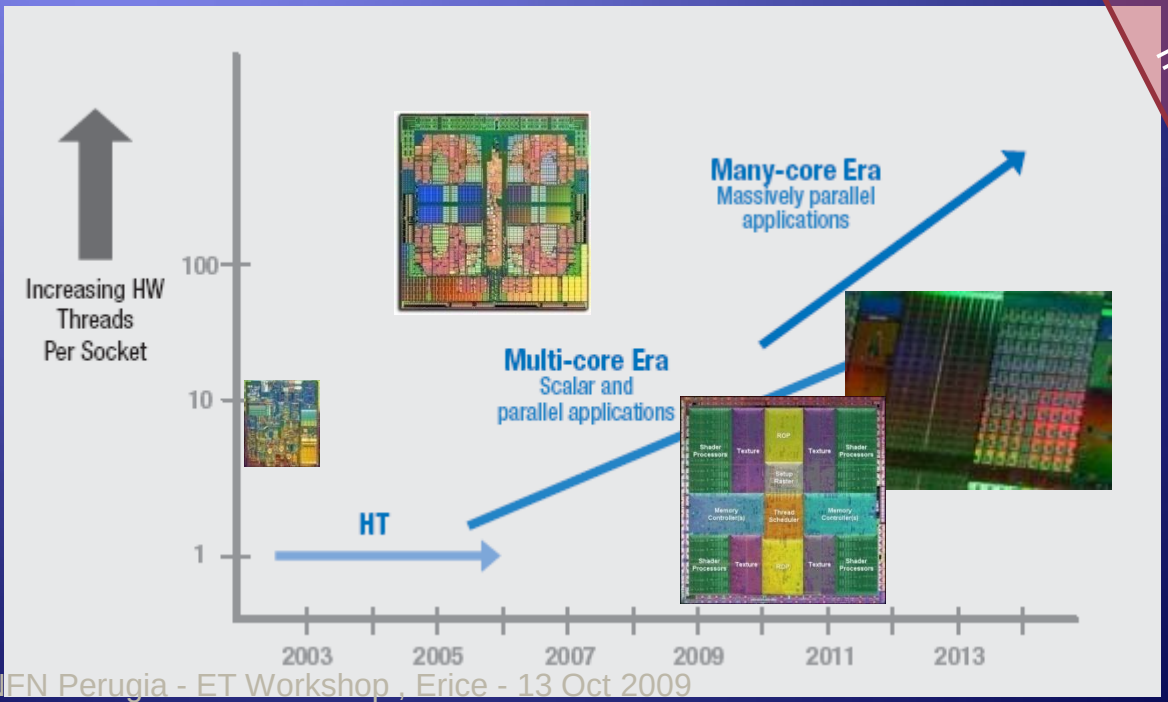
cuInspiral: a GPU high arithmetic intensity library for coalescing binaries detection.

Dr. Leone B. Bosi – INFN Perugia

2° Annual ET Workshop – Erice(TP) Italy
13 October 2009

Technological outlook

- Most important chip semiconductor maker are working in order to limit problems due to integration scale reduction.
- In fact last 10 years the processors architectures are changed a lot, introducing parallelization at several architectural levels.
- That evolutive process will continue in a deeper manner, moving to the so called “many-core” era.

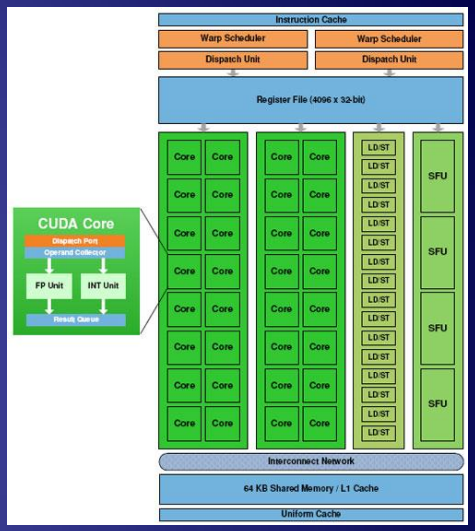


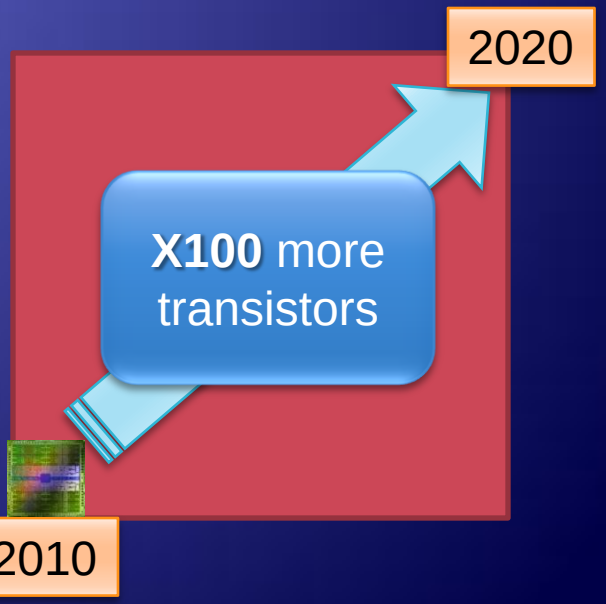
Ref. to my talk 2-Oct ET F2F meeting for more info.

Intel ref.

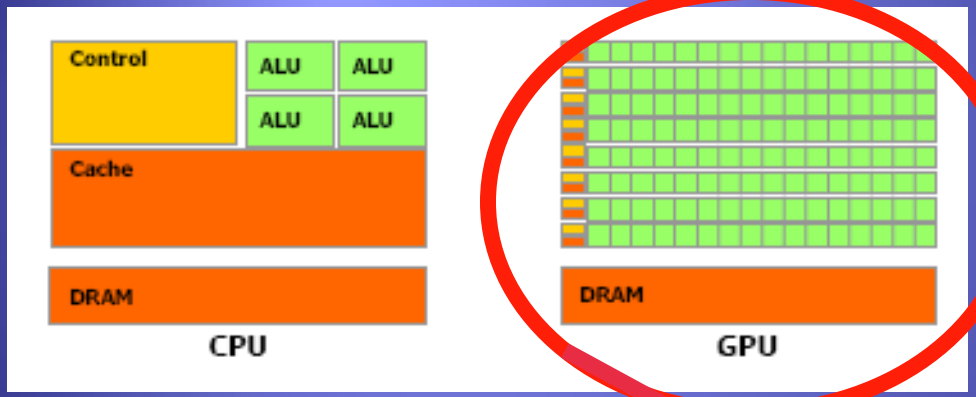
Some performance considerations:

