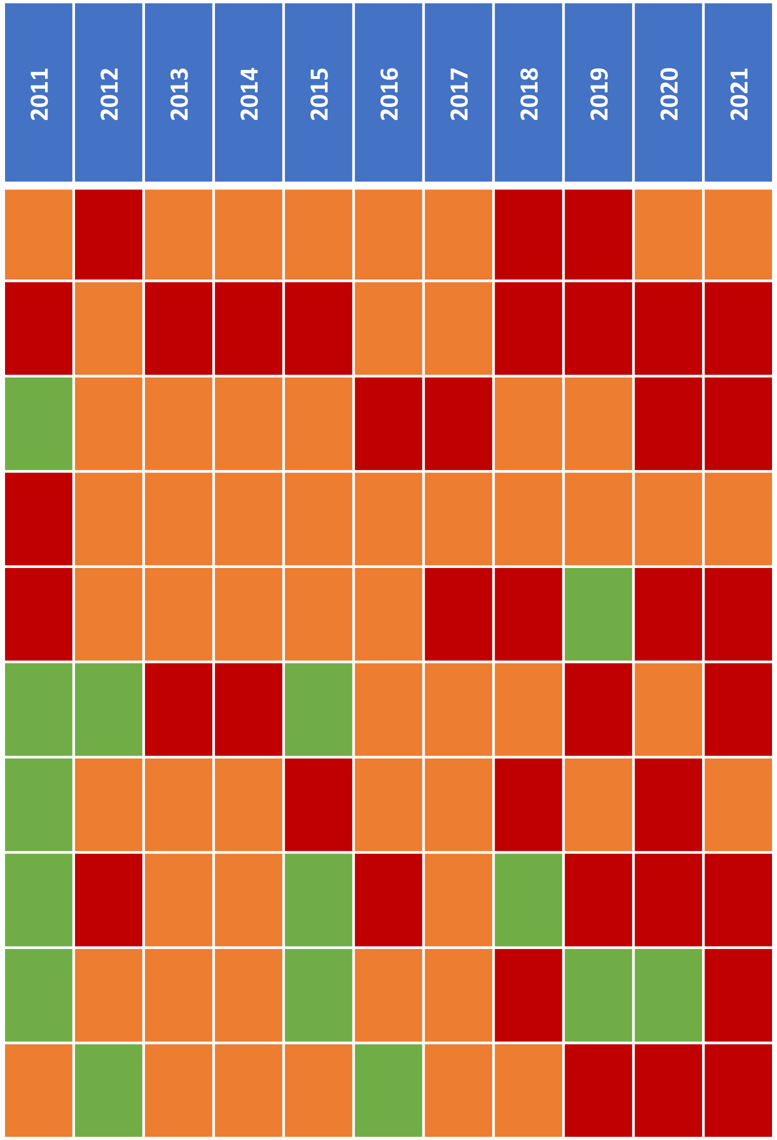
# L’enjeu des machines accélérées




## Top 500 list : the 10 fasten supercomputer

**Past decade** **Nov. 2021**

- **#1 (Nov 2021) : Fugaku (Japan) :**
**7,5 M cores ARMFX64, ~ 0,5 ExaFlops**

Color code :

- ~ Accelerator (GPU)
- ~ Many cores
- ~ General purpose processor

- **General tendency is on accelerator based system (GPU computing)**
- **Or Many Cores systems (1M+ cores)**
- **Convergence of HPC, AI & HPDA**

| Rank | System | Cores |
|---|---|---|
| 1 | **Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu<br>RIKEN Center for Computational Science<br>Japan | 7,630,848 |
| 2 | **Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM<br>DOE/SC/Oak Ridge National Laboratory<br>United States | 2,414,592 |
| 3 | **Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox<br>DOE/NNSA/LLNL<br>United States | 1,572,480 |
| 4 | **Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC<br>National Supercomputing Center in Wuxi<br>China | 10,649,600 |
| 5 | **Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE<br>DOE/SC/LBNL/NERSC<br>United States | 761,856 |
| 6 | **Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia<br>NVIDIA Corporation<br>United States | 555,520 |
| 7 | **Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT<br>National Super Computer Center in Guangzhou<br>China | 4,981,760 |
| 8 | **JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, Atos<br>Forschungszentrum Juelich (FZJ)<br>Germany | 449,280 |
| 9 | **HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, DELL EMC<br>Eni S.p.A.<br>Italy | 669,760 |
| 10 | **Voyager-EUS2** - ND96amsr_A100_v4, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100 80GB, Mellanox HDR Infiniband, Microsoft Azure<br>Azure East US 2<br>United States | 253,440 |

