NVIDIA Grace Superchip Early Evaluation for HPC Applications

IWAHPCE24, Nagoya

Fabio Banchelli * (BSC) Joan Vinyals-Ylla-Catala (BSC) Josep Pocurull (BSC)

Marc Clascà (BSC) Kilian Peiro (BSC) Filippo Spiga (NVIDIA)

Marta Garcia-Gasulla (BSC) Filippo Mantovani (BSC)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MareNostrum4

General purpose block

Emerging Technology Clusters (CTEs)

Intel Xeon Platinum

CTE Power - IBM POWER9 + NVIDIA Volta GPUs

CTE AMD - AMD Rome + AMD Radeon Instinct MI50

CTE Arm - A64FX processors

Previous experience with Arm-based clusters

- Mont-Blanc EU Project
  - Odroid-XU
  - Nvidia Jetson-TX

  - Cavium ThunderX
  - Marvell ThunderX2

- BSC-Huawei collaboration
  - Kunpeng

- ISC23 Student cluster competition
  - Ampere Altra MAX

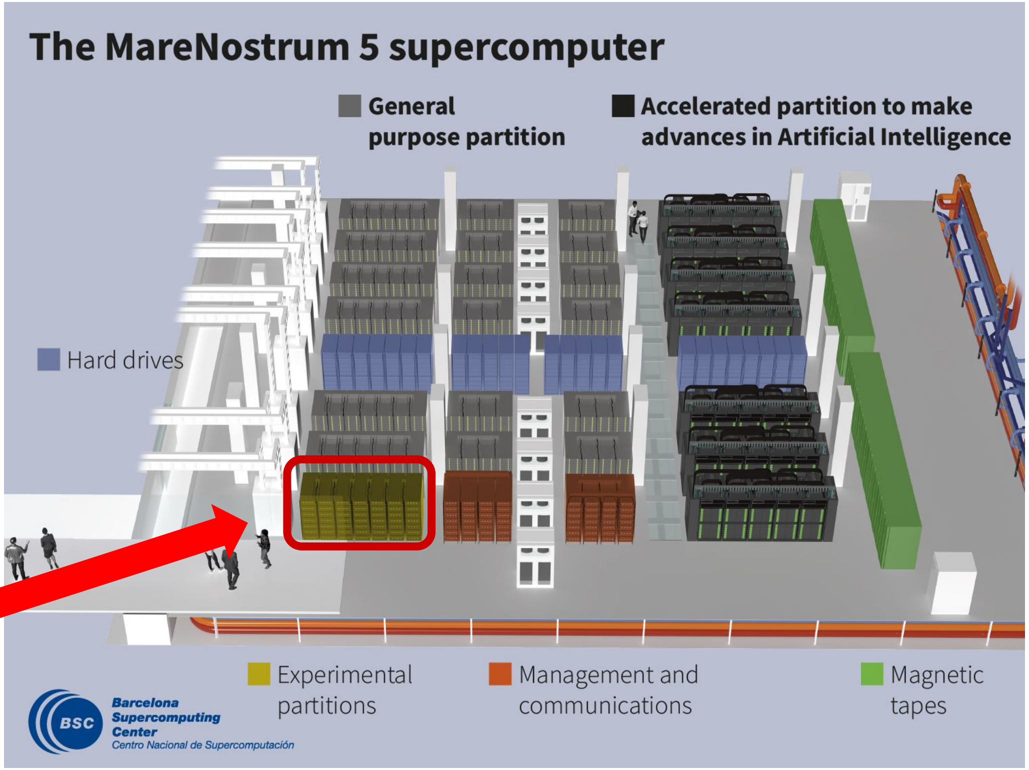
MareNostrum5

Goal

- ~~First in the Top500~~
- Run/Solve diverse scientific problems

Structure

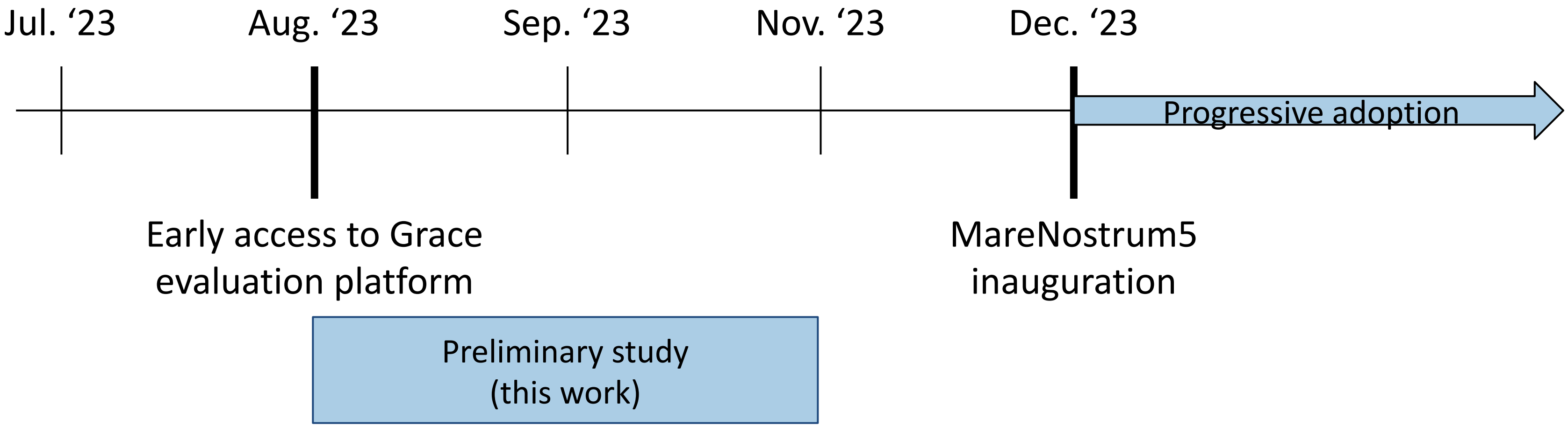
- General purpose partition
  - Intel CPUs
- Accelerated partition
  - Intel CPUs + Nvidia GPUs
- Experimental "Next generation"
  - Nvidia Grace CPUs



MareNostrum4 $\longrightarrow$ MareNostrum5

