

# CROCO

Coastal and Regional Ocean COmmunity model

# CROCO Webinars #2

## A Walkthrough of CROCO on GPUs

Friday, April 10th, 2026

Presenter : Cyril Nguyen (CNRS)

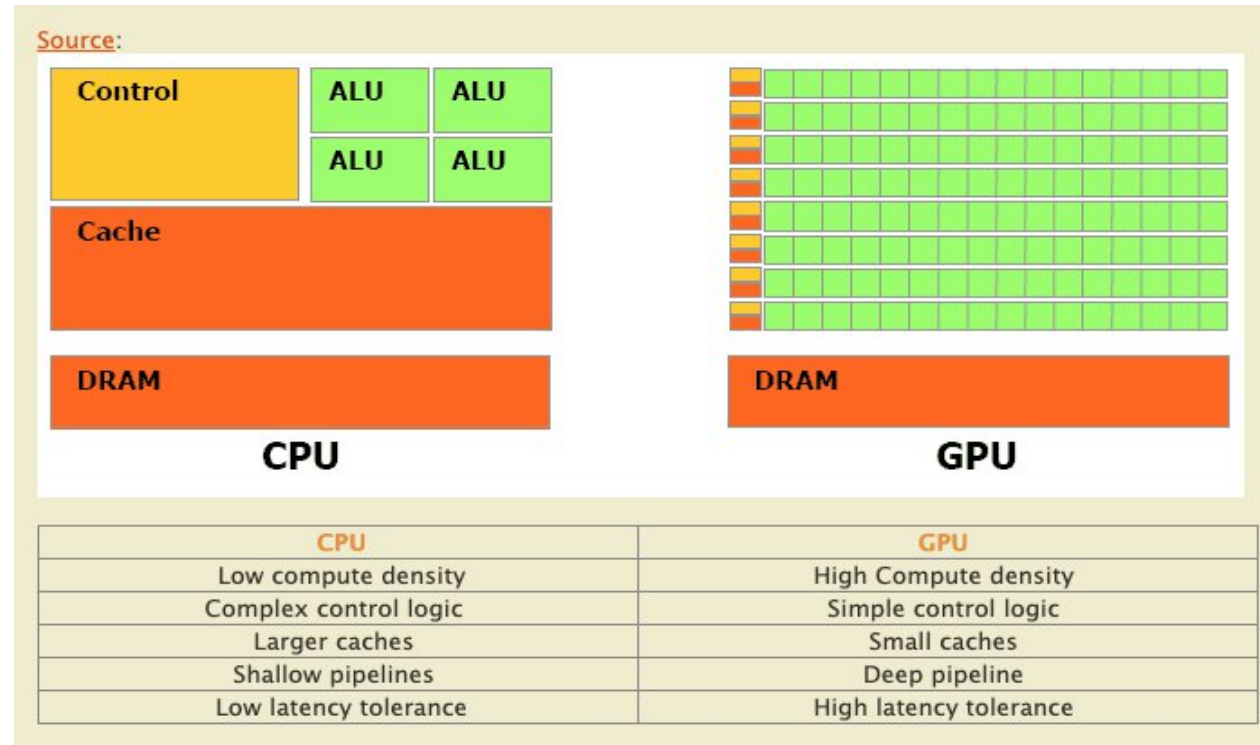
On behalf of the CROCO development team



# Why GPUs?



# Hardware



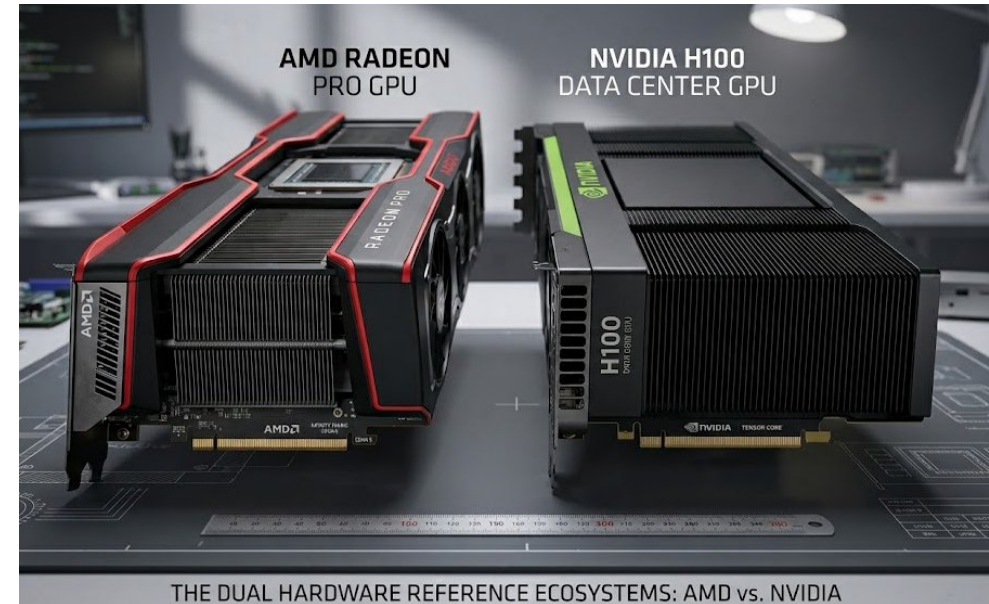
- CPU "general-purpose" / GPU specializes in parallel computing
- IDLE

# Software Stack & Toolkit

AMD / NVIDIA

<https://developer.nvidia.com/hpc-sdk>

- Fortran Compiler
- OpenACC Directives
- GPU Math Libraries
- Optimized for Your CPU
- Multi-GPU Programming (MPI)
- Nsight Performance Profiling
- **LINUX**(Windows Subsystem for Linux ?)



