Fastest way to run python…is to **NOT** run python

Quivron Loïc

Github

Loïc Quivron

Data Scientist at Macq

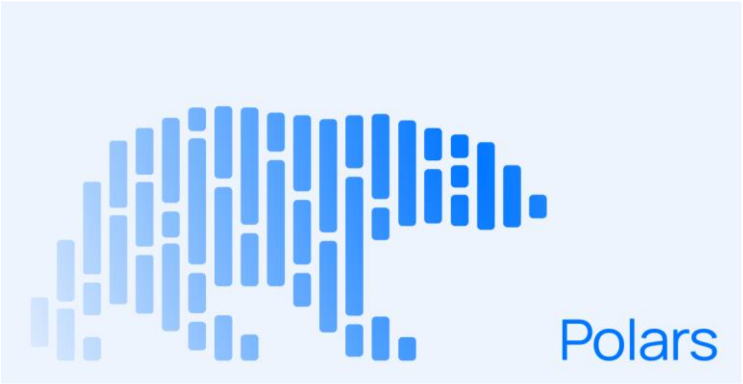
Traffic Forecasting Model

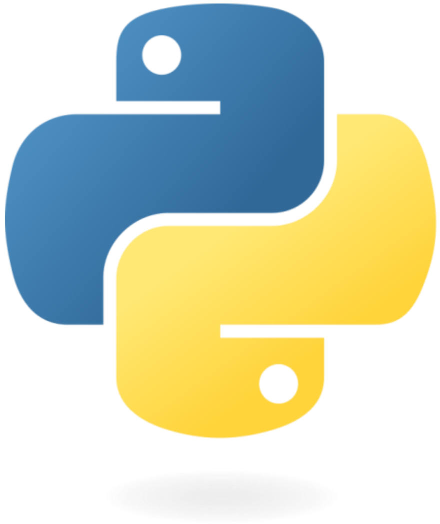
Dashboarding

Data Engineering

How to make python **Faster** ?

