TEXTAROSSA

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

Alessandro Lonardo

for the INFN TEXTAROSSA team

Workshop sul Calcolo nell'INFN

23 - 27 maggio 2022, Paestum

INFN TEXTAROSSA Team

@INFN Roma – APE Lab


A. Lonardo


P. Vicini


F. Lo Cicero


F. Simula


M. Martinelli


P. S. Paolucci


R. Ammendola


A. Biagioni


P. Cretaro


O. Frezza


C. Rossi


M. Turisini

@INFN CNAF


F. Giacomini


L. Cappelli

@INFN Pisa


T. Boccali

@INFN Padova


S. Montangero

The TEXTAROSSA Project

- 6 M€ budget, co-funded project
  - EuroHPC Joint Undertaking, H2020 G.A. 956831 / MISE
- 36 months (April 2021 – March 2024)
- The focus of the TEXTAROSSA project is the **HW/SW co-design of heterogeneous processing nodes to boost energy efficiency in the execution of a set of relevant scientific codes**, leveraging the best partitioning of the applications among the heterogeneous resources of the node to achieve the best trade-off between **Time-to-Solution** and **Energy-to-Solution**.

11 partners from 5 countries:
ENEA, Fraunhofer, INRIA, ATOS, E4, BSC, PSNC, INFN, CNR, IN QUATTRO, CINI (Politecnico di Milano, Università di Torino, Università di Pisa),
LTP: Universitat Politecnica de Catalunya (UPC), Université de Bordeaux.

TEXTAROSSA Main Goals

1. **Energy efficiency and thermal control**
   - innovative **two-phase cooling technology** at node and rack level, fully integrated in an optimized multi-level runtime resource management
2. **Sustained application performance**
   - efficient exploitation of highly concurrent accelerators (GPUs and FPGAs) by focusing on data/stream locality, efficient algorithms and programming models, tuned libraries and innovative IPs
3. **Seamless integration of reconfigurable accelerators**
   - by extending field-proven tools for the design and implementation such as Vitis and OmpSs@FPGA to support new IPs and methodologies
4. **Development of new IPs**
   - for mixed-precision AI computing, data compression, security, power monitoring and control, task scheduling, and low-latency communication
5. **Integrated Development Platforms**
   - by developing two architecturally different, heterogeneous Integrated Development Vehicles (IDVs),

Co-Design-centric process workflow

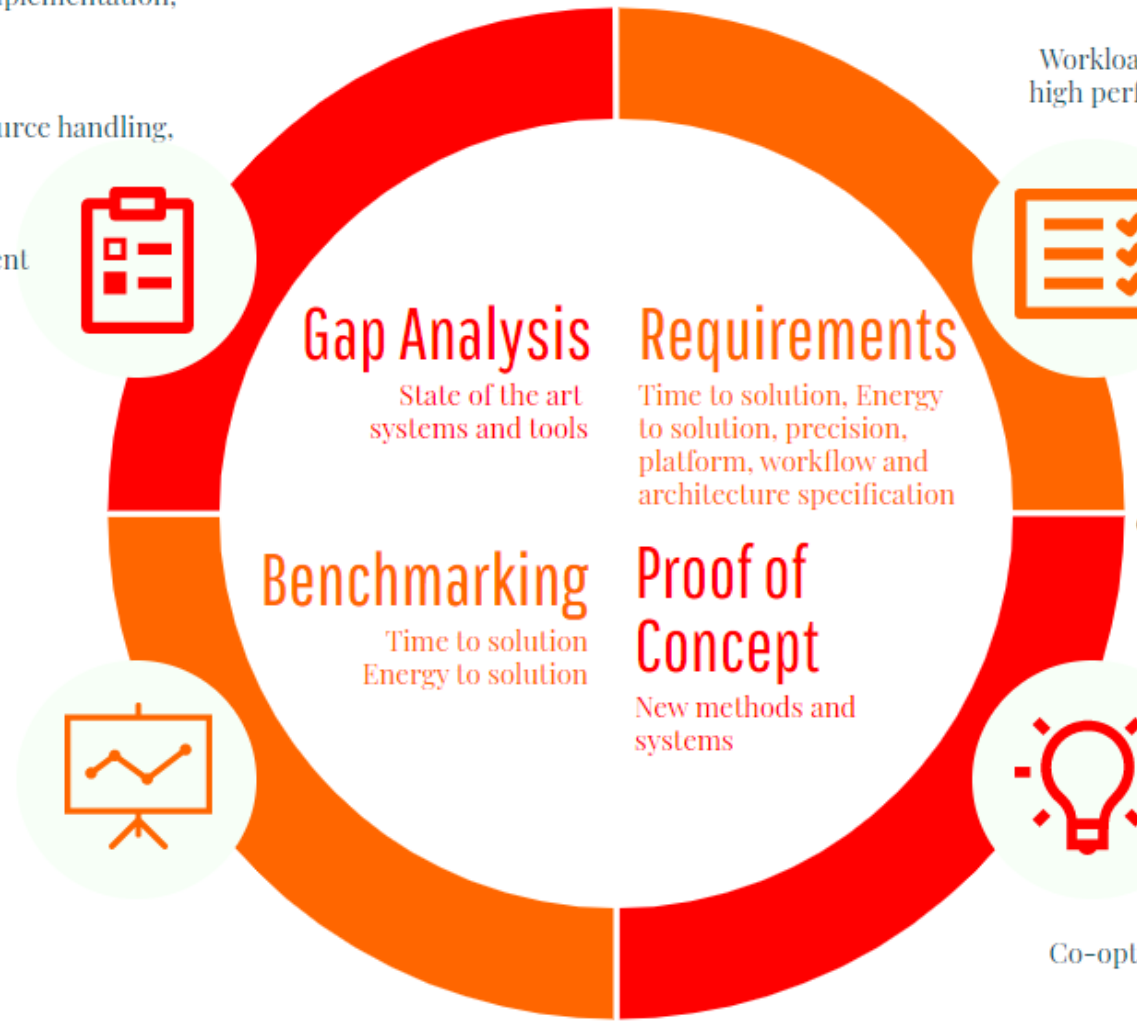
User Applications
Computing models, implementation, algorithms, AI, HPDA

Runtime Services
Execution model, resource handling, fault tolerance, I/O

Programming Models
Toolchains, development tools

System Architectures
CPU, GPU, FPGA, Transprecision, memory, I/O, network

Hardware Platforms
Node and rack



User Applications
Time/Energy to solution, Precision, Data locality

Runtime Services
Workload-specific, parallel workflow, high performance I/O, time to solution

Programming Models
Design, verification, IP integration, emulation debugging, optimization

System Architecture
IP/SoC, low power, ARM SVE, RISC-V, bandwidth/latency

Hardware Platform
Power supply/management cooling, thermal management

User Application
New methods and algorithms

Runtime Services
Energy-aware management

Programming models
Toolkits for heterogeneous architectures

System Architectures
Co-optimized with toolchain and API

Hardware Platforms
Energy efficient and heterogeneous

Applications and Libraries driving co-design

MathLib
CNR, FHG, INRIA, ENEA

High performance numerical methods for HPC, HPDA, HPC-AI, including linear algebra, and graph computation

UrbanAir
Air Pollution Model
PSNC

Modelling and forecasting of the concentration and dispersion of air pollutants at meso-scale and city-scale

RAIDER
INFN

Real-time data analytics on heterogeneous distributed systems, processing data streams through Deep Neural Networks

TNM Quantum Simulation
INFN

Tensor Network Method to study in and out of equilibrium properties of strongly correlated many-body quantum systems

Brain Simulation
DPSNN
INFN

Distributed and Plastic Spiking Neural Network model of the brain cortex behavior

