# Hardware Accelerator for HPC in bioinformatics

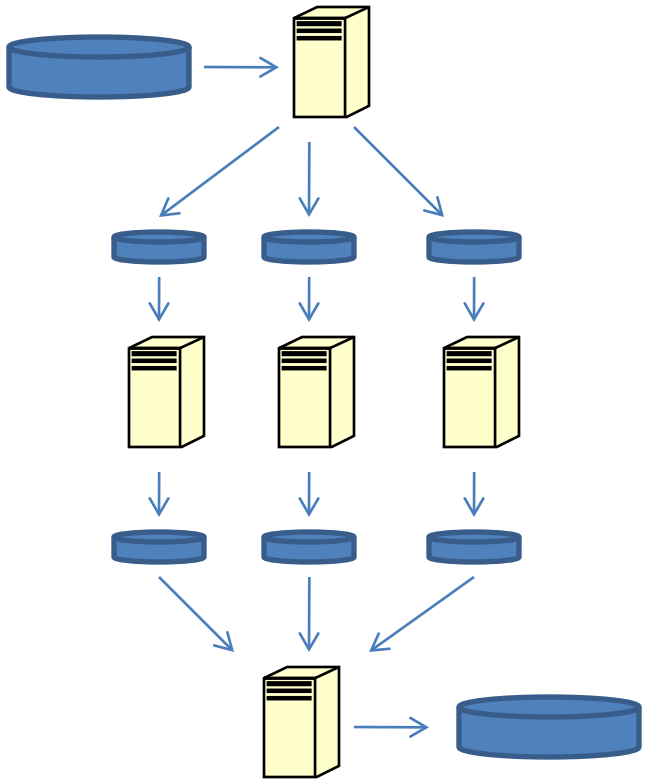
Dominique Lavenier

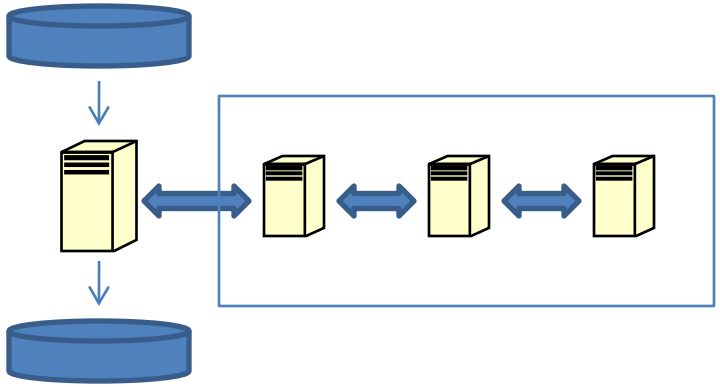
ENS Cachan Bretagne / IRISA

EPI INRIA Symbiose

# Parallelism

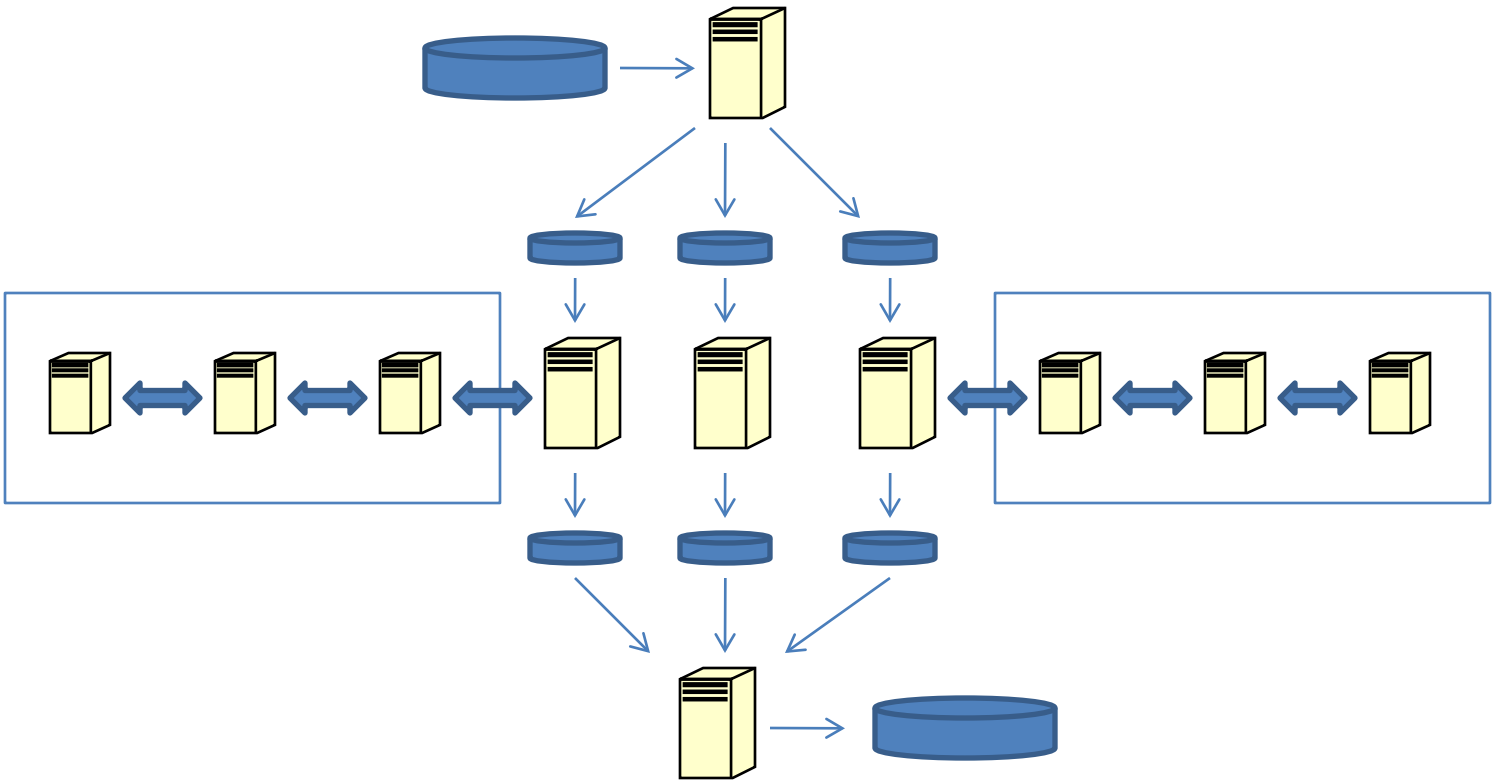
- Different ways to parallelize computation





# Parallelism

- The 2 approaches are not antagonist



# Why using accelerators ?

- Cheaper
