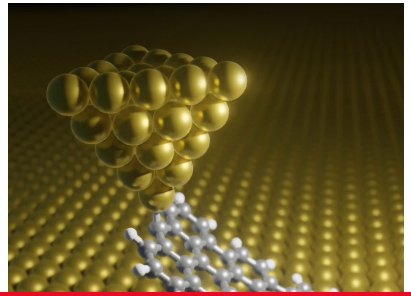
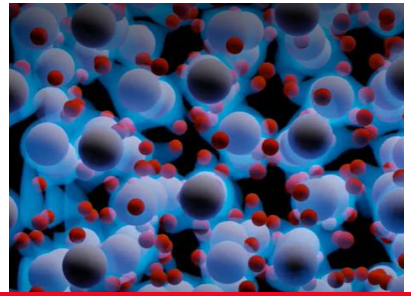
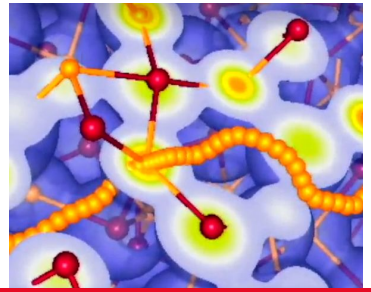
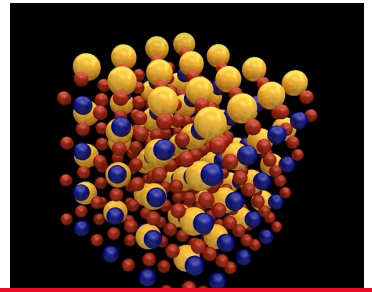
MAX DRIVING THE EXASCALE TRANSITION

ISC High Performance

MaXimizing portability and performance of material modelling on EuroHPC clusters

Workshop: Readiness of HPC extreme scale applications

Laura Bellentani, HLST - CINECA
l.bellentani@cineca.it

CINECA

MATERIAL DESIGN AT EXASCALE

MAX

LIGHTHOUSE CODES

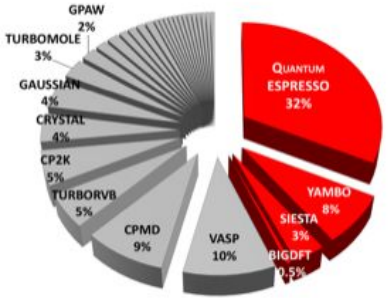
DOMAIN EXPERTS & CODE DEVELOPERS

HPC EXPERTS & DATA CENTRES

TECHNOLOGY & CO-DESIGN PARTNERS

Coe for HPC applications in material science

exploit **frontier HPC**
for material science research in strong link with **scientific communities**



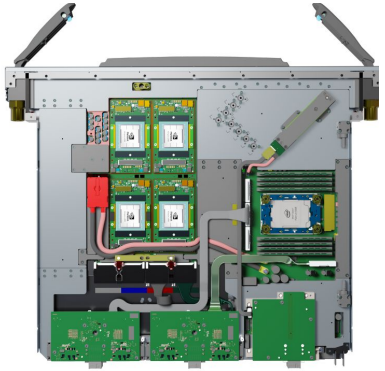
complementary | open source | HPC oriented | global impact

READINESS FOR EXASCALE

PORTABILITY

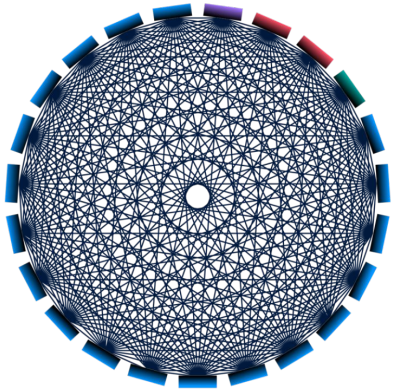
Support for multiple programming models and hardware vendors.

EFFICIENCY



Optimized hardware utilization for performance and energy.

SCALING

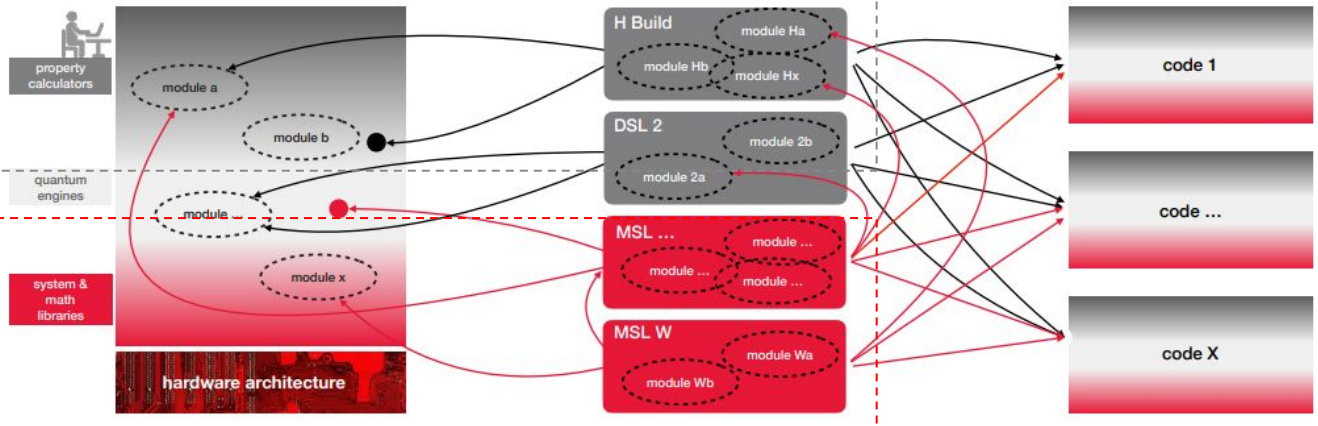


Demonstrated ability to scale across thousands of cores.

SEPARATION OF CONCERNS

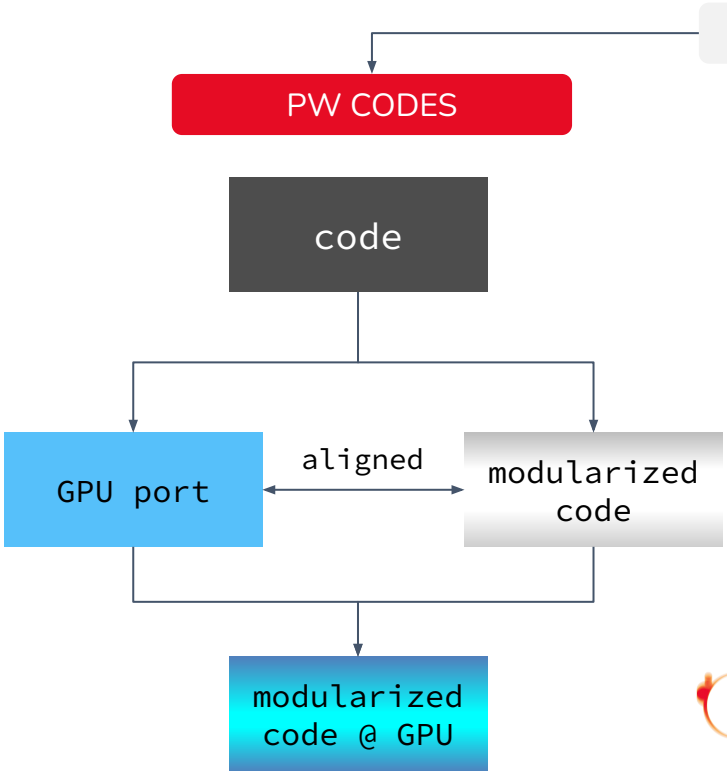
The performance critical layers must be hidden from the higher level functions

