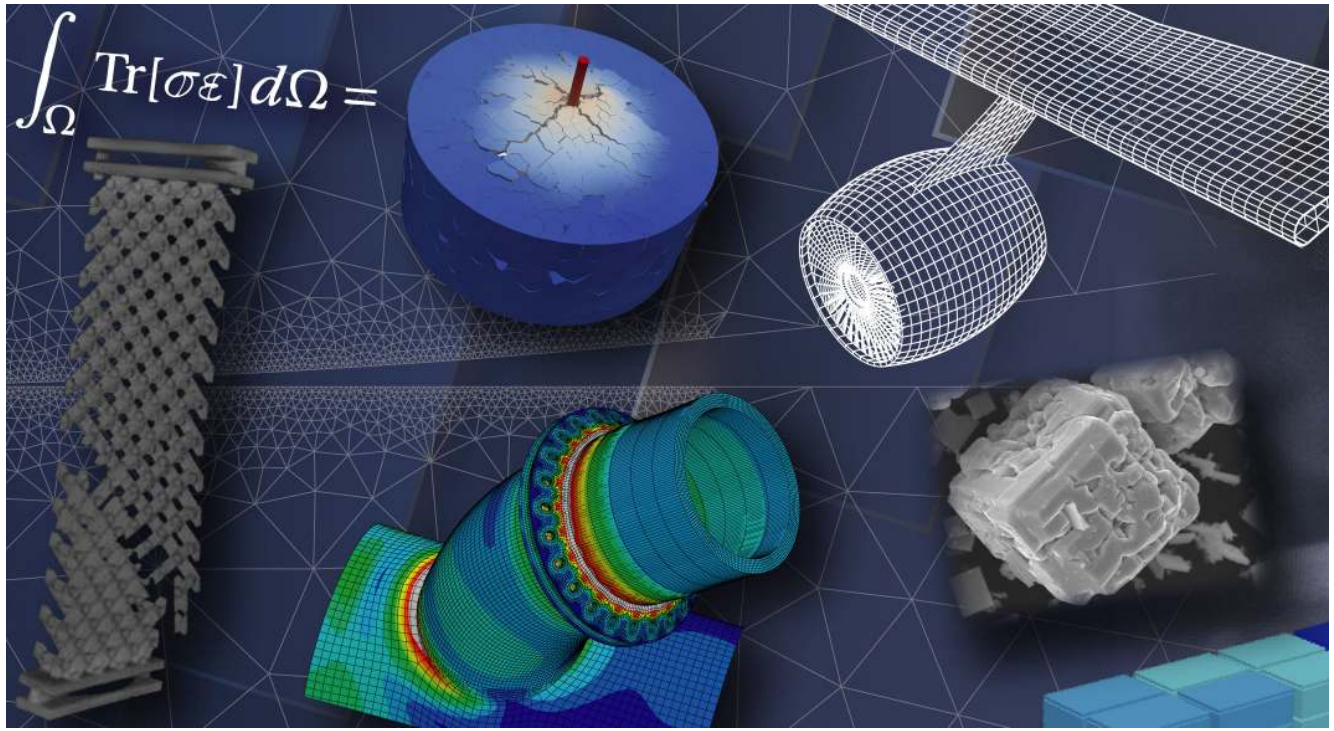
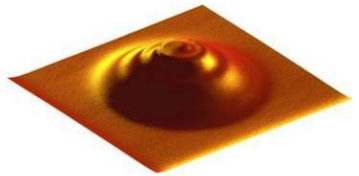
Parallel computing in Matlab using MPI4.0 via the Caryam C/C++ interface