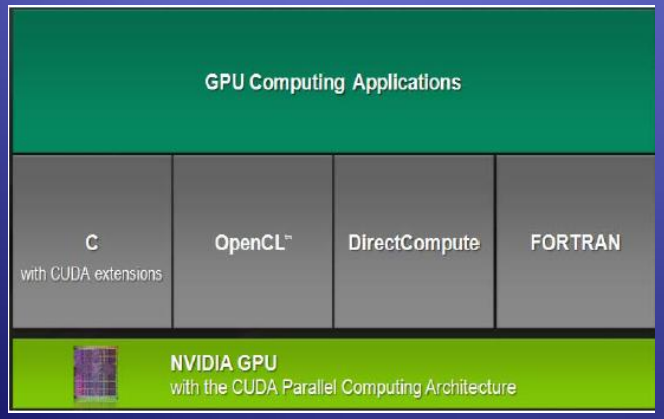
- These new architectures require a complete different programming models.
- In the close future, in the manycore era, computing power will be distributed across multiple cores (10000>?) on a single processor, and many processors on a single board.
- Performance achievable from these architecture is not predictable because depends on algorithm and relation with:
  - Memory/registries architecture model
  - Intercommunication
  - Serial portion of the algorithm

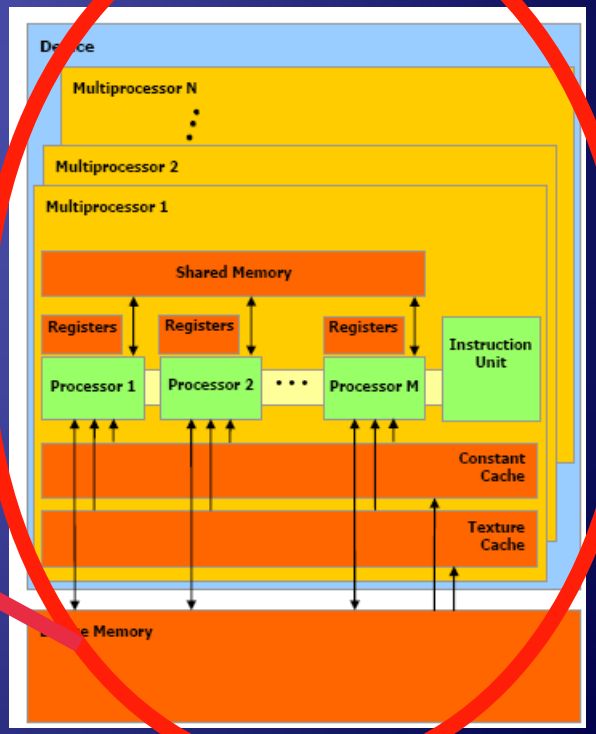




CPU – GPU (nvidia) architectural model comparison

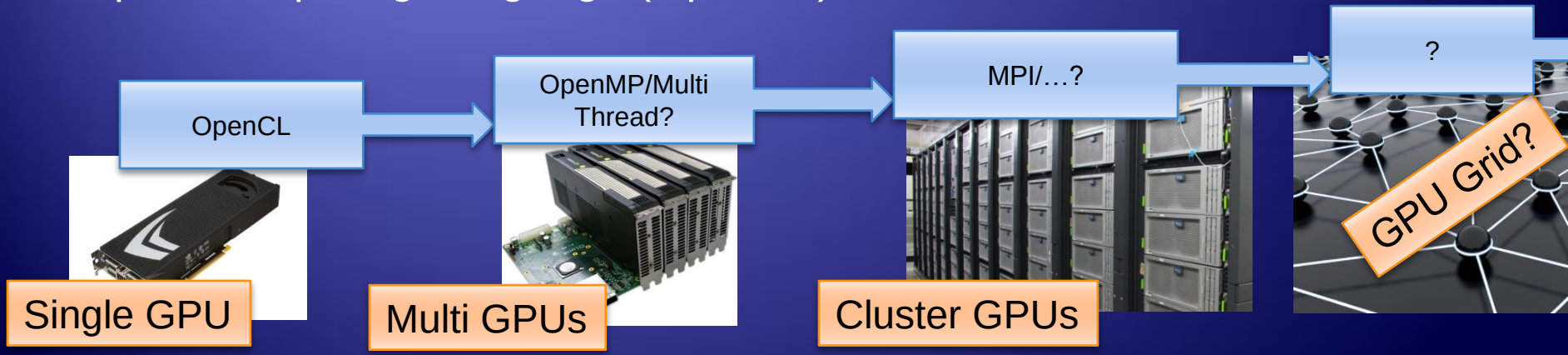




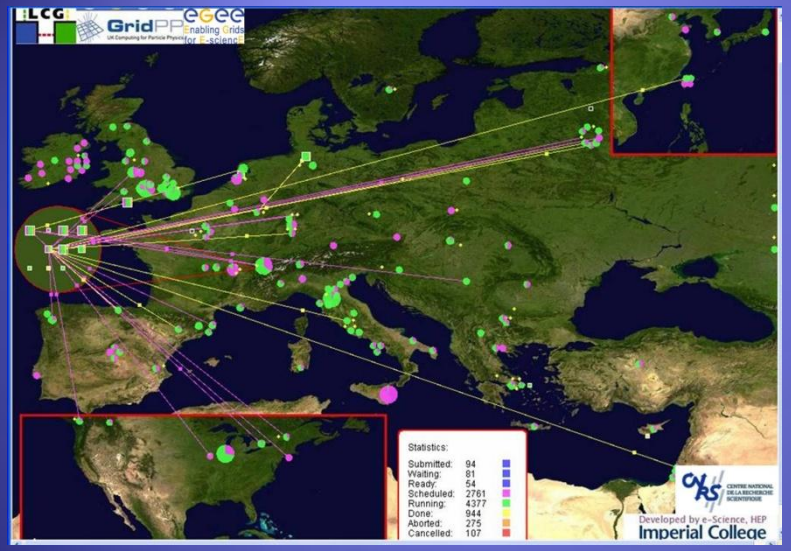


GPU computing and programming paradigms

- The architectural differences between GPU and CPU are evident, in particular the way how the relations between cores, memory, shared memory and IO subsystem are organized
- Moreover different chip producers implement different solutions with different characteristics and instructions sets
- Recently, several important efforts have been done by Apple, Intel, NVIDIA , AMD-Ati, Sony, … in the direction of programming standardization for parallel architecture: The Khronos Group, and the Open Computing Language (OpenCL) definition.



