Journal of Physics: Conference Series **396** (2012) 042059 doi:10.1088/1742-6596/396/4/042059

APEnet+: a 3D Torus network optimized for GPU-based HPC Systems

R Ammendola[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 Tor Vergata, Italy
[2] INFN Roma, Italy

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

Abstract. In the supercomputing arena, the strong rise of GPU-accelerated clusters is a matter of fact. Within INFN, we proposed an initiative — the QUonG project — whose aim is to deploy a high performance computing system dedicated to scientific computations leveraging on commodity multi-core processors coupled with latest generation GPUs.
The inter-node interconnection system is based on a point-to-point, high performance, low latency 3D torus network which is built in the framework of the APEnet+ project. It takes the form of an FPGA-based PCIe network card exposing six full bidirectional links running at 34 Gbps each that implements the RDMA protocol.
In order to enable significant access latency reduction for inter-node data transfer, a direct network-to-GPU interface was built.
The specialized hardware blocks, integrated in the APEnet+ board, provide support for GPU-initiated communications using the so called PCIe peer-to-peer (P2P) transactions.
This development is made in close collaboration with the GPU vendor NVIDIA.
The final shape of a complete QUonG deployment is an assembly of standard 42U racks, each one capable of 80 $TFLOPS/rack$ of peak performance, at a cost of 5 $k€/TFLOPS$ and for an estimated power consumption of 25 $kW/rack$.
In this paper we report on the status of final rack deployment and on the R&D activities for 2012 that will focus on performance enhancement of the APEnet+ hardware through the adoption of new generation 28 nm FPGAs allowing the implementation of PCIe Gen3 host interface and the addition of new fault tolerance-oriented capabilities.

1. Introduction

Lattice Quantum Chromo-Dynamics (LQCD) is still today one of the most demanding scientific application in terms of computing power, as latest LQCD codes require several PFLOPS-year[1]. Typical LQCD algorithms exploit the peculiarities of locality and uniformity in their datasets, which clearly states the features that a computing platform must have to be suitable for LQCD:

- a high performance elementary arithmetic data path engine targeted for matrix algebra;
- a low latency, high performance multi-dimensional torus network optimized for next-neighbours communications.

Besides, these same features can be effective on other numerical simulations which share the same symmetries.
These requirements have driven the design of past generations of custom LQCD-dedicated

Published under licence by IOP Publishing Ltd

computing machines: purely custom-made systems, where every critical component was designed *ad-hoc* (APE systems, QCDSP and its successors QCDOC and BlueGene [2, 3, 4, 5, 6, 7, 8], CP-PACS [9]) and clusters of commodity-hardware interconnected by multi-dimensional torus network *ad-hoc* designed (APEnet [10, 11] and more recently QPACE [12]).
Nowadays, the Non-Recurring Engineering costs of ASIC development at the current VLSI technologies (45 nm and smaller) require multi-million dollars investments; at the same time, GPUs hold the stage with their impressive FLOPS peak performance (more than 1 TFLOPS per device) and excellent FLOPS per Watt and FLOPS per dollar ratios.
As a consequence, GPUs are becoming mainstream in the HPC arena: they feature prominently in the current TOP500 list[13] with three of the top five positions sporting GPU-equipped systems, where GPUs are conceived as accelerators.
Yet, room is still left for interconnection networks which are efficient, scalable and provide GPU-specific optimizations and integration, to further push GPU acceleration of scientific HPC clusters.

The APEnet cluster interconnect is such a technology, starting its development in 2003; the design of the latest version, APEnet+, has been proposed in [14, 15] and shows now the first impressive benchmark results.
On top of this, an HPC initiative, QUonG (lattice QUantum chromo-dynamics ON Gpu), was proposed by INFN to deploy a platform dedicated to scientific computations — tuned to the LQCD algorithms requirements — that is built on commodity multi-core processors, coupled with latest generation GPUs and interconnected by APEnet+ cards.
The structure of the paper is the following: in the next section (2) we introduce the APEnet+ interconnect, whose preliminary bechmarks results are shown in section 4. Section 3 describes the first QUonG installation at INFN Roma and in section 5 we list our next months R&D ideas to improve the APEnet+ card in order to achieve the best performances.

