



DE LA RECHERCHE À L'INDUSTRIE

## **TRUST/TrioCFD, démonstrateur CExA pour la DES**

Pierre Ledac

DES/DM2S/SGLS/LCAN



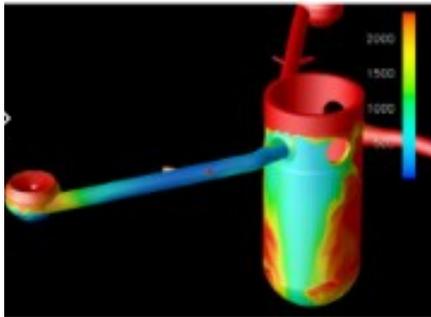
Plateforme de thermohydraulique (DES/DM2S/SGLS/LCAN)

Application TRUST dédiée à la CFD

- Mécanique des fluides

- Incompressible/faiblement compressible
- Mono/Diphasique
- Suivi d'interfaces

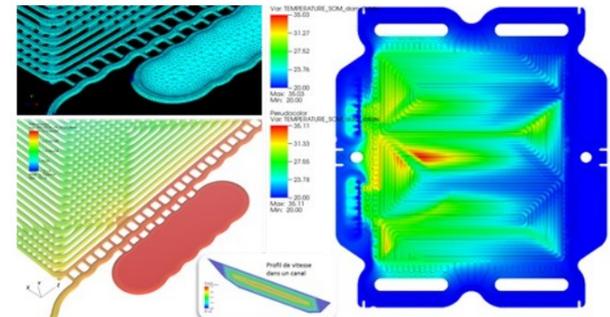
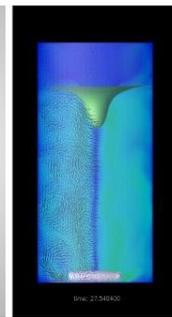
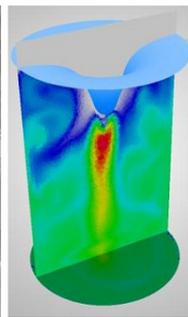
- Domaines d'application



Réacteur REP



Agitateur à effet vortex

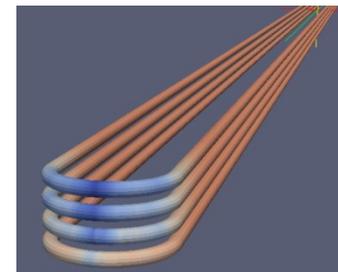
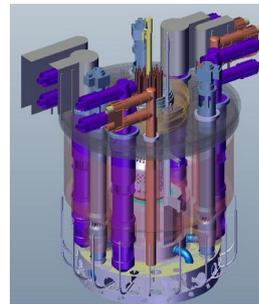
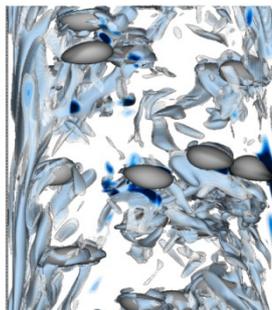
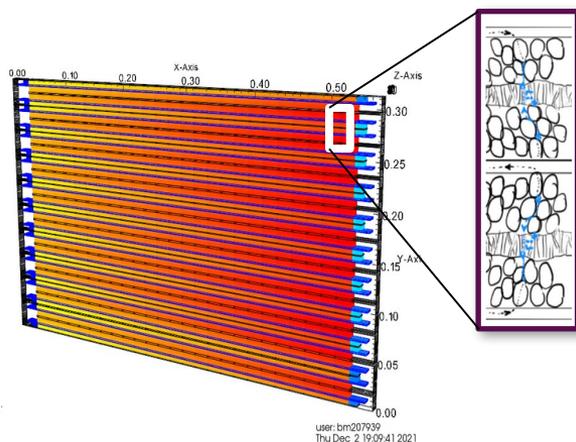


Pile à combustible

- C++, MPI, OpenSource <https://github.com/cea-trust-platform>

# APPLICATIONS SUR TRUST

- TrioCFD : CFD monophasique + Front Tracking
- FLICA5 : code thermohydraulique diphasique niveau composant (code poreux)
- SympyToTRUST (STT) : code multiphysique pour la simulation (entre autres) des batteries et piles à combustible
- CATHARE3D: modélisation 3D du cœur réacteur couplée au système
- Trio-IJK : modélisation des écoulements diphasiques à l'échelle locale instantannée
- TrioMC : modélisation de la thermohydraulique du cœur des RNR
- GENEPI+ : modélisation de la thermohydraulique des GV
- PAREX+ : code de chimie 0D (pas de maillage)



2014

- **First** use of GPU in TRUST (**PETSc**)
  - Single node GPU, limited to one solver (GMRES/Jacobi)



2020

- Test **AmgX**, Nvidia GPU library
  - Multi-node GPU, more solvers available (CG/Multigrid)



2021

- Add **AmgX** library (Nvidia) to TRUST (1.8.3)
- Nvidia Hackathon participation
  - Challenge TRUST team to evaluate **OpenACC** approach (parallel pragma directives)

2022

- First study with a GPU **partial** accelerated **TrioCFD** (Jean-Zay)
- Partial port on **AMD** GPU with **OpenMP** on **Adastra** (GENCI contract)

2023

- First run with a GPU **fully** accelerated TRUST (Topaze)

