# Outlines in hardware and software for new generations of exascale interconnects

*Roberto* Ammendola[2], *Andrea* Biagioni[1], *Carlotta* Chiarini[1], *Paolo* Cretaro[1], *Ottorino* Frezza[1], *Francesca* Lo Cicero[1], *Alessandro* Lonardo[1], *Michele* Martinelli[1,*], *Elena* Pastorelli[1], *Pier Stanislao* Paolucci[1], *Luca* Pontisso[1], *Cristian* Rossi[1], *Francesco* Simula[1], and *Piero* Vicini[1]

[1]Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Roma, Rome, Italy
[2]Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Roma Tor Vergata, Rome, Italy

**Abstract.** RED-SEA (https://redsea-project.eu/) is a European project funded in the framework of the H2020-JTI-EuroHPC-2019-1 call that started in April 2021. The goal of the project is to evaluate the architectural design of the main elements of the interconnection networks for the next generation of HPC systems supporting hundreds of thousands of computing nodes enabling the Exascale for HPC, HPDA and AI applications while providing preliminary prototypes.
The main technological feature is the BXI network, originally designed and produced by ATOS (France). The plan is to integrate in the next release of the network – BXI3 – the architectural solutions and novel IPs developed within the framework of the RED-SEA project.
The consortium is composed of 11 well-established research teams across Europe, with extensive experience in interconnects, including network design, deployment and evaluation.
Within RED-SEA, INFN is adopting a hardware/software co-design approach to design APEnetX, a scalable interconnect prototyped on latest generation Xilinx FPGAs, adding innovative components for the improvement of the performance and resiliency of the interconnect. APEnetX is an FPGA-based, PCIe Gen3/4 network interface card equipped with RDMA capabilities being the endpoint of a direct multidimensional toroidal network and suitable for integration in the BXI environment. APEnetX design will be benchmarked on project testbeds using real scientific applications like NEST, a spiking neural network simulator.

## 1 Introduction

Refined accuracy, shorter time to solution and management of the exponential growth of data are the main requirements for simulations of complex phenomena — e.g. multi-physics, multiple phases heterogeneous workflows — on Exascale systems in both scientific and industrial fields of the coming decade. Efficient networks supporting massively parallel processing systems (hundreds of thousands of nodes, millions of cores) will be one of the pillars in the next generation of Exascale deployments. To allow for emerging data-centric and AI-related

*e-mail: michele.martinelli@roma1.infn.it

applications to scale efficiently at Exascale level and beyond, novel interconnects must support power-efficient accelerators and compute units, just to mention two of the many features required.

The RED-SEA project pursues extreme scale computing and data driven technologies, leveraging key European competences and backgrounds that include BXI as well as output from a number of EU-funded projects on interconnects and HPC that strive to anticipate the requirements of systems in the 2022-2025 timeframe. A further key target of the project is to create and uphold European Intellectual Properties (IP), consolidating European technological partnerships around BXI and the interconnection topic while fostering the growth of the related hardware/software ecosystems and community. RED-SEA is reaching out to ARM and RISC-V architectures to jumpstart EPI-related IPs for new interconnects but is also designing effective network interfaces for varied workloads in order to create a programmable platform that is highly heterogeneous and can make use of state-of-the-art interconnect technologies. RED-SEA proposes different network interface cards for different workload scenarios. Each NIC is an end-point of the BXI environment, having Network interfaces specialized for the target application and offering the BXI link as I/O pathway.

The INFN duty in this environment is two-fold: (i) to design the APEnetX NIC — based on a PCIe interface that communicates with either CPU and GPU accelerators — and implement a prototype exploiting Xilinx Alveo FPGAs, (ii) to benchmark the reference NEST application on the prototype in support of the co-design paradigm for such activity. In summary the INFN target in RED-SEA is to develop network IPs that are optimized for the kind of network traffic generated by spiking neural network applications.

In this paper the latest generation of the APEnet Network Interface Card based on FPGA is presented (section 3) and preliminary latency and bandwidth results are reported (section 4).

### 1.1 Related work and previous projects achievements

The Interconnect market landscape has no pure players following the acquisition of Mellanox by Nvidia in 2019. The proprietary Slingshot [1] interconnect will be embedded in the HPE/Cray future system and Intel removed Omni-Path [2] from its roadmap. "Home-grown" technology for both the processor and the interconnect is planned for the US, Japanese and Chinese Exascale environments. Thus, Atos BXI [3] remains the only HPC interconnect independent from the compute solution available to European HPC solution providers.

