OBELICS Task 3.4 – LAPP contribution

Thomas Vuillaume, on behalf of LAPP

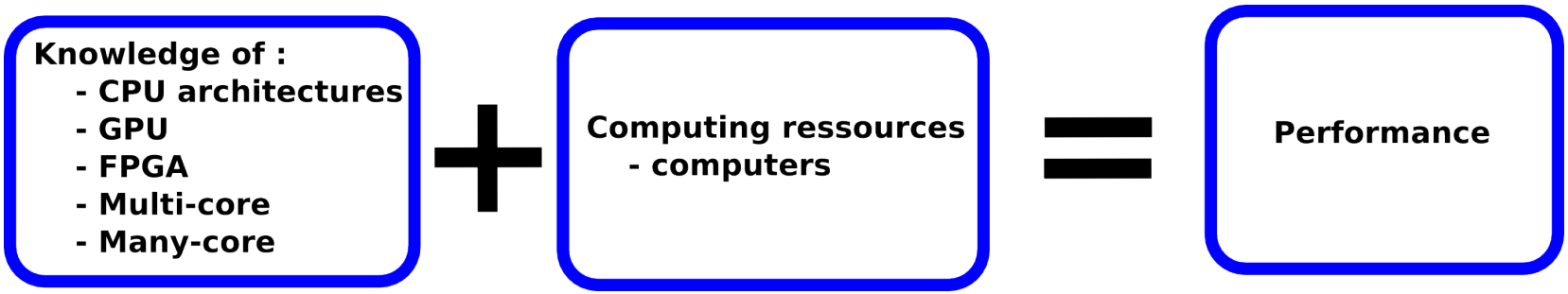
{Pierre Aubert, Thomas Vuillaume, Jean Jacquemier, Gilles Maurin, Armand Fiasson, Giovanni Lamanna}

Outline

What is High Performance Computing

Aim

Use the computer (CPU, GPU, FGPA, multi-core, many-core, ...) as efficent as possible



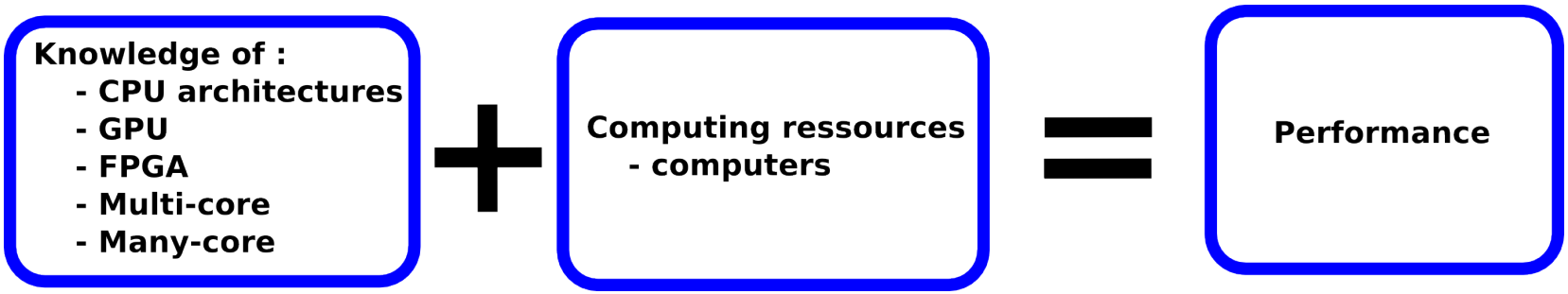
Do we need specific computers ?

