Evoluzione del Calcolo Scientifico per la Fisica Teorica

F. S. Schifano

[1] University of Ferrara and INFN-Ferrara

Workshop Commissione Calcolo e Reti INFN 2008 LNGS
June 10^{th} – 13^{th} 2008

Outline

Evolution of scientific computing for theoretical physiscs:
custom vs commodity computing systems

- Spin-Glass and LQCD: two computing challenge theoretical physics problems
- Spin-Glass and LQCD Engines: past, present and future
- conclusions

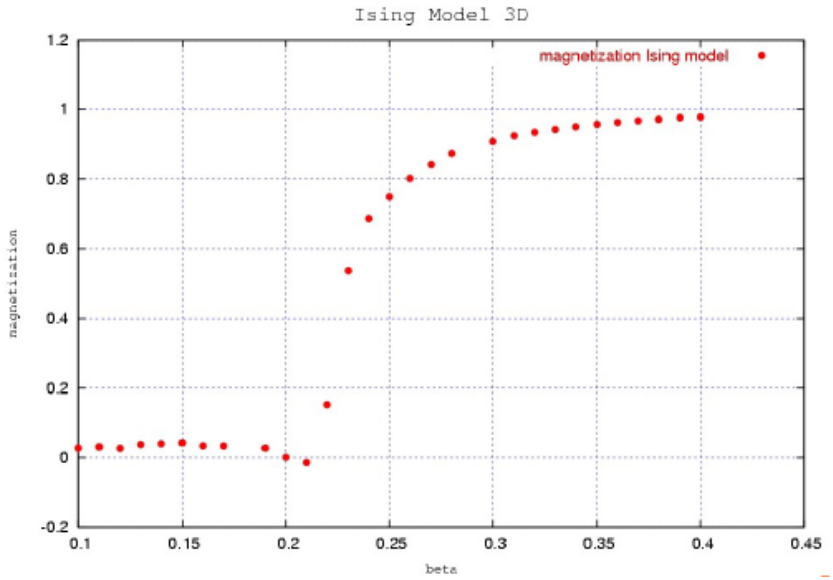
Disclaimer

Standard computing for theoretical physics, e.g. use of commercial systems possibly via a GRID infrastucture (THEOPHYS) is obvious for all of you and not covered in this talk.

Spin-Glass

The Spin-glass model is a statistic model proposed as a **toy** model to study some behaviours of the most complex macroscopic systems, e.g.:

- disordered magnetic materials
- real flow glasses (e.g. Notre-Dame windows)

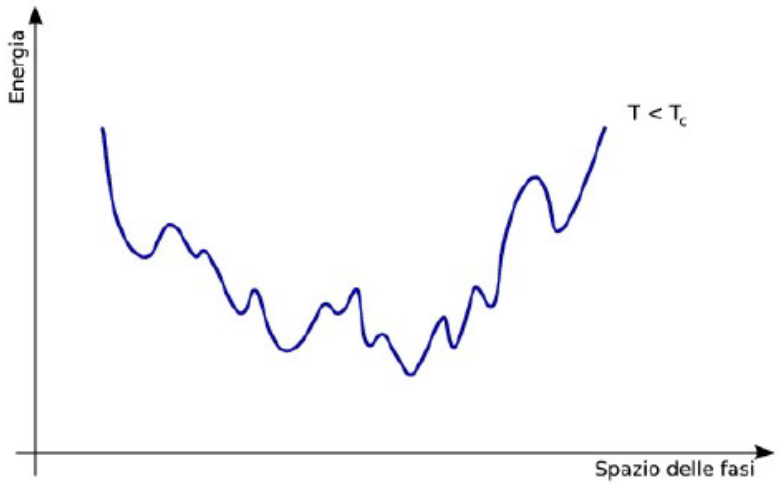


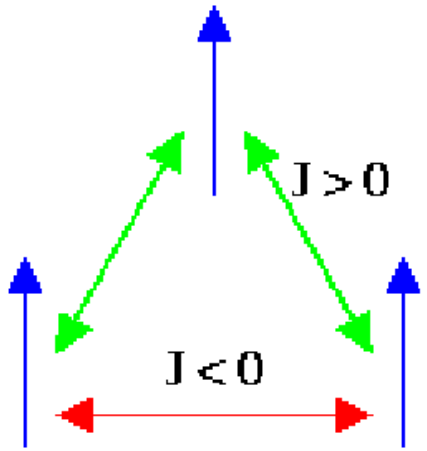
for example: transition temperature of magnets beyond which they lose their magnetic state, explained in terms of elementary magnetic dipoles (spins) attached to each atom in the material.

Spin-glass is a complex problem(1)

Averaging physics observables over all possible configuration S of the discrete 3D lattice of spins is extremely difficult due to **frustation** effects. The energy function:

$$E(\{S\}) = -\Sigma_{\langle ij \rangle} J_{ij}\sigma_i\sigma_j, \quad \sigma_i, \sigma_j \in \{+1, -1\}, \quad J_{ij} \in \{+1, -1\}$$

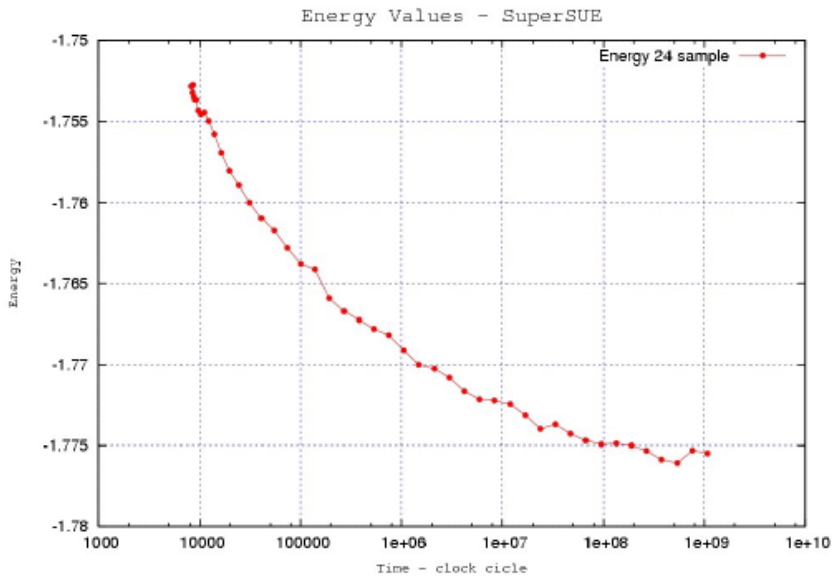




Spin-glass is a complex problem(2)

