AuroraScience

web.infn.it/aurorascience

Luigi Scorzato (ECT*)
for the AuroraScience Collaboration

Lattice 2010 - Villasimius

Motivations

Lattice QCD community and application driven HPC systems

apeNEXT

QCDOC

QPACE

BlueGene

Janus

Motivations

Provide suitable computing tools for our LQCD calculations

Ensure interaction between HW developers and scientists

Knowing how a machine works inside may give significant competitive advantages.

Motivations

In recent years, the number of scientific fields where HPC has become essential for competition has encreased considerably.

Funding agencies see the investments in HPC as highly strategic in order to boost progress in many scientific fields efficiently (optimization of resources, dissemination of know how)

Experience from our community highly recognized
⇨ more funding opportunity for LQCD

AuroraScience

- Develop and Build a prototype of science-optimized HPC system:

  ➡Test Board (last year)→ 25 Tflops (this summer)→~100 Tflops (next year)

- Convincing evidence that a cost-effective Pflops-scale system can be built.

- Based on industry-grade building blocks

  ➡ Intel processors: high perf., clear roadmap, standard tools.

- Develop a no-frills communication system

  ➡ APE-like 3D network, minimal overhead

- Good for LQCD, but also for other scientific fields

  ➡ Get them involved to get Science out of it (in trivial and non-triv. contexts)

→ ***AuroraScience***

AuroraScience Collaboration

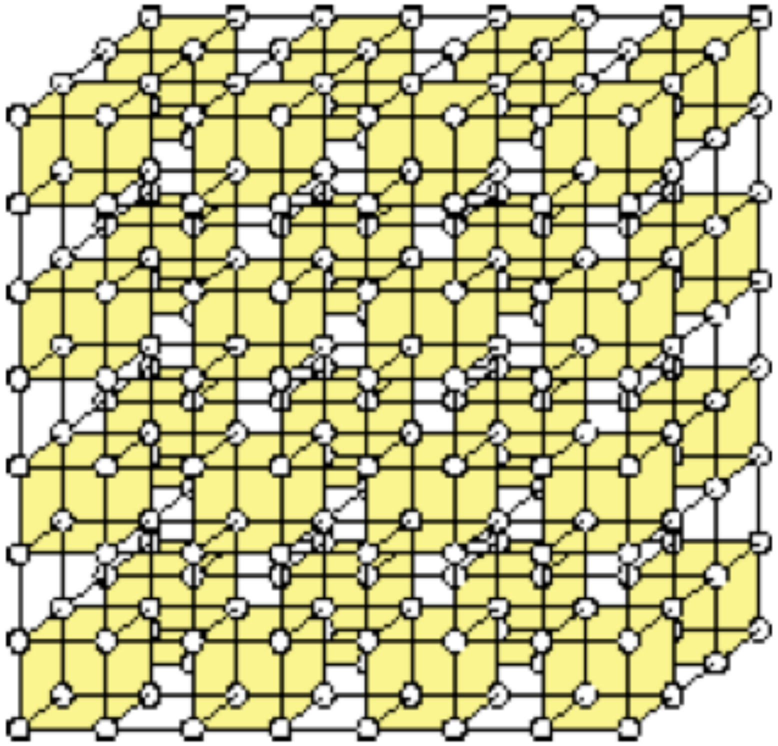
Scientific Coordinator: R. Tripiccione (INFN)

- *ECT*/FBK.* (Director A.Richter) Coordinating Institution.
- *University of Trento* (Groups of Nucl-th G.Orlandini and F.Pederiva)
- *Fondazione E.Mach*. (Group of Bioinformatics R.Velasco)
- *Agenzia Provinciale per la Protonterapia [ATreP]*. (Group of M.Schwarz)
- *INFN* (Groups of Ferrara, Parma, Milano-Bicocca).
- *DEI-Padova.* (Group of Computer Science of G.Bilardi).

Funded by: Provincia Autonoma di Trento (PAT)
& Istituto Nazionale di Fisica Nucleare (INFN)

Industrial partners: **Eurotech** & **Intel**.