High Energy Physics
INFN

Optimization of high energy physics simulation and data analysis frameworks

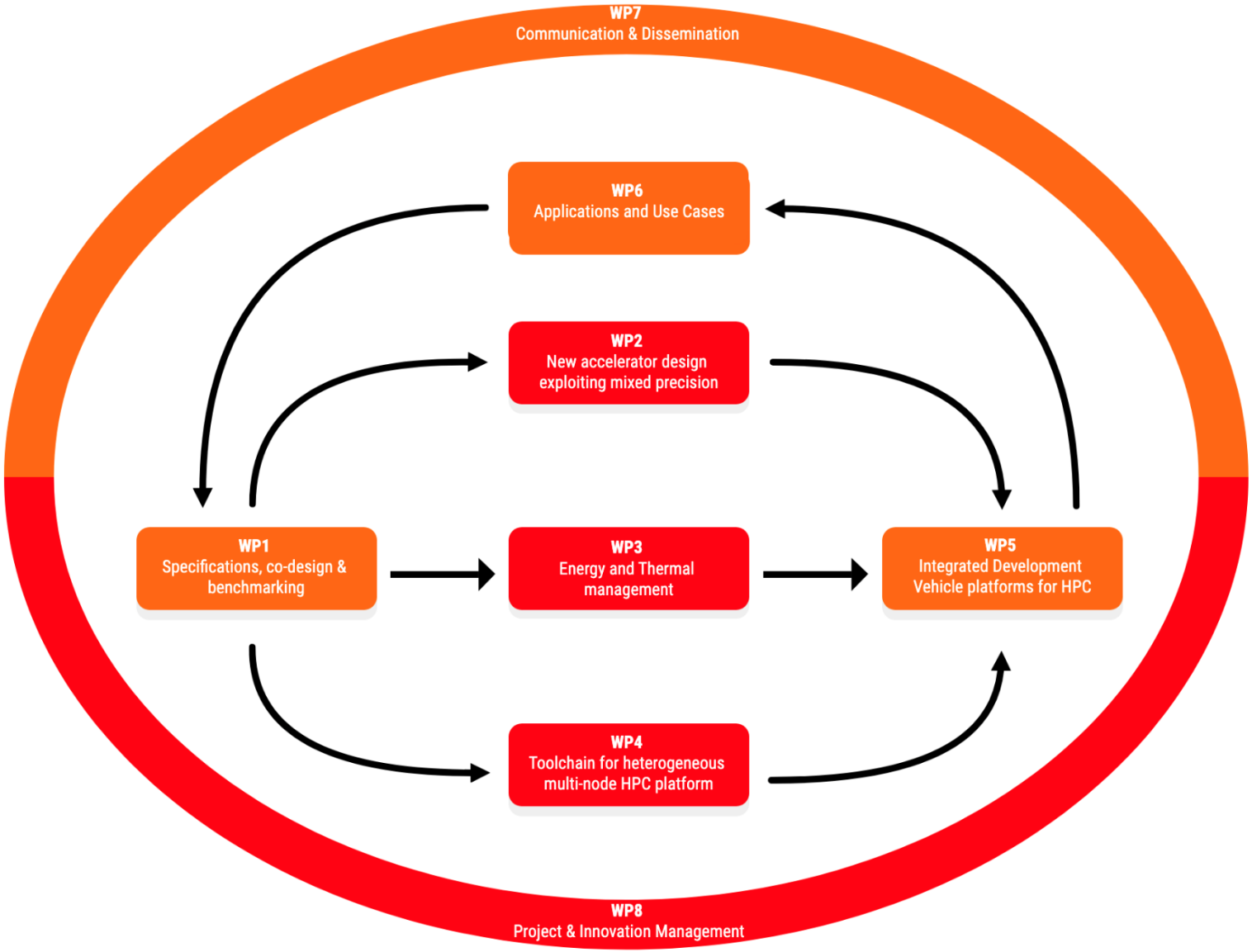
Smart Cities
Danger Detection
CINI

Smoke and fire detection in a smart city context, implemented through Convolutional Neural Networks on edge servers

Oil & Gas
Reverse Time Migration
FHG

High performance, energy efficient Reverse Time Migration for Oil & Gas and Geo-Services applications

TEXTAROSSA Work Package Structure



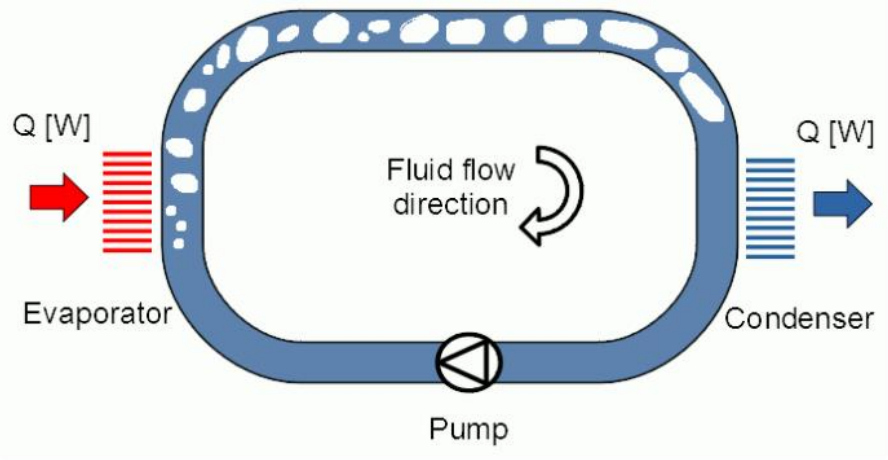
WP2 (HW IPs) Objectives

- Boosting the performance of HPC systems based on GPPs through FPGA/FPSoC acceleration
- Focus on emerging topics (**e.g. AI, Security, HPDA, …**)
- Complementary IPs to these developed in H2020 EPI SGA1
- Technology independent design to easy future migration in ultra-scaled technology
- Research-oriented activities
- RISC-V extensions
  1. **Accelerators with mixed-precision for AI computing and data compression**: design of IP for PPU (Posit Processing Unit) to be connected as co-processor to RISC-V by UNIPI and implemented in FPGA
  2. **eXtreme Secure Crypto IP**: hw acceleration for Post-Quantum Cryptography and Homomorphic Encryption.
- **IP for fast task scheduling** on FPGA
- **IPs for low-latency intra-node & inter-node communication on FPGA**

WP3 (Energy and Thermal Mgmt) Objectives

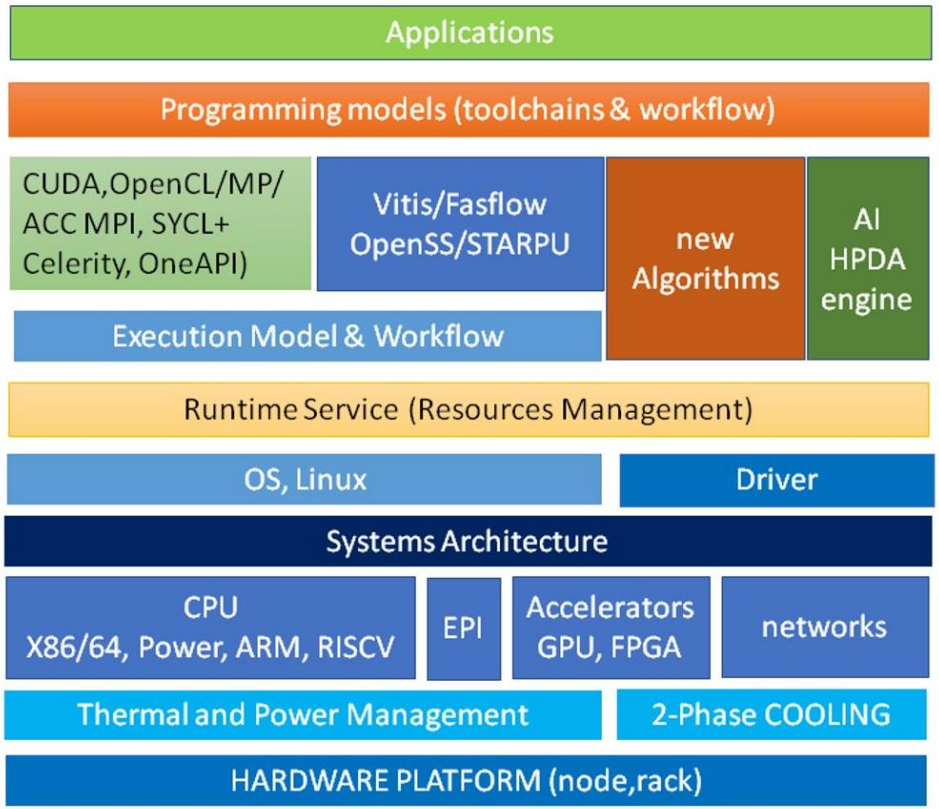
Integration of innovative thermal management solutions into the exascale computing systems: servers' cooling and thermal control strategies.

- An **innovative two-phase cooling** technology will be developed
- The two-phase cooling technology will be implemented at **server board (node) and at rack level.**
- Develop the **thermal management control strategies** to keep under control the thermal profile of the node/rack for energy, performance and reliability optimization.