PROPERTY CALCULATOR & HAMILTONIAN BUILDER
architecture-agnostic, reusable among codes



DOMAIN-SPECIFIC/MATHEMATICAL LIBRARIES
code-agnostic, externalized, fine-tuned for hardware

STRATEGIES FOR GPU PORTING



COMPLEMENTARY

PW CODES

LOCALISED BASIS

spread compute intensive parts

offloaded data accessed through all the execution

QUANTUMESPRESSO

Yambo

www.flapw.de

Fleur

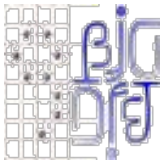
STRATEGIES FOR GPU PORTING

COMPLEMENTARY

PW CODES

LOCALISED BASIS

localized compute intensive regions



siesta™

offload leveraged by external libraries, e.g. ELSI, ELPA, PEXSI,

code

modularized code

modularized code @ GPU

THE STATUS

	QUANTUM ESPRESSO	YAMBO	SIESTA	BigDFT	FLEUR
DEUCALION	D	D	W	S	S
	S	S	D ◆	S	S
	S	S	D ◆	S	S
DISCOVERER	M	S	D ◆	S	S
KAROLINA	M	M	D ◆	D	D
	M	M	D ◆	S	S
LEONARDO	M ◆	M ◆	M ◆	M	M
	M ◆	M ◆	M ◆	M ◆	M ◆
LUMI	M ◆	M	D ◆	S	S
	M ◆	D ◆	D	D	W
MareNostrum5	M ◆	D ◆	M ◆	D	D
	M ◆	D ◆	M ◆	S	D
MELUXINA	D	D	D ◆	S	S
	M	M	D ◆	S	S
VEGA	M	M	D ◆	S	S
	M	M	M	M	S