Libraries to explore

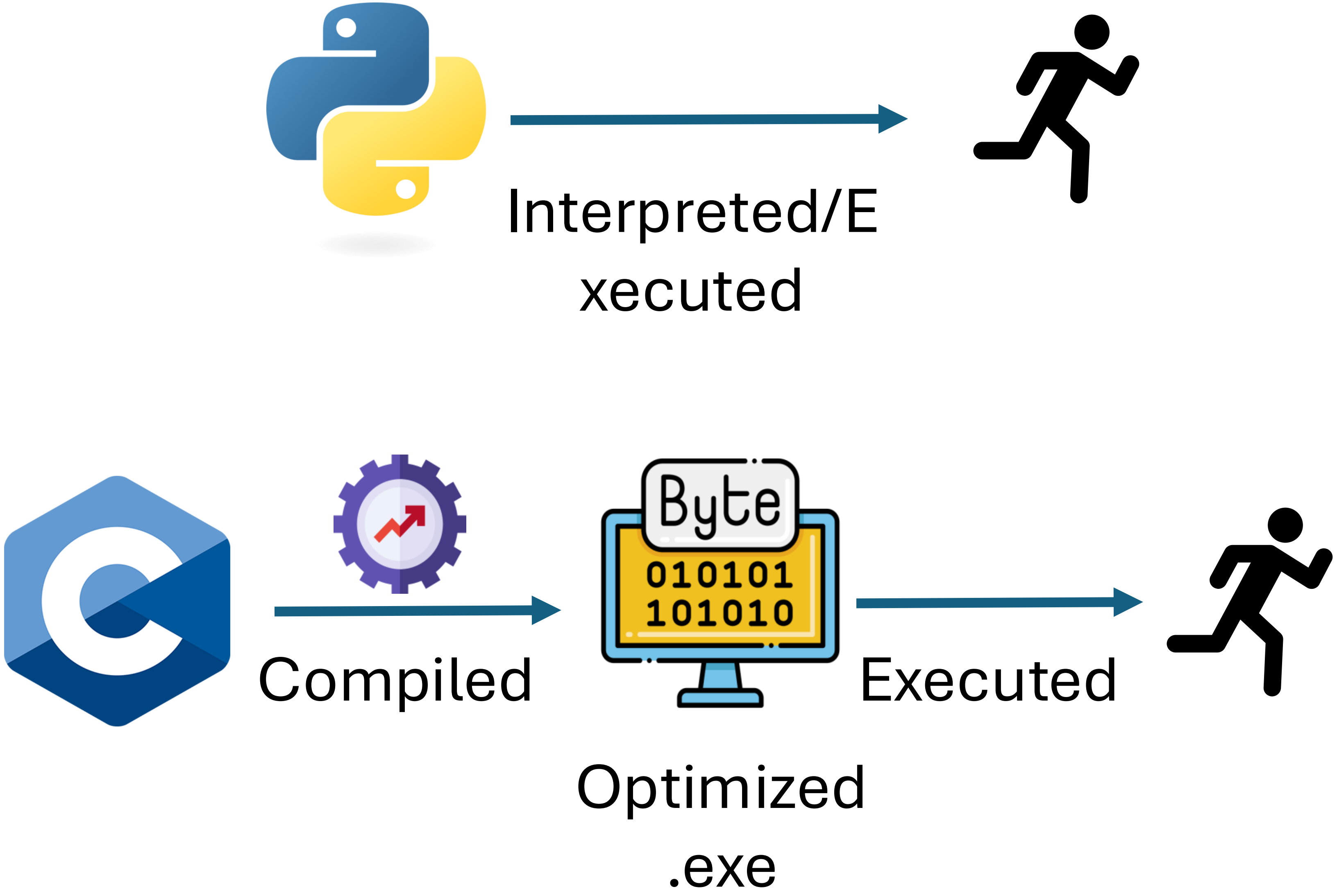




VS

Interpreted

Compiled



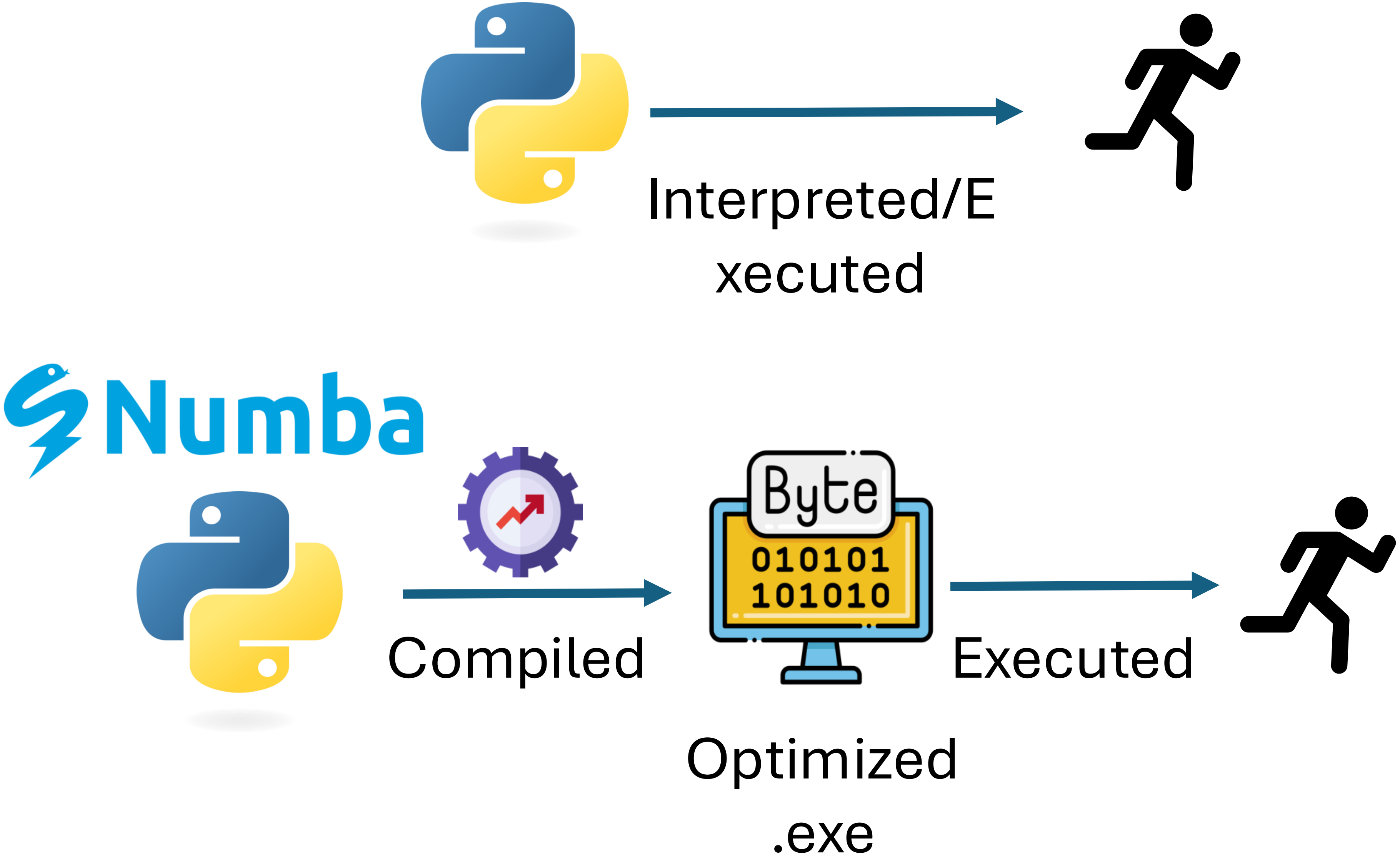
Interpreted/Executed

Compiled

Executed

Optimized

.exe

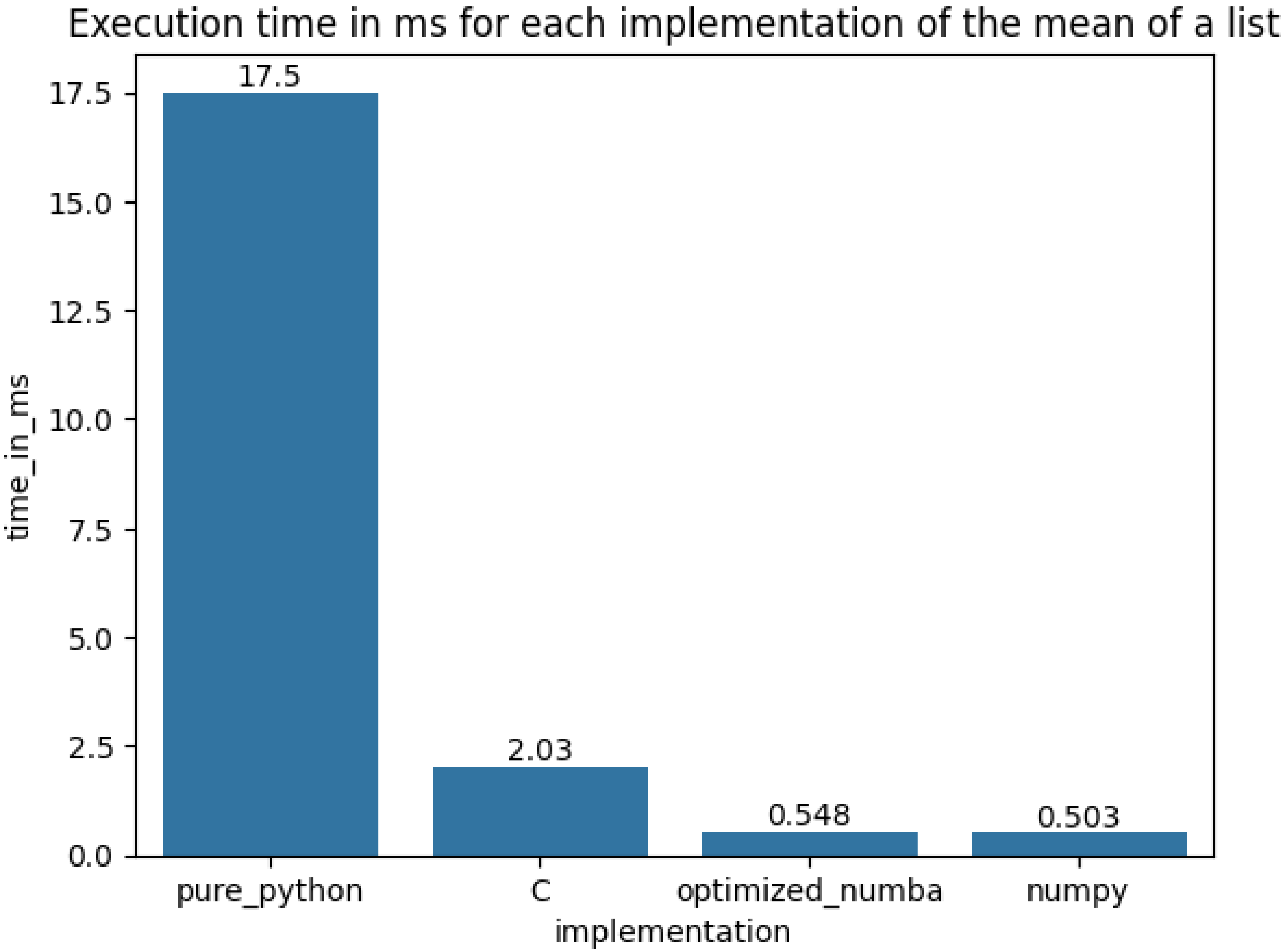


Interpreted/E