To bring a system of $48^3, 64^3, 80^3$ lattice points to thermal equilibrium:

- the system must be followed over 10^{12}, 10^{13} MonteCarlo steps
- on $\mathcal{O}(100)$ indipendent system (replicas)



Spin Glass Simulation Algorithm: Ising Model

Spin are represented as bit-variable arranged on a 3D lattice, only first neighbour interactions are considered:

1. compute the local energy E, summing all contribution from nearest neighbor spins (and taking the corresponding coupling into consideration);
2. flip the value of the spin $\sigma' = -\sigma$
3. compute the new local energy E'
4. compute the energy change: $\Delta E = E' - E$
5. if $\Delta E < 0$ the new value of the spin σ' is accepted
6. if $\Delta >= 0$ then the new state is accepted if $\rho < e^{-\beta\Delta}$, where ρ is a pseudo-random number ($0 \leq \rho \leq 1$) and β is defined as the inverse of the temperature.

The Ideal Spin Glass Engine

- an orderly structure (e.g. a 2D grid or 3D-grid) of a large number of **update engines**
- each update engine executes the same algorithm on a different subset of the physical mesh (**SIMD computing**)
- its architectural structure has to be extremely simple:
  - the data path processes one bit at a time
  - memory addressing is regular and predictable (**data prefetch**)
- memory bandwith requirements are **huge**: 7 (or more depending by the model) bits are necessary to process one single bit
- memory should be local to the processor

A Realistic Spin Glass Engine: The PC Approach

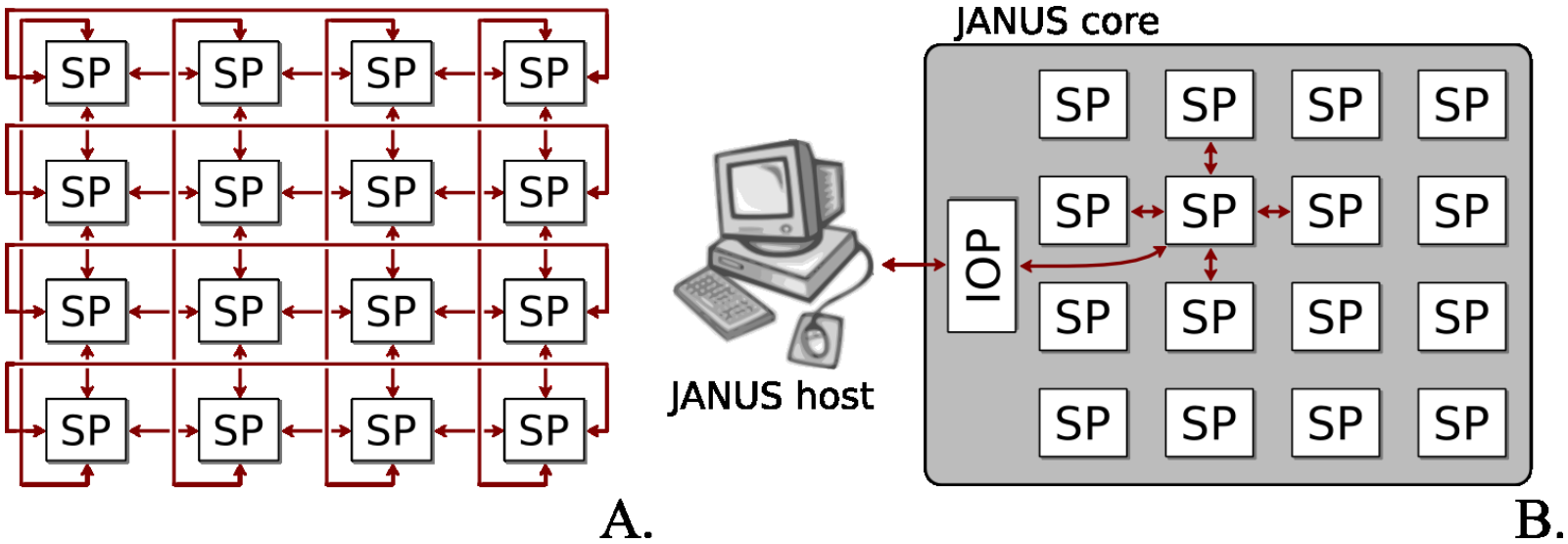
Using commercial PC as spin-glass engines: two different algorithms are commonly used:

- the Synchronous Multi-spin Coding (SMC):
  - one CPU handles one single systems
  - replicas are handled on a farm of several CPU ($128 - 256$)
  - at every step each CPU update in parallel ≤ 4 spins (bottleneck is the number of floating point random numbers can be generated in parallel)
- the Asynchronous Multi-spin Coding (ASMC) approach):
  - each CPU handles several ($64 - 128$) system in parallel
  - replicas are handled on CPU farm (smaller than previous)
  - at every step each CPU update in parallel $64 - 128$ spins of different systems
  - on each single CPU radom number is shared among all systems

A Realistic Spin Glass Engine: The JANUS Approach

The basic hardware elements:

- a 2-D grid of 4 x 4 FPGA-based processors (SP's)
- data links among nearest neighbours on the grid
- one control processor on each board (IOP) with 2 gbit-ethernet



JANUS is a project carried out by BIFI, University of Madrid, Estremadura, Rome and Ferrara, and by Eurotech.

Janus vs PCs Performance(1)

- traditional architecture boost performance by processing several **replicas** of the system, mapped on the bits of their **long** data words (AMSC code)
- the same random number is shared by all replicas (introduce nasty but manageable correlations)
- however too-many replicas (> 256) become quickly useless and SMSC is required

System	SMSC	AMSC	Note
Janus core (16 FPGAs)	1ps	-	
Intel Core Duo 2, 2.4 GHz	3000ps	700ps	128 replicas
IBM Cell 1 SPE, 3.2 GHz	-	224ps	32 replicas

Note: performance on Cell-BE are achived assuming the lattice size fits all in the LS: $L < 15^3$ not useful for real simulations.

