



PROGRAMME  
DE RECHERCHE  
NUMÉRIQUE  
POUR L'EXASCALE

# Ecosystème logiciel: modèles de programmation et pile logicielle

H. Barucq, A. Buttari and R. Namyst

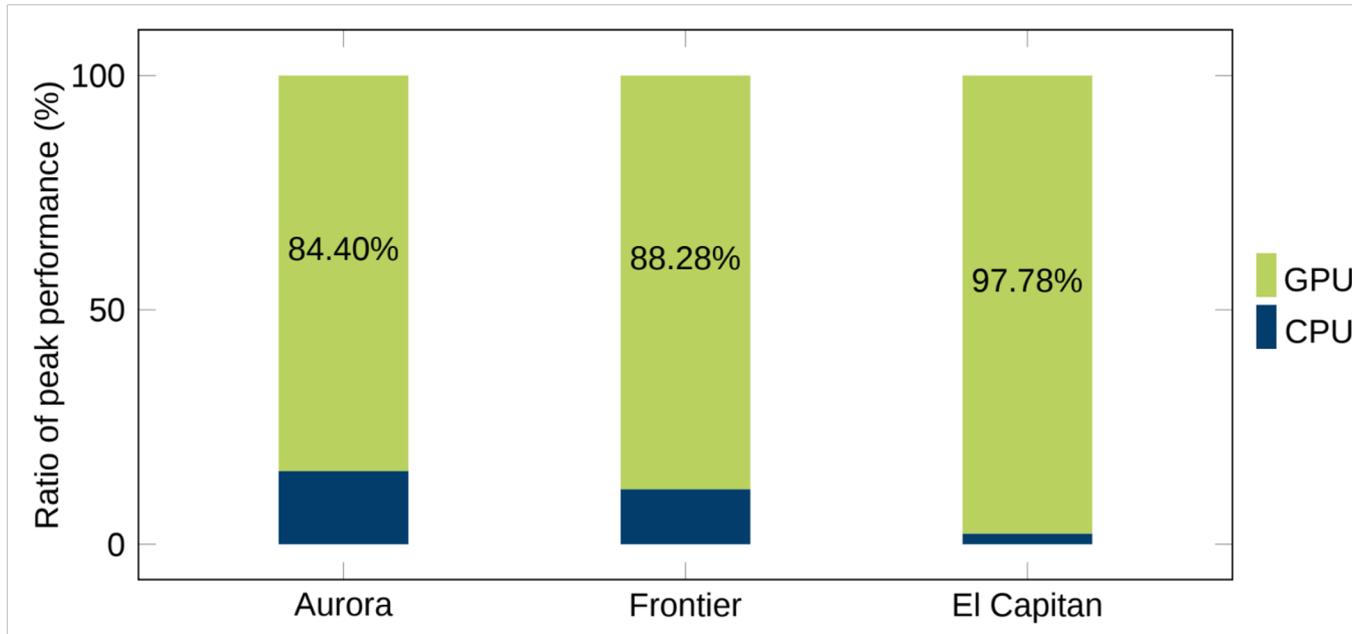
# The elephant in the room



# The elephant in the room: GPUs



# The elephant in the room: GPUs



courtesy of Ivo Kabadshow

# Software

Software features that force us to rethink our algorithms and software

## Issues

- Heterogeneity
- Scale
- Variety

## Objectives

- Efficiency/scalability
- Portability (functional and performance)
- Maintenance and evolution

# The situation

Different technologies address different needs:

- Composability?
- Portability?
- Maintainability/durability?
- Productivity?
- Sovereignty?
- Education?

