Analysis of performance improvements for host and GPU interface of the APENet+ 3D Torus network.

R Ammendola A[1], A Biagioni[2], O Frezza[2], F Lo Cicero[2], A Lonardo[2], P S Paolucci[2], D Rossetti[2], F Simula[2], L Tosoratto[2] and P Vicini[2]

[1]INFN Roma II, Via della Ricerca Scientifica 1 - 00133 Roma, Italy
[2]INFN Roma I, P.le Aldo Moro 2 - 00185 Roma, Italy

E-mail: `piero.vicini@roma1.infn.it`

Abstract. APEnet+ is an INFN (Italian Institute for Nuclear Physics) project aiming to develop a custom 3-Dimensional torus interconnect network optimized for hybrid clusters CPU-GPU dedicated to High Performance scientific Computing. The APEnet+ interconnect fabric is built on a FPGA-based PCI-express board with 6 bi-directional off-board links showing 34 Gbps of raw bandwidth per direction, and leverages upon peer-to-peer capabilities of Fermi and Kepler-class NVIDIA GPUs to obtain real zero-copy, GPU-to-GPU low latency transfers. The minimization of APEnet+ transfer latency is achieved through the adoption of RDMA protocol implemented in FPGA with specialized hardware blocks tightly coupled with embedded micro-processor. This architecture provides a high performance low latency offload engine for both trasmit and receive side of data transactions: preliminary results are encouraging, showing 50% of bandwidth increase for large packet size transfers. In this paper we describe the APEnet+ architecture, detailing the hardware implementation and discuss the impact of such RDMA specialized hardware on host interface latency and bandwidth.

1. Introduction

Scaling towards exaFLOPS systems in HPC requires to select an high performances interconnection network solution able to also tackle with requirements of low power consumption, high efficiency and resilience. Most of the newest HPC platforms, as testified by TOP500 [1] and the Graph500 that enlist world's current most powerful computing systems, are mainly large clusters interconnected by a switch-less, multi-dimensional toroidal mesh. This solution has been proven effective to ensure lower costs in terms of power and equipment compared with a fat-tree one with comparable interconnect technology. Moreover scaling such mesh is easier, even while maintaing comparable or better communication latency for computing algorithms with a good degree of locality.

At the same time, TOP500 shows that GPU accelerators are gaining market in the supercomputing arena.

On this path we designed and deployed a PCIe network board for tera-scale clusters, the APEnet+ card [2], based on state-of-the-art FPGA. With its six fully bidirectional, point-to-point links, the APEnet+ board can effortlessly turn an assembly of off-the-shelf many-core x86_64 systems into a cluster with a 3D-torus network topology; an overview of the board architecture is given in Sec. 2.

The Network Interface is also able to directly access GPUs memory leveraging upon the peer-to-peer (P2P) capabilites of Fermi- and Kepler-class NVIDIA GPUs. This is a first-of-its-kind feature for a non-NVIDIA device that allows unstaged off-board GPU-to-GPU transfers with unprecedented low latency, as described in Sec. 2.1.

The overall architecture of APEnet+ has been evolutionarily improved by adding or re-engineering its parts in order to relieve or bypass the bottlenecks. The set of performance figures that this evolving design allowed to achieve are comparable with those of top offerings from the commercial world for the same PCIe generation; Sec. 3 describes more in-depth the areas where most of the work was performed.

Thanks to the reconfigurability of the FPGA platform and the modularity of our architecture, it has been possible to reuse APEnet+ in different contexts, for example as a low-latency interface between a read-out board and a GPU farm in the data acquisition system of the low level trigger of a high-energy-physics experiment (NaNet project)[3, 4].

Short term future roadmap foresees the delivery of a new release of the APEnet+ card relying on 28nm FPGAs and PCIe Gen3 bus. A glimpse on this work is given in Sec. 4 together with some conclusions.

2. Architecture Overview

APEnet+ is a point-to-point, low-latency, 3D-torus network controller integrated in a PCIe Gen2 board based on Altera Stratix IV FPGA developed at ”Istituto Nazionale di Fisica Nucleare” (INFN).

The APEnet+ network architecture has, at its core, the *Distributed Network Processor* (DNP), which acts as an off-loading network engine for the computing node, performing inter-node data transfers.