xecuted

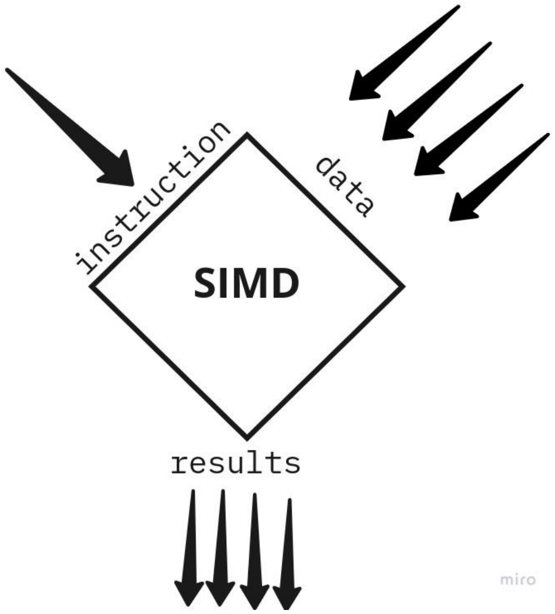
Compiled

Executed

Optimized
.exe


Execution time in ms for each implementation of the mean of a list
time_in_ms
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
17.5
2.03
0.548
0.503
pure_python
C
optimized_numba
numpy
implementation

NumPy
"Let's see who you really are!"
imgflip.com



Vectorization/SIMD

Vectorization/SIMD

Single Instruction Multiple Data :
Performing **one function** on **multiple data** at the same time

