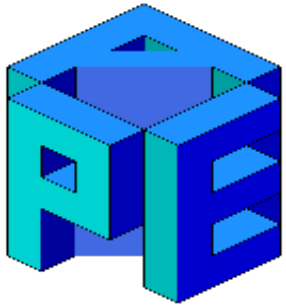




APEnet+, a custom 3D Torus network for GPU-based computing systems: development status and performance improvements for host and GPU interface.

Piero Vicini - INFN Roma

ACAT2013 – Beijing May 2013

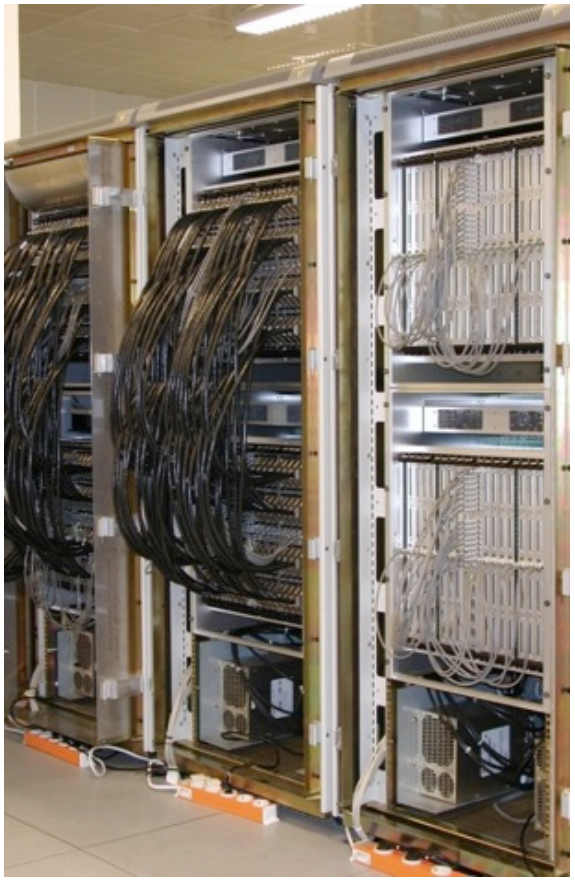
A little bit of history: from APE1 to apeNEXT

APE is a 25 years old project (!)

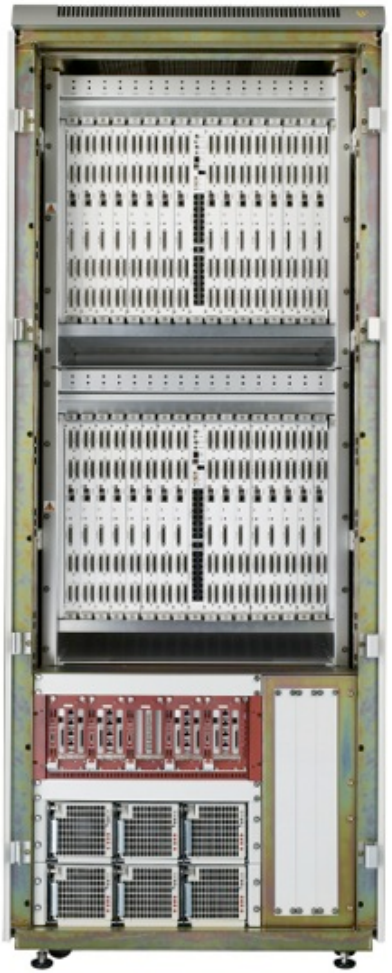
- MPP (APE1, APE100, APEmille, apeNEXT) & PC Cluster interconnection network (apeNET)
- FP Engine optimized for application + Smart dedicated 3D Torus interconnection network

APE1 (1988) 1GF, chipset Weitek

APE100 (1992) 25GF, SP, REAL "Home made" VLSI processors



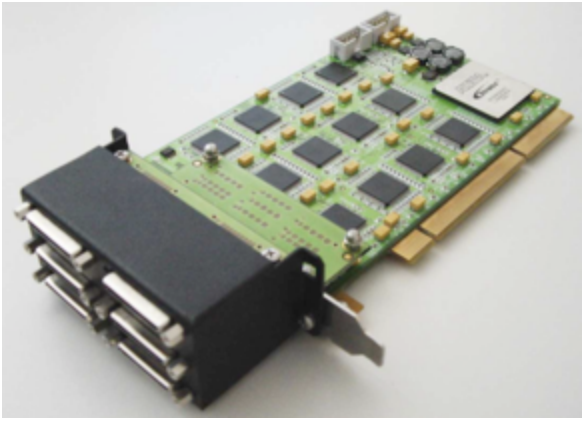
APEmille (1999) 128GF, SP, Complex
Italy+France+Germany collaboration

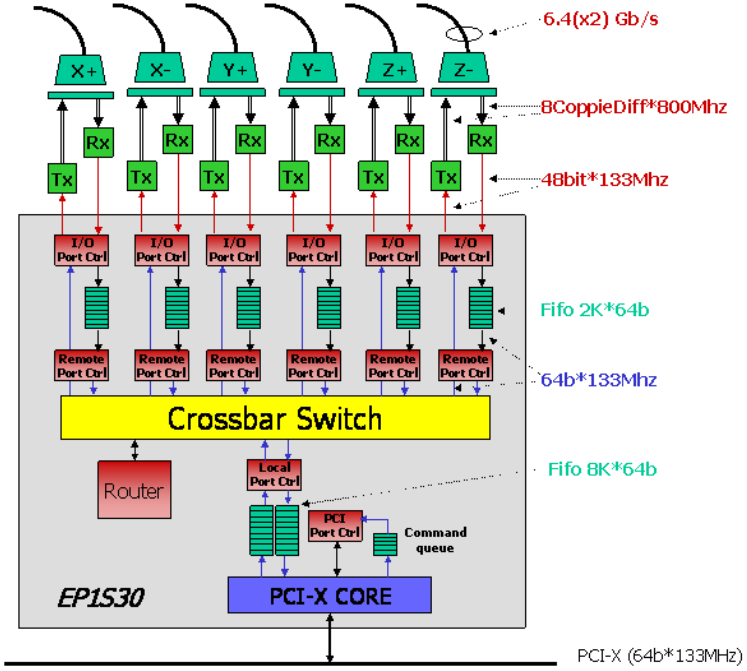


apeNEXT (2004)
800GF, DP, Complex

A little bit of history: apeNET

- APEnet: PC Cluster 3-d torus network
  - Integrated routing and switching capabilities
  - High throughput, low latency, "light-weight" protocol
  - PCI Interface, 6 Links full-bidir on torus side



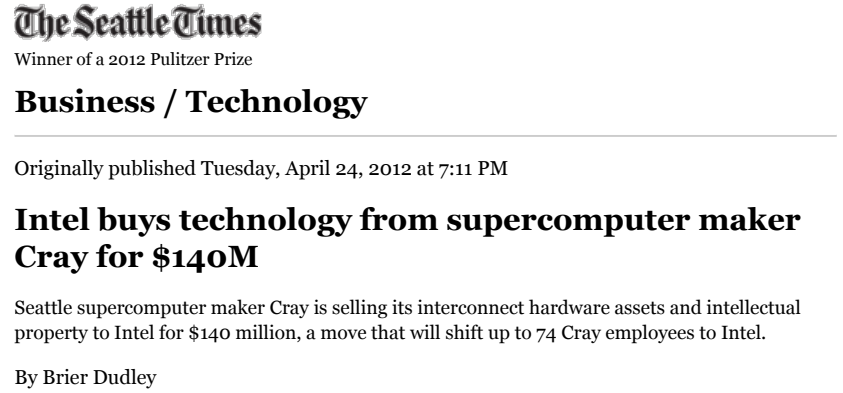


- History
  - 2003-2004: APEnet V3 (PCI-X)
  - 2005: APEnet V3+
    - same HW with RDMA API
  - 2006-2009: APEnet goes embedded
    - DNP, D(istributed) N(etwork) Processor
    - EU SHAPES project co-development
  - 2011: APEnet+
    - PCI Express, enhanced torus links

The (many) million dollars question….

Time changes but some facts are still true.

1) The hunger for floating point computing power remains unchanged
2) Towards the ExaFlops, main keywords remain unchanged
   - Floating Point Engines allowing efficient execution (i.e. high ratio flop/watt) of scientific applications -> ***high ratio flop/watt***
   - Smart and efficent specialized interconnection system to scale up to systems made of huge number of computing nodes (100-10000-100000-….) -> ***3D Torus network***



The Seattle Times

Winner of a 2012 Pulitzer Prize

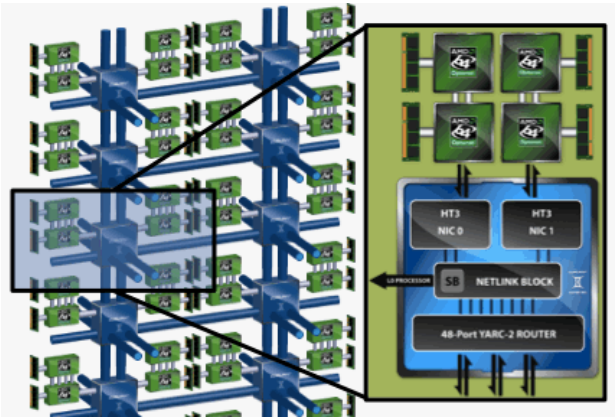
Business / Technology

Originally published Tuesday, April 24, 2012 at 7:11 PM

Intel buys technology from supercomputer maker Cray for $140M

Seattle supercomputer maker Cray is selling its interconnect hardware assets and intellectual property to Intel for $140 million, a move that will shift up to 74 Cray employees to Intel.

By Brier Dudley



Cray Gemini is a 3D Torus network quite similar to APE (sigh…)

   - Dense and robust low power system assembly required
3) NRE costs for custom developments of ASICS, systems, networks reached absurd high levels…

So the question is:

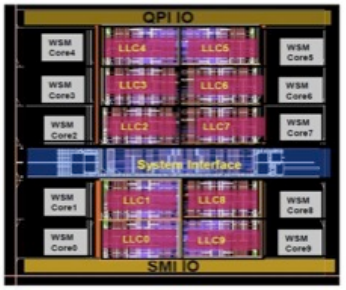
can we use commodities technologies to meet

the demands of computing power of modern scientific applications?

