Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Studying a new GPU treatment for chemical modules inside CAMP

Christian Guzman Ruiz, Mario C. Acosta, Guillermo Oyarzun, Matthew Dawson*, Oriol Jorba, Carlos Pérez García-Pando, Kim Serradell

Barcelona SuperComputing Center
***National Center for Atmospheric Research (NCAR)**

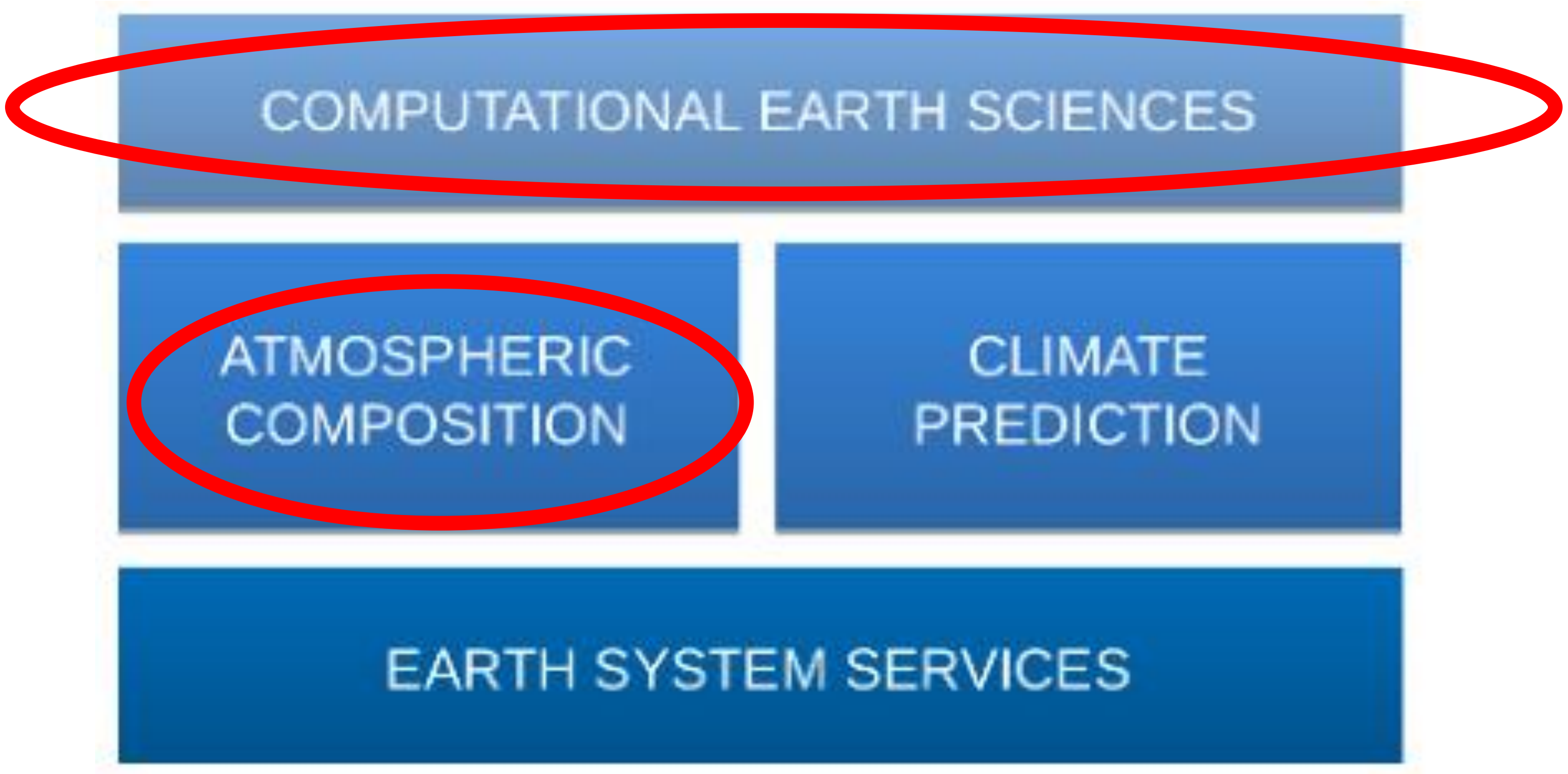
24/09/2021

19th workshop on HPC in meteorology

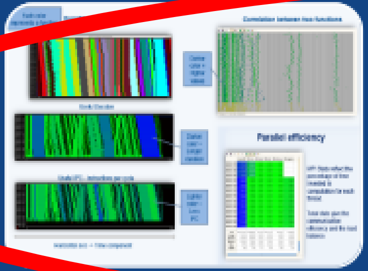
Outline

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Earth Sciences

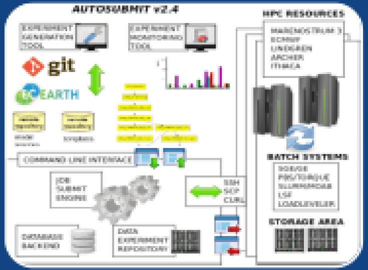


Computational Earth Science



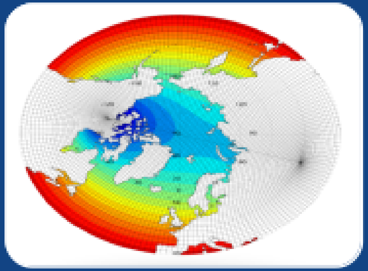
Performance Team

- Provide HPC Services (profiling, code audit, ...) to find main bottlenecks of our operational models
- Research and apply new computational methods for current and new platforms



Models and Workflows Team

- Development of HPC user-friendly software framework
- Support the development of atmospheric research software