2. APEnet+ card

The APEnet project aims at developing a network fabric which allows assembling an HPC cluster *à la* APE with off-the-shelf components. Its latest generation is the APEnet+ board [15]: a FPGA-based PCIe board with 6 fully bidirectional off-board links with 34 Gbps of raw bandwidth per direction, state-of-the-art signaling capabilities — up to X8 Gen2 bandwidth towards the host PC — and a Remote Direct Memory Access (RDMA) protocol that, leveraging upon peer-to-peer (P2P) capabilities of Fermi-class NVIDIA GPUs [16] allows real zero-copy, low latency GPU-to-GPU transfers.
The APEnet+ network architecture has, at its core, the Distributed Network Processor (DNP). The DNP acts as an off-loading reconfigurable network engine for the computing node, performing inter-node data transfers.
The current implementation is deployed on a high performance Stratix IV Altera® FPGA. The DNP hardware blocks structure, depicted in Fig. 3, is split into a *network interface* — the packet injection/processing logic comprising host interface, TX/RX logic, Microcontroller and GPU/IO Accelerator — a *router component* and multiple *torus links*.

2.1. Torus Links

Node-to-node communications flows over links which have a bidirectional Serializer/Deserializer (Ser/Des), error checking, DC-balancing and autonomous retransmission capability. The torus link block manages the data flow by encapsulating packets into a light, low-level *word-stuffing* protocol able to detect transmission errors via CRC. These links allow point-to-point, full-duplex connection of each node with its six neighbours with coordinates both increasing and decreasing along the X, Y and Z axes, forming a 3D torus; each channel has a (theoretical) raw bandwidth of 34 Gbps.

2.2. Router
The Router block establishes dynamic links among the 8 ports cross-bar switch, six external torus links and two internal links. The routing policy applied to communications on the port of the switch is *dimension-ordered* and the order of dimension evaluation is selectable at run-time.

2.3. Network Interface
The Network Interface block has basically two main tasks: on the transmit data path, it gathers data coming in from the PCIe port — either from host or GPU memory — fragmenting the data stream into packets which are forwarded to the relevant destination ports, depending on the requested operation. On the receive side, it implements PUT and GET semantics providing 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.

2.4. Micro Controller
A zero-copy, off-loaded implementation of the RDMA protocol requires the network card to be aware of and autonomously handle all details pertaining to buffers registered by the application. On top of this, x86/x86_64 Linux OS adds the complexities of virtual memory management.

To address this task, we use a soft-core micro controller (μC) on the FPGA, which executes a firmware implementing:

- a Look-Up Table (LUT) with memory addresses and lengths of registered buffers;
- a Page Table (PT), mimicking the Linux kernel data structure, to perform buffer decomposition into pages and virtual-to-physical address translations.

One LUT/PT instance is allocated once per OS process and once per GPU; each APEnet+ can be used by up to 4 OS processes at a time. At application startup, LUTs and PTs are allocated by the μC firmware and updated by the APEnet+ kernel driver; their management and querying is done by the μC, during the processing of both incoming and outgoing message requests.

2.5. GPU/IO accelerator
GPUs are self-contained systems with their own device memory; pushing and pulling data to and from the host memory is an inevitable encumbrance for any computing that is staged on the GPU. This chore is even more evident in Multi-GPU systems: in GPU-accelerated clusters, a memory transfer between GPUs belonging to different nodes implies, besides the necessary network transfer, either one more copy from the sending GPU to its host memory and another one from the receiving host memory to the receiving GPU. This has significant repercussions on the bandwidth and latency of transfers to and from GPU memory. On the other hand, GPUDirect technology from NVIDIA allows direct data exchange between GPUs – when they both are behind the same I/O Hub (IOH) — without any CPU intervention or intermediate buffer.

APEnet+ has the ability to take part in the so-called PCIe peer-to-peer (P2P) transactions [17]; it is the first non-NVIDIA device with specialized hardware blocks [18] to support the NVIDIA GPUdirect peer-to-peer inter-GPU protocol. This means that the APEnet+ network board can target GPU memory by ordinary RDMA semantics with no CPU involvement and dispensing entirely with intermediate copies (see Fig. 2). In this way, real zero-copy, inter-node GPU-to-host, host-to-GPU or GPU-to-GPU transfers can be achieved, with substantial reductions in latency. Preliminary results on latency measurements are very promising (see section 4).