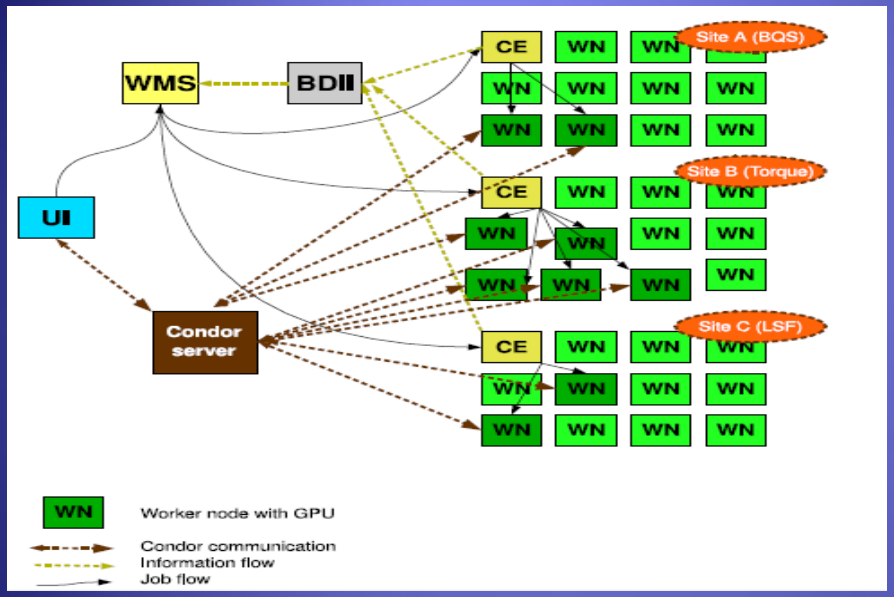
Grid in Europe
[Debreczeni Gergely work]



- Future of distributed computing ?
- LCG – EGEE I – EGEE II - EGI
- Enabling and interfacing, shared, heterogeneous resources, such as GPU
- Currently ~100k CPU, 8K users, 157 research group, ~100 PB storage
- GW community doesn't make use of it as it could

- Virgo – ET transition is not like LEP – LHC. Size of data will not multiply.
- As a consequence shipping data away will not be a problem !!!!
- Distributed computing will be very useful

Prototype Grid submission for heterogeneous hardware [Debreczeni Gergely work]



- Pilot jobs for low latency and user transparent analysis
- Metacluster over Grid clusters
- Sites with various properties connected together (GPU, Storage, etc....)
- New resources became available
- Virgo is supported by ~ 30 site – several thousand of new CPU !!!!!

- Grid will evolve quickly
- Instead of specific solutions, the technology should be considered for ET !!!
- It will became a standard infrastructure like GEANT is now ....

For more details ref. to http://www.grid.kfki.hu/afs/gdebrecz/web/amaldi08.pdf

CUDA programming model
Cuda Kernels and threads

- Parallel portions of an application are executed on the device as kernels
  - One kernel is executed at a time
  - Many threads execute each kernel

- Differences between CUDA and CPU threads
  - CUDA threads are extremely lightweight
    - Very little creation overhead
    - Instant switching

Some Definitions:

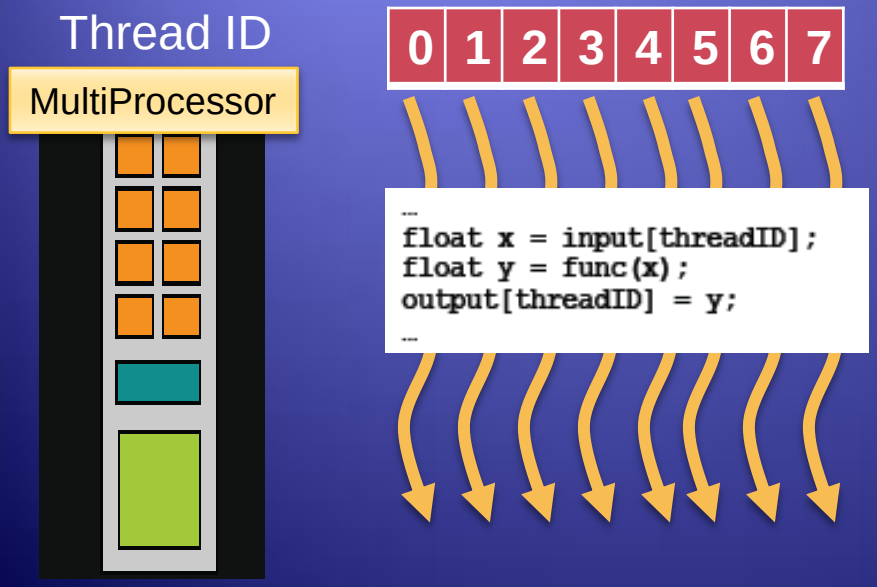
Device = GPU

Host = CPU

Kernel = function that run on the device

CUDA programming model Array of Parallel Thread

- A CUDA kernel is executed by an array of threads
  - All threads run the same code (kernel)
  - Each thread has an ID that used to compute memory addresses and make control decisions
- SIMT (Single Instruction Multiple Thread)



- Thread blocks are executed on Multiprocessors
- Thread blocks do not migrate
- Several concurrent thread blocks can reside on one multiprocessor

GPU computing application

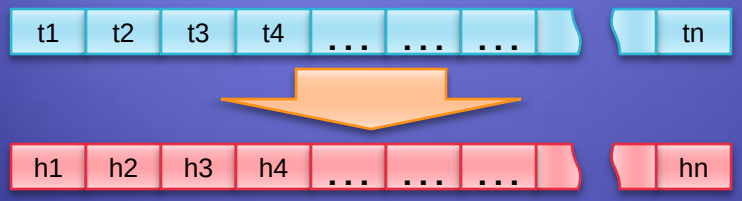
- GPUs address well problems that can be expressed as data-parallel computations:
  - When the same program is executed on many data elements in parallel
  - With high arithmetic intensity.
- Data-parallel processing maps data elements to parallel processing threads.
- Many applications that process large data sets can use a data-parallel programming model to speed up the computations.