Vectorization/SIMD

$$f(x) = a + b$$

Column A

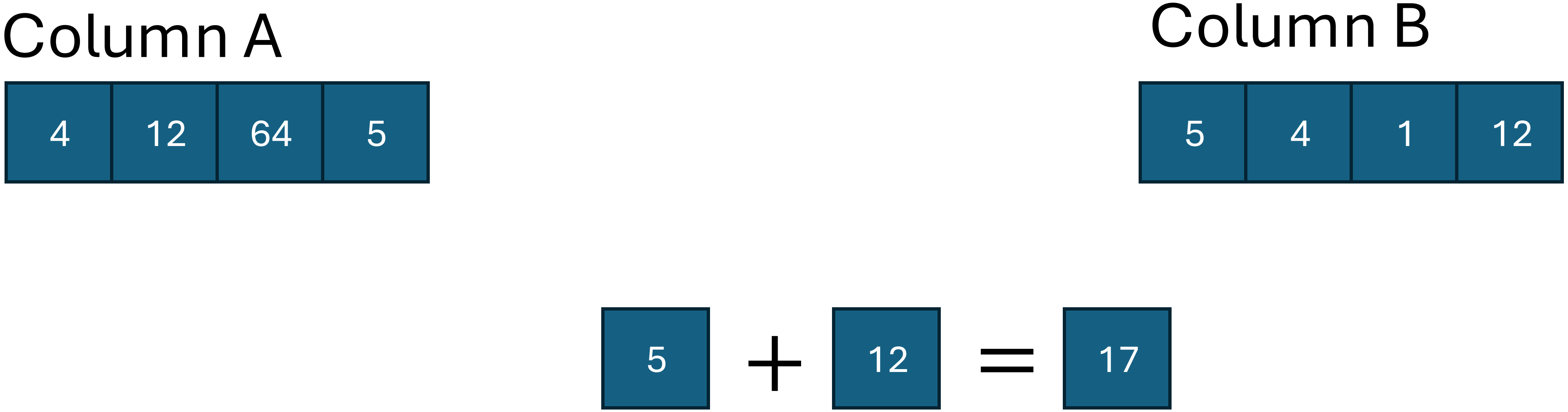
4	12	64	5

Column B

5	4	1	12

Vectorization/SIMD

$$f(x) = a + b$$



Column A

4	12	64	5

Column B

5	4	1	12

$$\boxed{5} + \boxed{12} = \boxed{17}$$

Vectorization/SIMD

$$f(x) = a + b$$

Column A

4	12	64	5

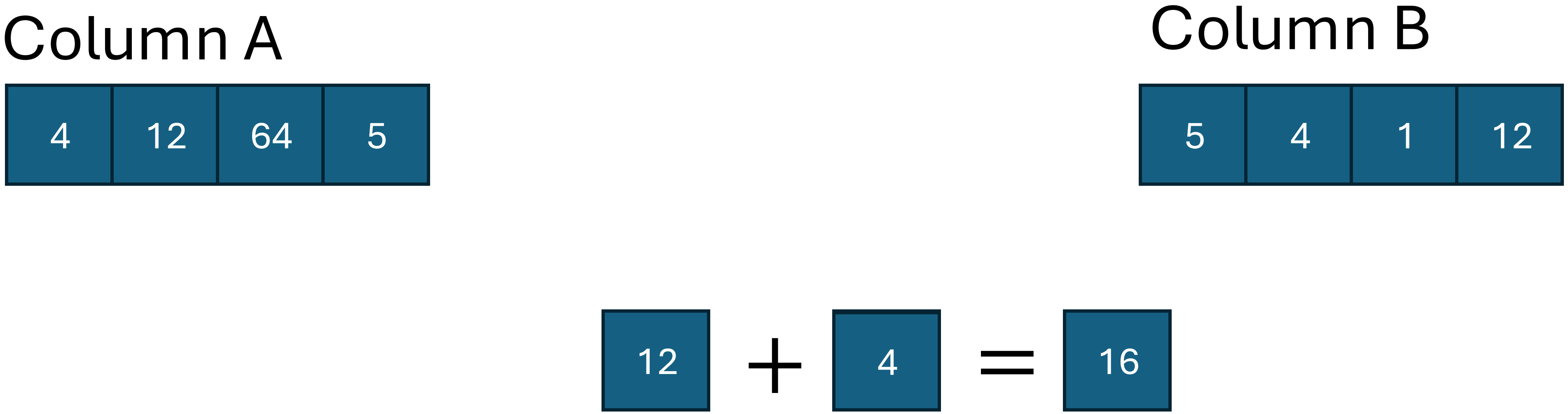