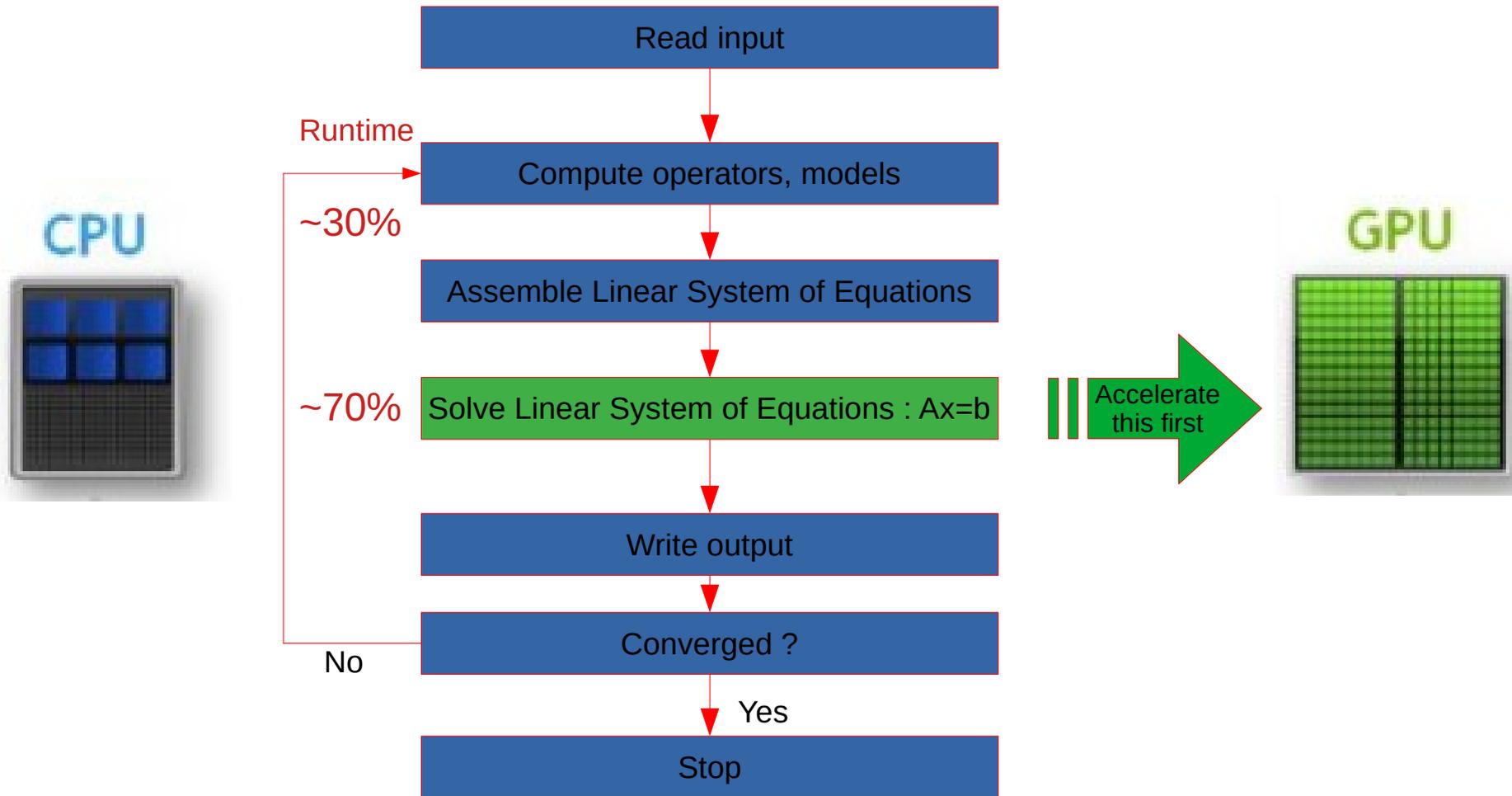
2024

- Enable **CExA** (**Kokkos** framework for CEA) in TRUST/TrioCFD

2025

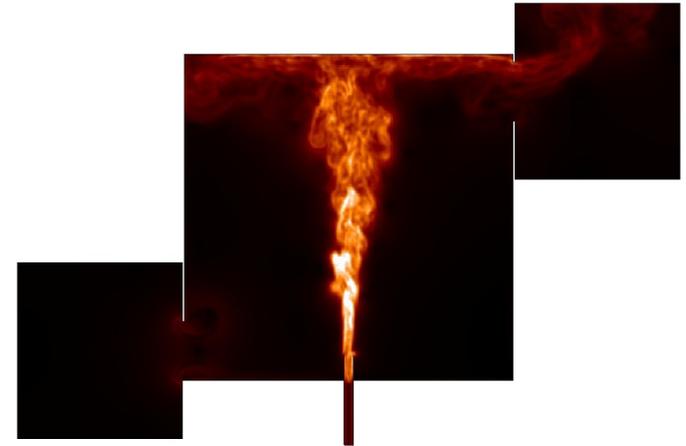
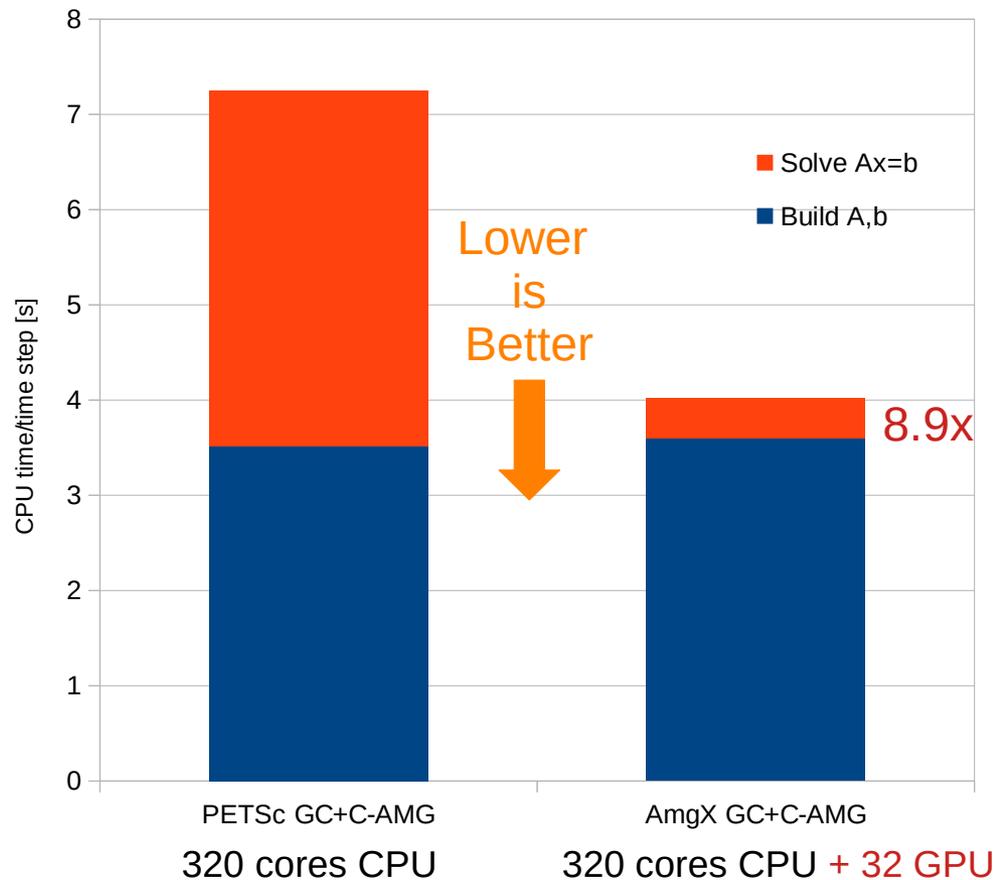
- French exascale supercomputer (**ARM** CPU/**Nvidia** GPU?)