**Applications:**  
JAX, PyTorch, Julia

**Algorithms, libraries:**  
MPI, PGAS, task-based...

**Kernels:**  
Kokkos, OpenMP,...

# Software sovereignty

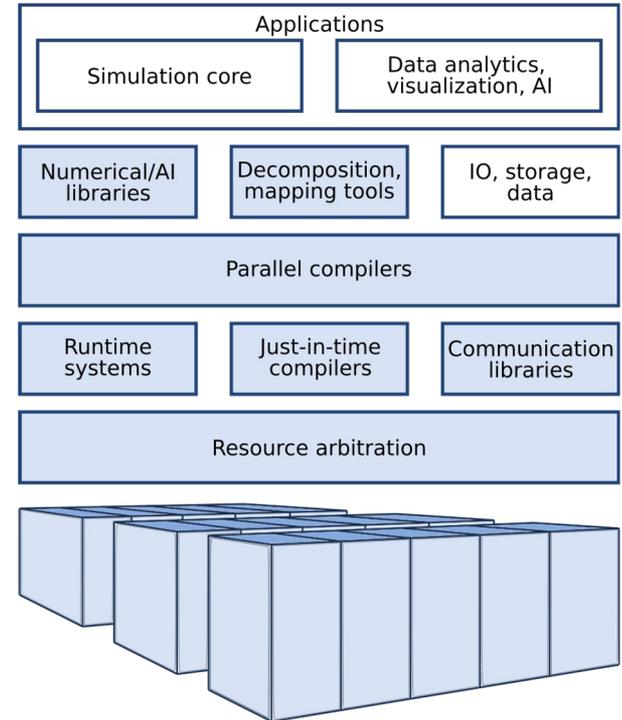
## What software?

## Which external software components do we critically depend on?

- SWOT of the UE software stack
  - GPU software ecosystem?
  - Low level communication libraries?

## How do we really setup a codesign approach with industry?

- EVIDEN
- SiPearl
- Etc.



# Directions in Software

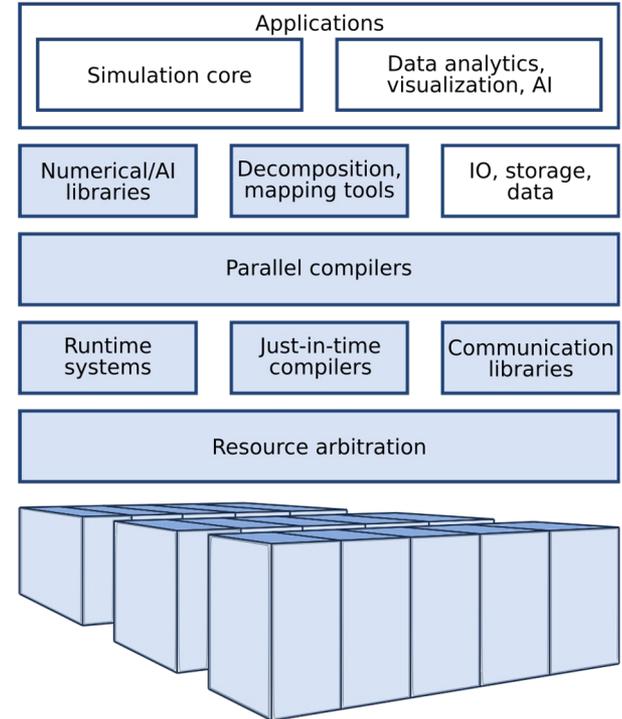
What will the post-exascale software stack look like?

## If machines are going “GPU-Only”

- What software is still relevant?
  - US GPU companies are developing complete GPU software stacks
  - How do we decrease our dependency on these providers?

## Will light come from hardware diversity and software composability?

- Seems a sound approach for HPC/IA/Quantum hybridation
- Disaggregated architectures



# Toward a unique (and boring) ecosystem?

The battle for the best programming model/language is over

**Generative Ais will soon be able to translate Fortran to C++**

- Or better, Fortran to Kokkos!

**Wait, why focusing on HPC-friendly languages?**

- GenAi can translate Fortran to Python (or JAX)
- So maybe we should prepare for the big jump to world.py
  - Are we gonna throw all our HPC expertise to trash?
  - Or shall all expertise be integrated into Python libraries?