M **Deployed** module available

D **Demonstrated** developers installation

S **Supported architectures** tested on similar architecture

W **Work in progress**

◆ **Benchmarks** configured and executed

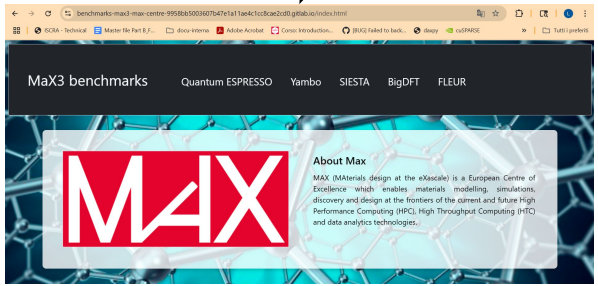
ARM
CPU
GPU

January 2025

Spack

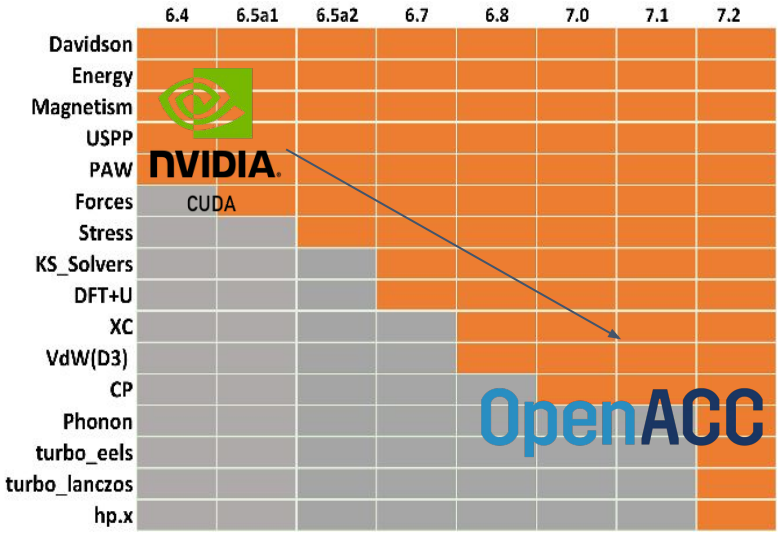
Score-P

JUBE BENCHMARKING ENVIRONMENT

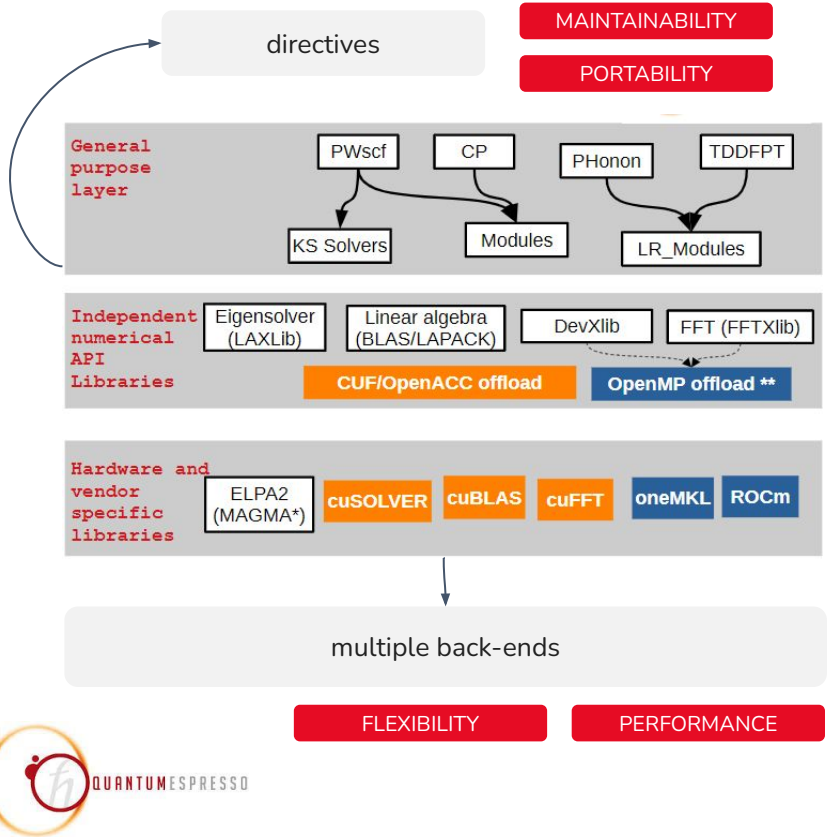


CINECA

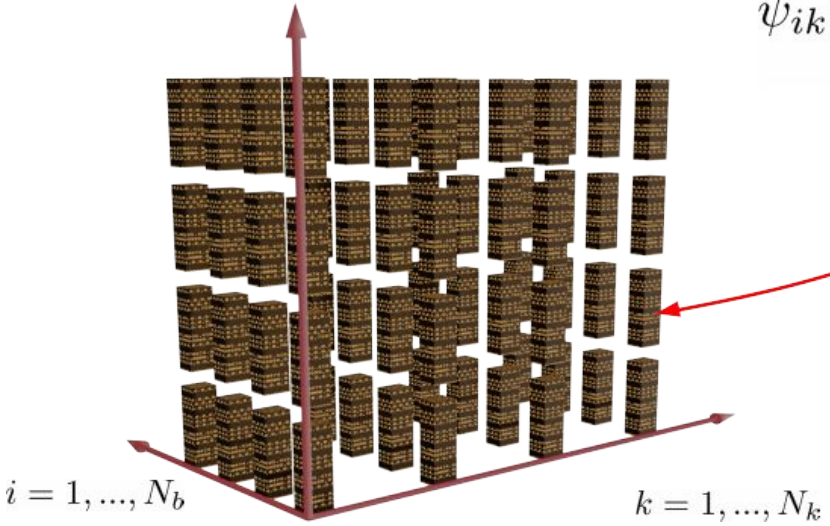
DIRECTIVE-BASED GPU OFFLOAD IN QE



- First porting with CUDAFortran
- Transition to OpenACC for maintainability
- New backend for AMD GPUs (OpenMP offload)



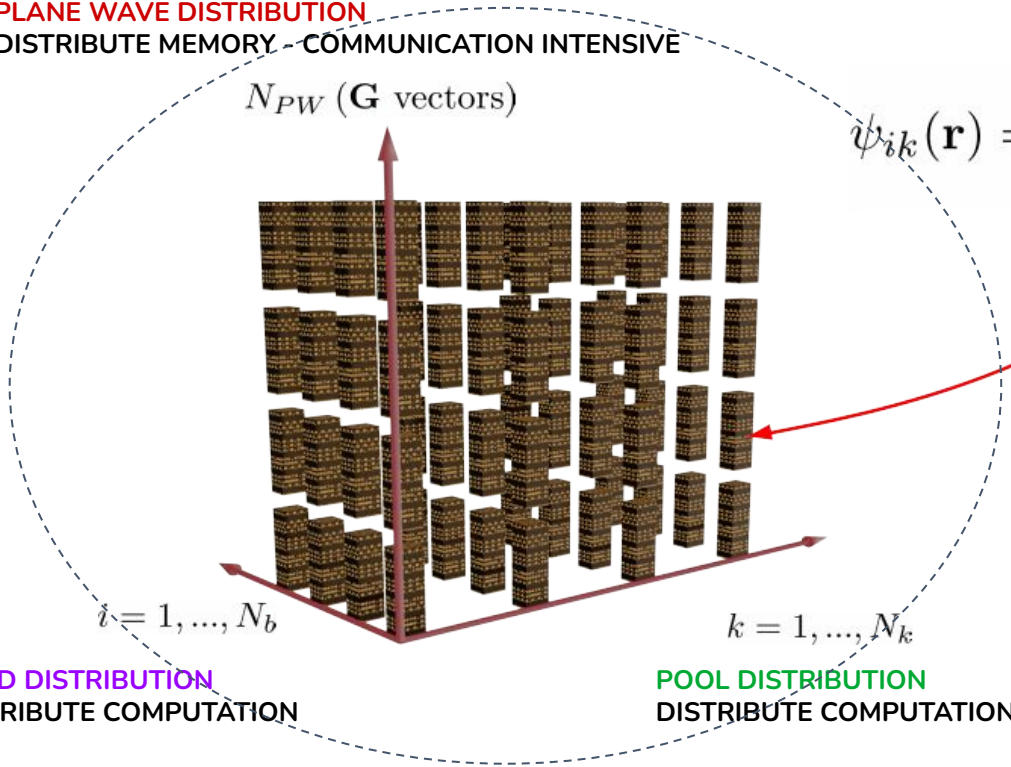
MPI HIERARCHIES



$$\psi_{ik}(\mathbf{r}) = \sum_{\mathbf{G}}^{N_{PW}} C_{\mathbf{G},ik} \frac{e^{i((\mathbf{G}+\mathbf{k})\cdot\mathbf{r})}}{\sqrt{\Omega}}$$

MPI HIERARCHIES

PLANE WAVE DISTRIBUTION
DISTRIBUTE MEMORY - COMMUNICATION INTENSIVE



$$\psi_{ik}(\mathbf{r}) = \sum_{\mathbf{G}}^{N_{PW}} C_{\mathbf{G},ik} \frac{e^{i((\mathbf{G}+\mathbf{k})\cdot\mathbf{r})}}{\sqrt{\Omega}}$$

