NaNet: a Low Latency, Real-Time, Multi-Standard Network Interface Card with GPUDirect Features

Piero Vicini (INFN-Rome)
On behalf of the NaNet collaboration

19th Real-Time Conference (RT2014)
GPU in HEP *low level* Trigger

A first case study: NA62 RICH detector, analysis of data-transfer latency and its jitter

Our solution: NaNet hardware

Preliminary results and future development

NaNet in KM3Net-IT experiment: fixed latency requirements

Conclusion and future work
• **General Purpose Graphic Processing Unit:** well established architecture in HPC arena
• Impressive peak performance (N*TFlops per chip)
  • Many-Core (~1000!) SIMD-like architecture
  • High local memory bandwidth: 140 GB/s -> 500 GB/s
  • “Green” and cost effective
  • Dedicated high level programming language (CUDA)
• Aggressive but feasible roadmap: much room for performance scaling
  • New features added generation by generation….
    • NVlink proprietary channel for GPU-to-GPU communication
    • 3D stacked memory to improve local memory capacity

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
Many activities to explore the use of GPU for HEP applications
- NA62, LHCb, ATLAS,…

The “canonical” way: High Level Trigger and offline data analysis
- no real-time strict constraints

More challenging: GPU in HEP Low level Trigger. Two main questions:
- Computing power: Are the GPUs fast enough to take trigger decision at tens of MHz events rate?
- Latency: Are the GPUs latency per event small enough to cope with the tiny latency of low level trigger systems? Is this small latency stable enough for usage in synchronous trigger systems?
Using GPUs in the NA62 L0 Trigger

L0 trigger:
- FPGA custom-design
- 10 MHz to 1 MHz
- Max latency 1 ms

Replace custom hardware with a GPU-based system performing the same task but:
- Programmable
- Upgradable
- Scalable
- Cost effective
- Exploit GPUs computing power to implement more selective trigger algorithms.
Using GPUs in the NA62 L0 Trigger: the RICH Detector Case Study

- Light focused by two mirrors on the two spots.
- 100 ps time resolution.
- 10 MHz event rate.
- ~18 hits per ring on average.
- 70 byte per event (max 32 hits per event).
- Computing on GPU: rough detection of particle speed (radius) and direction (centre) through matching of partial circular hit patterns.
- Network Protocol: UDP on GbE channel(s).
- System Throughput: 10 M events/s
  - Network bandwidth between 400 MB/s and 700 MB/s, depending on data protocol choice (hit finetime data).
- System response latency < 1 ms
  - determined by the size of Readout Board memory buffer storing event data to be passed to higher trigger levels.
NA62 readout board: TEL62
See PS 3-45: “The TEL62: a Real-Time Board ...”

- $\text{lat}_{\text{L0TP-GPU}} = \text{lat}_{\text{comp}} + \text{lat}_{\text{comm}}$
  - Estimate the two components and their fluctuations.
- Perform estimation on a **single** GbE channel
  - **Estrapolate** data for the design of the “full bandwidth” system.
- **lat\textsubscript{comp}**: time needed to perform rings pattern-matching on the GPU with input and output data on device memory using the MATH algorithm.

- Test setup: Supermicro X8DTG-DF motherboard (Intel Tylersburg chipset), dual Intel Xeon E5620, Nvidia M2070, Intel 82576 Gigabit Network Connection, kernel 2.6.32-358.6.2.el6.x86_64, CUDA 4.2, Nvidia driver 325.15.
\( \text{lat}_{\text{comm}} \) : time needed to receive input event data from GbE NIC to GPU memory.

- 20 events data (1404 byte) sent from Readout board to the GbE NIC are stored in a receiving host kernel buffer.
- \( T=0 \) start of send operation on the Readout Board.