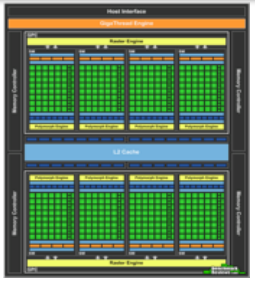
Piero Vicini – INFN Roma (piero.vicini@roma1.infn.it)

Peta(Exa)Flops scale enabling technologies: GPGPU

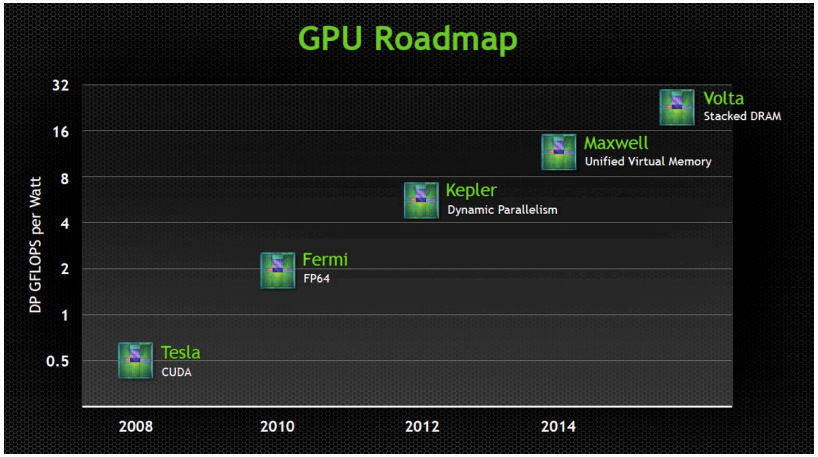
- General Purpose Graphic Processing Unit: impressive peak performance (N*TFlops per chip)
- Videogames market i.e. 10 G$/yr unified gaming and HPC chip architectures
- Architecture and characteristics fit with HPC scientific application (LQCD as an example…) requirements
  - Many-Core (>>100) SIMD-like architecture
  - High local memory bandwidth
    - 140 GB/s -> 500 GB/s
  - "Green" and cost effective
- Aggressive but (really!) feasible roadmap: much room for performance scaling
  - Easy peak performance scaling allowed by "tiled" architecture
  - New features added generation by generation….
  - Adoption of new technologies to improve performance

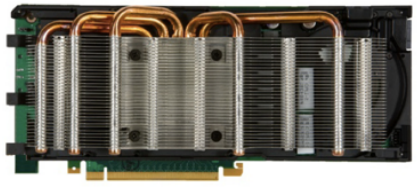


INTEL Westmere
+many caches - few processing

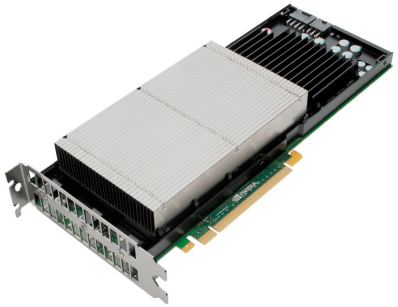


NVidia Fermi GPU
many computing units!!!



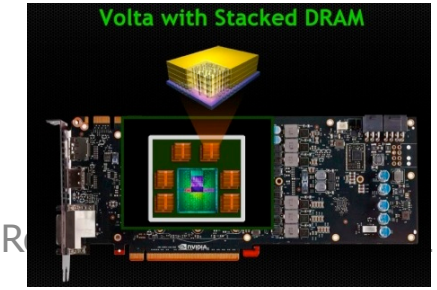


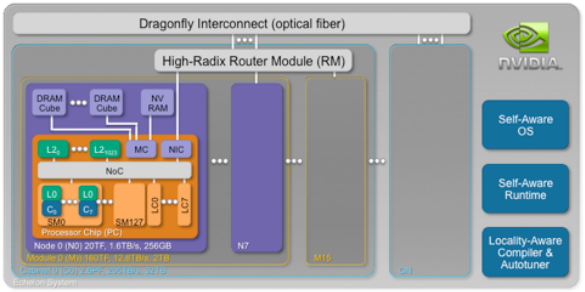
Nvidia Fermi (Tesla 20xx)
~500 core, **1 TF SP, 0.5 TF DP**
6 GB external memory (**150 GB/s**)



Nvidia Kepler (K20..)
~**2500** CUDA core, >**3x FP** (4 TF(SP)/ 1.3TF (DP))
6 GB external memory (**250 GB/s**)

Nvidia VOLTA
Est. **5x K20x** performance
1 TB/s memory bandwidth
New architectural improvements

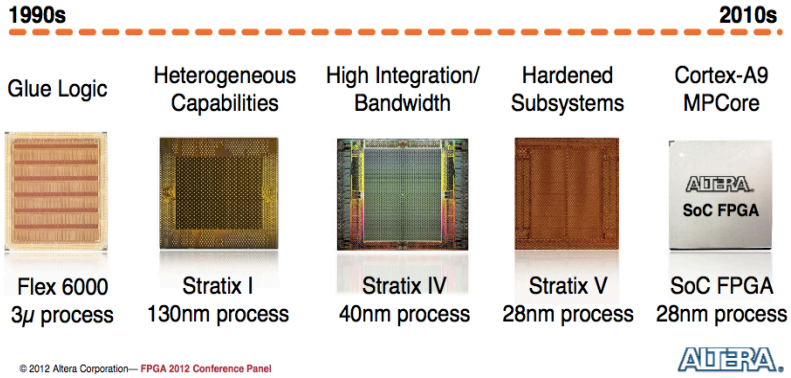




ECHELON: NVIDIA's ExaScale Computing
- 128 SM (1024 core) 160 GFlops each, **20 TFlops aggregated**
- Network: 150 GB/s; DRAM: **1.6TB/s**

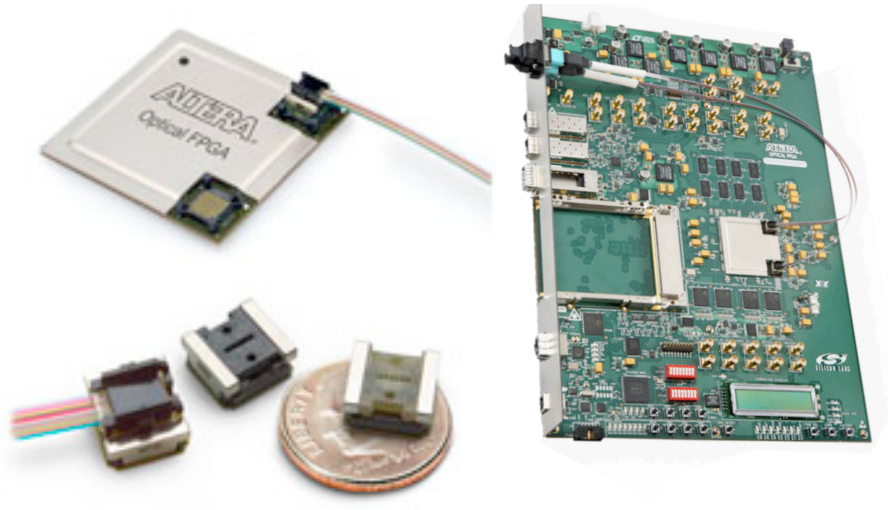
(Multi)PetaFlops scale enabling technologies: FPGA

- High–end FPGA-based systems are the ideal hardware to build custom network
- Most complex electronic devices leveraging on silicon process improving and state-of-the-art technologies



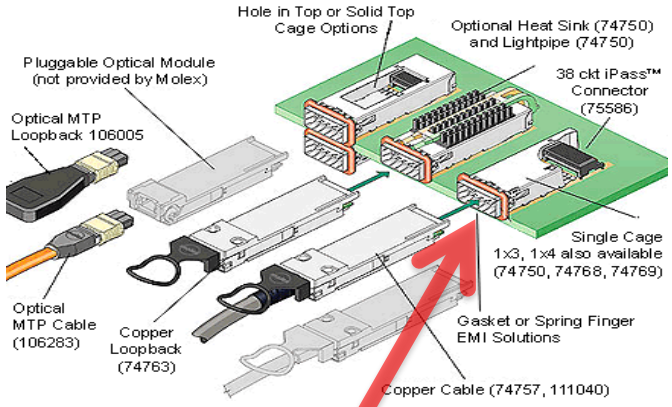
- Current devices (28nm) sport Tflops, (multi)Terabits I/O bandwidth, hardIP uP cores
  - Dual ARM @800MHz (!)
  - O(1) transceivers @28gbps, O(10) transceivers @10-14 gbps, O(100)100 transceivers @1-5 gbps
  - PCIe Gen3/2/1, 10G/40G/100G Ethernet, Serial RapidIO,CPRI (Fixed latency)
- Testbed for future interconnection technologies
  - Avago MicroPod up to 120gb/s full bidir
  - Proof of concept AVAGO + Altera
  - …Waiting for Optical on die

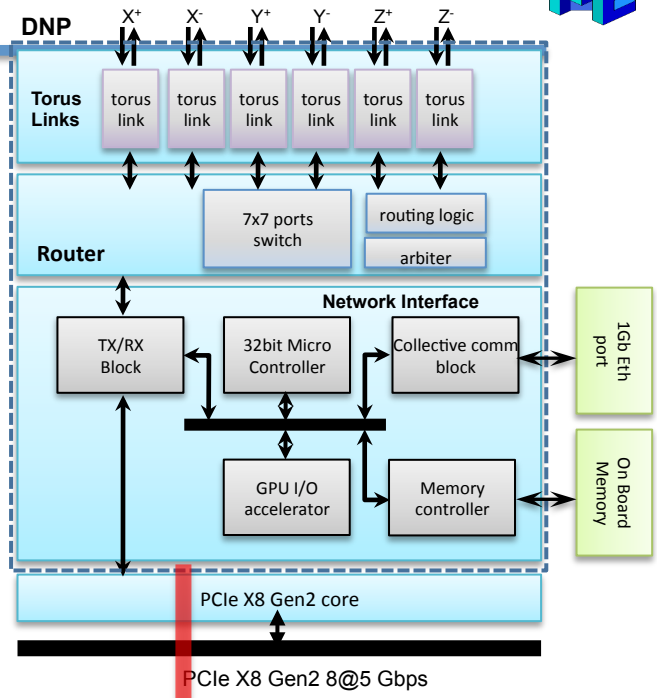
	ALTERA STRATIX V	XILINX VIRTEX 7
Logic Cell (up to)	1.05M	1.9M
Registers (up to)	1.58M	2.4M
Peak transceiver speed (Gbps)	14.1 28.0	13.0 29.0
Device serial bandwidth (Gbps)	936	930
PCIe interface	up to 4 Gen3,x8	up to 4 Gen3,x8
Memory interfaces	up to 7 banks x72 DDR3@800Mhz	DDR3 1.8Gbps
Embedded Memory size	55 Mb	65 Mb
DSP elementary block	4096 (Dual 18x18 Mul + 64bit Acc)	3960 (25x18 Mul + 48bit Acc)
I/O pins	up to 1125 (+ dedicated transceiver pins)	up to 1200 (+ dedicated transceiver pins)
Hardened IP	NIOS (proprietary) ARM A9 MIPS32 (soft core)	Dual ARM A9