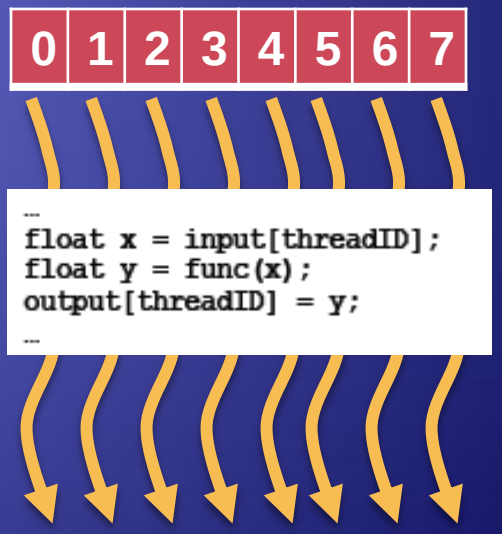
cuInspiral: GPU high arithmetic intensity CB library

- cuInspiral is under develop in INFN Perugia, using CUDA framework
- cuInspiral is the first library implementing a high arithmetic intensity strategy on GPU for coalescing binaries DA.
- All computation is made completely inside GPU space, using particular algorithmic solution e.g. reduction schema, permitting to gain performance from GPU shared memory.
- The actual prototype release provide at user level a set of functionalities, such as:
  - Taylor PN2 generator
  - Normalization
  - Matched filtering
  - Maximum identification
  - Other complex vector operations

culnspiral: Templates generation on GPU

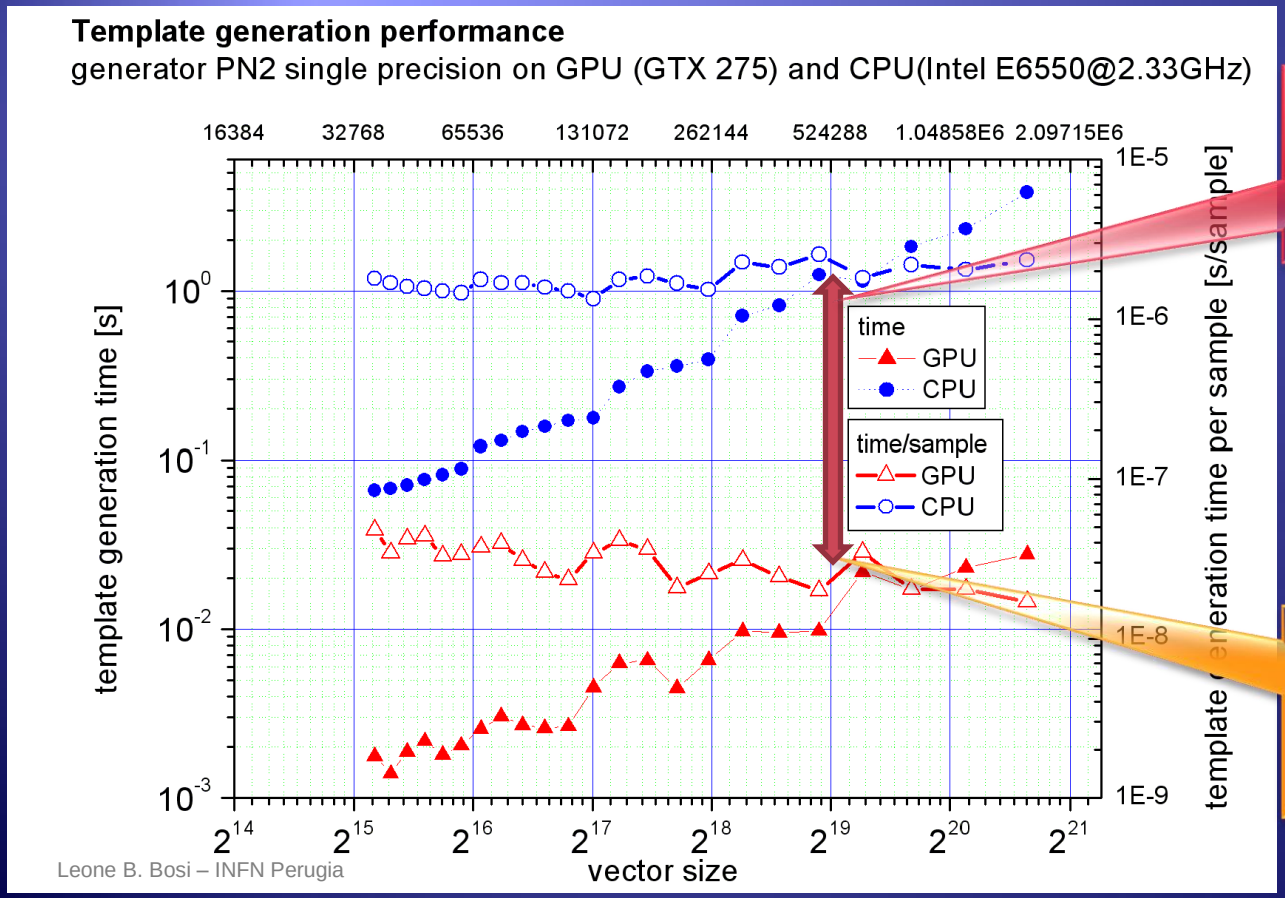
- Template generation is a typical computing problem that maps very well the GPU architectural model.