The Computing System

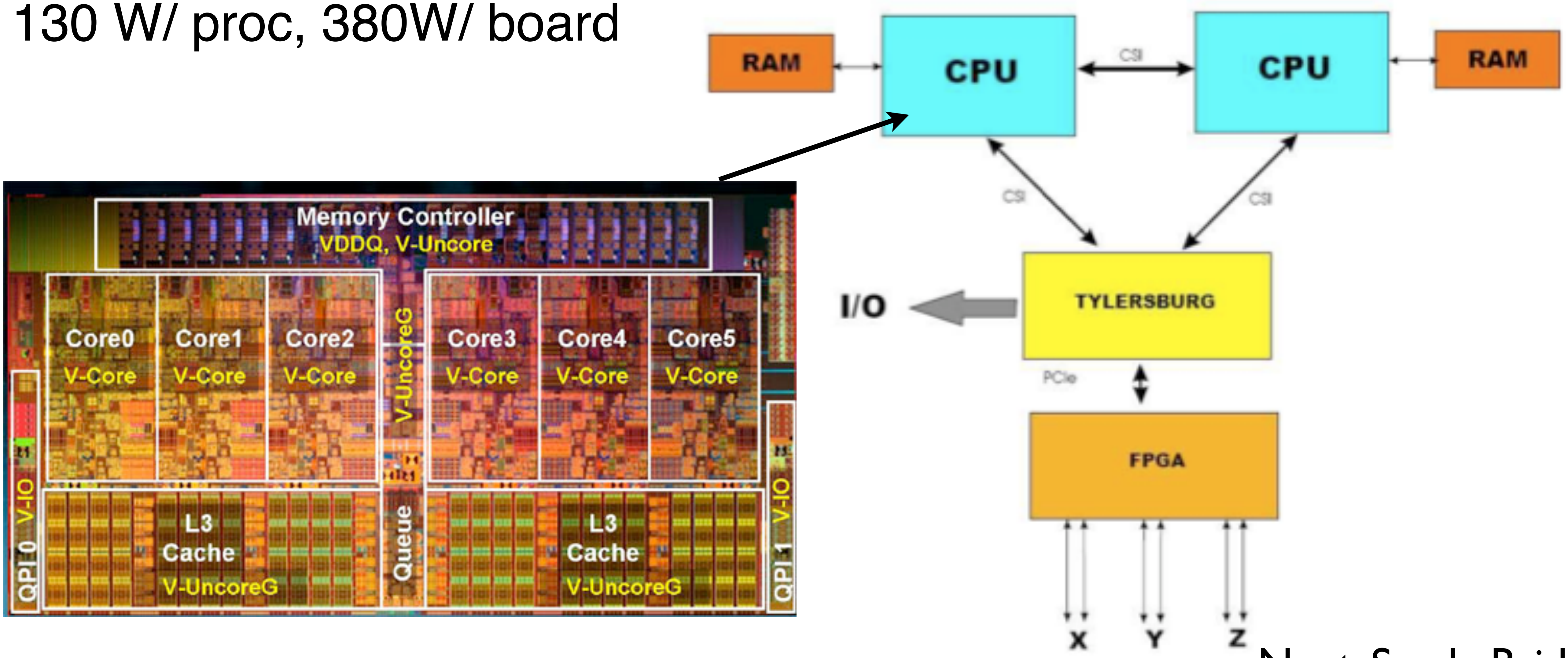


3D mesh of computing nodes

son of the **APE** and **QPACE** tradition

Processors

High End HPC Intel line of processors:
E.g. **Intel Westmere 6C** processor
6 cores x 4 instr/clk x 3.33 Ghz ➔ **80 Gflops**
256 KB L2 cache per core
12 MB L3 cache shared
130 W/ proc, 380W/ board



Next: Sandy Bridge
8 cores, ~200 GFlops

Note: room for "trivial parallelizations" will get narrower

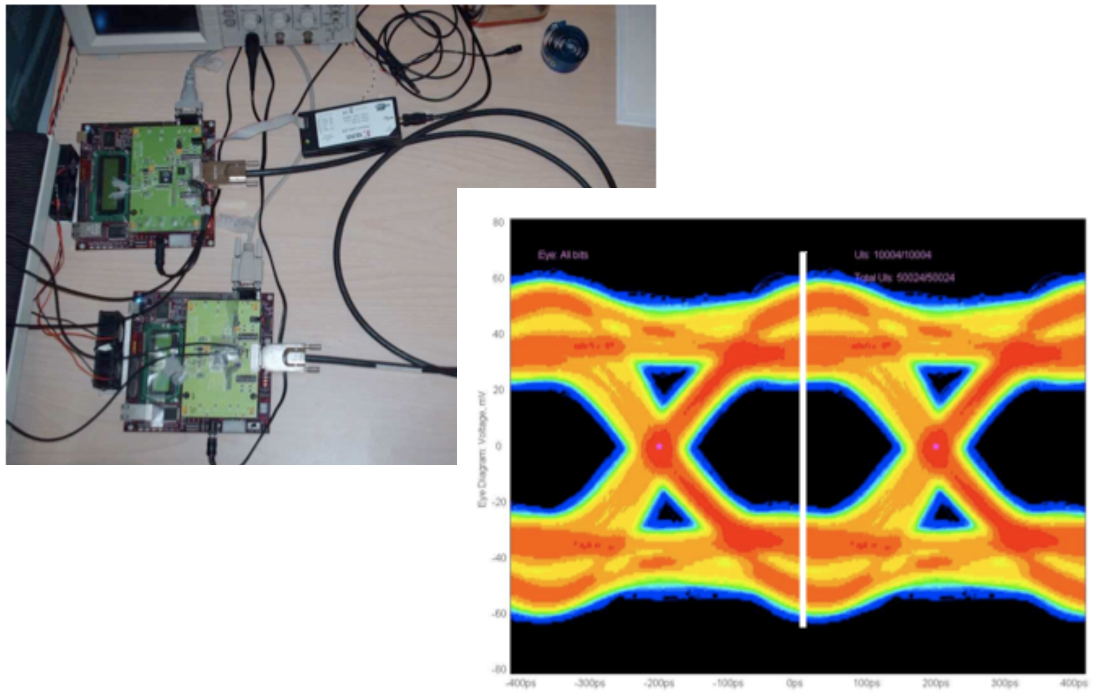
Aurora Torus Network ATN

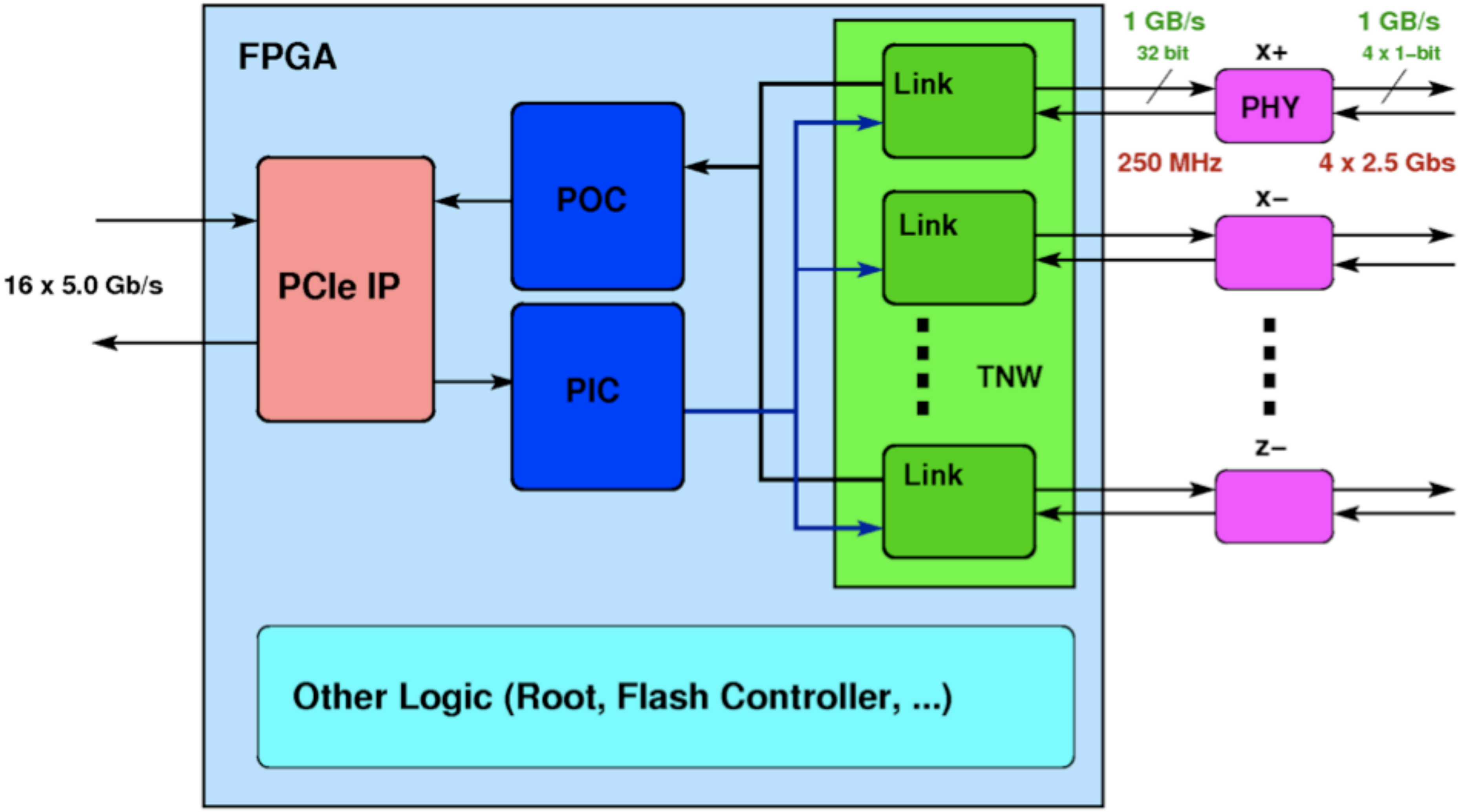
The network is conceptually simple but ... technically challenging