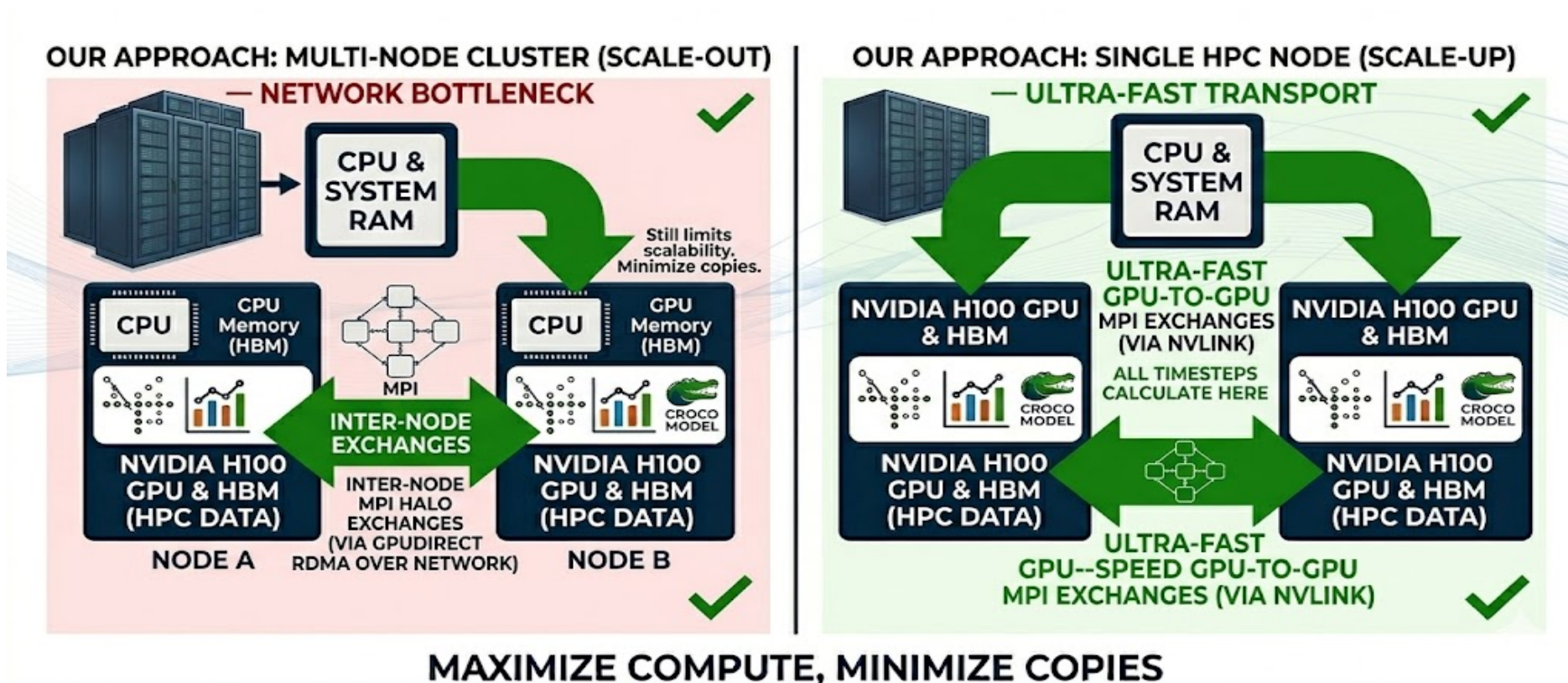
# Programming Strategy: OpenACC

- <https://www.openacc.org/>

```
!$acc kernels copyin(a, b) copyout(c)
!$acc loop
    do 10 i = 1, n
        c(i) = a(i) * b(i)
    10 continue
!$acc end kernels
```

- Copy cpu <-> gpu take times
- « Handmade » kernels

# Memory Philosophy: GPU-Resident



## next slides

- How compile and launch CROCO on GPUs
- Validation and performance results
- Change or add code : synthetic view
- Recipes

# Prerequisites & Environment

- To Compile and Run on GPU
  - Nvidia hpc toolkit for compiler
  - and/or MPI (Nvidia ou system)
  - cppdefs.h
- jobcomp detect OPENACC

```
#elif defined BASIN
/*
!
!
*/
# undef OPENMP
# undef MPI
# define OPENACC
# define UV_ADV
# define UV_COR
# define UV_VIS2
# define SOLVE3D
# define TS_DIF2
# define BODYFORCE
# define ANA_GRID
# define ANA_INITIAL
# define ANA_SMFLUX
# define ANA_STFLUX
# define ANA_BTFLUX
# define NO_FRCFILE
```