- ◆ **Detect** the most CPU expensive algorithms candidate to GPU



- ◆ **Benefit firstly** from dedicated linear algebra libraries (e.g. **AmgX** for GPU Nvidia)

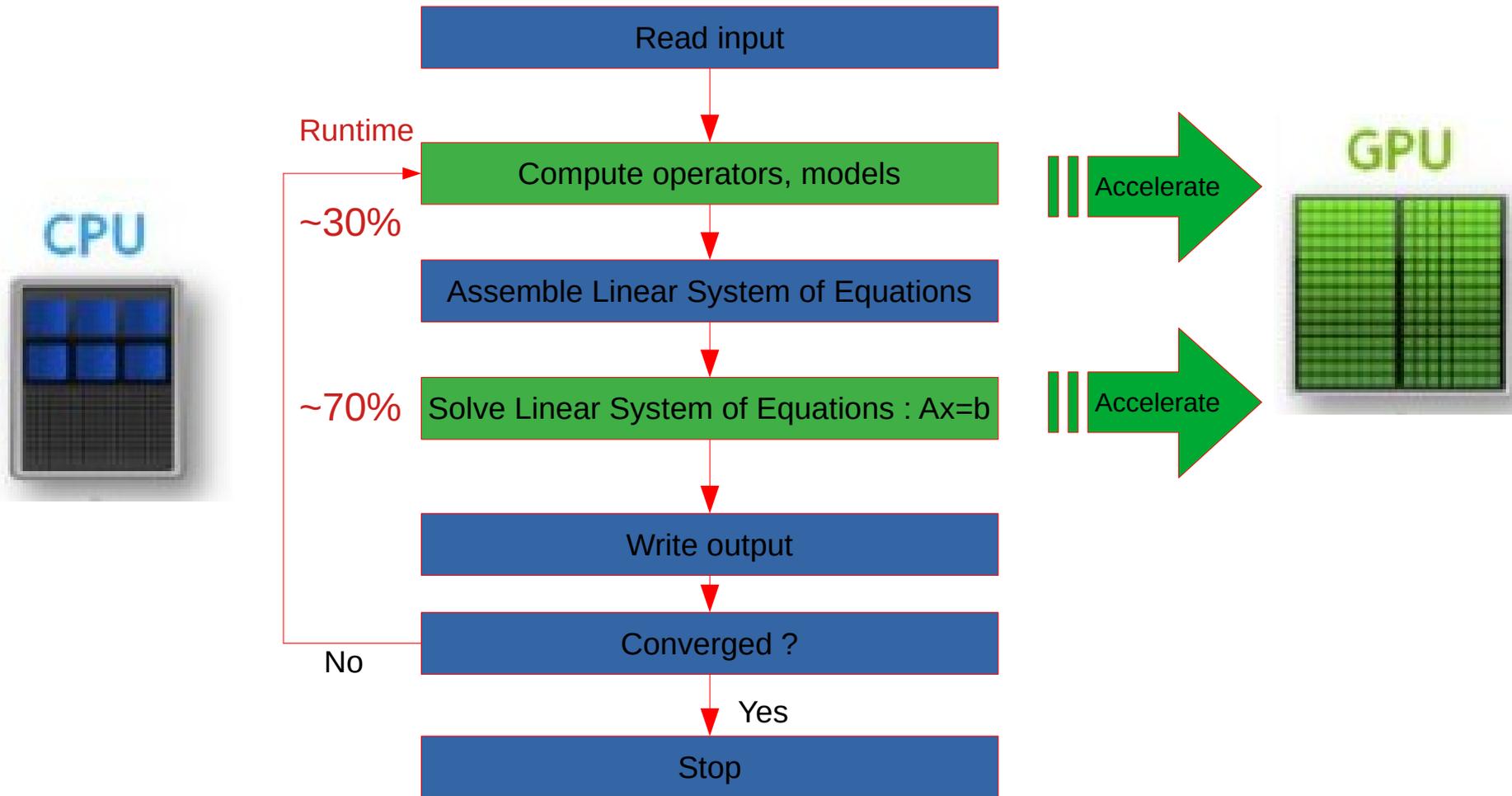
## DNS simulation (TrioCFD 1.8.3) on Irene Joliot cluster (TGCC)



