The Aurora Project: a status report

F. Di Renzo

University of Parma and INFN
AuroraScience Collaboration

Legnaro, Feb 16th 2011

OUTLINE

One short notice, just to start. While I stayed with the original title, you will hear in the following about Aurora and AuroraScience: Aurora is as a matter of fact both a research project and an industrial project. Well, machines are simply called Aurora. I will sketch a status report, mainly focusing the current status of our Aurora parallel system.

> You will probably notice that this is in the end a proper time for AuroraScience to state "where we stand".

- AuroraScience: the Collaboration and the Project
- The Aurora platform and our Aurora machine (the project vs what we have by now)
- The Aurora programming environment
- Applications on/for Aurora

AuroraScience and Aurora

AuroraScience is a joint-project of INFN and Provincia Autonoma di Trento.

- ECT* (*they host the installation*)
- INFN (Ferrara, Milano Bicocca, Parma)
- University of Trento (*Nuclear Physics and Protein folding*)
- DEI (University of Padova, *Algorithms and models of computation*)
- ATreP (*proton therapy*)
- FEM (*genomics*)

Aurora is an industrial project by Eurotech.

> **Eurotech, Intel to Collaborate on HPC Development**
>
> AMARO, Italy, May 14 -- Gruppo Eurotech, an Italian company specializing in embedded solutions and computer miniaturizatrion, and Intel, the world's largest producer of chips and world leader in silicon innovation, have signed a memorandum of understanding over several years of technological collaboration.Under the terms of the agreement -- signed during a visit by Pat Gelsinger, vice president and general manager on Intel's Digital Enterprise Group -- the two companies will work together on the development of HPC (high performance computing) systems based on Intel processors. These systems will address the computing needs of fluid dynamics and aerodynamic analysis, as well as scientific computing.

AuroraScience project basic facts and goals

- Officially started 31 July 2009
- 1st phase: 18 months (+6), 1852K€ (mostrly PAT, 300K€ INFN)
- 2nd phase: 18 months, roughly the same budget
  (to be confirmed after evaluation!)

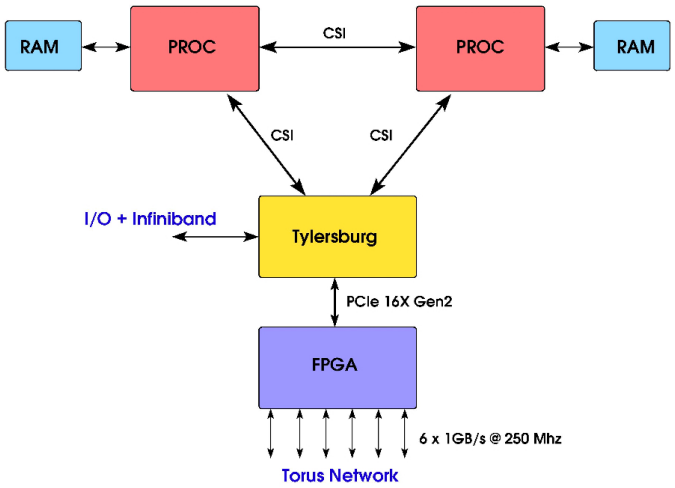
Actually the work unofficially started earlier (2008). Members of AuroraScience strongly inspired and guided the design of Aurora hardware. In Autumn 2008 Eurotech had the overall project mainly ready.

- budget goes mainly into positions and prototypes
- 1st phase: $O(20)$ TFlops machine (was expected summer 2010) ($\sim$0.7M€)
- 2nd phase: $O(xxx)$ TFlops machine (2011) ($\sim$1M€) ($xxx \sim 100$)

Our main goals:

- Design and development of a parallel system based on
  Intel multi-core CPU's and APE-like 3DT(orus) network
- Provide evidence of scalability to PFlops range
- Optimization of applications (not only LQCD ...)
- Development of programming strategy for multi-cores parallel systems

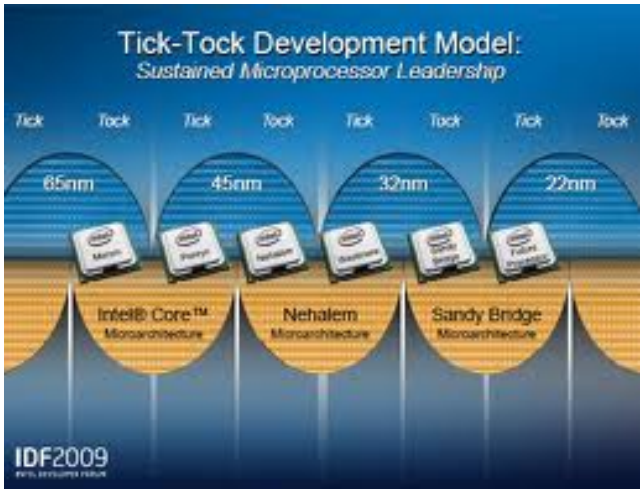
Aurora node card



- 2 CPUs
- 12 GB of RAM (6 GB per processor)
- 1 QDR 40-Gbit/s Infiniband adapter
- 1 FPGA Altera Stratix IV GX230
- 6 PMC-Sierra quad-link PHYs