- How difficult is it for a scientist to transition to a Grace-based system?
- How does the NVIDIA Grace CPU behave with complex scientific codes?
- How does the NVIDIA Grace CPU compare to other HPC systems?

MareNostrum4 ⟶ MareNostrum5



Systems under study

Early evaluation cluster

- Provided by Nvidia
- Engineering samples (not final product)

- Two hardware configurations
  - Grace-Grace
    - Three nodes based on the Nvidia Grace CPU Superchip
  - Grace-Hopper
    - Two nodes based on the Nvidia Grace-Hopper GPU Superchip

- Network
  - Infiniband NDR400

⚠️ **Engineering samples:** functional parts but only partially reflect the exact final product that will be available to the mass market.

Early evaluation cluster - Software

- MPI Library
  - OpenMPI/4.1.5rc2 + HPCX
- Compilers
  - GNU Compiler/12.3.0
  - Nvidia HPC Compiler/23.9
  - Arm HPC Compiler/23.04.1

- Shared filesystem
  - Home → NFS
  - Scratch → DDN Lustre
- Job scheduler
  - SLURM

Early evaluation cluster - Limitations

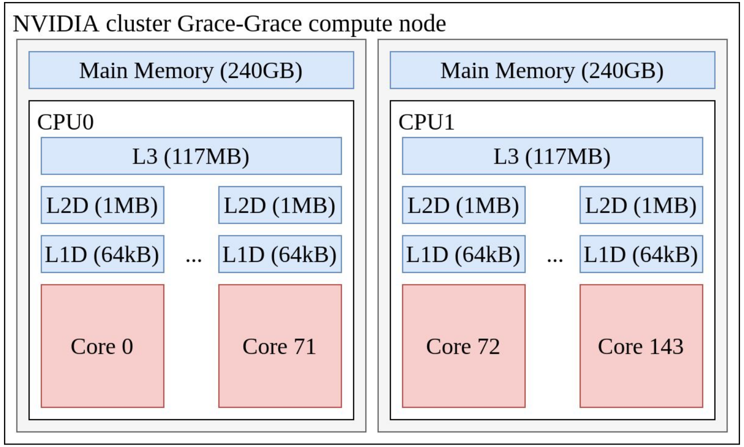
⚠️ **Engineering samples:** functional parts but only partially reflect the exact final product that will be available to the mass market.

