensors command.
An alternative method is to use the BMC infrastructure that provides the following power data:

- PCP power domain for CPU cores and mesh interconnects
- SoC power domain for SoC blocks, memory and PCIe controllers
- DDR4 power domain for memory IOs and DIMMs

4.1.2 GPU

In the context of the project only NVIDIA GPUs will be taken into account.
To perform power monitoring on NVIDIA GPUs, a useful tool is represented by the NVIDIA Management Library (NVML): a C-based programmatic interface for monitoring and managing various states within NVIDIA GPU devices. NVML is delivered in the NVIDIA vGPU software Management SDK, which enables

third party applications to monitor and control NVIDIA physical and virtual GPUS that are running on virtualisation hosts. Using NVML APIs, we have experimented with a simple tool able to measure the power consumption of a CUDA kernel in specific points of the device code.
However, the power value obtained through NVML APIs is updated every ~20 ms. Thus, this sampling interval is not suited for a precise evaluation of the power consumption profile of a CUDA kernel with a short execution time.

4.1.3 FPGA

Textarossa adopts the XIlinx U280 as reference platform for the project. Referring to FPGAs power monitoring, POLIMI developed a methodology to deploy into generic hardware design online power monitors, capable of periodic power estimate. They evaluated 2 possibilities of implemented power monitoring:

- Software power monitors: applications providing online power monitoring in cases where platform RTL description is not accessible, at the cost of a non-negligible performance overhead, low accuracy and limited temporal resolution for the power estimate.
- Hardware power monitors: dedicated hardware delivering highly accurate power estimates at high temporal resolution and without performance overhead at the cost of changing the RTL description of the computing platform.

More information about this methodology can be found in [6].

In addition to this, Xilinx® provides a suite of software tools that can assess power supply requirements of the device throughout each stage of the design cycle. For example, Vivado® power analysis feature performs power analysis through stages of: post-synthesis, post-placement, and post-routing. Also, Xilinx Runtime library (XRT) Linux kernel driver *xclmgmt* binds to management physical function and handles the access to in-band sensors (temperature, voltage, current, power etc.). In this context, POLIMI, UNIPI and INFN have started working together in order to characterise the power consumption of IPs developed in the Textarossa project.

4.2 Mixed-precision applications

The increasing interest in complex AI and video applications involving large convolutional networks require a trade-off between the low complexity of integers and the high accuracy of floats. To this aim new arithmetic types, like Bfloat and Posits, will be considered. The KPIs will consider not only the accuracy in detection and classification for the target application (i.e. Accuracy = Number of correct predictions divided by Total number of predictions) but also computational complexity and power model.

Target applications are:
- by UNIPI-CINI, some AI and video classification applications will be used for smart cities surveillance services such as man-down (detection from camera acquired images of people laying down, useful for people rescuing in case of natural disasters, wars,…) and people detection and social distancing check and covered-face detection (useful for Covid-19). These applications are further described in Section 4.1.
- by Fraunhofer supported by UNIPI-CINI, an optimised Reverse Time Migration (RTM) algorithm that is used for oil and gas exploration in seismic imaging. This microbenchmark is further described in Section 5.3.

These applications[3] will be tested on GPU (e.g. NVIDIA GPU like T4 and Jetson AGX) and FPGA. The complexity of a heterogeneous computing platforms requires the use of a common methodology to perform power measurements and manage a trade between accuracy, computational power and energy consumption. To this aim, similarly to what described in 3.1, the following tools will be considered.

For GPUs: To perform power monitoring on NVIDIA GPUs, a useful tool is represented by the NVIDIA Management Library (NVML): a C-based programmatic interface for monitoring and managing various states within NVIDIA GPU devices. NVML is delivered in the NVIDIA vGPU software Management SDK, which enables third party applications to monitor and control NVIDIA physical and virtual GPUS that are running on virtualisation hosts.

FPGAs: Referring to FPGAs' power monitoring, CINI-POLIMI has developed a methodology to deploy into generic hardware design online power monitors, capable of periodic power estimate. This methodology will be applied to the Posit Processing Unit designed by UNIPI-CINI to be integrated in FPGA technology.

4.3 Dynamic runtime system applications

The performance of applications based on runtime systems is impacted by 1) the way the applications are parallelized, 2) the internal implementation of the runtime systems, and 3) the scheduling decisions taken at runtime to distribute the tasks over the processing units. Concerning 1) the performance is clearly application dependent and is left aside from the current description. For 2), the internal implementation of a runtime system can be evaluated by measuring its overhead and its capacity to potentially hide data movement with computation when possible. We do not expect that including a new hardware device will change the quality of an existing runtime system, therefore we recommend performing sanity check but do not consider it relevant to include these aspects in the benchmarking in Textarossa. This is why 3) is certainly the more important criterion. To evaluate the scheduling, we propose to study the makespan and the amount of memory data transfer. With this aim, we propose to benchmark runtime system-based applications with three metrics: makespan (duration of the execution) in seconds, amount of memory transferred in GB, and occupancy of the processing units in percentage.

[3] Because of limited budget RTM will be evaluated on Posits only.

5 Individual evaluation plans

In Section 2, a high-level overview of applications is presented emphasizing which Textarossa hardware and software outcome will be demonstrated for each of the use cases. These applications represent a wide range of scientific domains and problems that are solved. This is the reason to introduce an individual evaluation plan. For each application it is explained briefly: i) the reason to improve, ii) how project objectives are targeted, iii) the key performance indicators related to accuracy (if needed), computational and energy efficiency, and iv) the evaluation plan. The KPIs and used technologies and tools (discussed in Section 3) are then summarized for each application.

5.1 Smart cities

Why to improve

CINI UNIPI is working on some AI and video classification applications that will be used for smart cities surveillance services such as:

- man-down (detection and successive tracking, from camera acquired images, of people laying down, useful for people rescuing in case of natural disasters, wars,…)
- people detection and social distancing check and covered-face detection (useful for Covid-19 prevention).

The trade-off among computational complexity, frame-rate of the application, accuracy of the detection and classification need to be improved.