Take the (hopefully) best off-the-shelf (CPU, IB) and plug a 3DT on top of it (plus liquid cooling)

CPU



- Intel has a (*Tick-Tock*) roadmap
- Eurotech has a MoU with Intel for cooperation in HPC
- AuroraScience is supposed to take advantage of cutting-edge technology

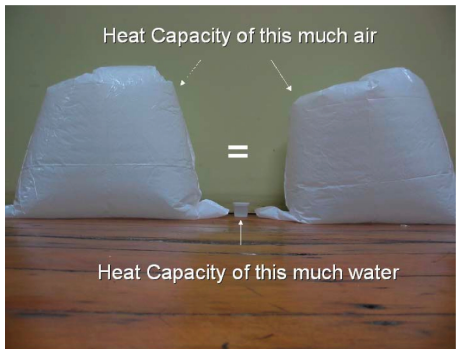
- 4-core Nehalem, 50 GFlops peak double-precision
  used in early prototypes node cards (we still have $O(20)$ boards)
- 6-core Westmere, 75 GFlops peak double-precision
  current stage; from Autumn, 61 node cards up (plus some spare)
- 8-core Sandy Bridge, $O(200)$ GFlops peak double-precision (AVX 256-bits!)
  available for 2011 prototype? Intel admitted they are late (see later)

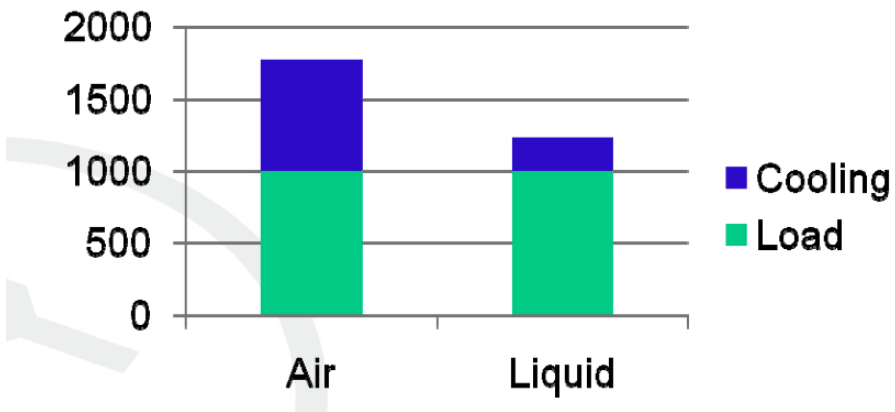
Aurora system

- 16 node cards make a crate (half-chassis), together with a DC/DC trayer (bottom) and a root card (top), delivering as now 2.5 TFlops peak performance
- 2 crates (half chassis) are sitting back to back on a floor
- 8 chassis would make a rack, delivering as now 40 TFlops peak performance (provided you provide some 120 KW ...)

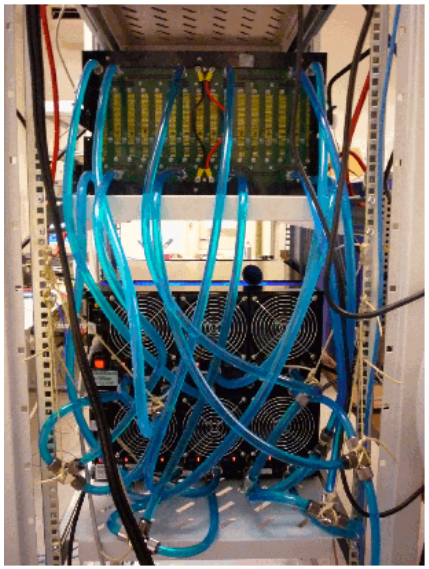
Liquid cooling

Liquid cooling (see the *cold-plate*!) is one of the key issue for a very compact design. *Quick-disconnect* technology first used in HPC.