NO

What is High Performance Computing

Aim

Use the computer (CPU, GPU, FGPA, multi-core, many-core, ...) as efficent as possible



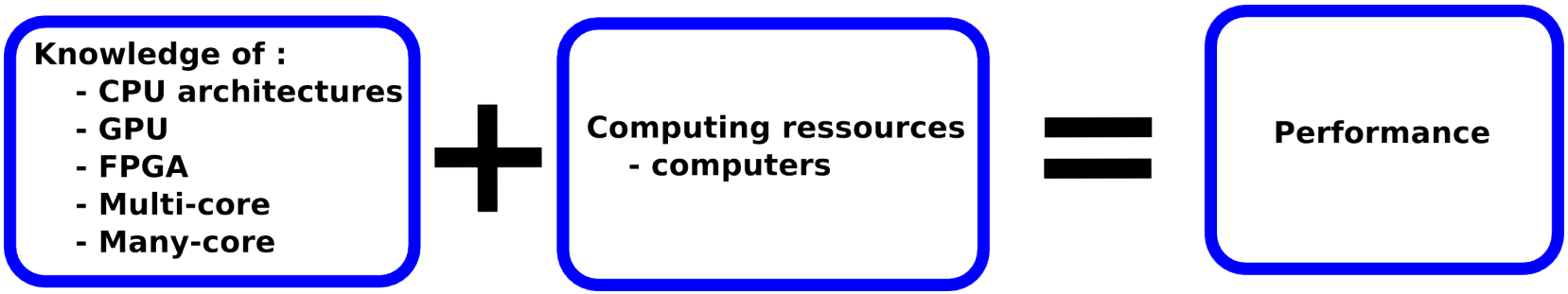
Do we need specific computers ?

NO

What is High Performance Computing

Aim

Use the computer (CPU, GPU, FGPA, multi-core, many-core, ...) as efficent as possible



Do we need specific computers ?

NO

HPC analysis

HPC CTA analysis

- Provide fast analysis steps
  - Vectorization
- Provide a data compression for the ADC signal
- Use a generated data format to make tests (Protobuf like)

Vectorization ?

CPU Recent Architectures

Architecture	Instruction Set	CPU	Nb float Computed at the same time
SSE4	2006	2007	4
AVX	2008	2011	8
AVX 512	2013	2016	16

Easy adaptation for coming architectures

Reduction optimization

Reduction Sum of all pixels photoelectron signal per camera

	Speed (cy/el)	Speed up
Classical	2.69842	1
Vectorized (GCC, SSE4)	0.702845	3.8

Reduction optimization

Reduction Sum of all pixels photoelectron signal per camera

	Speed (cy/el)	Speed up
Classical	2.69842	1
Vectorized (GCC, SSE4)	0.702845	3.8
Intrinsics Vectorized SSE4	0.226675	11.9
Intrinsics Vectorized AVX	0.11379	23.7

Barycenter optimisation

An element

x : position, y : position, a : amplitude

With SSE4 intrinsics

- 1 dimention (mean)
  - Speed 0.683944 cy/el
  - Speed 0.457595 cy/el twice interleaved
- 2 dimentions
  - Speed 0.910941 cy/el
- First and second momenta ($\bar{x}$, $\bar{y}$, $\overline{x^2}$, $\overline{y^2}$, $\overline{xy}$)
  - Speed 1.29088 cy/el

With AVX intrinsics

- 1 dimention (mean)
  - Speed 0.333342 cy/el
  - Speed 0.179133 cy/el twice interleaved
- 2 dimentions
  - Speed 0.343091 cy/el
- First and second momenta ($\bar{x}$, $\bar{y}$, $\overline{x^2}$, $\overline{y^2}$, $\overline{xy}$)
  - Speed 0.599513 cy/el