Thomas Verbeke, Anne-Sophie Mouronval, Aurelia Ruda, Pierre-Alain Boucard

Laboratoire de Mécanique Paris-Saclay
91190 Gif-sur-Yvette

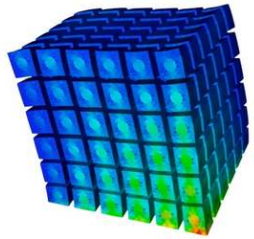
LMPS - A new lab





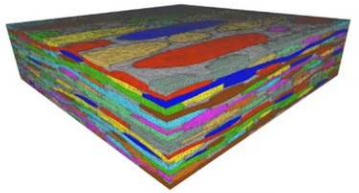
COMMET

COmportement des Matériaux, Modélisation, Expérimentation et Théorie



STAN

Science et Techniques Avancées en mécanique Numérique



MILA

MILieux Architecturés

OMEIR

Ouvrages, Matériaux, Environnement : Interactions et Risques

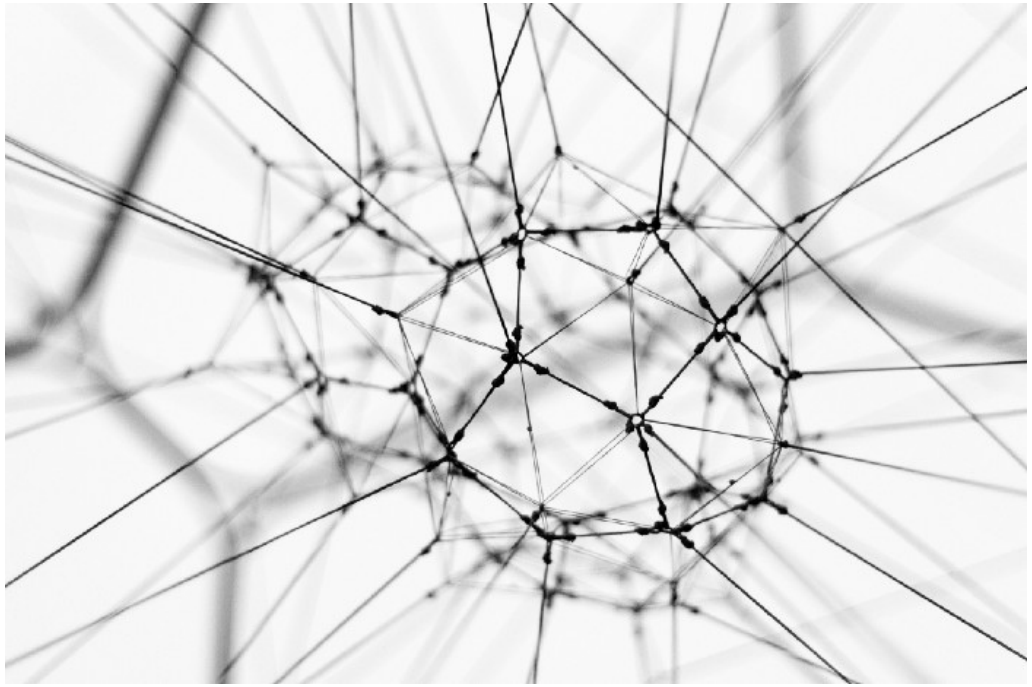
LMT → LMPS ← MSSMat

Since 1975 — Since 2022 — Since 1987

LMPS - Computing Resources

Personal laptop/station		1 x Intel Corei7 7820 @ 2.9GHz = 8 cores	1 x 16GB
Multi-users workstations		... 18 x 2 x Intel Xeon Silver 4116 @ 2.1GHz = 432 cores ...	18 x 256GB
Regional mesocenter RUCHE		(144|216) x 2 x Intel Xeon Gold 6230 @ 2.1GHz = 7680 cores	216 x 192GB

LMPS - Numerical Framework



Industrial codes

Home designed codes

Data

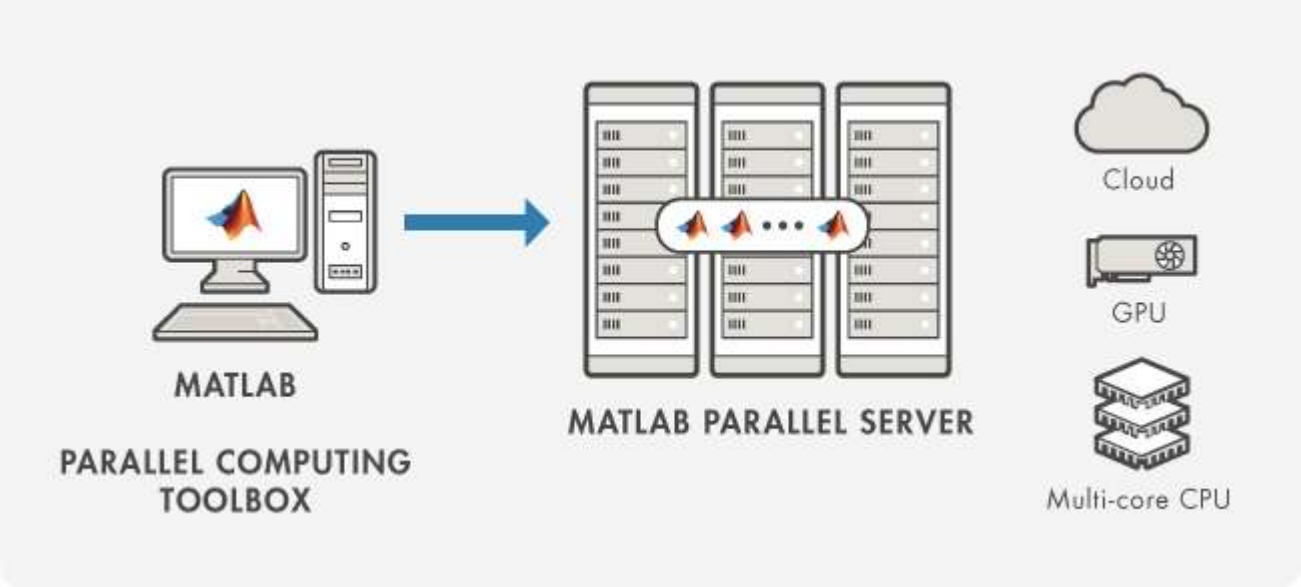
Languages

Trying to escape from technical debt

Scaling with MATLAB

MATLAB®

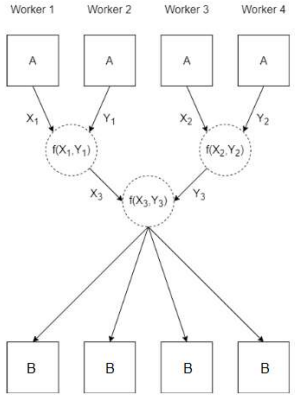
Suitable for fast prototyping



Parallel Computing Toolbox - Pros & Cons

MATLAB®

Based on the host system MPI library
Use third-party library like scaLAPACK



spmdReduce

	Interface : *spmd*	
`spmdSize`		`MPI_Comm_size`
`spmdIndex`		`MPI_Comm_rank`
`spmdBarrier`		`MPI_Barrier`
`spmdSend`		`MPI_Send`
`spmdReceive`		`MPI_Recv`
`spmdBroadcast`		`MPI_Bcast`
`spmdReduce`		`MPI_Reduce|MPI_Allreduce`
`spmdPlus`		`MPI_Allreduce('MPI_SUM')`
`spmdCat`		`MPI_Allgather`