Janus vs PCs Performance(2)

Assuming to perform 10^{12} montecarlo steps on a lattice size of 64^3 we have:

	Janus	AMSC	SMSC
processor	1 SP	1 CPU	1 CPU
statistic	1 (16)	1 (128)	1 (4)
wall-clock time	50 days	770 years	25 years
energy	2,7 GJ	2,3 TJ	78,8 GJ
processor	256 SPs	2 CPUs	256 (64) CPUs
statistic	256	256	256
wall-clock time	50 days	770 years	25 years
energy	43 GJ	4,6 TJ	20 (5) TJ

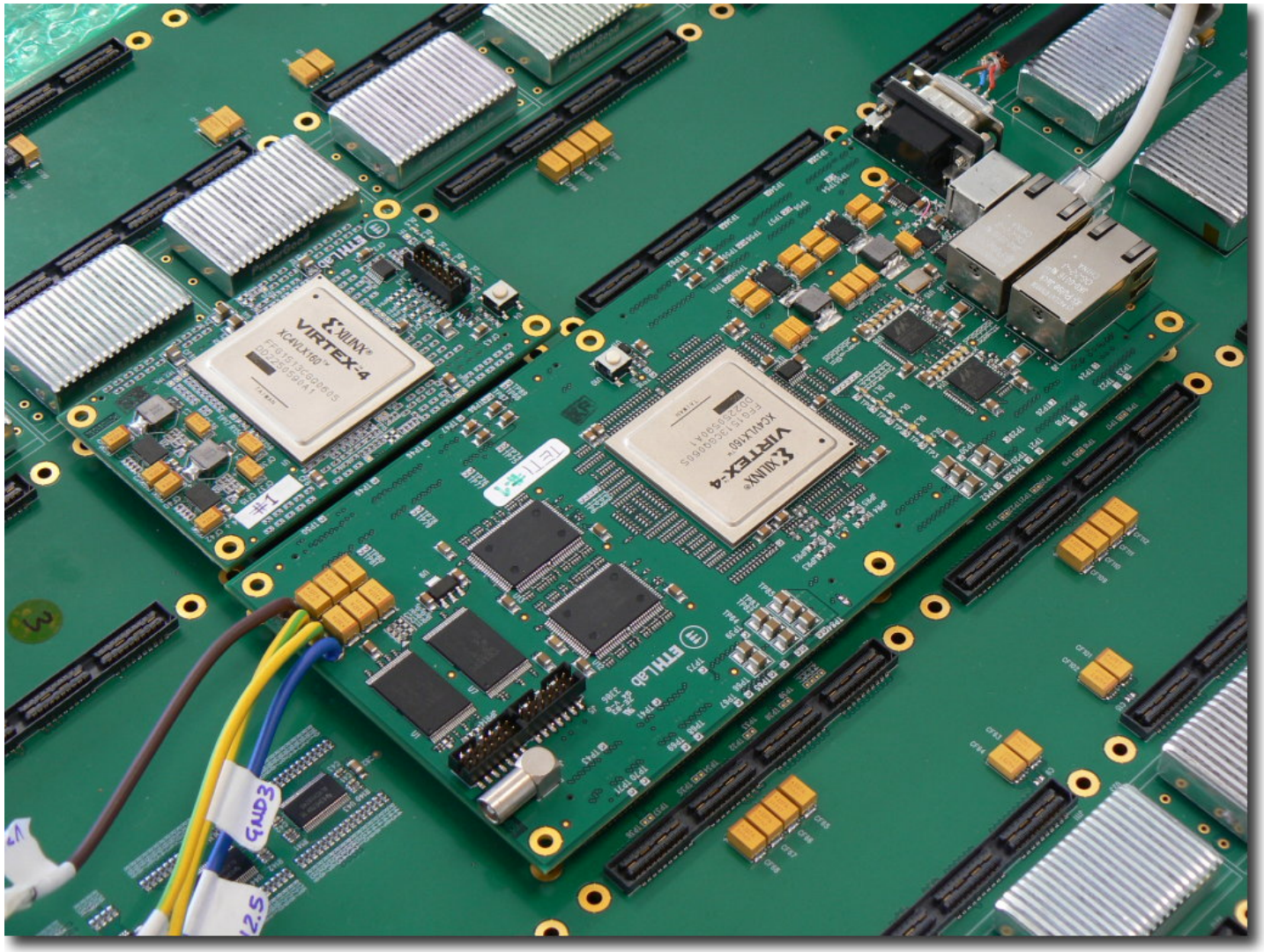
Tipical number of replicas is 256, using more than 256 CPUs is physically not very interesting !