- Saving electrical power
- Lower maintenance cost
- In situ computation

+


≈

# Why not using accelerators ?

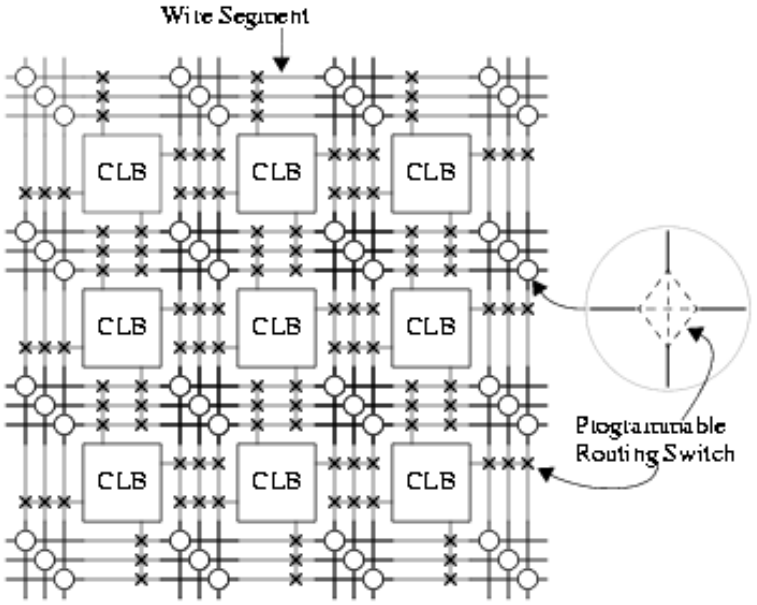
- Cannot do everything
  - Only a subset of algorithms can be (efficiently) parallelized on such platforms
- No standard
  - Hardware
  - Software
- No soft continuity across the continuous accelerator generations