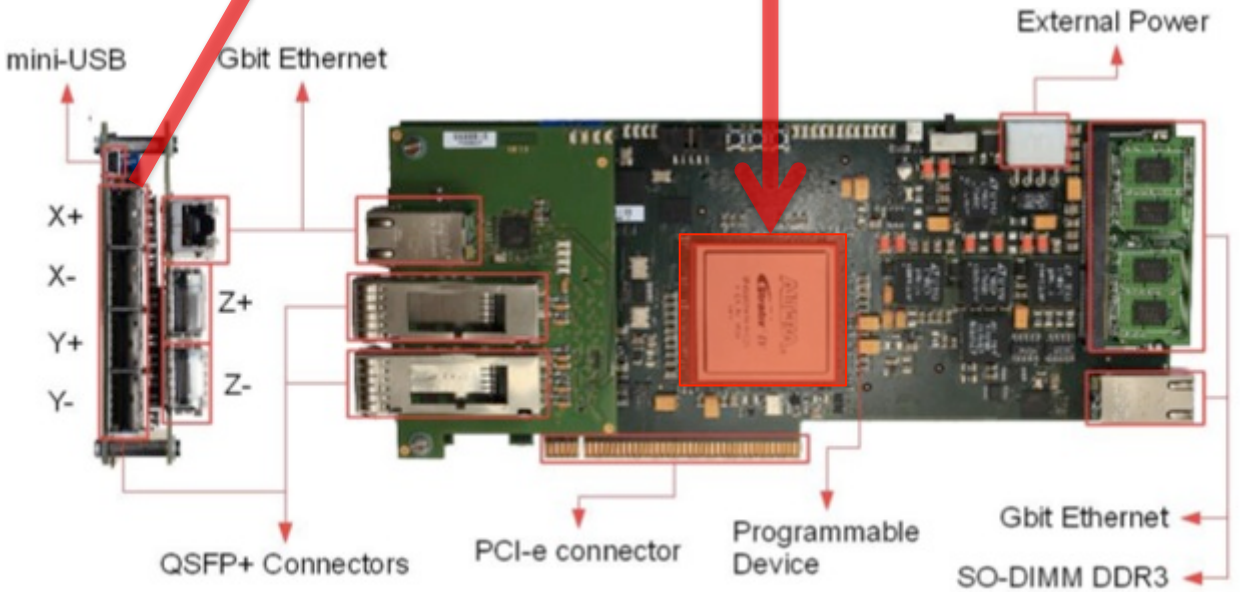


APEnet+ at a glance

- 3D Torus network
  - ideal for large-scale scientific simulations (domain decomposition, stencil computation, …)
  - scalable (APENEt+ today up to 32K nodes)
  - Cost effective: no external switches! 1 card+3 cables
- APEnet based on INFN DNP
  - RDMA: Zero-copy RX & TX !
  - Small latency & high bandwidth
  - GPU clusters features (APEnet+):
    - RDMA support for GPUs! -> no buffer copies between GPU and host.
    - Very good GPU to GPU latency
- APEnet+ card:
  - FPGA based (ALTERA St.IV EP4SGX290)
  - 6 full-bidirectional links up to 68 Gbps raw (~400 Gbps)
  - PCIe X8 Gen2 in X16 slot
    - peak BW 4+4 GB/s
  - Network Processor, off-loading engine integrated in the FPGA
  - Zero-copy RDMA host interface
  - Direct GPU interface
  - Industry standard QSFP+ cabling
    - Copper (passive/active), optical







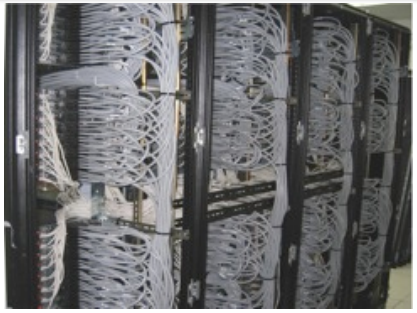
QUonG: GPU+3D Network FPGA-based

QUonG (QUantum chromodynamics ON Gpu) is a comprehensive initiative aiming to deploy an GPU-accelerated HPC hardware platform mainly devoted to theoretical physics computations.

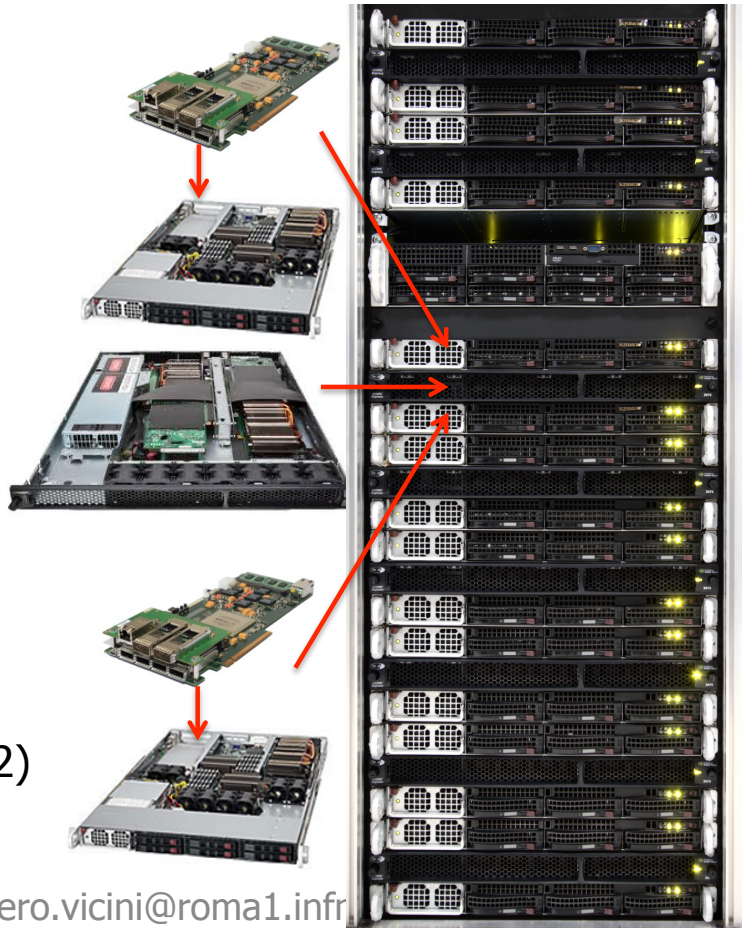
- Heterogeneous cluster: PC mesh accelerated with high-end GPU (Nvidia) and interconnected via 3-D Torus network
- Added value:
  - tight integration between accelerators (GPU) and custom/reconfigurable network (DNP on FPGA) allows latency reduction and computing efficiency gain
  - Huge hardware resources in FPGA to integrate specific computing task accelerators
    - ASIP
    - OpenCL (in the future..)
- Communicating with optimized custom interconnect (APEnet+), with a standard software stack (MPI, OpenMP, …)
- Optionally an augmented programming model (cuOS)
- Community of researchers sharing codes and expertise (LQCD, GWA, Laser-plasma interactions, BioComputing, Complex systems,…)

QUonG assembly

QUonG quick recipe:
1) Take N PC cluster nodes and plug 1 APEnet+ card per node
2) Plug M GPUs per node
3) Connect 6 cables for APEnet+ first neigh. connections
Add service networks (IB,Eth,..) as you like and
serve it to the computational physicists….



- QUonG Hybrid Computing Node:
  - Intel Xeon E5620 double processor
  - 48 GB System Memory
  - 2 S2075 NVIDIA Fermi GPU
  - 1 APEnet+ board
  - 40 Gb/s InfiniBand Host Controller Adapter
- QUonG Elementary Mechanical Unit:
  - 3U Sandwich:
    - 2 Intel dual Xeon servers
    - 4 NVIDIA Tesla M2075 GPU
  - 2 Vertex on the APEnet+ 3d network
- Software Environment
  - CentOS 6.3
  - NVIDIA CUDA 4.2 driver and dev kit
  - OpenMPI and MVAPICH2 MPI available
- Q2 2013: 16 nodes connected by the APEnet+ (4x2x2)
- Addition of few Tflops of Kepler GPUs during 2013

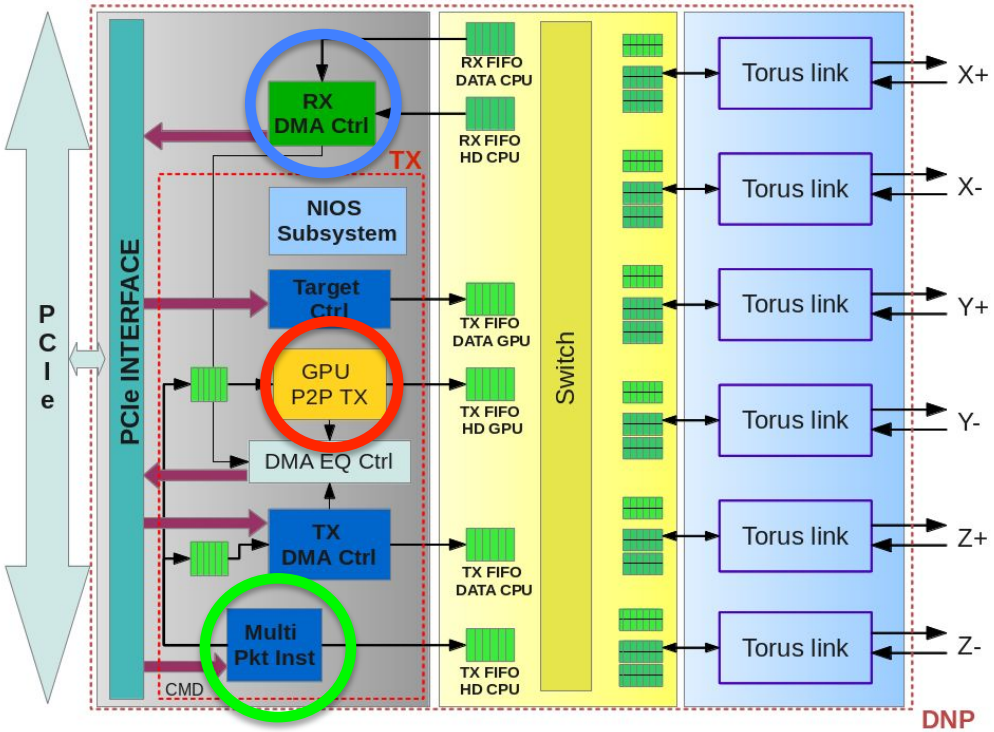


APEnet+ performance improvements: howto…

Iterative process leveraging on architectural refinements of high performance programmable components (FPGA) driven by target applications benchmarking

The rationale is that PCIe "SW driver only" driven transactions are slow (~uS for register access) so HW support is mandatory to squeeze performances…

1) Implementation of GPU Direct RDMA P2P acceleration hardware to lower small packet latency and enhance accelerator (GPGPU) interface performance
2) Integration of specific hardware blocks
   1) "Turbo-Rx" TLB hardware for fast virtual to physics destination address translation: bottleneck for RDMA RX side bottleneck
   2) Multiple DMA engines and "multi-packet instantiator" logic to reduce overhead between subsequent PCIe transfer