WP4 (Toolchains) Objectives

- Programming frameworks at the node level
  - Stream-based: FastFlow.
  - Task-based: OmpSs, StarPU.
  - High-Level Synthesis: Vitis.
  - and ”standard” ones (CUDA, OneAPI, SYCL,…)
  - Leveraging hetereogeneous resources.
- Design/test a novel approach
  - Integration of different frameworks (e.g. Vitis + COMM API for stream-based FPGA programming)
- To implement low-level features and interfaces
  - Power modeling, mixed precision, **communication**
- Improve performance and energy efficiency



WP5 (IDV platforms for HPC) Objectives

- **WP5 has two main objectives:**

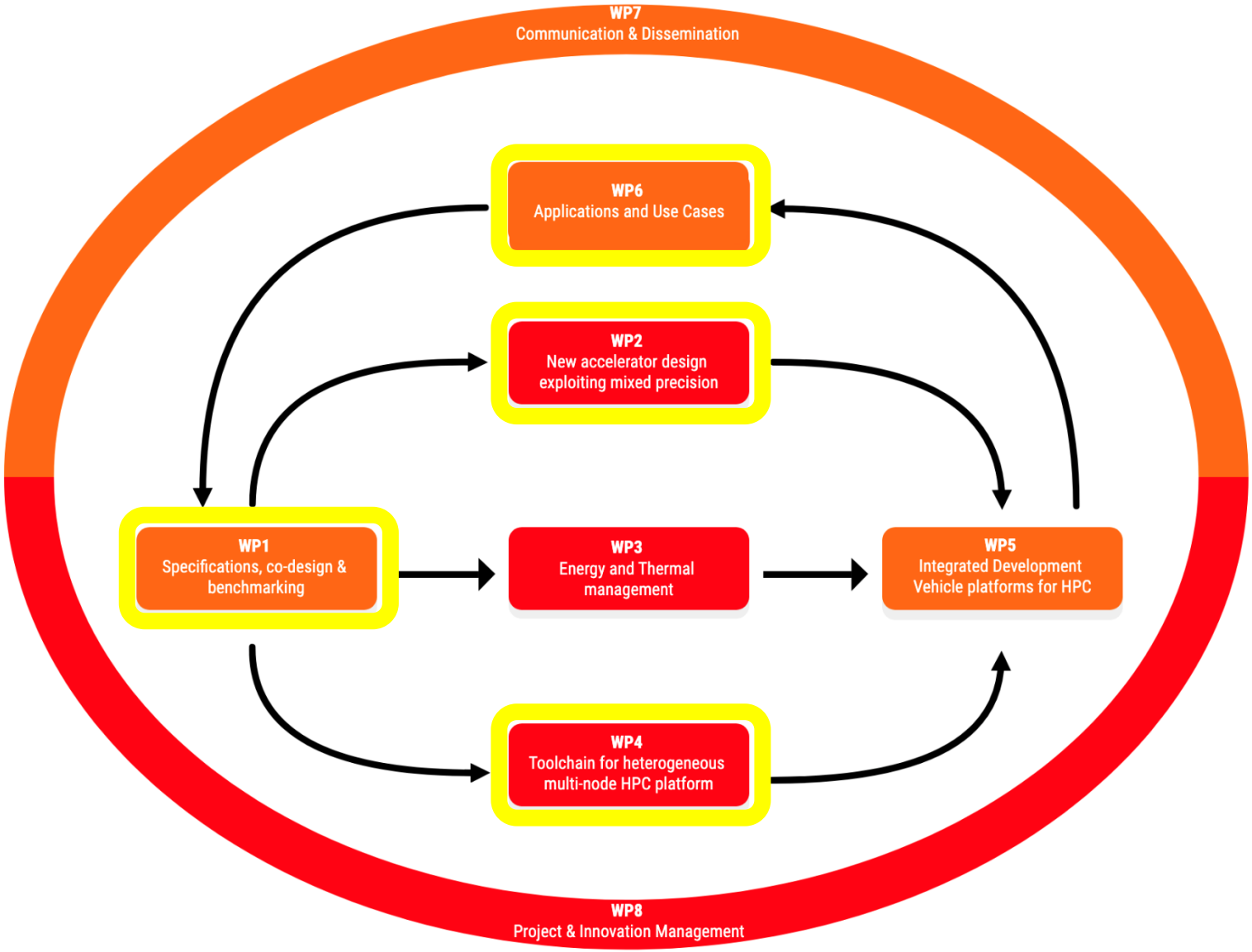
  - **Phase 1**: Adapt two commercial off-the-shelf nodes as per the specifications of WP1 and build two **Integrated Development Vehicles** (IDV-A developed by ATOS and IDV-E developed by E4) featuring the two-phase cooling technology as per the specifications of WP3.
    - Fixed specs for both node types:
      - **IDV-A (ATOS):** OpenSequana Blade, 2xINTEL Sapphire Rapids Processor + 4 NVIDIA Hopper GPUs.
      - **IDV-E (E4):** 2U Mt. Collins, 2xAmpere® Altra® Max Series Processor + Xilinx Alveo U280 FPGA.

  - **Phase 2**: Aggregate the prototype in a rack within a single cooling system according to the cooling components as defined in WP3. The placement of both types of nodes in a rack will follow a co-design approach according to the cooling architecture as defined in WP3.

WP6 (Applications and Use Cases) Objectives

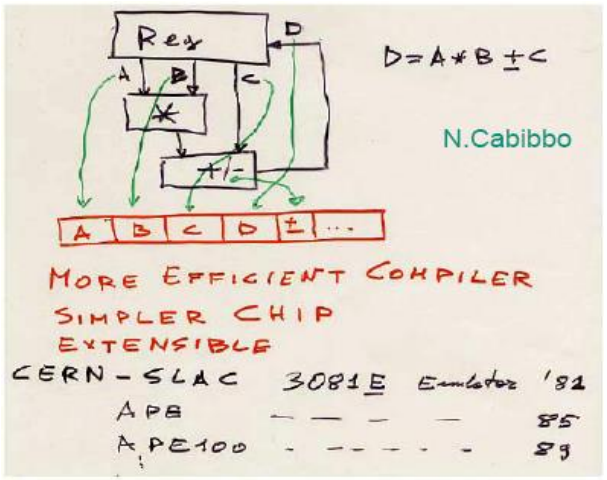
- Demonstration of developed TEXTAROSSA techniques for AI/HDPA/HPC real applications, steering of co-design
- Evaluation of TEXTAROSSA – Gap analysis (w.r.t. energy/performance improvement)
- Expected achievements
  - T6.1 benefits from heterogeneous resources
  - T6.2 benefits from compression / mixed-precision
  - T6.3 benefits from computation/communication orchestration
  - T6.4 techniques validation and impact measurement
- Key Performance Indicators
  - Energy-to-solution (FLOPS/W, FPS/W, SUPS/W, …)
  - Time-to-solution
  - Throughput

TEXTAROSSA INFN Work Packages



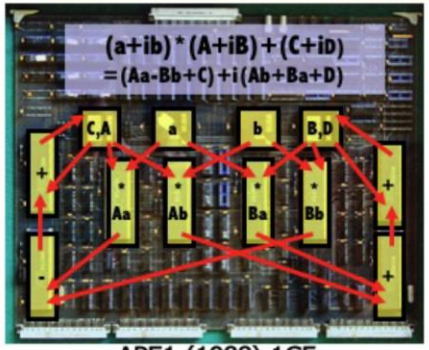
Background Technologies for TEXTAROSSA

APE Projects (1985-2004)

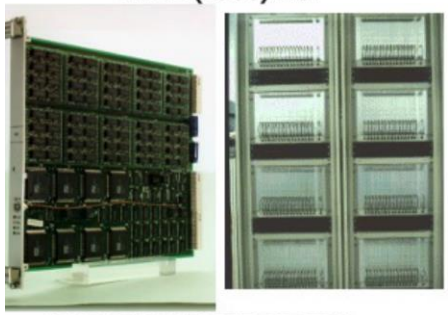


- Design and production of 4 generations of parallel machines optimized for LQCD and widely used across Europe.