- No micro-benchmarking
  - Floating-point performance
  - Cache hierarchy
  - Memory bandwidth

- No access to hardware counters
  - Perf
  - PAPI
  - BSC performance analysis tools

Nvidia Grace CPU Superchip

• Architecture:	Armv9
• Micro-architecture:	Neoverse V2
• Frequency:	> 3.20 GHz
• Number of sockets:	1
• CPUs per socket:	2
• Cores per CPU:	72
• L3 cache per CPU:	117 MB
• Memory channels per CPU:	8
• Memory per node:	480 GB
• Memory technology:	LPDDR5



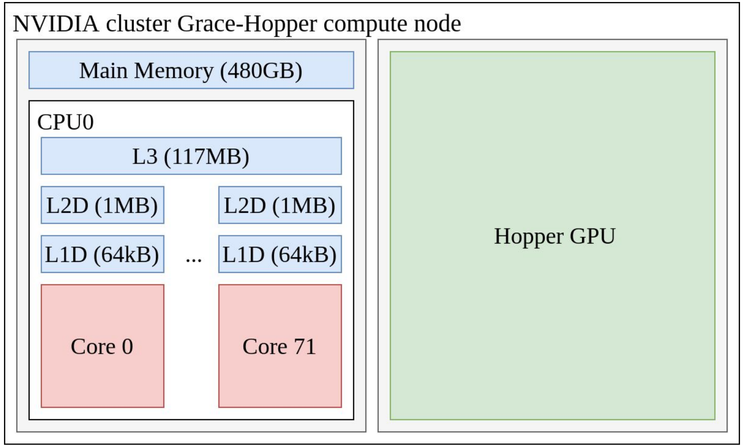
Nvidia Grace GPU Superchip

- Architecture: Armv9
- Micro-architecture: Neoverse V2
- Frequency: > 3.20 GHz

- Number of sockets: 1
- CPUs per socket: 1
- Cores per CPU: 72

- L3 cache per CPU: 117 MB
- Memory channels per CPU: 8

- Memory per node: 480 GB
- Memory technology: LPDDR5



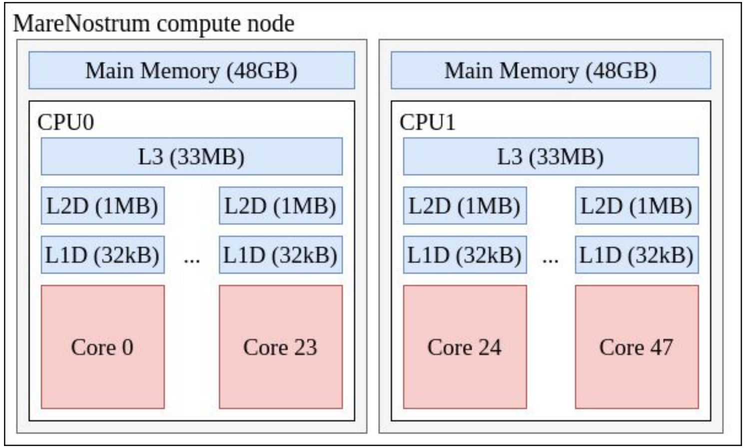
MareNostrum4 General Purpose partition

Architecture:	x86
Micro-architecture:	Intel Skylake
Frequency:	2.10 GHz (locked)

Number of sockets:	2
CPUs per socket:	1
Cores per CPU:	24

L3 cache per CPU:	66 MB
Memory channels per CPU:	6

Memory per node:	96 GB
Memory technology:	DDR4



Apps

Alya

- Computational mechanics
- Developed by BSC
- Written in Fortran
- Parallelized with MPI

- Input case
  - Finite element problem (16M elems.)
  - 1000 timesteps divided into phases
    - Matrix-Assembly
    - Boundary-Assembly
    - Solver

OpenFOAM

- Computational Fluid Dynamics
- Developed by OpenCFD Ltd.
- Written in C++
- Parallelized with MPI

- Input case
  - motorBike benchmark (5.2 million cells)
  - 10 timesteps

NEMO

- Forecasting in ocean and climate services
- Developed by EU consortium
- Written in Fortran
- Parallelized with MPI

- Input case
  - ORCA-1 configuration
  - 700 iterations

LAMMPS

- Molecular dynamics
- Developed by Sandia National Labs and Temple University
- Written in C++
- Parallelized with MPI

- Input case
  - 3d Lennard-Jones melt
  - Grid of 256 million atoms

PhysiCell

- Multi-cellular simulation
- Developed by BSC and Institut Curie
- Written in C++
- Parallelized with OpenMP