- ▶ Mini-GAMELAN geometry
- ▶ Structured mesh (VDF)
- ▶ 80M cells (250K/core)
- ▶ Unsteady DNS
- ▶ GC + C-AMG solver
- ▶ 50% time into solver

▶ **1.8x** acceleration for the simulation

- ◆ **Detect** the most CPU expensive algorithms candidate to GPU



- ◆ **Benefit** from dedicated linear algebra libraries (e.g. **AmgX** for Nvidia, **rocALUTION** for AMD)
- ◆ **Introduce** parallel directives (**OpenMP**) for the the most CPU expensive loops



DE LA RECHERCHE À L'INDUSTRIE

## Avancement TRUST sur GPU (OpenMP)

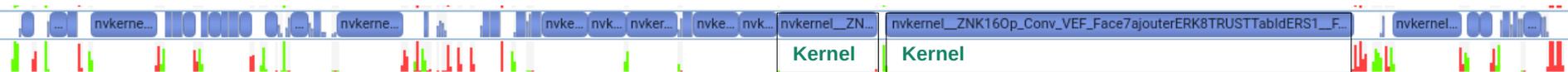
MonoGPU (2.6M DDL)

MultiGPU (2.6M DDL)

MultiNoeud-MultiGPU (80.8M DDL)

- 70 **kernels** OpenMP-target
- **Jeu** de 10 fonctions TRUST pour gérer:
  - données CPU ↔ GPU
  - lancements de Kernel
- **Opérationnel** sur:
  - PC (Nvidia A3000, A5000, A6000)
  - Jean-Zay (V100), Topaze (A100), Adastra (AMD MI250)
- **Optimal** (les données restent sur le device sauf CL et IO):
 

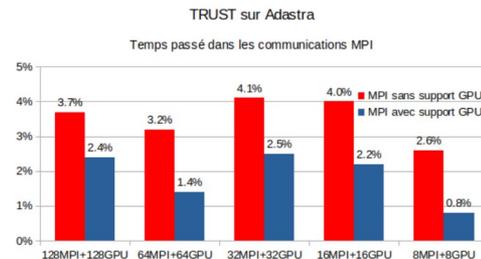
90% sur GPU 5% de copies H→D et D→H 5% IO/CPU