During the last years there were several attempts in the research field to implement high-speed networks targeting HPC computing onto FPGA-based platforms. The open source Corundum [4] platform encompasses a high-performance datapath, 10G/25G/100G Ethernet MACs, PCI Express Gen3 and custom PCIe DMA engines and extensible queue management supporting over 10000 queues coupled with a transmit scheduler and fine-grained hardware control of packet transmission. Limago [5] provides an open source implementation of a TCP/IP stack operating at 100Gbps and a fully customizable NIC. EasyNet [6] integrates an open source 100Gbps TCP/IP stack into a state-of-the-art FPGA development framework (Xilinx Vitis) providing MPI-like communication primitives for both point-to-point and collective operations as components in a High Level Synthesis (HLS) library.

The INFN APE Lab has a consolidated experience in the HPC interconnects. APEnet+ project [7] integrates a 3D Torus low latency interconnect equipped with an RDMA engine and a specialised HW/SW interface to NVIDIA GPUs implementing a "Direct-GPU" protocol [8] to avoid multiple hops inside the host. Within the framework of the ExaNeSt [9] project and in collaboration with FORTH, a hierarchical network interconnect based on FPGAs — ExaNet — was designed and deployed on a small/medium testbed of 128 Zynq

Ultrascale+ SoCs arranged into an all-to-all topology at the node level (a node is made of 4 interconnected FPGAs) and a scalable 3D Torus network for inter-node interconnect at the rack level. EuroEXA [10] leverages on ExaNet to push the concept of "hybrid topology scalable interconnect" (*Trifecta*™) at extreme scale. INFN designed an innovative "Custom Switch" based on a single FPGA Virtex Ultrascale+ implementing 2-hops all-to-all topology at board level, a 3D Torus network at rack level and a ExaNet-Ethernet 100/200G bridge for inter-rack connectivity.

## 2 RED-SEA project

The RED-SEA [11] project faces the challenge of preparing a next generation European Interconnect, capable of powering the EU Exascale systems to come. The project leverages on European interconnect technology (BXI) associated with standard and mature technology (Ethernet), previous EU-funded initiatives such as ExaNeSt [12], EuroEXA, ECOSCALE, Mont-Blanc [13], DEEP projects [14] and the European Processor Initiative (EPI), as well as open standards and compatible APIs.

RED-SEA explores innovative solutions on every fundamental facet of the network scenario with scalability (target size is around 100k nodes) being a major concern. The goal is developing a reliable end-to-end protocol for the transport layer and embedding it in an MPI-compliant communication library in order to optimize the performance and to meet the requirement of AI and data-centric applications. The bandwidth increase of serial links is one of the requirements of future interconnect technologies. The project targets a 200Gbps-per-direction link and innovative solutions to reduce the latency between emergent low power processor architectures (ARM and RISC-V) and the network, via the development of low latency interfaces equipped with multiple RDMA engines in such a way as to be coherent and guarantee compatibility with COTS HPC components. RED-SEA will propose and evaluate new QoS-provision mechanisms and congestion management solutions suitable for data-centric HPC environments. The hardware is empowered to take key decisions as soon as possible in order to increase the efficiency of the proposed techniques while mechanisms are designed and implemented immediately in the NIC or in the network switches. The integration of Internet Protocol, Ethernet and RoCE (RDMA over Converged Ethernet) traffic over an HPC interconnect is addressed. The project focuses on the development of MAC and PCS modular IPs supporting Ethernet and BXI traffic. An Ethernet gateway prototype connecting the HPC fabric to the Ethernet storage network is implemented in the framework of the project.

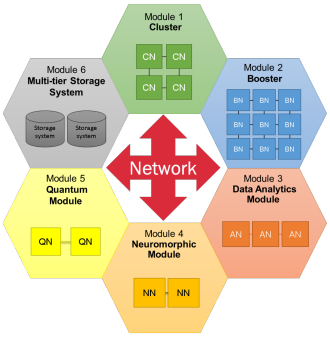


**Figure 1.** Schematic representation of the Modular Supercomputing Architecture.

### 2.1 Technological challenges

The optimization of the application workflows, leading to a major improvement of the production efficiency, has driven the definition of new computing systems in past years. The convergence of High Performance Computing (HPC), High-Performance Data Analytics (HPDA) and Artificial Intelligence (AI) workloads results in a Modular Supercomputing Architecture (see Figure 1). The Modular Supercomputing Architecture is an aggregation of resources that are organized to facilitate the mapping of several workflows exploiting a network federating specialized clusters. Each module is therefore a cluster providing specialized solutions to whatever the different workload may be — data analytics, neuromorphic, quantum computing and, of course, traffic towards the storage system. The Modular Supercomputing Architecture performance mainly relies on network quality to guarantee the efficient flow of data across the continuum of computing tasks.