- Input case
  - 1 million cells
  - Evenly distributed across rectangular box

Compiler compatibility

- Minor changes to code to be able to compile
  - Alya → Compiler bug with Nvidia compiler
  - NEMO → Use of non-standard intrinsic function
  - Physicell → Initialization of static member variables

Coding techniques that do not respect the standard

- We were unable to run NEMO compiled with the GNU compiler and the Arm compiler

	Alya	OpenFOAM	NEMO	LAMMPS	PhysiCell
MareNostrum					
GNU Compiler	✓	✓	✓	✓	✓
Intel Compiler	✓	✓	✓	✓	✓
NVIDIA Cluster					
GNU Compiler	✓	✓	×	✓	✓
NVIDIA Compiler	✓	✓	✓	✓	✓
Arm Compiler	✓	✓	×	✓	✓

Evaluation

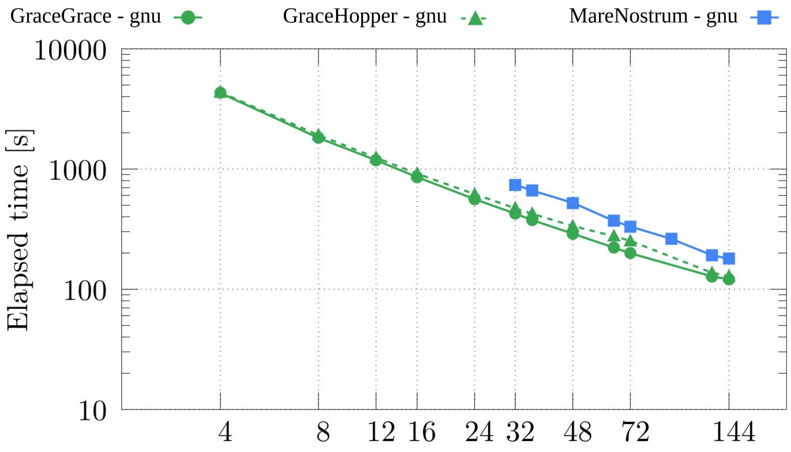
Alya

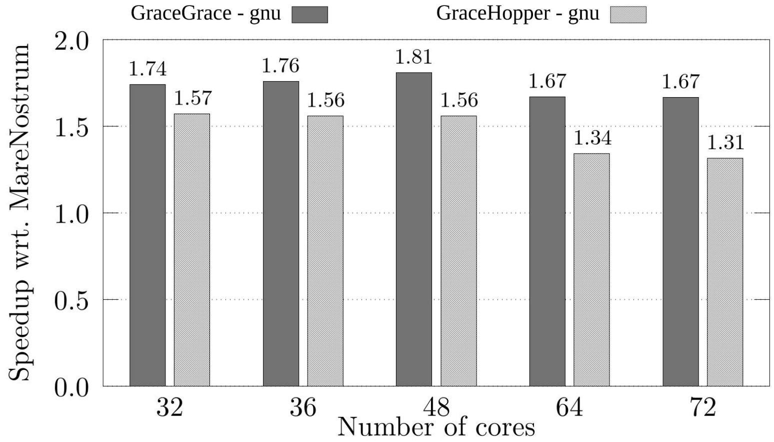
- Alya execution is mostly dominated by the Matrix-Assembly phase (compute bound)

- Grace-Hopper up to 64 cores
  - Speedup 1.57x
  - Frequency 1.60x wrt. MareNostrum

- Grace-Hopper 64 cores and beyond
  - Slight drop in scalability
  - Possibly due to memory bandwidth saturation

- Grace-Grace
  - Similar behavior as Grace-Hopper
  - Higher available bandwidth → Higher speedup