# Two families of accelerators

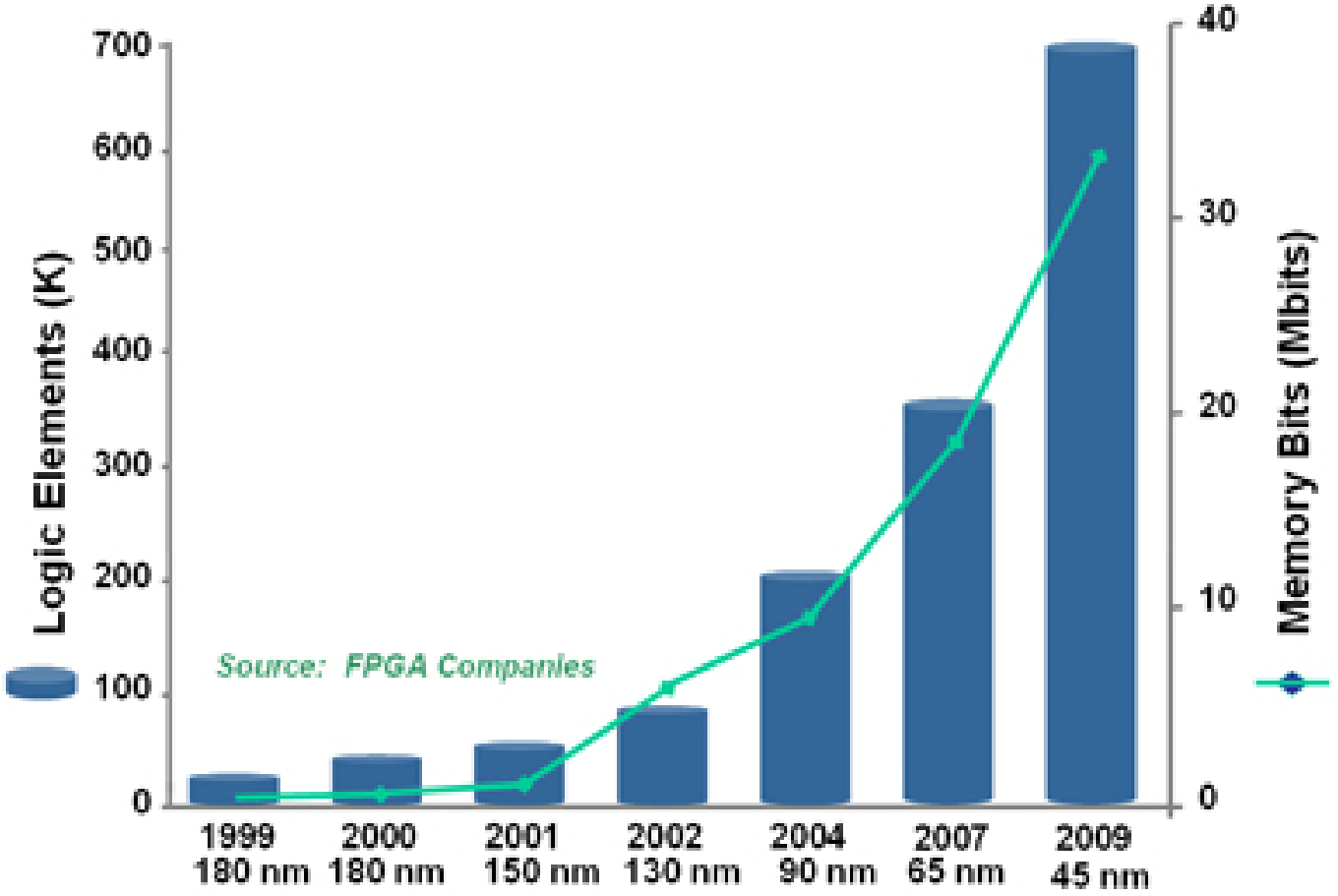
- FPGA
  - Reconfigurable computing
  - Bioinformatics : 15-20 years  (VLSI prototypes)
- GPGPU
  - General-Purpose computation on Graphics Processing Units
  - Bioinformatics : 3-4 years

# FPGA

- FPGA : Field Programmable Gate Array
- Reconfigurable interconnection network
- Matrix of reconfigurable computing elements
  - Logic gates
  - Memories
  - Arithmetic operators
- Frequency : 250 MHz
- Xilinx, Altera
- Reconfiguration time < 1 sec