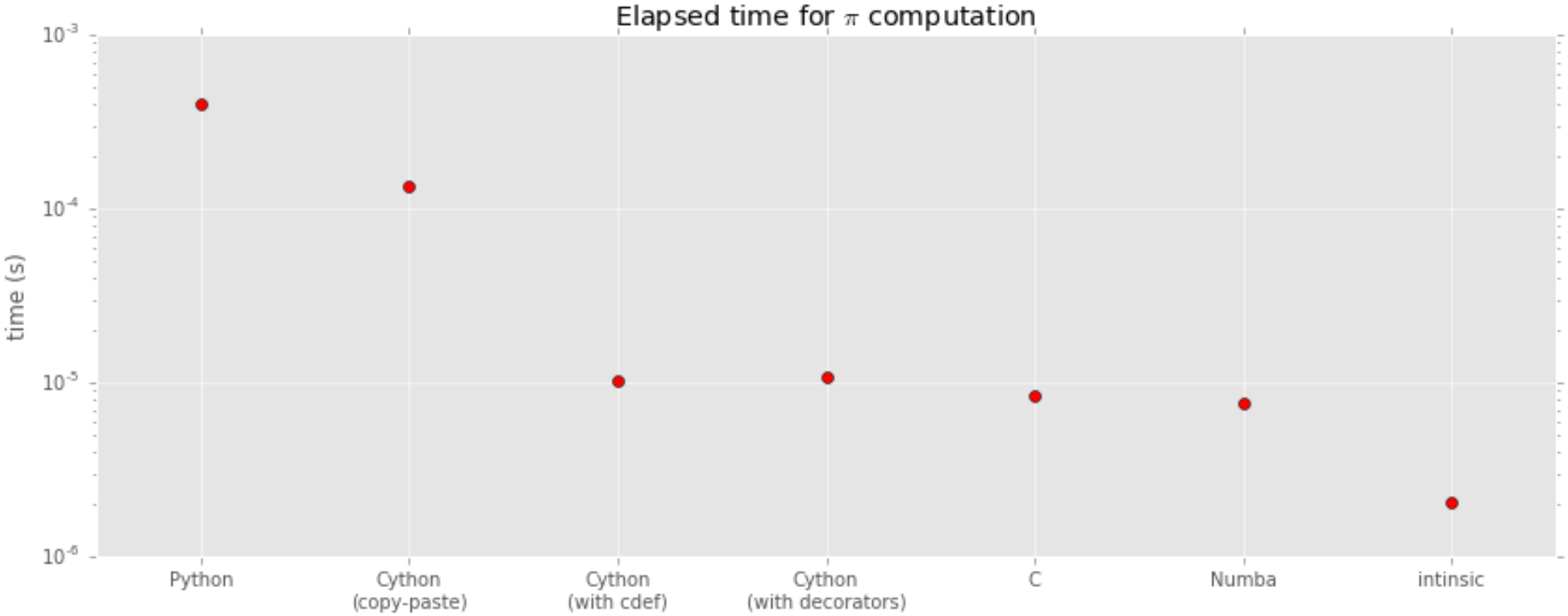
Application example : Hillas optimization

Performances on H.E.S.S. DATA

Processing time, observation with 4 telescopes (file 6 GB, 733547 events)

Mode	Hardware RAM Go	Total analysis time	Time per event
Sandard Analysis	5.7	8 min30 s	1.58 ms
Vectorized Analysis (SSE4, 4 float)	5.7	1 min10 s	0.2170 ms
Vectorized Analysis (AVX, 8 float)	16	7.6 s	0.0142 ms