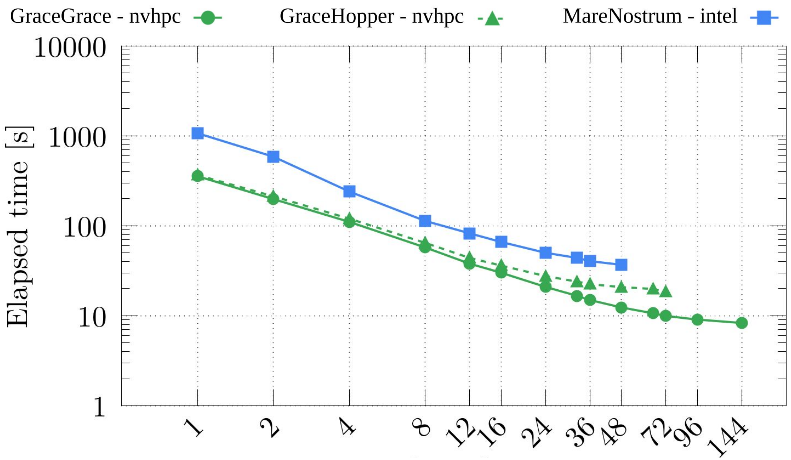


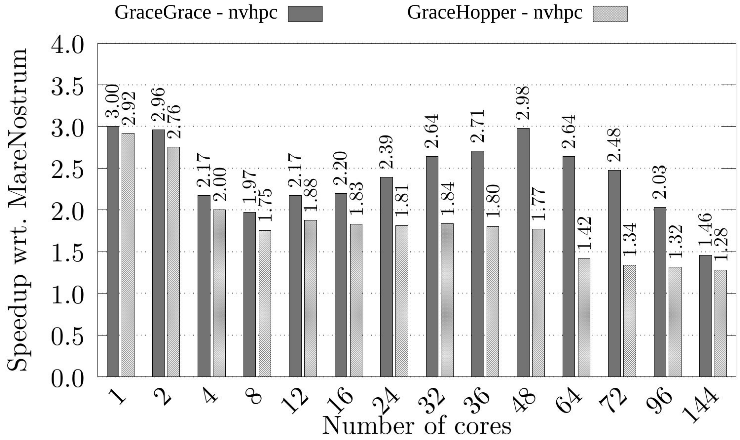
OpenFOAM

- Grace-Hopper and Grace-Grace cores always outperform MareNostrum

- Scalability curve flattens in all three clusters
  - MareNostrum first (peak 256 GB/s)
  - Grace-Hopper second (peak <= 500 GB/s)
  - Grace-Grace third (peak <= 1 TB/s)

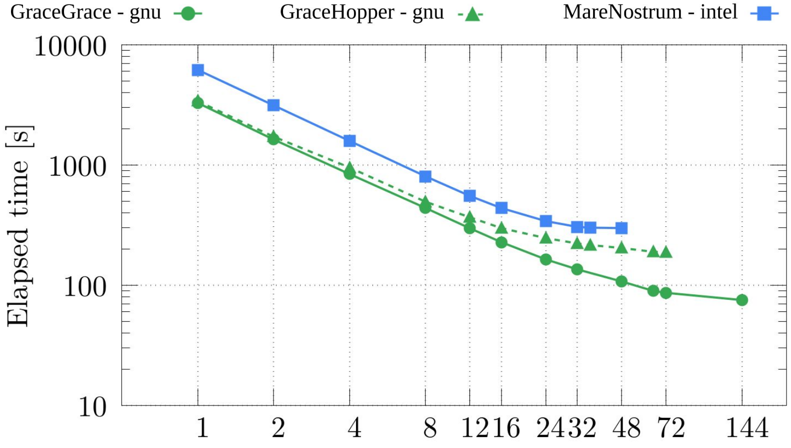
- Grace-Grace
  - Diminishing returns with 64 cores and over

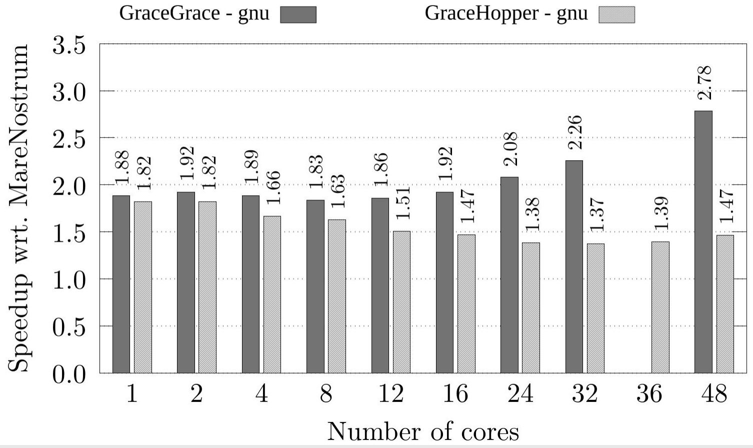




NEMO

- Similar behavior as with OpenFOAM
- Saturation point requires more cores → NEMO benefits from the higher memory bandwidth