IMAGE

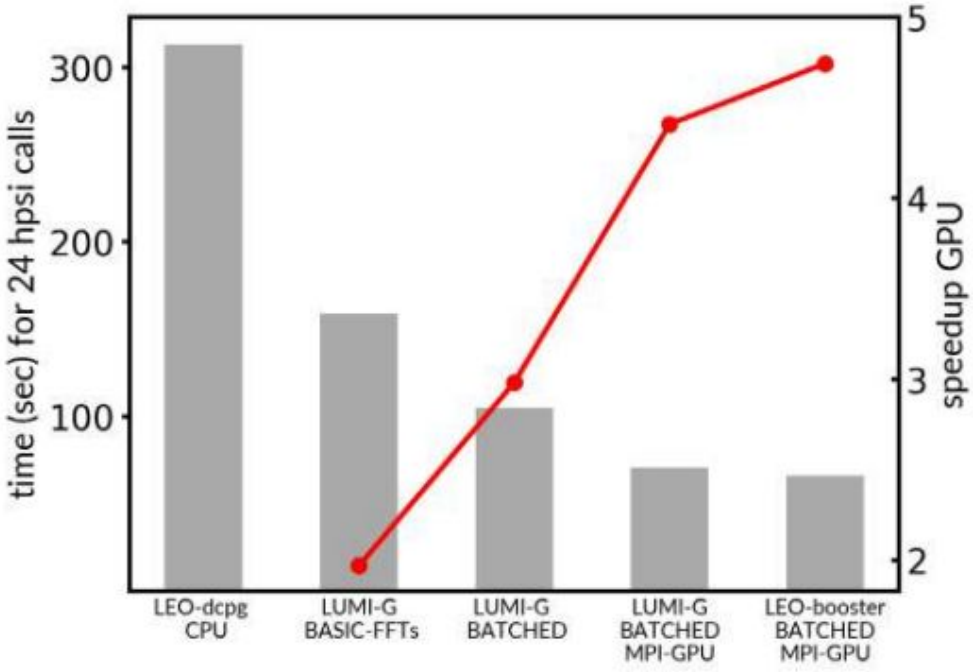
DISTRIBUTE REPLICAS

BAND DISTRIBUTION
DISTRIBUTE COMPUTATION

POOL DISTRIBUTION
DISTRIBUTE COMPUTATION

SPECIALIZED BACKENDS - memory distribution

CrI3- 480 atoms, 3240 electrons, 2 nodes



Driver of FFT distribution optimized for GPUs

1. non-blocking communications
2. batching for comm/compute overlap
3. GPUdirect

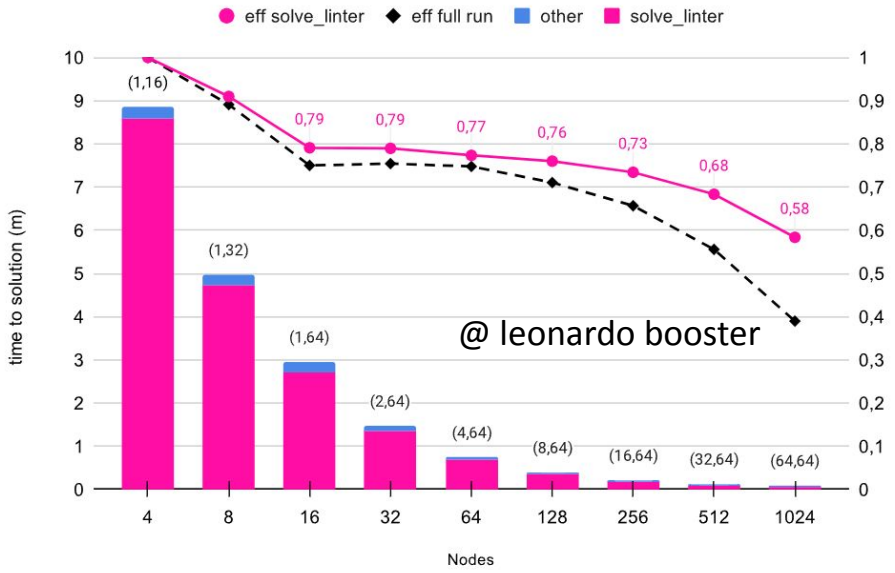
→ GPU execution outperform by ~7x.

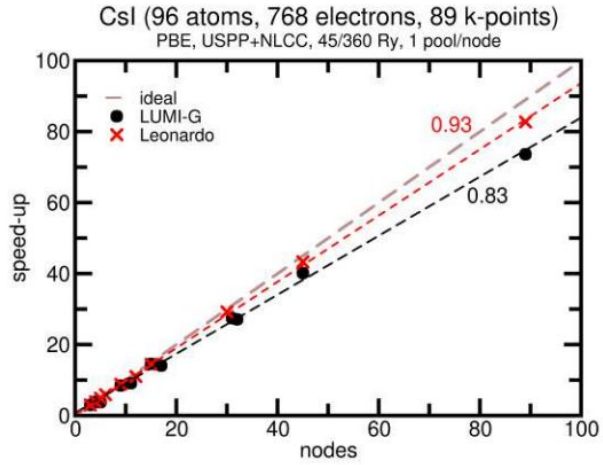
LUMI-G	4 AMD MI250X 64x2 GB HBM
LEO BOOSTER	4 Nvidia A100 64 GB HBM2
LEO DCGP	Sapphyre Rapids 112 cores

F. Ruffino et al.. Procedia Computer Science. 240. 52-60

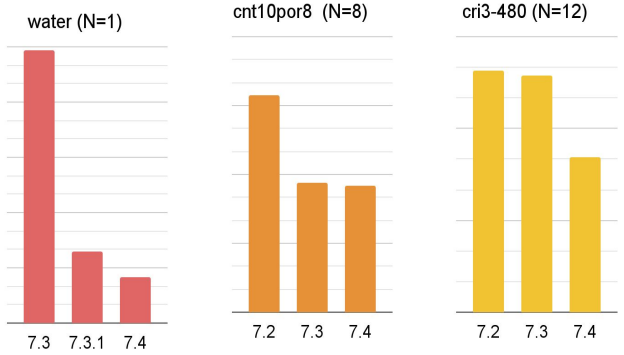
IMAGES AND POOLS - compute distribution

I. Carnimeo et al., , Journal of Chemical Theory and Computation 2023 19 (20)

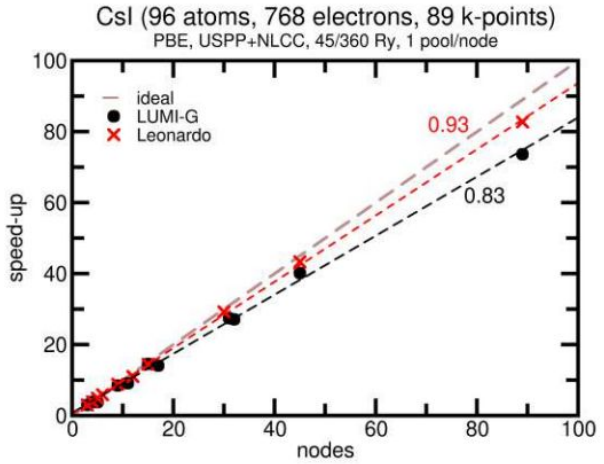


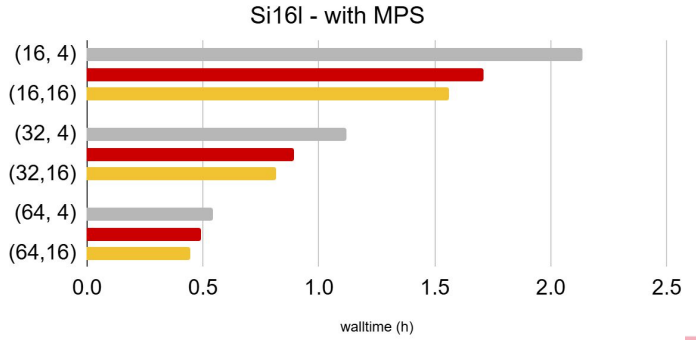


- Efficient over infiniband
- prone to overhead (small systems)
- improved data mapping, porting of small routines across versions
- Multi-process-service to improve GPU utilization



IMAGES AND POOLS - compute distribution



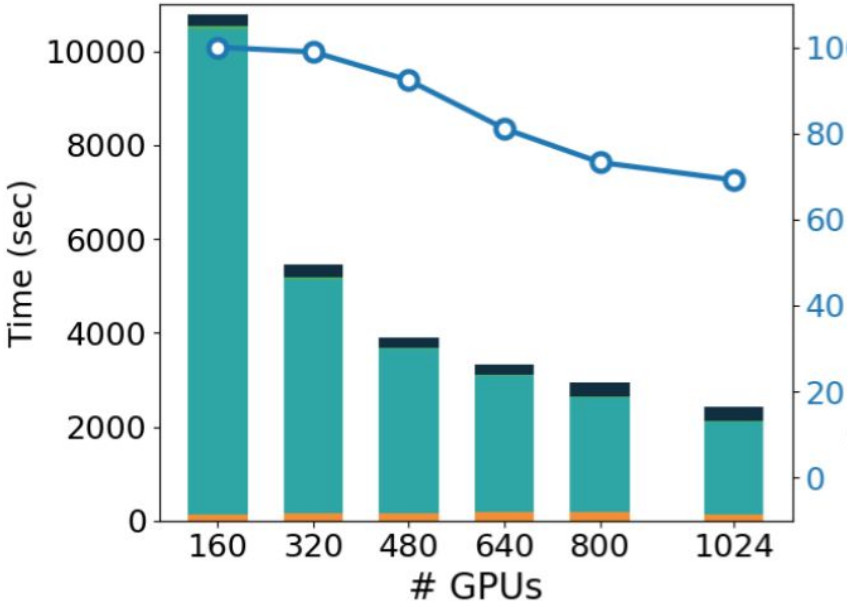


- Efficient over infiniband
- prone to overhead (small systems)
- improved data mapping, porting of small routines across versions
- Multi-process-service to improve GPU utilization