1 SP 40 W, 1 Janus-core 640 W, 1 PC 100 W

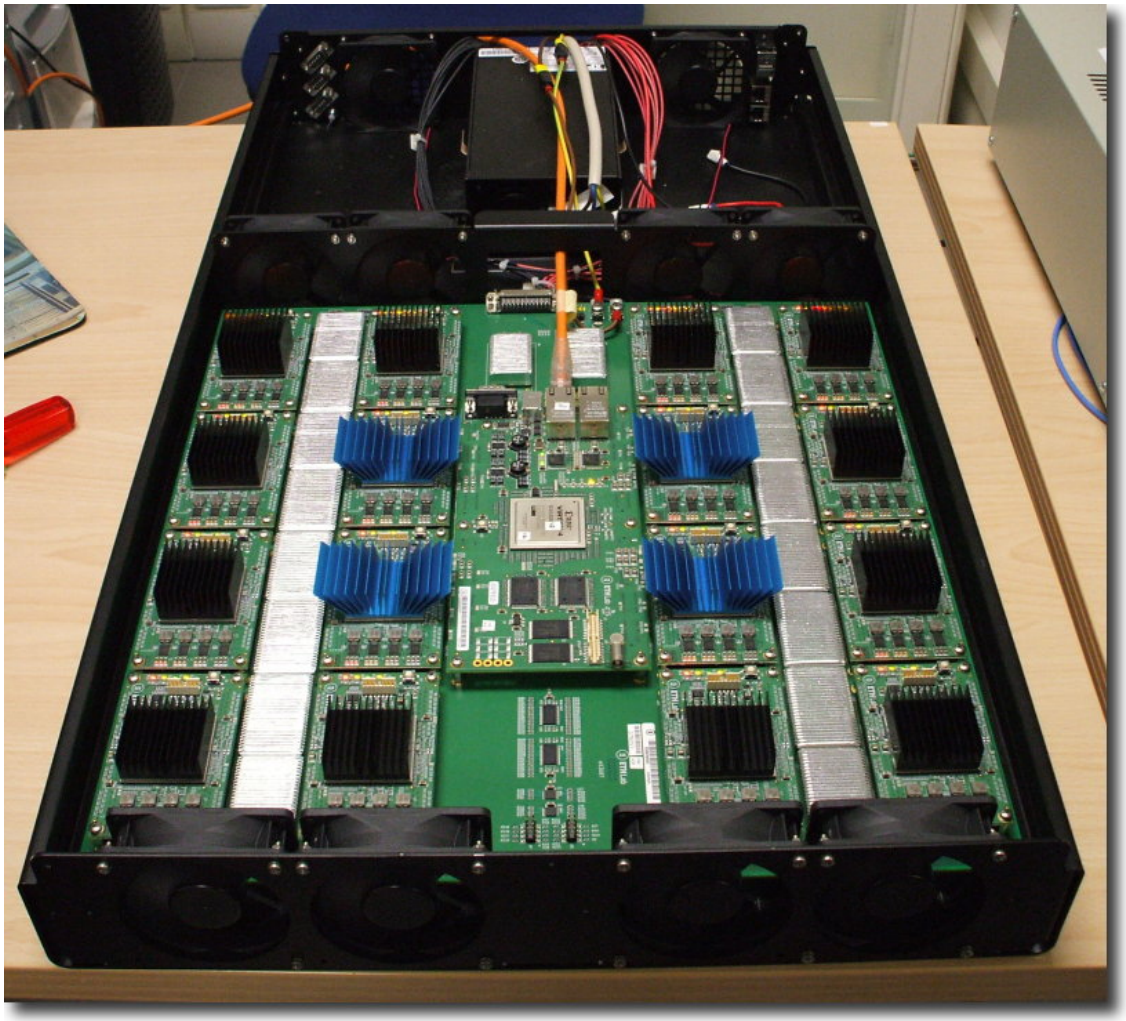
The Janus SP


XILINX
VIRTEX-4
XC4VLX160
FFG1513CGQ0605
DD2250590A1
TAIWAN
ETHLab
PC110610SL

The Janus IOP



The Janus core module



16 SP + 1 IOP

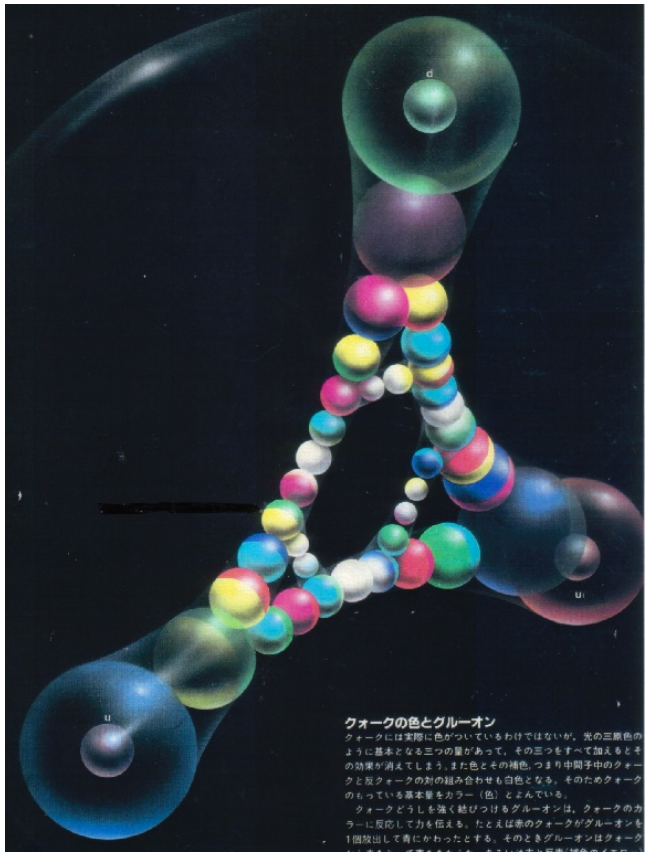
The Janus rack

16 JANUS core, 256 SPs

Quantum Chromodynamics on the Lattice (LQCD)

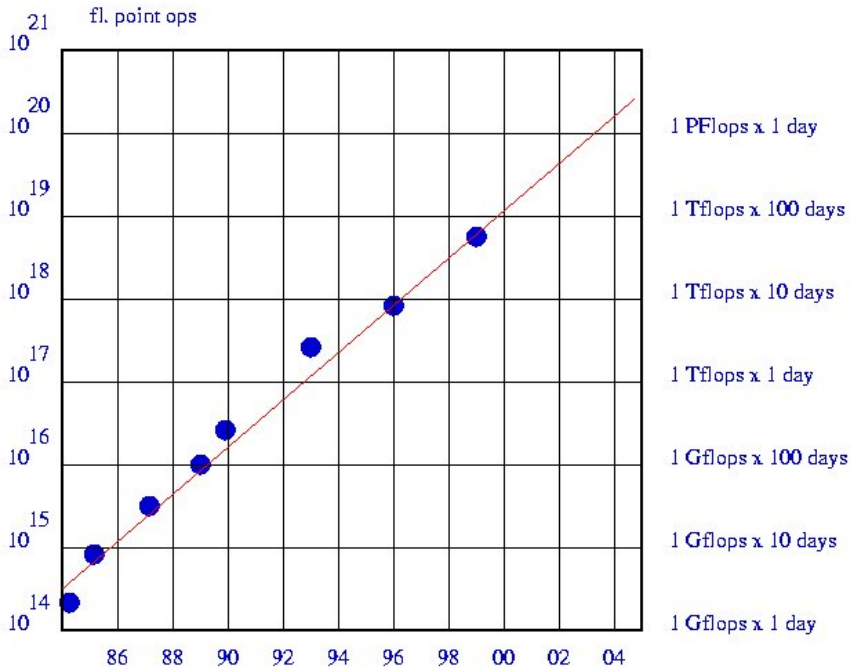
QCD describes the behaviour of hadrons (proton, neutron, ...)
$\Rightarrow$ important ingredient to understand high-energy experiments

- theory of strong interactions between
  - quarks (matter)
  - gluons (forces)
- quantitative predictions require numerical approaches
- Monte Carlo simulations on discrete 4D space-time lattice: $N > 10^7$ sites



The Computing Challenge of LQCD

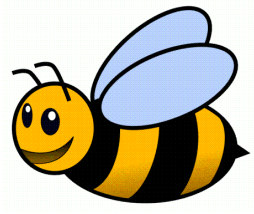
Petaflops performance is needed around 2010



Main computational cost: matrix-vector multiply (Dirac operator)

- structured sparse matrix ($12N \times 12N$, 8 non-zero entries/row)
- $W = N \cdot 1320$ FP operations (mainly complex arithmetics)
- $I = N \cdot (48 \ldots 168)$ complex operands (from memory)

Array Processor Experiment (APE)



Started in the mid 80s at INFN to provide computing resources for LQCD simulations.