Basin Example  
=====

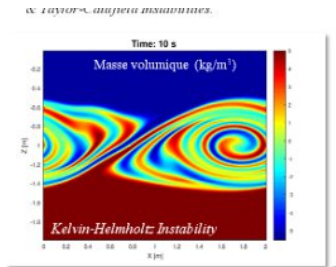
# jobcomp and run : Personal to Cluster

```
elif [[ $FC == pgfortran || $FC == nvfortran ]] ; then  
  CPP1="cpp -traditional -DLinux -DXLF"  
  CFT1="$FC"  
  FFLAGS1="-g -fast -r8 -i4 -mcmode=medium -Mbackslash"  
  #FFLAGS1="-g -O0 -C -Kieee -r8 -i4 -traceback -mnofma"
```

```
#  
# determine if OPENACC compilation is needed  
#  
unset COMPILEOPENACC  
echo "Checking COMPILEOPENACC..."  
if $($CPP1 testkeys.F | grep -i -q openaccisdefined) ; then  
  COMPILEOPENACC=TRUE  
  if [[ $FC == pgfortran || $FC == nvfortran ]] ; then  
    FFLAGS1="$FFLAGS1 -acc -Minfo=accel"  
  fi  
fi  
"
```

Cluster compile on node (Grace Hopper)

# Test Cases to Real Configs



- many test case
  - CVTK Ok
- Gibraltar
  - (tide, forcing...)
- Results on BASIN
- Gibraltar
- not all cppdefs.h combinaison

Acoustic Waves

Canonic Shear Instabilities

Convective Instability

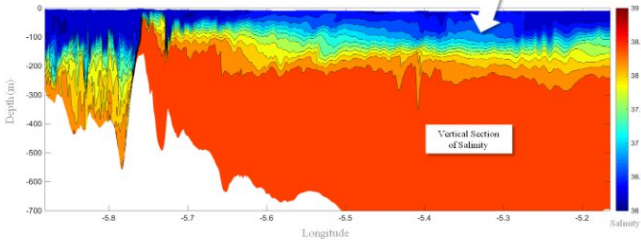
2-Layer Regimes

DSL Soliton

Gibraltar 3D Benchmark

Vertical Section (Gibraltar Experiment)

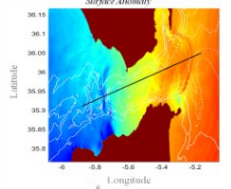
*M. Hill: Ph.D thesis, L. Bordalo (Purdue), Croco Dev. Team.*