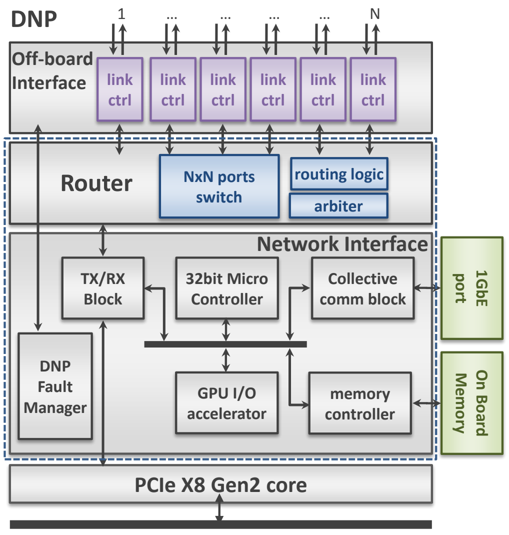

Figure 1. Overview of APEnet+. The DNP is the core of the architecture - composed by the Off-Board Interface, Router and Network Interface macro blocks - implemented on the FPGA. The system interfaces to the host through the PCIe bus.

The DNP has been developed as a parametric Intellectual Property library, meaning that some fundamental architectural features can be chosen among the available ones, others can be customized at design-time and new ones can easily be plugged in.

A highly modular design has been employed separating the architectural *Core* from the *Off-board Interface* block, connected by a customizable number of ports as shown in Fig. 1.

The Off-Board Interface manages the node-to-node communication flow over links equipped with a bi-directional Serializer/Deserializer, error checking and DC-balancing; it allows point-to-point, full-duplex connections for each node.

The Core block structure is split into a *Router* component, responsible for data routing and dispatch, and a *Network interface*, the packet injection/processing logic comprising host interface, Tx/Rx logic and μC.

The Router establishes dynamic links among the ports of the cross-bar switch, managing conflicts on shared resources.

The Network Interface block has basically two main tasks: on the transmit data path, it gathers data coming in from the PCIe port and forwards them to the relevant destination ports; on the receiving side, it provides hardware support for the Remote Direct Memory Access (RDMA) protocol, allowing remote data transfer over the network without involvement of the CPU of the remote node.

The GPU I/O Accelerator in Fig.1 allows to interface to the GPU for direct memory access, as described in Sec. 2.1.

APEnet+ features has been proved on the field by deploying the 3D toroidal network for the QUonG HPC platform at INFN Roma, Italy. QUonG is an INFN initiative targeted at developing a computing system dedicated to Lattice QCD computations; it is a massively parallel computing platform leveraging on commodity multi-core processors coupled with last generation GPUs as computing nodes interconnected by a point-to-point, high performance, low-latency 3D torus network, which is particularly suited to the transmission patterns of the set of algorithms LQCD belongs to. QUonG installation at the moment counts 16 nodes, each one equipped with 2 M2075 NVidia GPU and one APEnet+ board, and shows an aggregated peak performance of $\sim$32 TFLOPS [5].

2.1. Peer-to-Peer GPU memory access

A peculiar feature of APEnet+ can be exploited when the cluster nodes are equipped with Fermi- and Kepler-class NVIDIA GPUs: APEnet+ is the first non-NVIDIA device able to directly access their memory leveraging upon their peer-to-peer (P2P) capabilites. In this way Remote GPU-to-GPU data transfers are possible without staging and involving the CPU, resulting in a very low transfer latency. Figure 2 gives a pictorial view of the data flow on the TX side from the GPU memory through the Network Interface when the P2P is enabled, compared to what happens using a device that does not provide P2P access to the GPU memory as in Fig. 3. Without the P2P a copy is necessary from the GPU memory to the CPU memory to allow the Network Interface to dispatch the message. The same happens on the receiving side.

All things considered this APEnet+ feature allows to obtain a gain in reduced GPU-to-GPU transfer latency as shown in Fig. 4. A thorough description of APEnet+ P2P implementation can be found in [6] together with an example of an accelerated application; we remark that off-board transfer latency in the GPU-to-GPU case is competitive with current academic research in this field [7, 8].