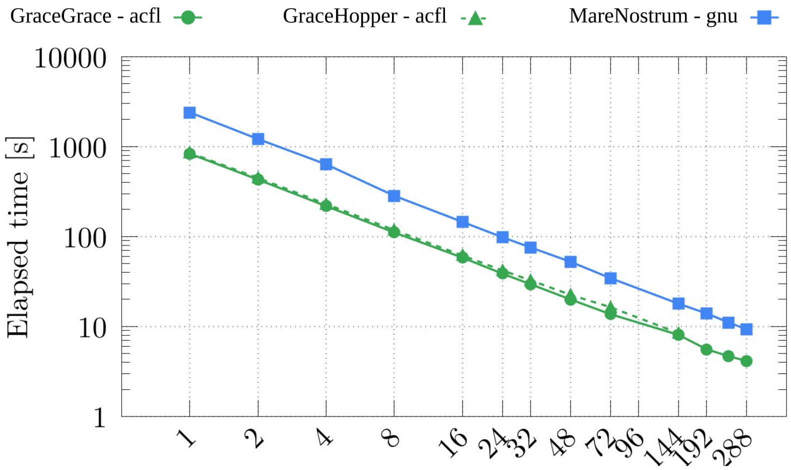


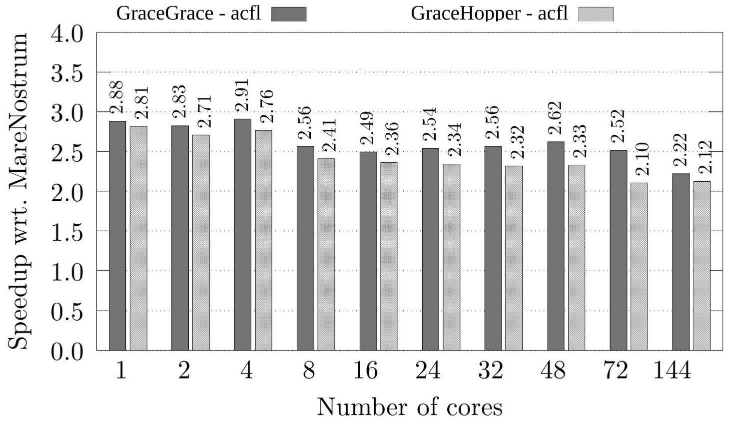
LAMMPS

- Little core-to-core performance difference between GraceGrace and GraceHopper
  - Having more memory channels does not seem to improve performance in LAMMPS

- Perfect strong scaling

- Always above 2.10x speedup wrt. MarenNostrum4

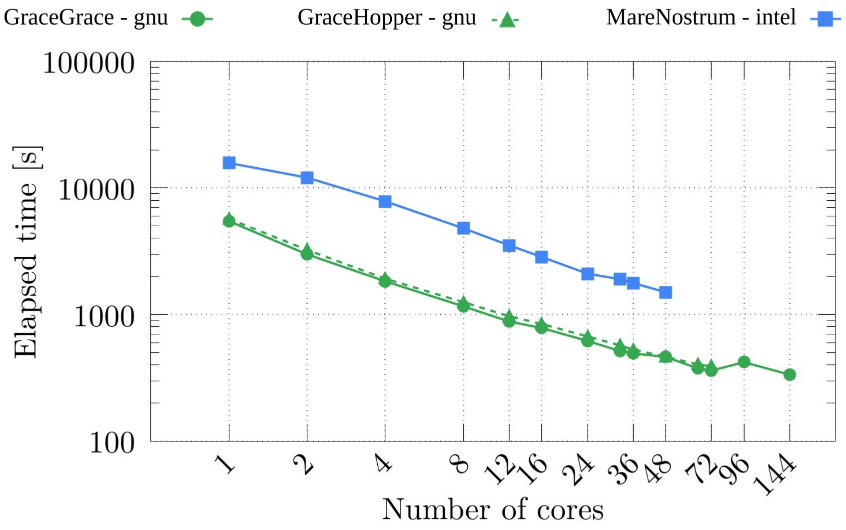




PhysiCell

- Overall bad scalability due to the application
  - Some regions of the code are not fully parallelized

- Grace-Hopper and Grace-Grace achieve the same performance
  - No benefit from higher memory bandwidth?
  - Maybe changing the thread mappings can improve performance