Data and Diagnostics Team

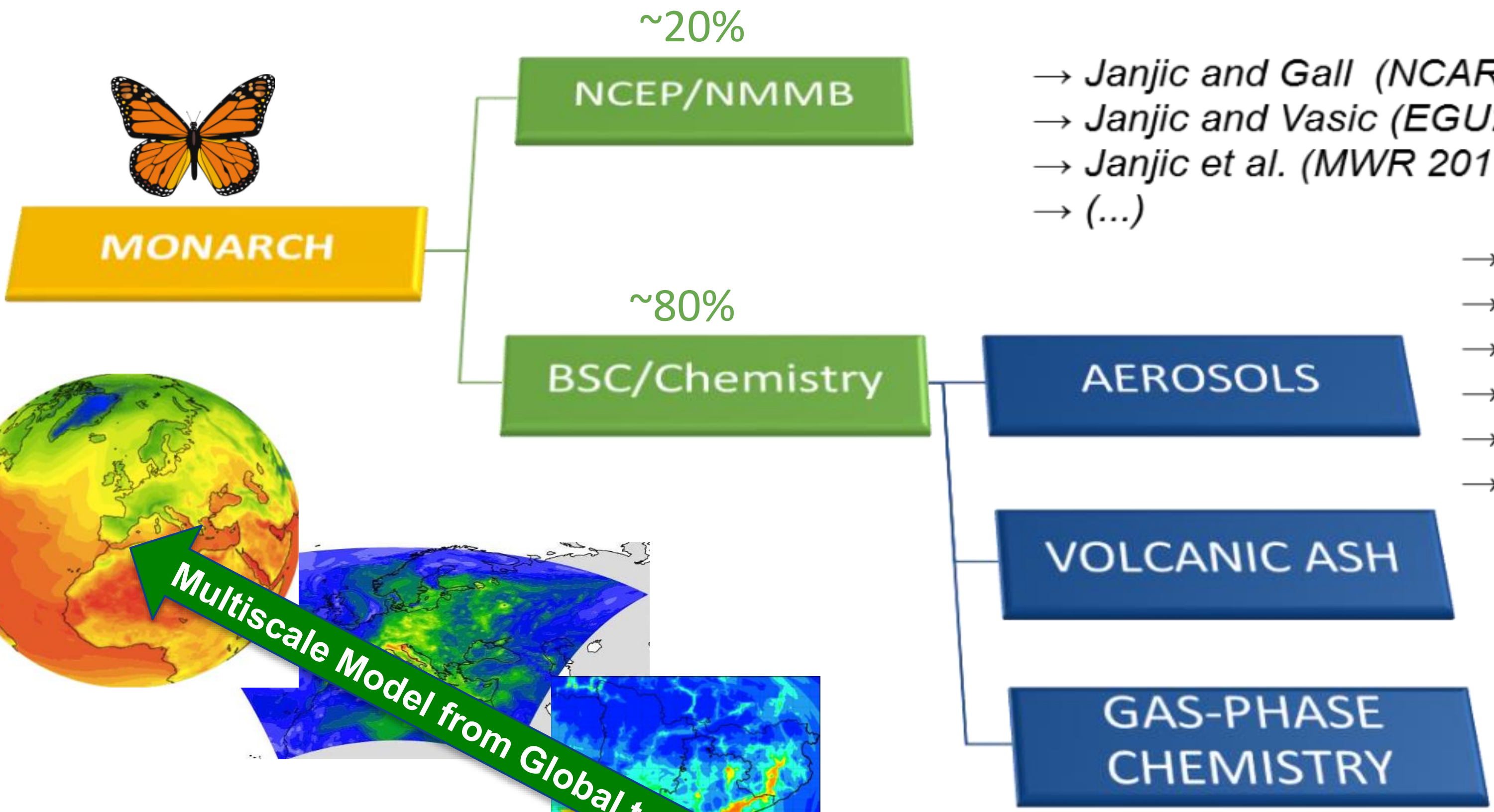
- Big Data in Earth Sciences
- Provision of data services
- Visualization

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Motivation

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

MONARCH: Multiscale On-line Atmosphere Chemistry Model

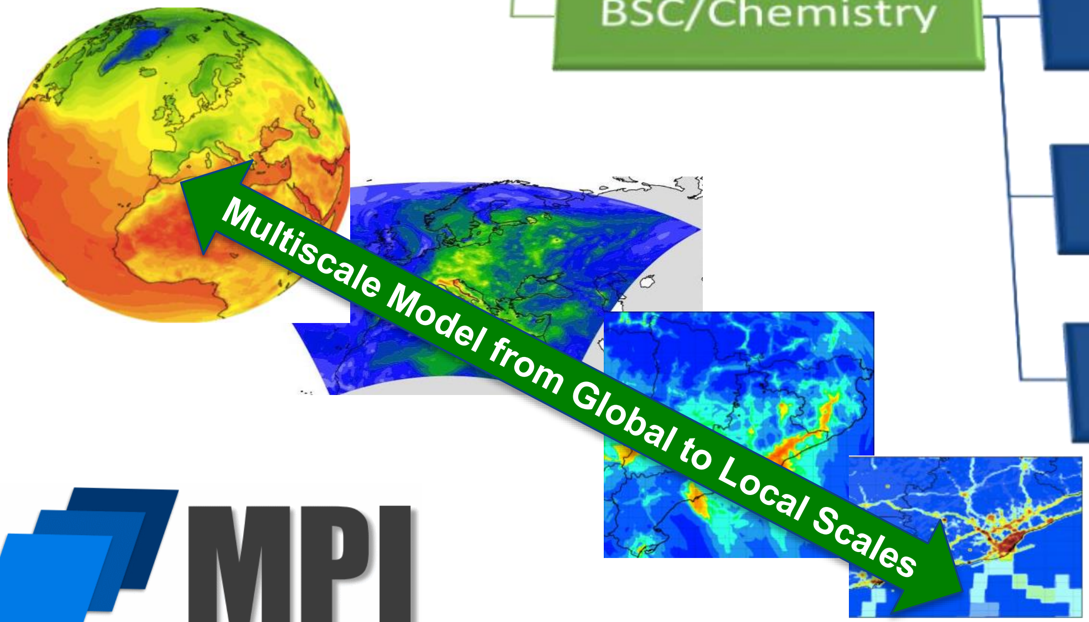


→ *Janjic and Gall (NCAR/TN 2012)*
→ *Janjic and Vasic (EGU2012)*
→ *Janjic et al. (MWR 2011)*
→ *(...)*

→ *Pérez et al. (ACP 2011)*
→ *Haustein et al. (ACP 2012)*
→ *Spada et al. (ACP 2013)*
→ *Spada et al. (AE 2014)*
→ *Spada (2015)*
→ *DiTomaso et al. (GMD 2017)*

→ *Martí et al. (ACP 2016)*

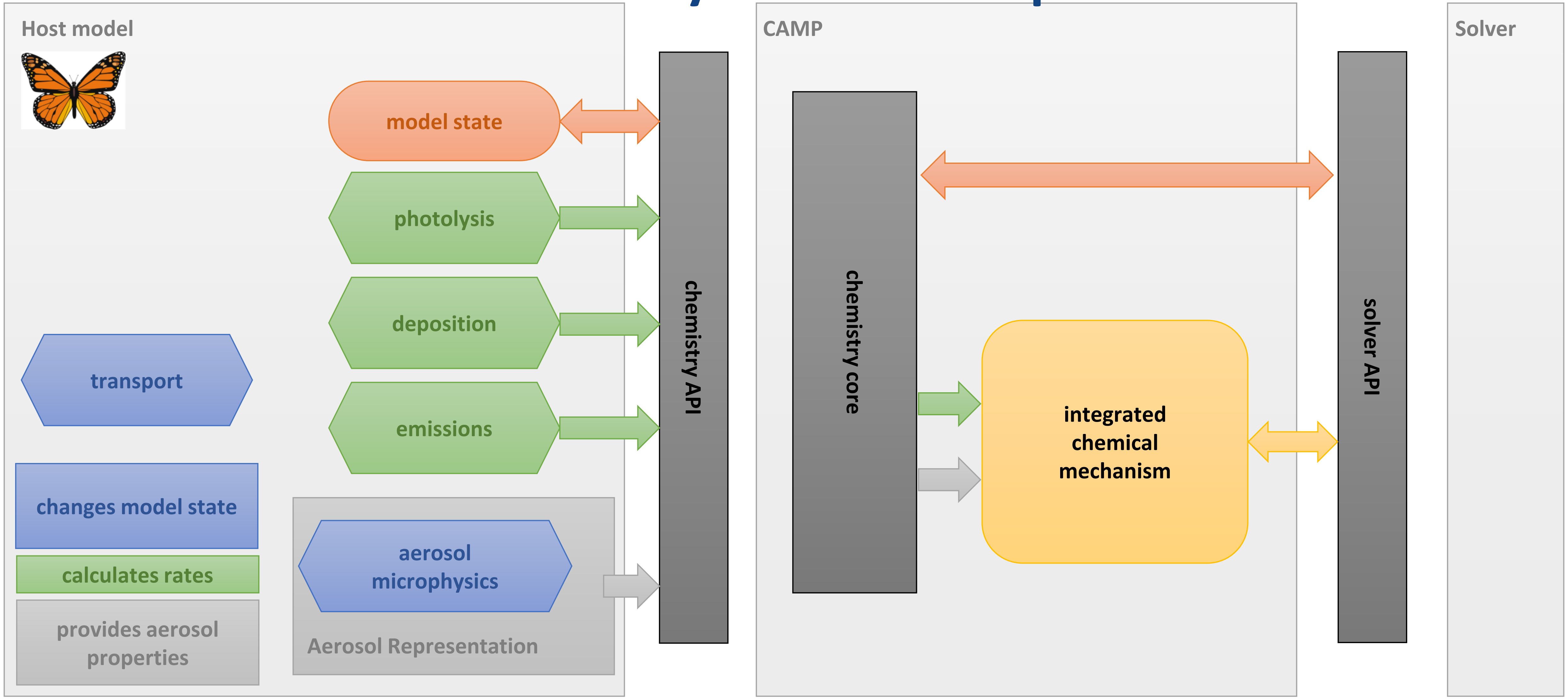
→ *Jorba et al. (JGR 2012)*
→ *Badia and Jorba (AE 2014)*
→ *Badia et al. (GMD 2017)*