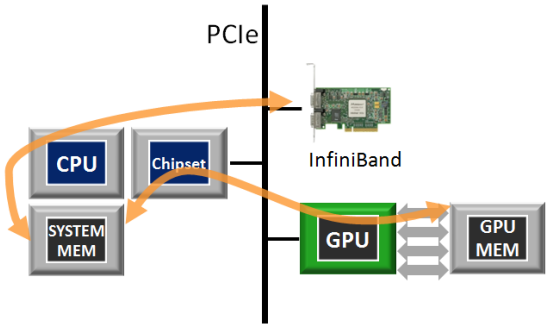

Figure 1. The *traditional* two steps necessary to transmit a GPU memory buffer across the network.

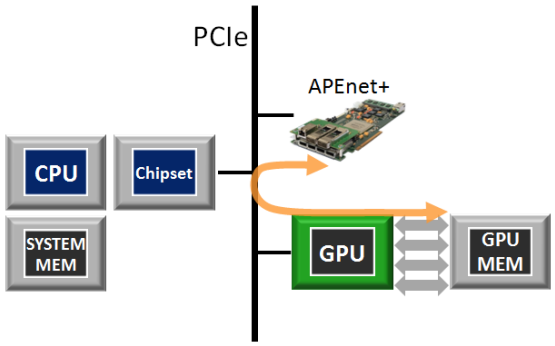

Figure 2. Here the P2P transfer over the PCIe bus enable for Zero-copy GPU memory TX/RX.

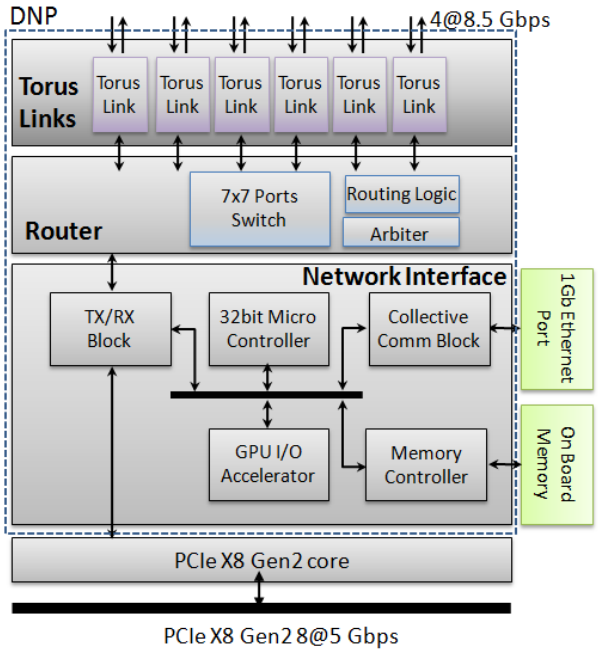

Figure 3. Overview of APEnet+. The DNP is the core of the architecture — composed by the Torus Links, Router and Network Interface macroblocks — implemented on the FPGA. The system interfaces to the host CPU through the PCIe bus.

2.6. The APEnet+ host adapter

The APEnet+ hardware is based on a recent generation Altera FPGA, the EP4SGX290, which is part of the Altera 40nm Stratix IV device family; it is equipped with 48 full-duplex CDR-based transceivers, 32 of them supporting data rates up to 8.5 Gbps each [19]. Though the device supports up to two PCIe links, the widest link lane width supported is 8[1]. The APEnet+ subsystem is a standard full-length, full-height, two I/O slots-wide PCIe card. The torus link mechanics is based on QSFP+ (Quad Small Form Pluggable+) [20], an electrical and mechanical standard for point-to-point links over copper wires or optical fibers. All the available FPGA transceiver blocks have been used to implement the 6 torus links and the PCIe x8 interface.

[1] Intrinsic channel bonding limitations prevent PCIe x16 configuration.

Due to the FPGA architecture limitations, this solution represents a balanced trade-off between torus link and PCIe link bandwidth.