# FPGA: density evolution

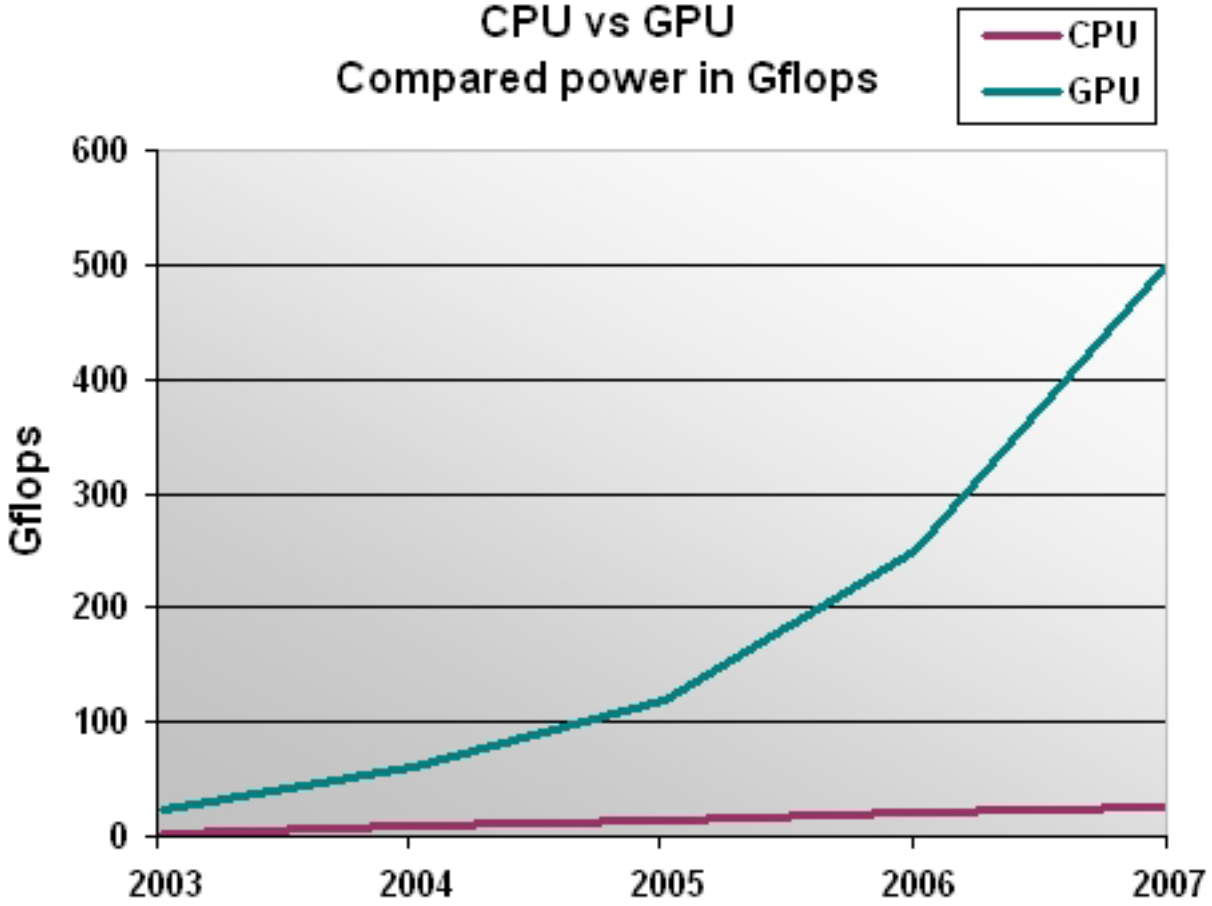


# FPGA: Capacity

*(example : Xilinx XC6VLX760)*

- 3000 × INTEL 4004,  X 5000 faster
  - 1st microprocessor, 2300 Tr, 60K inst/sec.
- 100 × microBLAZE
  - 32-bit microprocessor – 250 MHz
- 1000 dedicated processors for protein comparison
  - 8-bit processor, 250 MHz, 4 operations/cycle

# GPGPU


CPU vs GPU
Compared power in Gflops
CPU
GPU
Gflops
600
500
400
300
200
100
0
2003
2004
2005
2006
2007