- Each GPU thread computes a time sample

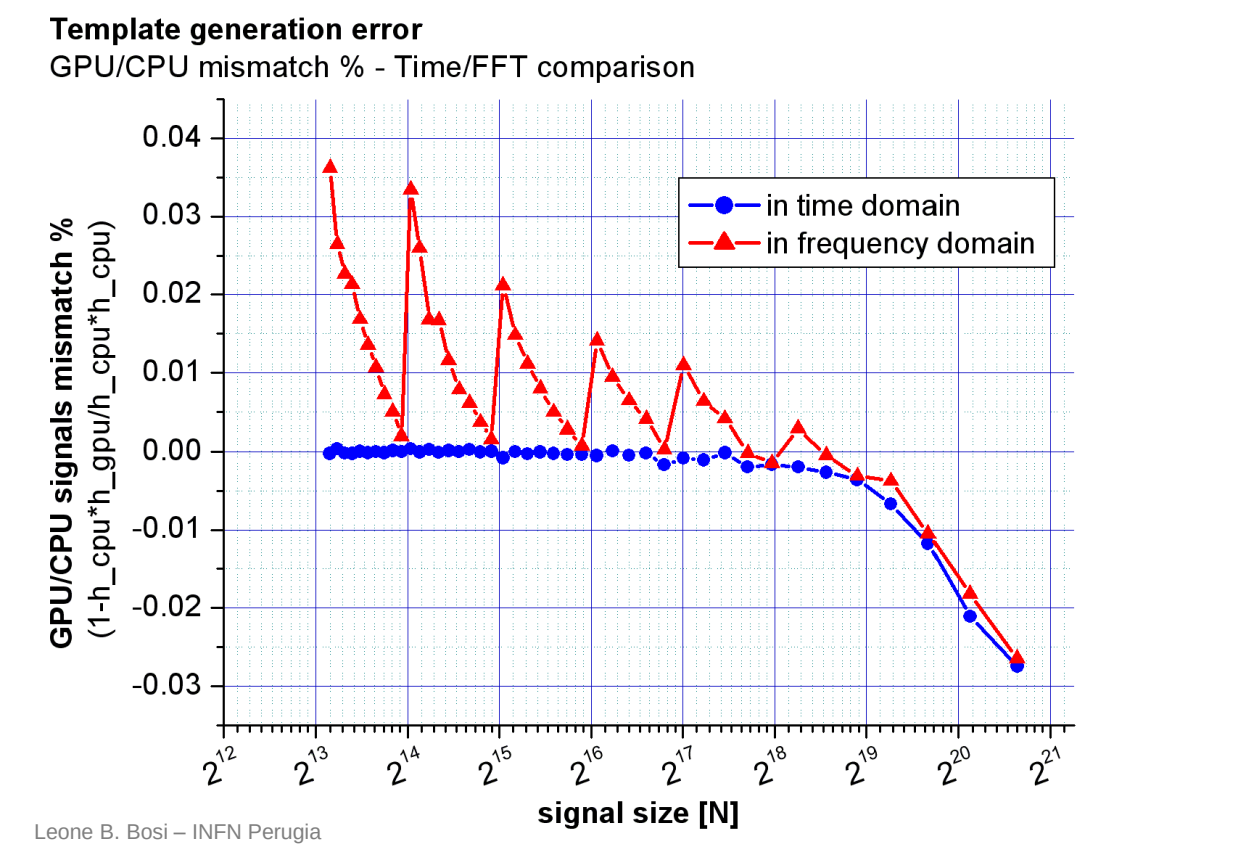
culnspiral:
PN2 template generation
[performance]



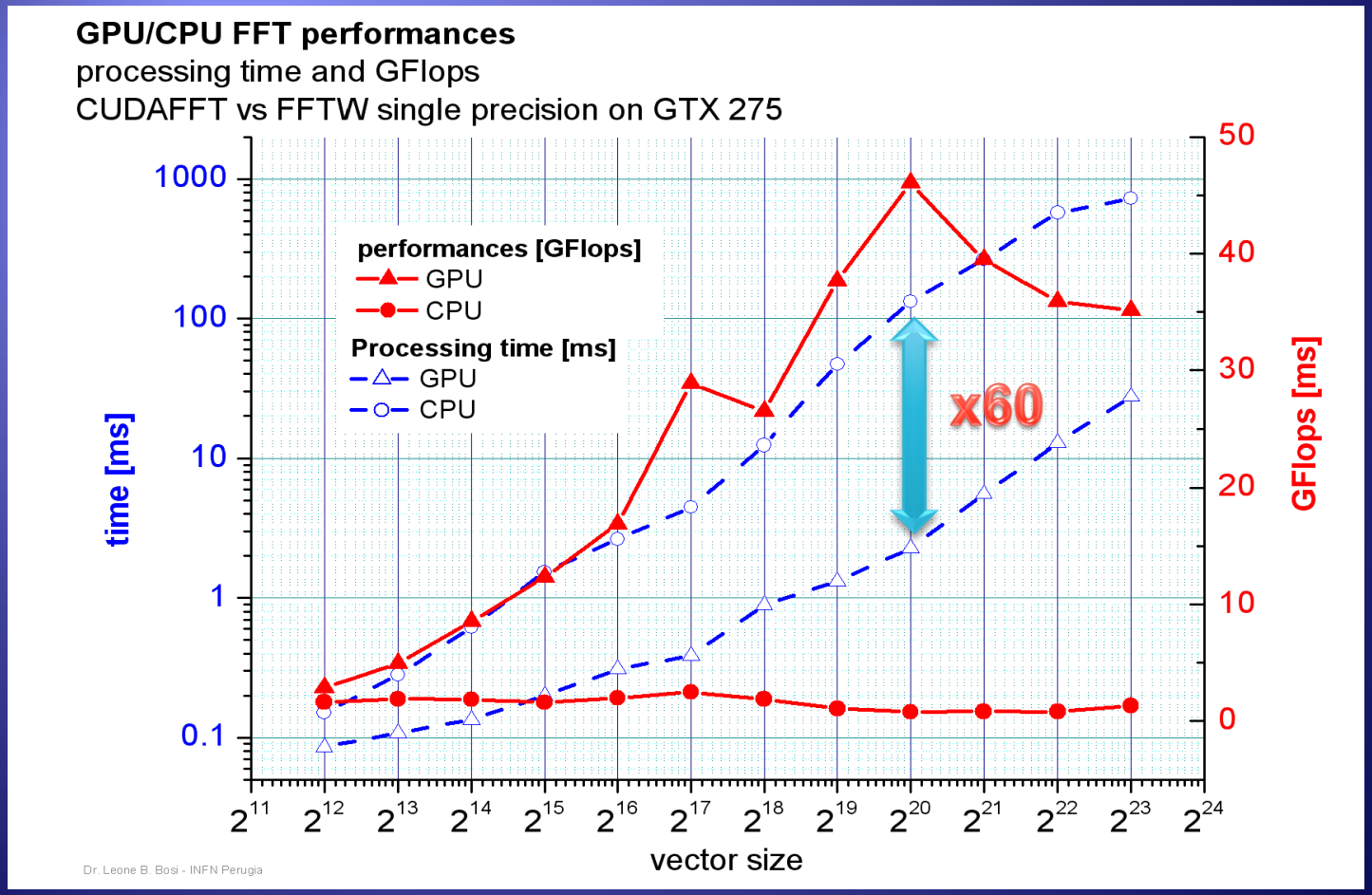
gain = X100

2-3 10^{-8} s/sample

culnspiral:
PN2 template generation
[Accuracy]



CUDAFFT vs FFTW: proc time | GFlops (single precision)



News about: CudaFFT

Actual GPU architecture can provide better performances with a more optimized code.