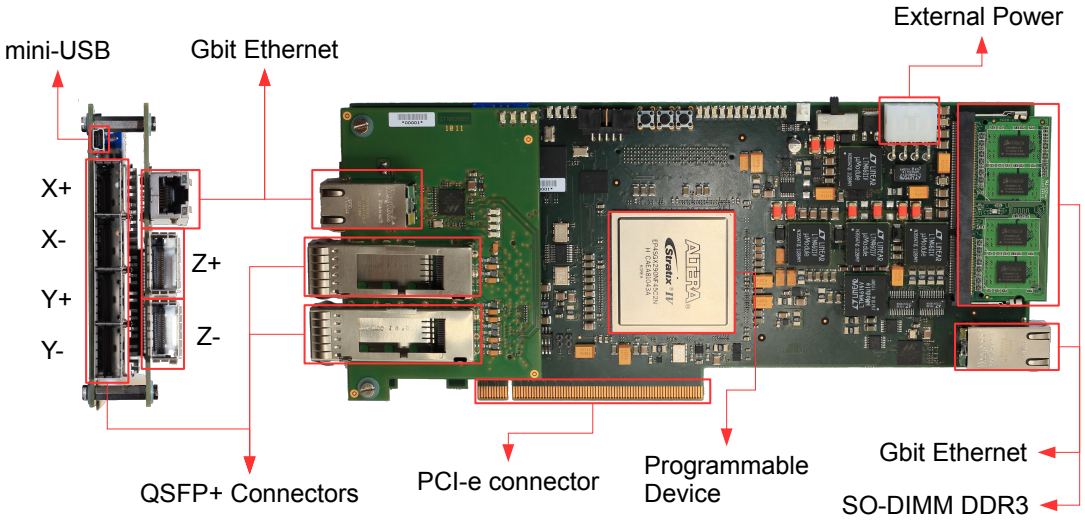

Figure 4. A photo of the APEnet+ card; it is an almost full-length double-wide PCIe card. The FPGA, the 6 link cable connectors and the SODIMM memory module are highlighted. Due to mechanical limitations of the development kit, two links are on a piggy-back module.

2.7. APEnet+ fault tolerance features

When scaling to peta/exa-scale in HPC systems, techniques that aim to maintain a low Failure In Time (FIT) ratio are mandatory.

Our idea is to apply a systemic approach by splitting the fault-tolerance problem into two areas: *fault awareness* and *fault reactivity*. A system is *fault-aware* once it has a global, consistent picture of its own health status, made up of the collected diagnostics of its single devices. The assessment of the health status of each subcomponent can be referred to as *local fault detection*. To obtain the complete awareness it is necessary to propagate the status information through the system hierarchy. Once the awareness is achieved, decisions can be made on how to *react* to faults. APEnet+ provides a way to obtain the awareness: a dedicated HW block is able to gather information about the network fabric status and to detect *faults* (*e.g.* link malfunctions) or *critical events* (*e.g.* temperature over threshold). Moreover, it can be part of a mutual watchdog mechanism, together with the companion software component running on the host node: the two devices monitor each other and, in case of a host breakdown, APEnet+ is able to send diagnostic messages towards the first neighbouring nodes so that fault awareness is preserved and up to date.

3. QUonG HPC platform

The QUonG modular system is a cluster of hybrid elementary computing nodes, based on commodity Intel-based host processors and NVIDIA GPUs acting as floating point accelerators. Multiple QUonG elementary computing nodes are arranged in a cubic mesh using the INFN custom 3D torus network — the APEnet+ network architecture — to efficiently interconnect

the elementary computing nodes, allowing the system to scale at PFLOPS level with $10^4 \div 10^5$ nodes.

Our reference cluster is made of a replication of the "QUonG elementary computing unit", a combination of multi-core CPU, GPU accelerator and an APEnet+ card. It is a flexible and modular platform that can be tailored to different application requirements by changing the GPU vs CPU ratio.

Given the computational costs of typical state-of-the-art LQCD calculations, the constraints that come from an efficient implementation of such codes on GPU-based platforms and the APEnet+ hard limits in terms of communication bandwidth, the best configuration for an LQCD-dedicated QUonG unit is 1 multi-core CPU, 2 NVIDIA GPUs and 1 APEnet+ card.

The QUonG elementary mechanical assembly is a 3U stack consisting of two servers, each one equipped with two multi-core CPUs and an APEnet+ card, which *sandwich* one NVIDIA S2075 multiple GPU system with 4 NVidia M2075 GPUs. This assembly collates two QUonG *elementary computing units* — *units* from now on — where each server hosts one APEnet+ board and one interface board to drive 2 (out of the 4) GPUs inside the S2075; a QUonG unit is topologically equivalent to two vertexes of the APEnet 3D mesh.