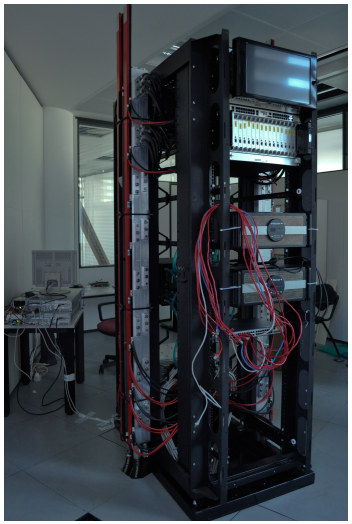
Liquid cooling

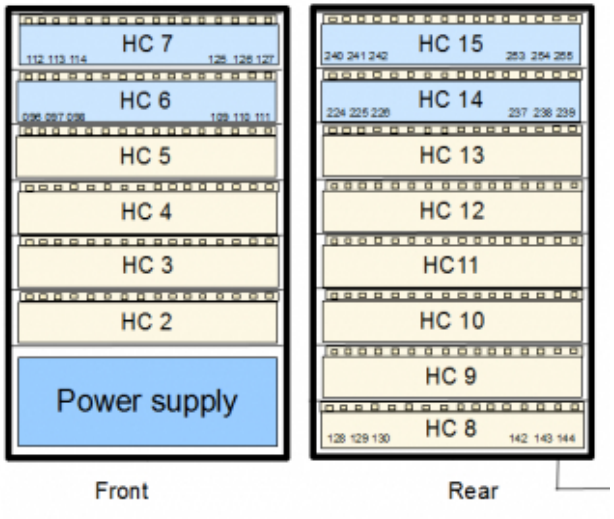


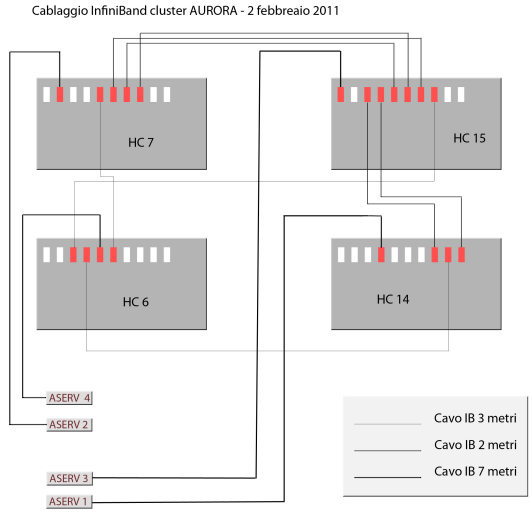
On the left, the first cooling apparatus in Ferrara. On the right, the first stage of the system in Trento. As now, chiller on the roof of the building (and pipes under the floor!)

- The cooling system infrastructure took some time: ready end of June
- Working at the design specs (1.5 lt/min per node, for a ΔT of $O(20)$)
- Nodes grouped in blocks of 4
- Some minor problems; one bigger problem a few weeks ago
  1 DCDC trayer had to be changed (plumbers to be blamed, actually)
- Free cooling almost all the time (10 out of 12 months)

Our current Aurora installation

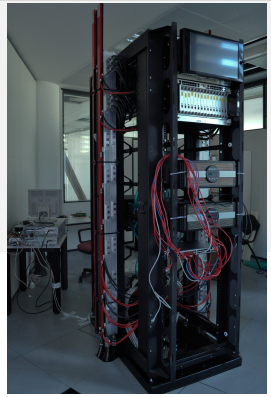






- At the moment, we have 61 rev. E (plus 3 Nehalem-rev. D boards) nodes (10TFlops), i.e. 2 floors (3DT closes in all three directions)
- extra 10TFlops to come
- 2 (plus 2) servers give access to the system and provide general services.
- Nodes root file-system installed and configured cloning that of the master.
- A NAS is almost ready; fibre channel; 12+12 SAS disks (600GB and 2TB) GPFS plus NFS; a lot of I/O tests (Lemon)

Alive since summer time (check it on the web ...)



Ganglia .sourceforge.net

Cluster Report for Mon, 11 Oct 2010 09:26:09 +0200 Get Fresh Data

Metric load_one Last hour Sorted descending Physical View

AuroraScience Grid > Aurora-Cluster > --Choose a Node

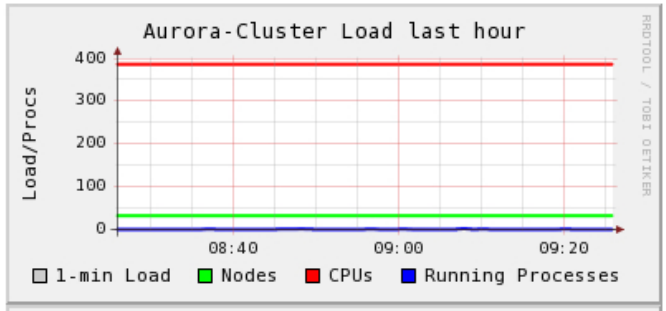
Overview of Aurora-Cluster

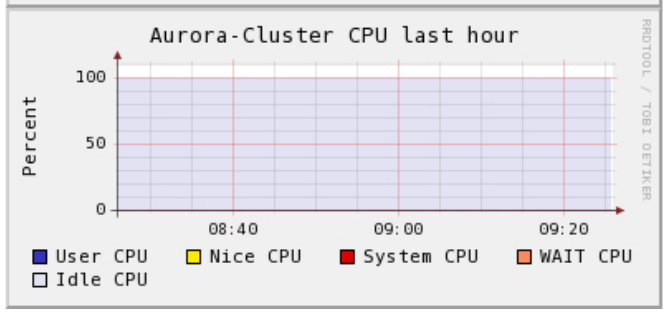
CPUs Total: **384**
Hosts up: **32**
Hosts down: **0**