Project objectives and strategic goals addressed

Energy efficiency - Implementing the smart cities surveillance algorithm on different platforms: e.g. heterogeneous CPU-GPU integrated at blade level and using for CPU both ARM (Neoverse N1 or Cortex-A) and x86 (I7 or Xeon) and for GPU T4 or A100, or integrated at SoC level like in Jetson-AGC and Jetson-Orin. By comparing time and energy performance of different platforms:

- with CPU only or CPU+GPU assess the usefulness of an accelerators.
- with ARM vs x86 based CPU to assess the usefulness of ARM cores for HPC

Sustained application performance - This activity is devoted to exploit hybrid architectures. By comparing time and energy performance of the different platforms:

- with CPU only or CPU+GPU assess the usefulness of an accelerators.
- with ARM vs x86 based CPU to assess the usefulness of ARM cores for HPC

Seamless integration of reconfigurable accelerators - The accelerator used will be based on NVIDIA GPU. Assessment of the easy integration of ARM based multicore CPU similar to that adopted in EPI (Rhea) with Nvidia GPUs will be made.

EPI - For the tests will be used ARM-based platforms like Ampera Altra (with ARN Neoverse N1) and like Fujitsu A64FX (with ARM SVE) that are representative of the core used in Rhea (EPI1). Assessment of the usefulness of ARM cores similar to those used in EPI for HPC applications like smart cities surveillance will be made.

Opening for new usage domains - Adoption of HPC for smart cities services. Adoption of HPC not only for scientific calculus but also for smart cities surveillance involving analysis of video streams.

KPIs

KPIs will be:

- the achieved frame-rate,
- accuracy of the detection and classification (i.e. Accuracy = Number of correct predictions divided by Total number of predictions)
- power consumption of the application implemented on the target platform

Evaluation

The evaluation will be carried out porting the algorithms in different platforms: e.g. heterogeneous CPU-GPU integrated at blade level and using for CPU both ARM (Neoverse N1 or Cortex-A) and x86 (I7 or Xeon) and for GPU T4 or A100, or integrated at SoC level like in Jetson-Orin.

5.2 MathLib

Why to improve

CNR is working on a mathematical software library for hybrid architectures, featuring NVIDIA GPUs at node level [7][8]. Mathematical software libraries provide a large resource for high-quality and reusable software components upon which applications can be rapidly constructed. They are building blocks for solving main mathematical problems, including radically new algorithms and methods at a low level that domain scientists can transparently reuse in form of basic components with very little need of specific mathematical and computer science expertise. CNR is developing computational kernels required in sparse matrix computations and iterative linear solvers which are widely applied in Scientific Computing and Data Analysis. Focus is on node-level efficiency, by exploiting at the best Nvidia GPU high throughput, and scalability when multiple nodes are needed for computations whose dimensions largely exceed the memory resources of a single computing node, such as those stemming from leading-edge HPC applications.
The kernels include:

- Sparse matrix – vector multiplication (SpMV);
- Sparse matrix power kernel (SpMPK);
- Sparse matrix – matrix multiplication (SpMM);
- Maximum Weight Matching in undirected graphs (MWM);
- Communication Avoiding Conjugate Gradient method (CACG);
- Algebraic MultiGrid preconditioners (AMG).

Project objectives and strategic goals addressed

Energy efficiency - As mentioned in the "4 Pillars Framework for Energy Efficient HPC Data Centers", one of the basic guidelines in energy efficient computing is the optimization and acceleration of algorithms and software libraries that provide a reduction of the elapsed time of applications and, as consequence turn, a significant cut in energy consumption. We optimize MathLib numerical kernels to have large reduction in execution times by exploiting GPUs at the node level and consequently reduce energy consumption. We will demonstrate benefits in terms of increasing problem size (dofs) per Watt to solution, by effective exploitation of multiple GPUs on single node as well in a multiple nodes setting. Energy measurement tools proposed in projects will be used according to the methodology described in deliverables from WP1.

Sustained application performance - Our main activities are devoted to exploit hybrid architectures to solve sparse linear systems up to ten billion of dofs on thousands of GPUs. We will demonstrate algorithmic and implementation scalability in solving systems up to billions of dofs and benefits with respect to Nvidia GPUs state of the art libraries.

ETP4HPC - Our algorithms and their software implementations aim to face some of the challenges of the SRA5 in the Research Domain on *Mathematical methods & algorithms, in the context of robust methods and algorithms enabling extreme scalability. We define new algorithms and parallel design pattern aimed to combine different forms of parallelism of heterogeneous architectures as well as to reduce data communication and synchronization costs to enable extreme scalability in sparse matrix computation.* New mathematical libraries for sparse matrix computations, including communication avoiding Krylov solvers and preconditioners, will be introduced.

KPIs

Main objective is a sustained performance and scalability for solving problems at extreme scales. Therefore, main parameters will be execution times and speedup in strong and weak scalability regime. Concerning iterative linear solvers, specific parameters also include number of iterations to reach a given accuracy and execution time per iteration for increasing number of unknowns and parallel cores. Memory footprint is also an issue to efficiently face extreme scales; therefore, key parameter will be also memory requirements of algorithm implementations. In the same way, parameters related to energy efficiency will include number of iterations per Watt and number of problem unknowns (dofs, degree of freedom) per Watt.

Evaluation

The plan for final evaluations on some well-known benchmarks, such as sparse linear systems coming from discretization of scalar partial differential equations of Poisson-type, includes the use of some clusters of hybrid nodes embedding Nvidia GPUs and, when possible, the IDV based on Nvidia GPUs from WP5. Main risks are related to the usage of tools, eventually proposed in the project, for energy consumption measurements, and the need to have access, at a reasonable stage of the project, to the IDV-A based on Nvidia GPUs.

5.3 RTM

Why to improve

Reverse Time Migration (RTM) is used for oil and gas exploration in seismology. Migration algorithms usually need to digest input shot data up to the terabyte range to create 3D images. Days and weeks of cluster compute time are common. For this reason, the users of seismic algorithms are sensitive to compute time. Other migration algorithms approximate the wave equation to reduce the compute time, e.g., the Kirchhoff migration uses the high frequency ray approximation. However, these approximations have drawbacks, e.g. when it comes to resolve steep dips in salt domes. RTM is much better here in resolution but much more expensive in compute time as well. As TTI RTM algorithms are even more expensive than Isotropic or VTI RTM algorithms, the latter two are the most frequently used RTM algorithms in practice. Isotropic and VTI RTM algorithms are both quite memory bound.
Here calculations are done typically in 32bit floating point. Reduction of data precision (at least partial, so mixed precision) to 16 bits could halve the consumed memory bandwidth and double the throughput of the kernel. So the gap in cost between RTM and cheaper migration methods is reduced.