A QUonG rack counts 14 elementary mechanical assemblies into a 42U standard rack enclosure (see Fig. 5). The aggregated peak performances are 58 and 29 TFLOPS in single and double precision respectively, with an estimated power consumption around 25 kW and at an estimated cost around 300 k€. The peak performance could reach 80 TFLOPS SP when eventually upgrading to NVidia S2090.

At the beginning of 2012 we started integrating a QUonG system. At the end of March 2012 we have at INFN Roma six assembled QUonG units for an aggregated performance of 25 TFLOPS (SP).

4. APEnet+ benchmark results

The APEnet+ benchmarks were performed on a QUonG experimental platform mounting SuperMicro servers with dual Xeon 56xx processors, 24GB system memory, running CentOS 5.7 x86_64, one NVIDIA C2050 GPU in an x16 Gen2 slot. The APEnet+ cards used were preliminary and used a reduced link speed of 13 Gbps.

Our benchmark program is coded using the APEnet RDMA APIs and is basically a one-way point-to-point test involving two nodes, similar in spirit to the OSU MPI bandwidth test[2]. Basically, the receiver node allocates a buffer on either host or GPU memory, registers it for RDMA, sends its address to the transmitter node, starts a loop waiting for N buffer received events and ends by sending back an acknowledgment (ACK) packet. The transmitter node waits for an initialization packet containing the receiver node buffer (virtual) memory address, writes that buffer N times in a loop with RDMA PUT, then waits for a final ACK packet.

Fig. 4 shows the timings plot for our benchmark test with small message sizes (up to 4 KB). This plot is our estimate of the APEnet+ one-way communication latency[3]. 'H' stands for Host and 'G' for GPU; the lower couple of lines (red and green) are for TX of Host memory, while the upper ones (blue and purple) are for GPU memory TX. The test does not take advantage of any special optimization trick for small messages, like copying of data in temporary buffers; we still need work for the pipelining capability of the APEnet+ HW, so we expect it to perform not very differently from a round-trip test. In the near future, we plan to analyze and optimize the round-trip as well as the bidirectional bandwidth benchmarks.

[2] `http://mvapich.cse.ohio-state.edu/benchmarks/`

[3] Note that this is not the half round-trip latency as usually reported in literature.

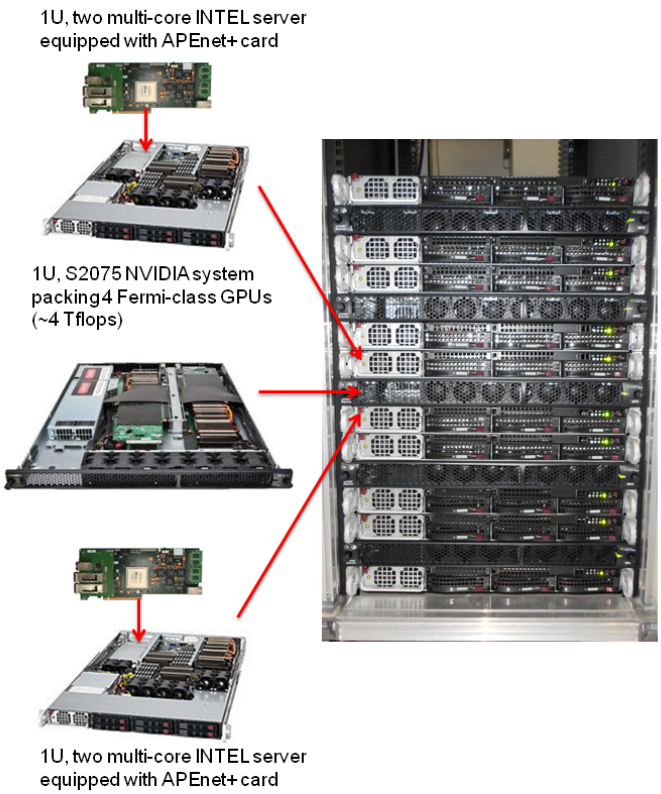

Figure 5. Pictorial view of a QUonG rack and a zoom of the QUonG elementary mechanical assembly.

From the plot, a factor 1.5 penalty is apparent when the transmitting buffer is located on the GPU; this is dependent on details of the P2P protocol that we are not entitled to disclose and which will be addressed in the near future. The initial 5-6 μs latency shown in Fig. 4 is a promising result that is expected to be improved upon.