- SIMD parallelism
- Scalability enabled by the 3D Torus Network
- Custom VLIW VLSI numerical processor
- DSL compilers and system software

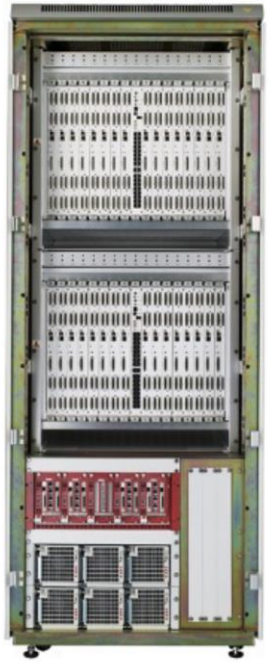


APE1 (1988) 1GF



APE100 (1992) 25GF, SP, REAL

APEmille (1999) 128GF, SP, Complex



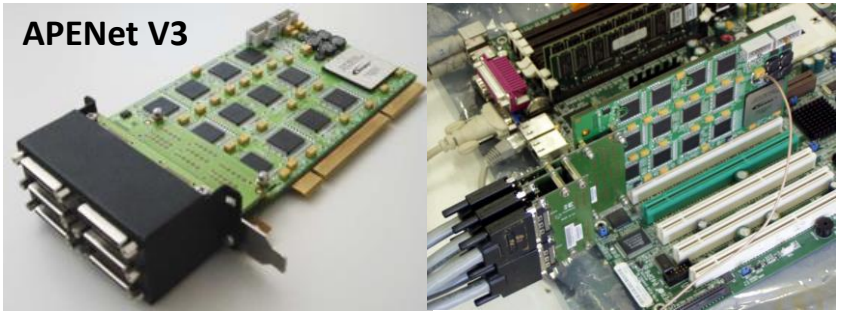
apeNEXT (2004) 800GF, DP, Complex

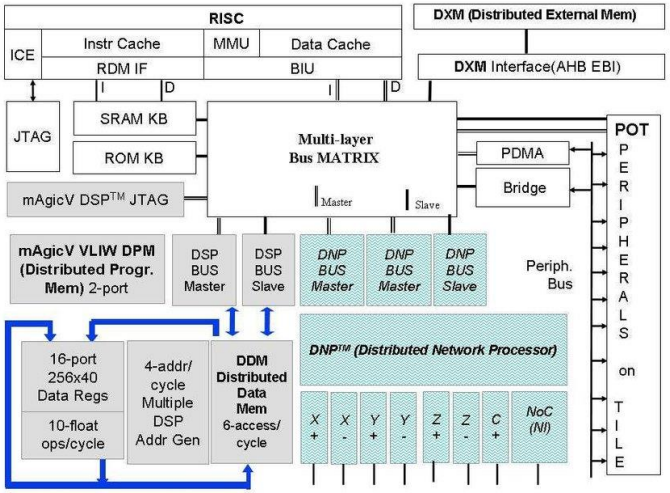
APEnet: 3D Torus Network for Hybrid Clusters

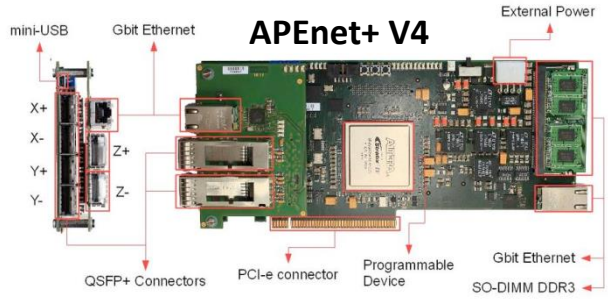
- FPGA-based NICs integrating switching and routing capabilities
- High throughput, low latency, lightweight network protocol
- PCI Interface, 6 Links full-bidir

APEnet history:

- **2003-2005: APEnet V3 (PCI-X), RDMA API**
- **2006-2009: APEnet goes embedded, AMBA-AHB integration with ARM9**
  - DNP, Distributed Network Processor
  - **EU SHAPES** project co-development
- **2012: APEnet V4 aka APEnet+ (PCIe Gen2)**
  - NVIDIA GPUDirect RDMA
  - **EU EURETILE** project co-development
- **2014: APEnet V5 (PCIe gen3)**





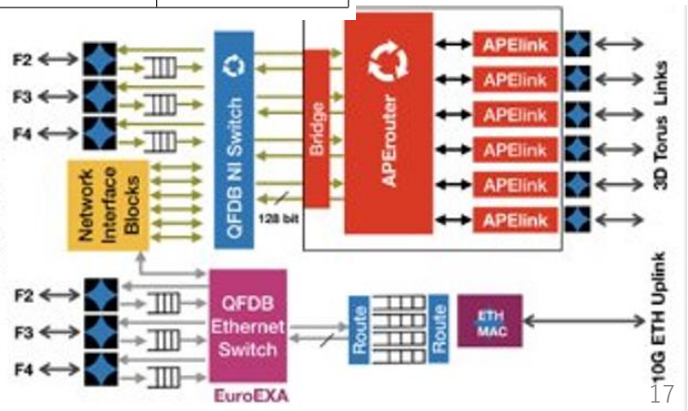


ExaNeST H2020 FET-HPC 2014 (2015-2019)

- **Evaluation of Exascale-enabling technologies**
  - Low-latency unified Interconnect (compute & storage traffic)
  - Fast, distributed in-node non-volatile-memory
- **Extreme compute-power density**
  - ARM-based (v8, 64 bit) microserver + FPGA accelerator
  - Advanced totally-liquid cooling technology
- **Real scientific and data-center applications: HW/SW Co-Design**
- **Prototype (July 19) : 768 ARM cores; 192 FPGAs; 3 TB of DDR3 memory**

	Hierarchy	Fanout	Switching	Topology	Bandwidth	Latency
Tier 4	System	500 Racks	Optical			
Tier 3	Rack	3 chassis	10GbE (ExaNet)	Fat-Tree (Torus)	10 Gbps	
Tier 2	Chassis	9 mezzanines	ExaNet	3D-Torus	4x10 Gbps	400 ns per hop
Tier 1	Mezzanine	4 nodes	ExaNet	Ring	2x10 Gbps	400 ns per hop
Tier 0	Node	4 FPGAs	ExaNet	All-to-All	16 Gbps	400 ns
FPGA	Unit	ZU9				
CORE		A53				

- **INFN contributions**
  - Architecture definition and system integration
  - Identification of the requirements and prototype benchmarking through spiking neural network code (DPSNN)

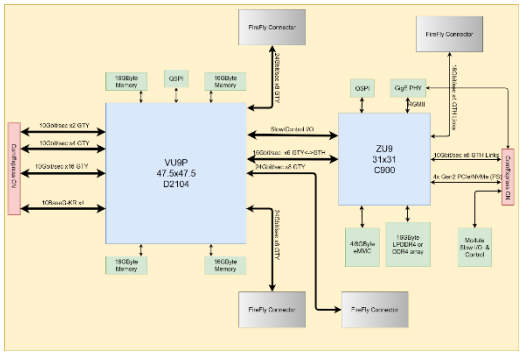


EuroEXA H2020 FET-HPC 2016 (2017-2021)

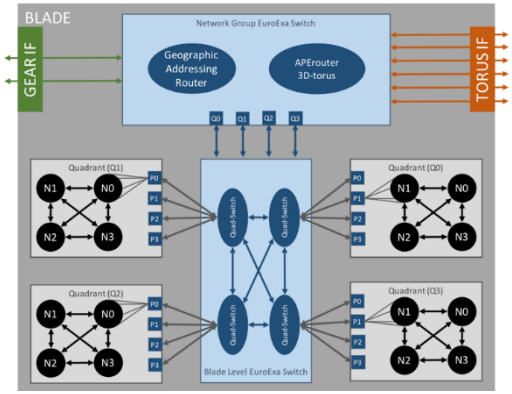
EuroEXA leverages on ExaNeSt, ExaNoDe and ECOSCALE results to deliver a world-class HPC pre-Exascale demonstrator

- **Energy Efficiency**
  - Tighter integration, customization and hardware acceleration
  - Advanced cooling
  - Mitigation of data transfer cost (memory compression, hyperconverged storage)
- **Scalability**
  - UNIMEM architecture
  - Unified (for data and storage traffic) low latency, high throughput, RDMA-based interconnect
  - **Hierarchical network topology**
- **INFN contributions**
  - **Network design at sub-rack level**
  - **Benchmarking through applications:**
    - Spiking neural network simulator (DPSNN)
    - Lattice Boltzmann Method (LBM)

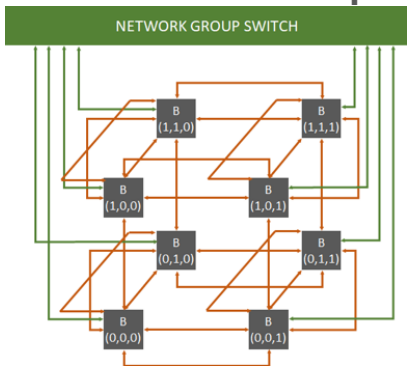
CRDB: Node



Blade



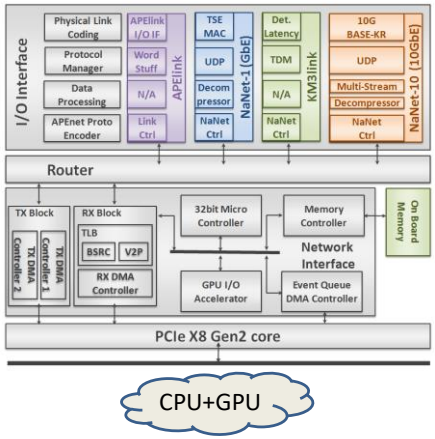
Network Group

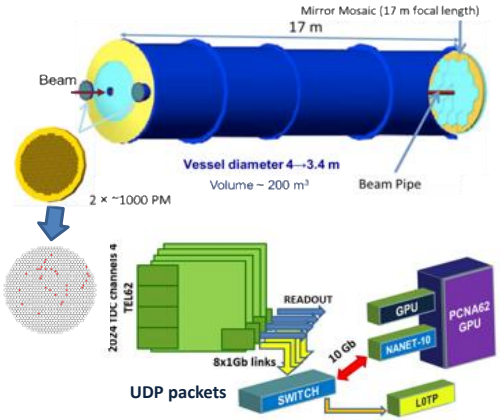


NaNet Project (2013-): Use of HPC Tech in HEP

NaNet: a family of FPGA-based PCIe Network Interface Cards

- Bridging the front-end electronics and the software trigger computing nodes.
- Supporting multiple link technologies and network protocols.
- Enabling a low and stable communication latency.
- Processing data streams from detectors on FPGA (data compression/decompression and re-formatting, coalescing of event fragments, …).
- Optimizing data transfers with GPU accelerators.
- **Real-time stream processing on heterogeneous devices.**