- \( \text{Lat}_{\text{comm}} = 104 \text{ us} \approx 3.5 \times \text{Lat}_{\text{comp}} \)

- i.e. total latency is dominated by communication latency:
- Fluctuations on the GbE component of \( \text{lat}_{\text{comm}} \) may hinder the real-time requisite, even at low events count buffer sizes.

sockperf: **Summary**: Latency is 99.129 usec
sockperf: Total 100816 observations; each percentile contains 1008.16 observations

sockperf: \( \text{<MAX> observation} = 657.743 \)
sockperf: \( \text{percentile 99.99} = 474.758 \)
sockperf: \( \text{percentile 99.90} = 201.321 \)
sockperf: \( \text{percentile 99.50} = 163.819 \)
sockperf: \( \text{percentile 99.00} = 149.694 \)
sockperf: \( \text{percentile 95.00} = 116.730 \)
sockperf: \( \text{percentile 90.00} = 105.027 \)
sockperf: \( \text{percentile 75.00} = 97.578 \)
sockperf: \( \text{percentile 50.00} = 96.023 \)
sockperf: \( \text{percentile 25.00} = 95.775 \)
sockperf: \( \text{<MIN> observation} = 64.141 \)
NaNet-1: a custom 1GbE NIC with GPUDirect RDMA capability

■ Challenge:
  ❑ lower system communication latency and its fluctuations.

■ How?
  1. Injecting directly data from the NIC into the GPU memory with no intermediate buffering.
  2. Offloading the CPU from network stack protocol management, avoiding OS jitter effects.

■ NaNet solution:
  ❑ Re-use the APEnet+ design, implementing GPUDirect RDMA.
  ❑ Add a network stack protocol management offloading engine to the logic (UDP Offloading Engine).
APEnet+: a 3D NIC for HPC with GPUdirect RDMA capability

**APEnet+ Card:**
- FPGA based (ALTERA EP4SGX290)
- PCI Express x16 slot, x8 gen2 signaling
- Fully bidir 3D torus links, 34 Gbps/channel

**APEnet+ Logic:**
- Network Interface
  - NIOS II 32 bit uP
  - RDMA engine
  - Virtual Memory Management support
  - GPU I/O accelerator.

**GPUDirect P2P**
- PCIe P2P protocol between Nvidia Fermi/Kepler devices and APEnet+
- GPUDirect P2P allows direct data exchange on the PCIe bus with no CPU involvement
  - *Latency reduction for small messages*
NaNet

- Multistandard support of I/O interface
  - Off-the-Shelf: 1-GbE, 10-GbE (work in progress).
  - Custom: APElink (34gbps/QSFP), KM3link (work in progress).
- GPU I/O accelerator
  - GPUDirect v2 (custom NVP2P)
  - GPUDirect RDMA
- CPU offload – NO OS Jitter
  - Transport layer off-loader module
  - Virtual Memory Management
- NaNet Controller protocol adaptation between on-board and off-board protocol
- Altera NIOS II microcontroller
- Custom Logic (application-specific task)
- DDR3 memory controller
- PCIe Gen2 X8 host interface

I/O Interface

- Physical Link Coding
- Protocol Manager
- NaNet Controller
- APElink I/O IF
- Word Stuff
- Link Ctrl
- APElink
- TSE MAC
- UDP
- NaNet Ctrl
- Det. Latency
- TDM
- NaNet Ctrl
- 10G BASE-R
- KM3link
- UDP
- NaNet Ctrl
- 10G BASE-R
- NaNet Ctrl
- NaNet Ctrl

Router

- TX/RX Block
- 32bit Micro Controller
- Custom Logic
- Network Interface
- On Board Memory
- GPU I/O accelerator
- memory controller
- PCIe X8 Gen2 core

I/O Interface

- Multistandard support of I/O interface
  - Off-the-Shelf: 1-GbE, 10-GbE (work in progress).
  - Custom: APElink (34gbps/QSFP), KM3link (work in progress).
- GPU I/O accelerator
  - GPUDirect v2 (custom NVP2P)
  - GPUDirect RDMA
- CPU offload – NO OS Jitter
  - Transport layer off-loader module
  - Virtual Memory Management
- NaNet Controller protocol adaptation between on-board and off-board protocol
- Altera NIOS II microcontroller
- Custom Logic (application-specific task)
- DDR3 memory controller
- PCIe Gen2 X8 host interface

19th Real-Time Conference (RT2014), May 2014, Nara (Japan) - P. Vicini, INFN Rome
NaNet-1 Implementation