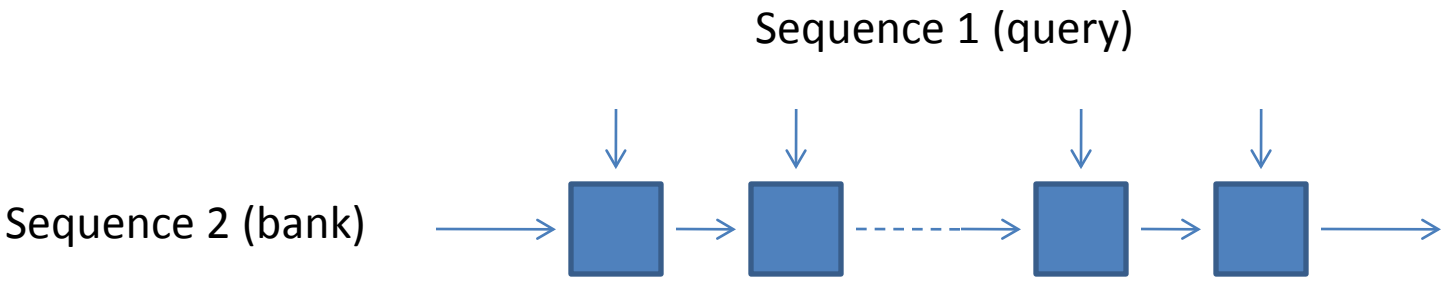
# NVIDIA GeForce GTX 295

- 480 processing units
- 2 chips
  - 1.4 billion transistors
  - 1.24 GHz
- 1788 GFlops
  - MADD+MUL, single precision
- Memory: 1.8 GB GDDR3
- CUDA programming language
- 400 Euros
  - 25 cents / GFlops

# Some algorithms

- Sequence comparison
  - Dynamic programming (S&W)
  - BLAST
  - HMMER
- RNA folding
- Docking

# Sequence Comparison
## - Dynamic Programming & FPGA -




- Optimal parallelism
  - N processors → speedup = N
- Well suited for FPGA
  - custom data path (8 / 12-bits)
  - simple integer arithmetic
  - very regular computation (simple control)

# Sequence Comparison
## - Dynamic Programming & FPGA-

- Commercial products

  - SeqCruncher, Active Motif
    - Speedup : 10-30 versus SSEARCH-SSE2

  - Cube, CLCbio
    - Speed : SW as fast as BLAST

# Sequence Comparison
## - Dynamic Programming & GPU -

- Two Parallelization schemes
  - Data parallelism
    - Sequences are sorted by length
    - One thread assigns to one pair-wise comparison
  - Systolic-like parallelism
    - Many threads contribute to a single pair-wise comparison

# Sequence Comparison
## - Dynamic Programming & GPU -

- Best implementations
  - University of Padova, Italy
    - SWCUDA : speedup = 2-3 versus SSEARCH-SSE2
    - 2 x NVIDIA GTX 8800
    - *BMC Bioinformatics, 2008, 9:510*
  - Nanyang Technological University, Singapore
    - CUDASW++ : speedup = 5-8 versus SSEARCH-SSE2
    - NVIDIA GTX 295
    - *BMC Research Notes* 2009, 2:73

# Sequence Comparison BLAST

- PLAST *(Parallel Local Alignment Search Tool)*
  - EPI Symbiose, INRIA, Rennes
  - BLAST algorithm revisited
  - Identical heuristics (seeds)
  - Bank to bank comparison
  - Combine different levels of parallelism
    - Multithreading (multi- many-cores)
    - Vectorization (SIMD instructions : SSE)
    - FPGA accelerators, GPU boards
  - Speedup vs multithreaded BLAST
    - SSE : x3 - x5
    - GPU : x5 - x10 (2 x NVIDIA Tesla boards)
    - FPGA : x10 – x30 (SGI ALTIX 350 + RASC-100 – 2 x virtex 4)

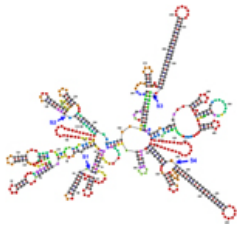
# BLAST - FPGA

- Tera-BLAST, Active Motif
  - SeqCruncher
  - Example : 4289 proteins X 192 bactera genomes (775 Mbp)
    - Tera T-BLASTN → 5760 sec.
    - TPLASTN-SSE 8-CPU cores → ~ 1000 sec.