Parallel Computing Toolbox - Pros & Cons

Wonderfull ergonomy!

```
spmd
    A = spmdIndex;
    B = spmdReduce(@max,A);
end
```

```
spmd
    B = spmdCat(spmdIndex);
end
```

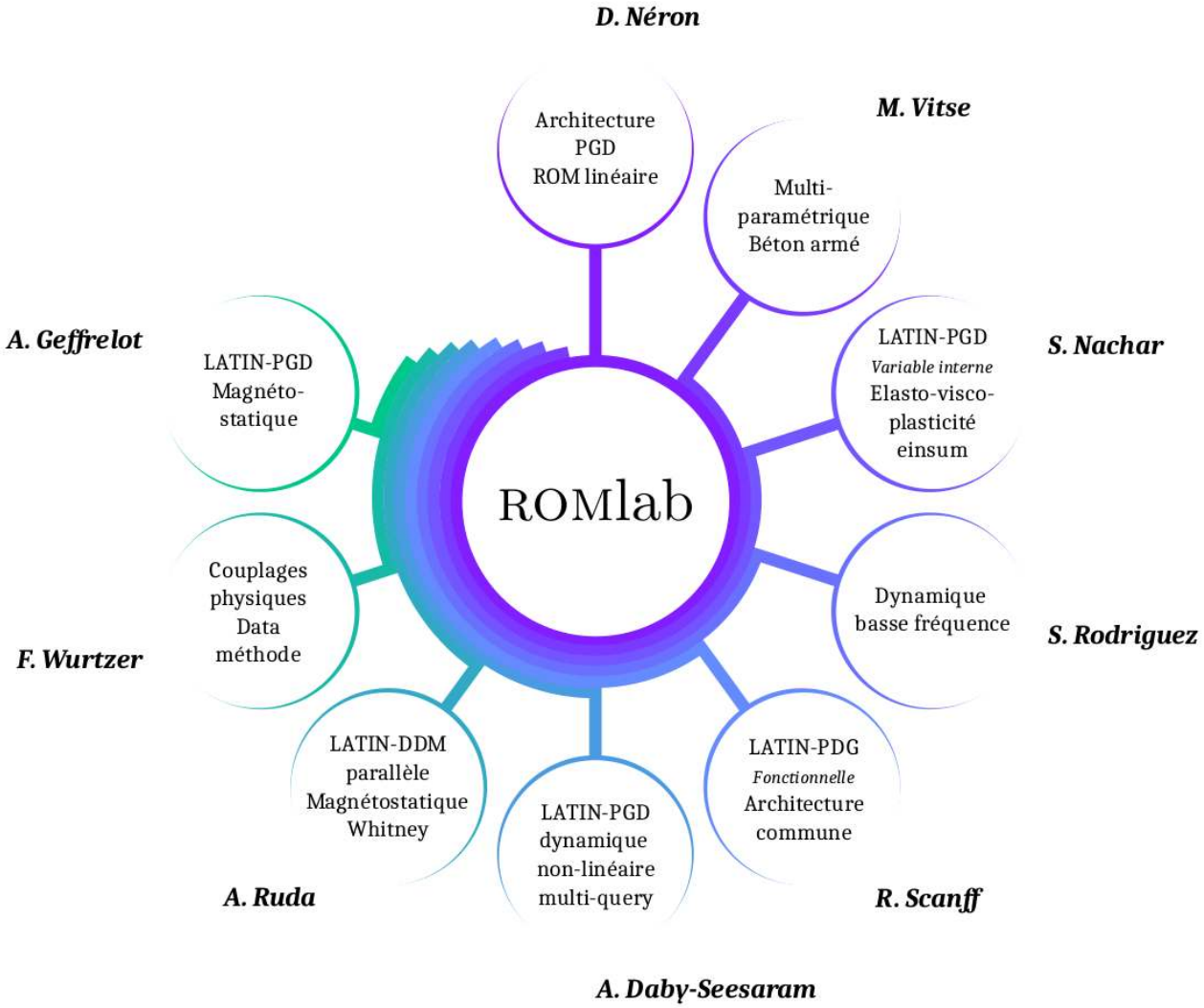
But:

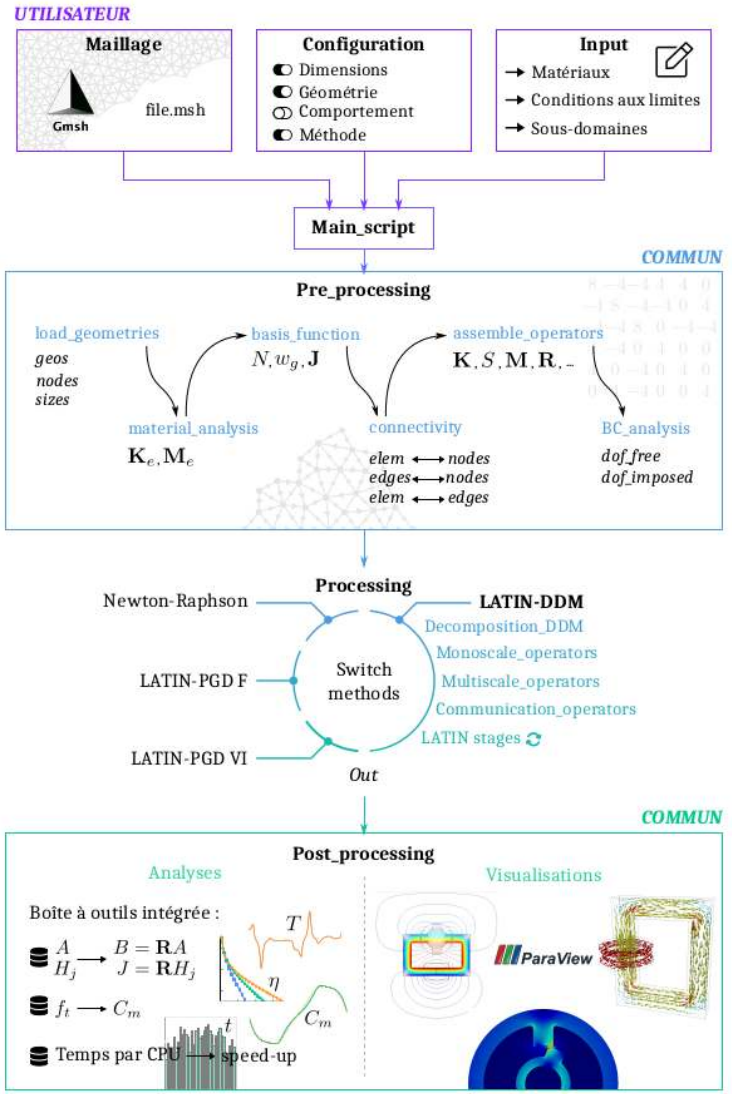
High memory demand.

Limited to 12 processes & intranode only.

Too few MPI functionalities.

Home designed code - ROMlab





Mex Interface, how to?

MATLAB®

Coding Mex gateway functions (by hands)

Until v2017b

mexFunction (C)

Entry point to C/C++ MEX function built with C Matrix API

Since v2018a

matlab::mex::Function

Base class for C++ MEX functions

- copy on write semantics
- access to data with iterators

Using Matlab coder (by hands)

MATLAB Coder

Generate C and C++ code from MATLAB code

Since v2011a

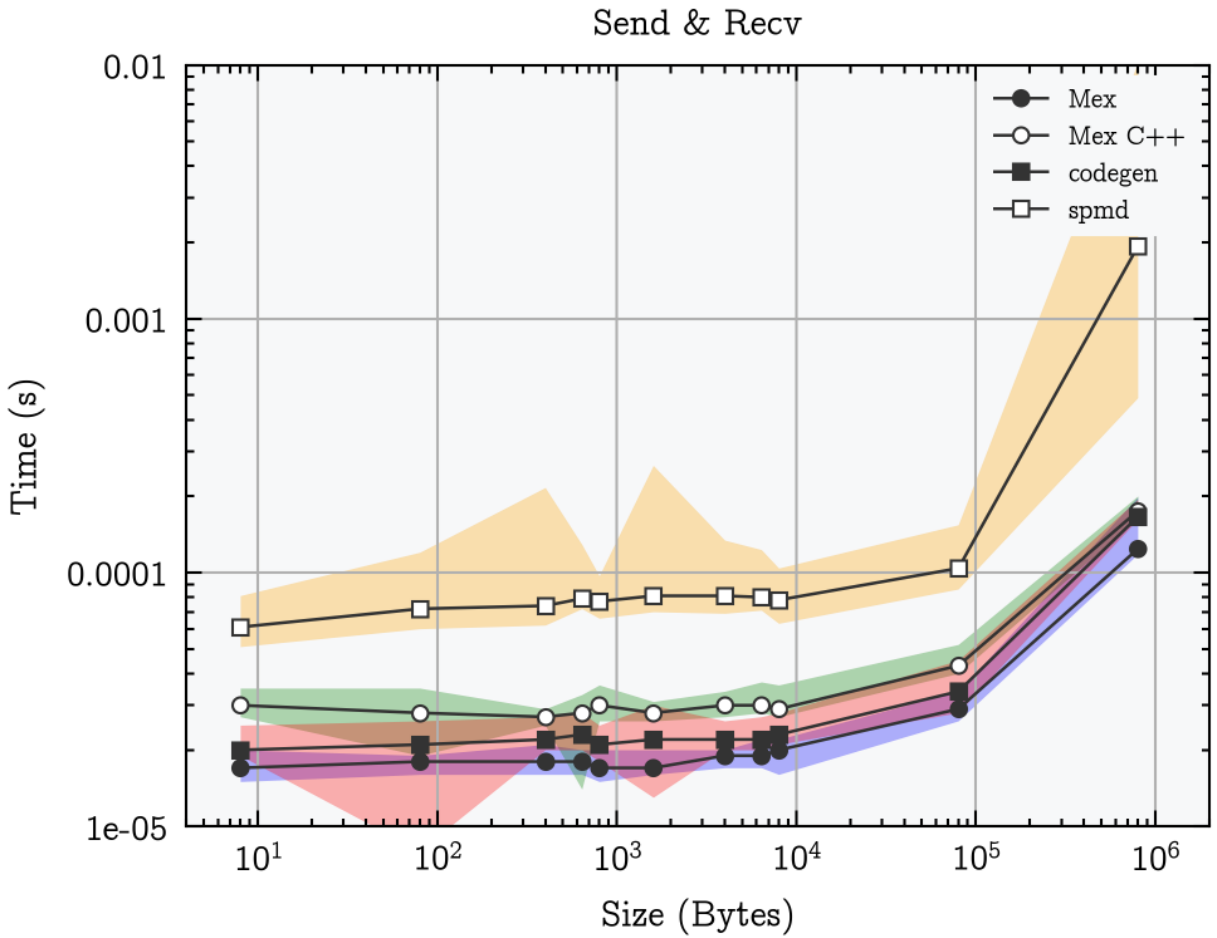
```
function y = myAdd(u,v) %#codegen
y = u + v;
end
```

```
codegen -config:mex myAdd.m -args {1,2} -args {int8(2),int8(3)}
```