- Implemented on the Altera Stratix IV dev board (EP4SGX230KF40C2)
- PHY Marvell 88E1111
- HAL based Nios II microcontroller firmware (no OS)
  - System Configuration and Initialization
  - Virtual Memory Management for the GPUDirect RDMA (TLB filling)
NiosII UDP Offload:
- Objective: **offloading the UDP protocol management** from the Nios II microcontroller.
- Collects data coming from the Avalon Streaming Interface of the Altera Triple-Speed Ethernet Megacore (TSE MAC) and redirects UDP packets into a hardware processing data path.
- Current implementation provides a single 32-bit width channel @6.4 gbps (six times greater than the GbE Specs)

NaNet Controller:
- Encapsulation of GbE packets in the APEnet+ protocol (and vice versa for output stream)
  - 32-bit data words coming from the Nios II subsystem into 128-bit APEnet+ data words.
  - Addition of Header/Footer words and redirect in corresponding FIFO
NaNet-1 Test & Benchmark Setup

- Supermicro SuperServer 6016GT-TF
  - X8DTG-DF motherboard (Intel Tylersburg chipset)
  - dual Intel Xeon E5620
  - Intel 82576 Gigabit Network Connection
  - Nvidia Fermi M2070 and Kepler K20Xm
  - kernel 2.6.32-358.6.2.el6.x86_64, CUDA 4.2, Nvidia driver 325.15.
- NaNet-1 board in x16 PCIe 2.0 slot
- NaNet-1 GbE interface directly connected to one host GbE interface
- Common time reference between sender and receiver (they are on the same host).
- Ease data integrity tests.
Communication Latency

- **NaNet-1 M2070**
- **NaNet-1 K20Xm**
- **Sockperf Kernel 2.6.33 M2070**
- **Sockperf Kernel 2.6.33.9-rt31-EL6RT M2070**

RT kernel: less fluctuations but also more latency....
Total latency of the GPU-Based RICH L0-TP using NaNet-1 (Fermi M2070)

Communication + Kernel Latencies (M2070)

Receive Buffer Size [Bytes]

![Graph showing latency vs. receive buffer size](image)

- Kernel Processing Latency (M2070)
- Total Latency
- NaNet-1 Communication Latency

Max throughput > 1M events/s

19th Real-Time Conference (RT2014), May 2014, Nara (Japan) - P. Vicini, INFN Rome
Total latency of the GPU-Based RICH L0-TP using NaNet-1 (Kepler K20X)

Communication + Kernel Latencies (K20Xm)

Receive Buffer Size [Bytes]

Latency [us]

Receive Buffer Size [Number of Events]
NaNet-1 APElink Communication + Kernel Latencies (M2070)

Receive Buffer Size [Bytes]

Latency [s]

Receive Buffer Size [Number of Events]
Note:

- we performed measures in worst case condition…
  - communication and processing tasks were serialized
- while, in normal operation, data communication and GPU processing are overlapped!
  - Circular list of receive buffers: when one of the buffers is full, the GPU kernel is launched, data arriving from the network are accumulated in the next buffer in the circular list concurrently with processing.

Summary:

- A RICH detector Level 0 Trigger Processor GPU-based system using NaNet-1 (1 GbE link) performing the simple MATH processing is able to sustain a throughput of \(~1.6\text{M events/s}\) with real-time behaviour.
- Using a single APEnLink channel instead of the GbE one, the system shows a throughput of \(~10\text{M events/s}\)

--- good scaling with link bandwidth.
NaNet-10 is required to achieve L0-TP required performances (10M events/s)

First functional prototype ready for mid-october 2014 test beam
- run in parallel and “parasitic” mode….
- Aggregation of 4 GbE channels payload in a single 10GbE link through Switch HP2920
NaNet-10 and last generation FPGAs (Stratix V)

- Implemented on the Altera Stratix V dev board but possible future porting on cheaper board (Terasic TR5-F40W)
  - PCIe gen2 x8 but developing PCIe Gen3 (8 GB/s)
  - Faster embedded Altera transceivers (up to 14.1 Gbps)
  - hardened 10GBASE-R PCS features to support 10 Gbps