Avg Load (15, 5, 1m):
0%, 0%, 0%
Localtime:
2010-10-11 09:25

Cluster Load Percentages

0-25 (100.00%)

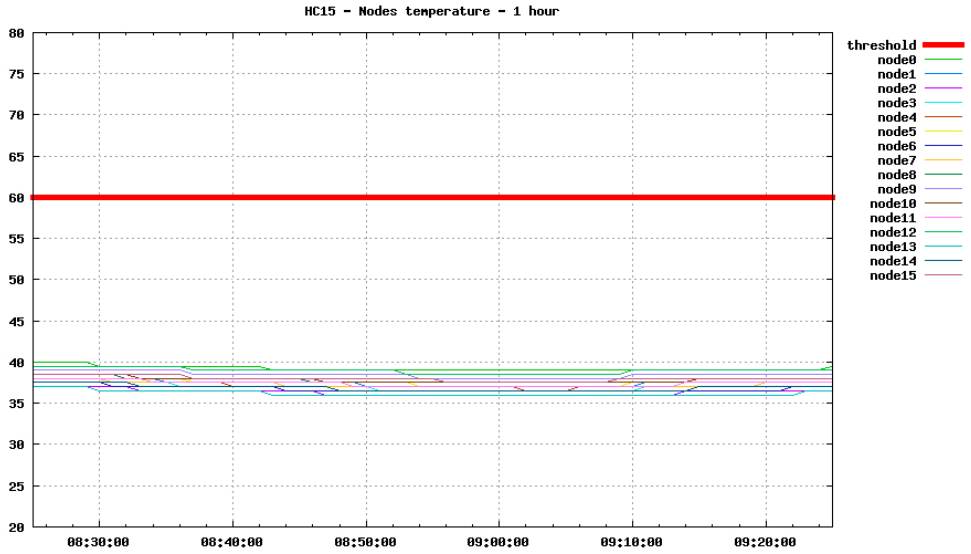




Aurora interconnection networks

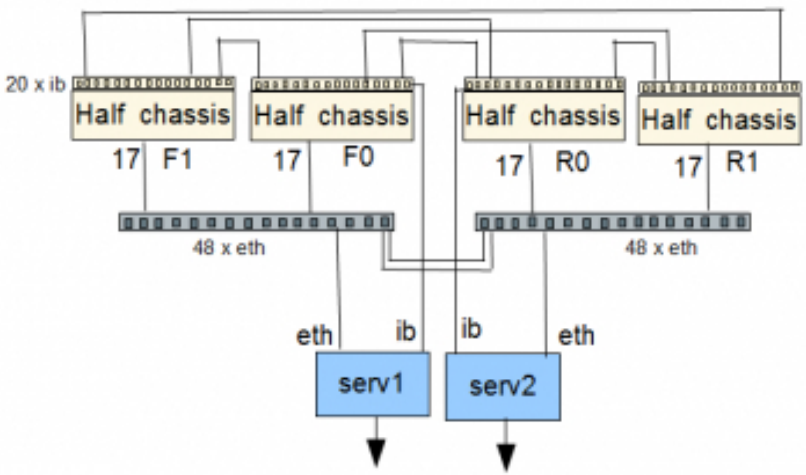
- Infiniband switched network, used for IO and general communication patterns
  - the root board hosts a 20-port infiniband switch
  - 16 ports are connected to the node card of the chassis
  - 4 ports are used for cabling a multi-chassis machine
- a 3D-Torus network for nearest neighbor communications (based on TNW)
- gigabit-ethernet switched network:
  mainly used for boot, monitor and debugging
- a tree-like fast (25MHz) network for global synchronizations a-la APE
- a tree-like slow (10 MHz) network for monitoring and slow synchronizations

Root card



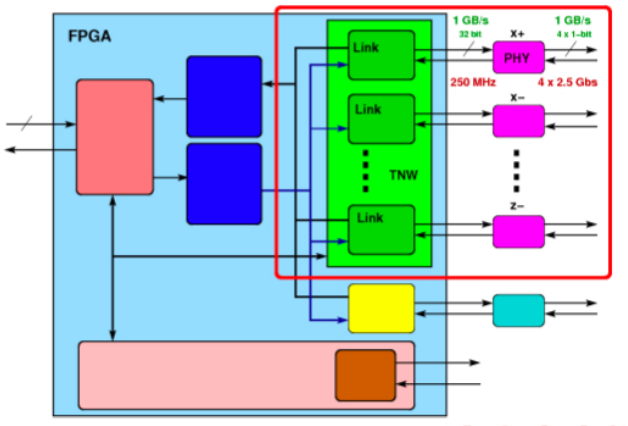
- Root card can have more functionalities than it has now...
- Used for power management, temperature monitoring, IB connections
- Some work done on the tree-like fast global synchronization network (apeNEXT legacy)
- The slow tree-like network is mainly Eurotech stuff