Vertical Section of Salinity

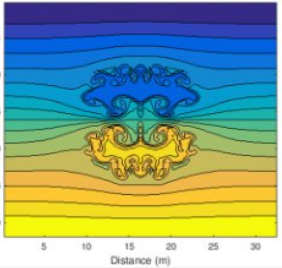
Internal Waves

Surface Anomaly

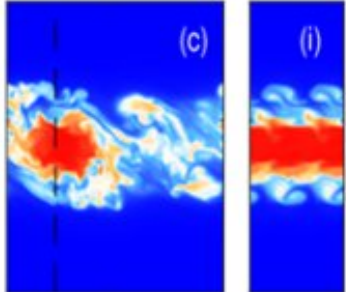


Latitude

Longitude



Baroclinic Jet



Dv Sections of the Bay of Biscaia

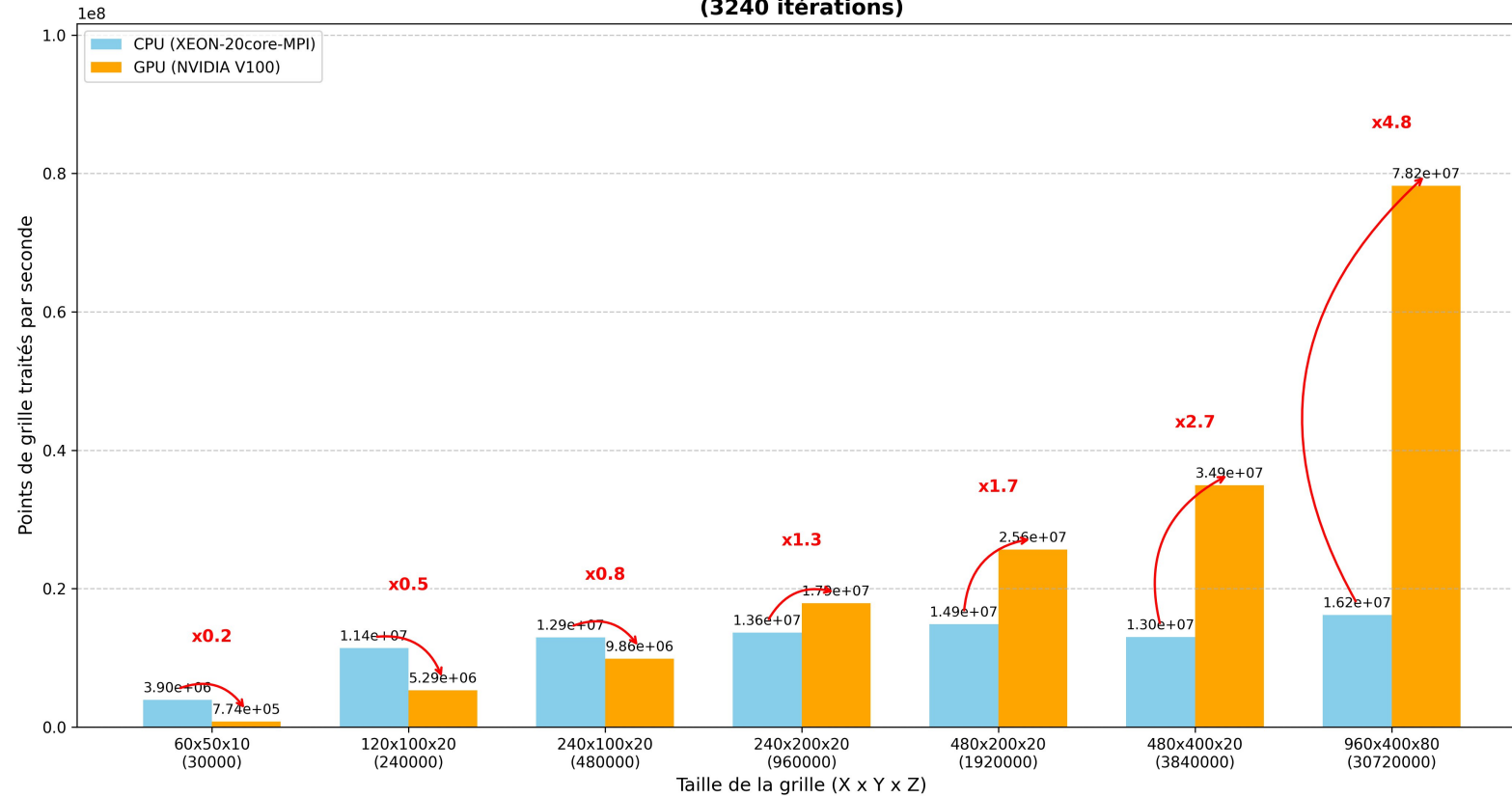
t = 45

Double upward-downward instability

# Performance Results

## Nuwa cluster (LAERO)

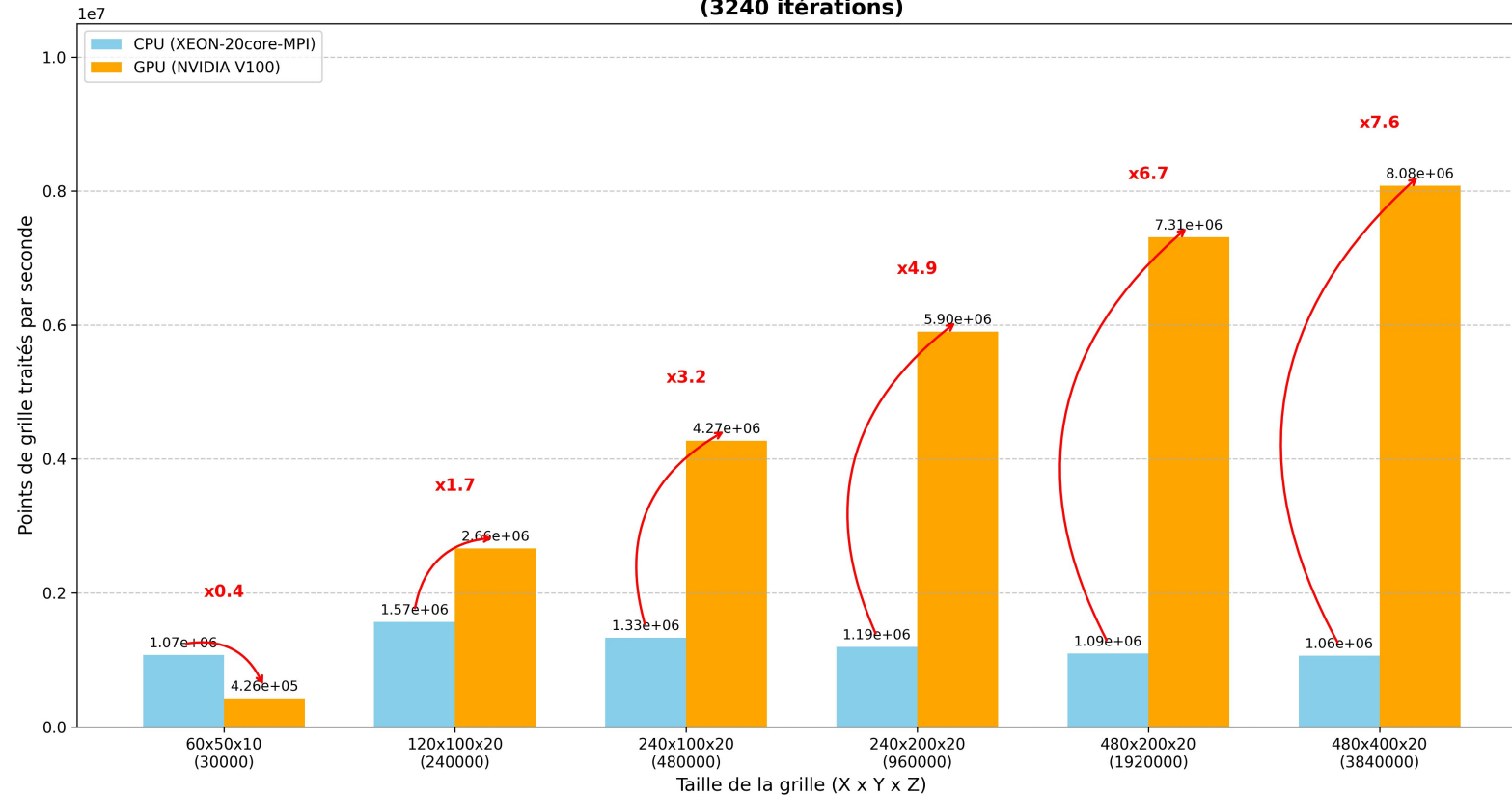
Performances HYDRO : XEON-20core-MPI vs NVIDIA V100  
Légende : Test cluster de calcul  
(3240 itérations)