Project objectives and strategic goals addressed

Energy efficiency - If usage of reduced precision for RTM is feasible the algorithm uses memory bandwidth more efficiently increasing the number of propagations per second while not increasing the energy consumption in the same amount. Further a future reduced precision ALU might consume less energy per FLOP than a 32 bit ALU. Although not benchmarked in this project the potential in energy efficiency is estimated up to factor 2.

Sustained application performance - If RTM algorithms based on reduced precision are feasible such an algorithm would consume less memory bandwidth. Because isotropic RTM is memory bound these algorithms would speed up the number of propagations per second. Although not benchmarked in this project the expected potential speed up is up to factor 2.

EPI - Specialized HPC processors might include reduced precision ALUs in the future. Seismic applications might profit from reduced precision simulations by faster results at less energy costs if the reduced precision image quality is sufficient for seismic applications. In this case EPI processors might enter the oil and gas market.

ETP4HPC - RTM is a widely commercially applied algorithm. Having more energy efficient approaches to calculate RTM could support European oil and gas companies to evaluate seismic data sets in a more energy efficient way.

KPIs

The drawback of using reduced or mixed precision might be a reduced image quality. As the better image quality of RTM versus cheaper methods is the main reason to use RTM in the first place, retaining an acceptable level of image quality is crucial.
This work analyses the possibility to reduce the floating-point precision at least in parts of the RTM algorithm to increase the throughput of the algorithm. Important boundary conditions that need to be kept up are the accuracy and the stability of the numerical results.

Evaluation

Simple test examples are created. These contain a velocity model and shot data. The shot data and the respective source data are propagated in time given the velocity model. Using the imaging condition from these propagations in time images are computed. This procedure is performed in different reduced floating point precision formats and mixed floating point precision formats. The resulting images are compared to images which are computed fully in single precision. Seismic experts will evaluate the images.

Further the numerical stability versus the time step size will be evaluated. Here a forward propagation of the wave signal and the total energy within the 3D volume for each time step will be computed. In the stable time step region the total energy should stay constant over simulated time. Different time steps will be evaluated to determine the stable time step region numerically.

5.4 HEP

Why to improve

The necessity to be able to execute scientific code on heterogeneous architectures is evident in many domains, from High Energy Physics, genomics, astrophysics to medical physics. Extrapolated needs for the next decades surpass what standard CPU evolution can allow. The most promising path to affordable computing lies in the utilization of better performance/cost computing solutions, like those offered by accelerator technologies; on top of this, the same technologies are expected to be present in most HPC centres, and available to users only if their codes can be executed efficiently. The main obstacle is the difficulty to redesign algorithms in a suitable manner for each different architecture, due to the lack of knowledge and of manpower.
For the Textarossa project, two high energy physics applications were identified:

- a track reconstruction algorithm for the CMS detector developed by the Patatrack team [9]
- the CLUE algorithm, a cluster algorithm for high granularity calorimeters [10]

Project objectives and strategic goals addressed

Energy efficiency - CLUE and Pixeltrack are single-source code HEP applications using the Alpaka library or SYCL. They can be compiled on several backend processing architectures, thus gaining in energy efficiency by a proper selection of the execution platform, without the need of customizing the code. We expect a x2-x5 improvement factor in energy efficiency (reconstructed events/J) comparing the CPU with the CPU+GPU versions of the executable.

Sustained application performance - CLUE and Pixeltrack are single-source code HEP applications using the Alpaka library or SYCL, they can be compiled on several backend processing architectures, thus gaining in throughput by a proper selection of the execution platform, without the need of customizing the code. An analysis of the CLUE and Pixeltrack lead us to expect a x5-x10 improvement in terms of throughput (reconstructed events/s) comparing execution on CPU and CPU+GPU.

Seamless integration of reconfigurable accelerators - There is a significant risk that the objective could not be achieved since there are no SYCL or Alpaka back-end available at the moment for Textarossa reference FPGA plaftorms

KPIs

In the kind of above applications, the measurement of the latency as the delay between invoking an operation and getting its response is not a representative metric as much as the throughput, i.e., the number of computing tasks per time unit. Therefore, the KPI considered for HEP applications is a throughput metric: number of reconstructed events per second. In particular, for the track reconstruction algorithm, reconstructed events are particle tracks in the detector; instead, for the CLUE algorithm, reconstructed events are the assignment of a cluster to each point.

Evaluation

The goal is to obtain a single heterogeneous software per application that can be run in parallel on multiple backends, taking advantage of the characteristics of each architecture. To evaluate the results, the comparison of the performance obtained using the serial code running on the CPU versus the parallel and heterogeneous code running on multiple backends (CPU, GPU, FPGA) is of main interest. The performance is measured as the time spent to reconstruct N events.

5.5 NEST-GPU

Why to improve

The main reason for developing a CUDA version of the CPU-only NEST neural simulator was to tap into the large floating point compute resources available on NVIDIA GPUs, in order to speed-up the integration of the large systems of differential equations as required by the setup and dynamics simulation of complex neural networks that an in-silico neurophysiology experiment implies. Any improvement in this regard can either push the size and the complexity of what can be achieved by such experiments on non-extreme scale HPC platforms. Moreover, shrinking the power envelope could even demonstrate the feasibility for NEST-GPU to drive an embodied agent, which would be useful for robotics applications.
A more thorough description of the NEST-GPU application can be found in [11], while an up-to-date comparison to the CPU-only sibling application NEST running on a cluster and using MPI communications can be found in [12].

Project objectives and strategic goals addressed

Energy efficiency - NEST-GPU development does not directly address the energy efficiency problem but being a GPU-supporting version of the CPU-only NEST neural network simulator, the expectation is, for a defined *in-silico* neurophysiology experiment chosen as benchmark, that the much higher compute made available on the GPU can cut the runtimes so as to reduce the overall energy-to-solution when comparing between the two versions. Taking into account only the power consumption of the GPU hardware and ignoring that of the hosting platform and given testing done up to now, a tenfold reduction in energy-to-solutions seems attainable, at least for NVIDIA devices based on the Ampere architecture; for the upcoming Hopper architecture – which is advertised to have twice the thermal design power and three times the FLOPs – the assessment requires direct testing on a proper IDV.

Sustained application performance - The compute made available by the GPU has up to now shown improvements between 16× and 30× when comparing runtimes on a single GPU vs. single multicore server-grade CPUs in published works – it is up to the current work to extend the testing to a multi-node, multi-GPU setup on a proper IDV and precisely measure what can actually be achieved. Given the single GPU results obtained up to now and the fact that the proper IDV is expected to provide a GPU with an architecture advertised to have 3× the compute and the memory bandwidth compared to what NEST-GPU has been tested on up to now, a tenfold or more increase in time-to-solution performance compared to NEST running on CPU seems a conservative and attainable estimate.