Figure 7 compares the improvements brought about by the P2P support: switching off the P2P optimizations, APEnet+ results are those on the middle (green) line; in this case timings include two `cudaMemcpy()`'s to pull data from GPU memory to host memory and vice-versa on the receiving side. Enabling the P2P feature, the results are those on the lower (red) curve. From the plot it is clear that the difference between the two curves is about 7 μs, which is approximately twice the cost of a single `cudaMemcpy()`. The APEnet+ card shows a latency improvement roughly by a factor 3. To give an idea, the points on the upper (blue) line come from a bidirectional latency test of MVAPICH2, MPI over InfiniBand from the Ohio State University [4], which does not exploit the P2P feature to boost the transfers.

The plot in Fig. 8 is the result of a very preliminary bandwidth benchmark. It should be noted that there is no such difference between lines having the same source buffer memory type, *i.e.* the type of destination memory has no significant impact. As mentioned previously, the

[4] `http://mvapich.cse.ohio-state.edu/performance/mvapich2/inter_gpu.shtml`

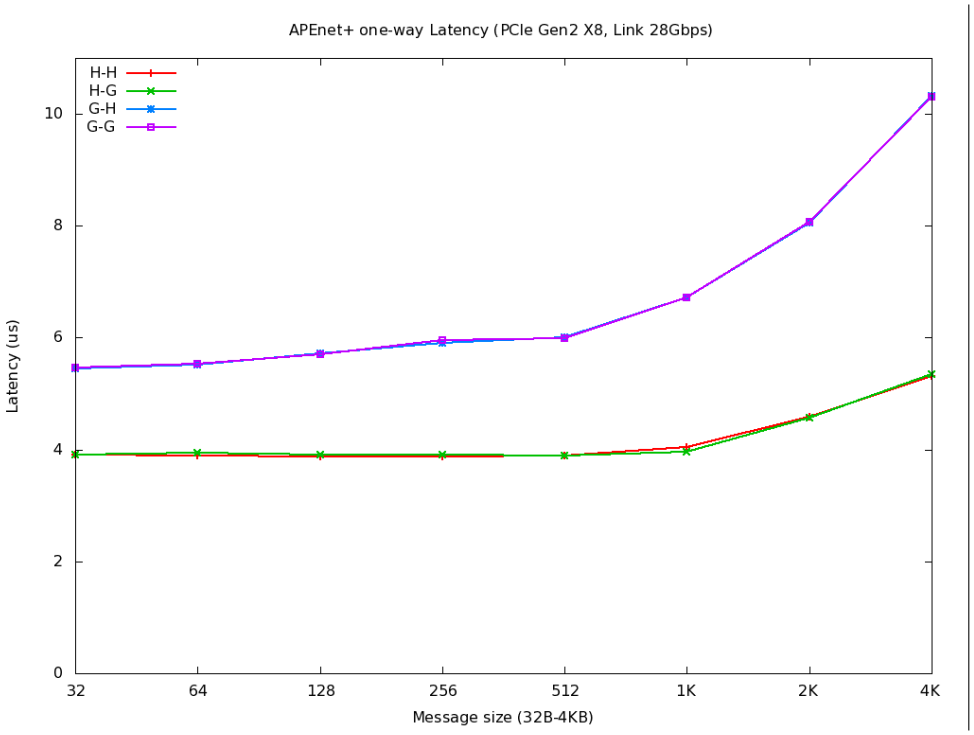

Figure 6. Characterization of the APEnet+ board: Latency.