# Performance Results

## Nuwa cluster (LAERO)

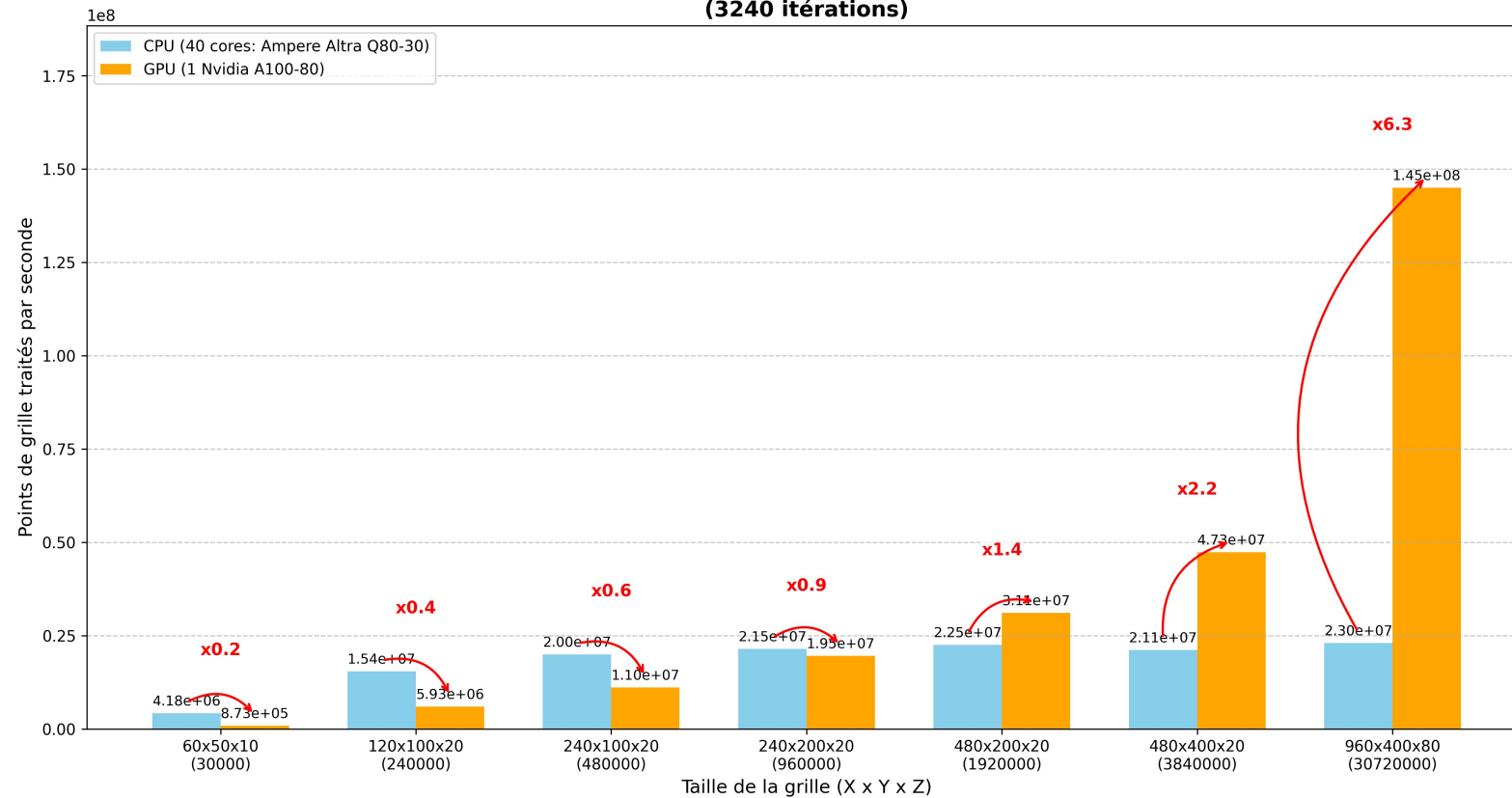
Performances NBQ : XEON-20core-MPI vs NVIDIA V100  
Légende : Test cluster de calcul  
(3240 itérations)