# French supercomputers

## France (GENCI & meteo-france) is following the general tendency

- IDRIS : Jean-Zay : ~ 2696 GPU Nvidia V100 ( 5/6 of JZ peak performance), 28PF
- CINES : Adastra (2022) : AMD manycore (Genoa) + GPU (MI200), 70PF
- CEA/TGCC : AMD Rome, 128 cores by nodes, 300 000 cores, similar to many core system
- CEA/TGCC : 256 GPU NVIDIA V100
- CEA/TGCC : 80 Nodes Arm Fujitsu A64FX (48 cores by socket, FUGAKU-like)
- Meteo-France : Belenos/Taranis : AMD EPYC ~300 000 cores

**But, for now, keep a large part of general purpose processors…**

## Next Step : French Exascale computer –> 2024 + ?

- EuroHPC call for exascale supercomputer (50% funded by Europe) : ~2023-2024
  - 2 hosting site candidates : TGCC (France) & FJZ (Germany)
- Should use European technology
  - ARM Zeus core + Titan Risc-V accelerator + BXI interconnect
- Due to delay on the roadmap and to fulfil the expected targets (performance , consumption) => would be probably mostly composed of GPU accelerators : Nvidia, AMD or Intel
- Announced part of general purpose processors : 20 – 40 % : Price or peak performance ?
  - 20 % of price => ~ 5% of peak performance
- "Exascale France " project : prepare the applications to exascale