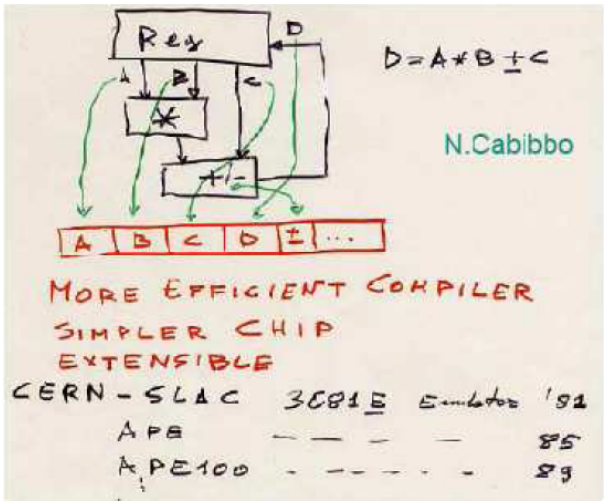
- efficient complex FP arithmetics:

$$a \times b + c, \qquad a, b, c \in \mathbb{C}$$

- accurate balance between computation and memory access or communication:

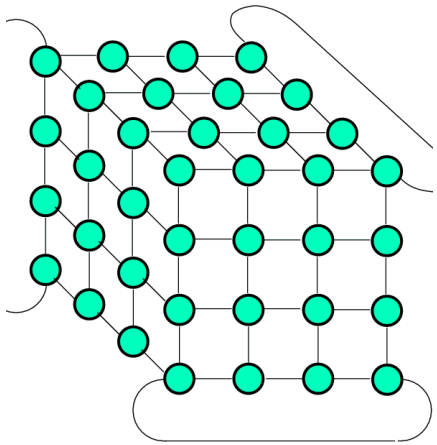
$$W(N)/P = I(N, m)/B$$

- nearest neighbour communications



Optimization of the APE Architecture for LQCD

- 3D torus network
- slow clock
  $\Rightarrow$ **low power**
- integrated memory and communication interface
  $\Rightarrow$ **compact design**
- Very Long Instruction Word (VLIW) architecture
  $\Rightarrow$ **optimized scheduling at compile-time**
- large register file instead of cache
  $\Rightarrow$ **predictable and synchronous execution**
- RAS (ECC, status registers, . . .)
  $\Rightarrow$ **run-time of single program execution** $\mathcal{O}(days)$



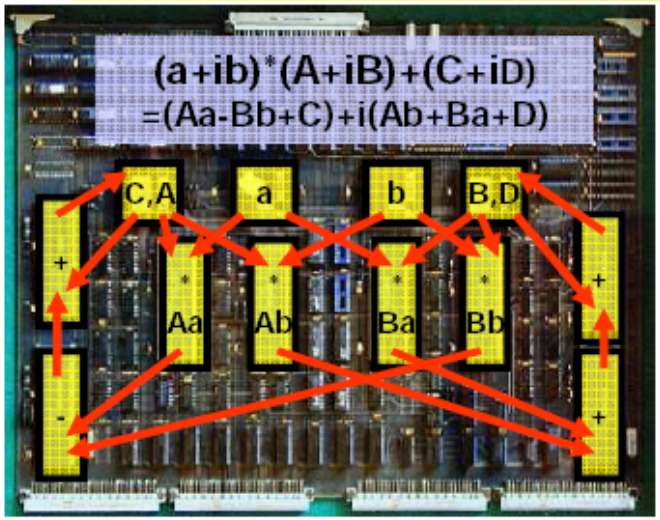
History of LQCD Machines

	EU	US	Japan
1990 - 1995	APE100 300 Gflops	Columbia2	
1995 - 2000	APEmille 2 Tflops	QCDSP	**CP-PACS**
2000 - 2005	apeNEXT 15 Tflops	QCDOC	**PACS-CS**
2005 -	—	**BG/L,P,Q**	—

- Since 1996 APE has become an international European collaboration (France, Germany, and Italy)
- $>$ 25 of the 100 most relevant publications on LQCD since 1995 have been obtained APE machines (20 out of the 79 most cited articles, Spires)

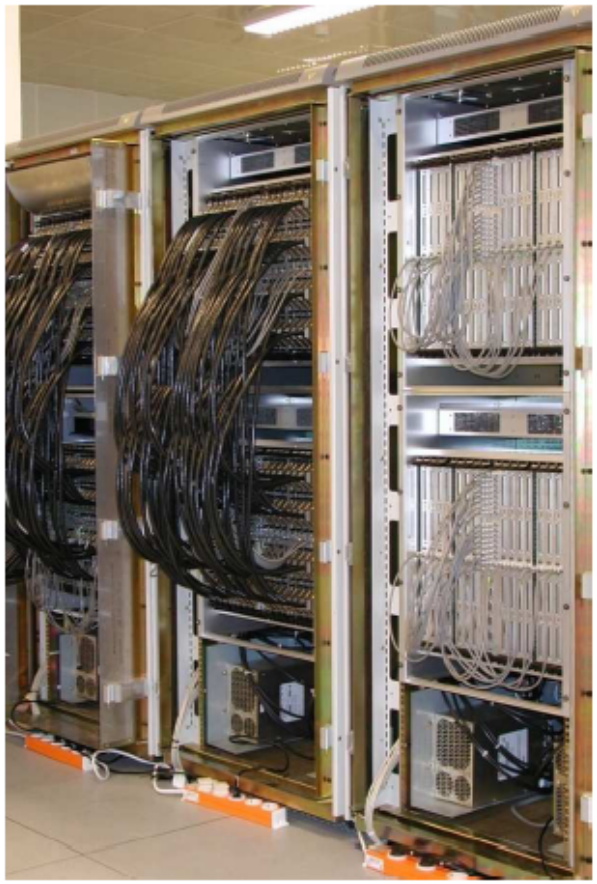
Today 15 Tflops apeNEXT are installed in Europe (DP, $\varepsilon \approx 40\%$)