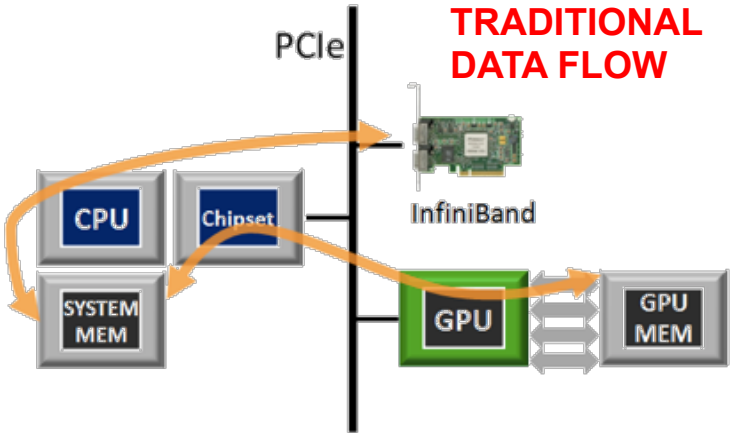
GPU-P2P in APEnet+

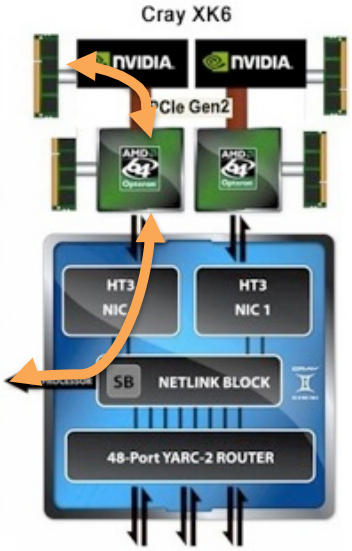
APEnet+ is 1st non-NVidia device to implement Fermi P2P protocol

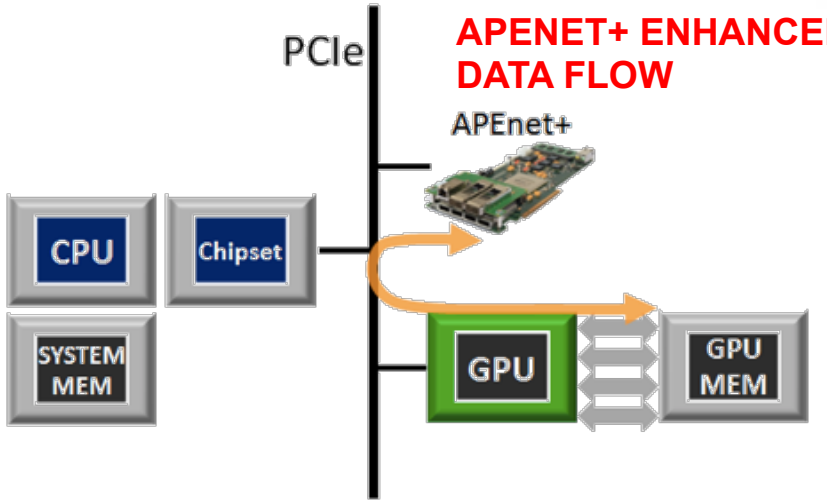
Peer-to-Peer means:
- Data exchange on the PCIe bus
- No bounce buffers on host

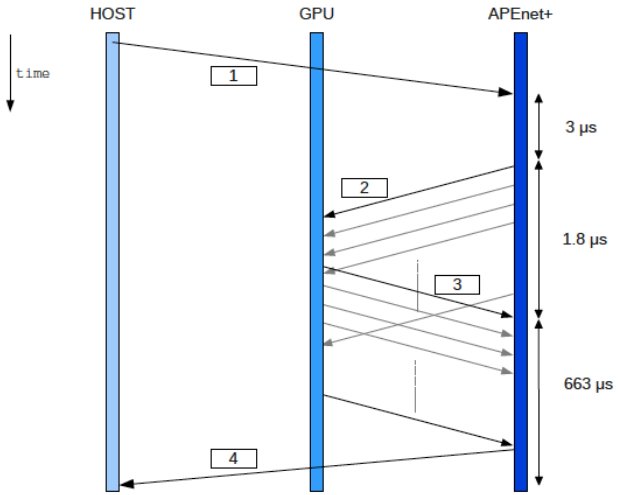
APEnet+ P2P support
- cutting-edge HW/SW technologies developed jointly with Nvidia
  - APEnet+ board acts as a peer
  - APEnet+ board can read/ write "directly" GPU memory

Direct GPU access
- Specialized APEnet+ HW block
- GPU initiated TX
- Latency saver for small size messages

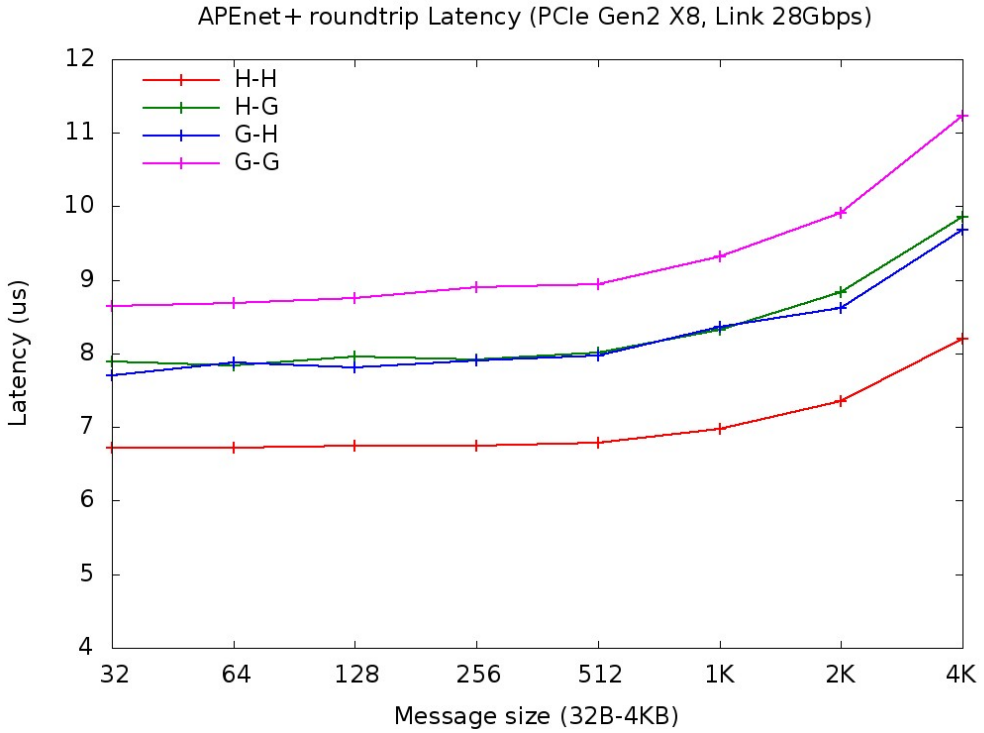


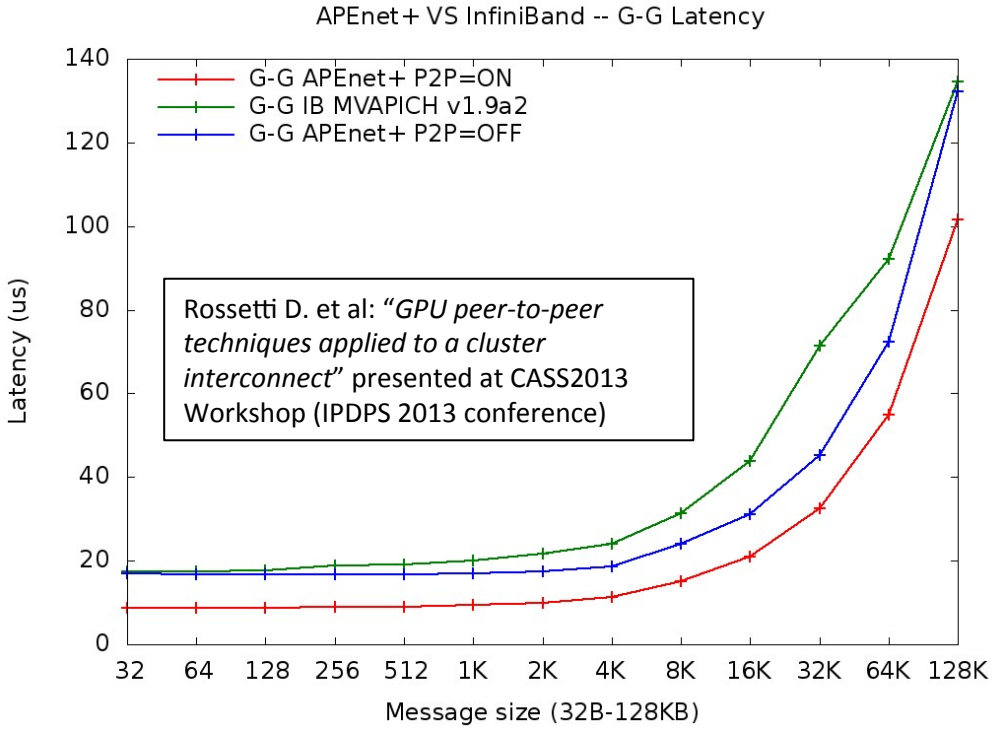






P2P effects on latency

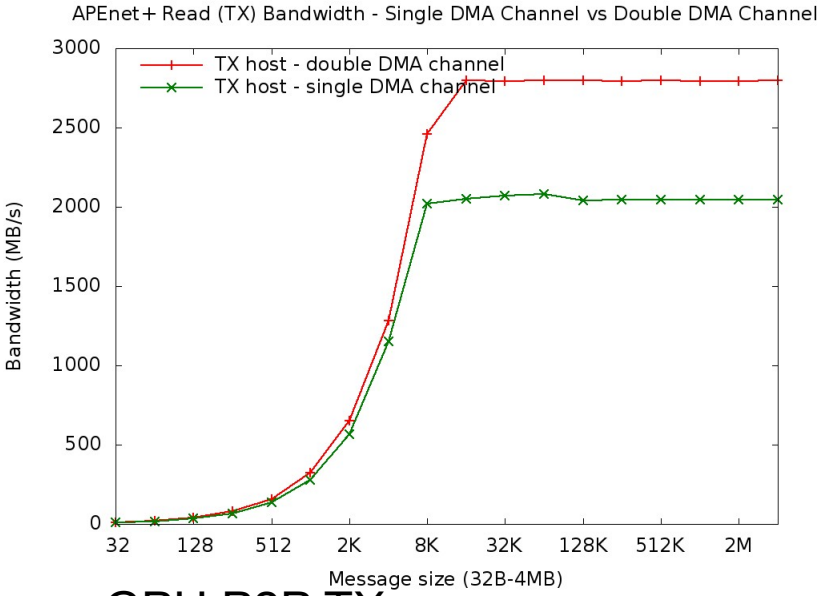




- Latency
  - APEnet+ G-G latency is lower up to 128KB
  - APEnet+ P2P latency ~8.5 us
  - APEnet+ staging latency ~16.8 us
  - MVAPICH/IB latency ~17.4 us
- P2P=OFF
  - cudaMemcpyD2H/H2D() on host bounce buffers
  - Buffers pinned with cuMemHostRegister
  - cuMemcpy() ~10 us