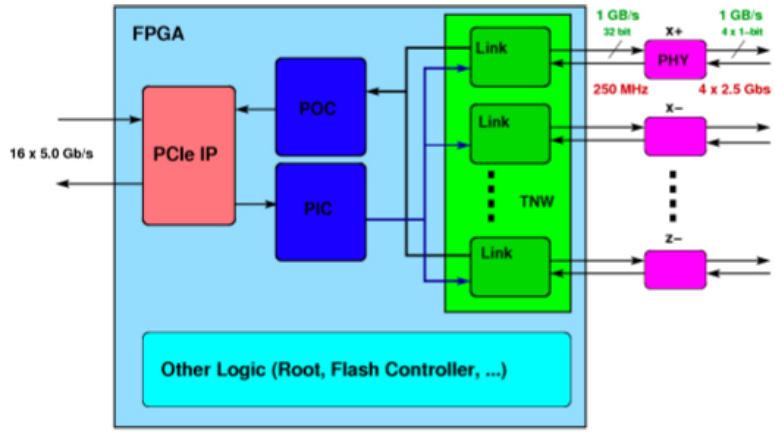
IB



- IB network is at the moment quite a minimal one ...
- ... but we have the freedom to envisage different configurations ...
- ... for both inter-node communications and I/O.
- /home exported to the nodes via NFS over IB

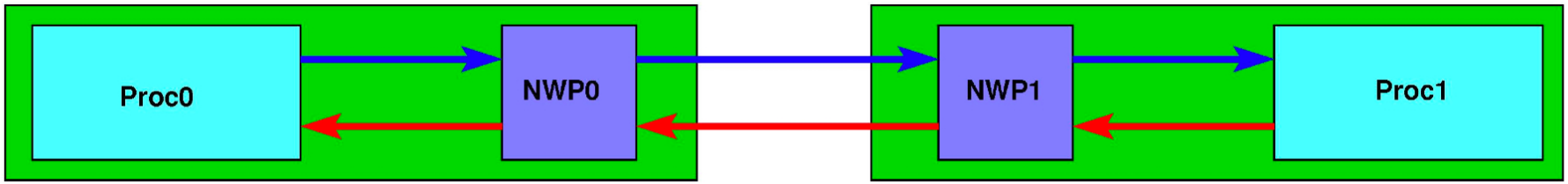
3DTorus, based on TWN (by M. Pivanti, F.S. Schifano, H. Simma)





- TNW had been made availabe to QPACE and has been made availbale to Aurora.
- In QPACE *FlexIO* interface by IBM; in Aurora everything is AuroraScience.
- TNW-project ported onto the Altera FPGA and processor interface adapted to Intel CPU.
- Easy loading of the configuration-firmware of the FPGA on the flash-memory.
- Driver and low-level library adapted and optimized for Intel CPU instruct. set.

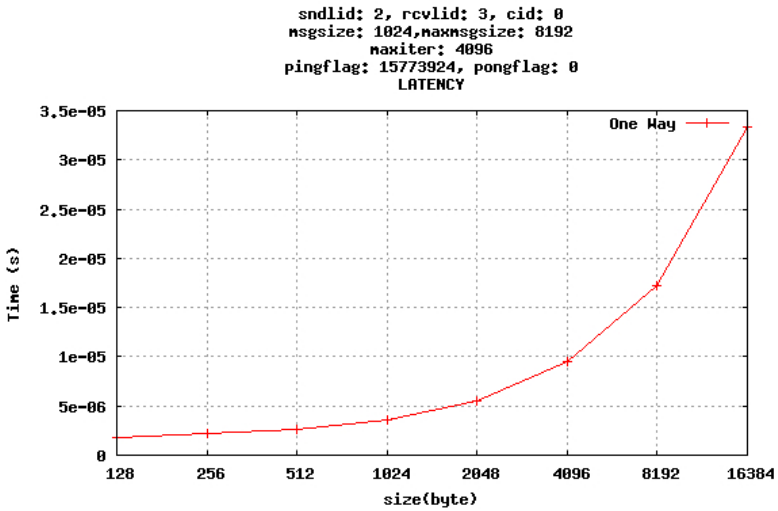
TNW (by M. Pivanti, F.S. Schifano, H. Simma)