# Performance Results

## Turpan cluster (calmip)

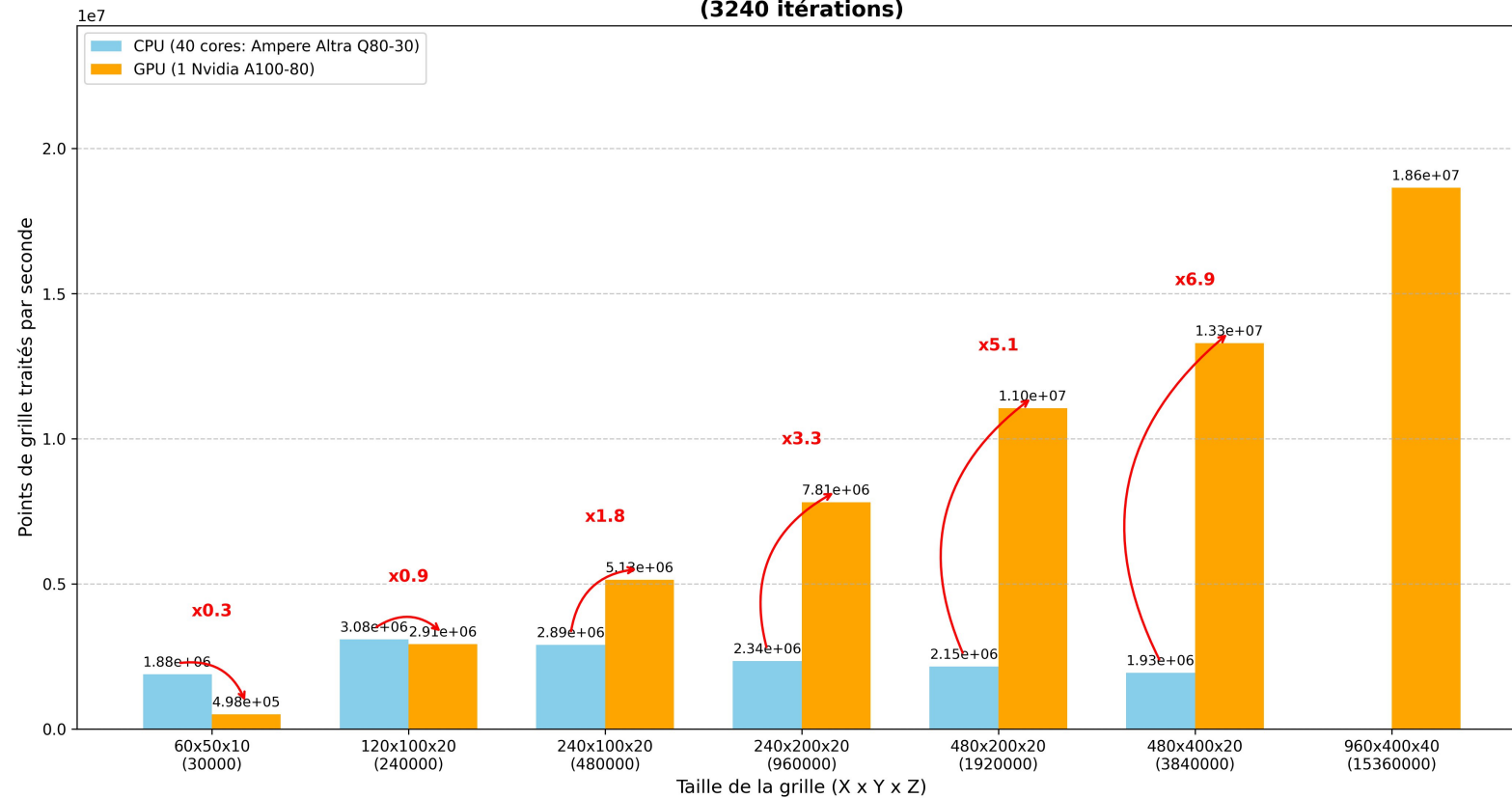
**Performances HYDRO : 40 cores: Ampere Altra Q80-30 vs 1 Nvidia A100-80**  
**Légende : Bench cluster Turpan**  
**(3240 itérations)**