From APE1 to apeNEXT

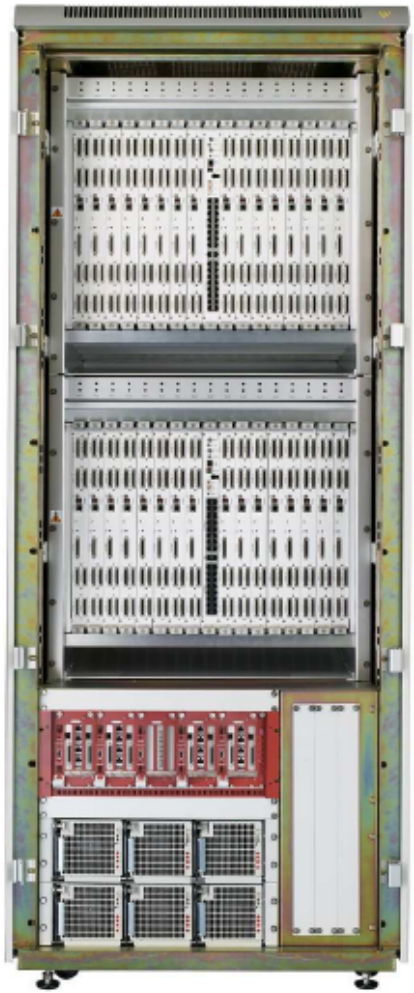


APE1 (1988) 1GF

APE1000 (1992) 25GF, SP, REAL



APEmille (1999) 128GF, SP, Complex



apeNEXT (2004) 800GF, DP, Complex

Goals for Future Machine (2010)

- high integration: 100 Tflops / m^3 (peak)
- power efficiency: 250 W / Tflops (peak)
- price / performance: $\approx$ 5 M€/ Pflops (peak)

Key elements:

- high FP-performance processor (400 Gflops)
- processor directly coupled to network by fast IO interface
- nearest-neighbours network, 3D torus

How to Reach the Goals ?

Current wisdom for the processor: custom design of processor ASIC no more **competitive**/needed:

- commodity processors have become efficient for QCD (SSE, cache-aware algorithms, ...)
- commodity processors allow scalable system architectures (power consumption, integrated memory and IO interface)

Possible alternatives are:

- use BlueGene[L,P,Q] systems . . . or . . .
- . . . do **better**(?) and interconnect commodity processors like
  - GPU
  - IBM Cell-BE processor
  - new generation of multi-core Intel processors

  through a custom network directly coupled to the processor.

The Blue Gene Project

Around 2003-2004 the idea that QCD machines are not just a toy is endorsed by IBM which started the Blue Gene Project.
Basic elements:

- derived from QCDOC
- based on the idea of interconnecting simple and low-power processors (PowerPC) through a 3D torus network
- huge investiments in porting a large set of applications
- BG/L (first generation) marginally cheaper than dedicated machines
- carries the big blue brand ...

Today 220 Tflops BlueGeneP are installed in Jülich (Germany), 1/4 used for LQCD (DP, $\varepsilon \approx 30\%$).

Can we have such installation in Italy ? However . . .

. . . is the price of BlueGeneQ going to be affordable in 2010 ?

Graphics cards

- Developed by Universities of Budapest and Wuppertal
- Very cheap, but
  - Precision/rounding issues
  - Not integrated into scalable architecture (yet)
  - Programming challenges

QPACE: Project Goals

- QPACE = **QCD PA**rallel computing on **CE**ll
- Design of a massively parallel QCD prototype (with suitability for other applications in mind)
- Key components:
  - **Enhanced Cell BE processor** PowerXCell8i
  - **Custom network processor**
  - custom boards and system integration
- Timescale
  - End 2008/early 2009: small prototype running
  - Spring 2009: large prototype at $O(400)$ TFlops peak installed
- Secondary goals:
  - Gain experience with a multi-core CPU in a massively parallel environment
  - Ensure availability of strong-scaling architectures

QPACE collaboration / Credits

- **Academic partners**

University of Regensburg	N. Meyer, S. Solbrig, T. Wettig
University of Wuppertal	Z. Fodor
DESY Zeuthen	D. Pleiter
Research Lab Jülich (FZJ)	M. Drochner, N. Eicker, Th. Lippert
University of Ferrara	F. Schifano, R. Tripiccione
University of Milano	A. Nobile, H. Simma
University of Padova	G. Bilardi

- **Industrial partner**

IBM Development Lab Böblingen	G. Goldrian, O. Wohlmuth

Performance of Dirac Operator

Local $4^3 \times L_0$ lattice on each SPE ($L_0 = 16, 32, 64$)

Model:

	cycles
T_{FP}	$10.6\ 10^3 \cdot L_0$
T_{LS-LS}	$2.3\ 10^3 \cdot L_0$
T_{LS-MM}	$32.8\ 10^3 \cdot L_0$

Benchmark: LS-MM access only

	QS20			CAB		
L_0	10^3 clk	ε_{MM}	ε_{FP}	10^3 clk	ε_{MM}	ε_{FP}
16	41.6	79 %	**25 %**	41.7	79 %	**25** %
32	39.7	82 %	27 %	40.0	82 %	26 %
64	39.1	84 %	27 %	43.7	75 %	24 %

For $L_0 = 32$ and 4-KB pages, each SPE needs at least 208 TLB entries

QPACE overview

- **Nodes:**
  - PowerXCell8i Enhanced Cell BE
    - ★ 100 GFlop/s double precision peak
    - ★ IEEE rounding
  - 4 GBytes external DDR memory
- **Network:**
  - Custom designed
  - nearest-neighbour communication, 3-dimension torus topology
  - Design goals:
    - ★ 1 GBytes/sec per link and direction
    - ★ LS-to-LS DMA capabilities with $< 1\mu$sec latency
  - FPGA based
- **System:**
  - Custom designed

QPACE overview (2)