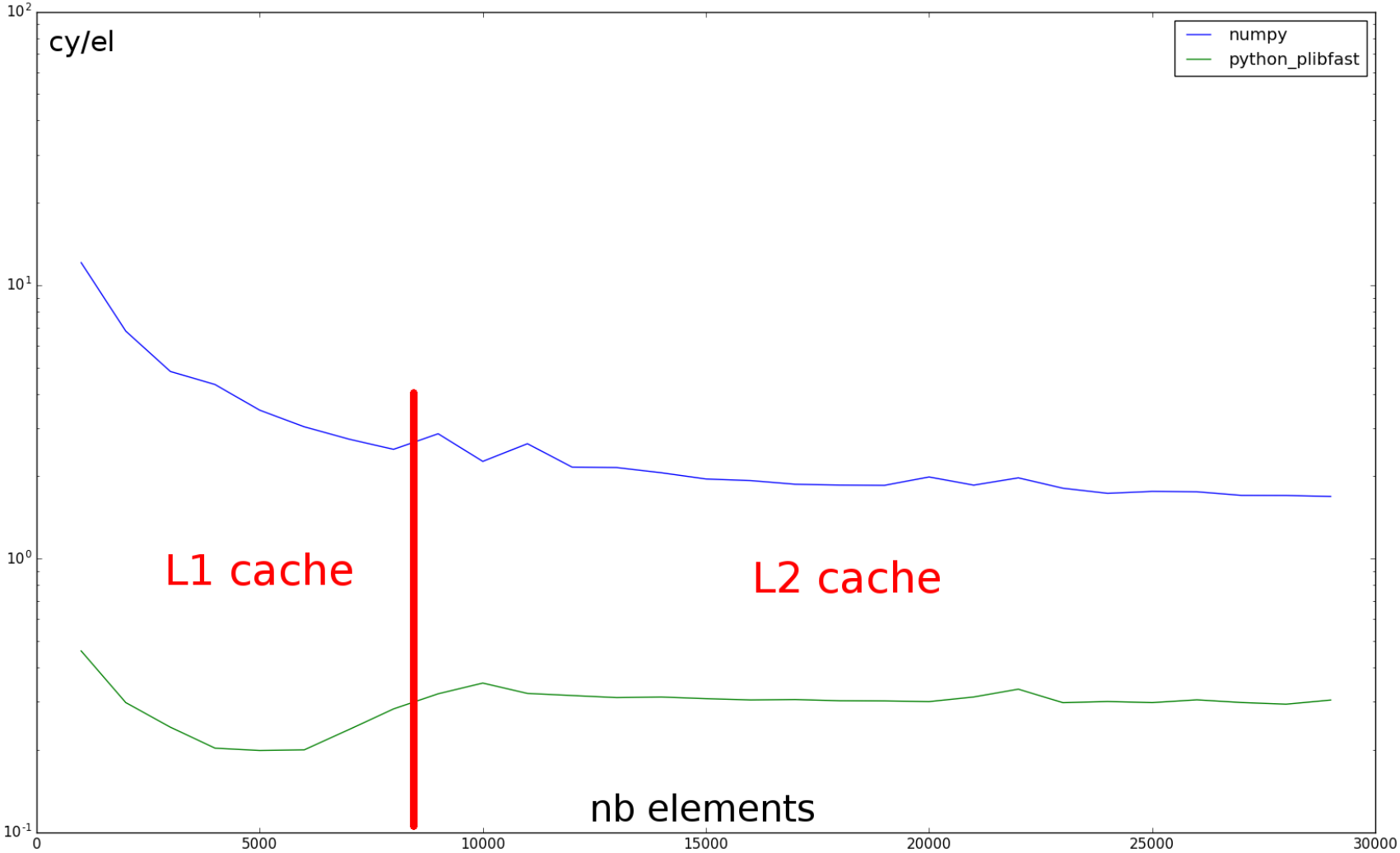
Use of our fast reduction method in Python



- Performance for intrinsic function : 2.9 cy/el
- 4 times faster than Numba
- Called by Python