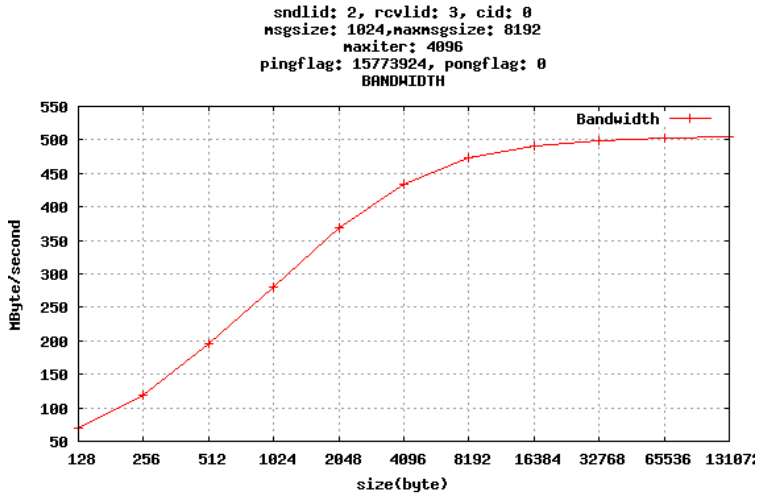


- Proc1 provides credit to NWP1
- Data are moved from Proc0 to NWP0
- NWP0 sends data to NWP1
- NWP1 checks data for errors and sends back a ACK/NACK
- NWP1 moves data to Proc1
- NWP1 notifies Proc1 that data are available

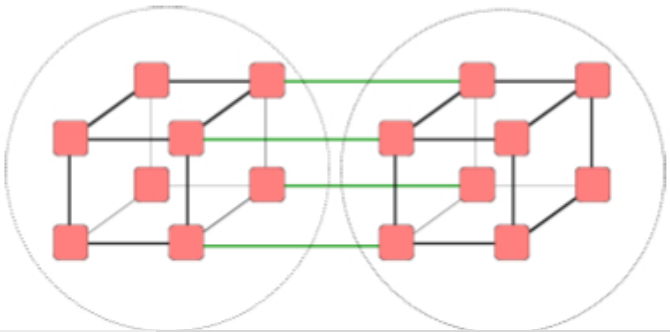
No explicit synchronization (barrier, rendez-vous) between Proc0 and Proc1

3DTorus, alive and kicking





- Bandwith measured by ping-pong and checked consistency in systolic communications. Checked aggregate bandwith.
- Communications tested along all the directions, with various payload and different *virtual channels*.

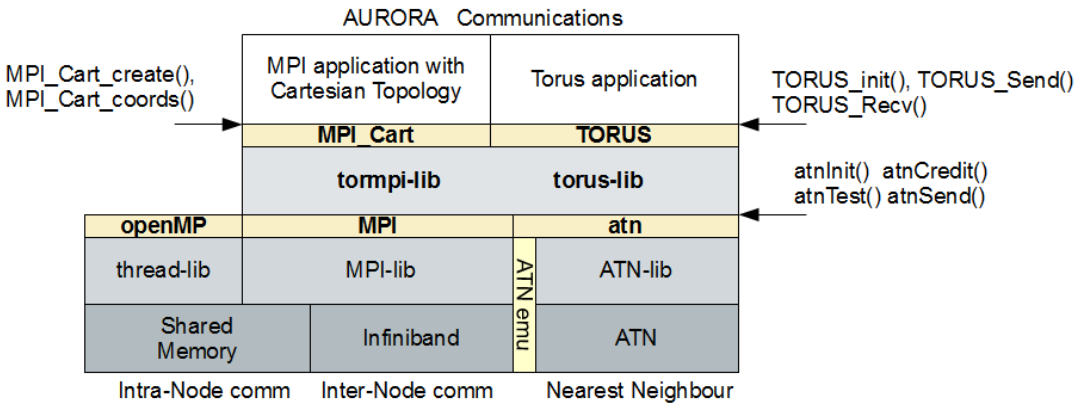


Programming Aurora communications

User programs can call both low level atn communication functions (threads) ...

```
int atnSend   (uint lid, uint cid, void * txbuf, uint txoff, uint len);
int atnCredit (uint lid, uint cid, uint rxoff, uint len, uint nid);
int atnPoll   (uint lid, uint cid, uint rxoff, uint len, void * rxbuf, uint nid);
int atnTest   (uint lid, uint cid, uint rxoff, uint len, void * rxbuf, uint nid);
```

... and high level TORUS or torMPI communication functions (processes).



- torMPI mimics MPI
- TORUS focuses nearest neighbor communications (3Dtorus + shared mem)
- they both at the moment rely on a *proxy* process
- TORUS + MPI can be better than MPI ...