I. Carnimeo et al., , Journal of Chemical Theory and Computation 2023 19 (20)

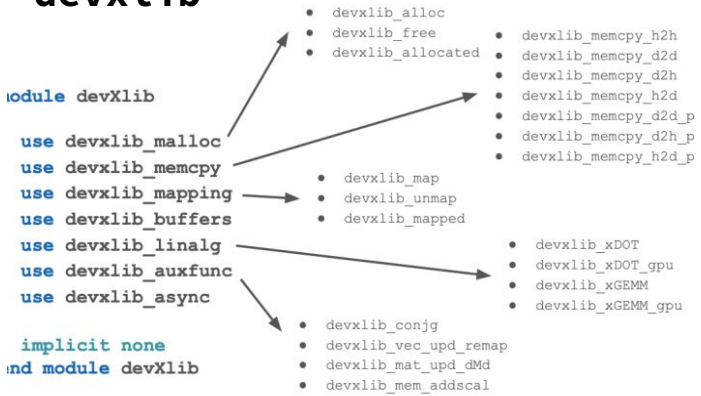
DEVXLIB IN YAMBO

D Sangalli *et al* 2019 *J. Phys.: Condens. Matter* 31 325902

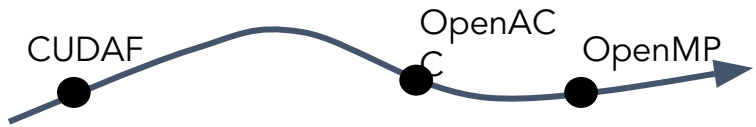


linear algebra, FFTs on GPUs, specialized backends, custom kernels. data mapping

devxlib

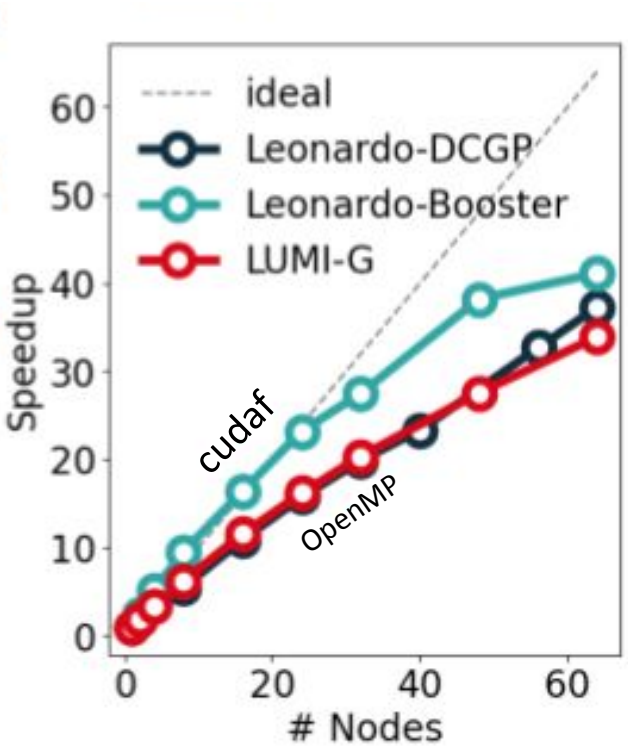


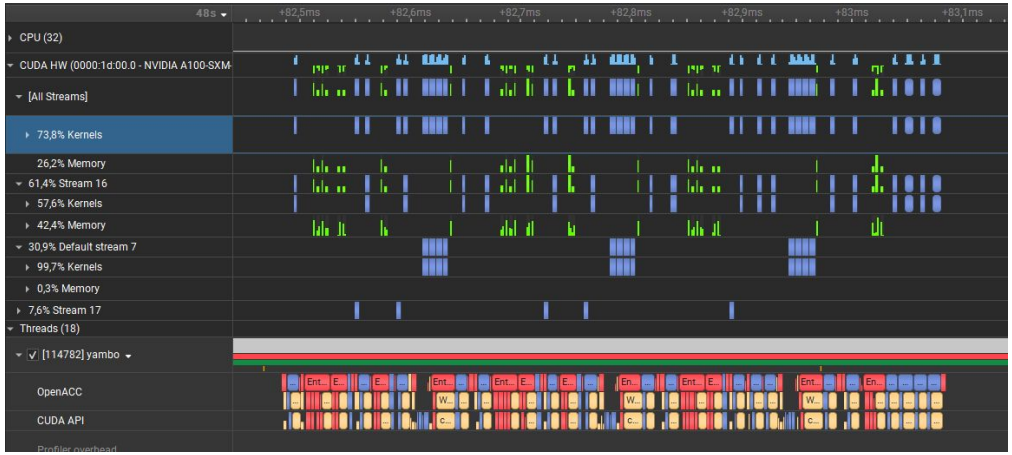
- yambo offers fairly independent parallelization layers
- porting to new programming model streamlined by devxlib