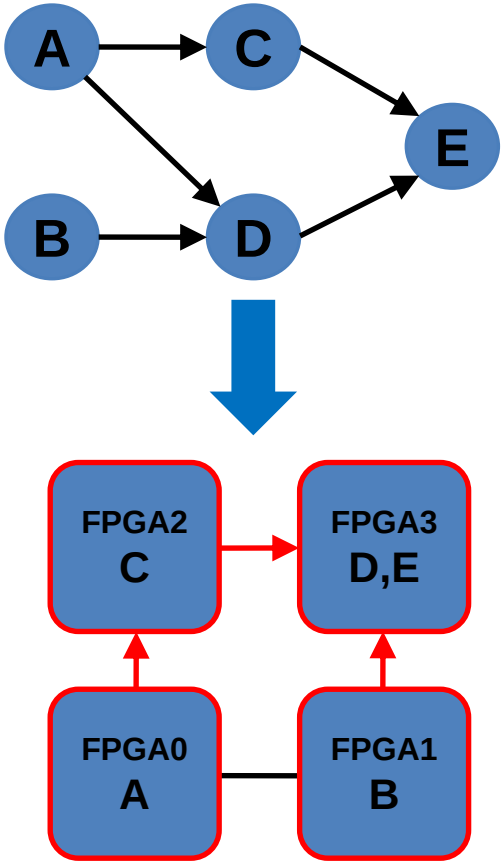
NA62 GPU-RICH @CERN

	NaNet-1	NaNet³	NaNet-10	NaNet-40
Year	Q3 - 2013	Q1 - 2015	Q2 - 2016	Q3 - 2019
Device Family	Altera Stratix IV	Altera Stratix V	Altera Stratix V	Altera Stratix V
Channel Technology	1 GbE	KM3link	10 GbE	40 GbE
Transmission Protocol	UDP	TDM	UDP	UDP
Number of Channel	1	4	4*	2
PCIe	Gen2 x8	Gen2 x8	Gen3 x8**	Gen3 x8
SoC	NO	NO	NO	NO
High Level Synthesis	NO	NO	NO	YES
nVIDIA GPUDirect RDMA	YES	YES	YES	YES
Real-time Processing	Decomp.	Decomp.	Decomp. Merger	?

HW IPs and Toolchains

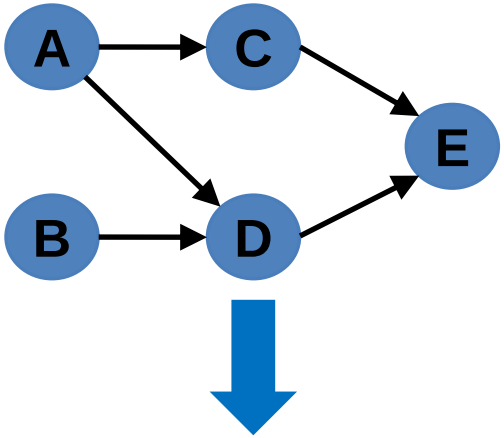
INFN Contribution to WP2/WP4: APEIRON

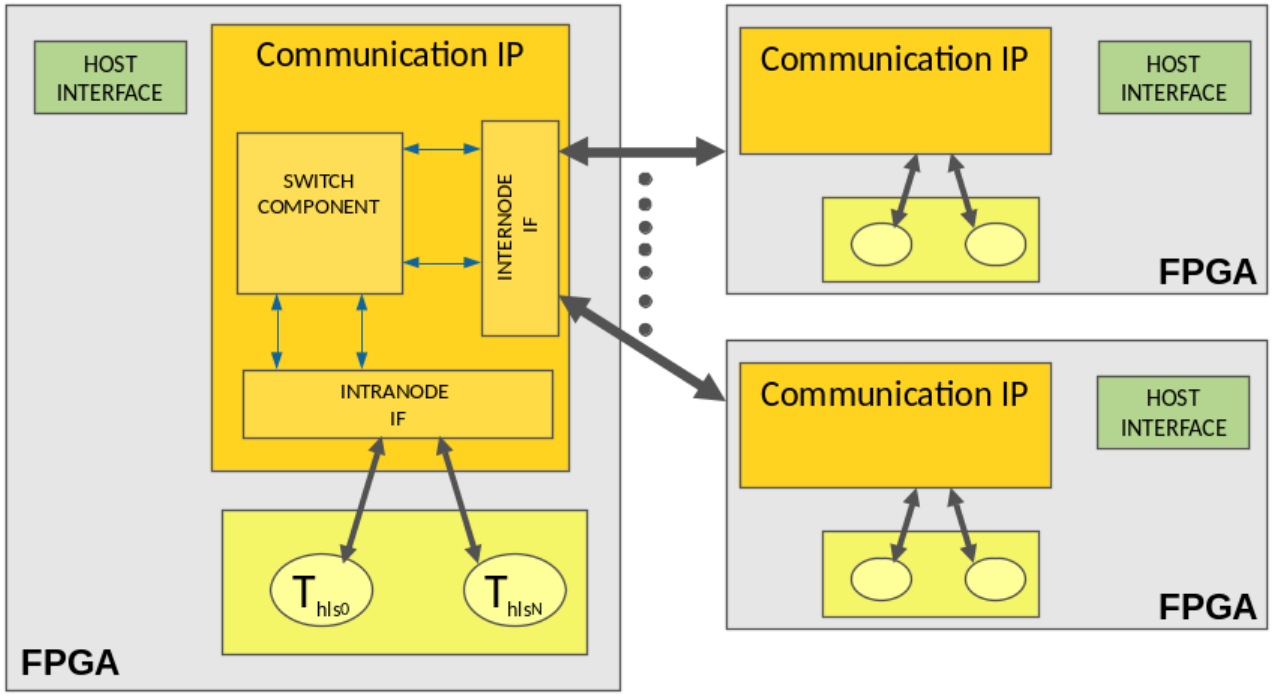
- Goal: offer hardware and software support for the execution on a system of **multiple interconnected FPGAs of applications developed according to a dataflow programming model**
- Map the directed graph of tasks on the distributed FPGA system and offer runtime support for the execution.
- Allow users with **no (or little) experience in hardware design tools** to develop their applications on such distributed FPGA-based platforms
  - Tasks are implemented **in C++ using High Level Synthesis tools (Vitis).**
  - Simple **Send/Receive** C++ communication API.



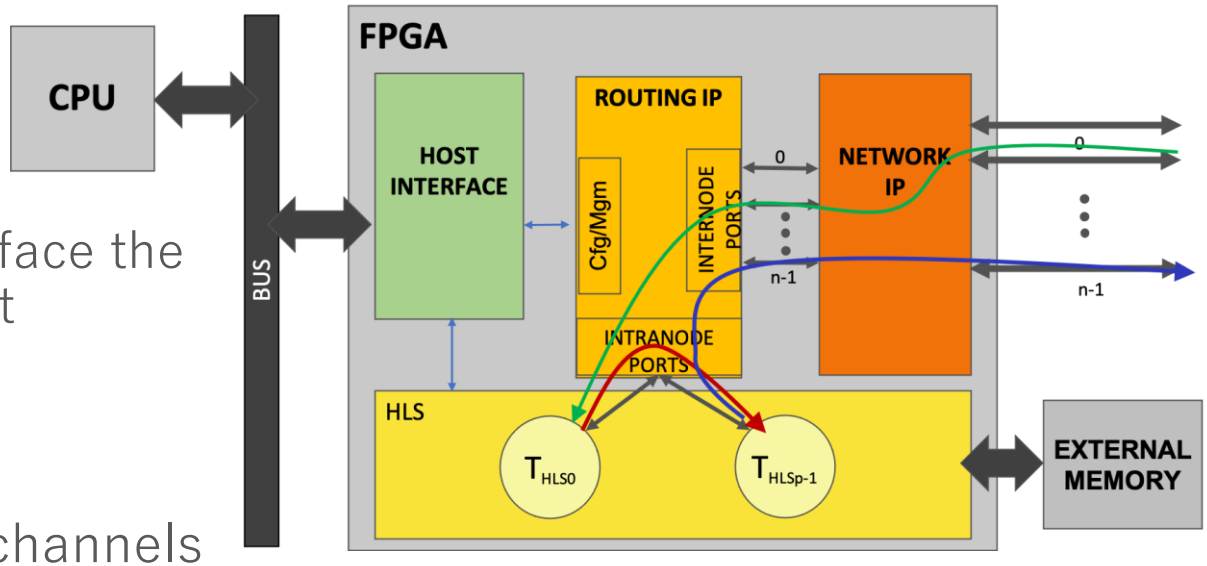
INFN Contribution to WP2

- INFN is developing the IPs implementing a **direct network** that allows **low-latency** data transfer between processing tasks deployed on the same FPGA (intra-node communication) and on different FPGAs (inter-node communication).