Opening for new usage domains - The high density of compute offered by GPUs makes amenable to simulation spiking neural networks of magnitude (e.g. complete brain areas or even whole brains of mammalians) which on non-heterogeneous HPC platforms would require unfeasibly large deployments. Moreover, it would be useful to make available spiking neural network simulations – which are currently quite cumbersome – to the low-power, embedded platforms employed in robotics applications; by targeting embedded NVIDIA GPUs as the Jetson, NEST-GPU could usher the employ of spiking neural networks on such devices. Testing NEST-GPU on the IDV-A should verify that the simulated cortical slice can be scaled up to sizes that would be otherwise unmanageable on non-heterogeneous HPC systems of comparable footprint while the most useful outcome in sight of a robotics application would be the ability to reach 'real-time' (1 simulated second takes 1 wall-clock second). While this latter is NOT a direct objective for NEST-GPU in Textarossa, assessing the performance on the IDV can give valuable insights regarding this supplementary goal.

KPIs

As mentioned, the KPIs are those related to the size, complexity and achievability (which usually means bringing down the timeframe of a simulation to a manageable level) of a neurophysiology experiment, therefore time-to-solution (as how long it takes to simulate *e.g.* 1s of a neural network of predefined size), the synaptic activity (the ratio of synaptic events to the actual runtime) and the energy-to-solution (as the energy dissipated throughout this runtime) will be considered.

Evaluation

The evaluation of the mentioned KPIs is obtained defining some average sized network that can be representative of a sufficiently broad set of experiments and simulating via NEST-GPU 1s of activity of such network (possibly a little longer if we concede some warm-up period to let the system reach a steady state) on the reference platform while measuring the elapsed time, the number of synaptic events and the power consumption throughout.

5.6 RAIDER

Why to improve

Real-time (also called "online" in the specific context) particle identification (PID), or partial particle identification (*e.g.*, electron identification), is a critical task in High Energy Physics experiments: it enhances

the suppression of background physics events, allowing to keep the bandwidth that data acquisition systems must forward to the analysis pipeline within a manageable level.
In this section, we refer to "event" as the instantaneous physical situation or occurrence associated with a point in spacetime, characterised in our systems by different information obtained through several physical detectors.
The system implementing the PID task must face two main requirements:

1. processing latency, often bounded to a few microseconds or less;
2. processing throughput, which can be in the order of 10^7 events per second.

FPGA devices are good candidates to be used as processing nodes to implement a dedicated computing architecture to perform PID, as these devices allow the design of AI algorithms through HLS tools, and the implementation of data transport (with support for a wide set of physical and transport layers protocols) and processing stages characterized by a highly predictable and low latency. A low-latency direct interconnection between FPGA nodes allows:

+ to scale the system to meet the throughput requirements, deploying multiple dedicated computational units (CUs) on several boards;
+ to gather data streams from different detectors, possibly processed according to a multi-layer architecture performing a distributed PID task.

A desirable development of such an architecture has been identified in the CERN NA62 experiment: in fact, RAIDER application seems suitable for the timing requirements of the Level 0 of the NA62 trigger, allowing a possible implementation of a PID system based on the use of neural networks trained for ring reconstruction over the events coming from the NA62 RICH detector.
A more detailed description of this workflow can be found in [13].

Project objectives and strategic goals addressed

Energy efficiency - RAIDER application we addressed the energy efficiency goal along two directions:

- Energy efficient dataflow processing: we designed and deployed on FPGA AI-based inference processing pipelines with a very limited FPGA resource footprint. This was accomplished thanks to the accurate selection of the minimal neural network model ensuring adequate inference accuracy and to the usage of quantization techniques that allowed us to use 8-bits and 16-bits fixed point data types for the implementation on FPGA.
- Energy efficient I/O and intra/inter-FPGA communication mechanism: the application is based on the APEIRON framework exploiting the INFN Communication IP. Implementing direct communication between tasks deployed on FPGAs without involving CPU and system bus resources, the Communication IP improves the energy efficiency of the FPGA execution platform. Furthermore, part of its intra-FPGA ports can be used to implement I/O channels, again without the need of staging data on host or FPGA memory, and thus limiting data movement and the associated energy consumption.

Our workflow for the deployment of Artificial Neural Networks on FPGA starts with the modelling of the ANN in Tensorflow/Keras (see D4.1).We expect a O(10) improvement factor in energy efficiency of the RAIDER application in comparison to the ones obtained on a CPU only and CPU+GPU platforms performing the same computational task in Tensorflow/Keras.

Sustained application performance - The performance KPI for the RAIDER application is the number of processed RICH detector events per second. The RAIDER application is based on the APEIRON framework, that leverages the INFN Communication IP, whose design was motivated by the following considerations:

- The direct communication between computing tasks deployed on FPGAs avoids the involvement of the CPUs and system bus in the data transfers.
- Bypassing the intervention of the host network stack, communication latency is reduced while bandwidth for small massages is increased.

- Since communication operations are implemented on a completely “hardware” path, deterministic latency is achieved, in accordance with the real-time requirements.

The above listed features of communication IP have a direct impact on this specific project goal.
A speedup of 10x is a realistic objective for the final version of the application. We plan to reach this goal either improving the performance of the single analysis pipeline and the number of pipelines integrated on a single or on multiple interconnected FPGAs.

Seamless integration of reconfigurable accelerators - RAIDER is the main use case for the APEIRON framework, that enables the straightforward deployment of distributed dataflow applications on Multi-FPGA systems. Demonstration of the workflow for the deployment of (AI-based) distributed real-time analysis pipelines using the APEIRON framework will be conducted .

Development of new IPs - The INFN Communication IP is the key enabling technology for the APEIRON framework. As main use case forAPEIRON, RAIDER will help testing and possibly drive the refinement of the functionalities of the Communication IP. Functional and performance validation of the INFN communication IP will be provided.

Opening for new usage domains - RAIDER implements a very high throughput, distributed AI-based real-time data analysis pipeline and thus represents a good example of HPDA application. We will demonstrate the feasibility of implementing a distributed HPDA task on a multi-FPGA system, showing significant gains in terms of energy efficiency end sustained performance in comparison with alternative execution platforms of similar cost/complexity.