A tour on the machine

```
[root@aserv1 ~]#
[root@aserv1 ~]# atninfo -h
/opt/aurora/bin/atninfo [-h] [-a] [-q] [-t] [-T]
a) List of active partitions
q) Queue status
t) Top processes on active partitions
T) Top processes on all nodes
[root@aserv1 ~]#
```

```
[root@aserv1 ~]# atninfo -q
List of queues :
   ROUTE-QUEUE  ENABLED QUEUED
   rq_16x1x1      no     0
   rq_4x1x1       yes    0
   rq_4x2x1       yes    0
   rq_8x1x1       no     0
   rq_8x2x1       yes    0

   EXEC-QUEUE         ENABLED   RUNNING
   eq_16x1x1_112          no      0
   eq_16x1x1_240          no      0
   eq_4x1x1_112           no      0
   eq_4x1x1_114           no      0
   eq_4x1x1_120           yes     0
   eq_4x1x1_122           no      0
   eq_4x1x1_240           yes     0
   eq_4x1x1_242           yes     0
   eq_4x1x1_248           no      0
   eq_4x1x1_250           yes     0
   eq_4x2x1_112           no      0
   eq_4x2x1_114           no      0
   eq_4x2x1_240           yes     0
   eq_4x2x1_242           no      0
   eq_8x1x1_112           no      0
   eq_8x1x1_120           no      0
   eq_8x1x1_240           no      0
   eq_8x1x1_248           no      0
   eq_8x2x1_112           no      0
   eq_8x2x1_240           no      0
   test           yes     0
[root@aserv1 ~]#
```

```
[root@aserv1 ~]#
[root@aserv1 ~]# atninfo -a
List of active  partitions:
4x1x1_120:  anode120 anode121 anode126 anode127
4x1x1_242:  anode242 anode243 anode244 anode245
4x1x1_250:  anode250 anode251 anode252 anode253
4x2x1_114:  anode114 anode115 anode116 anode117 anode122 anode123 anode124 anode125
4x2x1_240:  anode240 anode241 anode246 anode247 anode248 anode249 anode254 anode255
[root@aserv1 ~]#
[root@aserv1 ~]#
[root@aserv1 ~]#
```

- User can inspect status of the machine ...
- ... typically active partitions, queue status.
- There are *execution* and *routing* queues.

A tour on the machine

... and of course one has to decide at some point what partitions are active.

```
[root@aserv1 ~]#
[root@aserv1 ~]# atnadmin -h
/opt/aurora/sbin/atnadmin [-h] [-a] [-l] [q] [-t] [-d <part>] [-n <new>] [-s <part>]
l) List of available partitions
a) List of active partitions
q) Queues status
t) Top processes on active partitions
T) Test node clash among  partition <part> and active partitions
c) Configure closure on active partition <part>
d) Delete active partition <part>
n) New partition: activate and configure closure  <part>
r) Reset active partition <part>
s) Status active partition <part>
f) Force answer YES
[root@aserv1 ~]# _
```

A tour on the machine

Many options for *atnsub* command are actually oriented to TORUSlib applications.

```
[root@aserv1 ~]# atnsub

 Usage:

    atnsub -n <node_partition> -s <execution_script>

      [-l <local_partition>]
          local nodecard partition
          (useful for rankfile generation)

      [-f|-b|--forward|--backward]
          logical mapping of internal rank
          (default: forward)

      [-c <cpu_affinity>]
          e.g.: -l 3x2 -c 1,2,3,0,7,8
          (default: 1,2,3,4,5,6,0,7,8,9,10,11)

      [-N name]
          declares a name for the job
          (default: execution script's name)

      [-v <env_variable>]
          e.g.: -v VAR0,VAR1=x,VAR2=y
          (value of VAR0 grabbed from current env)

      [--tormpi]
          use the TORMPI rankfile generator

      [--bypass]
         read the vars inside atnsetup.conf and
         export that as env variables;
         also tell the TORUS library to read that
         from the env, instead from file
```

```
      [--proxy=2|4|6|8|10|any]
          atnproxy autostart on cores 6,11

          =2   support 2 procs,
               1 x socket 0 + 1 x socket 1 (-f -l 1x2)
          =4   support 4 procs,
               2 x socket 0 + 2 x socket 1 (-f -l 2x2)
          =6   support 6 procs
               3 x socket 0 + 3 x socket 1 (-f -l 3x2)
          =8   support 8 procs
               4 x socket 0 + 4 x socket 1 (-f -l 4x2)
          =10  support 10 procs
               5 x socket 0 + 5 x socket 1 (-f -l 5x2)
          =any support from 1 to 10 procs,
               no constraints on socket

      [--walltime=<time>]
          e.g, 15 min: --walltime=15:00

      [--err=<err_path>|--out=<out_path>]
          redefine path for error or output stream


Environment variables available for execution script:

    AURORA_JOBID     job's unique identifier
    AURORA_HOSTFILE  path to hostfile
    AURORA_RANKFILE  path to rankfile
    AURORA_NP        number of processing nodes
    AURORA_BTL       btl parameter for mpirun
```

Applications on/for Aurora

As it was clear from the list of research lines, there are a variety of applications to keep Aurora busy ...