Trade logical simplicity with:

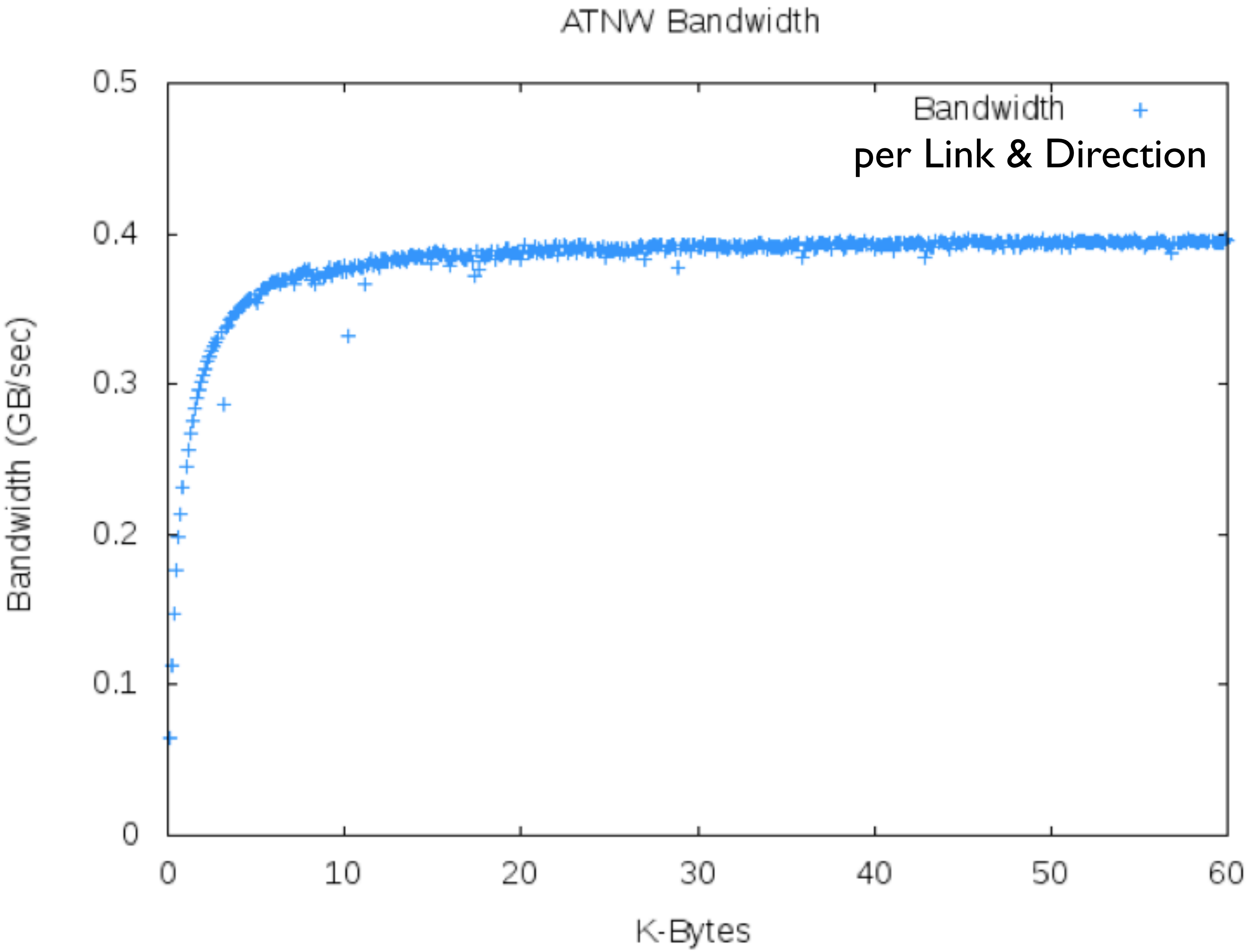
-High Bandwidth
(goal: 6 x 1 GB/sec)

-Low-Latency
(goal: ~ 1 micro-sec)