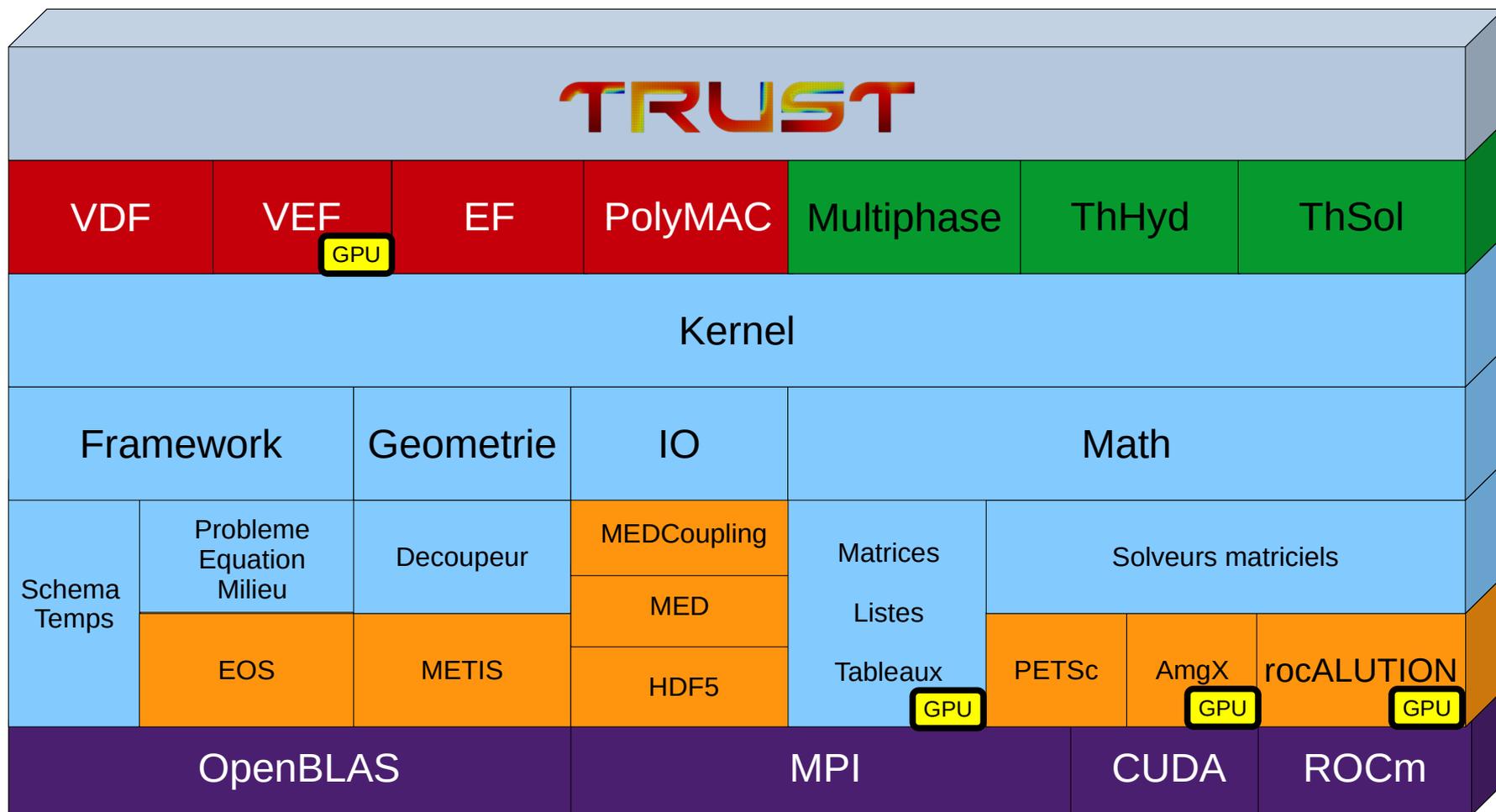


- // Communications GPU-GPU **implémentées**

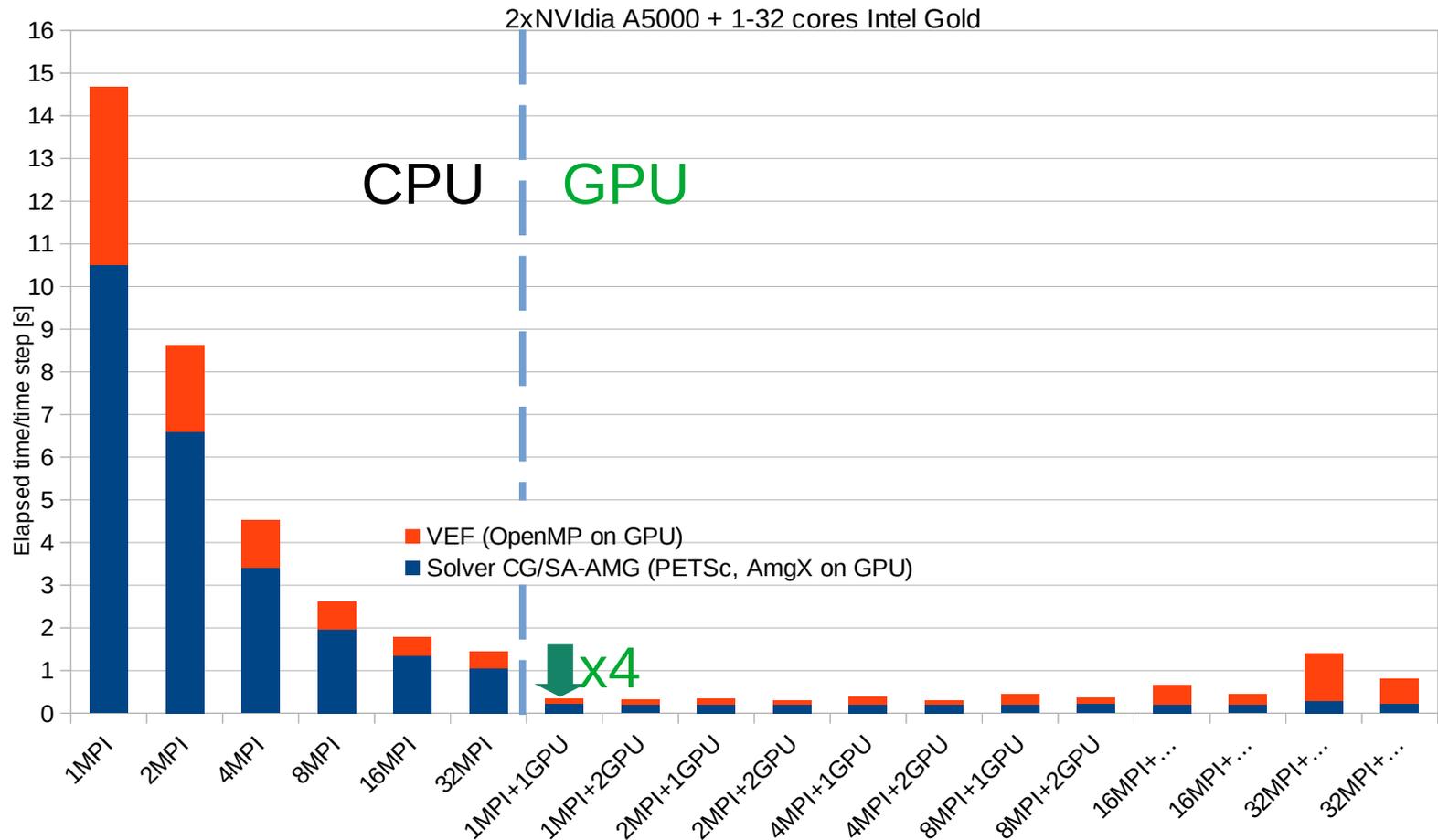
mais pas encore fiable

- -30 à -70% sur le coût des communications MPI dans TRUST:





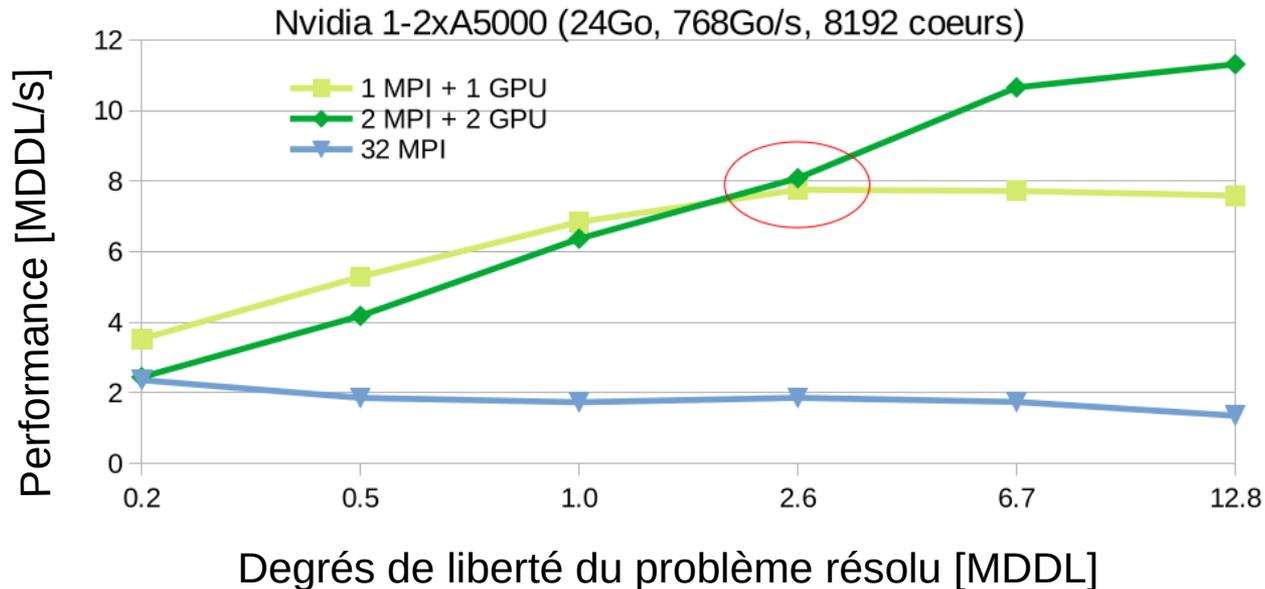
Poiseuille 3D simulation with TRUST 1.9.2 (2.6e6 cells)



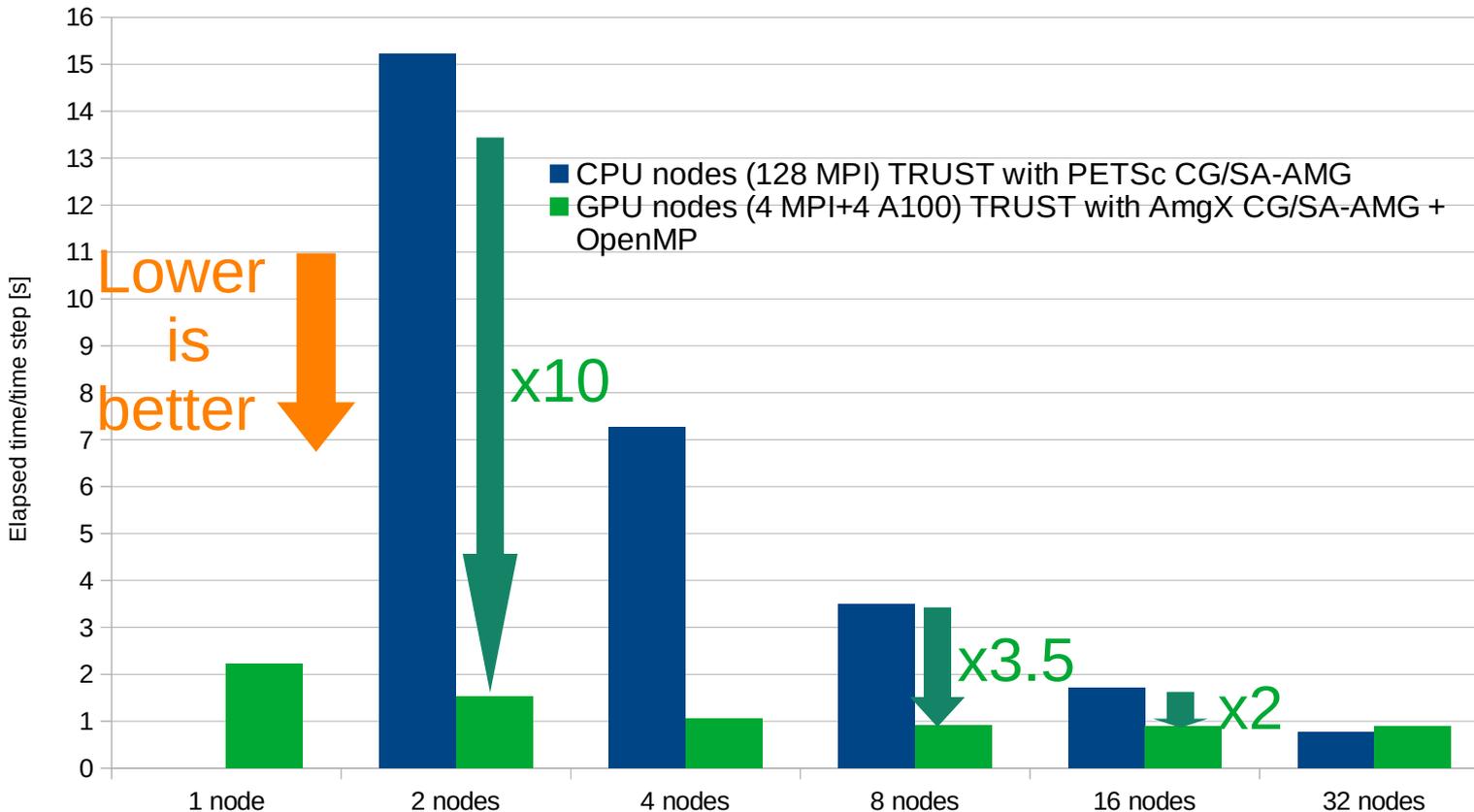
- **Accélération x4** mais **gain faible** à utiliser 2 GPUs
  - Device **sous utilisé**