INFN in WP2: IPs for low-latency FPGA commun.



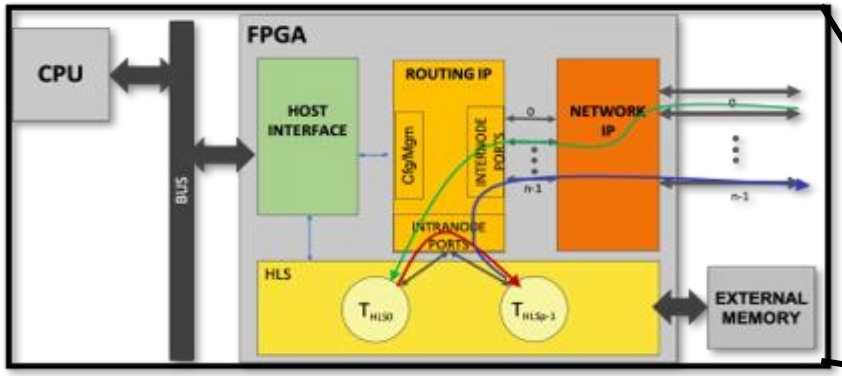
- **Host Interface IP:** Interface the FPGA logic with the host through the system bus.
  - PCI Express Gen3 → Gen4
- **Network IP:** Network channels and **Application-dependent I/O**
  - APElink 32 Gbps → 64/100 Gbps
  - UDP/IP over 10-25 GbE → 40/100 GbE
- **Routing IP**
  - Routing of intra-node and inter-node messages between processing tasks on FPGA.

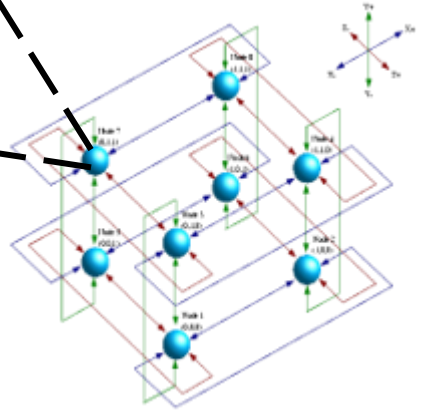
- Implemented as incremental development on APEnet IPs over XILINX platforms.
- **Deliverable D2.5**
  - Intermediate database at M18
  - Deployed in the IDVs (WP5) at M30

TEXTAROSSA: APEIRON – Workflow

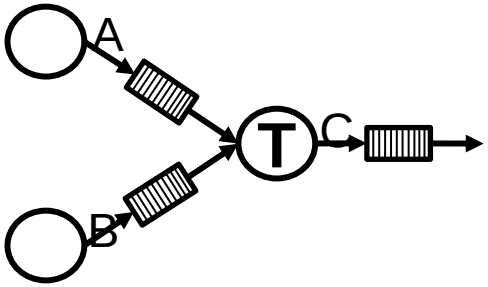
- For Xilinx FPGAs, APEIRON leverages the Vitis flow to generate the bitstream
- HLS kernels are written in C++, there are no particular restrictions, apart from the top-level interfaces (channels)
- A YAML configuration file is used to describe the kernels interconnection topology, specifying how many input/output channels they have

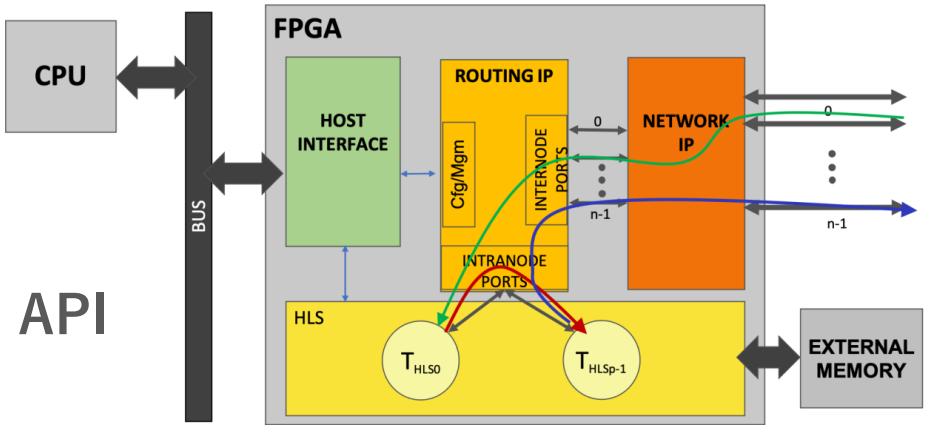
```yaml
kernels:
  - name: krnl_compute1
    input_channels: 4
    output_channels: 3
    switch_port: 1

  - name: krnl_compute2
    input_channels: 2
    output_channels: 1
    switch_port: 2

  - name: krnl_compute3
    input_channels: 1
    output_channels: 1
    switch_port: 3

  - name: krnl_compute4
    input_channels: 1
    output_channels: 1
    switch_port: 4
```





APEIRON HLS C++ Communication Primitives





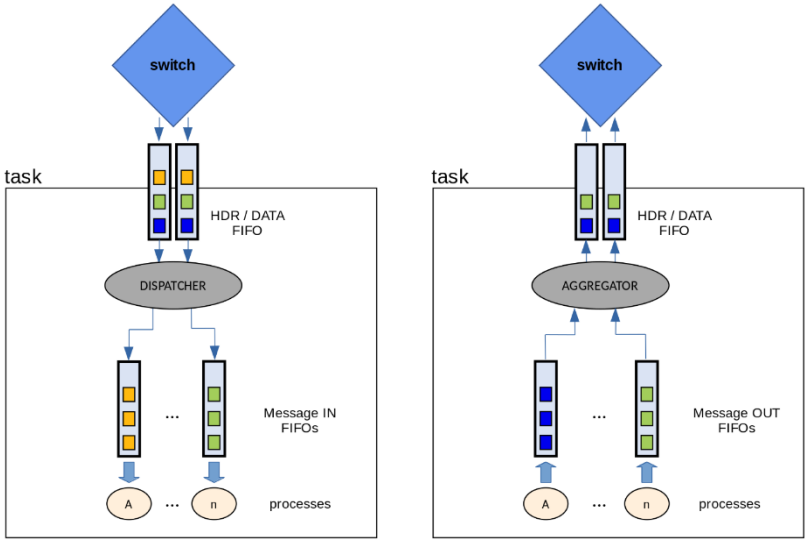
- **APEIRON C++ HLS Communication API**
  - **send**(msg, size, dest_node, task_id, ch_id)
  - **receive**(ch_id)

Where :

dest_node are the n-Dim coordinates of the destination node (FPGA) in a n-Dim torus network.

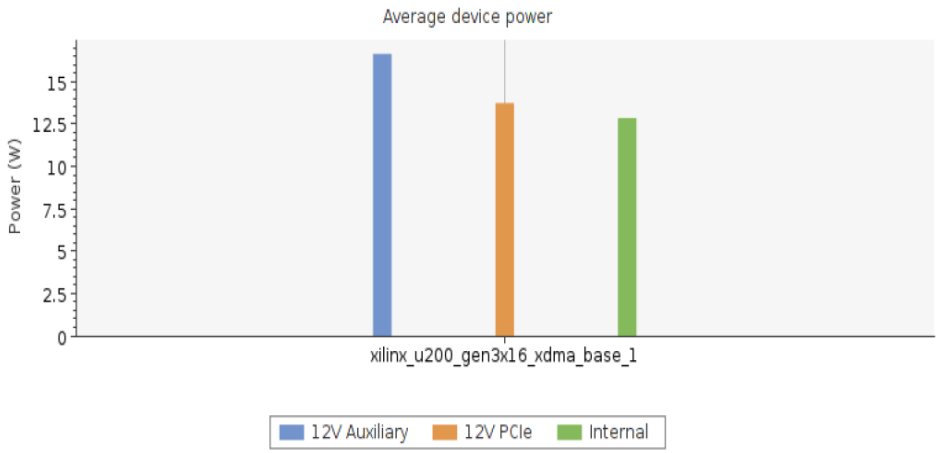
task_id is the local-to-node receiving task (kernel) identifier (0-3).