- LQCD
  - ECT* group (TMQCD ported to TorMPI, Laplacian-Heaviside method for hadronic correlators, ...)
  - i.s. RM123
- Lattice Boltzman fluidodynamics (Ferrara)
- Spin glasses (Ferrara)
- NSPT (Parma)
- Diffusion MonteCarlo for Nuclear Physics (TN)
- Simulation of the dynamics of bio- and macro-molecules by Dominant Reaction Pathway (DRP) methods (TN)
- Genomics (FEM, with major controbutions from DEI-PD)
- Parallelization of Geant4 (ATreP)

Lattice Boltzman - our recordman

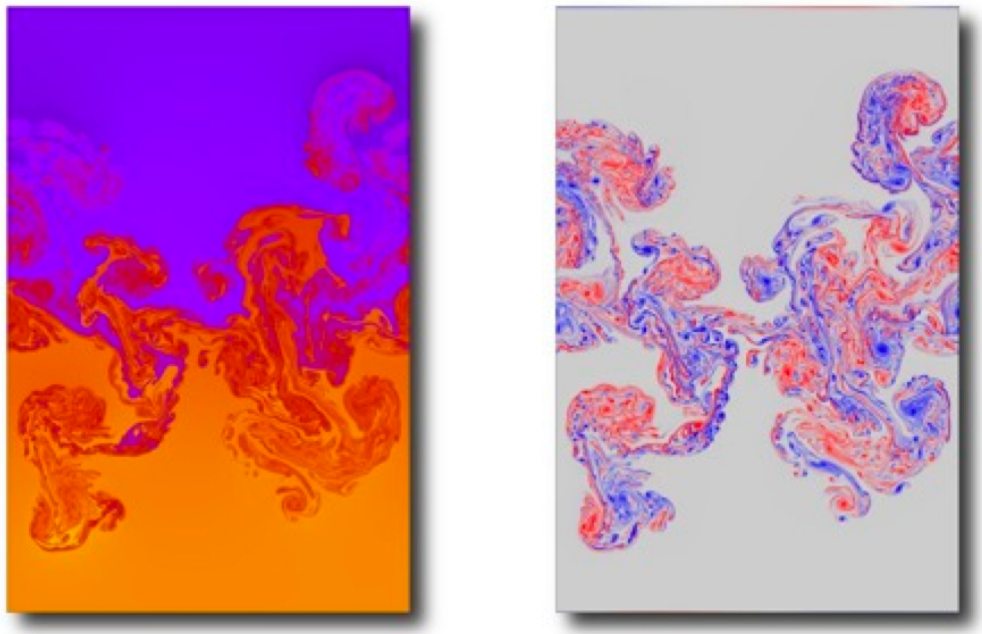


Figure 3.2.5. Snapshots of temperature and vorticity in the evolution of a Rayleigh-Taylor instability, simulated on a grid of 512 × 1000 sites, using 8 processing nodes.

Up to 38% efficiency!

NSPT - an example that profits from both networks

The master formula for inverting (order by order) the Dirac operator:

$$
\begin{aligned}
\psi^{(0)} &= M^{(0)\,-1}\xi \\
\psi^{(1)} &= -M^{(0)\,-1}M^{(1)}\psi^{(0)} \\
\psi^{(2)} &= -M^{(0)\,-1}\left[M^{(2)}\psi^{(0)} + M^{(1)}\psi^{(1)}\right] \\
\psi^{(3)} &= -M^{(0)\,-1}\left[M^{(3)}\psi^{(0)} + M^{(2)}\psi^{(1)} + M^{(1)}\psi^{(2)}\right] \\
&\cdots \\
\psi^{(n)} &= -M^{(0)\,-1}\sum_{j=0}^{n-1} M^{(n-j)}\psi^{(j)}
\end{aligned}
$$

We have 12 cores on a node and we set up 3 processes on each node

- $M^{(0)\,-1}$ trivial in momentum-space
  - 2 cores (2 processes) devoted to FFTW ...
  - ... rearranged in a 1-dim ring MPI FFTcommunicator, via Infiniband
- $M^{(i)}$ easy in configuration space
  - 8 cores computing in a multi-thread scheme
  - 1 core devoted to MPI-RMA data to/from MPI FFTcommunicator
  - 1 core devoted to Torus communications

What now?

- 10 TFlops up and running, extra 10 to come (first phase), made available to many applications
- LQCD is working: easy to *enter and run* for people who have not developped the system

... and now?

- We will have SandyBrdige *engineering samples* (late?) in summer time ...
- ... but we propose the installation of a significant machine quite soon.
- There is a window of opportunity, with current technology!
- With a moderate (shared) economic effort, we can get back to having a significant HPC installation in Italy.