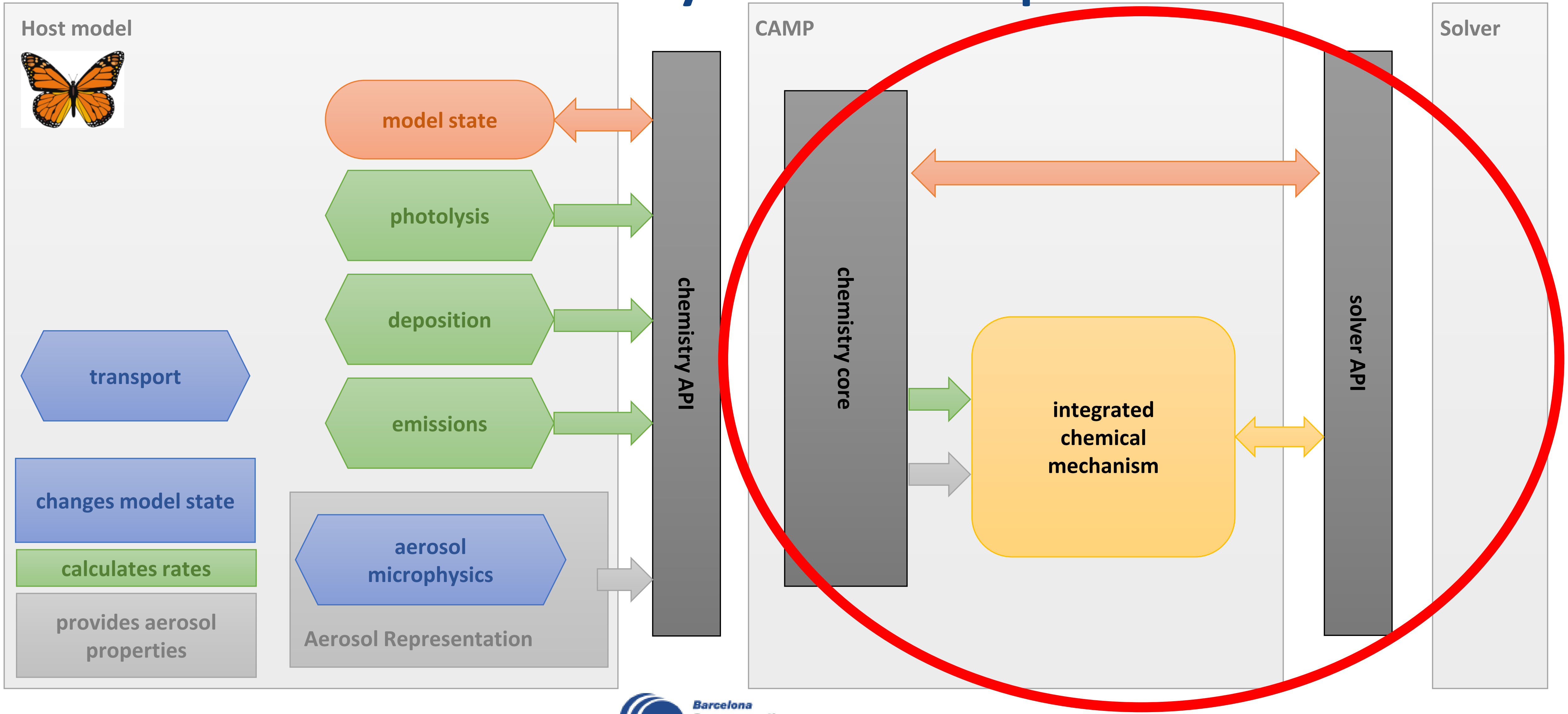
MPI

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

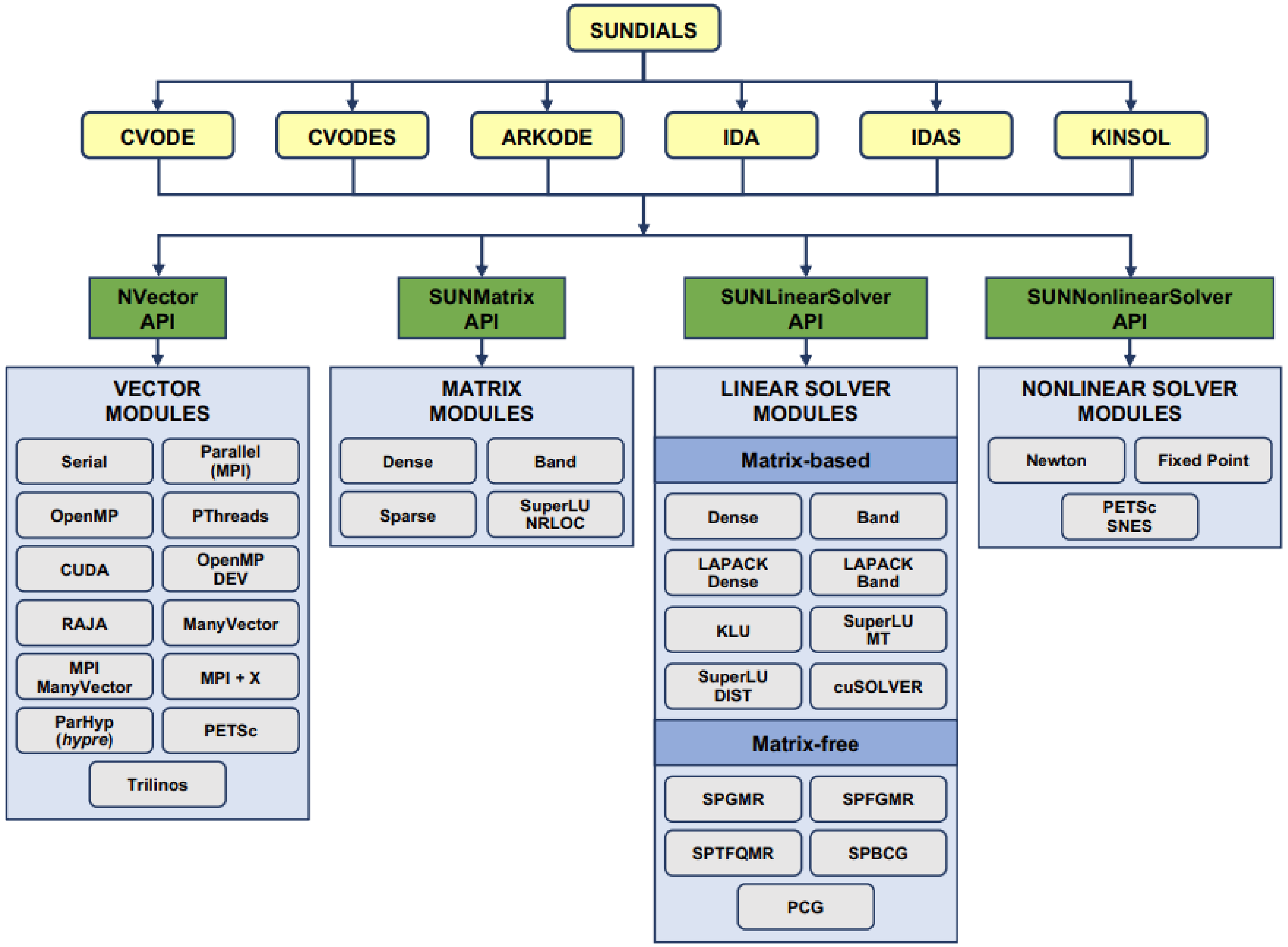
CAMP: Chemistry Across Multiple Phases



CAMP: Chemistry Across Multiple Phases

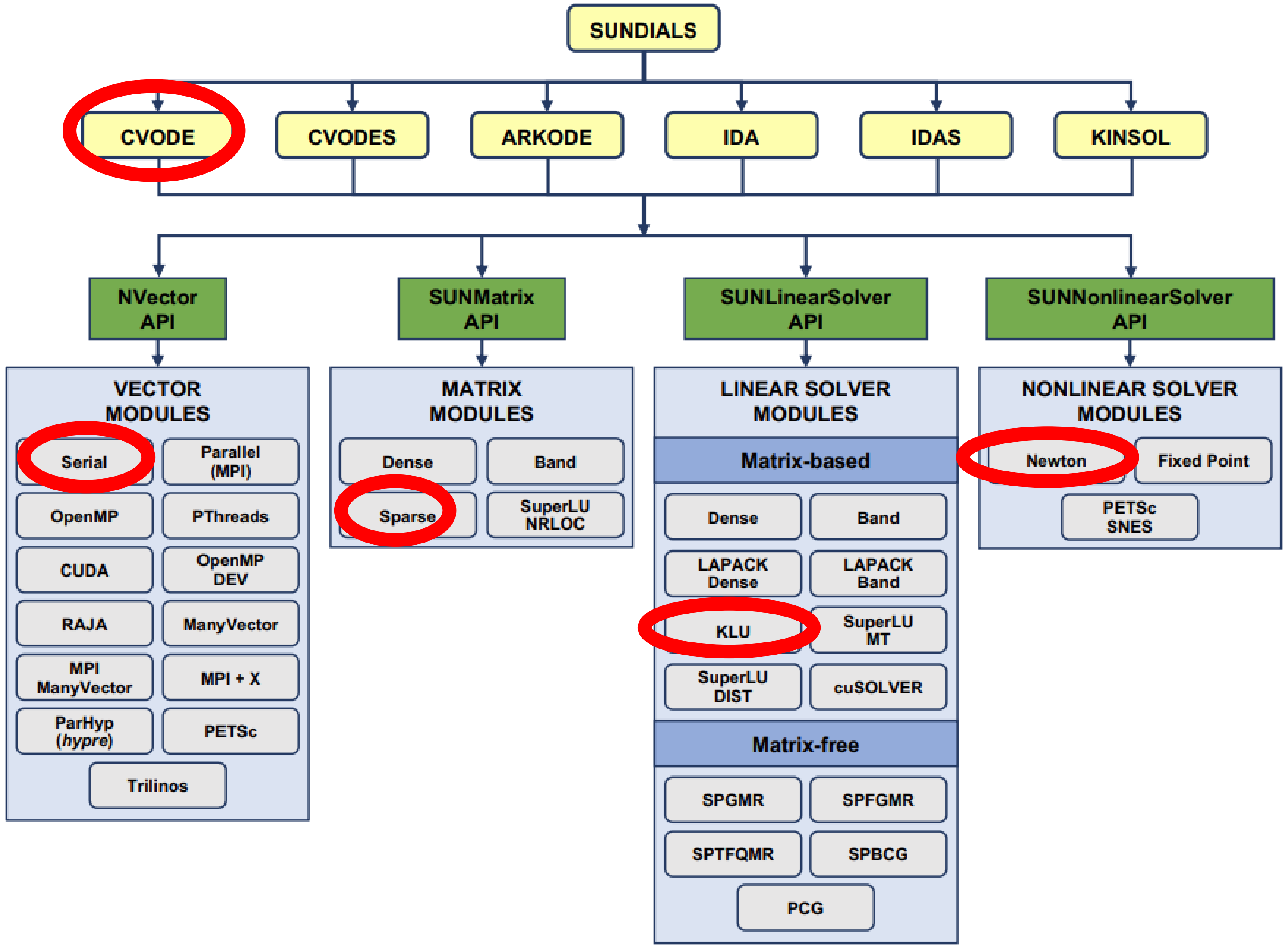


ODE Solvers



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

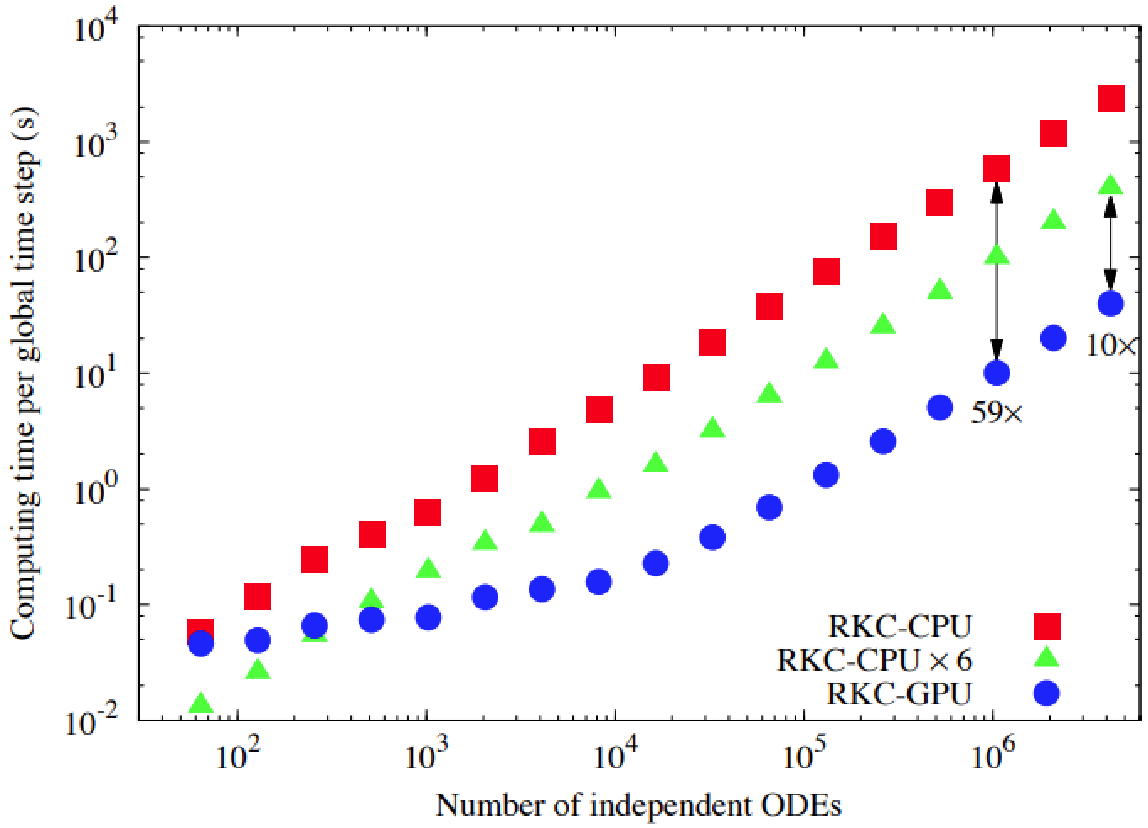
ODE Solvers


SUNDIALS
CVODE
CVODES
ARKODE
IDA
IDAS
KINSOL
NVector API
SUNMatrix API
SUNLinearSolver API
SUNNonlinearSolver API
VECTOR MODULES
Serial
Parallel (MPI)
OpenMP
PThreads
CUDA
OpenMP DEV
RAJA
ManyVector
MPI ManyVector
MPI + X
ParHyp (*hypre*)
PETSc
Trilinos
MATRIX MODULES
Dense
Band
Sparse
SuperLU NRLOC
LINEAR SOLVER MODULES
Matrix-based
Dense
Band
LAPACK Dense
LAPACK Band
KLU
SuperLU MT
SuperLU DIST
cuSOLVER
Matrix-free
SPGMR