Network Interface

- TX/RX Block
- 32bit Micro Controller
- UDP offload NaNet
- GPU I/O accelerator
- memory controller
- PCIe X8 Gen3 core

PCIe X8 Gen3 core

4x10GbE (QSFP+)

On Board Memory

PCIe gen3

QSFP+ to 4 SFP+ cable

19th Real-Time Conference (RT2014), May 2014, Nara (Japan) - P. Vicini, INFN Rome
**KM3Net-IT Experiment**

- **KM3Net-IT**: an European deep-sea research infrastructure, hosting a neutrino telescope with a volume of cubic kilometer at the bottom of the Mediterranean Sea.

- "Phase-2 tower" composed of:
  - 8 floors
  - 8 m bars,
  - vertical dist. = 40 m, $H_{tot} = 450$ m
  - 32 OM, 18 hydrophones
  - oceanographic instrumentation

- Current **read-out system** employs a large number of NO state-of-the-art components
  - 2.5 Gb/s optical link per 800Mb/s payload
  - 2 twinned FCM boards per floor
  - Many PCs for HW read-out hosting
  - ...

- When scaling to KM3 many cost/size/power/reliability issues!!!

Details in: *RTA 2-2: Ameli F. “The KM3Net-IT Tower prototype detection unit”*
Why NANET3?

NaNet3 specifications for future enhanced read-out system

- more (4) channels/PCIe board
  - i.e. less PC, less read-out boards,…. ✓
- Link speed-up to 10Gb/s
  - to get 4:1 multiplexing ratio using an optional “Octolink” aggregating board … ✓
- GPU Direct
  - for support to future enhanced read-out and trigger system GPU-based… ✓
- “Fixed Latency” clock distribution for under-water events time-stamping
  - OK for FCM (Xilinx) ✓
  - Altera Stratix IV/V TBD….

Current implementation is being developed on TERASIC DE5-NET

- Altera Stratix V based dev board
- PCIe Gen3 x8
- 4 independent sfp+ ports (up to 10Gb/s)
- Deterministic Latency mode for transceiver
Deterministic latency link inter-operability

Nanet3 On-shore (StratixV)  Fcm Off-Shore (Virtex5)

Clock (+ Data)  Clock (+ Data)

Off shore Recovered clock  Off shore Recovered clock

From on shore data  To on shore data

Preliminary but encouraging results

• Testbed: FCM vs Terasic DE5-Net
  • Custom hw mode for FCM Transceivers (Xilinx)
  • Latency deterministic mode for Stratix V Transceiver
  • 2mt copper and 2 mt long fiber

• Test:
  • 12 hours of periodic (~s) Tx clock reset to verify pll locking and rx word alignment
Conclusions

- NaNet design has been demonstrated effective in different contexts thanks to its modularity and high performance/low latency (RDMA and GPUDirect) features:
  - NA62 L0-TP GPU-based
    - Demonstrated real-time data communication between the NA62 RICH read-out system and the GPU-based L0 trigger processor over a single GbE link
    - Demonstrated scalability up to multiple 10GbE read-out channels (NA62 requirements…)
  - KM3Net-IT on-shore read-out system:
    - Demonstrated different vendors, high-end FPGAs serial links inter-operability (with Deterministic Latency)
    - Compliant with Real-time constraints of TDM processing
    - Ready for future enhancements (Link aggregation, GPU base trigger…)
- Work in progress
  - 10 GbE support on multiple platforms (Terasic DE5-Net, Altera Stratix V dev. Kit,…)
  - Adoption of last generation FPGA devices for PCIe Gen3 integration and faster transceivers (→ higher I/O bandwidth)
  - NaNet3 TDM hardware design for KM3Net-IT
- Nanet-10 and Nanet3 design will be completed and tested on final site before the end of 2014