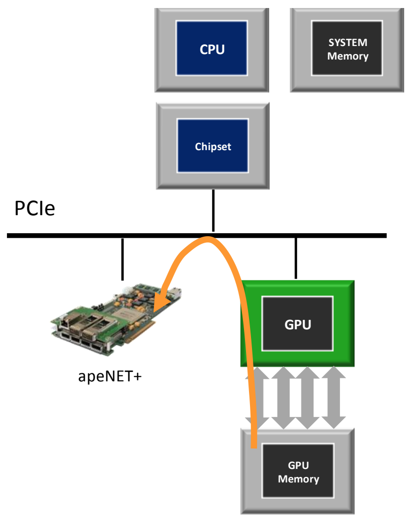

Figure 2. APEnet+'s GPU TX flow exploiting P2P.

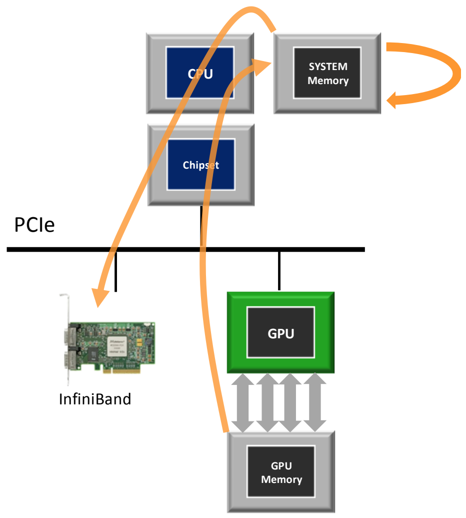

Figure 3. Traditional GPU TX data flow without the use of P2P: a copy GPU-to-CPU memory is necessary to complete the data send.

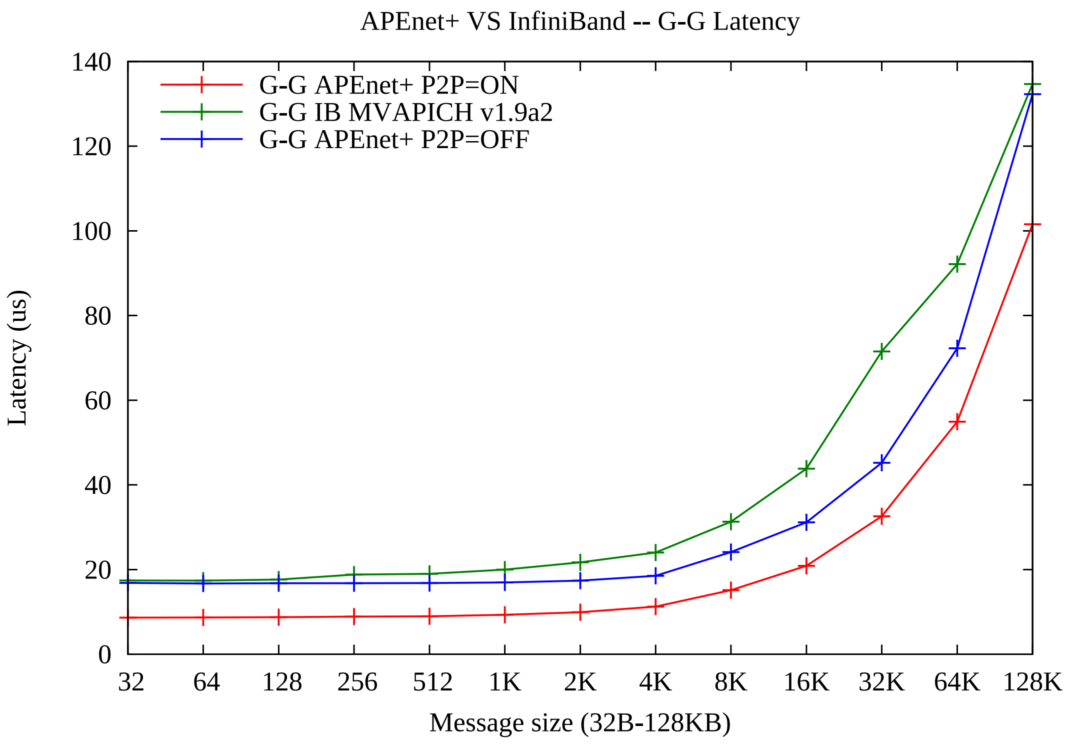

Figure 4. Plot showing the effect of APEnet+ P2P on GPU-to-GPU data transfer latency. Results are also compared with latencies of an Infiniband interconnect of the same PCIe generation, obtained using MVAPICH v1.9a2.