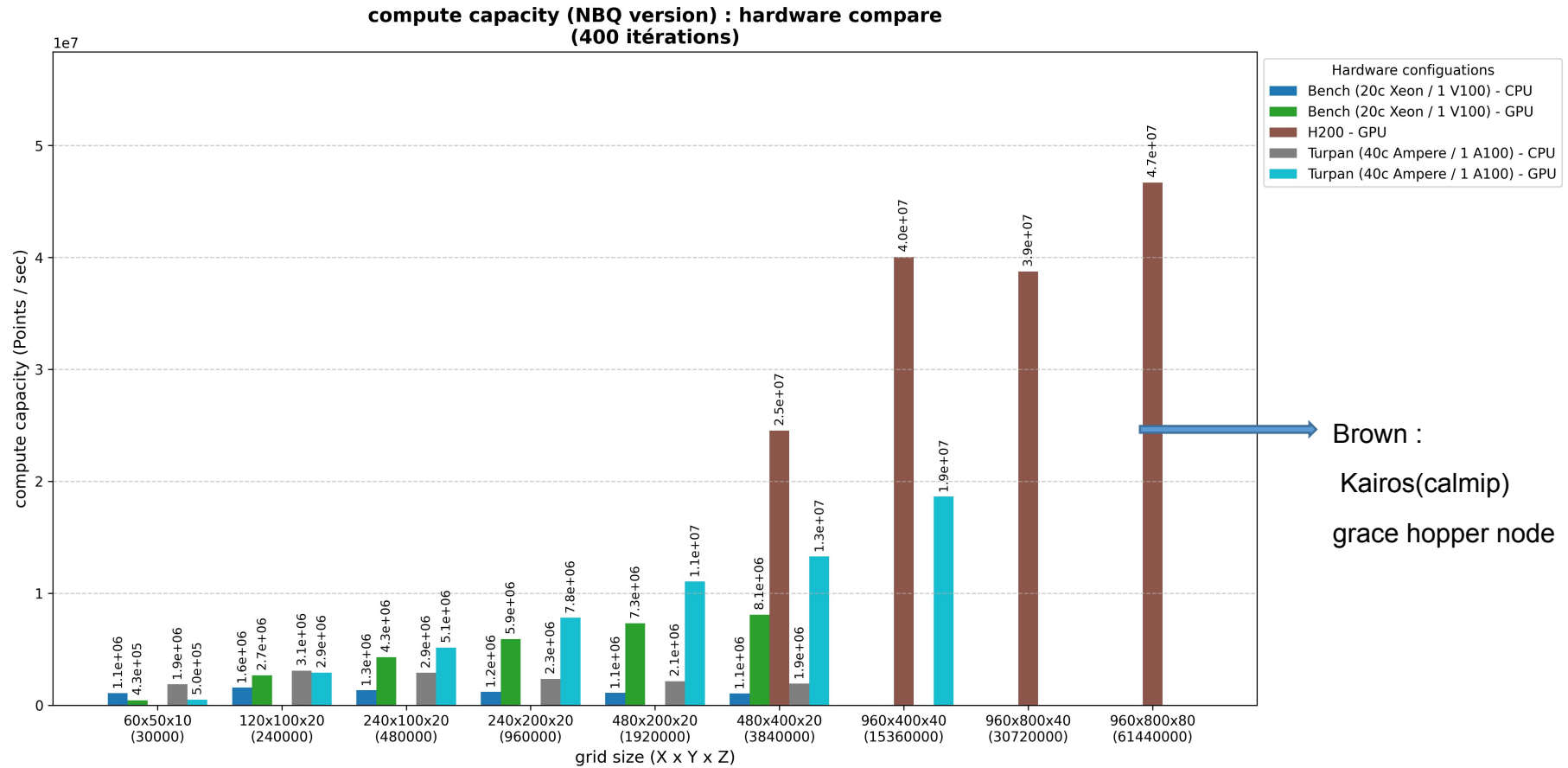
# Performance Results

## Turpan cluster (calmip)

Performances NBQ : 40 cores: Ampere Altra Q80-30 vs 1 Nvidia A100-80  
 Légende : Bench cluster Turpan  
 (3240 itérations)



# Hardware resume

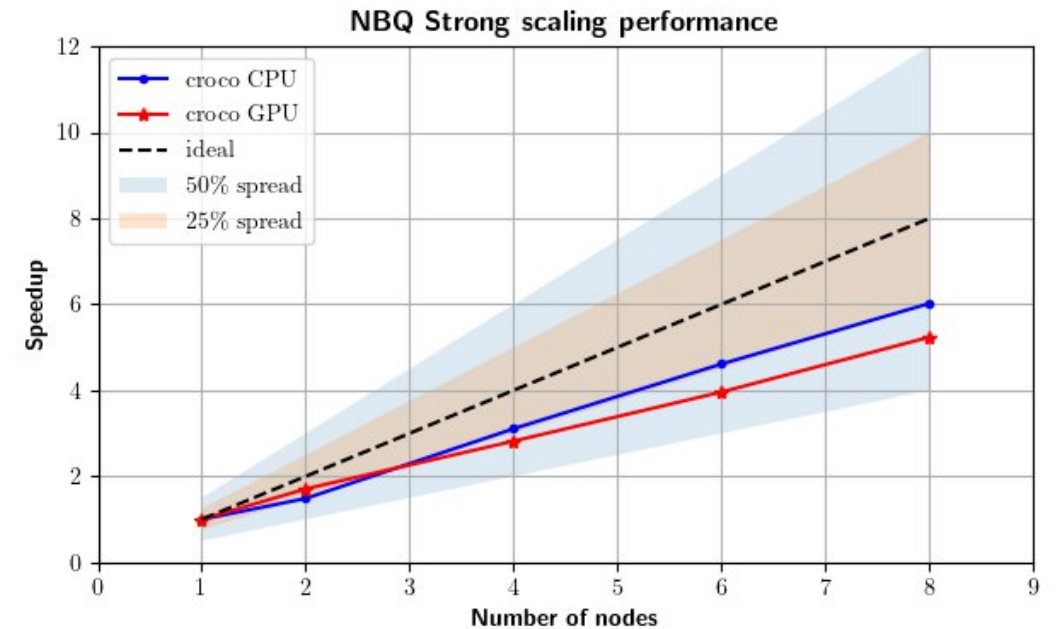
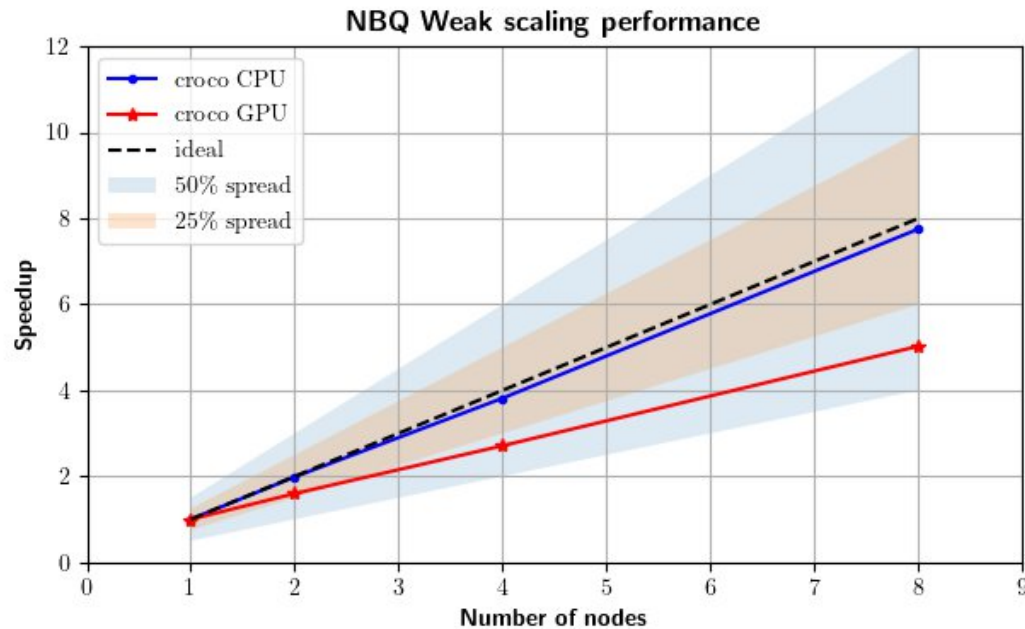