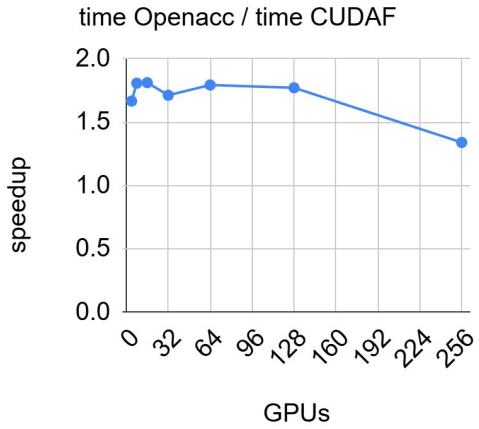


Yambo

DIRECTIVE BASED PROGRAMMING MODEL IN YAMBO

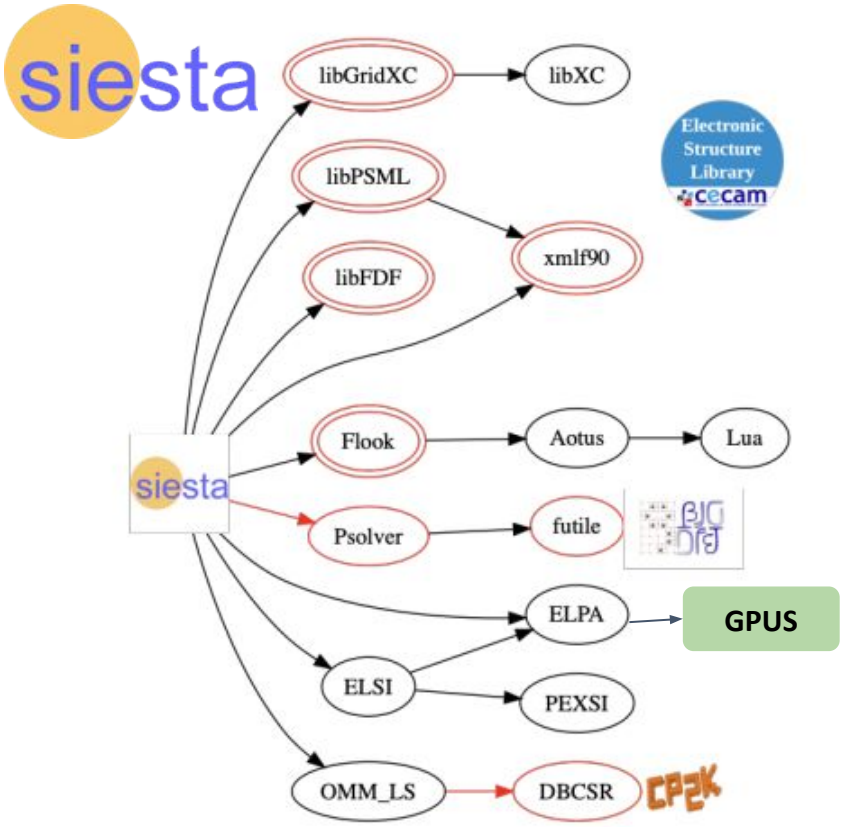


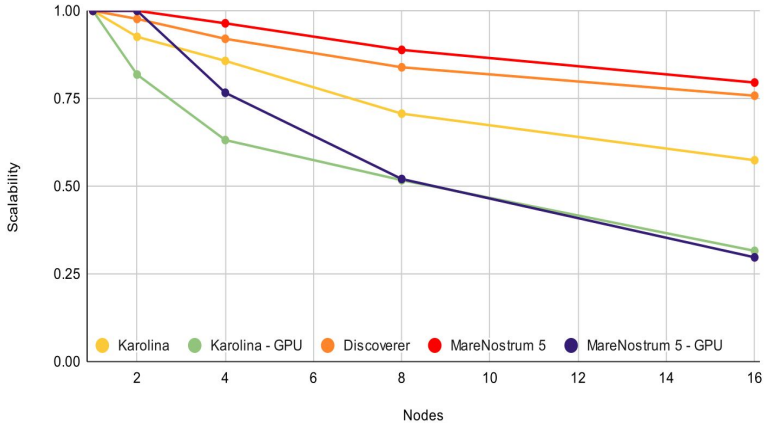


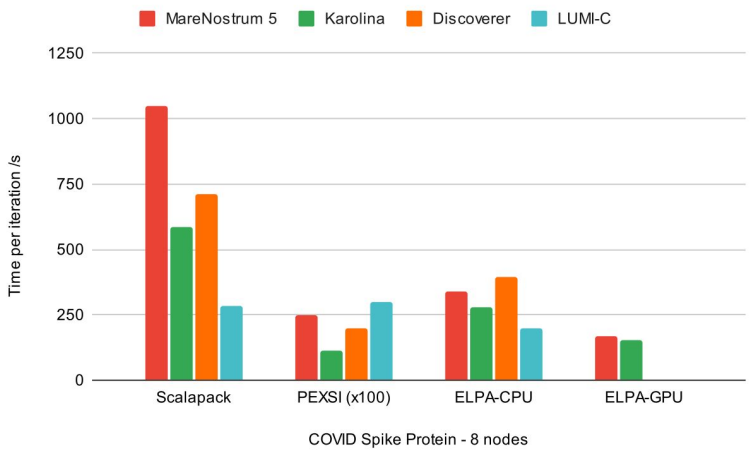


- Scalability reduces for directive-based programming models
- Testing async, device resident data mapping
- Integration of new solvers (cusolvemp)