Column B

5	4	1	12

$$\boxed{64} + \boxed{1} = \boxed{65}$$

Vectorization/SIMD

$$f(x) = a + b$$



Column A

4	12	64	5

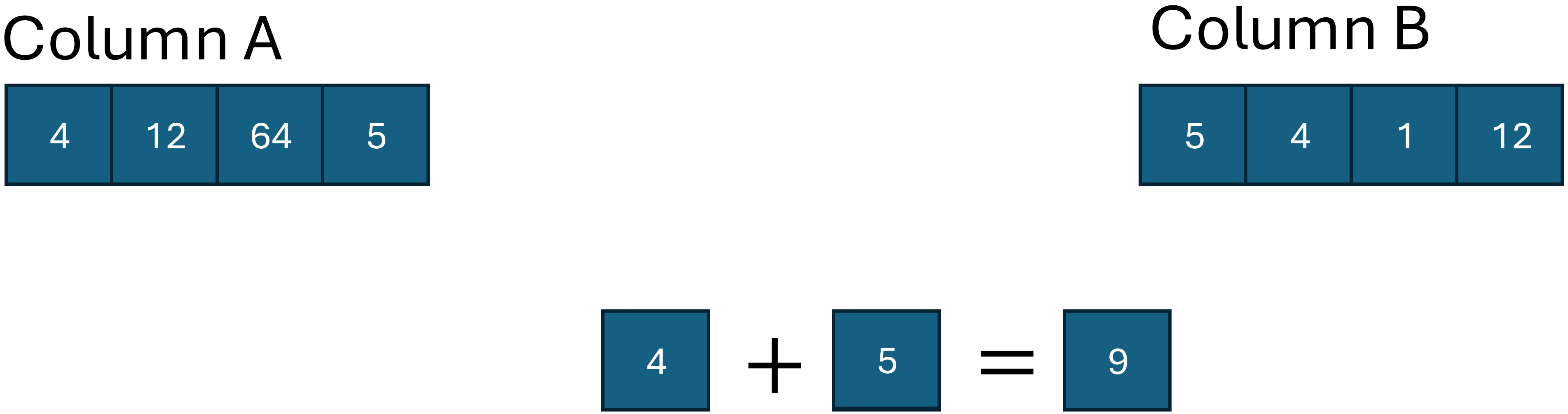
Column B

5	4	1	12

$$\boxed{12} + \boxed{4} = \boxed{16}$$

Vectorization/SIMD

$$f(x) = a + b$$



Column A

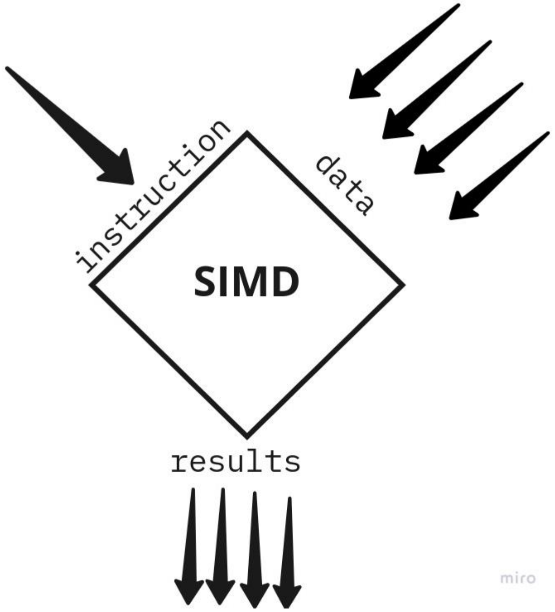