- Importance de faire travailler le GPU de façon intensive

Higher  
is  
better

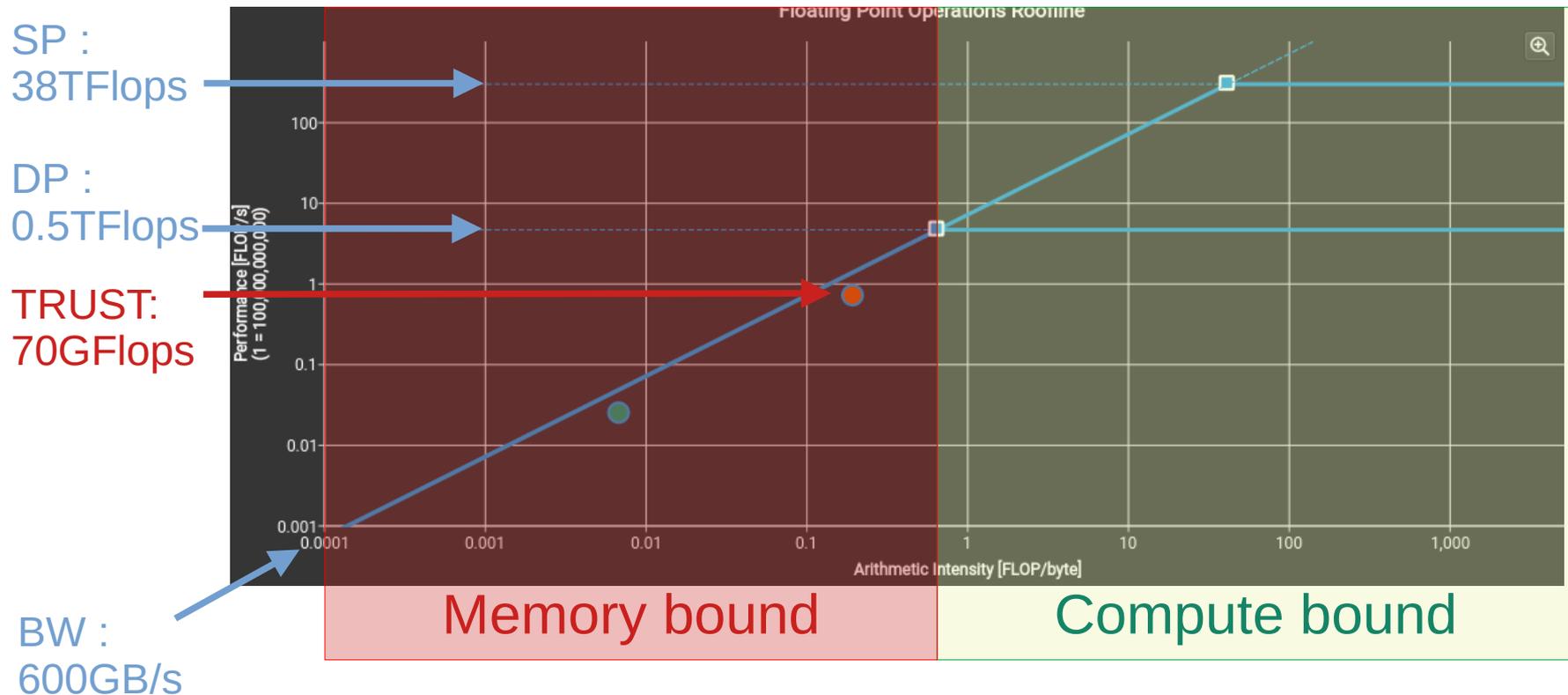


- Sur GPU, l'efficacité **augmente** avec la taille du problème
- Règle pour définir les ressources en GPU: **2-3 MDDL par device** ?
  - Dépend de la physique, du device, si communications directes, etc...