Test	Bandwidth	GPU/method	Nios II active tasks
Host mem read	2.4 GB/s		none
GPU mem read	1.5 GB/s	Fermi/P2P	GPU_P2P_TX
GPU mem read	150 MB/s	Fermi/BAR1	GPU_P2P_TX
GPU mem read	1.6 GB/s	Kepler/P2P	GPU_P2P_TX
GPU mem read	1.6 GB/s	Kepler/BAR1	GPU_P2P_TX
GPU-to-GPU loop-back	1.1 GB/s	Fermi/P2P	GPU_P2P_TX + RX
Host-to-Host loop-back	1.2 GB/s		RX

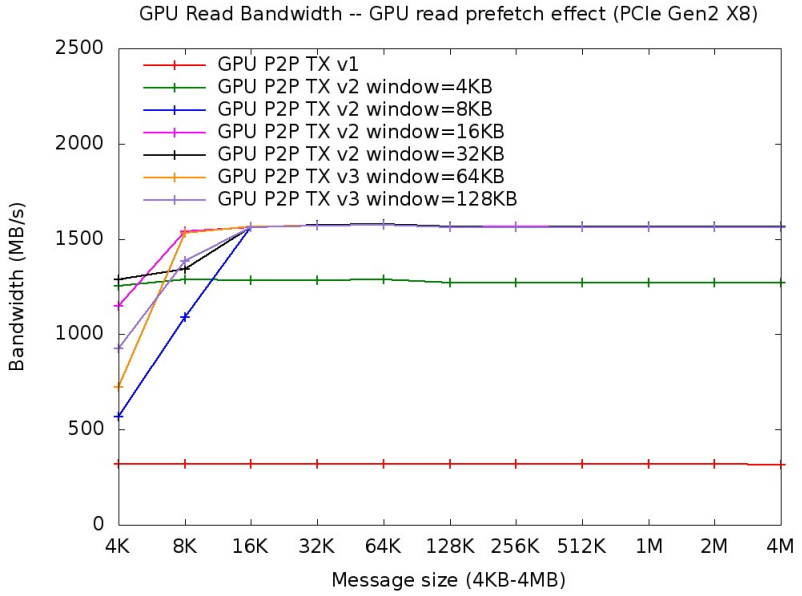
APEnet+ TX bandwidth



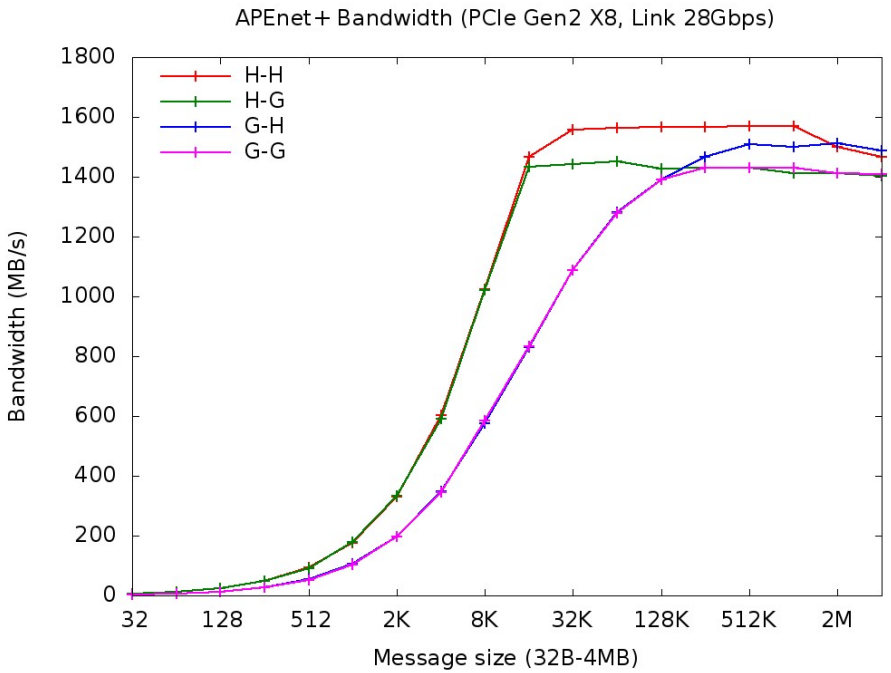
- HOST TX:
  - Completely handled by kernel driver
  - 2 DMA channels implemented
  - PCI data read request pipelined
  - Latency between 2 consecutive Host Memory Read Request
    - Single DMA ~900ns
    - Double DMA ~50ns
  - BW ~2.8GB/s (40% improvement)

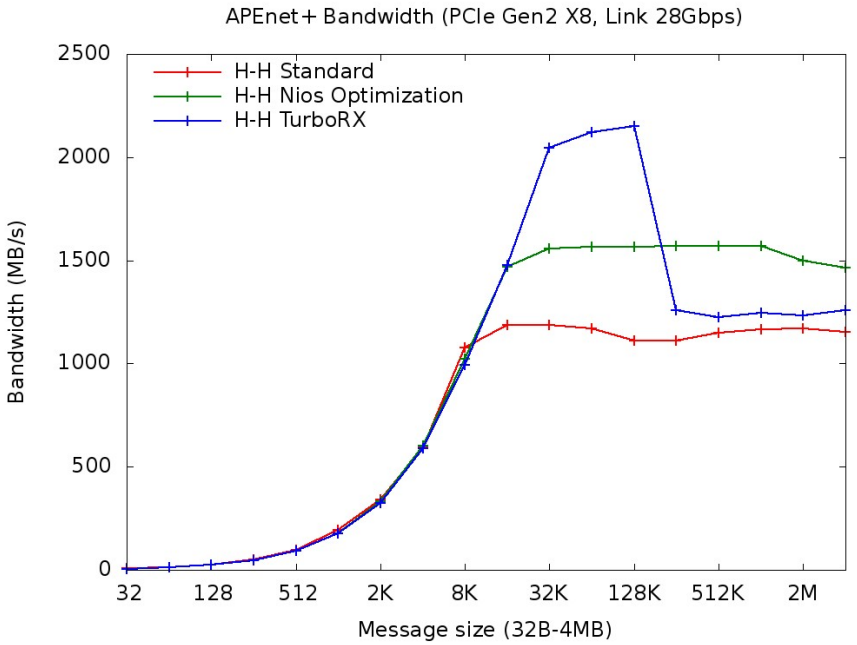
Biagioni A.. et al: "*Second Report on Innovations on HW Intellectual Properties*" EURETILE FP7 project euretile.roma1.infn.it

- GPU P2P TX:
  - v1 ~ 0.3GB/s
    - Single packet request support
    - No data prefetch (GPU mem is not latency opt)
  - v2 ~ 1.4GB/s
    - read request acceleration (a read req every 80ns)
    - limited pre-fetching window to 32KB (hiding response latency)
  - v3 ~ 1.6GB/s
    - Unlimited pre-fetching window
    - Higher bandwidth in loopback test



Bandwidth optimization for large packets: TLB hardware





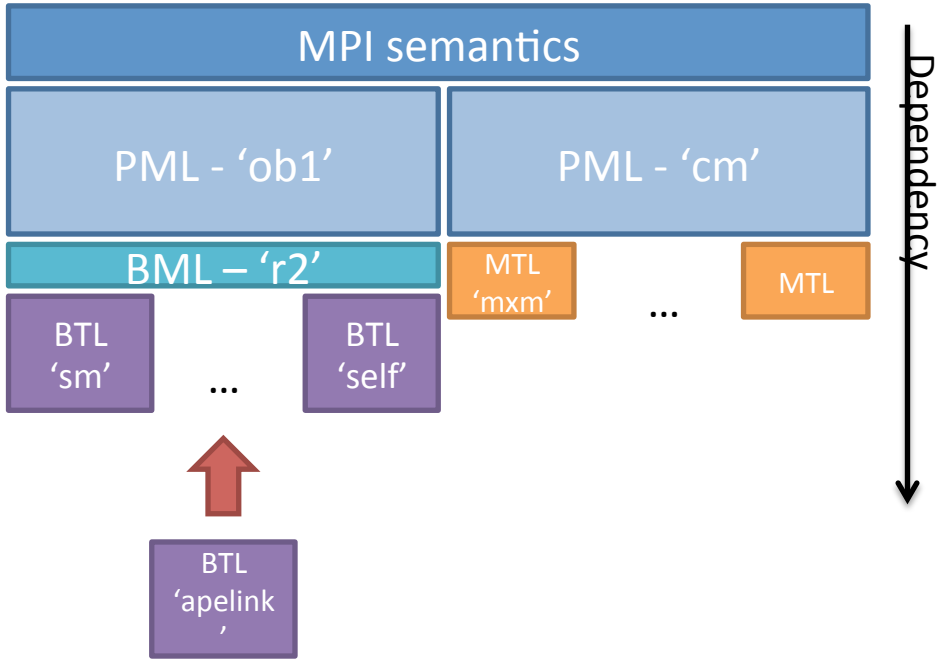
- Host RX ~1.6 GB/s
- GPU RX ~1.4 GB/s (switching GPU P2P window before writing)
- Limited by RX LOGIC RDMA Virtual-to-Physical (V2P) Translation, most demanding task.
- When handled by Nios II:
  - Firmware not optimized: ~1.2GB/s, Latency ~2.2us
  - Nios Firmware Optimization ~1.6GB/s, ~1.6us
- When handled by HW (TurboRX):
  - First implementation. Host RX only! >2.2GB/s
  - At hardware level, TLB hardware from 3000ns -> 124 ns (x30) for 128 KB buffer size. Work in progress to push up to 1-4 MB packets

Ammendola et al: "*VIRTUAL TO PHYSICAL ADDRESS TRANSLATION FOR AN FPGA-BASED INTERCONNECT WITH HOST AND GPU RDMA CAPABILITIES*." Submitted to FPL2013 conf.

SW stack and GPU optimizations

Current APEnet programming model

- native RDMA API:
  - RDMA buffer registration: pinning and posting combined
  - single message transmission async queue
  - async delivery of completion events (both TX and RX)
- MPI for APEnet+ OpenMPI based
  - The Byte Transfer Layer framework provides a set of components for raw data transfer for send/receive and RDMA based inteconnects.
  - The apelink BTL uses the RDMA API to program the APEnet+ device
  - early prototype….



From, here where?

1) Can we do something more to increase application performances on hybrid CPU+GPU systems i.e. can our target application benefits from a more powerful torus network card?

2) What about our commercial "competitors"?