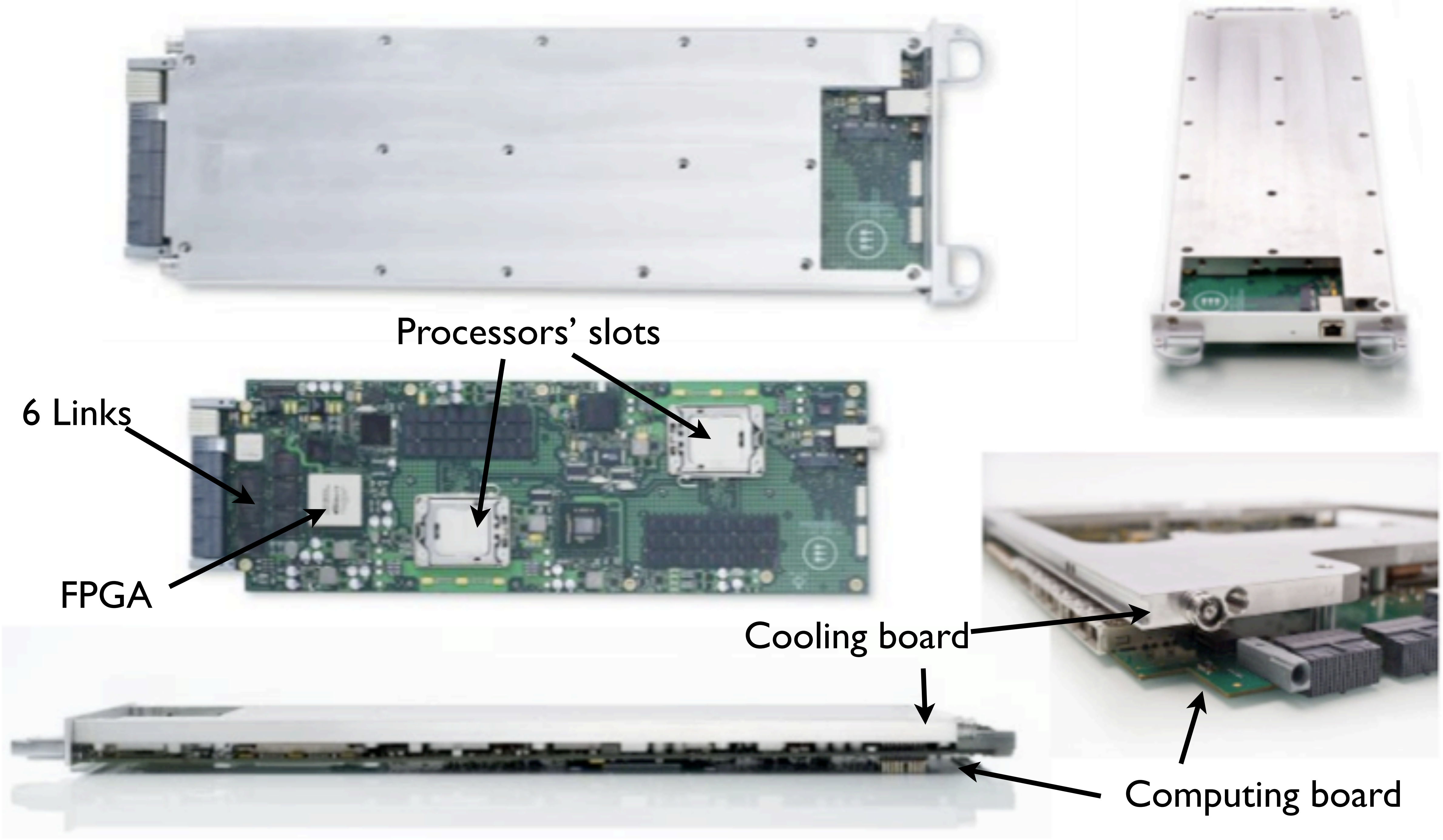
Aurora Torus Network ATN



Transfer time for 1 packet = 128 B is ≈ 1.96 μsec

See poster by Marcello Pivanti

Node Card



Processors' slots

6 Links

FPGA

Cooling board

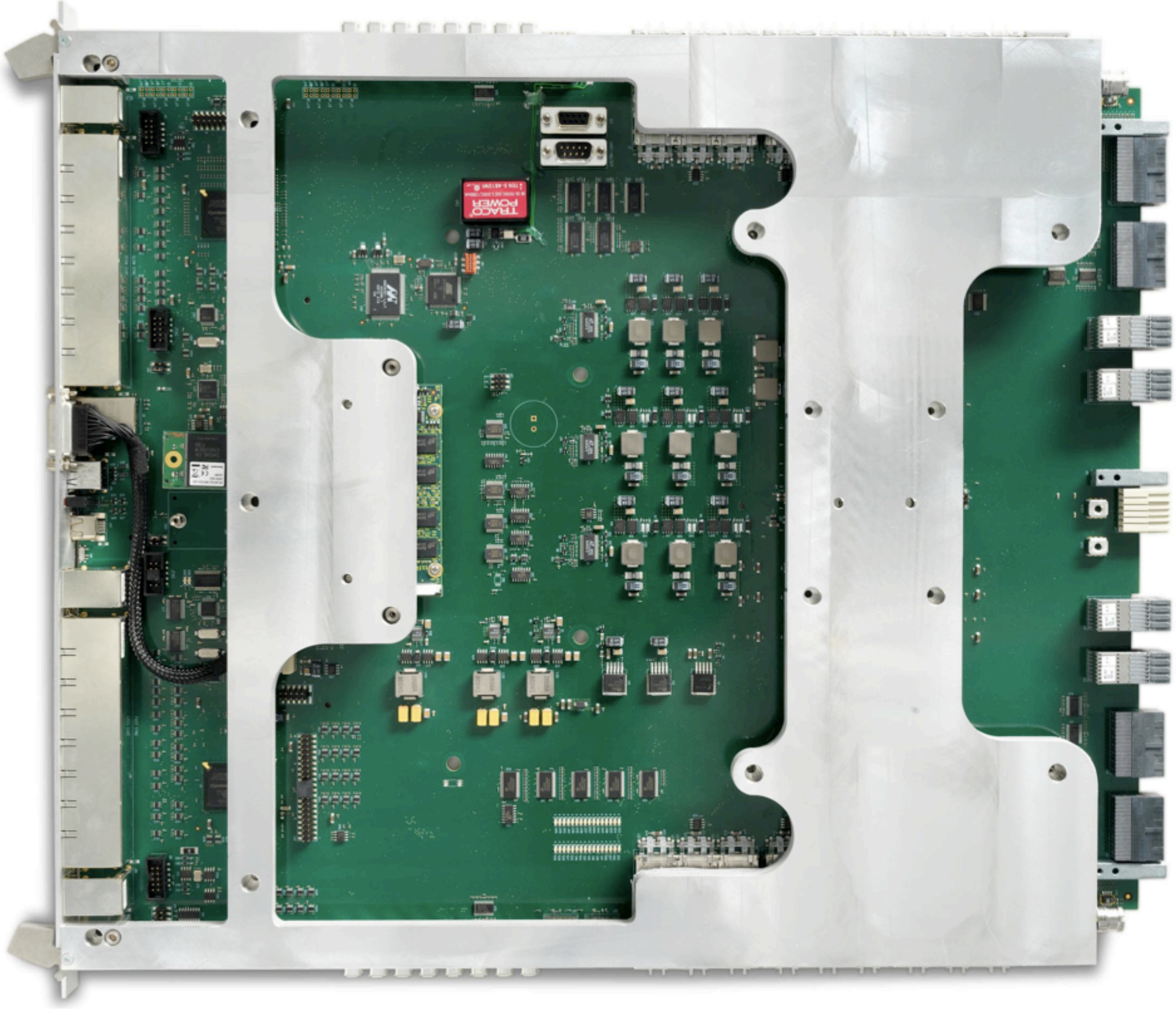
Computing board

Developed by Eurotech
with substantial inputs by AuroraScience

EuroTech
A MEMBER OF EUROTECH GROUP

The other Networks and the Root Card

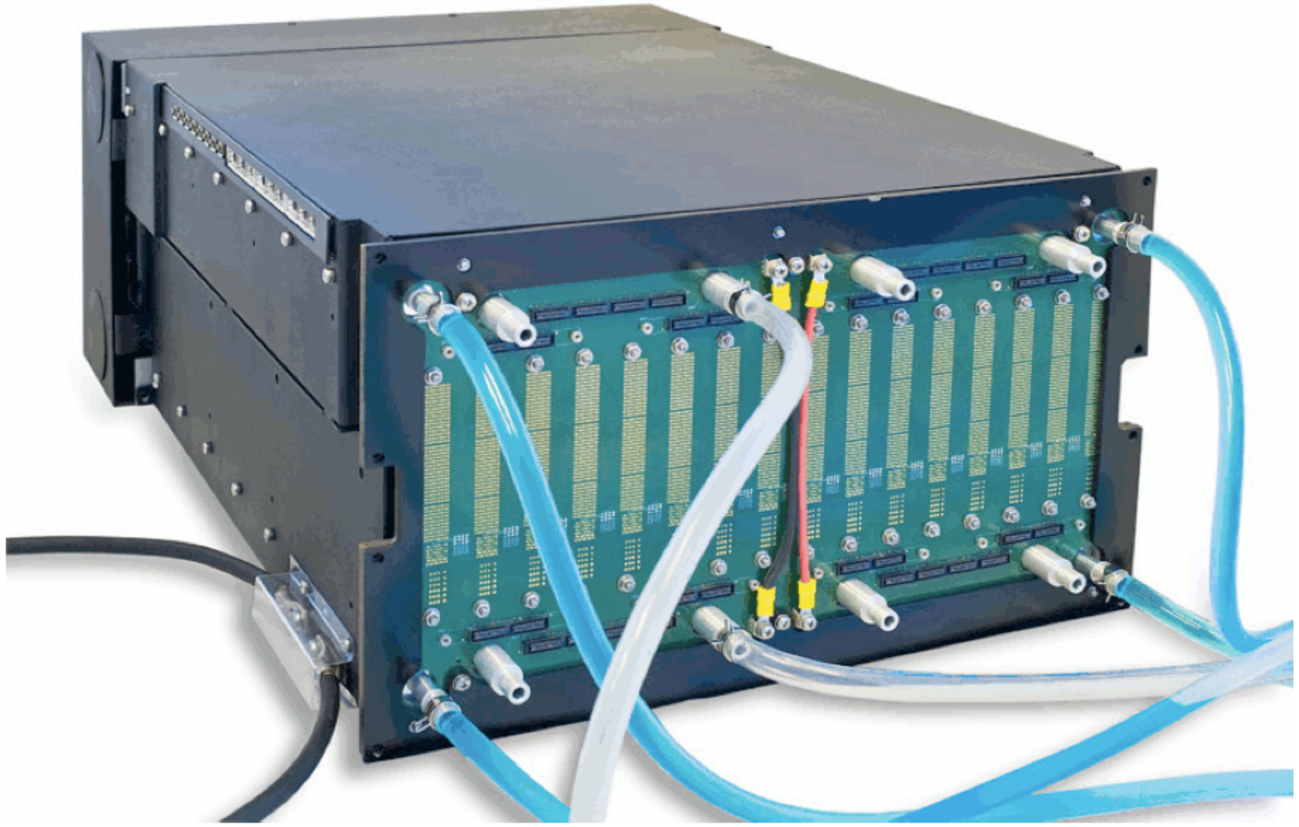
Besides the **3D torus network** designed by AuroraScience, the system inclues also an **Infiniband (IB) network** with one (36 ports) switch per chassis (16 node cards) and a **Syncronization network** with tree topology (also APE inspired).



EuroTech

A MEMBER OF EUROTECH GROUP

Chassis



Cooling System

Schedule of Installation

✓ **Spring 2009.** First (Nehalem based) node card ready (45 nm, 100 GFlops, ~300 W)

✓ **August 2009**. Official start of the project

✓ **Last week.** 32 nodes / 2 chassis (Nehalem based) powered on and hosted at Eurotech (copy of firmware ongoing).

➡ **Mid July.** Upgrade to 48 nodes / 3 chassis (Westmere based).

➡ **Mid August.** Complete installation at FBK of 10 chassis (25 TFlops).

➡ **2011.** If everybody is happy, final prototype of ~100 TFlops to be installed.

Scientific Applications

- LatticeQCD - ECT*, INFN
- Molecular Dynamics - UniTN
- Lattice Boltzmann - Ferrara
- Quantum Monte Carlo for Nuclear Structure - UniTN
- Linear Algebra for Nuclear Reactions - UniTN
- Bioinformatics - FEM
- Monte Carlo for Radiotherapy - ATreP
- Application Independent Optimisation - UniPD

Scientific Applications

We consider applications:

- Either with a long experience of designing optimized codes for a given architecture and even designig dedicated architectures for the algorithm.... (LQCD, Nuclear Physics, Lattice Boltzmann, Molecular Dynamics).

- Or with an emerging large need of computing power and a relevant social impact (Bioinformatics, Medical Physics).

- All committed to invest now human resources to work on the scalability of their algorithms and the to optimize of the common computing architecture.