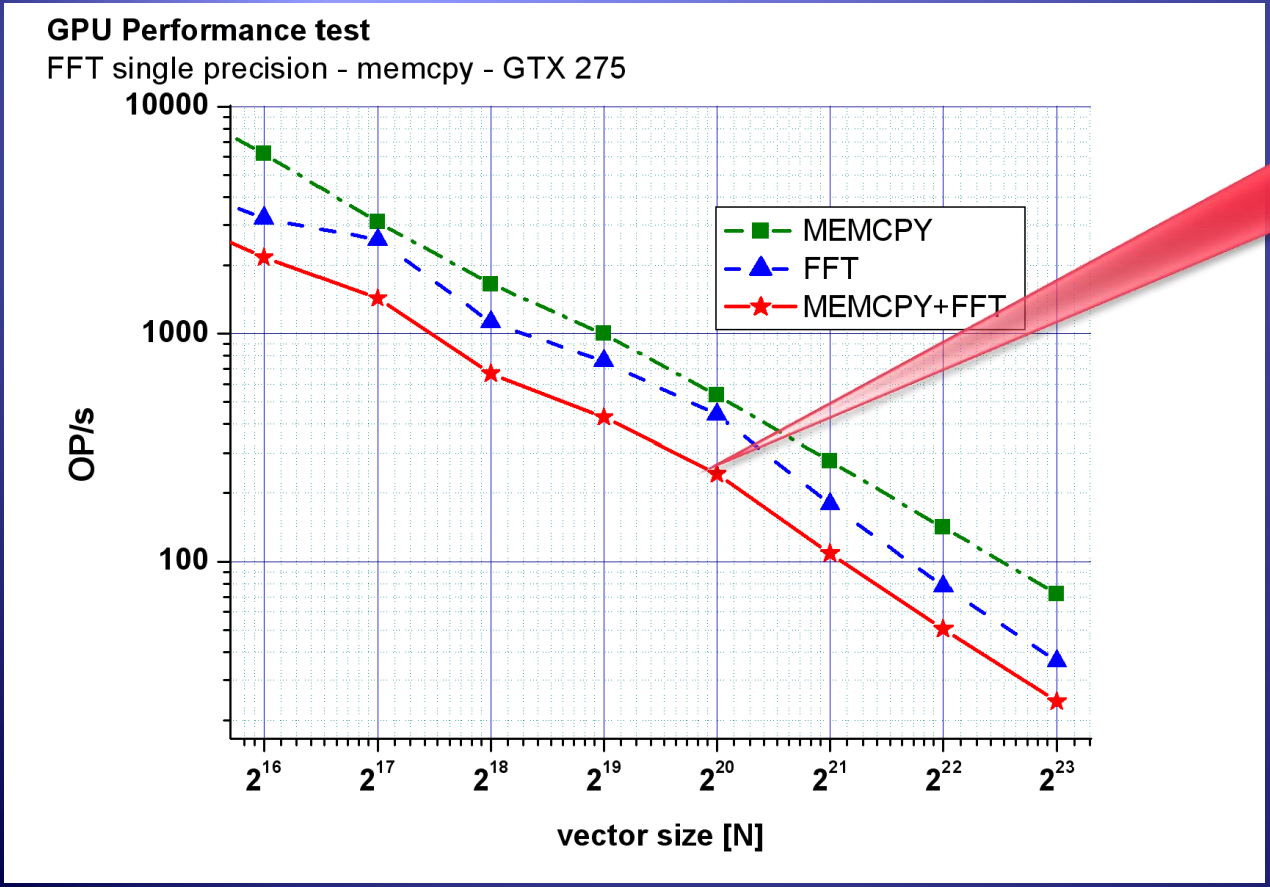
→The actual CuFFT library doesn't perform a "real" real-to-complex algorithm but executes a masquerade complex-to-complex transformation

→The CuFFT code has big margin of optimization. Some independents coding reports a factor **X3-X4** ,e.g. the paper:

"High Performance Discrete Fourier Transforms on Graphics Processors" Naga K. Govindaraju, Brandon Lloyd, Yuri Dotsenko, Burton Smith, and John Manferdelli - Microsoft Corporation

→We could expect for the next year a CUFFT-FFTW performance gain of a factor **X180** on the next CUDA versions, due to algorithms optimizations.