**=> Not a clear idea of what would be the hardware architecture and technology for the future Exascale French machine**

# CALCULATEURS PRE-EXASCALE

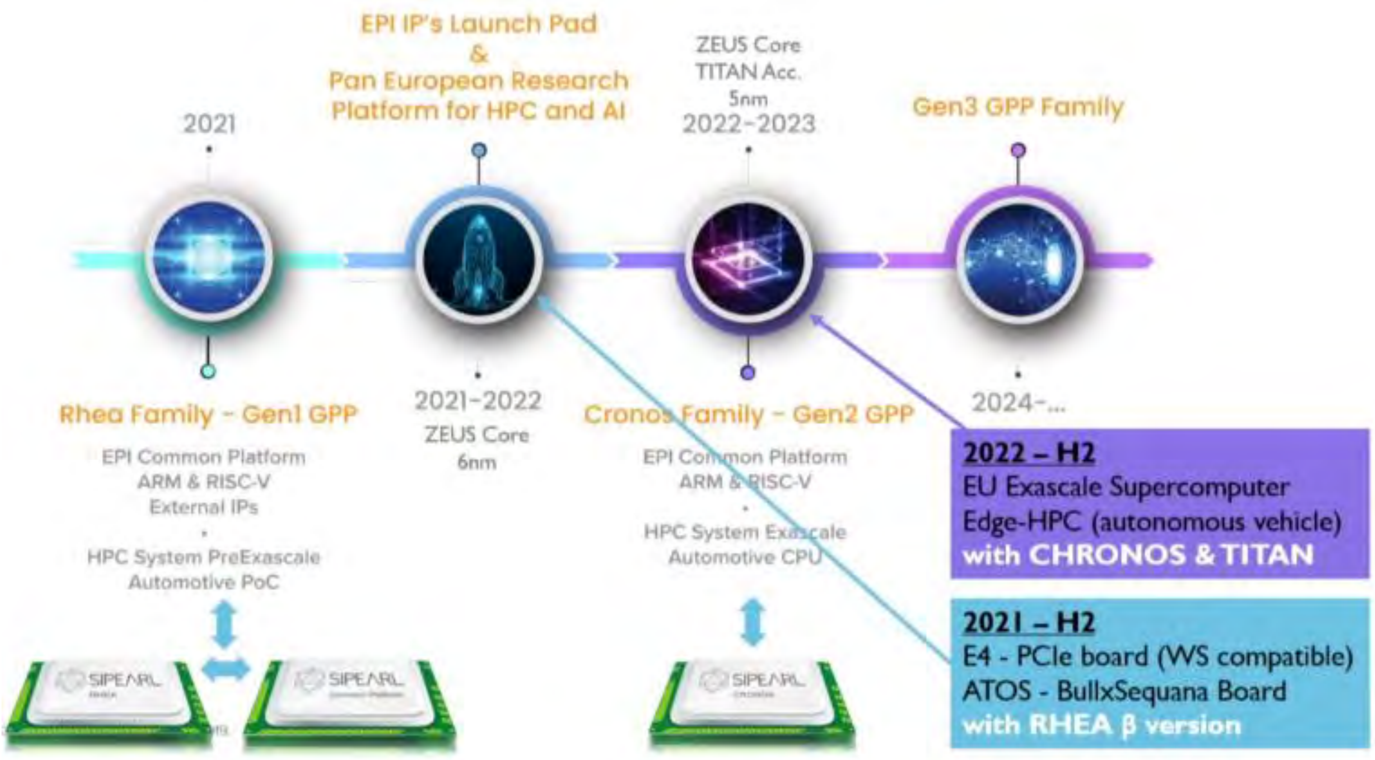
## Statuts

EuroHPC

| | LUMI | LEONARDO | MARE NOSTRUM 5 |
|---|---|---|---|
| Leader<br>Consortium | CSC – Finlande<br>10 partenaires (SE, DK, NO, CH, BE, PL, EE, CZ, IS) | CINECA- Italie<br>5 partenaires (AT, GR, HU, SI, SK) | BSC – Espagne<br>3 partenaires (PT, TK, HR)<br>~200 Pflop/s (peak) |
| Caractéristiques | HPE<br>550 Pflop/s (peak)<br>CPU : AMD EPYC Trento<br>GPU : AMD Instinct MI200<br>4 X 100 Gb/s | Atos<br>250 Pflop/s (peak)<br>CPU : Intel ICL<br>GPU : nVIDIA A100<br>2 x 100 Gb/s | Annulation de l'appel d'offres<br>**Reloaded ??** |
| Mise en production | Lumi C : sept. 2021<br>Lumi G : mars/avril 2022 | Fin travaux site : déc 2021<br>Mise en production : Q3 2022 | |