SPFGMR
SPTFQMR
SPBCG
PCG
NONLINEAR SOLVER MODULES
Newton
Fixed Point
PETSc SNES

Related works

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Chemistry in the GPU: CUDA



Kyle E. Niemeyera,b,1, Chih-Jen Sungb, Accelerating moderately stiff chemical kinetics in reactive-flow simulations using GPUs, 2018

Configuration	Median CPU exec time (s)	Median accelerated exec time (s)	Performance over CPU
Intel Xeon X5650 + M2070	4.502	0.999	4.50×
Intel Xeon E5-2680 v3 + K80	1.476	0.283	5.21×
IBM POWER8 + P100	3.040	0.149	20.40×

Configuration	MPI Processes	CPU exec time (s)	Accelerated exec time (s)	Performance over CPU
2 × 6-core Intel Xeon X5650 +	2 MPI processes	5199	2358	2.27 ×
2 × NVIDIA M2070	12 MPI processes	1388	1368	1.01 ×
2 × 12-core Intel E5-2680 v3 +	4 MPI processes	7362	3384	2.17 ×
2 × NVIDIA K80	24 MPI processes	1756	1473	1.19 ×
2 × 10-core IBM POWER8 +	4 MPI processes	2294	918	2.50 ×
4 × NVIDIA P100	20 MPI Processes	814	437	1.86 ×

Michail Alvanos and Theodoros Christoudia, GPU-accelerated atmospheric chemical kinetics in the ECHAM/MESSy (EMAC) Earth system model , 2017

...and more

Our contribution

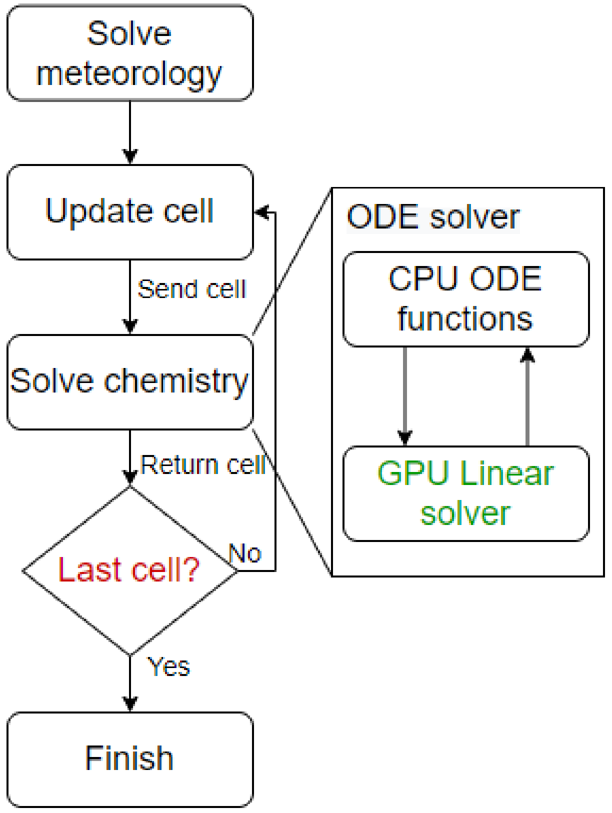
- Development on CAMP
- Allow integration of GPU kernels (as a library) without developing an entire chemical module to GPU
  - A novel approach computing multiple cells simultaneously
- Performance evaluation of GPU algorithms over CPU
  - Linear solver scheme
  - Adapted to avoid unnecessary delays with synchronization between GPU blocks

Multi-cells strategy

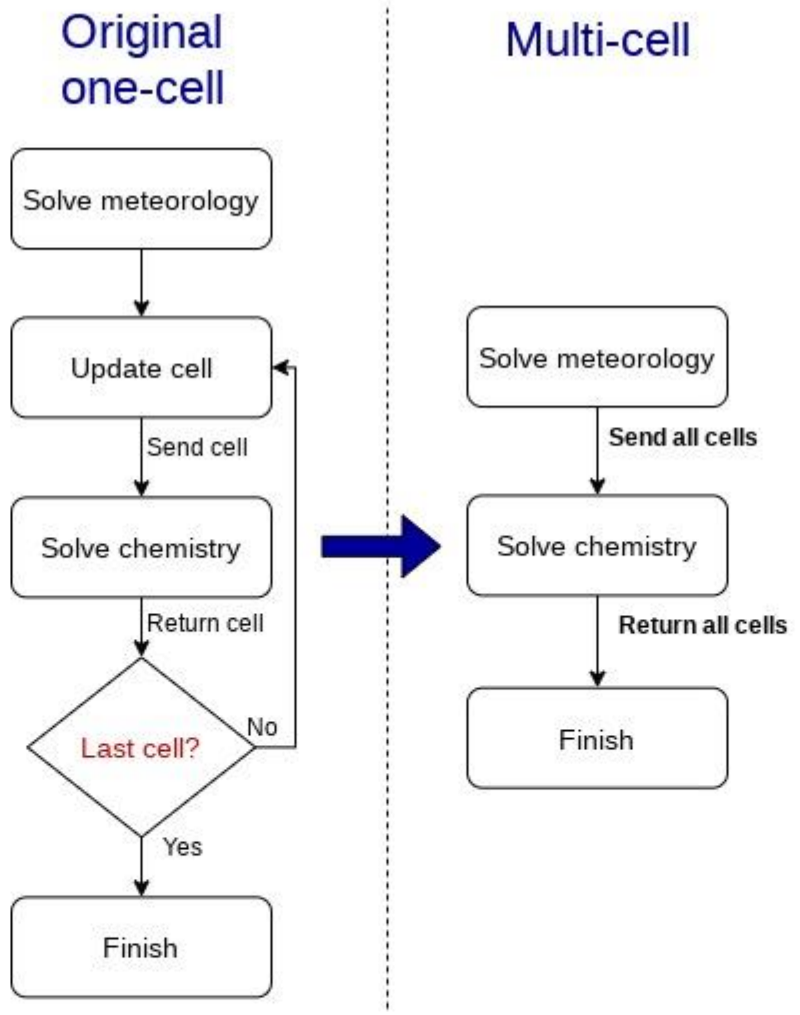
Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Multi-cells