# scalability

Test On Jean-ZAY IDRIS :  
 1 node : 40 cores & 4 V100  
 on 8 nodes GPU ~ 2 times faster than CPU

2048x2048 horizontal grid and 50 vertical levels : 1 node  
 to  
 8192x4096 2048x2048 points and 50 vertical levels : 8 nodes

2048x2048 horizontal grid and 50 vertical levels







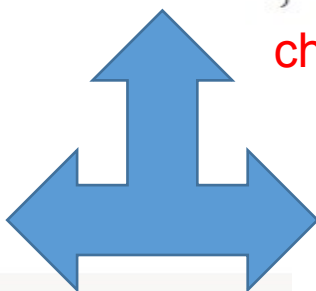
# development side

```

!
do j=JV_RANGE
  DOLOOP2D_R(IV_RANGE, JV_RANGE)
  do i=IV_RANGE
    DC(i,0)=0.
    FC(i,0)=0.
  enddo
  do k=1,N
    do i=IV_RANGE
      DC(i,k)=0.5*(Hz(i,j,k)+Hz(i,j-1,k))*om_v(i,j)
      DC(i,0)=DC(i,0)+DC(i,k)
      FC(i,0)=FC(i,0)+DC(i,k)*v(i,j,k,nrhs)
    enddo
  enddo
  do i=IV_RANGE
    FC(i,0)=(FC(i,0)-DV_avg2(i,j))/DC(i,0)
  enddo
  do k=1,N
    do i=IV_RANGE
      v(i,j,k,nrhs)=(v(i,j,k,nrhs)-FC(i,0))
      Hvom(i,j,k)=DC(i,k)*( v(i,j,k,nrhs)
# endif
&
      enddo
    enddo
  enddo
ENDDOLOOP2D
!

```

change\_loops.py



CPU

GPU

```

do j=Jstr,Jendr
do i=IstrR,Iendr
DC(i,0)=0.D0
FC(i,0)=0.D0
enddo
do k=1,N
do i=IstrR,Iendr
DC(i,k)=0.5D0*(Hz(i,j,k)+Hz(i,j-1,k))*om_v(i,j)
DC(i,0)=DC(i,0)+DC(i,k)
FC(i,0)=FC(i,0)+DC(i,k)*v(i,j,k,nrhs)
enddo
enddo
do i=IstrR,Iendr
FC(i,0)=(FC(i,0)-DV_avg2(i,j))/DC(i,0)
enddo
do k=1,N
do i=IstrR,Iendr
v(i,j,k,nrhs)=(v(i,j,k,nrhs)-FC(i,0))
Hvom(i,j,k)=DC(i,k)*( v(i,j,k,nrhs)
&
      enddo
    enddo
  enddo

```

```

DO j=Jstr,Jendr
!$acc loop private(DC1D,FC1D) vector
DO i=IstrR,Iendr
DC1D(0)=0.D0
FC1D(0)=0.D0
do k=1,N
DC1D(k)=0.5D0*(Hz(i,j,k)+Hz(i,j-1,k))*om_v(i,j)
DC1D(0)=DC1D(0)+DC1D(k)
FC1D(0)=FC1D(0)+DC1D(k)*v(i,j,k,nrhs)
enddo
FC1D(0)=(FC1D(0)-DV_avg2(i,j))/DC1D(0)
do k=1,N
v(i,j,k,nrhs)=(v(i,j,k,nrhs)-FC1D(0))
Hvom(i,j,k)=DC1D(k)*( v(i,j,k,nrhs)
&
      enddo
    ENDDO
  ENDDO

```

# Troubleshooting

```
deleted block device:0x4019c0000000 size:1002388992 threadid=1  
deleted block device:0x4019c0000000 size:1002388992 threadid=1  
Accelerator Fatal Error: call to cuLaunchKernel returned error 2 (CUDA_ERROR_OUT_OF_MEMORY): Out of memory  
File: /usr/local/cuda-11.8/bin/nvcc (TEST_7UAN/RunGPU/ana_grid_tile.f) line:309
```

too much memory required

```
allocated block device:0x14cefbbaf200 size:267264 thread:1  
allocated block device:0x14cefbc00000 size:801792 thread:1  
allocated block device:0x14cefbcc3c00 size:267264 thread:1  
allocated block device:0x14cefbdb05000 size:881664 thread:1  
FATAL ERROR: data in PRESENT clause was not found on device 1: name=tab(istr:istr,jstr:jstr) host:0x5d236e0  
file:/nhoms/nguc/Devel2026/Webinaire/BASIN/BASIN_NBQ_NBQ3_60_50_10_30000_ERROR/CompileGPU/ana_grid_tile.f ana_grid_tile line:309
```


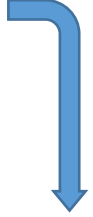
array is not on device

poor performance => look at **sequential** at compilation time

## practices

- Think Big
- Not all key combinations in `cppdefs.h` have been tested
- Numerics differ CPU vs GPU
- Cluster : accounting
- PC : GPU works, CPU have time

# future Horizon GPUs

- Maintenance : « handmade »
  - gitlab CI : include CVTK ACC test ...
- OpenACC ? (AMD GPU) 
  - Psyclone : fortran 2 fortran processing 
- No AGRIF (yet)
- What about CPU when GPUs run ?
- Scalability :
  - Expand mpi overlapping : branch martin\_lat\_hid\_mpi\_2d
- Questions ...