OFFLOAD VIA ACCELERATED LIBRARIES

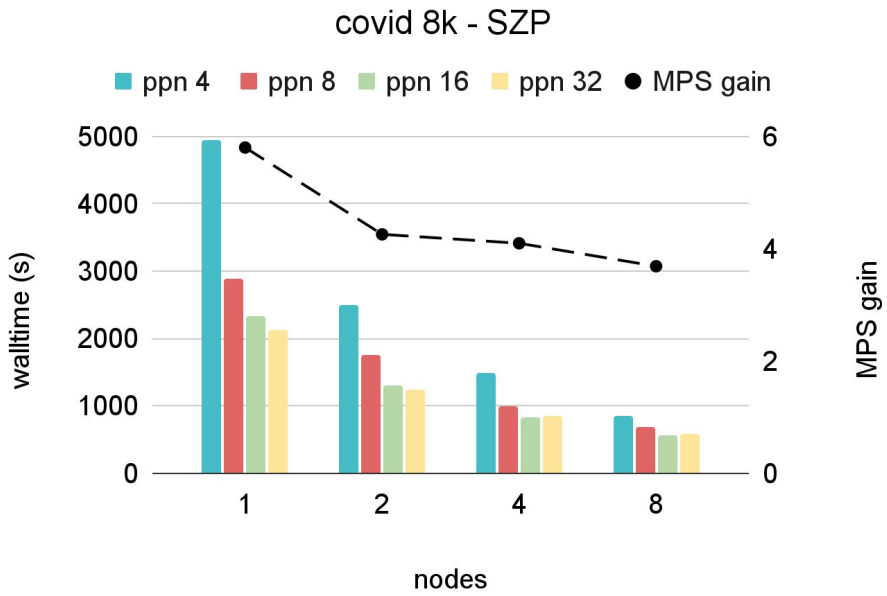


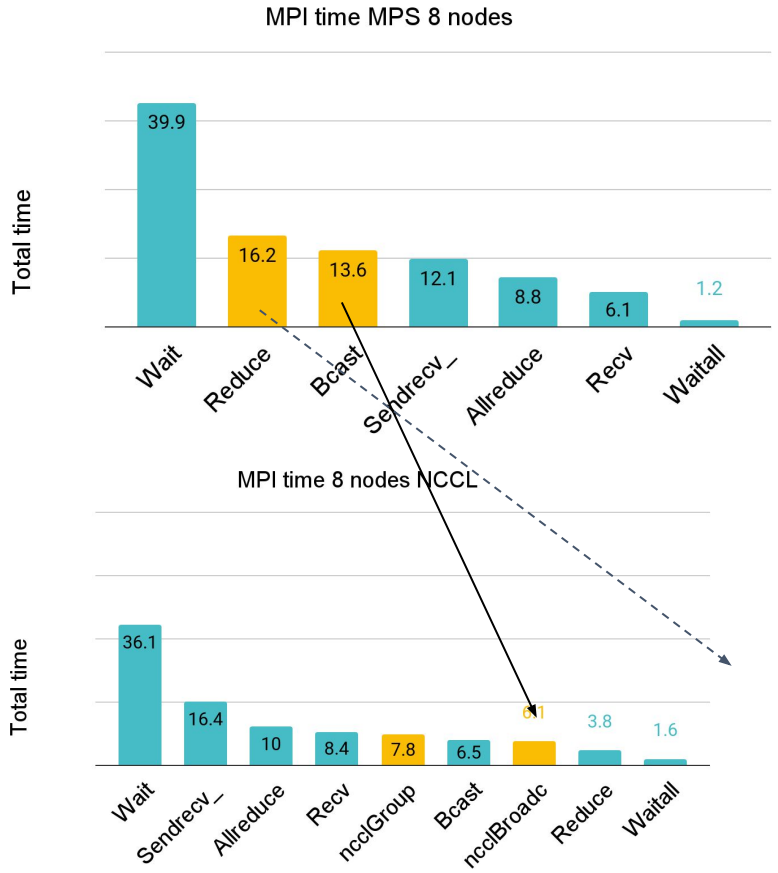




OFFLOAD VIA ACCELERATED LIBRARIES

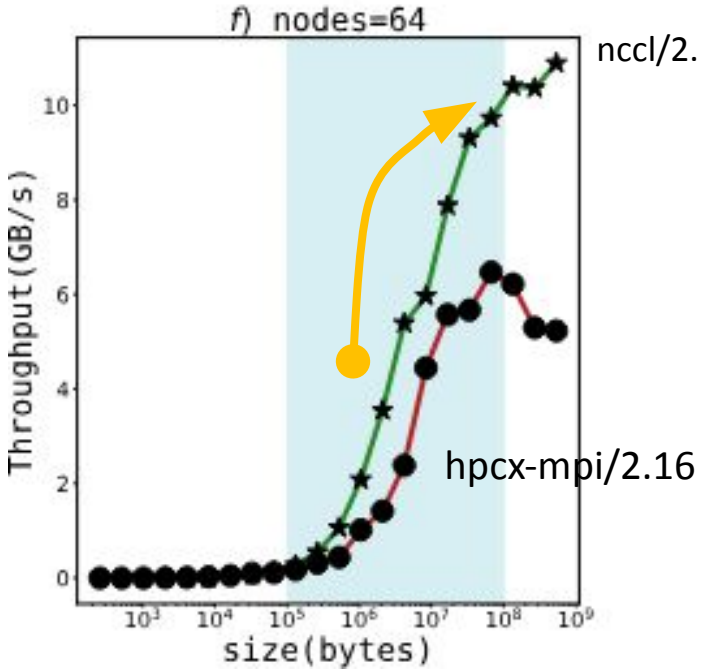
- no support to awareness in ELPA yet (only experimental)
- NCCL version involves only few collectives
- MPS quite effective to cover overhead of CPU staging

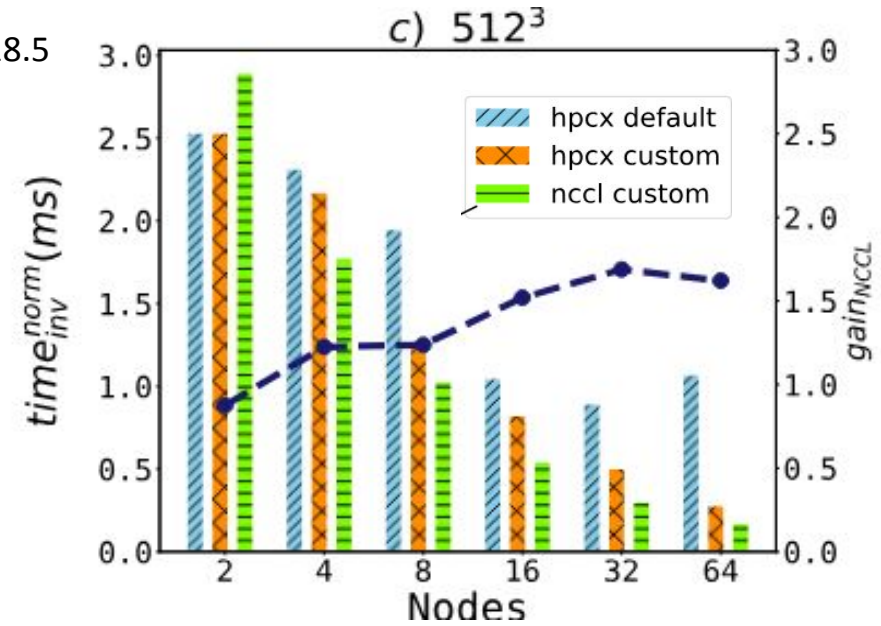




NEW BACKENDS FOR COMMUNICATIONS

PLANE WAVE distribution, FFTXlib (All-to-All)

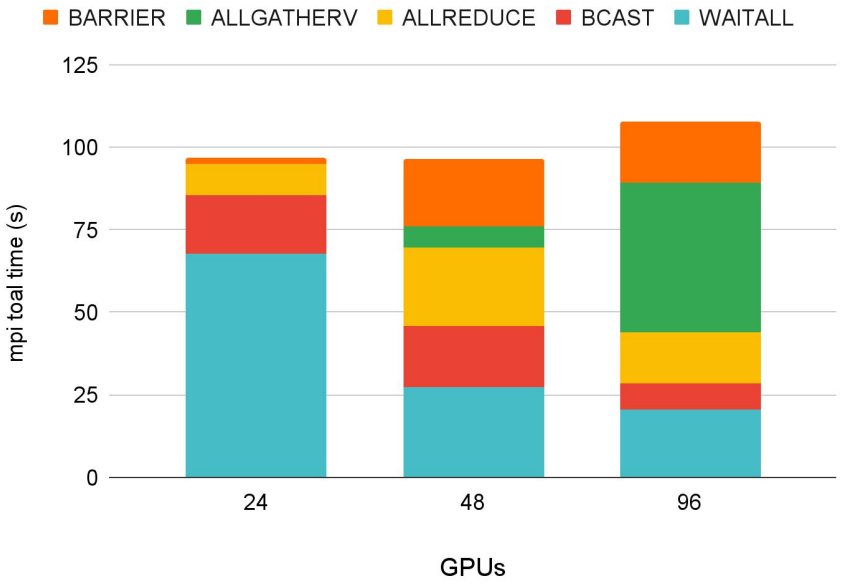


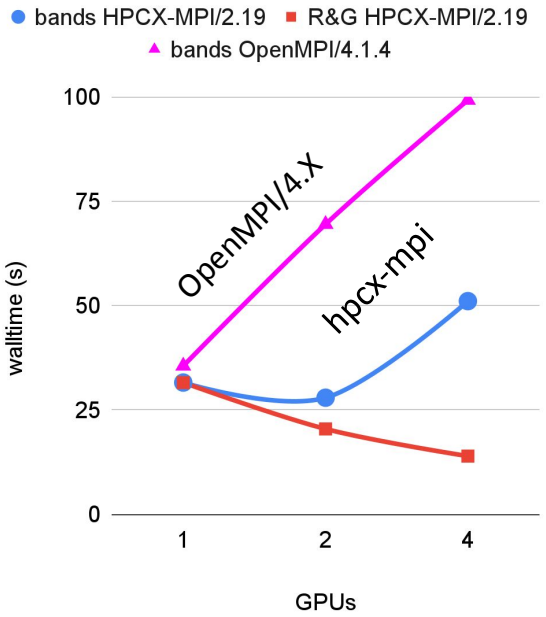


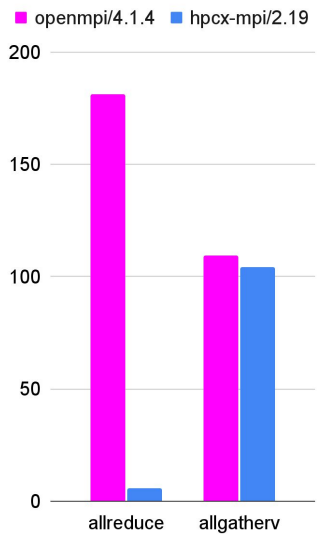
NCCL and HPCX-MPI with improved batching