KPIs

Besides the processing latency per event that must be less than the experimental requirement, and so it should be considered more as a prerequisite rather than a KPI, relevant KPIs for the RAIDER application are, for given values of accuracy and purity of the (partial) PID task:

1. the number of processed events per second, *i.e.*, the throughput;
2. the number of processed events per Joule.

Evaluation

The measurement of the events/s KPI is trivial. We will leverage the power measuring tools provided by the FPGA platforms producer to assess power efficiency. Furthermore, a joint activity with the POLIMI team, aimed at instrumenting the RTL code of the communication IPs and processing kernels in order to measure their power consumption, has just started. Since this methodology has already been used successfully in the past by the POLIMI team any particular risk at the moment for the measurement of the events/J KPI is identified.

5.7 TNM

Why to improve

Tensor network methods consist of techniques that represent the quantum state of *N* qubits as a series of tensor contractions. By trading some accuracy, this enables for example quantum circuit simulators to handle circuits with many qubits that would not be feasible to be simulated with exact methods because of the exponential growth of the Hilbert space. However, depending on circuit topology and depth, this can also get prohibitively expensive. This highlights the need for tensor network methods to be executed on heterogeneous architectures to efficiently exploit parallel computing and powerful GPU computation. Before moving towards mixed-precision methods, the effect of running different precisions for a complete simulation are reported.

Energy efficiency - The application plans to evaluate energy efficiency by comparing different simulation parameters and platforms, e.g., a comparison between single-precision and double-precision simulations. The expected outcome is an baseline of the energy consumption on classical computing platforms to enable potential future comparison between classical computers emulating quantum computers and quantum computers. For example, a comparison of different classical hardware ensure to have the optimal baseline for this comparison.

Sustained application performance - The key performance indicators that we added specifically address variables relevant to quantum systems and will give future users an intuition of the resources they need to solve a problem. Still, the problem will depend on the entanglement present in the quantum system. Collecting the KPIs for different example problems will allow to predict resource allocation for computational tasks on supercomputers or other cloud services. Predicting the resource allocation plays a key role for an audience which is not specialized on the simulation side, but wants to use quantum simulations as a user to solve their problem. Potential improvements in the performance due to parallelization of a factor 2 to 4 could already affect which problems can be tackled.

ETP4HPC - The ETP4HPC discusses the goals for quantum computing and HPC in their whitepaper "<QC | HPC> Quantum for HPC". As the tensor network methods are developed to solve quantum systems – especially quantum circuits are one of the target applications – the TNM is connected to the ETP4HPC agenda. The outcome is a classical emulator for quantum circuits which scales on HPC systems. Currently, we already have access to national supercomputers like marconi100 at Cineca, i.e., the Italian National Supercomputing Center, and develop and run simulations there.A further improvement of the scaling and user-experience seems possible.

Opening for new usage domains - Quantum computing may lead to new possible algorithms for solving problems beyond the reach of todays' classical computers. This involves problems in material science, drug design, chemistry, finanance, and optimization problems.The expected outcome is a classical emulator which can serve as a bridge to quantum computers until the error rates of quantum computers reach the level for the NISQ era or the fault-tolerant threshold for quantum error correcting codes.This emulator should help us to start a dialog with potential partners at universities or in industry to prepare them for the quantum computing era.

KPIs

In the kind of above applications several KPIs can be considered for estimating computational efficiency. For a quantum simulation with tensor network methods one can evaluate the performance by looking at the number of qubits that can be simulated per second with a fixed set of convergence parameters as the bond dimension between the link in the tensors. Another KPI is the number of gates per second that can be executed in a simulation of a quantum system with a given size. A possible direction to evaluate the KPI for energy consumption is to follow up on our previous work, which compares the energy consumption of a quantum circuit on a quantum processing unit against the same quantum circuit running with tensor networks [14].

Evaluation

To evaluate the results, the performance of the software executed by using the serial code running on the CPU versus the parallel and heterogeneous code running on CPU and GPU will be compared.

The goal of the performance measurements in terms of computational time is to have a meaningful benchmark and ensure the energy measurements are executed with a code that scales according to expectations. The goal of the energy efficiency measurements is to provide a baseline for the application

which can also be used in future comparisons of classical architectures as used in Textarossa versus an execution on quantum processing units (QPUs).

5.8 Chameleon (Mathlib)

Why to improve

Chameleon is a dense linear solver based on StarPU, i.e. it is parallelised with a task-based method and relies on classical Blas functions in the tasks. Consequently, its performance is critically tied to the scheduling of the tasks and the raw performance of the Blas functions on the target processing units. Chameleon has been massively used on distributed heterogeneous computing nodes equipped with multiple GPUs. However, the study of a large-scale dense linear solver with FPGAs has never been done. This is why we want to use FPGA and see how this can improve performance and/or energy.

Project objectives and strategic goals addressed

Energy efficiency: The objective is to schedule tasks in a way that parallel executions of Chameleon consume less energy compared to existing schedulers. The focus will be on developing a generic scheduler that can be utilized by any StarPU-based application. To achieve this, we will concentrate on the development of the new scheduler and its underlying strategies, and Chameleon will be a study case for validation and benchmarking. In order to make good decisions and distribute tasks across different processing units to obtain efficient executions, we will need to build performance and energy models for the various computational kernels used in Chameleon.

Sustained application performance: The goal is to utilize accelerators, including FPGAs, to reduce the makespan and/or energy consumption. Again, the aim is to provide a generic scheduler that can be applied to any StarPU-based application.

Opening for new usage domains: The objective is to offer an optimized version of Chameleon that can be integrated into other applications. This is important as there are numerous HPC applications that rely on a parallel linear solver.

KPIs

Two main KPIs are FLOP per second and FLOP per watt because main interest is in evaluate speedup and energy efficiency can be obtained in running with FPGA .

Evaluation

FLOP per second can be obtained easily. The FLOPS per watt needs hardware counters.

5.9 ScalFMM (Mathlib)

Why to improve

The fast multipole method is a well-known approach that allows for reducing the quadratic complexity when computing interactions in n-body problems. ScalFMM has been a pioneer by proving the first implementing FMM algorithm on top of StarPU, i.e. parallelised with a task-based method. ScalFMM can be executed on distributed computing nodes equipped with accelerator devices and used CPUs and GPUs concurrently. Said differently, when the GPUs are computing kernels for which they are efficient, we use CPUs at the same time for less GPU-friendly kernels. Currently, we implemented the two major FMM kernels with CUDA such that we can use the main common HPC architectures. However, we never study its energy efficiency or the use of FPGA.