In this landscape, the network is likely to become the next big bottleneck just as memory is in single node systems. The RED-SEA consortium proposes to address these challenges using (see Figure 2):

- High performance Ethernet as a federation network featuring state-of-the-art low latency RDMA communication semantics;
- BXI as the HPC fabric consisting of two discrete components, a BXI NIC plus a BXI switch, and the BXI fabric manager. BXI third generation will add new features and boost its performance to match the listed objectives.

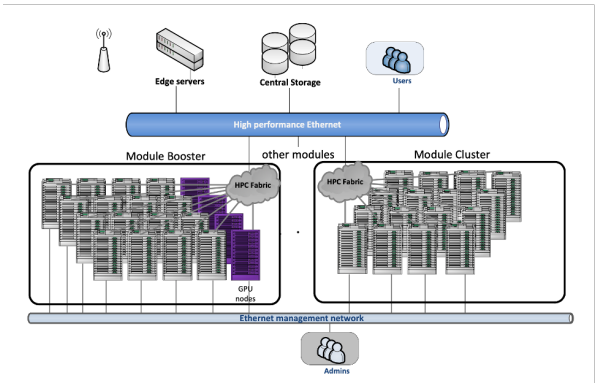


**Figure 2.** Network architecture of a Modular Supercomputing Architecture.

## 3 APEnetX board

INFN APEnet project is a long standing initiative aiming at designing and developing an FPGA-based, point-to-point, low-latency and high-throughput interconnect adapter to be employed in High Performance Computing clusters with a 3D toroidal network mesh. Several designs were deployed in the last 10+ years, each one characterised by the adoption of different PCI technologies, performance improvements at link level, routing capabilities, integration of dedicated engines for low latency data transport to/from computing accelerators (NVIDIA GPU). APEnetX, described in the following paragraphs, is the latest generation of APEnet interconnect architecture.

### 3.1 APEnetX HW architecture

INFN developed APEnetX, a low-latency and high-throughput NIC based on a PCIe Gen3/Gen4 interface to increase the capability of the network and to keep compatibility with off-the-shelf clusters.

The APEnetX prototype is developed on a Xilinx Alveo U200 board, which is built on the Xilinx 16nm UltraScale architecture and natively supports the QDMA IP. PCIe support is developed using QDMA IP by Xilinx, on the leftmost part of the figure 3: in green the routing and switching IP and the transmission control logic developed mainly during past European projects and optimized to meet the requirements of the RED-SEA project; the proposed Network Interface (APEni) supporting the RDMA semantics is in blue, which manages the user-level, zero-copy RDMA data transmission to/from memory, and offloads the host operating system. In the current implementation, the virtual memory address resolution is managed through the IOMMU of the Intel processor for the implementation of the direct I/O.

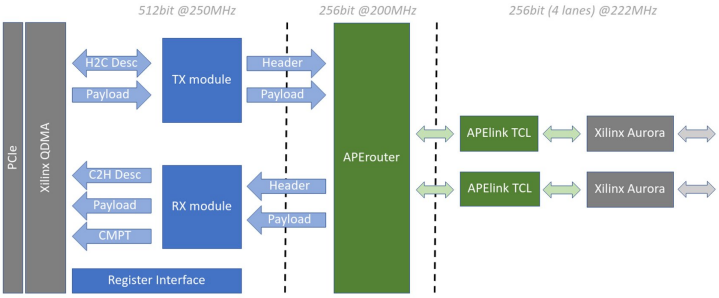


**Figure 3.** APEnetX architecture, showing the different blocks: PCI host interface, Router, Links

We have designed and developed the modules that manage the data flow through the PCIe interface, supporting the communication protocol of the Xilinx QDMA IP and encapsulating the data in a proprietary format, the APEnet protocol. Each packet is composed by a 32byte header and footer, plus payload with a maximum size of 4KB. The Queue Direct Memory Access (QDMA) subsystem is a PCI Express-based DMA engine optimized for both high bandwidth and high packet count data transfers. It is composed of the UltraScale+™ Integrated Block for PCI Express and an extensive DMA and bridge infrastructure that enables the ultimate in performance and flexibility.