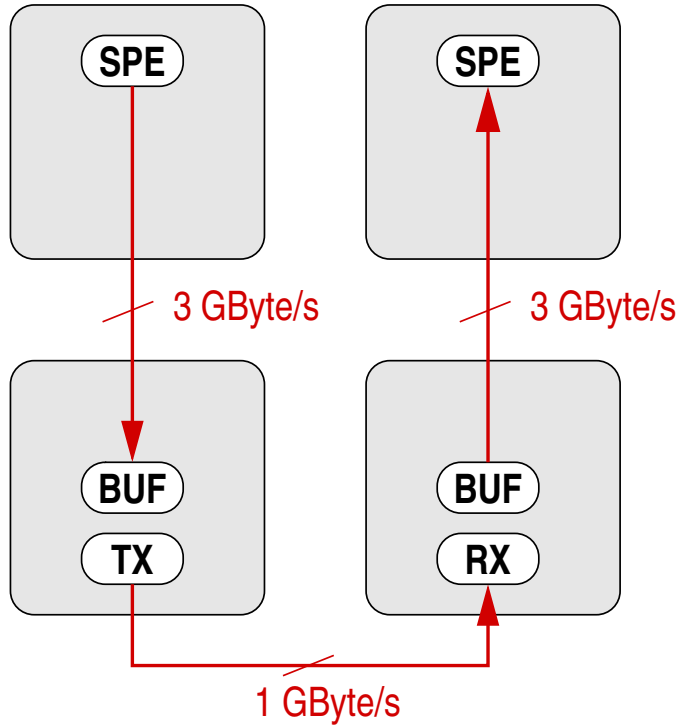
System parameters:

# CBE	$4 \times 2 \times 2$	$8 \times 4 \times 4$
Performance (peak, DP) [TFlops]	1.6	13
Power consumption [kW]	2.4	19
TFlops/kW (peak, DP)	0.7	

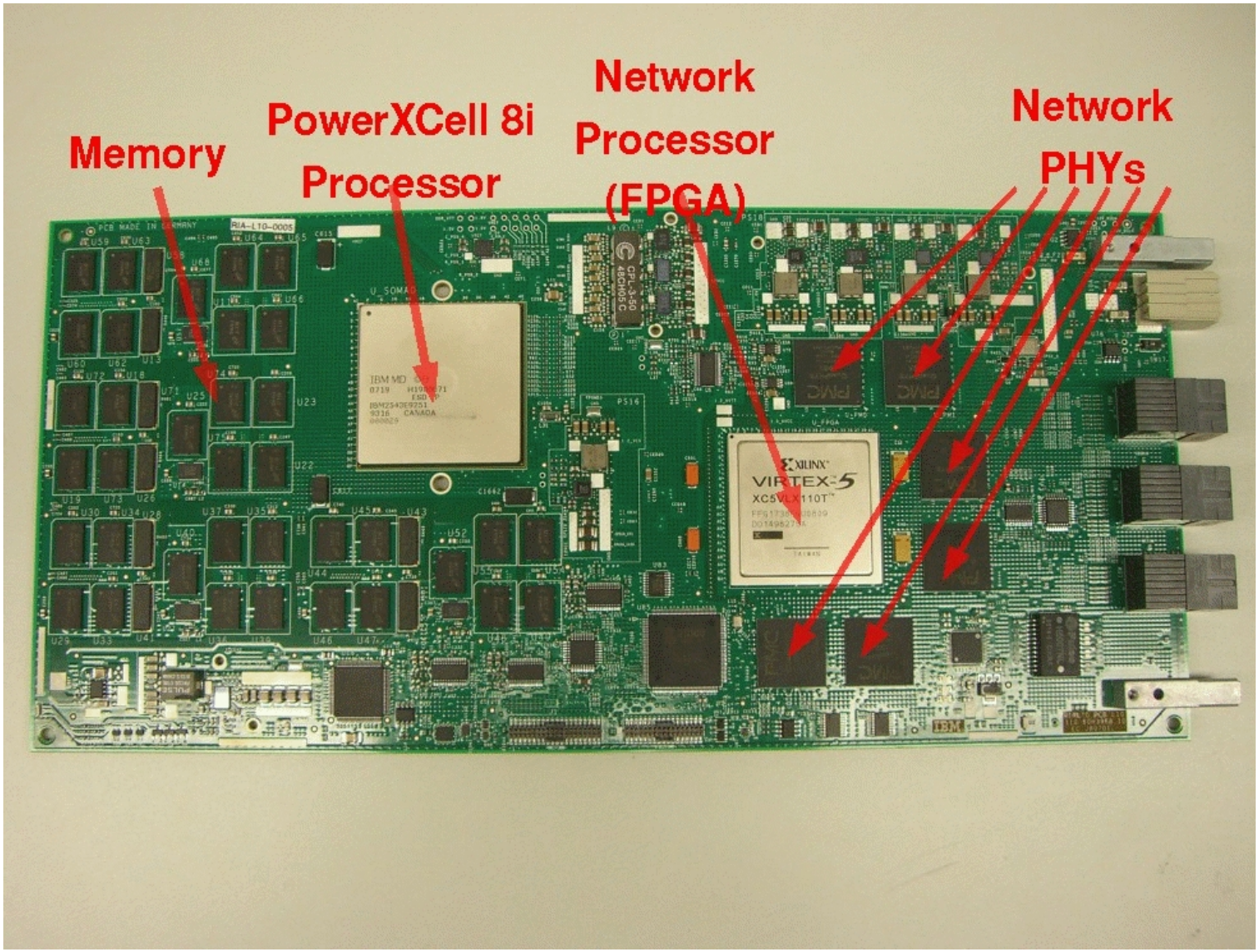
Application optimised network

Operations needed for communication:

- **Source: SPE performs DMA put to I/O device (=network processor)**
- Data is moved via send FIFO to sender
- Data is received and moved to receiver FIFO
- **Destination: DMA from I/O device to LS of SPE**



QPACE node card



The Aurora Project

As for QPACE, Aurora project aims to use new generation of commodity processors: the low-power multi-core processors of Intel.