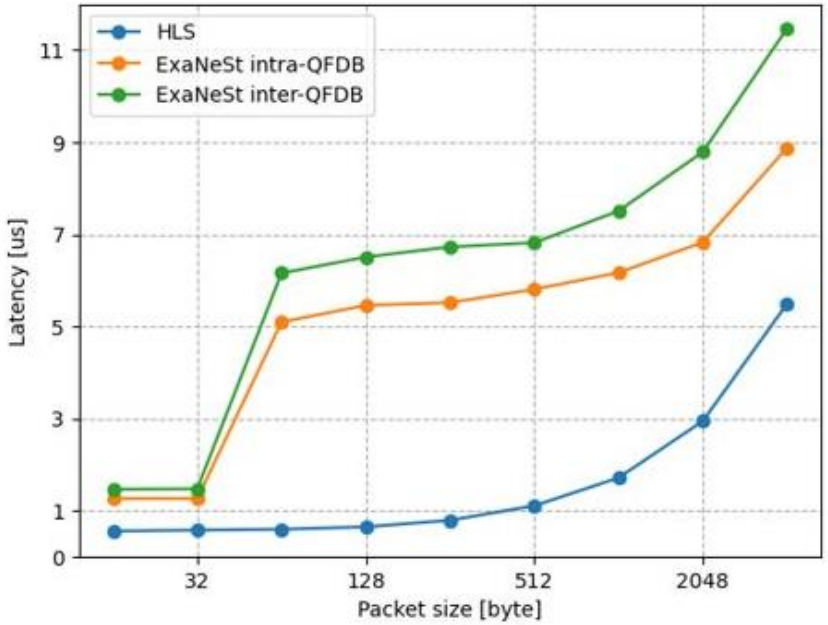
ch_id is the local-to-task receiving fifo (channel) identifier (0-127).

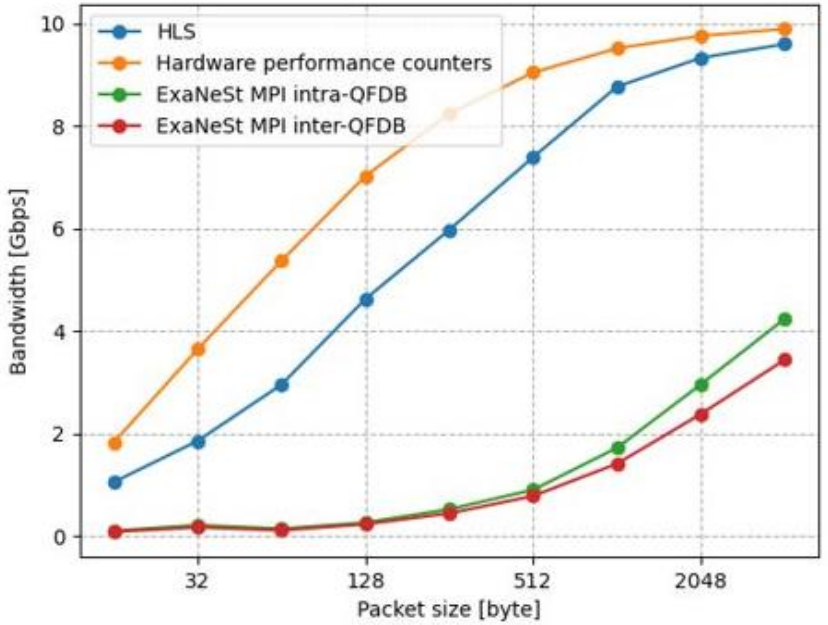


TEXTAROSSA: APEIRON Preliminary Results

- **No CPU and software stack overhead for communication**
  - Significantly reduced latency
  - Close to hardware limit BW
  - Reduced energy consumption (no CPU and system bus involvement), to be measured.







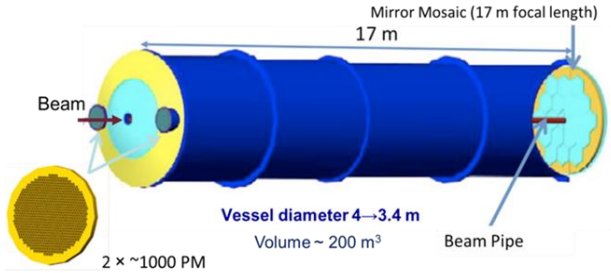
Applications & Use Cases

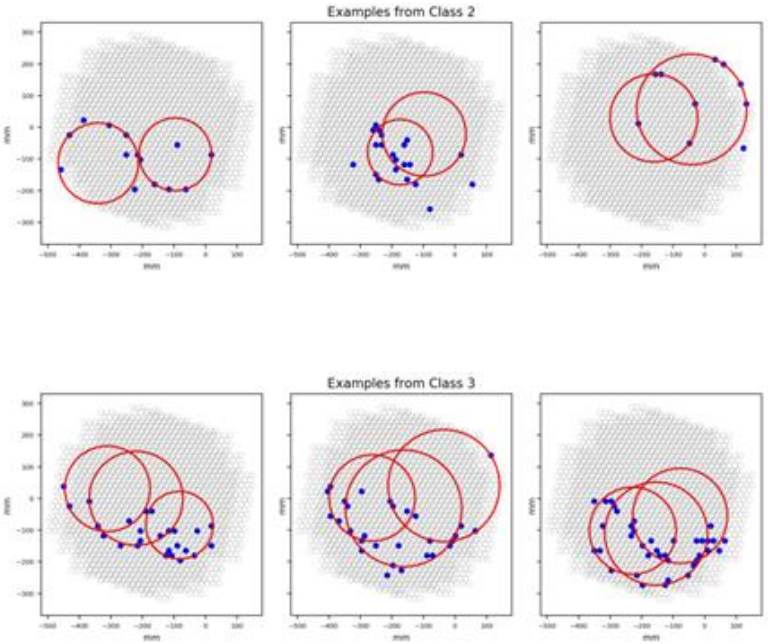
RAIDER

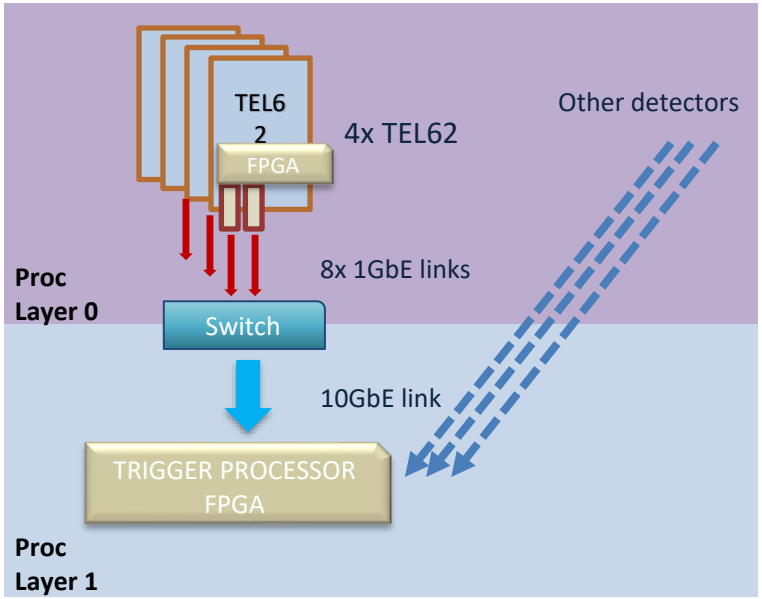
- **Real-time AI-based Data analytics on hetERogeneous distributed systems (dataflow processing).**
- **A Proof-of-Concept shall be delivered as a typical High Energy Physics experiment (CERN NA62) partial particle identification system with 10 Mevents/s througput and 1 ms latency requirements.**
- **Status**:
  - Studied and implemented on a single FPGA device several NN models (resource constraints, numerical representation, input format,...)
  - Started designing the distributed case (several interconnected FPGAs processing streams from different detectors)
- **Features to be exploited**:
  - Task 2.4: IPs for low-latency intra-node and inter-node communication
  - Task 4.1: Streaming Models
  - Task 4.4: Inter-FPGA Communication SW Stack
  - Task 4.5: Power modeling and use in runtime systems
- **is the prototype application already available**: single modules.
- **is the app or dataset ready for WP1 benchmarks**: single modules
- **what are the app-specific KPIs**
  - Processing pipeline Latency (us) and Throughput (Mevents/s)
  - Energy-to-solution: Mevents/Watt

RAIDER use case: Partial Particle ID with the NA62 RICH

- Located at the CERN SPS
- Measure rare kaon decay:
  - $k \to \pi\nu\nu$ with $BR(k \to \pi\nu\nu) = (8.4 \pm 1.0) \times 10^{-11}$
- Nominal event rate at L0: 10MHz
- Number of Cherenkov rings is a good candidate to improve L0 decisions: can we achieve required throughput with a good accuracy?