3. New hardware blocks for performance improvements

The use of FPGAs made possible for the development of APEnet+ a tight design cycle and modular and reconfigurable architecture. In this way we could evolve the architecture by working around the critical areas as shown by benchmarks, and remove or reduce the performance bottlenecks.

In the following section we describe some of this work performed on key areas of the IP.

3.1. Tx speed-up: Double DMA channel

The *Tx Block*, depicted in Fig. 5, is the DNP logic block in charge of loading data from CPU memory.

The host data buffer transmission is completely handled by the APEnet+ Kernel driver: PCIe data read transactions (Tx) are issued after the information contained in command-packets, each one composed by four 128bit-words: *Header*, *Footer*, *Cmd0* and *Cmd1*. The Tx Block uses the first two command words (Header and Footer) to encapsulate data coming from the host memory, creating data-packets that will be sent to the network, while Cmd0 and Cmd1 contain information used respectively to program the PCIe DMA data read – *i.e.* packet size and address – and to communicate the end of the transaction. Even commands are read in DMA mode, so the Host communicates to the reading logic the number N of incoming command packets and notifies when they are ready. Command-packets are placed in the host memory in a circular buffer; the read logic read and consume them through the *Multi Packet Instantiator*, possibly gathering more than one command in a single PCIe DMA read transaction.

In the first implementation these DMA read operations on the PCIe were completely sequential. In order to reduce the latency between two consecutive reading processes, we introduced a mechanism able to switch between two DMA channels (DMA0 and DMA1) popping alternatively new commands from the *Tx Cmd Fifo*.

Fig. 6 compares the read bandwidth measured using one or two DMA channel implementation, showing the bandwidth improvement equal to a gain $\sim 40\%$ on the TX side bandwidth.

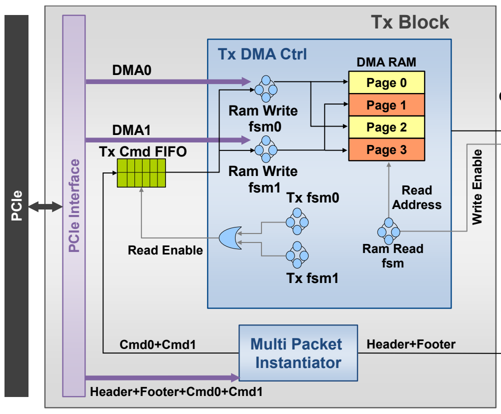

Figure 5. APEnet+'s CPU TX flow.

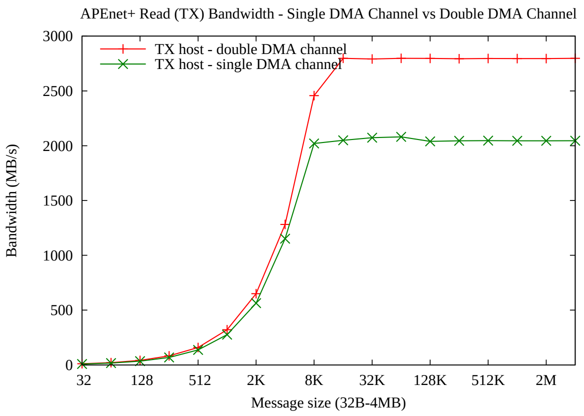

Figure 6. APEnet+ Tx single vs. double DMA results.

3.2. Rx speed-up: on-board memory management

The RDMA protocol implementation requires that the virtual memory management task has to be performed by the APEnet+ hardware on the receive side. Each packet header contains the virtual address of the destination buffer and once the packet reaches the remote destination, a virtual to physical address translation is needed to execute the DMA write in the host memory. So, the impact of this highly demanding task has to be limited to lower the latency of packet data transfer.