# EPI for HPC project : European Processor Initiative

**Goal : reach the European technological independence on computing processors**

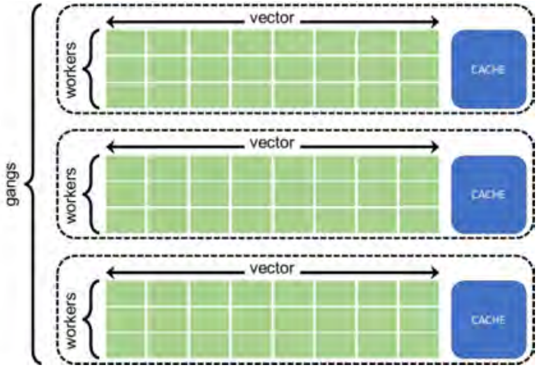


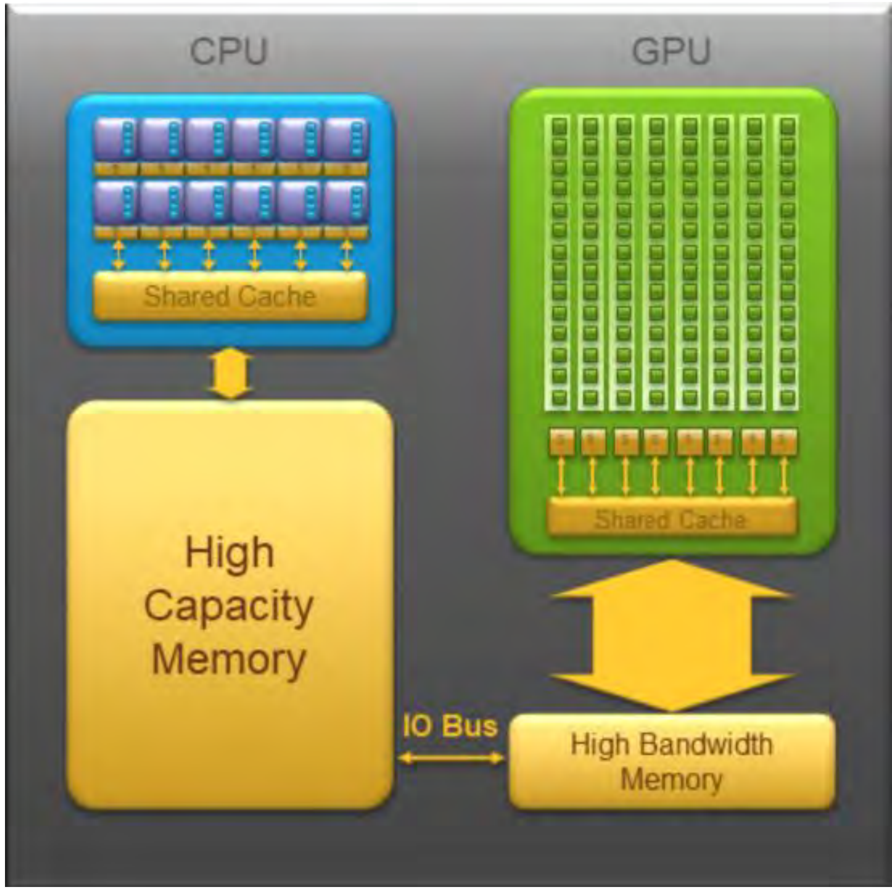

- **Common platform based on ARM64 (general purpose) and Risc-V for accelerator + BXI interconnect (ATOS)**
- **Growing interest for ARM of many country to achieve strategic path to exascale**
  - Japan (RIKEN), India (MEYTI-CDAC), South Korea (ETRI K-AB21), Europe…

**Accelerator hardware strategy seems not to be so well defined, and will not be ready for first exascale European computer**

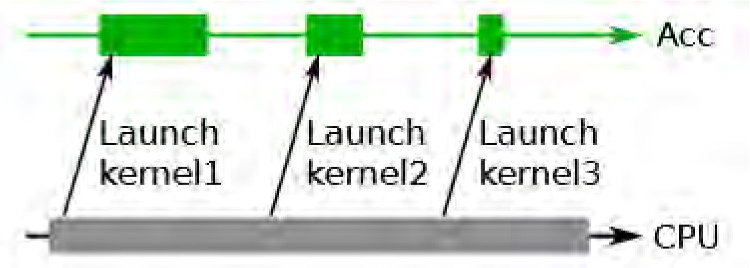
# GPU : how does it works ?

- Thousands of small core (3456 for Nvidia A100)
  - 9,7 TFs, 2TB/s memory bandwidth ( HBM2)
- 3 internal levels of parallelism
  - Coarse grain (gang / teams)
  - Fine grain (worker)
  - Vectorisation (vector)




- #Threads = #gang * #worker *# vector ==> O(10000 threads) to ensure performance
- SIMT paradigm : Single Instruction Multiple Threads
- Performance comes from quick switching between threads context to overlap memory access latency
- Each computing kernels (loops) are offloaded from the host (CPU) to the device (GPU)




- Computation on GPU is “free lunch”, performance bottlenecks come mainly from memory access and data transfer between host and GPU

# How is it programming ?

- **By the past, specific low level language (CUDA) or API (OpenCL)**
  - Very intrusive for models, difficult to port large part codes
- **Now recommendation is to use high level approach based on directives ; 2 main standards :**
  - OpenACC : most complete standard driven mainly by NVIDIA
  - OpenMP 4.5 / 5 : specific directives have been added to the standard for accelerator, late on OpenACC, but would more largely supported by other GPU vendors (AMD, Intel)
- **Other high level approach**
  - Language : kokkos, Raja… => not standard, need to rewrite
  - Domain Specific Language : STELLA, PSYCLON, home made…
- **Each loops (kernels) must be instrumented to be offloaded on the GPU**
  - Directives to manage loop parallelism (!$ACC parallel loop, $ACC kernels)
  - Directives to manage data transfer and allocation on GPU ( data clause)

```
!$ACC kernels
do i=1, n
   do j=1, n
   ...
   enddo

   do j=1, n
   ...
   enddo
enddo
!$ACC end kernels
```

```
!$ACC parallel copyout(a(:1000))
!$ACC loop
do i=1, 1000
 a(i) = i
enddo
!$ACC end parallel

!$ACC data copy(a(:1000))
do l=1,100
     !$ACC parallel
     !$ACC loop
     do i=1, 1000
      a(i) = a(i) + 1
     enddo
     !$ACC end parallel
enddo
!$ACC end data
```

```
!$acc parallel
!$acc loop worker vector
do i=1,nx
    A(i)=1.0_8
end do
!$acc loop worker vector reduction(+:somme)
do i=nx,1,-1
    somme=somme+A(i)
end do
!$acc end parallel
```