- inlining
- opaque objects
- avoid useless copy
- control stack/dynamic memory allocation

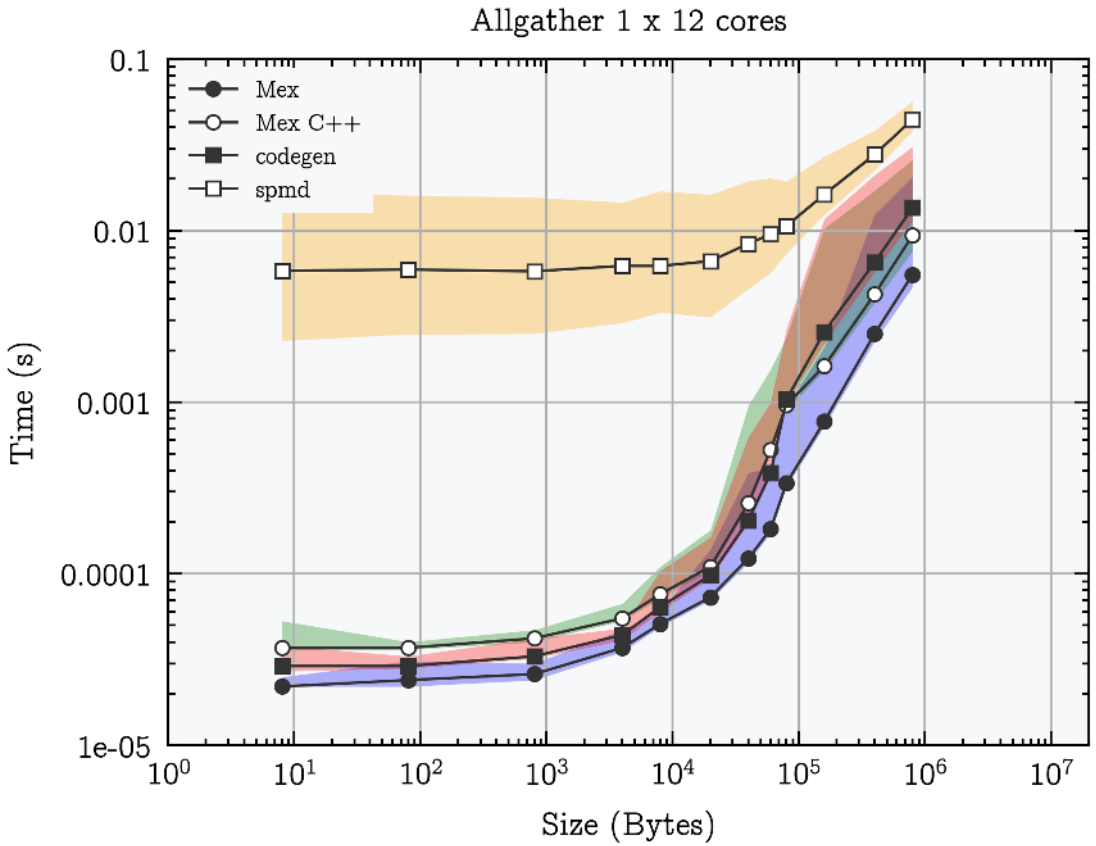
Benchmarks - Point-to-point communication