Host←→Device Memory I/O overhead

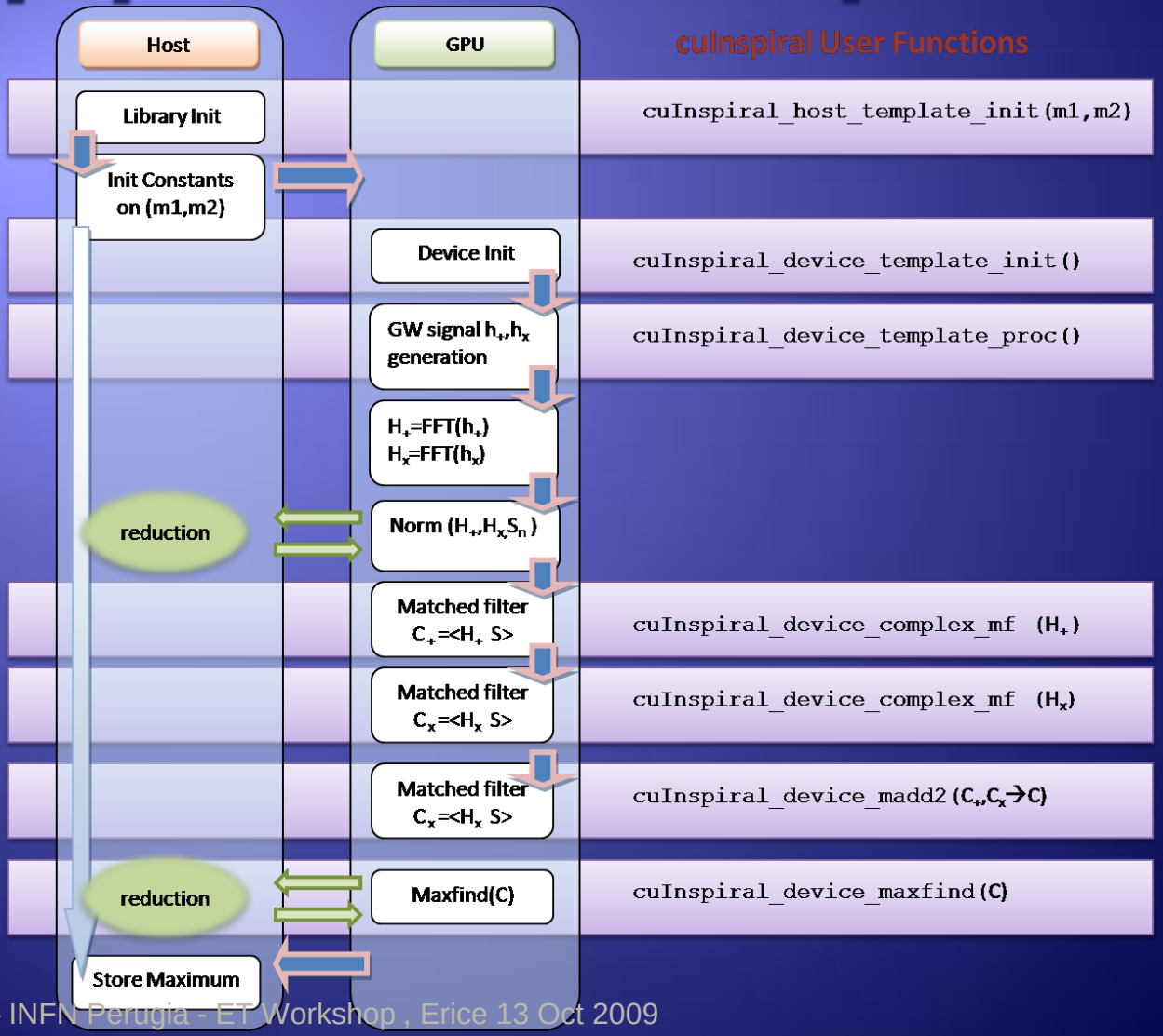


cuFFT wrapping approach looses performances due to memory IO operations

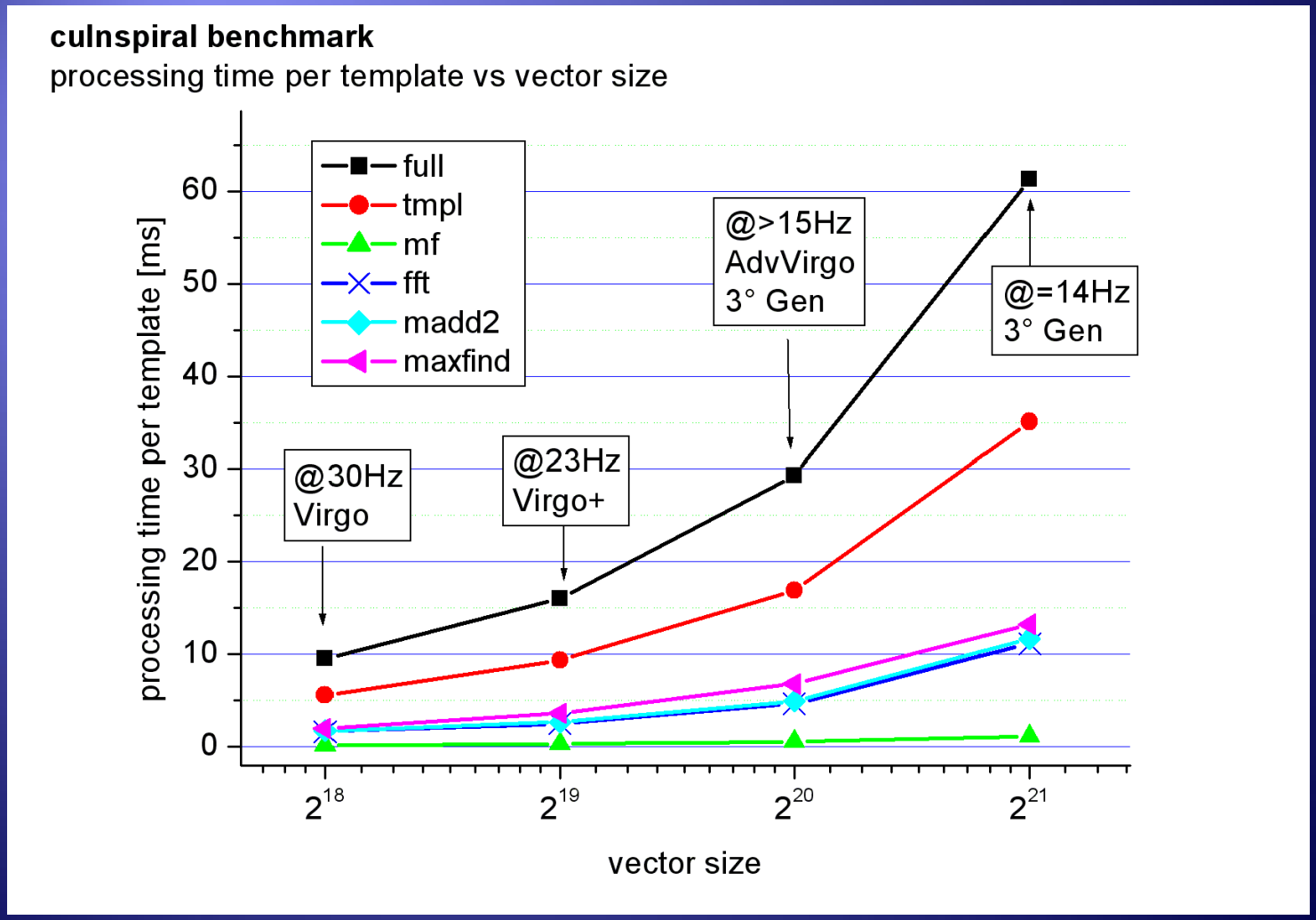
Host-Device memory IO kills performances

GPU express best performances in high arithmetic intensity conditions.

cuInspiral: CB pipeline description



culnspiral: pipeline performances (1):



cuInspiral: pipeline performance (2):

- If we consider analysis parameters of :
  - low.cutof.freq:24Hz,
  - vector length 2^20, fs=4kHz the
- The cuInspiral processing rate on GTX 275 is roughly of:

  30 TEMPLATES/SEC (lower limit)
- If we consider the online constrain processing with 4000 templates, we can estimate that analysis (detection only) can be performed with a couple of GPU 275 (=500 Euro)

Pipeline Gain (lower limit):
>x50 with GTX 275
→ expected with Fermi GPU: **x150**

Ref. to my Talk 2-Oct ET f2f meeting for more info.

