RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

METEO FRANCE

Perspectives on Systems and Retrospective on People

Ryad El Khatib / Météo-France
Tallinn, 29 March 2023

1. Perspectives on Systems

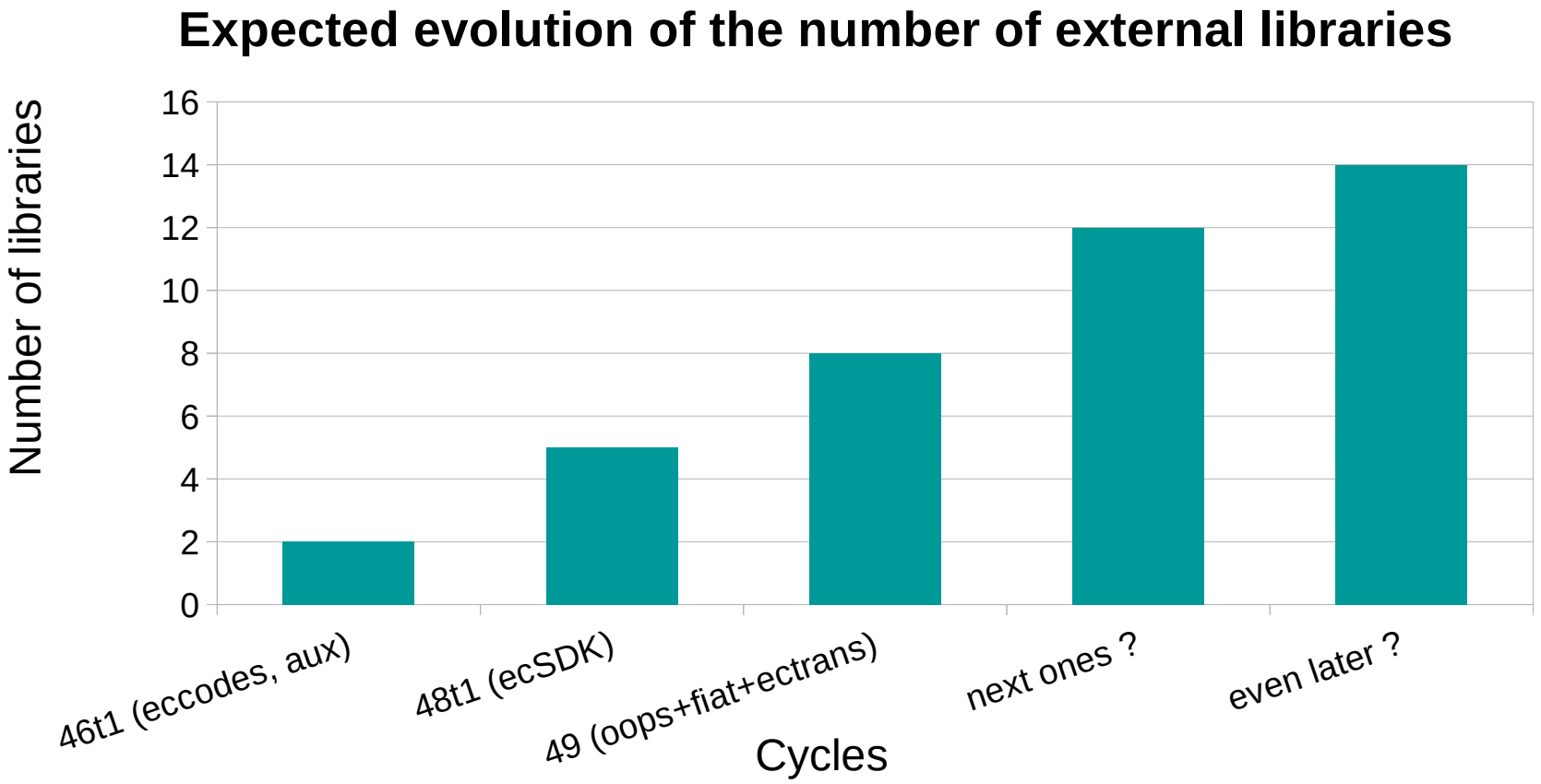
1) The quantic jump from cycle 48t* to cycle 49t0

2) GPUs (and some consequences) for dummies

1) The quantic jump from cycle 48t* to cycle 49t0

- New auxilary files (RTTOV new version and new RRTM data files)
- Several namelist key-variables movments from dynamics
- Project PHYEX (Meso-NH physics) for which mpa/ is its Arome interface
- Part of auxilary code « ifsaux » replaced by an external library « fiat » in another code repository
- Global spectral transforms « trans » replaced by an external library « Ectrans » in another code repository
- « algor » partly moved to fiat or ectrans
- oops_src replaced by «OOPS» in another code repository
- Also new fields in IFS GRIB files (5 snow layers, …)
- New compression algorithm in IFS GRIB files in cycle 48R1 (libaec) requiring an update of eccodes library
- Important code refactoring for GPUs, incl. .fypp files *(from cycle 48t3)*

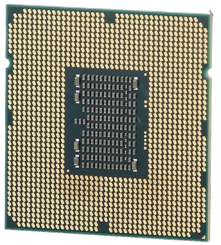
1) The quantic jump from cycle 48t3 to cycle 49t1



=> more complex management of libraries
through multiple repositories

2) GPUs (and some consequences) for dummies

- **GPUs can be seen like massively parallel processors without vectorization**. Still, vectorization is good because it is a sort of micro-parallelization
- **A huge number of points is needed to sufficiently feed a GPU**. Ideally, there should be nested loops on vertical levels and horizontal points.
- **Data should not be transferred from the CPU to the GPU and vice-versa, that would slow down the computation**. Better let the data stay on the GPU



CPU
Single Instruction
Multiple Data
(SIMD – n ~ 8)

VECTOR ENGINE
Vector pipelining
n=256

GPU
Single Instruction
Multiple Threads
(SIMT n ~20000)

2) GPUs (and some consequences) for dummies

Constraints :

- *Memory handling* : automatic arrays can hardly be used on GPU, so that all the data to be transferred should be grouped and allocated on the heap.

- *Computation* : may require loops to be interchanged and code to be reorganized, possibly in-lined, in order to present heavy computational parts and without calls to subroutines that would fragment them.

2) GPUs (and some consequences) for dummies

Strategy :

- *Reorganize the data layer*. This work has started with the help of a source code preprocessor : fypp (https://fypp.readthedocs.io/en/stable/fypp.html).

  Appears in cycle 48t3.

- *Reorganize the code* to isolate the pure computational part, by the creation of an adequate interface layer.

- Develop and use a software before compilation, to transform the source code and make it compliant with the targeted hardware (CPU, GPU, …) : loki, fxtran.

2) GPUs (and some consequences) for dummies

Consequences :

- **Re-learn** how the source-code is organized
- **No synthax array in calculations,** please
- Favourize in-lining : **No more functions encapsulated in modules**
- **Use structures**, not individual variables
- The number of .fypp files is increasing drastically

  => **developpers will have to learn how to code in .fypp files**