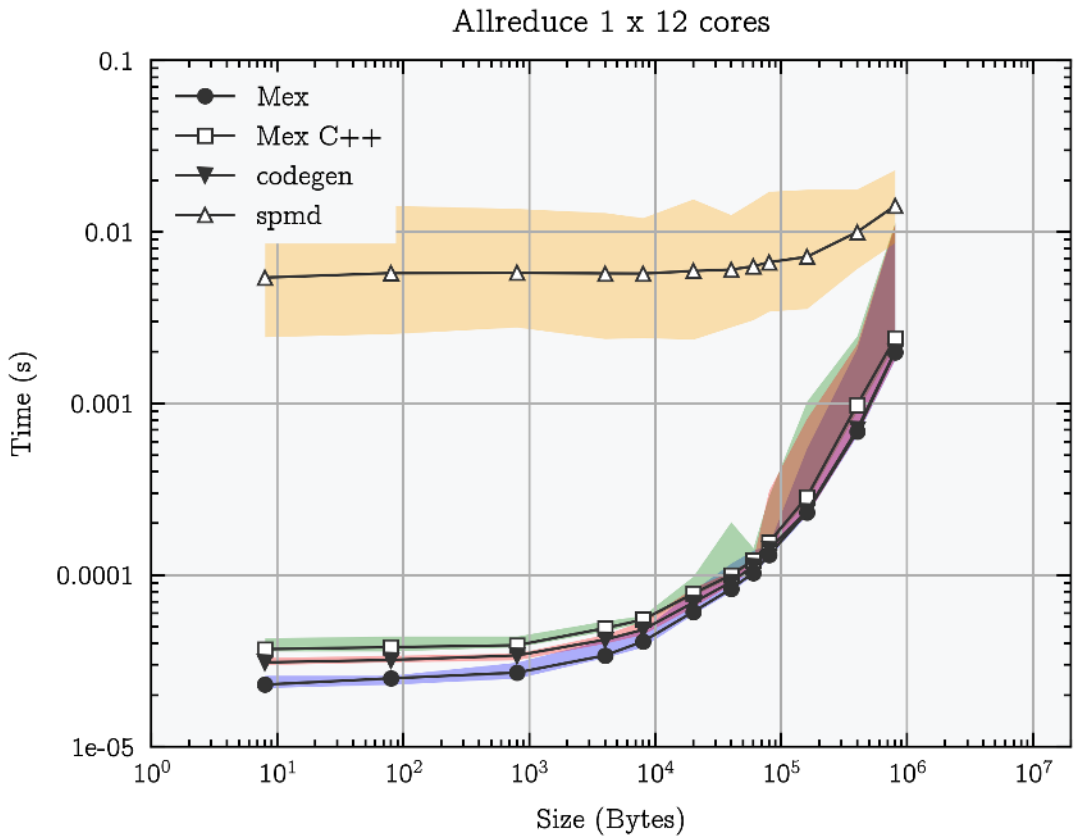
1 node Xeon Silver 4116 2.10GHz



Benchmarks - Collective communication

1 node Xeon Silver 4116 2.10GHz



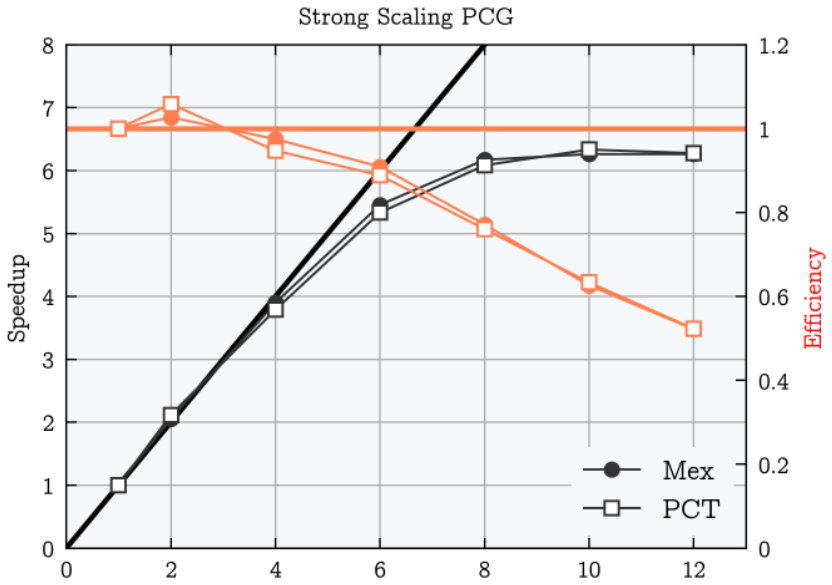


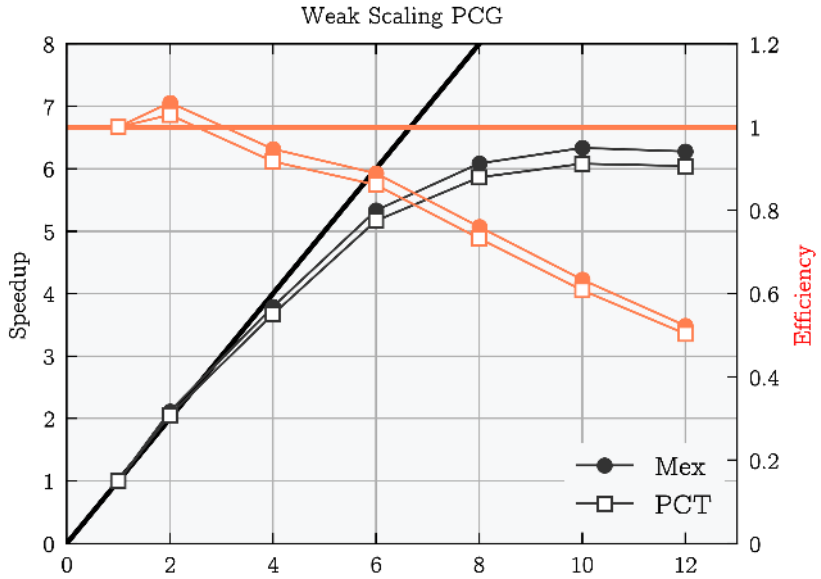
Benchmarks - Parallel Conjugate Gradient