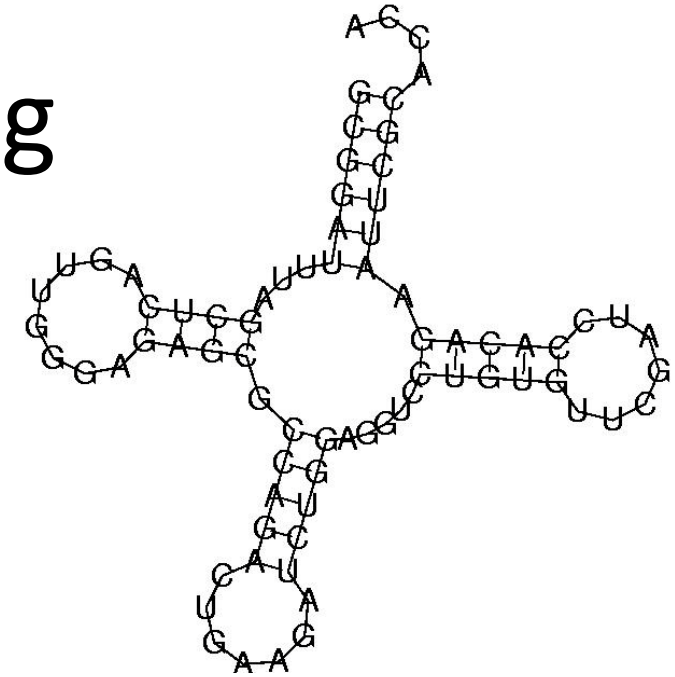
- Blast-n, Mitrionics, SGI
  - SGI RASC-100
  - Example : 3354 probes (25 pb) X 4 Gbp
    - Mitrion/RASC Blastn → 570 sec.
    - GASSST (1 CPU) → ~ 130 sec.

# HMMER

- Function
  - Profile hidden Markov models to do sensitive database searching using statistical descriptions of a sequence family's consensus
- GPU-HMMER
  - Open source software (www.mpihmmer.org)
  - Speedup : x 20 – 40  (Tesla C1060)
- FPGA : SeqCruncher
  - Active Motif's TimeLogic division
  - Speedup : x 60 compared to a 8-CPU core server

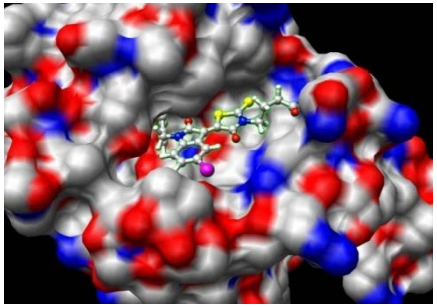


# RNA folding

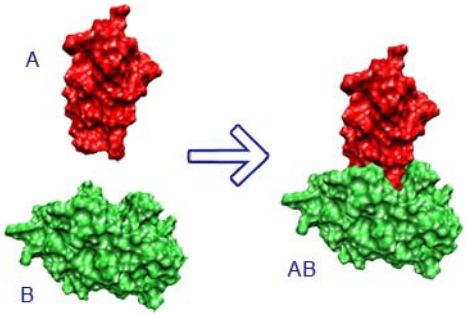


- UNAfold package
  - software package for nucleic acid folding and hybridization prediction
  - Time-consuming algorithm
    - Dynamic Programming
    - Complexity : $O(n^3)$
  - Parallelization on GPU
    - PhD these : Guillaume Rizk
    - Speedup :
      - X 20 (NVIDIA GTX 280)
      - X 40 (2 x NVIDIA GTX 280)
- Applications
  - miRNA prediction
  - statistical evaluation of secondary RNA structures
    - 1 RNA sequence (10kbp) : 2 weeks → 8 hours