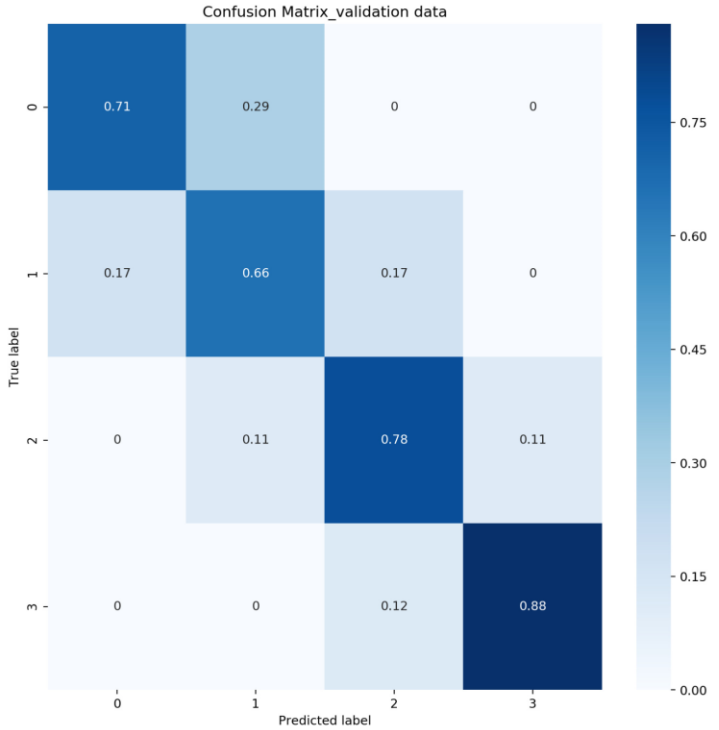


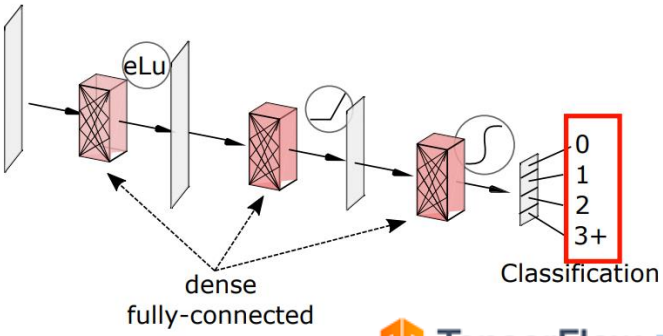


RAIDER Rings detection - Dense model on FPGA

Fully Connected

- Input: 64 hits per event
- Architecture: 3 fully connected layers
- Output: 4 classes (0, 1, 2, 3+ rings per event)
- Qkeras, quantization aware training:
  - ~75% average accuracy with low resource usage: LUT 14%, DSP 2%, BRAM 0% (VCU118)
- Latency: 22 cycles @ 150MHz
- Initiation Interval (II): 8 cycles



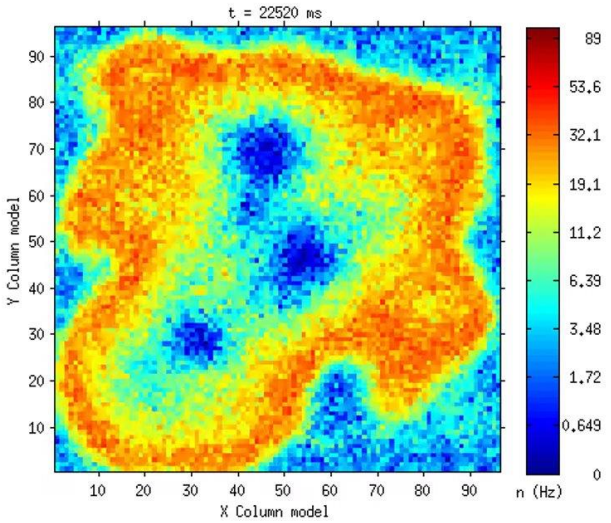


TensorFlow ➡ hls 4 ml ➡ VIVADO HLS ➡ IP integration in L0TP+ infrastructure

https://fastmachinelearning.org/hls4ml/

Nest GPU (as NEST on GPU)

- The engine driving the neural simulations is the Nest GPU code which is C++ with CUDA extensions and is production-ready
- The Python script detailing the experimental protocol is ready – a 1000ms simulation of dynamics of one hemisphere of cortex of mouse brain with a realistic connectome inferred from data obtained with optical imaging methods on anesthetized mice – and will be run by the Nest GPU engine on the reference platform.



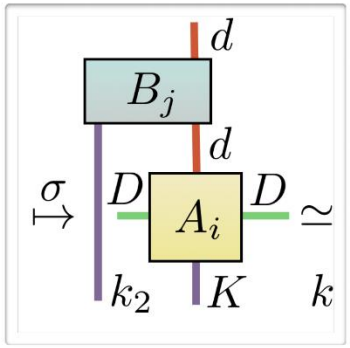
- As soon as the GPU-equipped is available, the simulation is ready to be benchmarked comparable with the same experiment on CPU-only engine (NEST).
- The specific KPI are:
  - Time-to-solution: Simulated-milliseconds-per-second
  - energy-to-solution: Synaptic UPdates per second (SUPs) per Watt

High Energy Physics high-level software tools

- For simulation, reconstruction (i.e. the transformation of detector signals to physics objects), data analysis
- Initial focus will be on the reconstruction software of the CMS experiment
  - Efforts are on-going to investigate parallelism and heterogeneous computing (CPU, GPU, possibly FPGA), based on TBB, CUDA, SYCL/OneAPI, Cupla/Alpaka, Vitis HLS, …
  - Some solutions are already in production, but investigation continues
- We have identified two software components, for particle tracking and calorimeter clustering
- Two directions of work
  - Use of GPUs and FPGAs via SYCL
  - Remote offloading of computation to specialized nodes
- Activity just started, due to delays in recruiting

Tensor Network Methods

TENSOR NETWORK ALGORITHMS



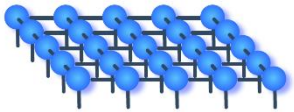
- *State of the art in 1D (poly effort)*
- *No sign problem*
- *Extended to open quantum systems*
- *Machine learning*
- *Data compression (BIG DATA)*
- *Extended to lattice gauge theories*
- *Simulations of low-entangled systems of hundreds qubits*
- *Extended to quantum field theories*

S. Montangero "Introduction to Tensor Network Methods", Springer (2019)

U. Schollwock, RMP (2005) *A. Cichocki, ECM (2013)* *I. Glasser, et al. PRX (2018)*

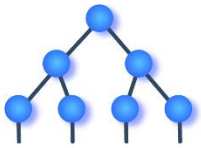
TENSOR NETWORKS STATES

$\psi_{\alpha_1,\alpha_2,\ldots\alpha_N}$ $\mathcal{O}(d^N)$



PEPS

$A^{\beta_1}_{\alpha_1} A^{\beta_1\beta_2}_{\alpha_2} \ldots A^{\beta_{N-1}}_{\alpha_N}$ $\mathcal{O}(Ndm^2)$



Tree Tensor Network

Tensor networks states are a compressed description of the system tunable between mean field and exact

Tensor network are state of the art methods for the simulation of many-body quantum systems, to understand complex quantum phenomena and to benchmark, verify and guide the developments of emerging quantum technologies (computers, simulations, sensors and communication).

Interpolation between mean field theory and exact description, faithful compression of the exponentially large many-body wave function.

INFN TEXTAROSSA Team

@INFN Roma – APE Lab


A. Lonardo


P. Vicini


F. Lo Cicero


F. Simula


M. Martinelli


P. S. Paolucci


R. Ammendola


A. Biagioni


P. Cretaro


O. Frezza


C. Rossi


M. Turisini

@INFN CNAF


F. Giacomini


L. Cappelli

@INFN Pisa


T. Boccali

@INFN Padova


S. Montangero