4	12	64	5

Column B

5	4	1	12

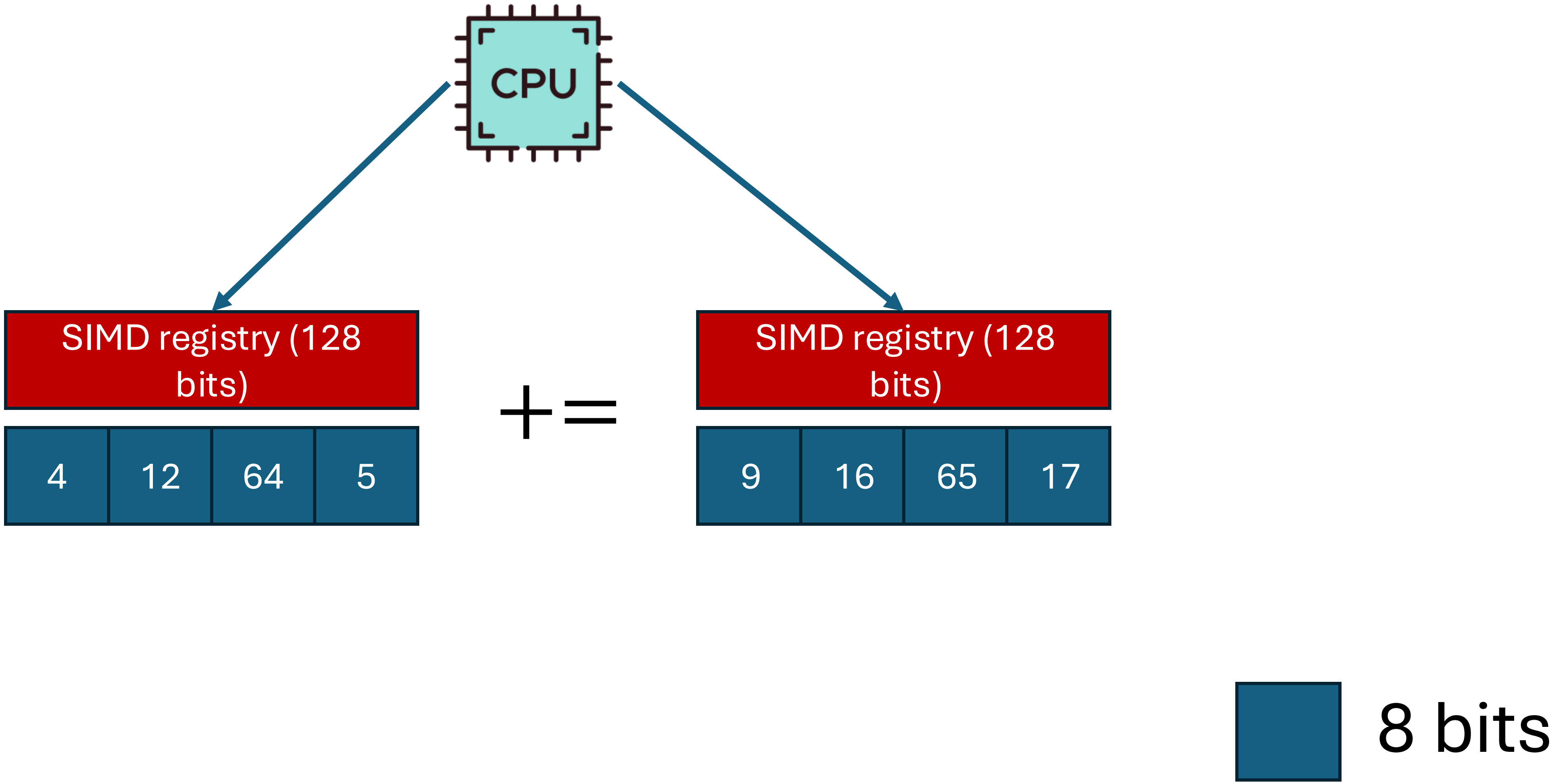
$$\boxed{4} + \boxed{5} = \boxed{9}$$

Vectorization/SIMD



4	12	64	5
+	+	+	+
5	4	1	12
=	=	=	=
9	16	65	17

SIMD -> single core parallelism



CPU

SIMD registry (128 bits)

4	12	64	5

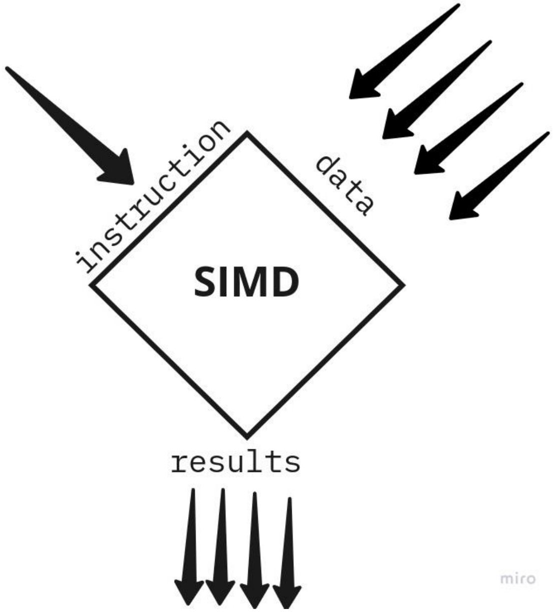
+=

SIMD registry (128 bits)