A simple bottleneck analysis

LQCD as an example

- Slightly modified performance model taken from Babich (STRONGnet 2010), from Gottlieb via Holmgren
- Balance condition: perfect overlap between computing time and communication time

GPU lattice	GPUs per node	Node lattice	Global lattice	# of nodes	# of GPUs	Req BW GB/2
16^3*16	2	16^3*32	64^3*128	256	512	4.3
16^3*32	2	16^3*64	64^3*128	128	256	4.0
32^3*32	2	32^3*64	64^3*128	16	32	2.1
16^3*32	4	16^3*128	64^3*128	64	256	7.4
32^3*32	4	32^3*128	64^3*128	8	32	3.7

Facts:

1) Current PCIe implementation limits the performance and scaling
   - apeNET+ bandwidth (PCI Gen2 x8 / 34 Gbps per link) allows "strong scaling" up to few tens of GPU
2) Tight time budget i.e. specific HW GPU-APEnet+ optimizations are needed to reduce transfer latency and overheads

Network card competitors…

Mellanox announced full support to RDMA GPU Direct ("Bar1 access")

- At GTC (GPU Tech. Conf, Mar 2013) presented a preliminary set of latency measures for Connect3-X InfiniBand adapter supporting GPU Direct protocol
- GPU Direct RDMA enabled board is available now to "selected customers"; GA in few months from now…

So, APEnet game over?

Not yet, if

- I/O interfaces performance remain comparable (better?)
- coupling of GPU Direct RDMA, High speed low latency network and (huge) FPGA resources is fully exploited
  - Hardware system specialization driven by application requirements
  - Adding "processing on network", changing routing functions, changing physical network topology, implementing in HW exotic "tasks", hw support to enhance system fault tolerance,…

APEnet+ customization: NaNet

NaNet: APEnet + NA62 cern Experiment

GPU L0 TRIGGER for HEP Experiments

Implement a RO Board-L0 GPU link with:

- Sustained Bandwidth > 600 MB/s, (RO board output on GbE links)
- Small and stable latency

Problem: lower communication latency and its fluctuations. How?

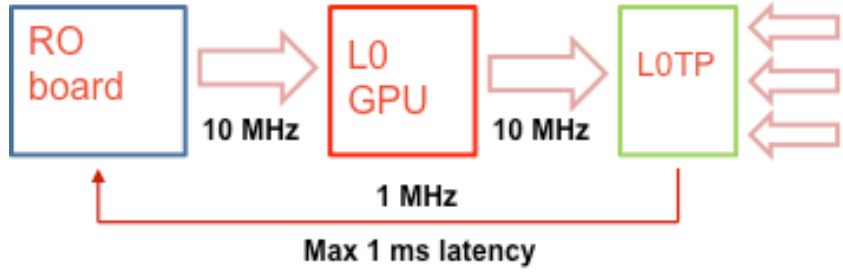
- Offloading the CPU from network stack protocol management.
- Injecting directly data from the NIC into the GPU(s) memory.

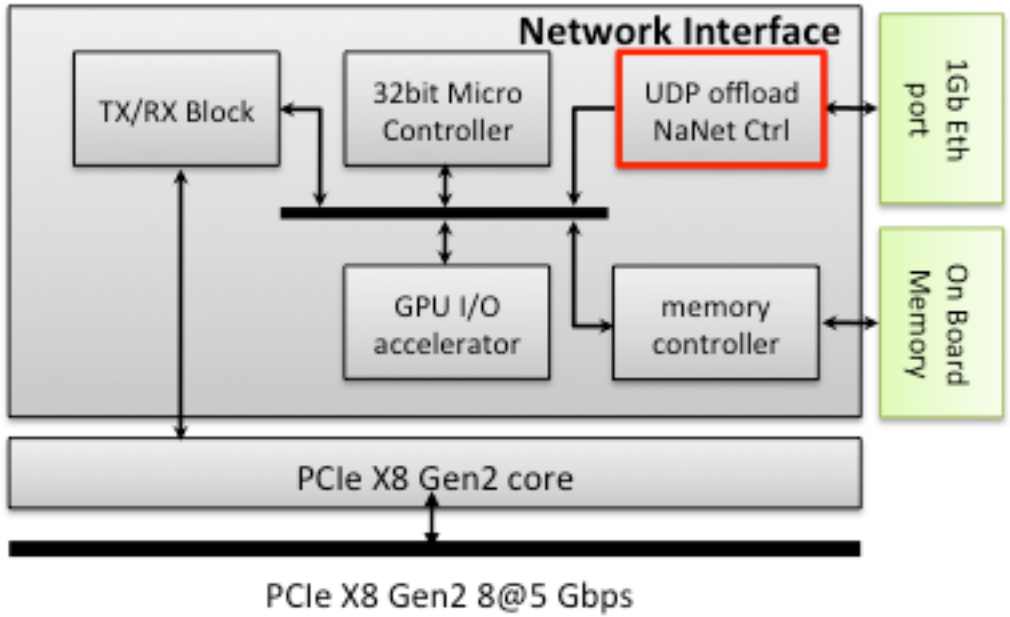
NaNet solution:

- APEnet+ FPGA-based NIC with an additional network stack protocol management offloading engine to the logic (UDP Offloading Engine).

Lonardo A. "*Building a Low-latency, Real-time, GPU-based Stream Processing System*" GTC2013 Conference

-→ More details in "*GPU for realtime processing in HEP experiments*" in this conference





APEnet: 2013-2014 (main) activities

- APEnet++: adoption of 28nm FPGA
  - PCIe Gen2 -> Gen3 (x2 data rate)
  - Torus links speed-up: from current 8.5 Gb/s to 14.5 Gb/s (x2)
  - Explore the PCI Gen3 x16 (with PLX technology bridge)
  - Explore the use of the embedded dual-core ARM processor to increase performance (Virt2Phys translation, P2P support,….)
- Explore V5 porting on EUROTECH Tigon systems
- Push on NVIDIA joint activities
  - Further optimization of P2P GPU-APEnet+ and Kepler, Maxwell,…. integration
- APEnet+ customization: add specific I/O interface and accelerators in FPGA
  - Low Level Trigger GPU-based for HEP collider (NA62, Atlas,…)
  - Distributed read-out for KM3 Neutrino Telescope (under evaluation)
  - Low latency coupling of read-out system and GPU computing for E-ELT (European Large Telescope) and for X-Ray microscopes imaging (LBNL)
  - Brain simulation: dedicated network for high speed connectoma simulation (DPSNN model)

Conclusions

APEnet+ and QUonG

- APEnet+ V4 in "massive" construction phase.
  - Demonstrated advantages of HW GPU Direct RDMA mechanism introduction in hybrid systems (CPU+GPU)
  - Assembled a medium-size prototype (32 Tflops) almost (…) "ready to use"
  - Software optimization (MPI, RDMA API) in progress

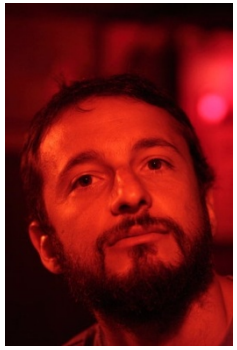
Roadmap 2013-2014

- APEnet+ Versione 5 (V5) based on 28nm FPGA: more room for performance improvements
  - PCIe Gen2->Gen3, Torus link speed enhancements, HardIP ARM coupling to APEnet exploration

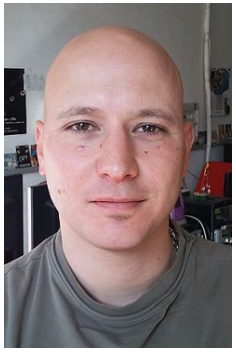
Also, fast moving towards new scientific environments

- High-Low level trigger of future HEP experiments
- APEnet+ as low latency GPU interface to read-out system
- Network specialization for computing platform for not "APE traditional" fields: complex systems, Brain Simulation, molecular dynamics…

Thank you! Questions or comments?


Roberto Ammendola


Andrea Biagioni


Ottorino Frezza


Francesca Lo Cicero


Alessandro Lonardo


Pier Stanislao Paolucci


Davide Rossetti


Francesco Simula


Laura Tosoratto


Piero Vicini

Partially supported by EU FP7 grant agreement EURETILE n. 247846

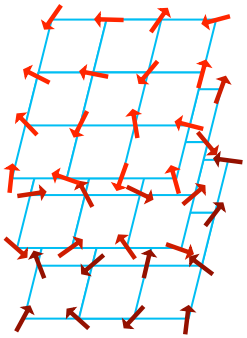
BACKUP SLIDES

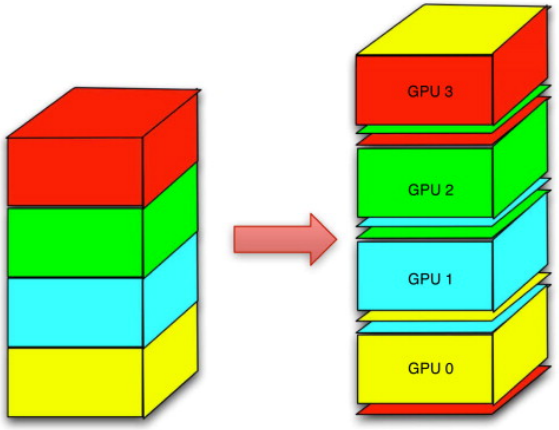
Applications highlights: HSG

- 3D Lattice of 3D versors, with randomly distributed, first-neighbours couplings:

$$H = -\sum_{i \neq j} J_{ij} \hat{\sigma}_i \cdot \hat{\sigma}_j$$

- Lattice is split along 1 dimension, with every GPU taking its slice. Boundaries must be kept in synch!