$$\int \frac{4}{1+x^2}\mathrm{d}x$$

Computation time comparison : numpy VS our library for reduction



Conclusion

Fast calculation library

- Reduction
- Barycenter
- First and second momenta calculation

Python

- Fast python module with reduction 4 times faster than C and Numba

https://gitlab.in2p3.fr/CTA-LAPP/PLIBS_8

Backups

CPU Architecture

CPU Architecture

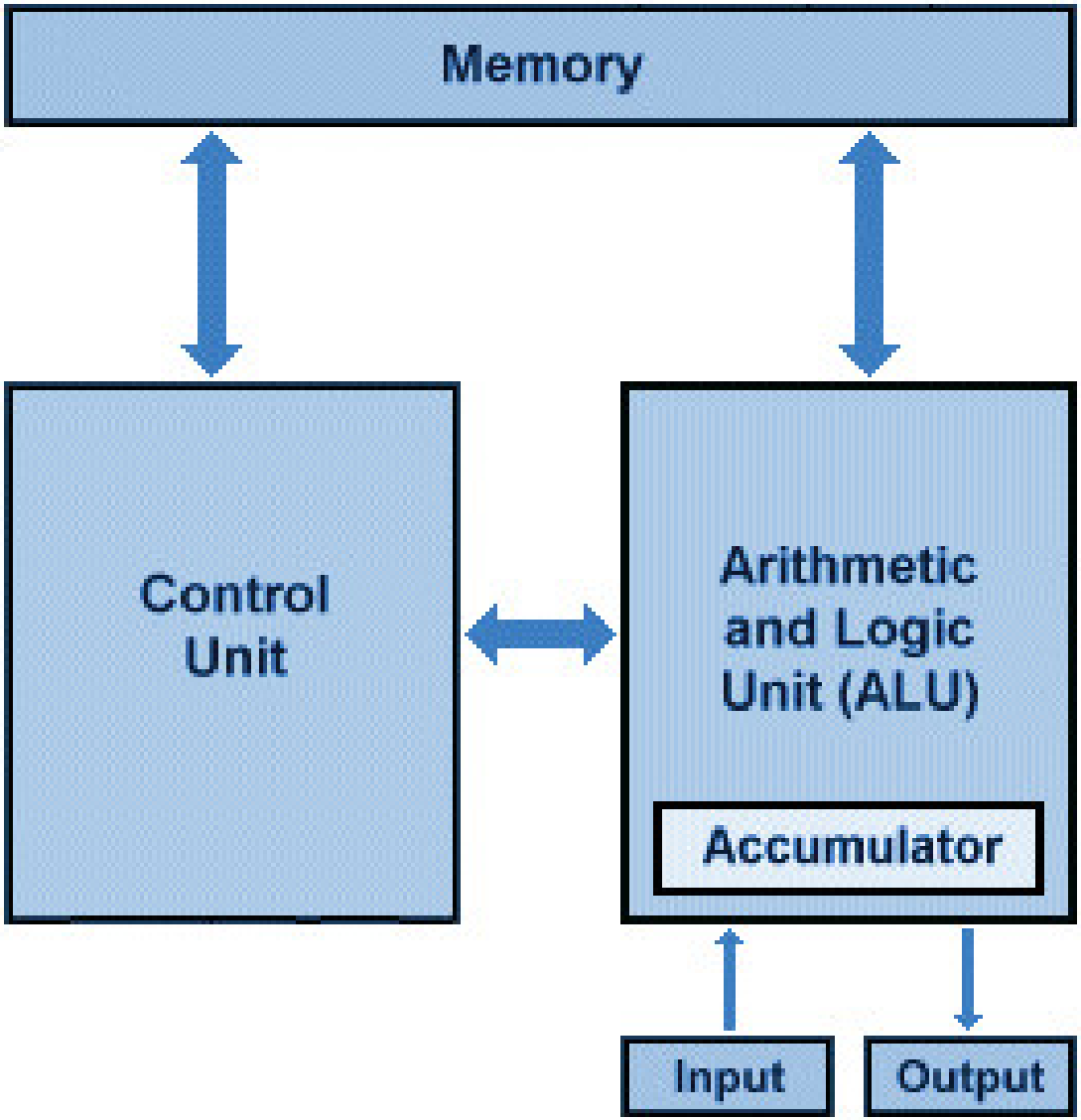
Von Neumann architecture 1945

Definition

Cycle : basis unit of time in a CPU

Time

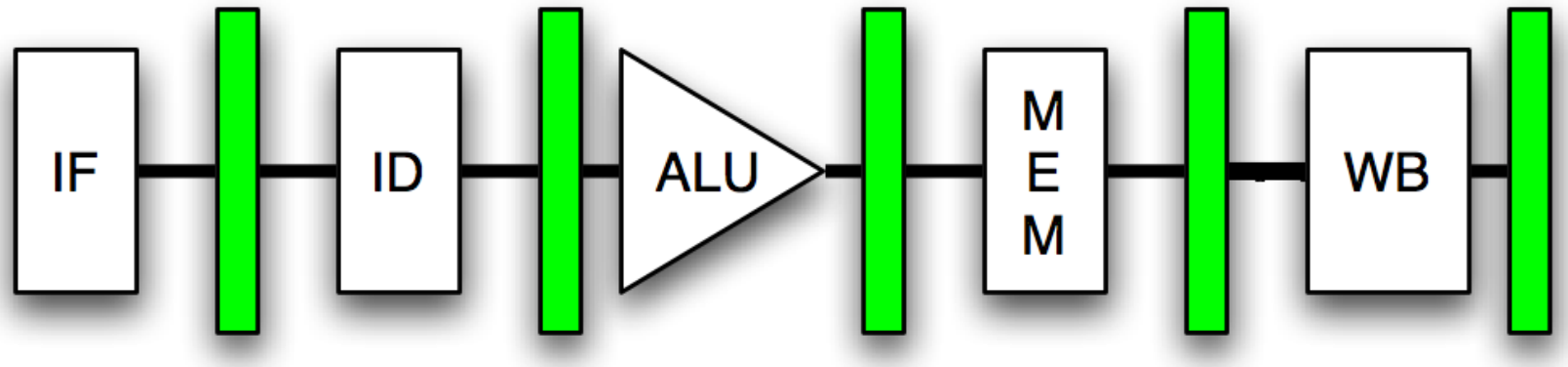
- 1 cycle per elementary operation (load, store, add, ...)
- 4 cycles per whole operation ($c = a + b$)