# DYNAMICO CPU Vs GPU

## "Contrat de progrès"

IDRIS+GENCI+HPE+LSCE

3 peoples for 3 months

OpenACC

## Only atmospheric dynamic

- No I/O

## Low resolution

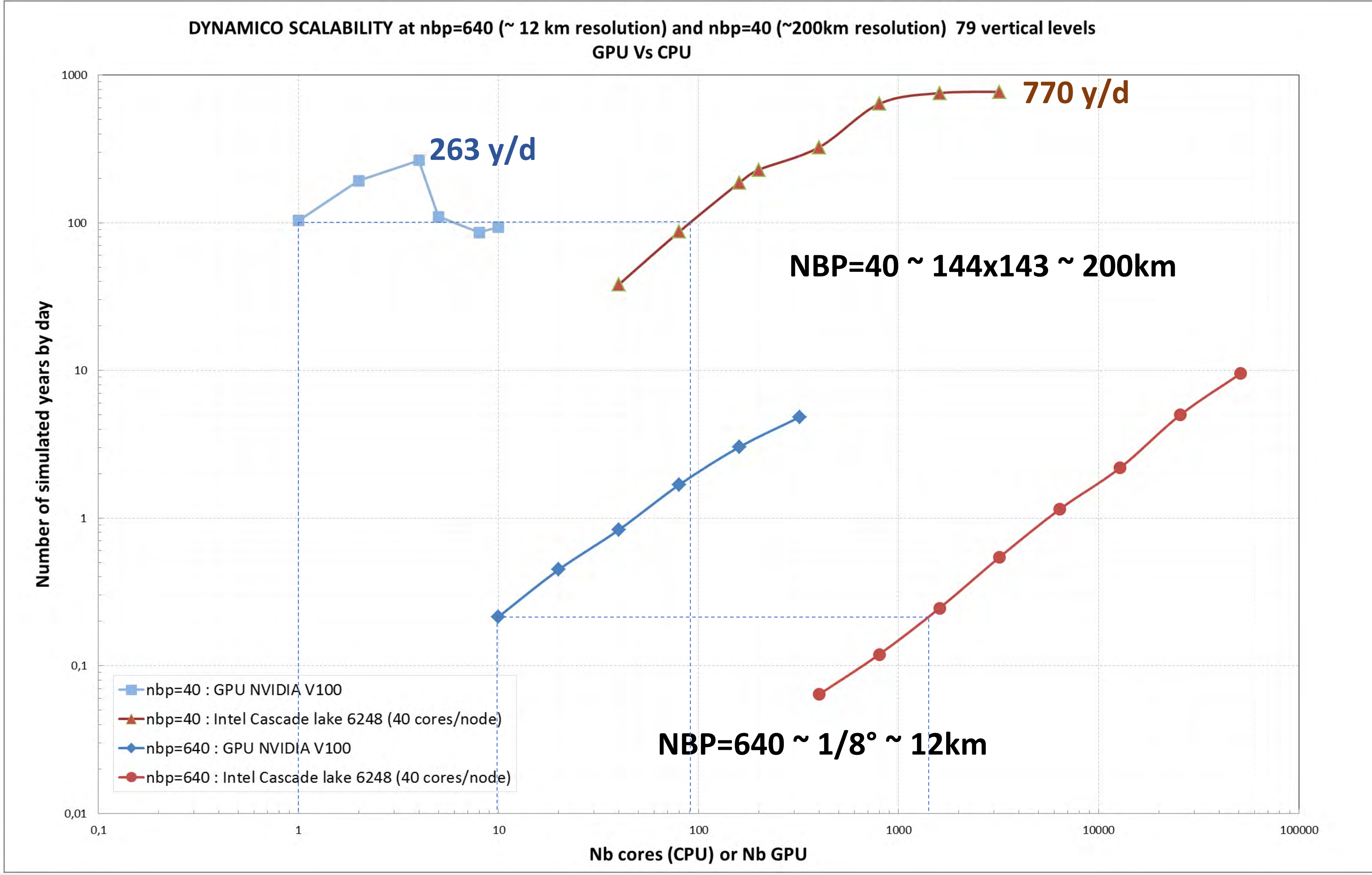
1 GPU (V100) ~= 4.5 Intel CC

~= 90 cores

## High resolution

1 GPU (V100) ~= 6 Intel CC

~= 120 cores



# IDRIS Hackathon on simple atmospheric physics

## Port of simple atmospheric physics with 21 parameters on GPU

- Idris Hackathon May 2021 (T. Dubos &al.)
- OpenACC
- ~2000 code lines, dynamics with DYNAMICO
- ~1 week work

| llm | ngrid (per GPU) | nbp | dt (dyn) | itau_phys | run_length | steps (phys) | CPU total | GPU total | speedup |
|---|---|---|---|---|---|---|---|---|---|
| 79 | 4000 | 40 | 480 | 5 | 864000 | 360 | 5,633 | 0,971 | 5,80 |
| 79 | 16000 | 80 | 240 | 5 | 432000 | 360 | 24,841 | 2,841 | 8,74 |
| 79 | 64000 | 160 | 120 | 5 | 60000 | 100 | 31,483 | 2,652 | 11,87 |
| 79 | 256000 | 320 | 60 | 5 | 30000 | 100 | 170,93 | 10,054 | 17,00 |

## Speedup : 1 GPU nodes Vs 1 CPU Nodes == 4 gpu V100 Vs 2 intel Cascace Lake

- Low resolution (200km) : 1 GPU ~ 58 cores CCL
- High Resolution (25km) : 1 GPU ~ 170 core CCL

**Greatest benefits for high resolution run**

**Low resolution runs exhibit interesting benefit only inside 1 GPU node**

**Potentially and probably no decreasing of elapsed time compared to massively parallel CPU computing**

# What to expect for a full ESM : “le diable est dans les détails”

**Full ESM : 500 000 ~ 600 000 code lines**

- **Each component ~ 50 000 Loc -> 250 000 Loc**
- **Not every thing would be accelerate**
  - Communications, I/O, Ill-formed code part, lake of man-power, late or new component
- **What would not be accelerated will be run by the host but on less CPU resources**
  - Generally 1 MPI process drive 1 GPU
  - Difficult to manage both openMP and GPU kernel within OpenACC (maybe better using only openMP5 only)
  - So 4 cores will be used to drive 4 GPU over the 40 cpu cores available (CC, IDRIS)