### 3.2 Software stack

Xilinx provides an open source device driver for the QDMA IP to be used as hardware validation and demonstrator. A user-space software stack to highlight the performances that the design can hit is also included. We slightly modified the Xilinx software, adding custom features and trying to reuse the available code as much as possible to avoid the loss of compatibility with future versions:

- Linux device driver: instead of using the already taken READ and WRITE syscalls, we use IOCTL as the main entry point for our custom software. In particular we use it to register and deregister buffers as needed by the RDMA semantics.

- User-space library: the interface between any user-space application and the kernel-space Linux device driver is the custom LibQCM dynamic library. This component is used mainly to prepare the data struct needed by the IOCTL device driver syscall.
- No regression tests: we implemented a series of bare-metal tests to validate the software stack and the new hardware syntheses during the debug and development phase.

A send request by the user application is wrapped in a "transmission request" by the LibQCM library and then passed to the Linux device driver which forges a "tx descriptor" data structure that contains all the relevant information for the send phase (i.e. destination host coordinates, destination virtual address, packet size, etc.). On the receiving side, completions are the mechanism used by the hardware to notify the user about the presence of newly arrived data. As soon as received data has been DMA-written in memory, an "RX done" *completion event* is generated, DMA-written in memory and notified to the device driver. The user application can poll on the completion memory address to get informed about the presence of newly arrived data.

### 3.3 User-space completions optimization

The proposed completion mechanism requires the copy of the completion content between the Linux kernel memory address space and the user-space memory address space.

A first optimization is to have a different completion address for every receiving buffer. The user-space software checks only a single memory area, allocated in the memory address space of the user process and then remapped in the kernel memory address space, where the kernel copies the new completion. The steps can be summarized as follow:

- User allocates a completion buffer: this is the memory area where the user application expects to receive the confirmation of data reception;
- the Linux device driver module remaps this memory area in kernel space: the completion buffer becomes shared between user-space and kernel-space. After the remap, all changes in the memory region will be reflected on both mappings;
- at the reception, user data is (asynchronously) DMA-copied in memory by the hardware and then a completion is issued to inform the software about the presence of new data;
- the Linux device driver module copies the completion data to the (remapped) user address;
- a bit in the completion is used as "ready" signal from the kernel-space module to user-space application.

A better implementation, in terms of latency and efficiency, is to exchange the completion buffer virtual address during the initial handshake, so the source node knows where the completion must be written at transmission time and this information is then included in the "tx descriptor" (see 3.2) and propagated along with the packet over the network so the receiver node can DMA-write the completion data directly at the user-space completion virtual address without the device driver intervention.

## 4 Results

To validate the hardware design and the software stack, we used a testbed composed of two Supermicro SuperWorkstation 7049GP-TRT, $2 \times 8$-cores 4200-series 14nm Intel Xeon Scalable Silver Processors (Cascade Lake) running at 2.10GHz. The operating system was a GNU/Linux Centos 8 with a 4.18.0 kernel. Two different APEnetX boards were connected together point-to-point through custom "apelink" protocol with QSFP+ cables.

We implemented a synthetic test to show the improvement of the optimized version of the software presented in Section 3.3: a simple ping-pong test repeated many times, then averaging their runtimes and plotting them against the message size. As can be seen in Figure 4, the naive implementation latency is around 10$\mu$s while with the optimized version the latency drops to around 6$\mu$s.

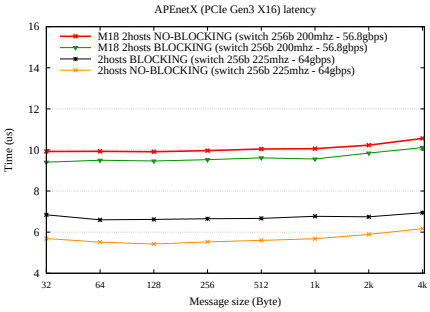


**Figure 4.** Latency results using the naive completions implementation and the optimized version. The "blocking" version waits for the end of transmission before issuing the next, while the "non-blocking" returns immediately to the user.

The total bandwidth achieved in this early phase of the development is plotted in Figure 5. The test is a simple send of a number of packets up to a specific size, with the total time measured and used to obtain the total bandwidth.