Conclusions

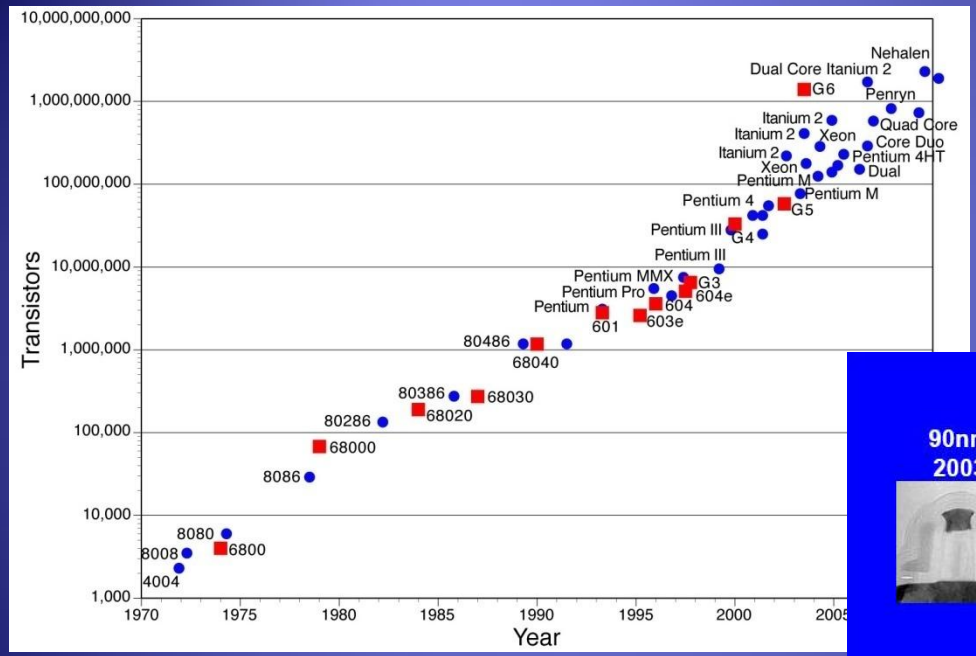
- **cuInspiral** is a prototype library developed in INFN Perugia to perform and evaluate high arithmetic intensity computation on GPU about CB data analysis.
- This work is in the perspective of the so called “manycore era”, important for ET purposes.
- With this version we have reported preliminary performances on a full CB detection pipeline on these new architecture (on nvidia GTX275). In particular:
  - Gain factor **X100** about templates generation respect to identical CPU implementation. (This factor is expected to be roughly constant also for higher PN approximation or others generators).
  - Gain factor **X60** about FFT, using cuFFT library, but in the close future, new cuFFT versions promise to have **X120-180** or more.
  - Number of **X30-35** templates processed per seconds with vector size= 2^20
- It has been reported how the cuInspiral approach permit to obtain very impressive gain from these manycore architectures.

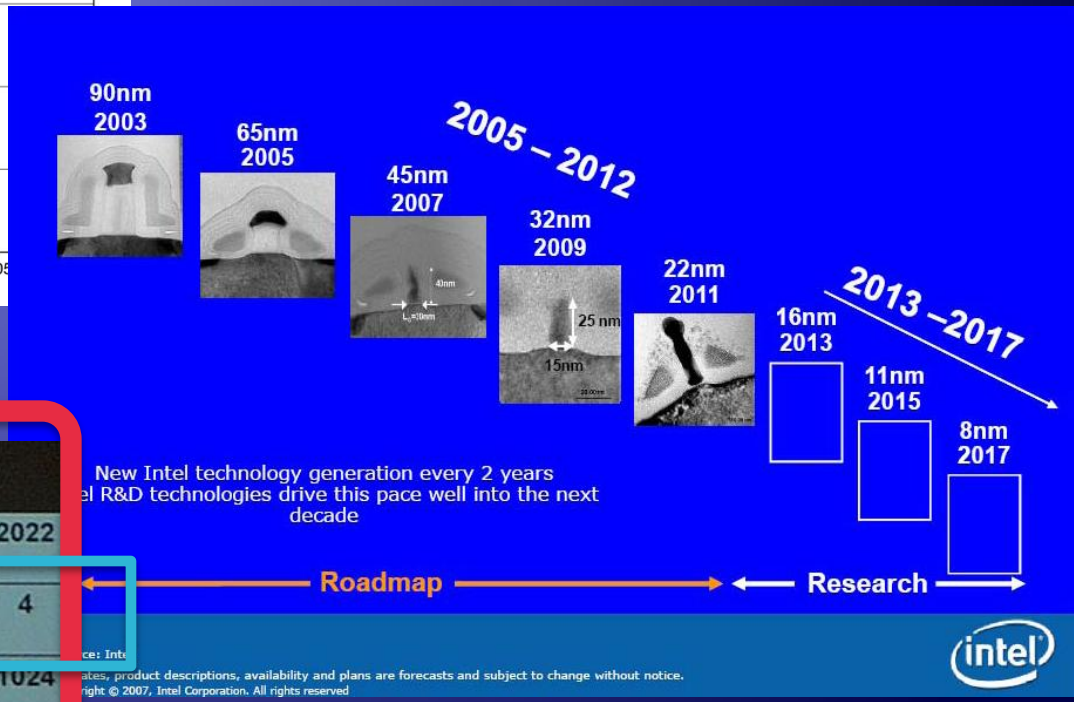
What next

- New GPU generations (e.g. Nvidia Fermi processor) have been or are going to be released in 2010 with a performance factor of x3-4 respect to the results reported on this talk.

- **cuInspiral** library has been merged with **MaCGO** (***Manycore Computing for future Gravitational Observatories***) experiment.
- MaCGO is an INFN Project, members of INFN, Chem. and IT Dep. and Dr. Leone B. Bosi is the National Rep.
- In the project roadmap of the next two years there are several important point:
  - Production of a first release of a numerical library for GW-DA on manycode architecture.
  - Implementation of dedicated algorithms for Multi-GPU topology.
  - Use of a more general programming language OpenCL, for such kind of computing architectures.

Extra 1





Technology Outlook

High Volume	2008	2010	2012	2014	2016	2018	2020	2022
Technology Node (nm)	45	32	22	16	11	8	6	4
Integration Capacity (BT)	8	16	32	64	128	256	512	1024

Intel ref.

Extra 2 forecast

- Given a template bank for ET, We can define to truncate critical long inspiral by chosing properly the low frequency cut-off (e.g.4Hz)
- With this choice we can reduce the max template length for this analysis step → 3600s
- **2000000** templates (length=2hours@4kHz), today using a cuInspiral GPU like library to process a single timeslice we require: 2000000 x 0.8 s=**18 days**!!
- If we renormalize respect the previous estimated gain factor (@2020 forecast) **3x100*0.4=120** we obtain

  →4hours computing time
- It seems plausible that by 2020 computing innovation we will be able to pursue ET requirements for this task.

Extra 3 forecast

We could try to make a projection of the available computing power by 2020, in the context of CB like algorithms, making some assumptions:

1. We can start considering that the actual firsts attempts of manycore architecture provides a factor **x150** in single precision respect CPU implementation.
2. We can consider a Moore's law factor at that time of **x100**
3. From the experiences coming from massive parallel architecture, usually performances are reduced significantly by communication overhead, thus we take **x0.4** (it could be even worse)
4. We obtain :
   - a gain of a factor **6000** respect the actual CPU implementation.
   - Equivalent to **5 TFLOPs or higher** on a single manycore processor by 2020.