Project objectives and strategic goals addressed

Energy efficiency: The objective are the same as for Chameleon. In addition to Chameleon, ScalFMM will be used for validating our new scheduler. Of course, the computational kernels that will be port on FPGA are different as we plan to port the P2P and maybe the M2L kernels.

Sustained application performance: Same as for Chameleon.

Opening for new usage domains: Same as for Chameleon, with the exception that the Fast Multipole Method (FMM) is less used thans linear solvers. However, the FMM is still a critical component in several HPC applications.

KPIs

Two main KPIs are n-body interactions per second and n-body interactions per watt because main interest is in evaluate speedup and energy efficiency in running with FPGA. Energy efficiency will be also evaluated for the existing GPU version.

Evaluation

N-body interactions or FLOP per second is obvious. Whereas interactions or FLOP per watt needs hardware counters. Once these counters will be available, FPGA kernels for the P2P operators and benchmark will be evaluated. Comparisons when using GPUs or FPGAs to highlight the situations where one is better than the other will be carried out.

5.10 DNN Inference

Why to improve

We are interested in large scale inference of huge models (Transformer based essentially, GPT like models). The context is the following: on a set of heterogeneous resources (both CPUs and GPUs), the goal is to perform a very large number of inference tasks. We focus on optimization at the scale of a single heterogeneous node (IDV-A like) because it is unlikely to perform inference on several nodes due to communication costs. On the other hand, we are aiming at a context where there are a lot of tasks and we can scale up without difficulty (relying on data parallelism), so it's definitely an HPC like application. The objective is to optimize (latency, throughput) under memory and performance constraints (of heterogeneous resources). We identified three types of opportunities that make (non-trivial) parallel solutions at the level of the node appealing, and that have not been yet explored in the literature.

- the first one is to fulfil the memory constraints on the different nodes (which can lead to split the network on several resources in order to store the weights in a distributed way)
- the second one is of course to maximize the throughput (the number of inferences per second). This can lead to distribute the network on several nodes, in order to perform each elementary layer (GEMM or tensor operation) on its favourite resource.
- the third is to minimize the latency (maximum or average) of the inferences, i.e. the time it takes to process each inference. This can also lead to splitting the network, to execute independent branches in parallel on different resources.

We are interested in the trade-off between static and dynamic runtime (StarPU at the moment) strategies. Indeed, it is probably necessary to add a dynamic component because the performance of computing and communication resources are difficult to predict accurately, and part of the placement decisions must be made at runtime.

Project objectives and strategic goals addressed

Sustained application performance - Efficient use of accelerators to decrease latency and increase throughput. Demonstrate that throughput can be increased by leveraging the heterogeneity of resources, by showing that we can go faster collectively than by using resources individually.

Opening for new usage domains - Extending parallel computation on heterogeneous resources to inference. Increase throughput and decrease latency.

KPIs

The main KPIs are throughput and latency (sustained application performance) and the opening for new usage domain.

Evaluation

Throughput and latency are easy to measure and evaluate. The efficient usage and integration of accelerators can be evaluated by proving that the overall throughput/latency with CPUs and GPUs can be larger than the sum of throughputs/latencies on each type of resources.

5.11 UrbanAir

Why to improve

In the UrbanAir we deal with weather forecasting which then influences how pollutants are transported and dispersed within the cities. One of the challenges is to efficiently and effectively represent complex building structures which affect contaminants flow. To model the problem accurately, there is a need for vast of computational resources. In order to be able to simulate larger domains, we need to improve. CPU+GPU realisation on multiple nodes is considered to shorten execution time, and to increase in energy efficiency. Additionally, we want to investigate whether implying mixed precision can lead to increased efficiency by minimising communication time.

Project objectives and strategic goals addressed

Energy efficiency - The energy efficiency is planned to be increased by using GPU accelerators and applying mixed-precision. Tools for measuring energy consumption will be used to validate the improvements. We expect not only overall improvement in energy efficiency, but also the ability to compute larger problem size with the same energy consumption.

Sustained application performance - The application performance is planned to be increased by using GPU accelerators and possible by applying mixed-precision.

Opening to new usage domains – Although UrbanAir is primarily targeted at modelling air quality in urban areas, the same solver is now used for improving energy production from renewable energy sources.

KPIs

The main part to be adapted to heterogeneous resources is an iterative solver, with the aim of dividing it into smaller kernels. Iterations/s and iterations/watt for respectively computational and energy efficiency will be considered.

Evaluation

The kernels will be benchmarked on currently available hardware for the baseline measurements. Iterations/s will be collected programmatically, while iterations/Watt need some energy measurement tools developed on WP4. The progress will be measured on a regular basis, on the available testbed, and at the end of the project it will be compared against IDV-A and project tools.

5.12 External applications

In addition to the applications described in the project DoA, some other applications are used by the partners to better measure, improve the outcome of the project and evaluate the results. In this section we described these external applications with their related KPIs. As can be seen in the description the KPIs of these applications are closely related to the KPIs of the previously selected applications.

5.12.1 Deflate

Why to improve

Deflate [15] is a lossless compression algorithm based on the sequential use of the LZ77 [16] and Huffman [17] coding. As it is widely used (it is part of the gzip compressor) and is time demanding, we want to accelerate it through FPGA. Our objective is to reach a compression throughput which does not limit the I/O bandwidth of today NVM disks, i.e. we are targeting a compression speed in the order of 3 GB/s, not reachable by available CPU implementations.
Together with the previous throughput objective, we target also energy efficiency that will be quantified, along with the speed, through the Energy Delay Product (EDP), i.e. the product between the time spent to compress data and the energy employed for the compression.

Project objectives and strategic goals addressed

Energy efficiency - Implementation on FPGA device, which is much less power hungry than CPU. Reduction of the EDP with respect to the CPU gzip or zlib implementation.

Sustained application performance - Wide use of parallelism and deep pipelining, to sustain the reading of W bytes/cycle (target W=16). Increase of sustained throughput respect to CPU implementation (gzip or zlib implementation).

KPIs

Throughput: bytes read/s
EnergyDelay Product = (time spent to compress input data) x (energy used during compression)

Evaluation

The proposed KPIs (throughput [B/s] and EDP [Js] will be evaluated by comparing their value both for FPGA and CPU implementations of Deflate algorithm.