Symbiose INRIA



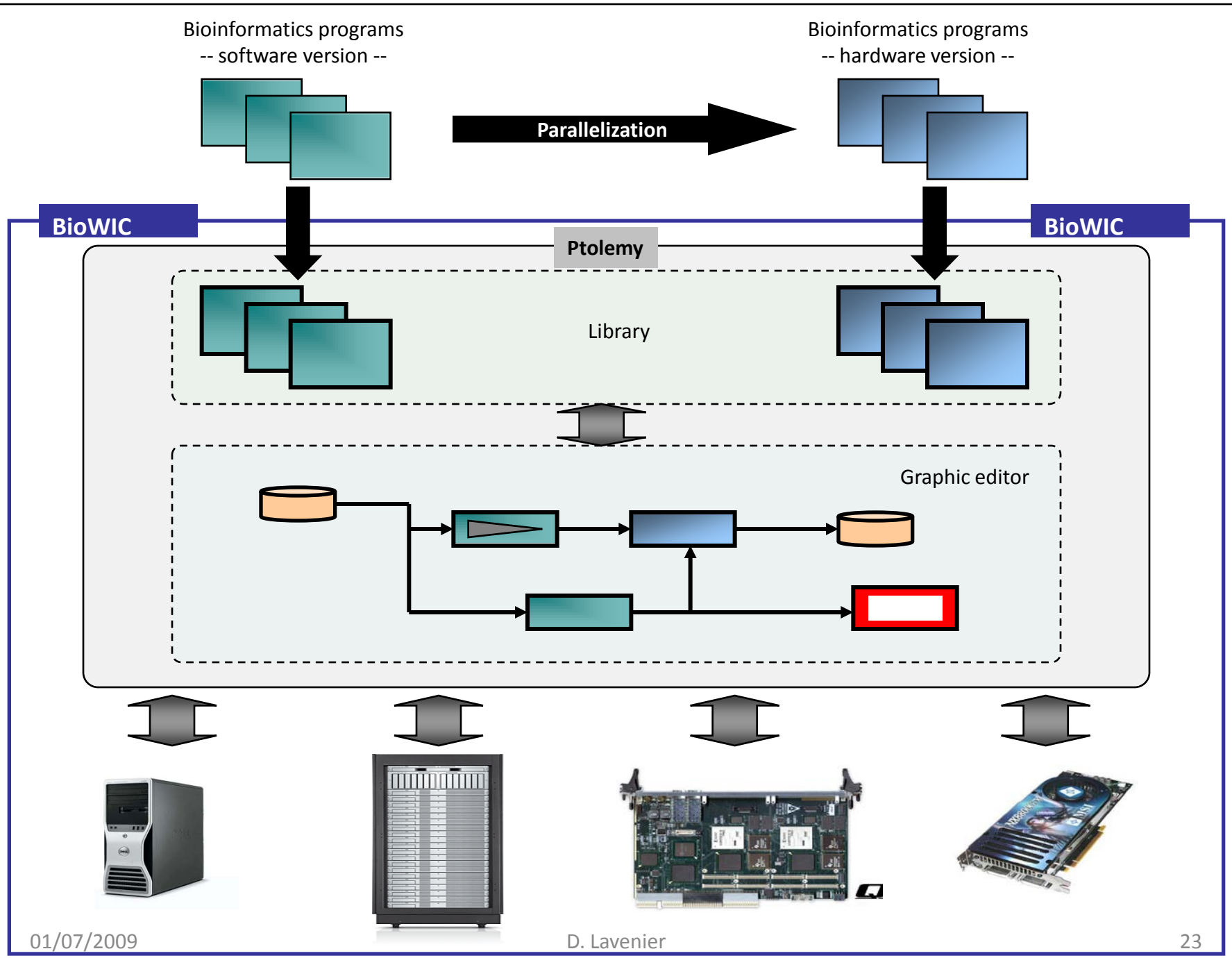
# Docking
# - GPU -




- PIPPER
  - Boston University, CAAD lab.
  - Speedup = 17 (Tesla C1060)
- FT-Dock
  - EPI INRIA Symbiose
  - Speedup = 7 (Tesla C870)
- AutoDock
  - Silicon Informatics
  - Speedup = 8 to 12 (Tesla server)

# BioWIC Project

- Bioinformatics Workflow for Intensive Computations
  - Fast design of pipelines
    - Workflow adapted to bioinformatics environments
  - Fast execution of pipelines
    - Target platforms :
      - FPGA
      - GPU
      - Clusters
- ANR project : jan. 2009 – dec. 2011


Bioinformatics programs
-- software version --
Parallelization
Bioinformatics programs
-- hardware version --
BioWIC
BioWIC
Ptolemy
Library
Graphic editor

# Conclusion

- End of the steadily increasing of the microprocessor clock frequency
- More genomic data → More computational power → More parallelism
- All level of parallelism must be taken into consideration
  - SIMD → SSE instructions
  - Multithreading
    → multi- many-cores
  - Hardware accelerators
    → FPGA / GPU
  - Clusters
  - Grids