In the first implementation we delegated the complete task of memory management to the soft-core Altera Nios2 μC clocked at 200 MHz. At run-time, a table with user registered buffers and pages is loaded into the μC on-chip RAM by the APEnet+ device driver. A queue of incoming message requests is built upon packet arrivals: for each packet in queue the registration of the destination buffer is checked, and buffer information is gathered, including the buffer decomposition into pages and V2P address translation.

Such a μC implementation has shown a severe performance penalty with an average of 600 clock cycles to perform a single virtual address processing. To speed up the execution of these tasks we designed a specialized Translation Lookaside Buffer (TLB), similar to a modern CPU's Memory Management Units. A TLB is an associative cache where a limited amount of entries can be stored in order to perform memory management tasks, significantly reducing Nios2 processing time in case of a cached translation — a 'hit' — or forwarding the operation to the μC in case of a 'miss'. In Fig. 7 is depicted how the TLB interacts with the rest of the design. The current hardware TLB implementation takes 31 clock cycles (logic is running at 250 MHz) to perform the complete task of buffer look-up and address translation.

With a simple inter-node bandwidth test it is possible to measure the advantage given by

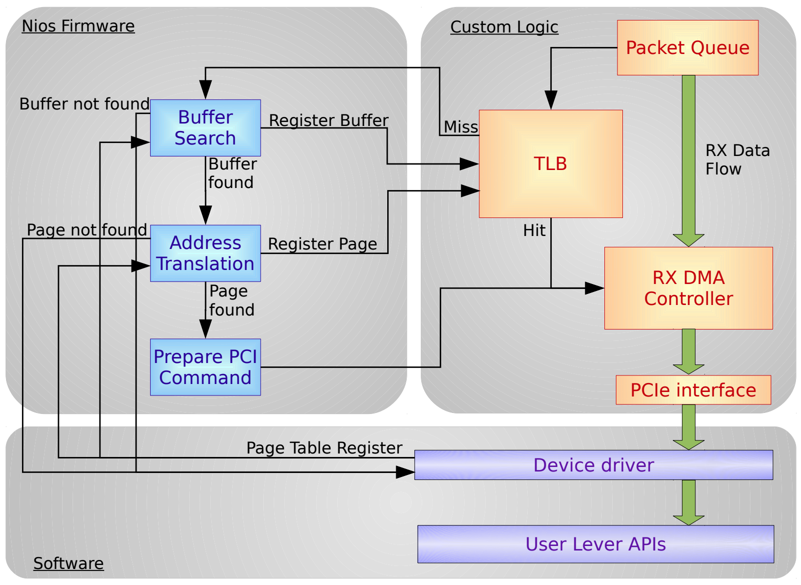

Figure 7. Schematic representation of memory management tasks. Three areas are interested: Software, with Page Table Registration; μC Firmware, interacting with the Device Driver for Registration and error handling (Buffer or Page not found events); Custom Logic, with the TLB in charge of packet processing and instructing the DMA engine logic.

the use the TLB. Results are shown in Fig. 8 for a TLB with 32 cacheable entries, where a comparison is made with the same test with the TLB disabled. Both Host- and GPU-bound data transfer are plotted, showing a performance benefit for message size up to 128 kB in the Host case and to 2 MB in the GPU case, due to the limited number of entries in this test (32) and to the memory paging size (4 kB for host memory, 64 kB for GPU memory).