5.12.2 BSC-HPC Benchmark

Why to improve

The BSC-HPC Benchmark is composed of 4 different applications (N-Body, Spectra, Cholesky and Matrix Multiplication, see D1.4 for more details) that reflect the characteristics of several different typical HPC applications (apart from themselves). The applications are of interest as their behaviour reflects the expected behaviour of general HPC problems using the given programming model and consequently are a good tool to evaluate the characteristics of the OmpSs@FPGA framework or, in general, a given architecture (I.e. a RISC-V manycore using a Fast Task Scheduling hardware)

Project objectives and strategic goals addressed

Energy efficiency, sustained application performance - As described in sections 3.4, 3.8 and 3.9, the BSC-HPC Benchmark is used as a method to evaluate the improvements of the OmpSs programming model and the OmpSs@FPGA framework in a given platform (IDV-E). Our desired outcome would be to increase the performance and energy efficiency of the system by at least a 50%. Note that this means increasing it over the state-of-the-art performance in the same physical platform, so all the benefits will come from the programming model/framework/fast task scheduler IP.

KPIs

The KPIs measured are GFlops (Cholesky and Matrix Multiplication) and GPairs per second (N-Body and Spectra) for performance and GFlops and GPairs per Watt for energy efficiency.

Evaluation

The KPIs will be evaluated by measuring them when executed in the IDV-E platform and will be compared against either other previous results in different platforms or base results (using a simple version of the programming model). Note that these applications could not be executed in the IDV-E platform before the project as it is a new platform, so comparing them with other previous implementations in the same platform is not possible as other previous implementations simply do not exist.

5.13 Summary

The evaluation will be based on benchmarking KPIs defined individually per each application. However, a more global approach can be applied by using a common set of indicators defined within WP1 and WP6. Some of the KPIs are easier to measure, such as time-to-iteration or time-to-solution, as they require only small integration to the code, and no external tools or access to hardware performance counters is required. Some of them, such as energy efficiency, need additional tools and sensors enabled for the underlying hardware infrastructure.

The KPIs are related to the solved problems, and some of them are common for several applications. Individual KPIs are summarised in Table 3, however an update is expected with the next D6.2 deliverable after these are liaised with WP1 outcomes.

App name	KPI - computational efficiency	KPI - energy	KPI - accuracy
Smart cities	execution time/speedup on GPU vs. scalability vs. accuracy	Power model on target GPU and on FPGA	Yes
Mathlib-CNR	execution time/speedup/strong and weak scalability; number of iterations to a fixed accuracy/time per iteration for iterative solvers	Iterations/Watt; Dofs/Watt	Yes (user's parameter dependent)
RTM (benchmark kernel only)	Feasability of Posit based reduced precision approach to increase throughput	No	No
HEP	Events / s	Events / J	No
NestGPU	Simulated s/ s	SUP / J (Synaptic Updates per Joule)	No

RAIDER	Events / s	Events/J	Yes
TNM	Qubits / s Gate / s	Qubits / Watt s Gates / Watt s	No
DNN Inference	Throughput and latency	No	No
Mathlib-INRIA	Flops/s | Interactions/s	Flops/Watt, Iterations/Watt	No
UrbanAir	iterations/s, simulated time/s	Iterations/Watt	No
Deflate	Throughput B/s	Energy delay product Js	
HPC-Benchmark	GFlops/s, GPairs/s	GFlops/Watt, GPairs/Watt	

Table 3 Individual KPIs

The KPIs for computational efficiency are:

- Time per iteration, used by iterative solvers where performance can be judged based on how many seconds are needed per each iteration and number of iterations to a user's defined accuracy.
- Simulated time/s (or timesteps/s), used by the solvers which iterates through simulated time, the more timesteps are calculated within one second the better the performance is.
- Interactions/s, used by n-body simulations where the number of interactions between particles is representative of the performance.
- Events/s, used by trigger systems in physics experiments where we refer to "event" as the instantaneous physical situation or occurrence associated with a point in spacetime, characterised in our systems by different information obtained through several physical detectors.
- Qubits/s (and Gate/s), with an equal fixed set of convergence parameters for a quantum simulation with tensor networks method, e.g. fixed bond dimension, the performance can be evaluated looking at what is the size of the system n, in terms of number of qubits, that can be simulated within a second. In some other application, for a given n-qubit system, the performance can be evaluated looking at the number of quantum gates within a second that can be executed.
- FLOP/s, a general performance KPI to indicate how many floating-point operations per second can application achieve.
- Throughput B/s, data (bytes) read per second
- Energy delay product J*s, time spend multiplied by energy used

The KPIs for energy efficiency are similar to the computational ones, except that it is measured for every watt of power consumed.

Accuracy KPI: accuracy in detection and classification for the target application (i.e. accuracy = number of the correct predictions divided by total number of predictions) vs. computational complexity and vs. used arithmetic; accuracy in iterative linear solvers (i.e., number of correct digits in the solution, as required by users).

Table 4 provides a mapping between application and programming model, software/tools to be used and hardware to be exploited, which will be used for the final evaluation.

Application	Description	Partnerr	Platform		Processor	Programming model	Textarossa Tool/Technology	KPI to test the app
MathLib	Basic Numerical Linear Algebra kernels for dense matrices	INRIA	IDV-A		CPU/GPU	Task-based	StarPU	1) Throughput, 2) Energy efficiency
				IDV-E	CPU/FPGA	Task-based	StarPU	1) Throughput, 2) Energy efficiency
MathLib	Basic Numerical Linear Algebra kernels for sparse matrices	CNR	IDV-A		CPU/GPU	Heterogeneous, distributed/shared (MPI/CUDA)	GPU accelerators/power consumption support tools/basic software toolchain. TAFFO is under investigation for possible use.	1) Strong/Weak scalability 2) Energy Consumption (Watt/dofs)
RAIDER	RAIDER is a high throughput online streaming processing application implemented on FPGA with the APEIRON framework and belongs to the HPDA domain. Its task is to perform particle identification (PID) on the stream of events generated by the RICH (Ring Imaging CHerenkov) detector in the CERN NA62 experiment at a rate of about 10 MHz, using neural networks.	INFN		IDV-E	CPU/FPGA	Natively streaming (APEIRON). A task-based version of the computational kernel is been produced for execution with OmpSs	APEIRON/Communication IP. OmpSs/Fast Scheduler	1) Throughput [events/s] 2) Energy efficiency [events/J]
TNM	Tensor Networks Methods for Quantum System Simulation.	INFN	IDV-A		CPU/GPU.	Heterogeneous (CUDA).	GPU accelerators/power consumption support tools/basic software toolchain	Performance: Gates/s, Qubits/s. Energy Efficiency: Gates/(W s), Qubit/(W s).