CPU Architecture

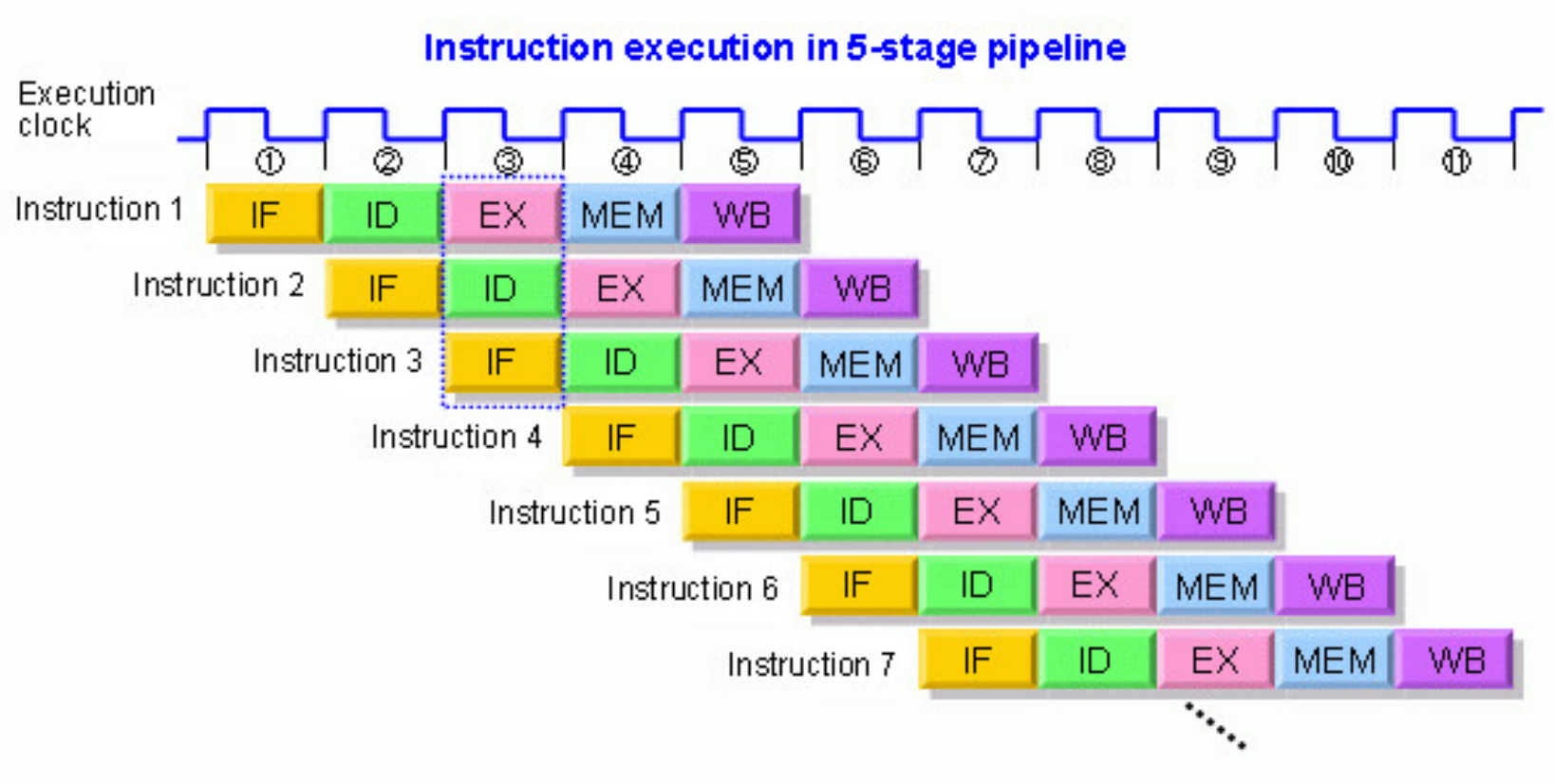
Pipeline approach

- IF : Instruction Fetch
- ID : Instruction Decode
- ALU : Execution
- MEM : Memory
- WB : Write Bytes



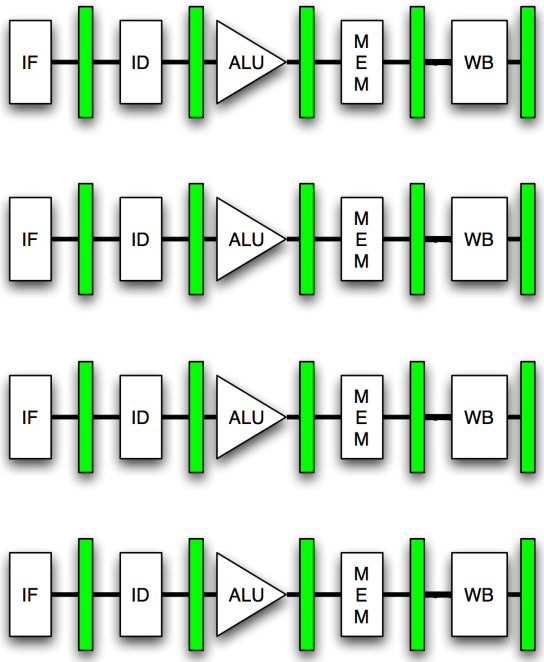
CPU Architecture evolution

Pipeline using

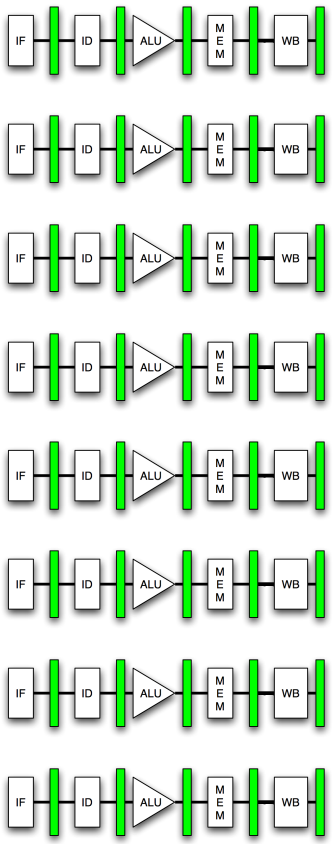


CPU Recent Architectures

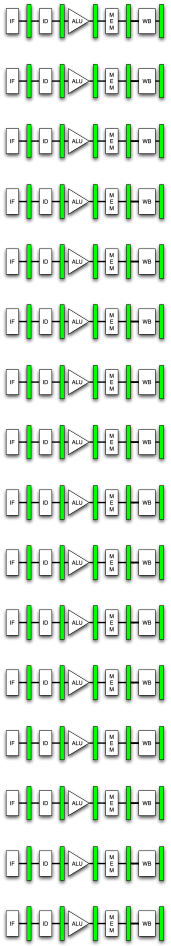
SSE4 4 floats
Instruction set : 2006
CPU : 2007



AVX 8 floats
Instruction set : 2008
CPU : 2011



AVX 512 16 floats
Instruction set : 2013
CPU : 2016



Data format

Efficent only if data are contiguous