Key elements:

- low power chip $\approx 80W$
- $\approx$ 100 GFlops DP peak per chip
- large cache $24MB$ per chip reducing requirements for network systems
- (more) standard programmability
- **out-of-order execution**
- **less control of the cache**

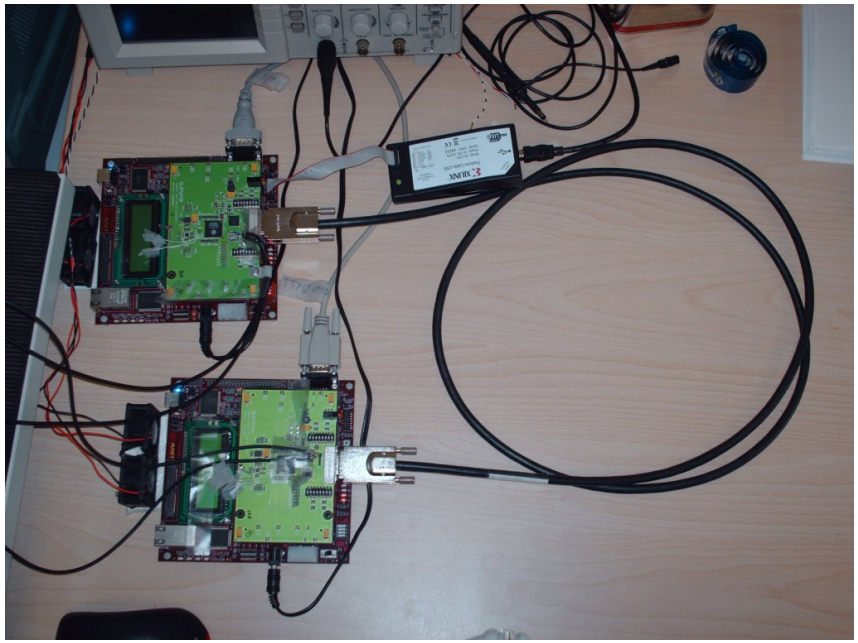
Performance for the Dirac operator is estimated $\approx$ 25% of peak.

Aurora is the poor man's QPACE: INFN + Italian Universities + Eurotech
. . . not yet approved, many details to be settled . . .

Network Processor

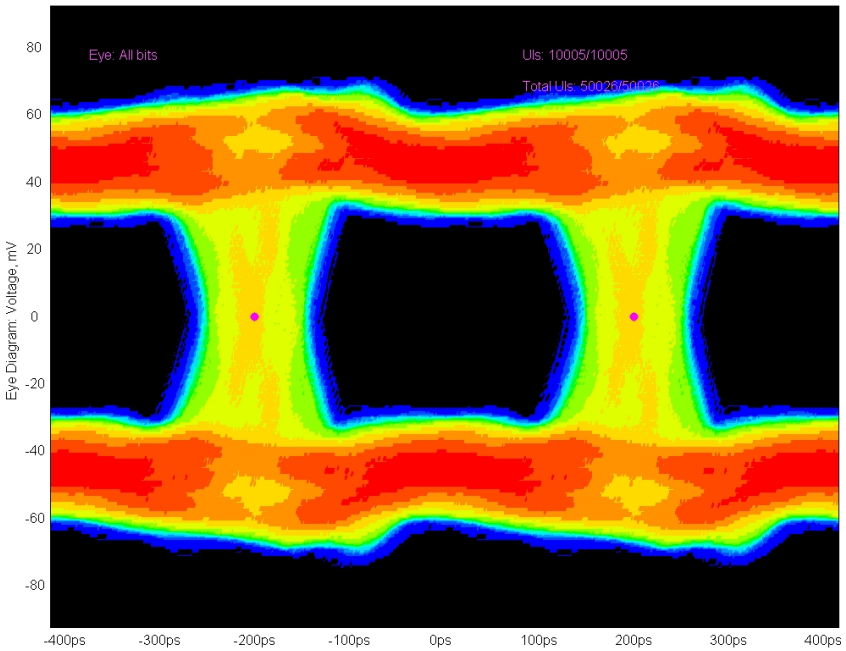
Both projects aim to interconnect commodity processors through a custom high speed, low latency 3D mesh torus network (**a la APE**).

- PCIe (10Gb) technology
- external PHY GL9714 (PMC8358), latency: $80(116) \times 4ns$
- FPGA based network processor

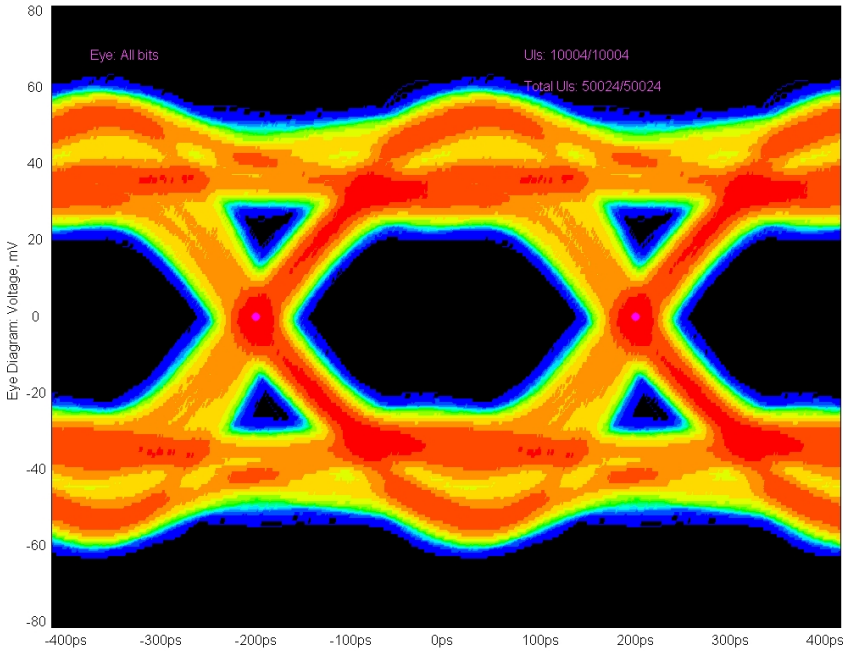


Test della Tecnologia dei Link

TX-eye: $\approx$ 550 mV



RX-eye: $\approx$ 420 mV



Summary and Conclusions

Spin Glass:

- commodity processors NOT provide the needed computing power
- **dedicated system** is the best approach

Lattice QCD:

- **past**: **dedicated systems** were the only viable choise
- **present**: the most used system is BlueGeneP
  - bunch of low-power simple processors
  - interconnected by a 3D-mesh torus
  - big effort on program porting
- **future**: LQCD ASIC design in not any more convenient/handy
  - use new generation of multi-core processor: low-power, large on-chip memory, high perfromance
  - interconnect them by a custom 3D torus network derived from experience of past Lattice-QCD machines