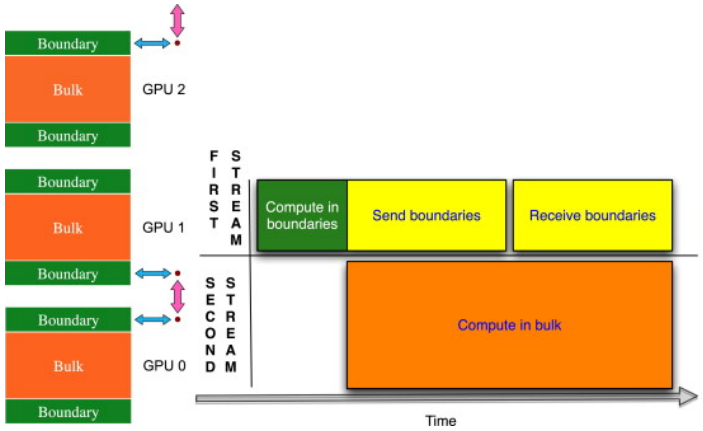


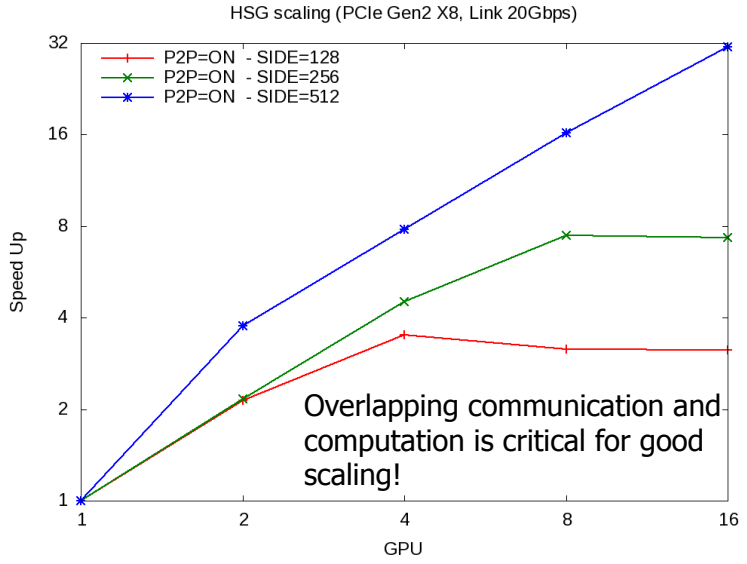


Time	Cluster I			Cluster II OMPI (PCIe X8)	Cluster I OMPI (PCIe X4)
	P2P=ON	P2P=RX	P2P=OFF		
T_{tot}	416	416	416	416	416
$T_{bnd} + T_{net}$	108	97	122	108	108
T_{net}	97	91	114	101	101

Rossetti D. et al: "*GPU peer-to-peer techniques applied to a cluster interconnect*" presented at CASS2013 Workshop (IPDPS 2013 conference)

P2P allows for 10-20% latency decreasing





Application highlights: Graph500*

- BFS on multi-GPUs
- Very preliminary results

on a 2x2 APEnet+ cluster

TABLE I. Traversed Edges Per Second, Strong Scaling, $|V| = 2^{20}$

NP	INFINIBAND	APENET
1	6.25389e+07	6.24038e+07
2	7.8924e+07	1.01101e+08
4	8.20081e+07	1.26543e+08

TABLE II. Traversed Edges Per Second, Weak Scaling, $|V| = 2^{SCALE}$

NP	SCALE	INFINIBAND	APENET
1	19	5.60594e+07	5.9808e+07
2	20	7.8924e+07	1.01101e+08
4	21	1.08637e+08	1.46482e+08

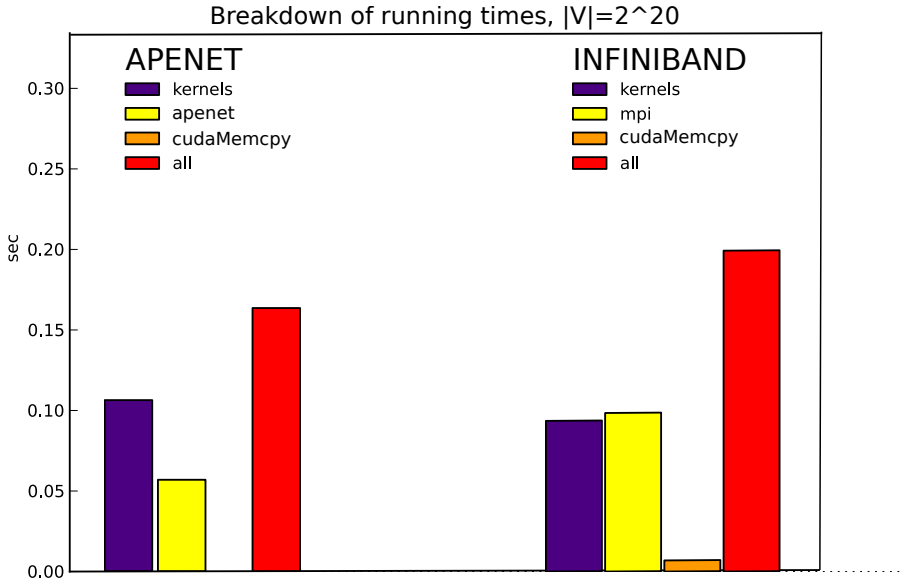


Fig. 5. Breakdown of the execution time on one out of four tasks for both APEnet and Infiniband.

*based on code from E.Mastrostefano, M.Bernaschi *"Efficient breadth first search on multi-GPU system"* submitted to Journal of Parallel and Distributed Computing.

Application highlights: event reconstruction in HEP exps.

GPUs for real time event selections?

GPU

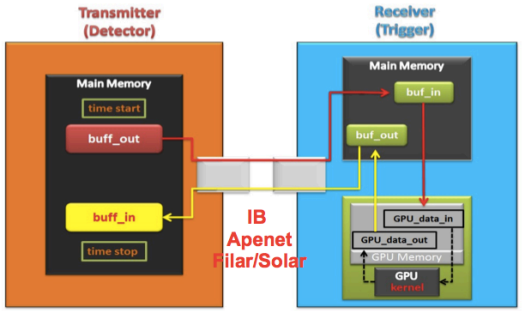
- A lot of computing power for highly parallelizable tasks;
- High level programming (CUDA, OpenCL);
- Commercial device → less expensive than dedicated hardware, continuous improvement of performance;
- NOT designed for low latency response

Real time events selection

- It is usually based on algorithms well suited for parallelization;
- A trigger system needs to be flexible, to be adapted to experiments changing conditions;
- It needs low latencies.

2

Data flow and measurements



Multiple loops, for each: ΔT = *Time stop -Time start*
Time measured in the transmitter using the time stamp counter register.

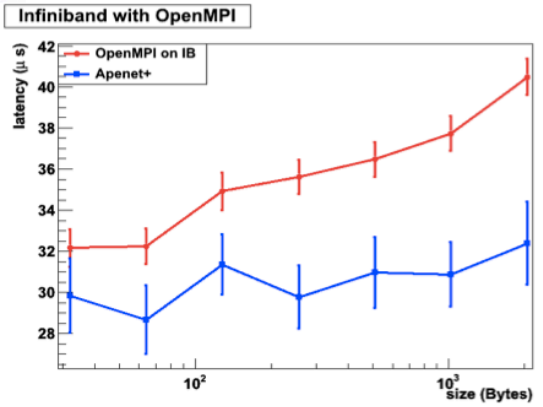
3 set of measurements:
1) data transfer only (N words IN → N words out)
2) data transfer + copy on GPU (N words IN → copy on GPU → N words out)
3) data transfer + copy on GPU + kernel (N words IN → copy on GPU → M words out)

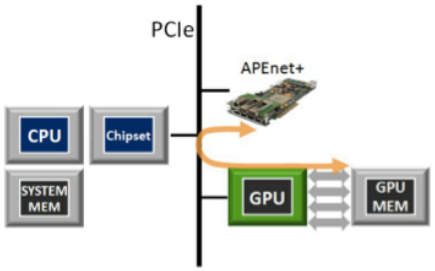
6

Amerio et al "*Applications of GPUs to Online Track Reconstruction in HEP Experiments*" NSS 2012

Data transfer + copy to the GPU

Apenet+





From 32 B to 2 kB
- IB: from 33 to 40 µs
- Apenet+: from 30 to 33 µs

Significant latency reduction with Apenet+ (direct GPU memory access)

12

LQCD on QUonG

C. Bonati (Pisa), G. Cossu (KEK), M. D'Elia (Pisa), A. Di Giacomo (Pisa) P. Incardona (Pisa)

First step (completed): single GPU implementation (see Comp. Phys. Comm. **183**, 853 (2012), arXiv:1106.5673).

We have:

- Staggered fermion action is already used in production runs on the study of the QCD phase diagram (e.g. Phys. Rev. D **83**, 054505 (2011), arXiv:1011.4515).
- openCL support for Nvidia and AMD GPUs (for Cuda/openCL comparison see arXiv:1106.5673).

Lattice $4 \times L^3$

L	16	32	48
Opteron (1 core)	65	75	140
Xeon (1 core)	30	40	50
apeNEXT crate	$\sim$2		

Table: NVIDIA C2050 time gains over CPU and apeNEXT (Opteron(tm) 2382 and Xeon(R) X5560)

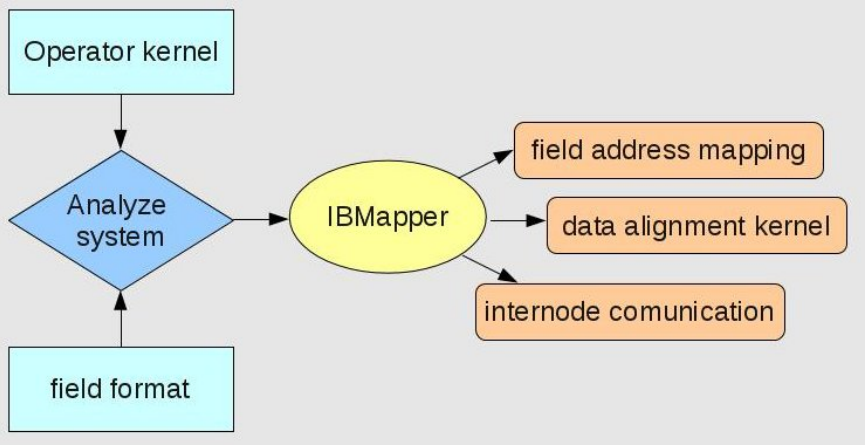
LQCD on GPU (2)

Multi-GPUs implementation

Key points:

- in-kernel fast field address mapping
- internode and P2P comunication overlapped with computation
- data alignment

C. Bonati (Pisa), G. Cossu (KEK), M. D'Elia (Pisa), A. Di Giacomo (Pisa) P. Incardona (Pisa)



multi-GPU scaling:

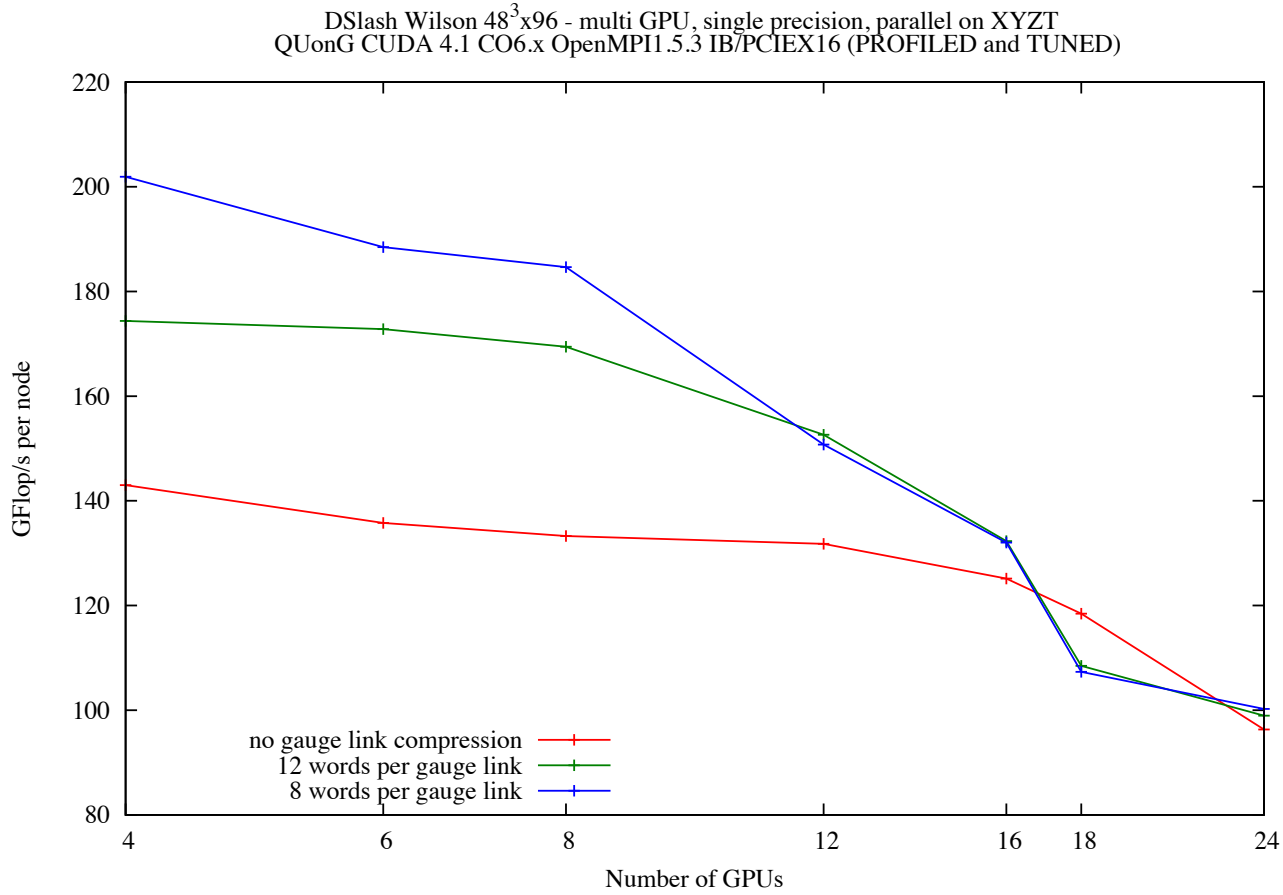
Lattice $4 \times L^3$

L	32	48	64
2 C2050	1.60	1.89	1.95
4 C2050 (infiniband)	2.45	2.98	3.34

Current development:

- implementation of improved discretization for fermions.
- integration with the APE-net project

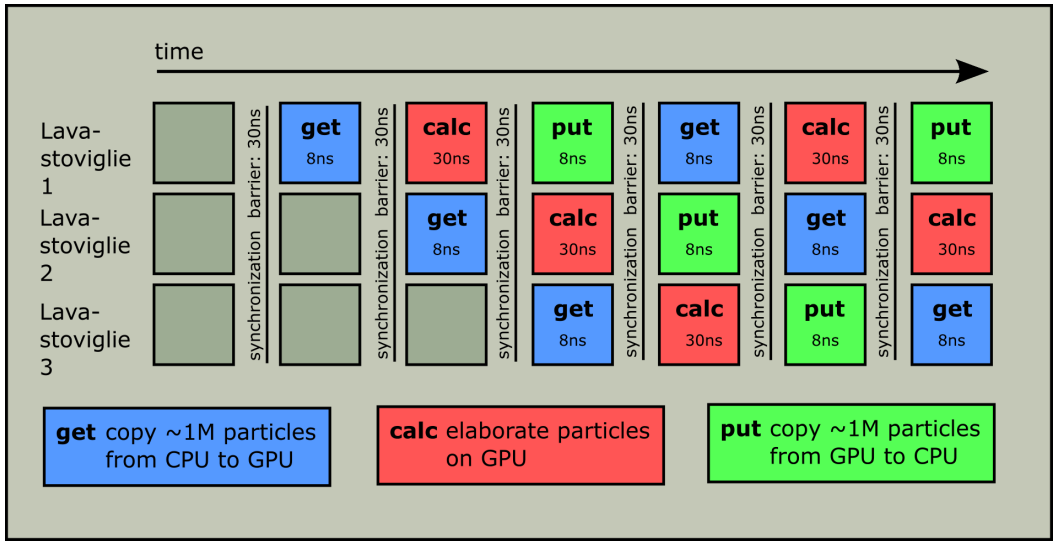
LQCD on GPU (2)

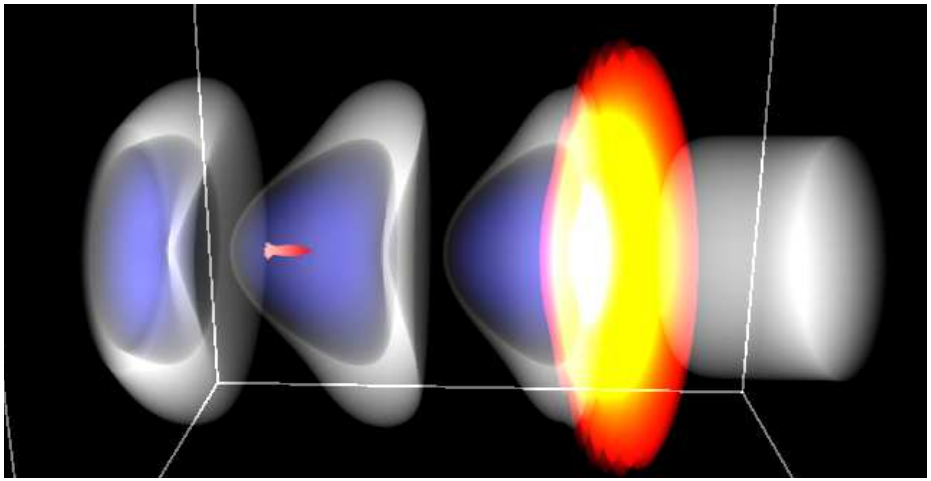

DSlash Wilson 48^3x96 - multi GPU, single precision, parallel on XYZT
QUonG CUDA 4.1 CO6.x OpenMPI1.5.3 IB/PCIEX16 (PROFILED and TUNED)
GFlop/s per node
Number of GPUs
220
200
180
160
140
120
100
80
4
6
8
12
16
18
24
no gauge link compression
12 words per gauge link
8 words per gauge link

PIC on QUonG

- PIC (Particle In Cell) code for laser-plasma acceleration
  - F. Rossi, P. Londrillo, A. Sgattoni, S. Sinigardi, G. Turchetti, *"Robust algorithms for current deposition and efficient memory usage in a GPU Particle In Cell code"* 15th Advanced Accelerator Concepts Workshop (AAC 2012)

4 Jasmine: a flexible, hybrid (CPU+GPU), PIC framework



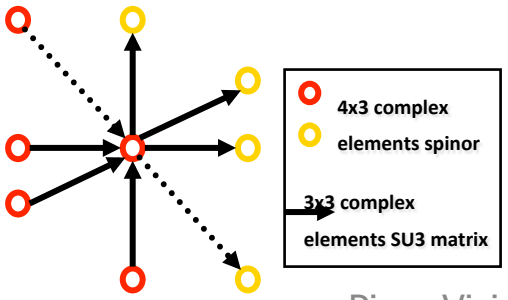


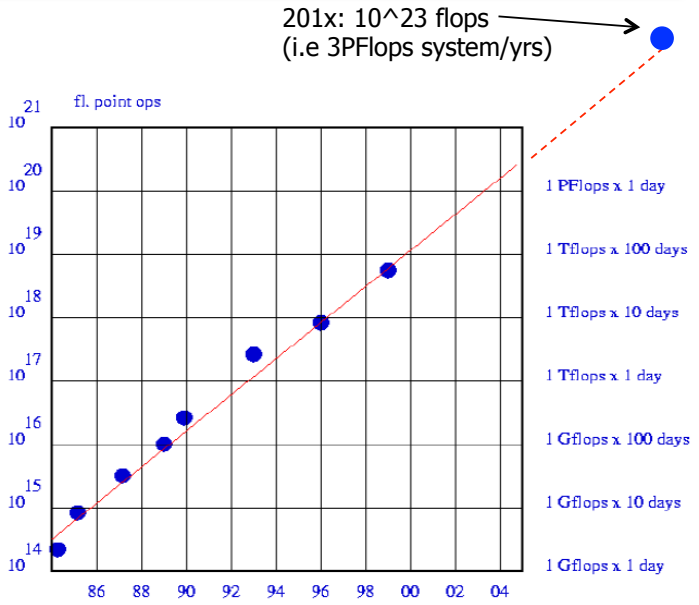
The (original) scientific case: LQCD (Lattice Quantum Chromo Dynamics)

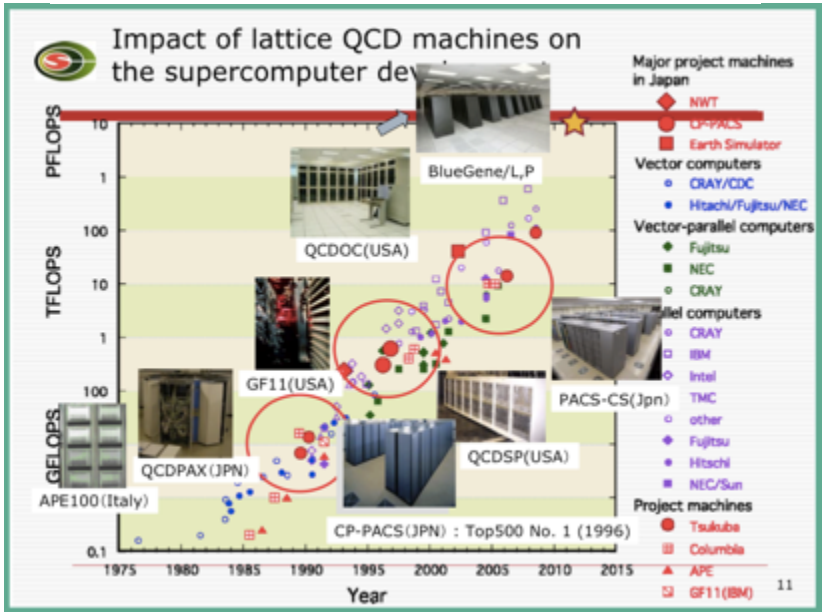
- The physical problem: QCD phase diagram and confinement
  - Low energy QCD and confinement are intrinsically nonperturbative phenomena.
  - In Lattice QCD a finite lattice is introduced as a nonperturbative gauge invariant regulator and observables are calculated by using Monte Carlo simulations.

- The numerical problem
  - In a LQCD simulation a lot of L × L linear systems have to be solved, with $L \sim 10^5 \div 10^6$ (!)
    - Need for dedicated high performances supercomputer (APE, Blue-Gene,…)

- Algorithmic peculiarities
  - Homogeneity
    - Ideally suited for SIMD parallelization on many core architectures
  - Locality
    - Parallel machine efficent scaling through the adoption of low latency, point-to-point 3D Torus network







Piero Vicini – INFN Roma (piero.vicini@roma1.infn.it)