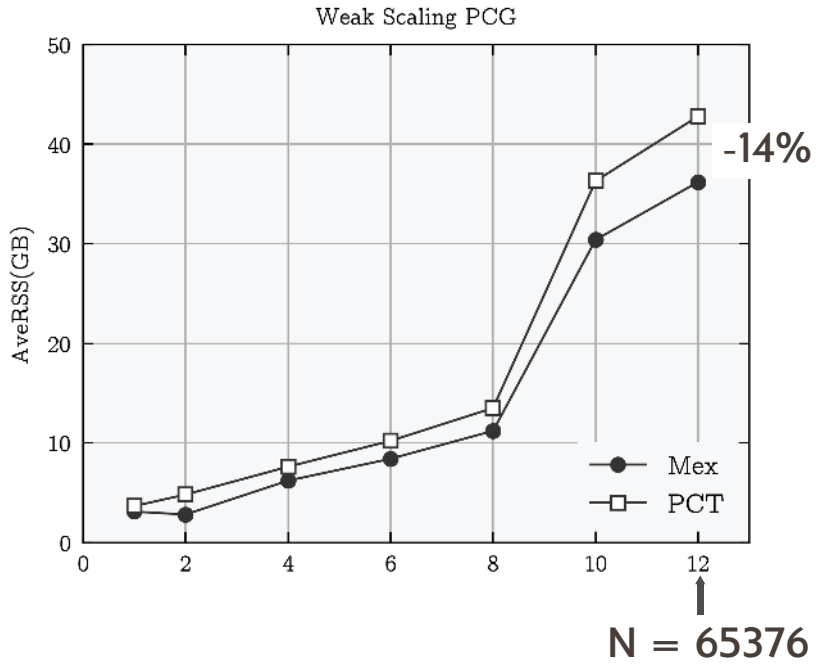
Square Dense matrix decomposed

- 10 iterations
- 1 Ax
- 3 waxpby
- 2 ddot

1 node Xeon Silver 4116 2.10GHz

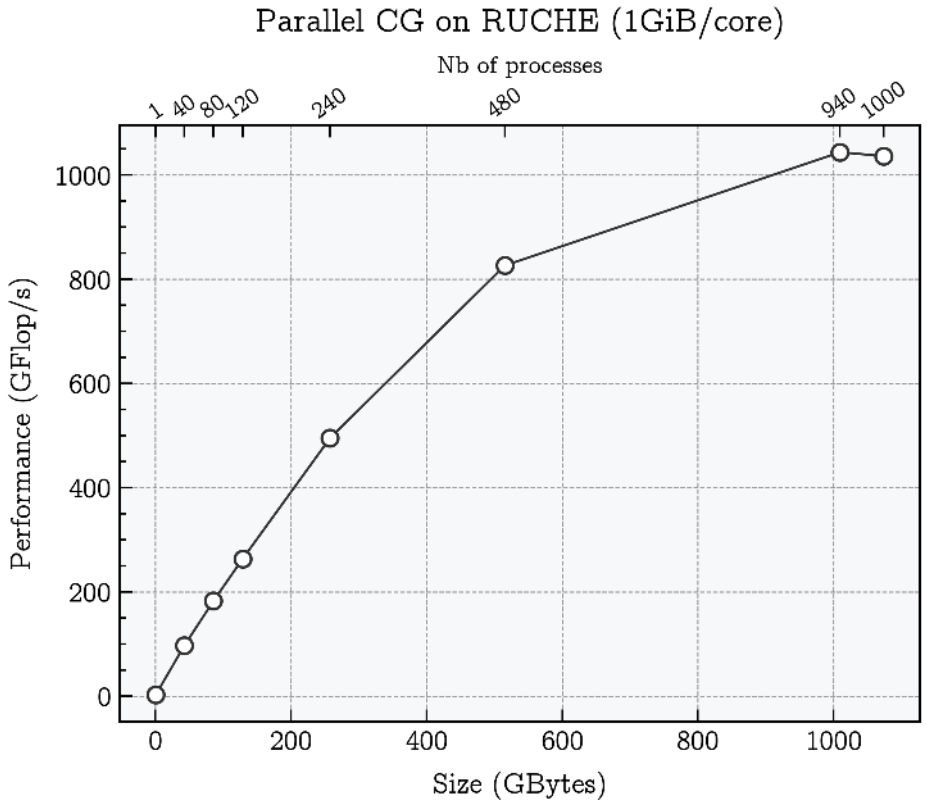


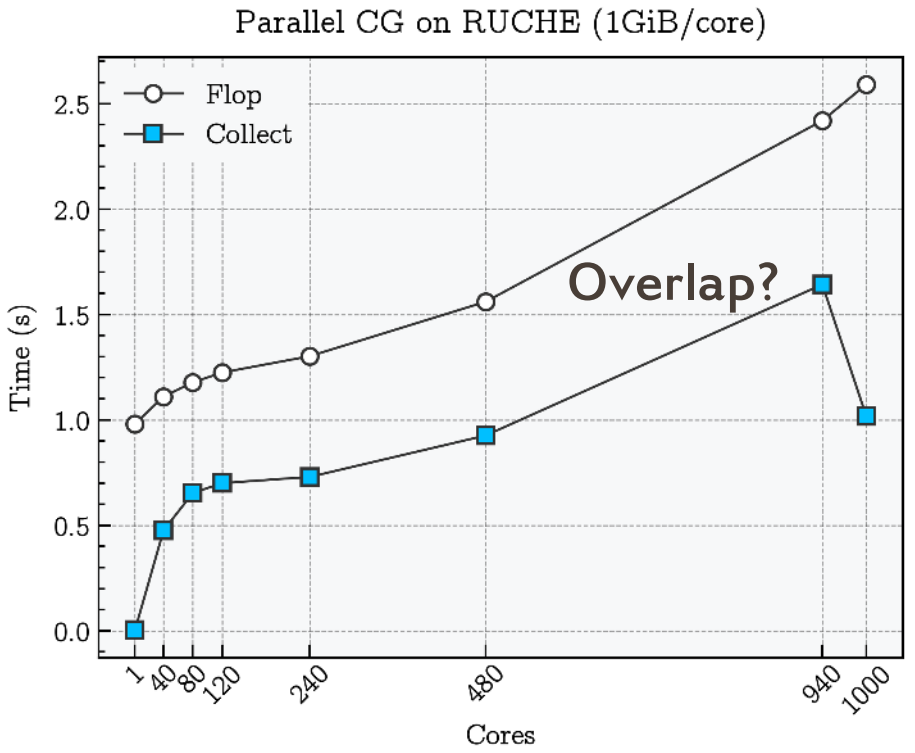




Benchmarks - Parallel Conjugate Gradient

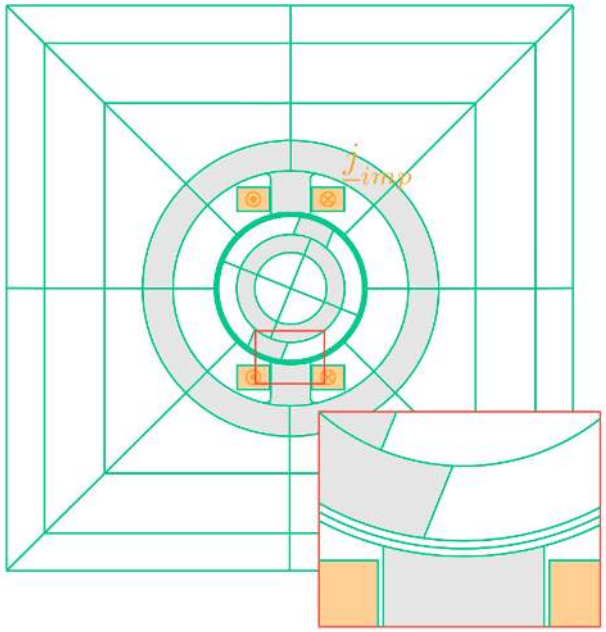