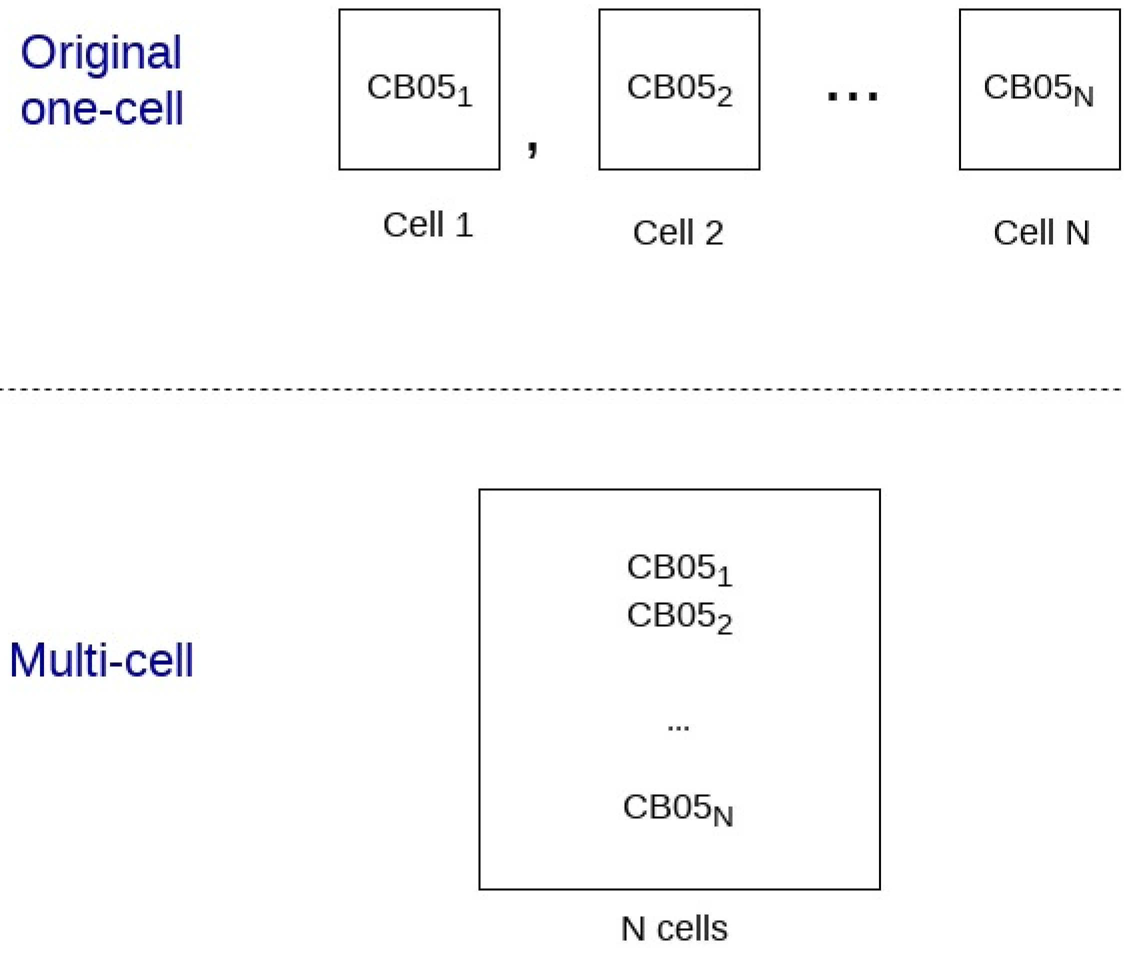
- **Background:**
  - **Domain decomposition** results in multiple cells per MPI process, computed in a loop
  - These cells can be solved in parallel with the GPU
  - A GPU kernel needs to receive all the cells at once in order to compute them simultaneously
- **Our approach:**
  - Adapt the chemical module to solve multiple cells in the base CPU single thread implementation



Multi-cells

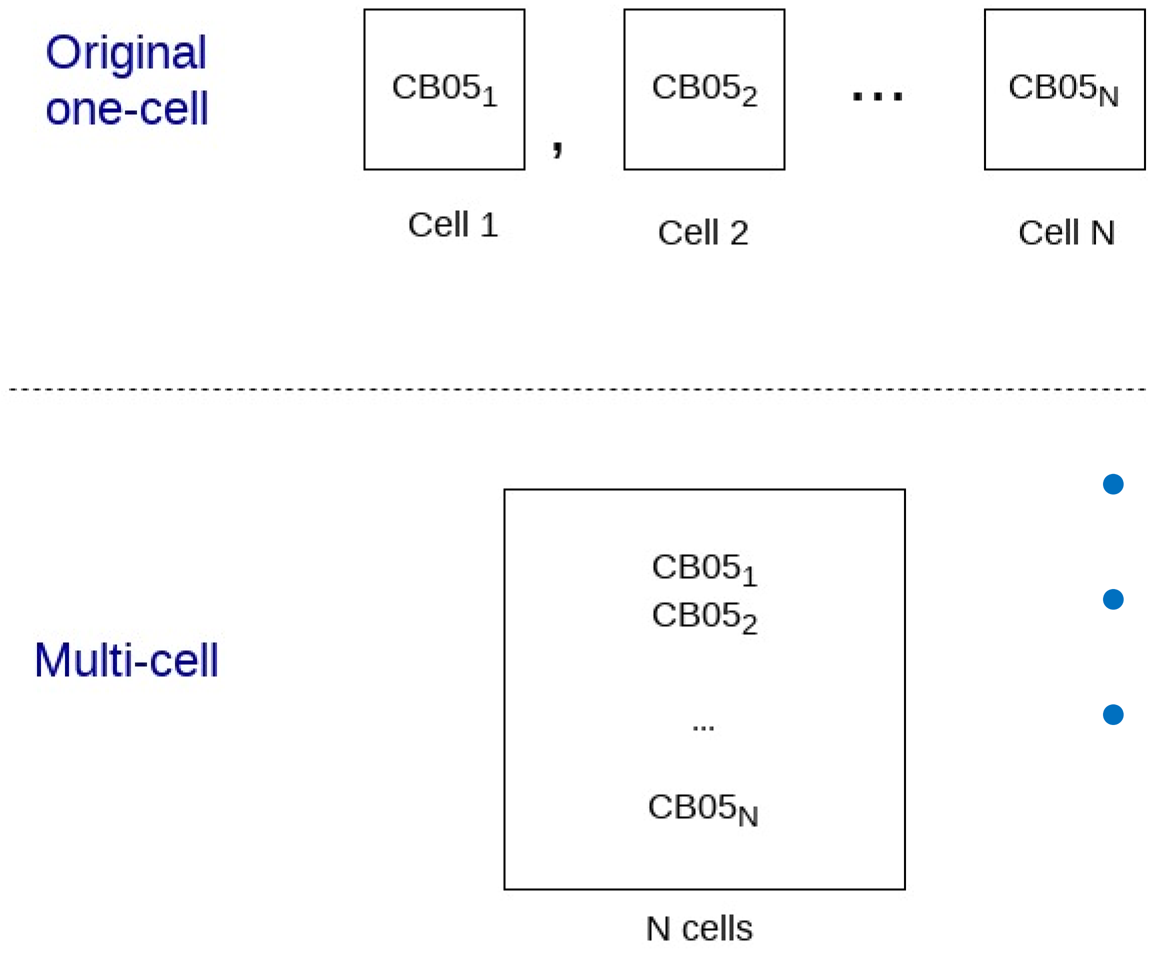

Original one-cell
Multi-cell
Solve meteorology
Update cell
Send cell
Solve chemistry
Return cell
Last cell?
No
Yes
Finish
Solve meteorology
Send all cells
Solve chemistry
Return all cells
Finish

Multi-cells



Group cells data

Multi-cells



- Species "replication": $O3_1$, $O3_2$.. O_N
- Common ODE solver parameters.
- More sensitive to different convergence speeds.