→ more to come in the next few months….
F. Ameli\textsuperscript{(a)}, R. Ammendola\textsuperscript{(b)},
A. Biagioni\textsuperscript{(a)}, O. Frezza\textsuperscript{(a)},
G. Lamanna\textsuperscript{(c)}, F. Lo Cicero\textsuperscript{(a)},
A. Lonardo\textsuperscript{(a)}, M. Martinelli\textsuperscript{(a)},
C. Nicolau\textsuperscript{(a)}, P. S. Paolucci\textsuperscript{(a)},
E. Pastorelli\textsuperscript{(a)}, L. Pontisso\textsuperscript{(d)},
D. Rossetti\textsuperscript{(e)}, F. Simeone\textsuperscript{(a)},
F. Simula\textsuperscript{(a)}, M. Sozzi\textsuperscript{(c)},
L. Tosoratto\textsuperscript{(a)} and P. Vicini\textsuperscript{(a)}

\textsuperscript{(a)} INFN Sezione di Roma
\textsuperscript{(b)} INFN Sezione di Roma Tor Vergata
\textsuperscript{(c)} INFN Sezione di Pisa and CERN, Switzerland
\textsuperscript{(d)} Università di Roma “Sapienza”
\textsuperscript{(e)} NVIDIA Corporation, USA

\url{http://www.roma1.infn.it/conference/GPU2014/}

19th Real-Time Conference (RT2014), May 2014, Nara (Japan) - P. Vicini, INFN Rome
Might a Real-Time kernel come to our rescue?

Main features of such kernels:

- predictability in response times
- reduced jitters
- microsecond accuracy
- improved time granularity

...but not a panacea...

To avoid other possible sources for latency, the CPUSPEED and IRQBALANCE services has been stopped. The INTERRUPT moderation was disabled either.
Bandwidth saturation at small packets...
NaNet-1 APElink Performance

Receive Buffer Size [Bytes]

Bandwidth [MB/s]

M2070 - Throughput
M2070 - Bandwidth

overlap between GPU computing and communication → 10M events/s
**NANET3 architecture and dataflow**

- **NaNet3 link phy:**
  - Altera Deterministic Latency Transceivers
  - 8B10B encoding scheme
  - Time Division Multiplexing (TDM) data transmission protocol
    - payload of different off-shore devices, multiplexed on continuous data stream at fixed time slot

- **Preliminary test of:**
  - Xilinx – Altera transceivers interoperability
  - Interoperable link with Fixed (Deterministic) Latency
Current read-out system employs a huge number, NO state-of-the-art components

- 2.5 Gb/s optical link
- 2 twinned FCM boards per floor
- Many PCs for HW read-out hosting

When scaling to KM3 many cost/size/power/reliability issues!!!
NaNet-1 Latency Benchmark

In a single host process:
- Allocate and register (pin) \( N \) GPU receive buffers of size \( P \times 1168 \) byte (\( P \times 16 \) events) in a circular list.
- In the main loop:
  1. Read TSC Register (\( \text{cycles}_{bs} \))
  2. Send \( P \) UDP packet with 1168 byte payload size over the host GbE intf
  3. Wait for a received buffer event
  4. Read TSC Register (\( \text{cycles}_{ar} \))
  5. Launch the GPU kernel on next buffer in circular list.
  6. Synch Streams
  7. Read (\( \text{cycles}_{ak} \))
  8. Record latency\_cycles\_comm = \( \text{cycles}_{ar} - \text{cycles}_{bs} \)
  9. Record latency\_cycles\_calc = \( \text{cycles}_{ak} - \text{cycles}_{ar} \)

- Results in good agreement with oscilloscope measures (TEL62).
Events can produce multiple ring hits patterns on PMs.  
- 95% of generated hits patterns produce a single ring.  
- Single ring search algorithms are used by multi-ring search algorithms.  
- Let’s focus on single ring pattern search algorithms.

**DOMH/POMH**: Each PM (1000) is considered as the center of a circle. For each center an histogram is constructed with the distances btw center and hits.
- **HOUGH**: Each hit is the center of a **test circle** with a given radius. The **ring center** is the best matching point of the test circles. Voting procedure in a **3D** parameters space.

- **MATH**: Translation of the ring to **centroid**. In this system a **least square method** can be used. The circle condition can be reduced to a **linear system**, analytically solvable, without any iterative procedure.