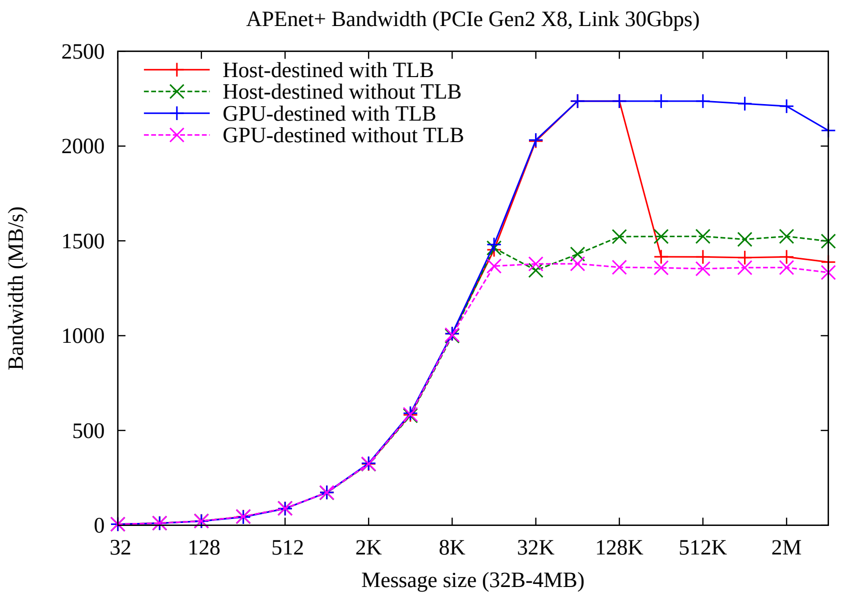

Figure 8. Bandwidth comparison in inter-node communication with (continuous lines) and without TLB (dashed lines). Use of TLB allows to reach a maximum bandwith of $\sim 2240MB/s$, instead of $\sim 1520MB/s$ in the host case and $\sim 1380MB/s$ in the GPU case when the TLB is disabled. The performance penalty for host-bound message size larger than 128 kB can be overcome by building a larger TLB.

4. Conclusion

APEnet+ is our custom Network Interface that can be used to implement high performance, low latency multidimensional toroidal networks for HPC clusters. A custom NIC for the QUonG cluster would have been impossibly lengthy and costly to build without the resources offered by FPGAs, which allow a tighter design cycle (prototyping, benchmarking and re-engineering for faster logic or bug fixes) and the fast development of a more reliable product. In this paper we described a couple of the iteration on this design cycle, that definitely helped to make the APEnet+ card the first P2P-enabled non-NVIDIA device, guaranteeing off-board GPU-to-GPU data transfers with with unprecedented low latency performances.

At the moment we are working to accommodate in our design the new features of next-gen FPGA: the all-round improvement coming from the 28 nm technology push, a PCIe Gen3 interface with upgraded encoding (from 8b/10b to 128b/130b) for doubled host bandwidth, acceleration from 8.5 Gb/s to 14.1 Gb/s for the transceivers, a more performing μC and an embedded 10GbE interface. While the 10GbE interface is mostly of interest for the evolution of the NaNet board [4], the resulting increased throughput will greatly benefit a new revision of the APEnet+ card, together with more resources to investigate application-specific acceleration methods or routing strategies.

Acknowledgments

This work was partially supported by the EU Framework Programme 7 project EURETILE under grant number 247846.

References

[1] *http://www.top500.org*

[2] Ammendola R, Biagioni A, Frezza O, Lo Cicero F, Lonardo A, Paolucci P.S, Rossetti D, Simula F, Tosoratto L and P Vicini 2012 *J. Phys. Conf. Ser.* **396** 042059

[3] Lonardo A *et al.* "Building a Low-Latency Real-time GPU-based stream processing system", *http://on-demand.gputechconf.com/gtc/2013/presentations/S3286-Low-Latency-RT-Stream-Processing-System.pdf*

[4] Lamanna G *et al.* 2013 to appear in *Proceedings of 15th International Workshop on Advanced Computing and Analysis techniques in Physics (ACAT)*

[5] Ammendola R, Biagioni A, Frezza O, Lo Cicero F, Lonardo A, Paolucci P S, Rossetti D, Simula F, Tosoratto L and Vicini P 2011 *Application Accelerators in High-Performance Computing (SAAHPC), 2011 Symposium* 113−122

[6] Ammendola R, Bernaschi M, Biagioni A, Bisson M, Fatica M, Frezza O, Lo Cicero F, Lonardo A, Mastrostefano E, Paolucci P S, Rossetti D, Simula F, Tosoratto L and Vicini P 2013 to appear in *Proceedings of 2013 IEEE International Symposium on Parallel and Distributed Processing Workshops (IPDPSW)*

[7] Hanawa T, Kodama Y, Boku T and M Sato 2013 "Interconnection network for tightly coupled accelerators architecture", in *2013 IEEE 21st Annual Symposium on High-Performance Interconnects (HOTI), San Jose, CA, U.S.A.* 79−82

[8] Bittner R, Ruf E and Forin A 2013 *Cluster Computing* 1−10 ISSN 138−7857 (*http://dx.doi.org/10.1007/s10586-013-0280-9*)