25 nodes Xeon Gold 6230 2.1GHz (RUCHE)

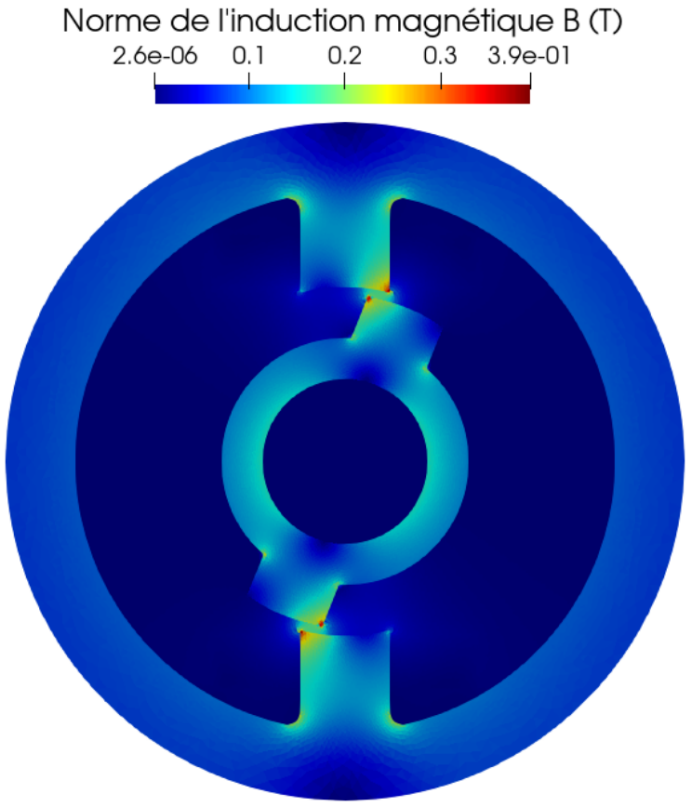




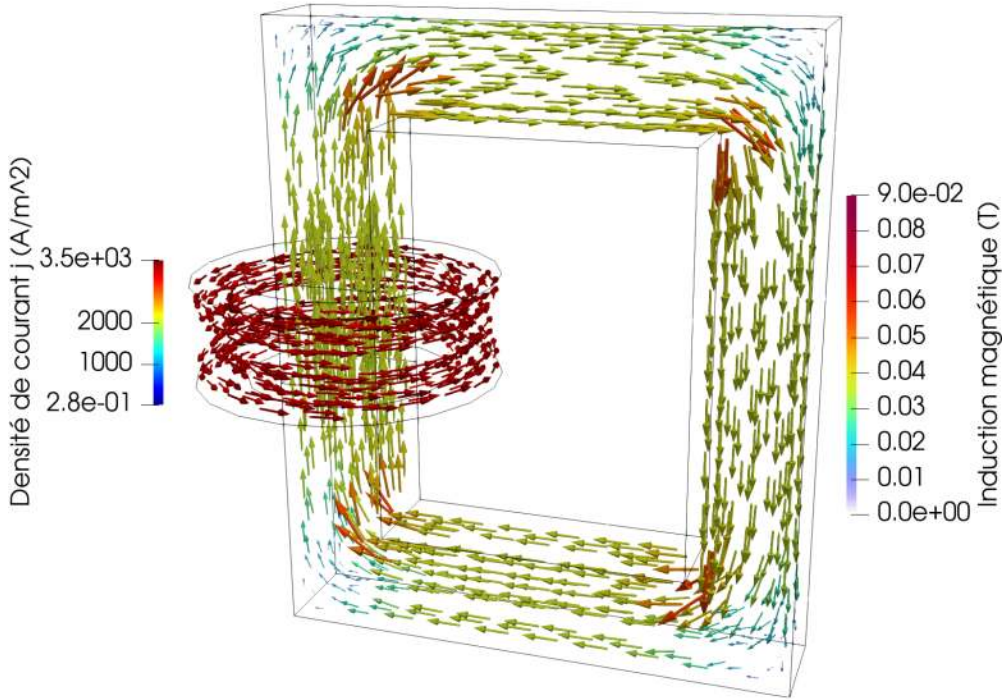
Application - ROMlab

Runs on RUCHE using MPI graph topology & Alltoallv between neighbors





2D Team24 problem on 62 sub-domains, 30.10^6 ddl



3D problem on 100 sub-domains, 2.10^6 ddl

Thank you!