Interface with the Applications

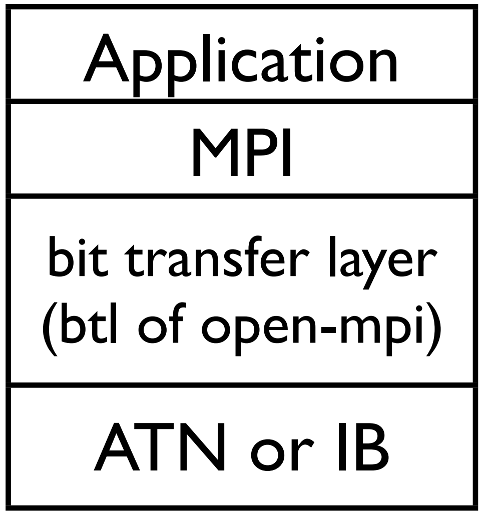
Standard x86 architecture with all associated tools.
Standard access to storage via IB and parallel file system (gpfs/lustre).
Standard MPI communications via Infiniband (IB).

On top of all this, custom 3D Torus Network (ATN)

The ATN primitives can be accessed
from any high level programming language

Interfacing the Applications with ATN

Putting ATN **below** a full MPI implementation (e.g. openmpi) would be possible and interesting, but it would add an unknown and uncontrollable overhead.


Application

MPI

bit transfer layer
(btl of open-mpi)

ATN or IB

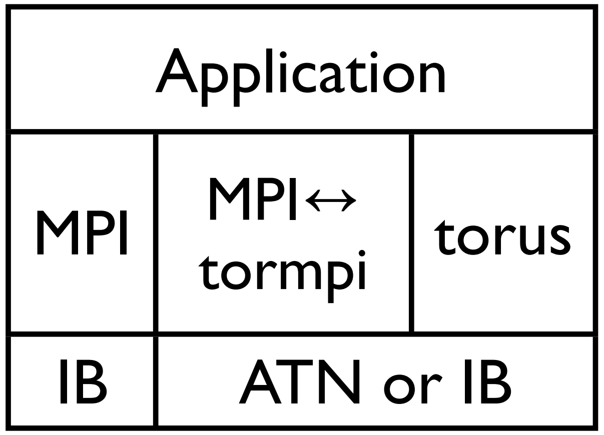
Torus library

Instead, we developed an interface **between** the Application and the ATN library with the goal of helping the porting

- First Set of Functions (tormpi_): wrapper of some MPI functions, which are constraint to have exactly the same prototypes and return status as the corresponding MPI functions. Require no modification on the code.

- Second Set of Functions (torus_): add more freedom to exploit the features of ATN, but require modifications of the code.

The code can use MPI_, tormpi_, torus_, next to each other


Application

MPI | MPI↔ tormpi | torus

IB | ATN or IB

Comments on the Torus lib

- **MPI** is used to start the jobs (mpirun); it is used in the initialization; it may be used for *local* operations but completely skipped in communications when we have a better option.
- The **overhead** introduced by the toruslib is minimal
- With this approach, I can start running the code since the beginning and gradually implement via the ATN first the most critical parts and later the others, if needed.
- Consistency of the coexisting MPI and ATN calls is ensured by the tormpi_init().

tmLQCD

- It is the main code used and developed by the ETM Collaboration. It reflects the many different physical interests of ETMC.
- We ported the full code to the ATN using the torus library described above.
- Here I concentrate in the benchmark of the Hopping Dirac Operator, which is the kernel of all critical computations.

tmLQCD

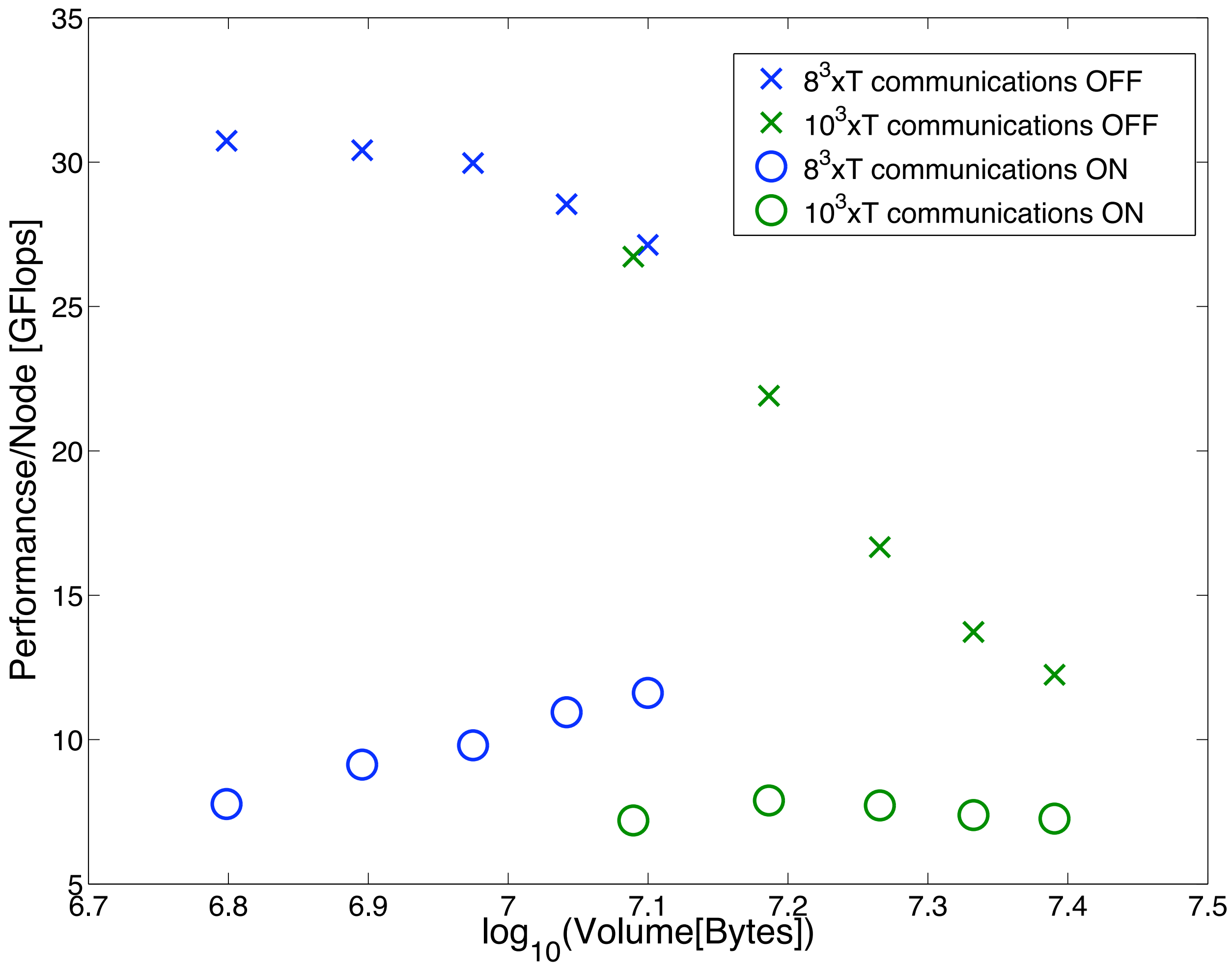
Preliminary benchmark, just after completing the torus library