HEP	Selection of High Energy Physics high level event reconstruction software: 1) Pixeltrack (track reconstruction algorithm for the CMS detector), and 2) CLUE (cluster algorithm for high-granularity calorimeters)	INFN	IDV-A		CPU/GPU	Heterogeneous computing: one source file that can be run in parallel on multiple heterogeneous backends. SYCL and Alpaka implementations.	GPU accelerators/power consumption support tools/basic software toolchain	1) Throughput [events/s] 2) Energy efficiency [events/J]
UrbanAir	EULAG model to forecast air quality over complex urban areas	PSNC	IDV-A		CPU/GPU.	Heterogeneous, distributed/shared (MPI/CUDA)	GPU accelerators/power consumption support tools/basic software toolchain. TAFFO is planned	Iterations/s, Iterations/Watt
Neural Networks	Convolutional NNs and Recurrent NNs as NN example with a focus on optimizing either inference but also computing-intensive training of Deep NN	INRIA	IDV-A		CPU/GPU	Task-based	StarPU	Throughput and latency
Nest-GPU	Spiking neural network simulation for brain modeling at Exascale. GPU-accelerated neural network simulator engine for in-silico experiments	INFN	IDV-A		CPU/GPU	Heterogeneous, distributed/shared (MPI/CUDA)	GPU accelerators/power consumption support tools/basic software toolchain	1) Throughput as time-to-solution [simulated seconds/s] 2) Energy efficiency as energy-to-solution [simulated seconds/J]
Surveillance systems (Cities)	Smart cities (people detection and tracking, people down counting in case of disaster)	UNIPI		GPP+GPU		Heterogenous	GPU accelerators/power consumption support tools/basic software toolchain	1) Throughput as time-to-solution [simulated seconds/s] 2) Energy efficiency as energy-to-solution [simulated seconds/J]

RTM	Reverse time Migration (wave equation), used for benchmarking only, not as application	FHG	Generic		CPU	Homogenous, single thread		Feasability of Posit based reduced precision approach to increase throughput

External Applications	**Description**	Author	Platform			Programming model	Tool	KPI
Deflate	Deflate/Image processing	ENEA		IDV-E	FPGA	streaming	FastFlow	throughput (GB/s), Energy Delay Product (EDP)
BSC-HPC Benchmark	N-Body, Cholesky, Spectra, Matrix Multiplication	BSC		IDV-E	FPGA	Task-based + Mixed OmpSs	OmpSs/Fast Task Scheduler	Performance (GFlops/s, GPairs/s), Throughput (GB/s, Tasks/s), Energy Efficiency

Table 4 Summary of applications and technologies

6 Future work

In this deliverable we discuss general and individual evaluation plan of the Textarossa technologies and uses cases. The next step is to benchmark each application with defined KPI, which will be the baseline measurements to compare with at the end of the project. The outcomes of this task are going to be described in the following deliverable – D6.2 Initial application benchmarks and results. The outcomes of WP1 shall be taken into account to extend the proposed evaluation metrics and the methodology of benchmarking. It is planned to derive such discussion in the next deliverable D6.2. It may be the case that the details of evaluation plan may require an update, such will be provided with the next deliverables.

7 References

[1] https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html

[2] Khan, Kashif & Hirki, Mikael & Niemi, Tapio & Nurminen, Jukka & Ou, Zhonghong. (2018). RAPL in Action: Experiences in Using RAPL for Power Measurements. ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS). 3. 10.1145/3177754.

[3] Intel Corporation 2015. Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3, System Programming Guide. Intel Corporation.

[4] Spencer Desrochers, Chad Paradis, and Vincent M. Weaver. 2016. A Validation of DRAM RAPL Power Measurements. In Proceedings of the Second International Symposium on Memory Systems (MEMSYS '16). ACM, New York, NY, USA, 455–470. https://doi.org/10.1145/2989081.2989088

[5] https://hpc.fau.de/research/tools/likwid/

[6] Cremona et al., Automatic identification and hardware implementation of a resource-constrained power model for embedded systems, Sustainable Computing: Informatics and Systems, Volume 29, Part B, 2021, 100467, ISSN 2210-5379, https://doi.org/10.1016/j.suscom.2020.100467.

[7] M. Bernaschi, P. D'Ambra, D. Pasquini, BootCMatchG: An adaptive Algebraic MultiGrid linear solver for GPUs, Software Impacts (Invited Paper). Vol. 6, 2020. https://doi.org/10.1016/j.simpa.2020.100041

[8] M. Bernaschi, P. D'Ambra, D. Pasquini, AMG based on Compatible Weighted Matching on GPUs, Parallel Computing. Vol. 92, 2020. https://doi.org/10.1016/j.parco.2019.102599

[9] Bocci A. et al, Heterogeneous Reconstruction of Tracks and Primary Vertices With the CMS Pixel Tracker, Frontiers in Big Data Journal, 2020.

[10] Rovere M. et al, A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High-Energy Physics, Frontiers in Big Data Journal, 2020.

[11] B. Golosio et al, Fast Simulations of Highly-Connected Spiking Cortical Models Using GPUs, Frontiers in Computational Neuroscience February 2021 (doi.org/10.3389/fncom.2021.627620)

[12] G. Tiddia et al, Fast Simulation of a Multi-Area Spiking Network Model of Macaque Cortex on an MPI-GPU Cluster, Frontiers in Neuroinformatics July 2022 (doi.org/10.3389/fninf.2022.883333).

[13] R. Ammendola et al, Progress report on the online processing upgrade at the NA62 experiment, 2022 JINST 17 C04002 (iopscience.iop.org/article/10.1088/1748-0221/17/04/C04002).

[14] Daniel Jaschke, Simone Montangero, Is quantum computing green? An estimate for an energy-efficiency quantum advantage.

[15] DEFLATE Compressed Data Format Specification version 1.3. https://www.rfc-editor.org/rfc/rfc1951

[16] Ziv J., Lempel A.: A Universal Algorithm for Sequential Data Compression. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-23, NO. 3, MAY 1977.

[17] D. A. Huffman. A method for the construction of minimum redundancy codes. Proceedings of the IRE, 40(9):1098–1101, September 1952.