COMMUNICATION REDESIGN

Band distribution (Allreduce,Allgatherv)

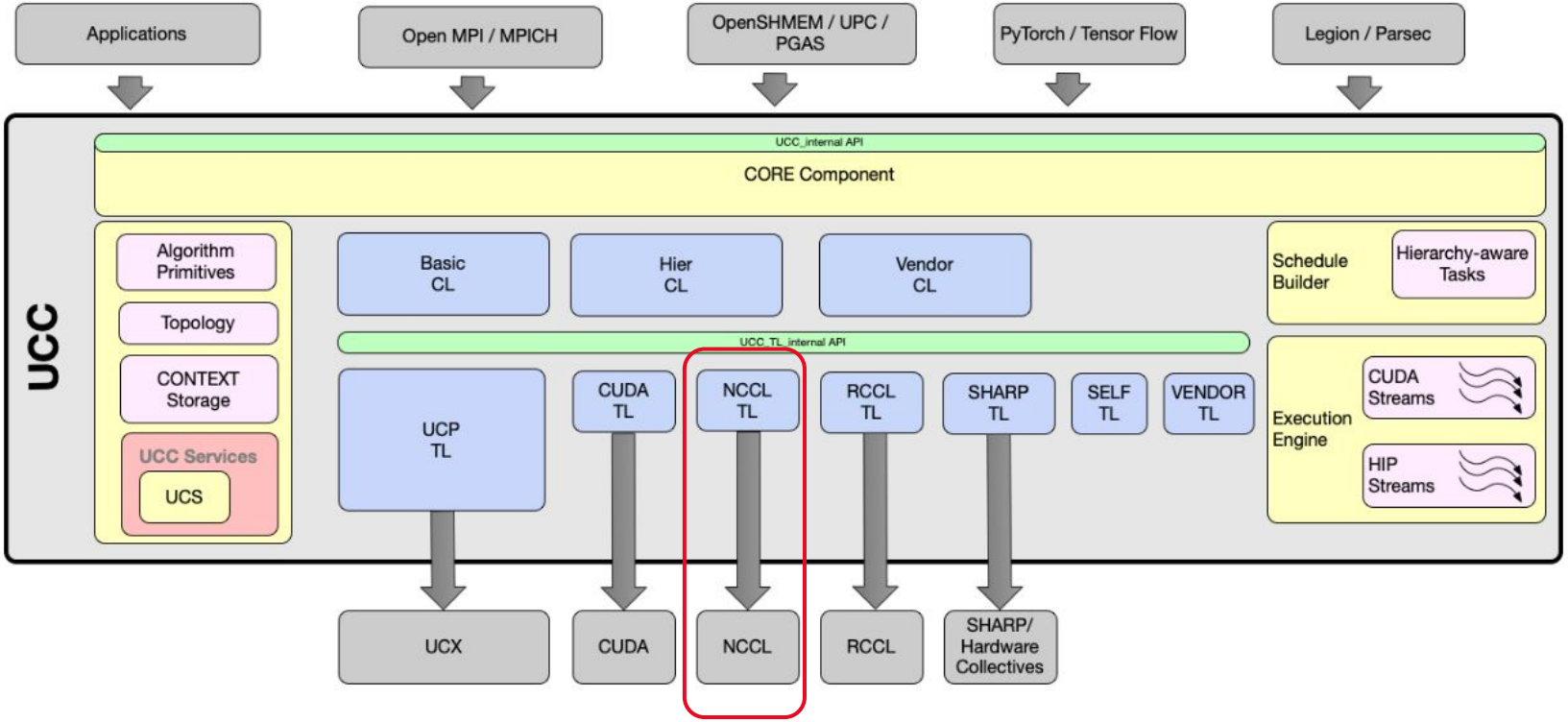






Potential to reduce the impact of Waitall to scale beyond R&G over infiniband
But allgather are slow, also over NVLlink

OpenMPI/5 vs HPCX-MPI



FUTURE PERSPECTIVES

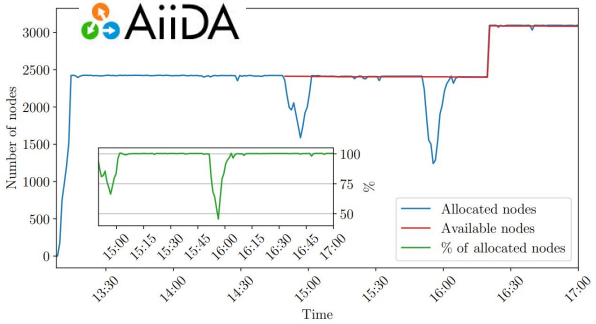
IMPROVING SINGLE-HARDWARE USAGE

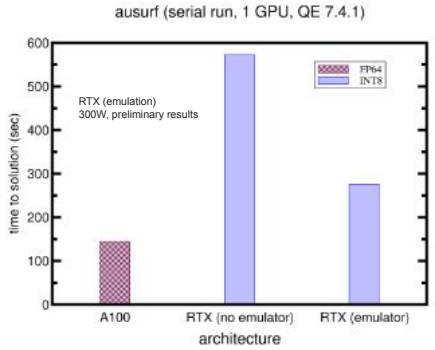
- emulators for future architectures
- Improved acceleration for small metallic systems

TOWARDS EXTREME-SCALING

- leveraging distributed libraries
- New workflows for exascale
  - improve interfaces and interoperability among codes
  - enable distribution of independent calculations
- HTC with AiiDA

https://doi.org/10.48550/arXiv.2505.20366Focus to learn more





https://doi.org/10.1016/j.cpc.2024.109439.

	CPU					GPU				optimized GPU			
#CPU	32	32	32	64	64	1	2	4	8	1	2	4	8
#GPU	0	0	0	0	0	1	2	4	8	1	2	4	8
#task(np)	32	32	32	64	64	1	2	4	8	1	2	4	8
#pool(nk)	8	16	32	32	64	1	2	4	8	1	2	4	8
time (s)	11400	10560	8040	5640	4500	107640	54780	27720	14100	16080	7860	4680	2153
time (m)	190	176	134	94	75	1794	913	462	235	268	131	78	36

		cuSOLVERMp		ScaLAPACK		
Nodes	Tasks/Node	Linear Algebra	getrf/getrs	Linear Algebra	SU LA	SU func
1	4	118.16	73.58	14760.00	124.9x	200.6x
4	4	116.46	61.05	4080.00	35.0x	66.9x
16	4	119.72	60.41	1063.00	8.9x	17.6x