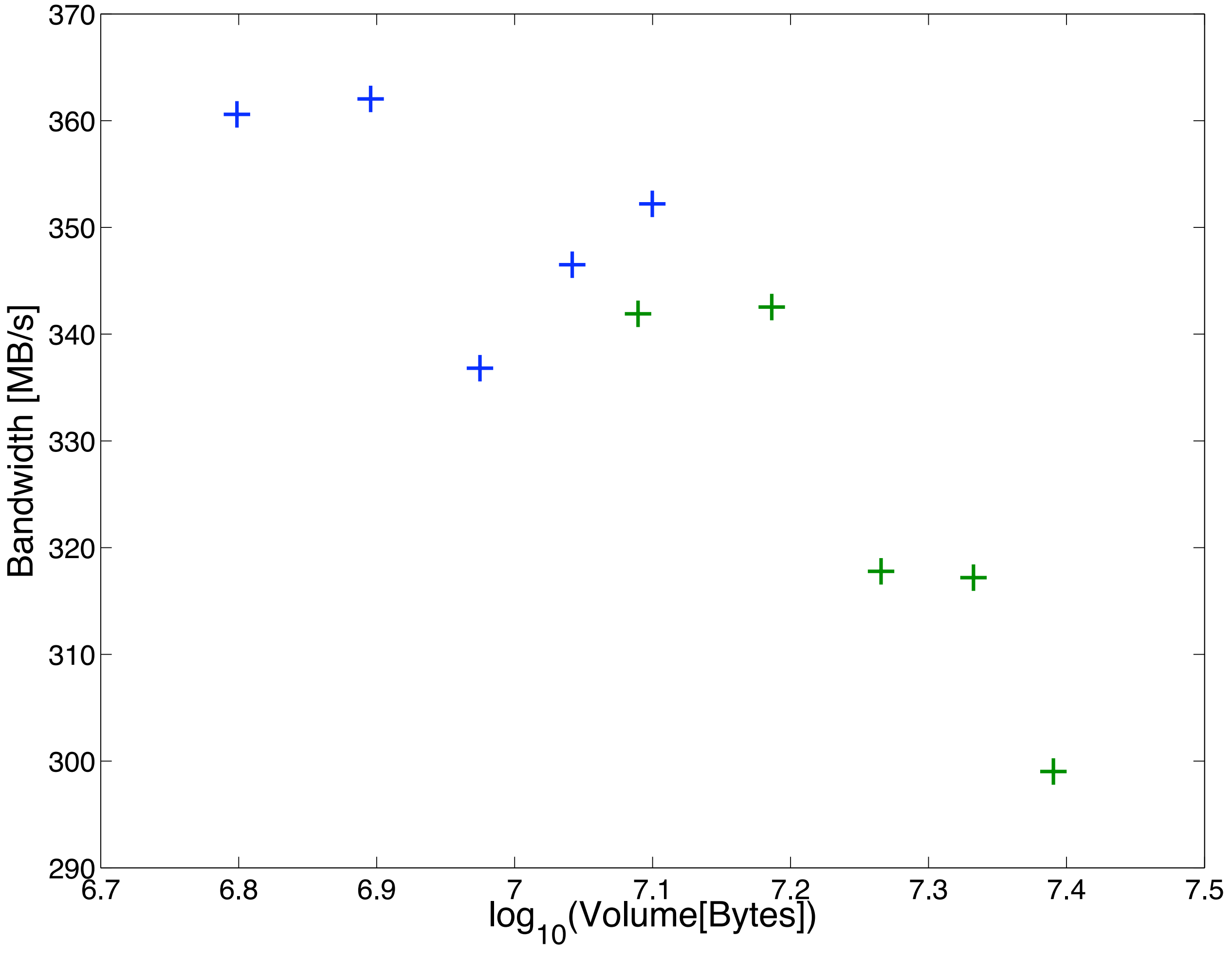
No further attempt to optimize yet.

Simply subst: MPI_Sendrecv() ➔ tormpi_sendrecv()