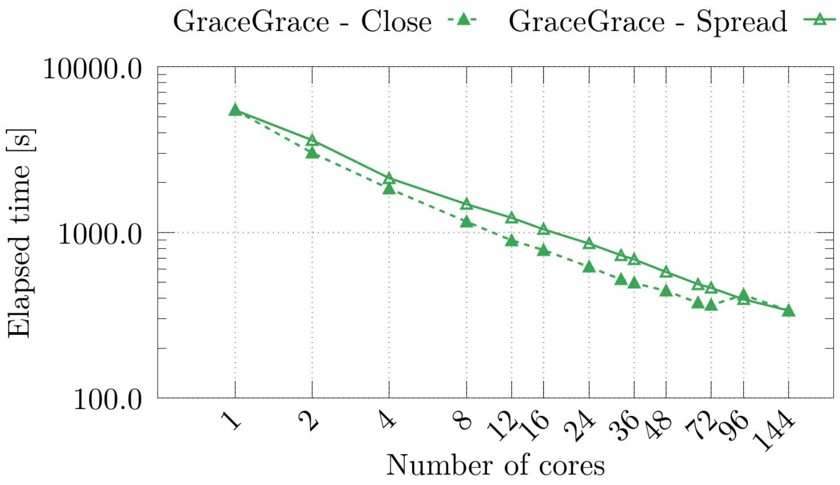


PhysiCell

- Close binding
  - Threads are mapped to contiguous cores
  - Thread count not balanced

- Spread binding
  - Threads are mapped to cores in alternating CPUs
  - Thread count is balanced

- Results show that close binding is better in PhysiCell
  - Memory allocation is not NUMA-aware
  - All threads access the same physical location
  - Threads from another CPU have higher latency and lowe bandwidth



Conclusions

How difficult is it for a scientist to transition to a Grace-based system?

✅ Compilers and libraries are well-prepared to support complex HPC codes

- Minor code changes (mostly due to non-standard coding techniques)
- Issue with binaries for NEMO

How does the NVIDIA Grace CPU behave with complex scientific codes?

✅ Scientific applications run out-of-the-box

- High core-count opens the door to more OpenMP parallelism

🏗️ Currently lacking tools to explore further

- Micro-benchmarking to measure performance peak
- Access to performance counters will give more insight

How does the NVIDIA Grace CPU compare to other HPC systems?

✅ All applications exhibited performance improvements wrt. MareNostrum

- Performance enhancement particularly pronounced in memory bound codes
- A portion of the improvement due to higher CPU frequency

✅ Core-to-core performance improvement out-of-the-box

End

Backup - Summary of hardware

	MareNostrum	NVIDIA Cluster	
		GraceHopper	GraceGrace
Cluster architecture			
Architecture	x86_64	Armv9	Armv9
Micro-architecture	Sykylake-X	Neoverse V2	Neoverse V2
Cores per socket	24	72	144
Sockets per node	2	1	1
Frequency [MHz]	2100	>=3200	>=3200
Full node floating-point performance			
Vector ISA	AVX512	SVE	SVE
Peak performance [Flop/cycle]	1536	1152	2304
Peak performance [GFlop/s]	3225.6	3801.6	7603.2
Full node main memory			
Number of memory channels	6	8	16
Size [GB]	96	480	480
Technology	DDR4-2666	LPDDR5	LPDDR5
Peak bandwidth [GB/s]	256	600	1200
Operative System			
OS distribution	SUSE Ent. Server 12 SP2	Ubuntu 22.04.2 LTS	
Kernel version	4.4.120-92.70-default	6.2.0-1009-nvidia-64k	

Backup - Compiler flags

Name	Version	Comments/Flags
MareNostrum		
GNU Compiler	gcc/12.1.0	-Ofast -march=skylake-avx512
Intel Compiler	intel/2021.4	-Ofast -xCORE-AVX512 -mtune=skylake
Math library	mkl/2021.4	Provided by environment modules
MPI library	impi/2018.4	Provided by environment modules
NVIDIA Cluster		
GNU Compiler	gcc/12.3.0	-Ofast -mcpu=native
NVIDIA Compiler	nvhpc/23.9	-O3 -tp=host
Arm Compiler	acfl/23.04.1	-Ofast -mcpu=neoverse-v2
Math library	armpl/23.04.1	Provided by environment modules
MPI library	openmpi/4.1.5-rc2	Provided by environment modules