9	16	65	17

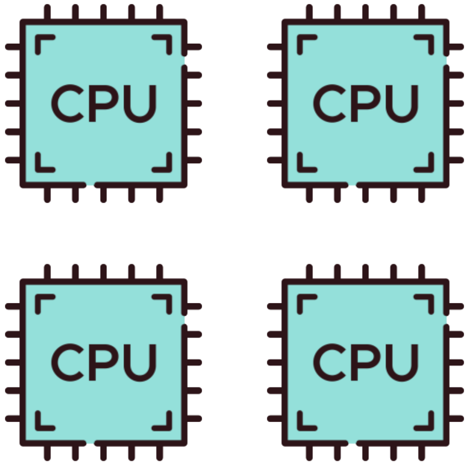
🟦 8 bits

Vectorization/SIMD

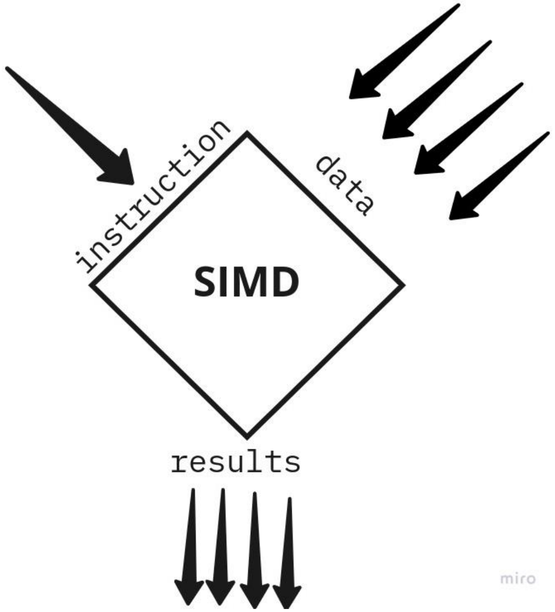


4	12	64	5
+	+	+	+
5	4	1	12
=	=	=	=
9	16	65	17

Why so Fast ?



Multi-threading



Vectorization/SIMD

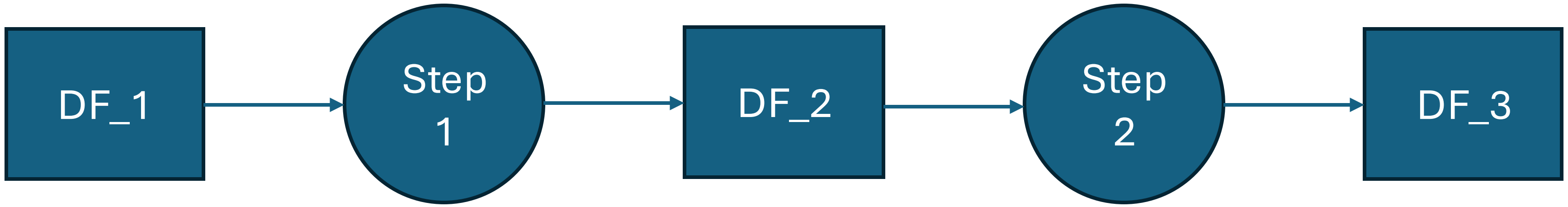
Lazy Evaluation

Parallelization

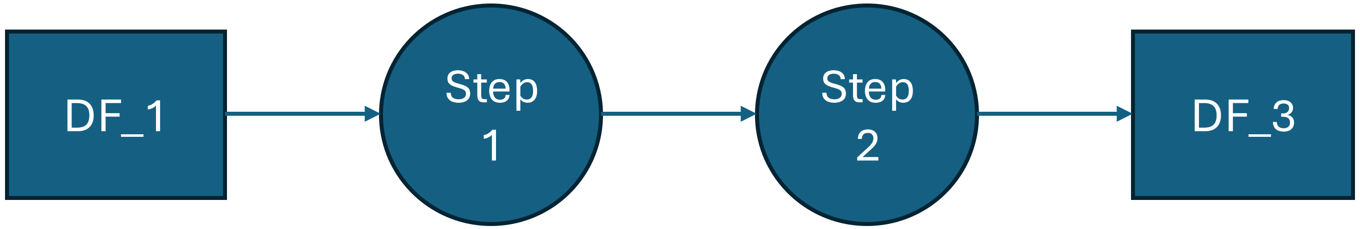
Lazy Evaluation

“Wake me up when I need to collect the data”

Eager



Lazy



Step 1 and 2 can be optimized

Conclusion

- Faster way to use python is to use library written in python or using numba to compile python in bytecode
- While numba is fast and easy, it is far from being ideal in production setting
- Numpy for scientific computations and Polars for DataFrame processing are the way to go