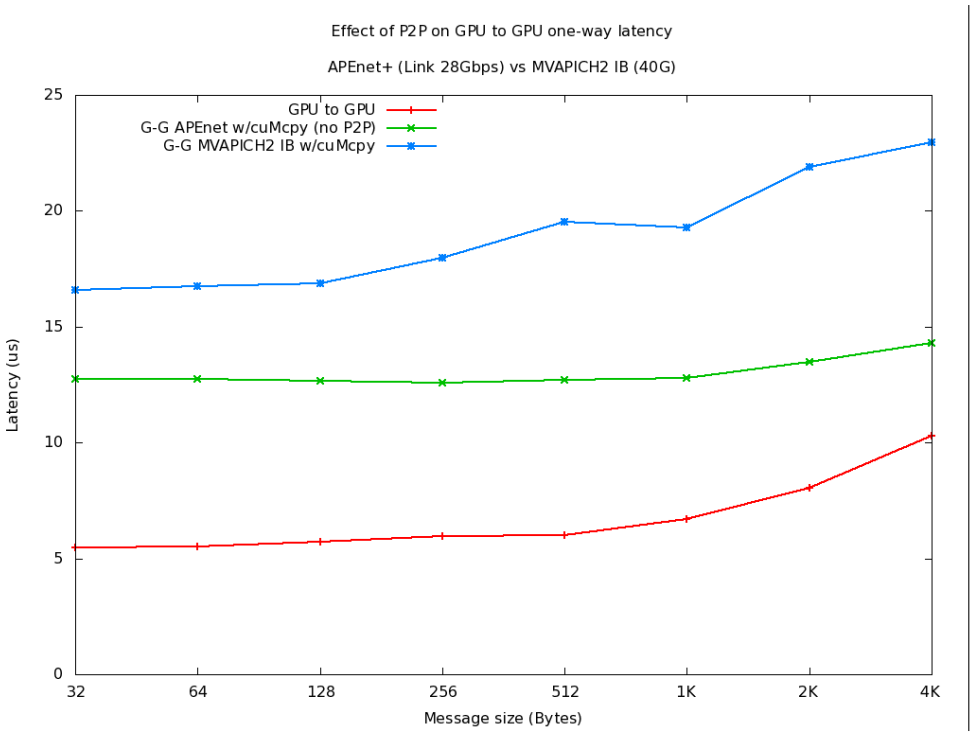

Figure 7. Characterization of the APEnet+ board: APEnet+ vs. MVAPICH over IB.

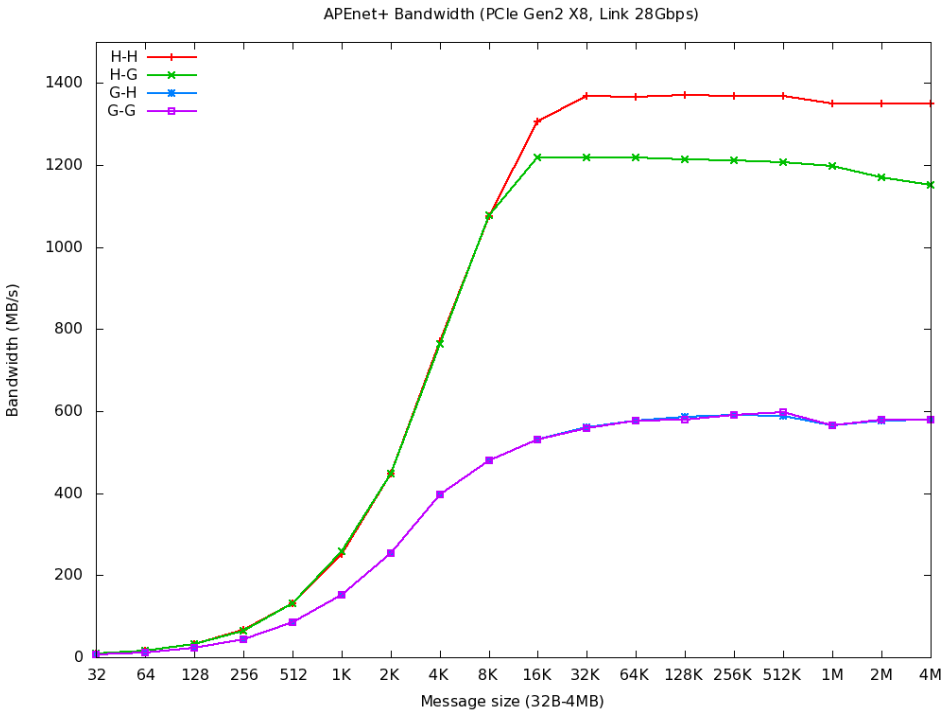

Figure 8. Characterization of the APEnet+ board: Bandwidth.

discriminating factor is the use of the GPU as a transmitter. The upper curves in figure 8 rise as expected up to a message size of 16 KB, where a plateau appears. To remove this limitation at the moment we are investigating on how to accelerate the receiving tasks performed by the μC. The lower curves — where GPU memory is the transfer source — show a low asymptotic bandwidth of roughly 600 MB/s. This is mainly due the high impact of the P2P read protocol, analogously to the effect seen in figure 4, which adds a constant overhead to the transmission of each 4 KB packet; in the current preliminary implementation, where the protocol is fully implemented in software on the μC firmware, the overhead is not overlapped among subsequent packet transmissions, so that it basically acts as multiple barriers, preventing the pipelining of the packet flow.