GPU Linear solving

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Algorithms

- KLU Sparse (CPU)
  - Integrated in CVODE
  - Dependency on CVODE and SuiteSparse routines
  - **Direct solver**

- Biconjutage Gradient (GPU)
  - Developed in house.
  - Based from CUBLAS library.
  - **Iterative solver**

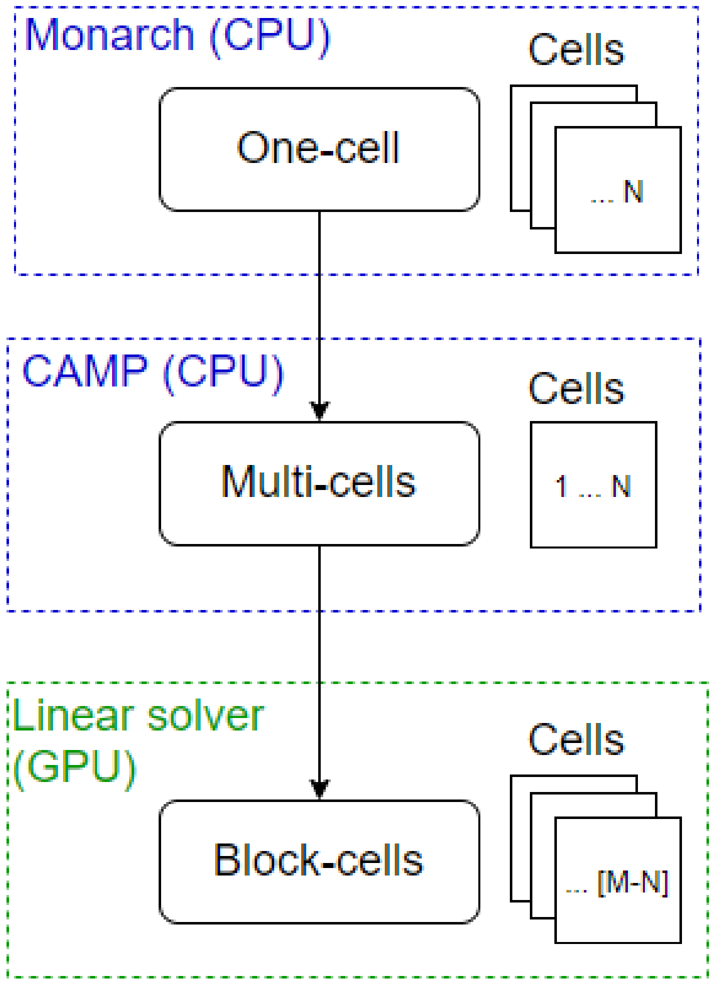
GPU Linear solving

- **Background:**
  - Solve a big linear system composed of multiple cells
  - Each cell contains multiple unknowns (species), solved by one-thread each (Number of Threads == Number of Species)
  - Base algorithm solves a single system, working over all the present unknowns (e.g. reducing an array to a single variable)
  - Expensive communication between GPU blocks (>= 50% overall execution time)

- **Our approach:**
  - Compute cells independently of the others

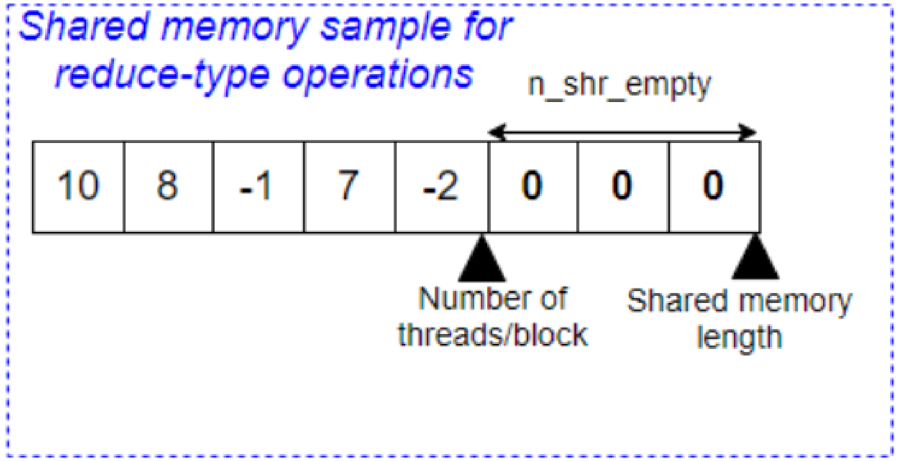
GPU Block-cells

- One-cell and Multi-cells are designed for CPU
  - One-cell computes the cells independently of each other
  - Multi-cells groups the cells into an unique system to solve
- Block-cells is used in GPU to avoid communication between GPU blocks
  - Cells are divided in blocks, containing X cells each block
  - The number of cells per block depends on the number of threads per block available in the GPU



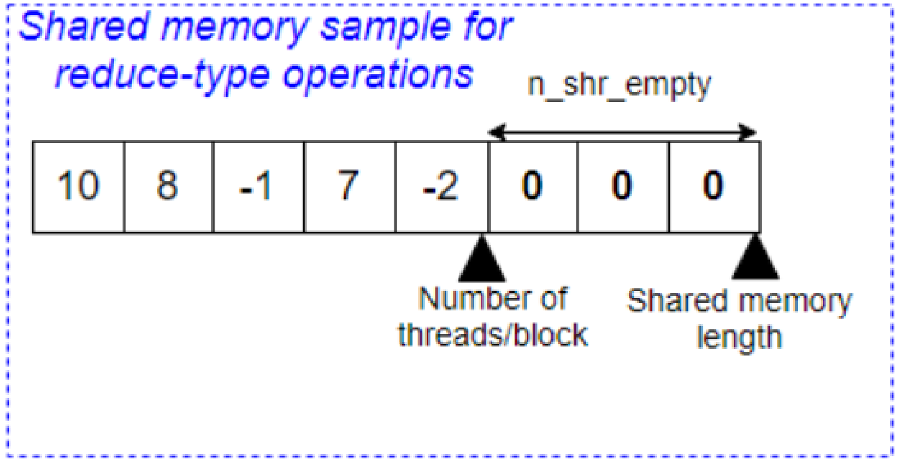
Kernel configuration: GPU blocks cells (N)

- **GPU Block contains multiple cells**
  - Chemical mechanism ⇨ 156 species (CB05)
  - GPU block ⇨ Max. threads/block (1024)
  - Cells per block ⇨ 1024/156 = 6.56 -> 6
  - Shared memory ⇨ 1024 variables (Empty slots from 936 to 1024)



Kernel configuration: GPU Block-cells (1)

- **GPU Block contains 1 cell**
  - Chemical mechanism ⇨ 156 species (CB05)
  - GPU block ⇨ 156 threads
  - Shared memory ⇨ 256 variables (Empty slots from 157 to 256)



Hardware and Software configuration

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Hardware

- **CTE-POWER cluster:**
  - **2 x IBM Power9** 8335-GTH @ 2.4GHz (3.0GHz on turbo, **20 cores** and 4 threads/core, total 160 threads per node**)**
  - 512GB of main memory distributed in 16 dimms x 32GB @ 2666MHz
  - 2 x SSD 1.9TB as local storage
  - **4 x GPU NVIDIA V100 (Volta) with 16GB HBM2.**
  - Compilers: GCC version 6.4.0 and NVCC version 9.1