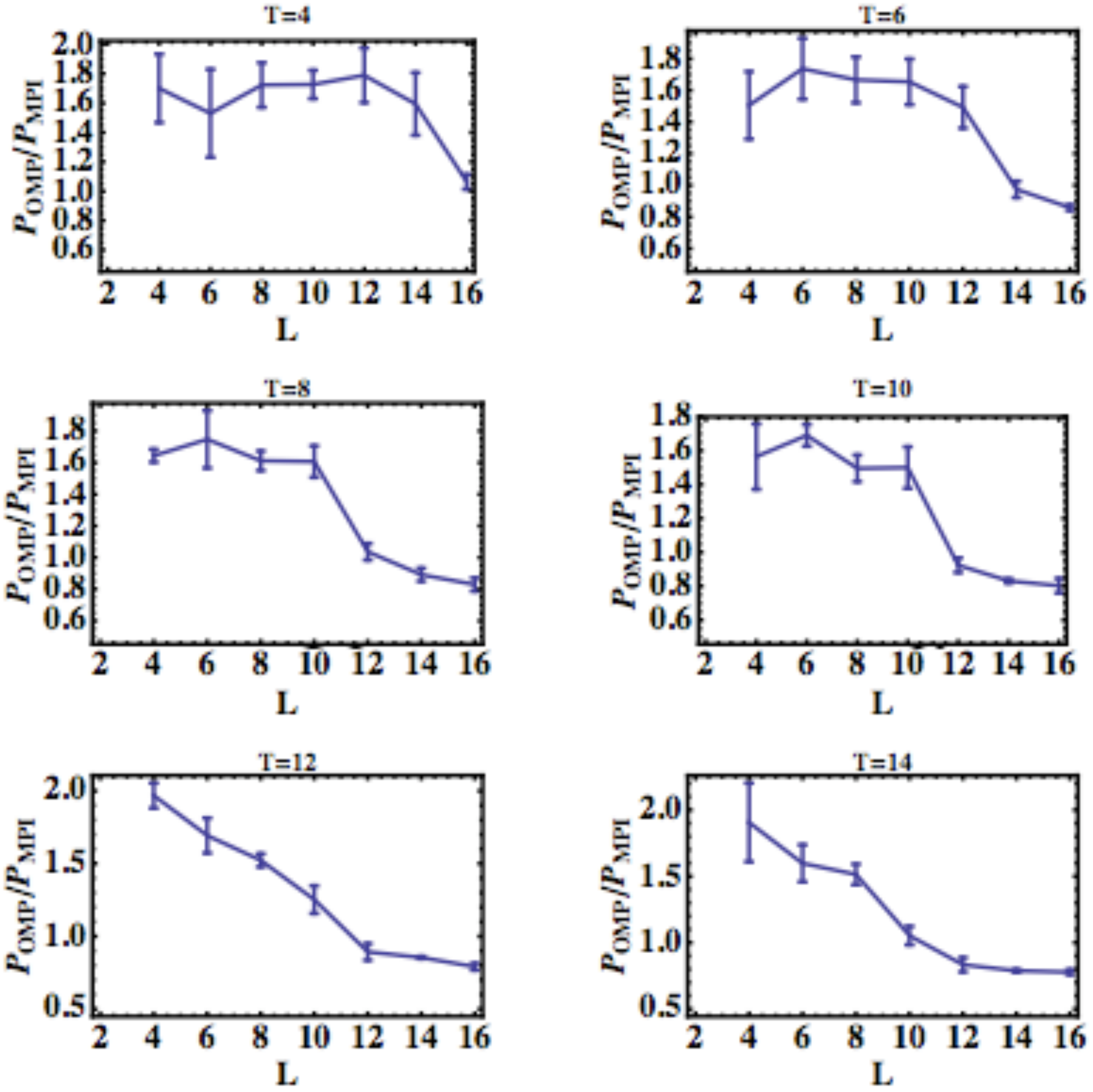
2 Nehalem boards connected via ATN in a ring.

In-board parallelization with 8 openMP threads





tmLQCD: threads vs MPI



A better parallelization strategy

Overview of parallelization strategies

- High ratio ρ = Network Latency / Time for Floating Point Ops

  ➡ Typical situation in Clusters.

  ➡ collect as much data as you can before send/recv

  ➡ Send/Recv the whole borders after a whole sweep over the lattice

  ➡ Original approach of the tmLQCD and most portable LQCD codes

- Low ratio ρ.

  ➡ Send/Recv each data just before you need it, in order to overlap commnications and computations easily and maximally.

  ➡ Ape remote addressing: U[x + up]

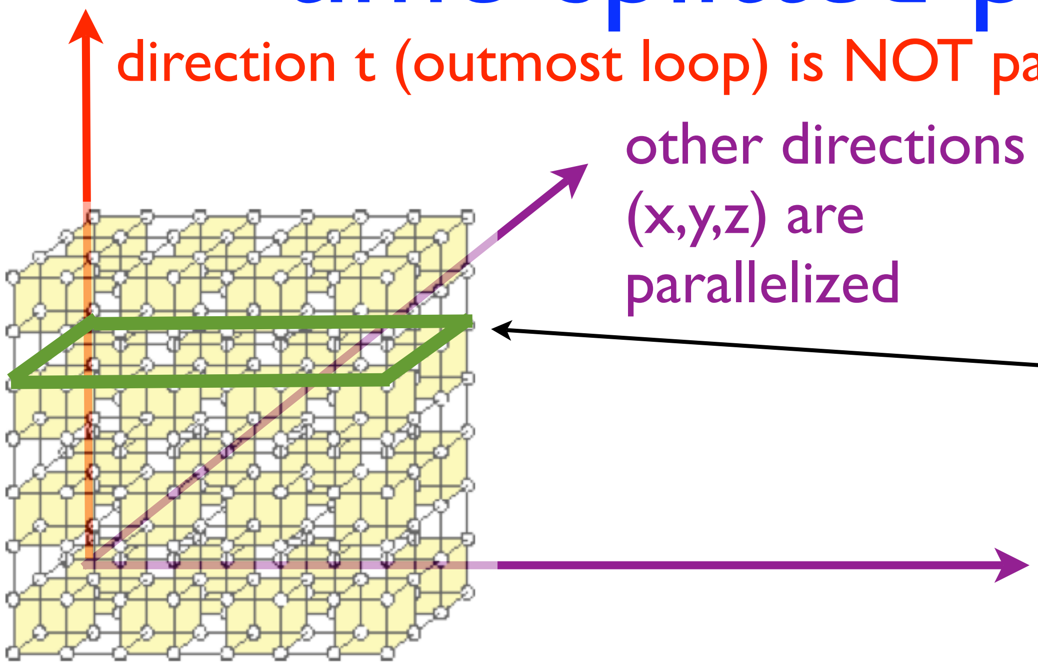
- Intermediate ratio ρ.

  ➡ Intermediate solution

  ➡ Split communications in order to overlap with computations. But still in big chunks.

  ➡ QPACE, AURORA

time-splitted parallelization (fermions):



direction t (outmost loop) is NOT parallelized

other directions (x,y,z) are parallelized

- Instead of sending all the boundaries after a full sweep on the lattice (after ~ L^3T computations),
- After each timeslice sweep (~ L^3 computations), send the 2D boundaries (t-x), (t-y), (t-z),
- Strategy introduced by the QPACE coll.

- The overhead of the communications increases, BUT:
- At any time of the computation, the data needed are at most 3 timeslices.
- The communication of the 2D boundaries (t,x=0,L), (t,y=0,L), (t,z=0,L) can overlap with the computation in the timelike links in the (t-1)-slice, since this only need local data.
- On L=8 lattices, the computation of timelike links in one timeslices (which can be completely overlapped with communications) is ~5μs. This should be > latency in order to cover it completely.
- In larger local lattices the constraint on the latency is even more favourable.

Conclusions

- I have given an overview of the AuroraScience project.
- A lot of work has been done in the past two year to develop the computing system and in particular the 3D torus network.
- The installation of the first prototype is ongoing and expected to be completed this summer.
- A lot work has been done to ease the porting of the applications.
- The tmLQCD code is working and I have shown preliminary (not optimized) benchmarks.
- Work on other applications is ongoing.