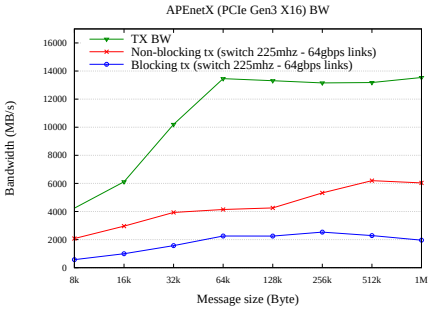


**Figure 5.** Preliminary bandwidth achieved with the current design. The TX BW is the maximum total bandwidth achievable, the plot is obtained by flushing the packets in the transmission phase. In the "non-blocking" version the last send is used as "flush", assuring the previous packets were sent. In the blocking version each send is awaited individually before proceeding.

In conclusion, we designed the APEnetX NIC based on a PCIe interface to communicate with both CPU and GPU accelerators in a RDMA fashion on Xilinx Alveo FPGA. Benchmark results showed that some custom components (e.g. our *custom switch*) are limiting the maximum performance achievable on the PCIe interface. In the next future, effort will be put to speed up and optimize the developed IPs.

## Acknowledgment

This work is supported by the EuroHPC JU initiative under specific grant agreements No. 955776 (RED-SEA) and No. 956831 (TEXTAROSSA). The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Greece, Germany, Spain, Italy, Switzerland.

## References

[1] D. De Sensi, S. Di Girolamo, K.H. McMahon, D. Roweth, T. Hoefler, *An In-Depth Analysis of the Slingshot Interconnect*, in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis* (2020), pp. 1–14

[2] M.S. Birrittella, M. Debbage, R. Huggahalli, J. Kunz, T. Lovett, T. Rimmer, K.D. Underwood, R.C. Zak, *Intel® Omni-path Architecture: Enabling Scalable, High Performance Fabrics*, in *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects* (2015), pp. 1–9

[3] S. Derradji, T. Palfer-Sollier, J.P. Panziera, A. Poudes, F.W. Atos, *The BXI interconnect architecture*, in *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects* (IEEE, 2015), pp. 18–25

[4] A. Forencich, A.C. Snoeren, G. Porter, G. Papen, *Corundum: An Open-Source 100-Gbps Nic*, in *2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)* (2020), pp. 38–46

[5] M. Ruiz, D. Sidler, G. Sutter, G. Alonso, S. López-Buedo, *Limago: An FPGA-Based Open-Source 100 GbE TCP/IP Stack*, in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)* (2019), pp. 286–292

[6] Z. He, D. Korolija, G. Alonso, *EasyNet: 100 Gbps Network for HLS*, in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)* (2021), pp. 197–203

[7] R. Ammendola, M. Bernaschi, A. Biagioni, M. Bisson, M. Fatica, O. Frezza, F. Lo Cicero, A. Lonardo, E. Mastrostefano, P.S. Paolucci et al., *GPU Peer-to-Peer Techniques Applied to a Cluster Interconnect*, in *2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum* (2013), pp. 806–815

[8] *Nvidia gpudirect: Enhancing data movement and access for gpus*, `https://developer.nvidia.com/gpudirect`

[9] M. Katevenis et al., Microprocessors and Microsystems **61**, 58 (2018)

[10] Biagioni, Andrea et al., EPJ Web Conf. **245**, 09004 (2020)

[11] A. Biagioni et al., *RED-SEA: Network Solution for Exascale Architectures*, in *2022 25th Euromicro Conference on Digital System Design (DSD)* (2022), pp. 712–719

[12] R. Ammendola, A. Biagioni, P. Cretaro, O. Frezza, F.L. Cicero, A. Lonardo, M. Martinelli, P.S. Paolucci, E. Pastorelli, F. Simula et al., *The Next Generation of Exascale-Class Systems: The ExaNeSt Project*, in *2017 Euromicro Conference on Digital System Design (DSD)* (2017), pp. 510–515

[13] A. Armejach, B. Brank, J. Cortina, F. Dolique, T. Hayes, N. Ho, P.A. Lagadec, R. Lemaire, G. López-Paradís, L. Marliac et al., *Mont-Blanc 2020: Towards Scalable and Power Efficient European HPC Processors*, in *2021 Design, Automation Test in Europe Conference Exhibition (DATE)* (2021), pp. 136–141

[14] N. Eicker, T. Lippert, T. Moschny, E. Suarez, for the DEEP project, Concurrency and Computation: Practice and Experience **28**, 2394 (2016), `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.3562`