Poiseuille 3D simulation with TRUST 1.9.2 (80e6 cells)  
Topaze (CCRT)

- A nombre de noeuds égal:
  - accélération nette sur **noeuds GPU** par rapport aux **noeuds scalaires**
  - **tant** que que le device est bien occupé (8 nodes: 2.5 MDDL/GPU)

Analyse “**roofline**” par profiler Nsight Compute (Nvidia A6000):



TRUST: comportement **similaire** des algorithmes sur CPU et GPU

- **Performance limitée** par la bande passante mémoire
- **15%** de la puissance crête double précision du GPU seulement...  
→ **Rematérialisation** : recalculer au lieu de stocker les données



DE LA RECHERCHE À L'INDUSTRIE

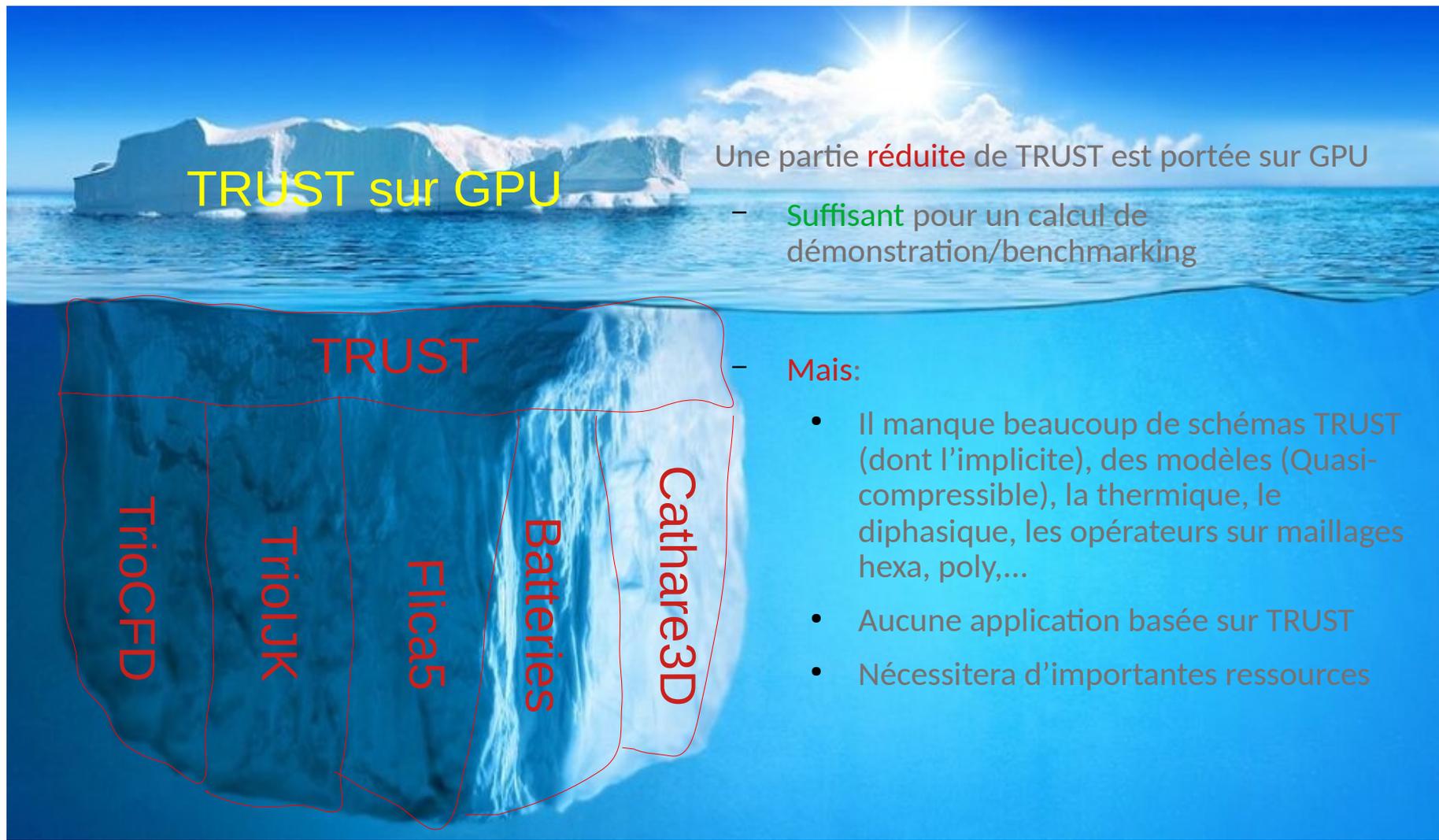
**In CExA we trust !**

An iceberg floating in the ocean under a bright sun. The visible tip of the iceberg is labeled 'TRUST sur GPU' in yellow. The much larger, submerged part of the iceberg is visible below the water line, illustrating that only a small portion of the total TRUST workload is currently handled by GPUs.

## TRUST sur GPU

Une partie **réduite** de TRUST est portée sur GPU

- **Suffisant** pour un calcul de démonstration/benchmarking



Une partie **réduite** de TRUST est portée sur GPU

- **Suffisant** pour un calcul de démonstration/benchmarking
- **Mais:**
  - Il manque beaucoup de schémas TRUST (dont l'implicite), des modèles (Quasi-compressible), la thermique, le diphasique, les opérateurs sur maillages hexa, poly,...
  - Aucune application basée sur TRUST
  - Nécessitera d'importantes ressources

- **Apports de CExA à TRUST/TrioCFD**
  - Portage **accéléré** des algorithmes sur GPU
  - **Portabilité** des performances
  - **Déploiement** de l'application
  - Force d'une **communauté**
  
- **Apports de TRUST/TrioCFD à CExA**
  - Retour **d'expérience** du portage GPU avec OpenMP
  - **Emulation**: performance actuelle du code à atteindre/dépasser
  - Equipé **motivée** !



kokkos



DE LA RECHERCHE À L'INDUSTRIE

**Merci de votre attention**