5. Future work

Next months research and development work will address the areas where we know there is significant headroom for improvement, like investigating about how to accelerate the receiving tasks performed by the μC and to obtain a full overlap among subsequent packet transmission, pipelining the packet flow. On the other hand, promising architectural enhancements will be applied once APEnet+ is updated to next generation 28 nm FPGA (like Altera Stratix V). The implementation of larger buffers will allow the handling of bigger packets and, given the market availability of PCIe Gen3-capable host platforms and GPUs, the increased number of transceivers on the FPGA board will allow a Dual PCIe Gen3 interface at 8Gbps and a better encoding (128b/130b), boosting APEnet+ bandwidth by a factor 4. The tranceiver frequency could be also raised up to 14 Gbps, bringing a 2X gain on the torus links bandwidth performances.

In the perspective of a complete QUonG HPC platform deployment, the development of APEnet+ fault handling/tolerance capabilities will be fully integrated with the software

environment in order to safely scale to multi-PFLOPS.

6. Acknowledgments

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

The authors would like to thank Massimiliano Fatica and Timothy Murray of NVIDIA Corp. for supporting the GPU P2P developments.

References

[1] M.J. Savage, *Nuclear Physics from QCD: The Anticipated Impact of Exa-Scale Computing*, arXiv:1012.0876v1.
[2] M. Albanese et al. *The Ape Computer: an Array Processor Optimized for Lattice Gauge Theory Simulations*, Comput. Phys. Commun. 45, 345- 353 (1987).
[3] N. Avico et al. *From APE to APE-100: from 1 to 100 gflops in Lattice Gauge Theory Simulations*, Comput. Phys. Commun. 57, 285- 289 (1989).
[4] A. Bartoloni et al. *An Overview of the APEmille Project*, Nucl. Phys. B - Proc. Suppl. **60**A:237-240, January 1998.
[5] F. Bodin et al., *The APENEXT project*, Proceedings of *Lattice2001 conference*, Nucl. Phys. B - Proc. Suppl. **106**:173-176, March 2002.
[6] P.A. Boyle et al., *Overview of the QCDSP and QCDOC computers*, IBM Journal of Research and Development, Vol. 49 Issue:2.3, 351-365, March 2005.
[7] D. Chen et al., *QCDOC: A 10-teraflops scale computer for lattice QCD*, Nucl. Phys. B - Proc. Suppl. **94**:825-832, March 2001.
[8] `http://www.research.ibm.com/journal/rd49-23.html`.
[9] S. Aoki et al., *Full QCD simulation on CP-PACS*, Nucl. Phys. B - Proc. Suppl. **60**A:246-254, January 1998.
[10] R. Ammendola et al., *APEnet: LQCD clusters à la APE*, Proceedings of *Lattice 2004* conference, Nucl. Phys. B - Proc. Suppl. **140**:826-828, March 2005 [arXiv:hep-lat/0409071v1].
[11] R. Ammendola et al., *Status of the APEnet project*, Proceedings of *Lattice 2005 conference*, PoS LAT2005 (2005) 100.
[12] G. Goldrian et al., *QPACE: Quantum Chromodynamics Parallel Computing on the Cell Broadband Engine*, Computing in Science & Engineering, Vol. 10 Issue 6, 46-54 November 2008.
[13] `http://www.top500.org`.
[14] R. Ammendola et al., *APEnet+: high bandwidth 3D torus direct network for petaflops scale commodity clusters*, [arXiv:1102.3796v1 [physics.comp-ph]].
[15] R. Ammendola et al., *APEnet+: a 3D toroidal network enabling petaFLOPS scale Lattice QCD simulations on commodity clusters*, Proceedings of *Lattice 2010 conference*, PoS LAT2010 (2010) 022
[16] NVidia GPUDirect technology `http://developer.nvidia.com/object/gpudirect.html`.
[17] `http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_Express_Base_r2_1_04Mar09_cb.pdf`
[18] R. Ammendola et al., *Mastering Multi-GPU Computing on a Torus Network*, Poster @ NVIDIA GPU Technology Conference 2010, San Jose (CA).
[19] `http://www.altera.com/products/devices/stratixfpgas/stratix-iv/transceivers/stxiv-transceivers.html`
[20] `ftp://ftp.seagate.com/sff/SFF-8436.PDF`