  - R, ratio between GPU and equivalent cpu core to reach the same SYD (or elapsed time) for the accelerated part : ex: Dynamico R ~ 100 cores/1GPU
  - So the part that would not be accelerated will run R times slowly
  - In this specific case, more than 99% of the computing work must be running on the GPU to shows some benefit

**What is important is not what had been ported onto GPU, but what has not been ported onto GPU**

  - More easy for model with small part of code that consume 99.9% of computing time
  - But climate models have a relatively flat profile…
- **Solution will consist to share computing work both on CPU and GPU**
  - Default behaviour is CPU are idle when GPUs compute kernels
  - openMP + GPU
  - Distribute work across idle CPU cores and GPU

**But it will consideratelly increase the degrees of complexity of the model**

**!! Importance of the code redesign !!**

# Conclusion

## The supercomputing landscape is changing in depth, introducing new paradigms…

- Convergence of ML, AI <-> HPC
- Massive integration of GPU-based accelerators (more than 90% of the peak perf.)
- No other more friendly alternative at short term.

**GPU computing is working very well for many scientific thematic and for IA**

**But future accelerator or many cores systems will probably not improved significantly time to solution for climate modeling (IPSL) but will make possible large ensembles and/or high resolution runs.**

**Major difficulties come from the volume of models, their heterogeneity and complexity.**

**Major code redesign & rewriting will be needed to achieve good performance on GPU.**

- How to make an efficient port without falling to "Ninja Programming" ?
- How to maintain porting for non-expert researcher without freezing new developments ?

**Emulate physical processes with AI can be a way to port efficiently part of models onto GPU**

- This way merit to be explored (TRACCS)
- But not (yet) proved to be working efficiently for climate modeling