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Software configuration

*Chemical mechanism	Species	Cells	Time-steps
CB05+chlorine	156	100 - 10,000	5

- *Based in a configuration for realistic tests for atmospheric models taken from CAMP paper (publication in progress)

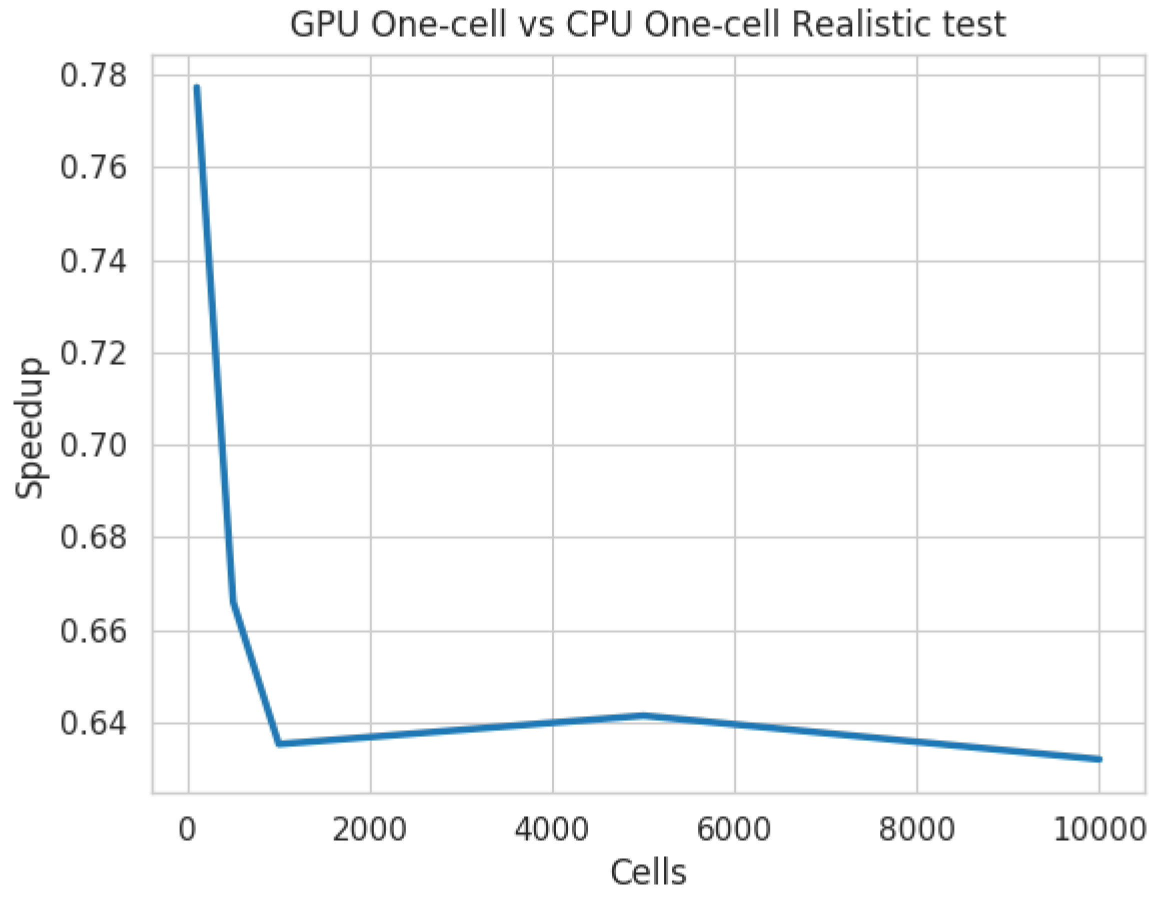
Kernel setting	Linear solver	Threads per block	Cells per block	Shared memory
GPU Block (1)	BCG	156	1	256
GPU Block (N)	BCG	1024	6	1024

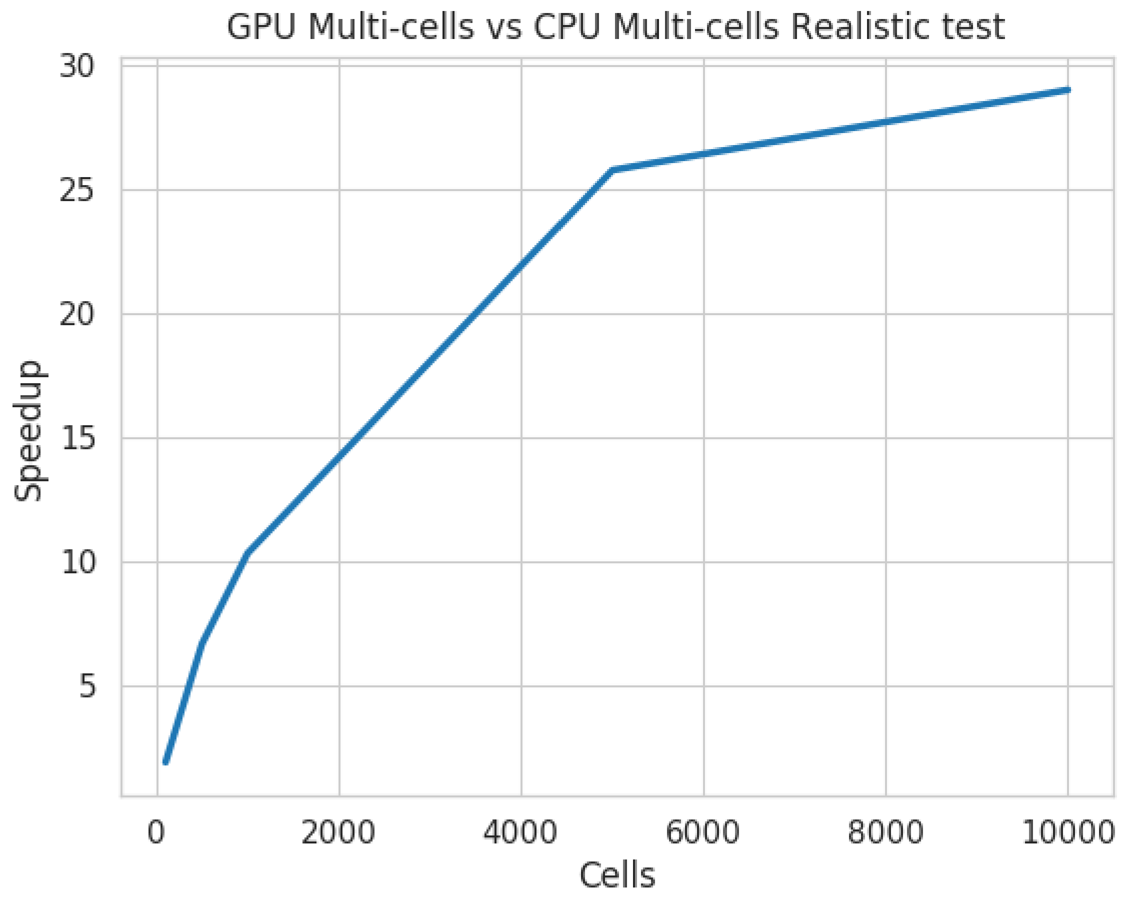
Linear solver	MPI processes	GPUs	Method
KLU	1, 40	0	One-cell, Multi-cells
BCG	1	1	One-cell, Multi-cells, Block-cells

Results

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

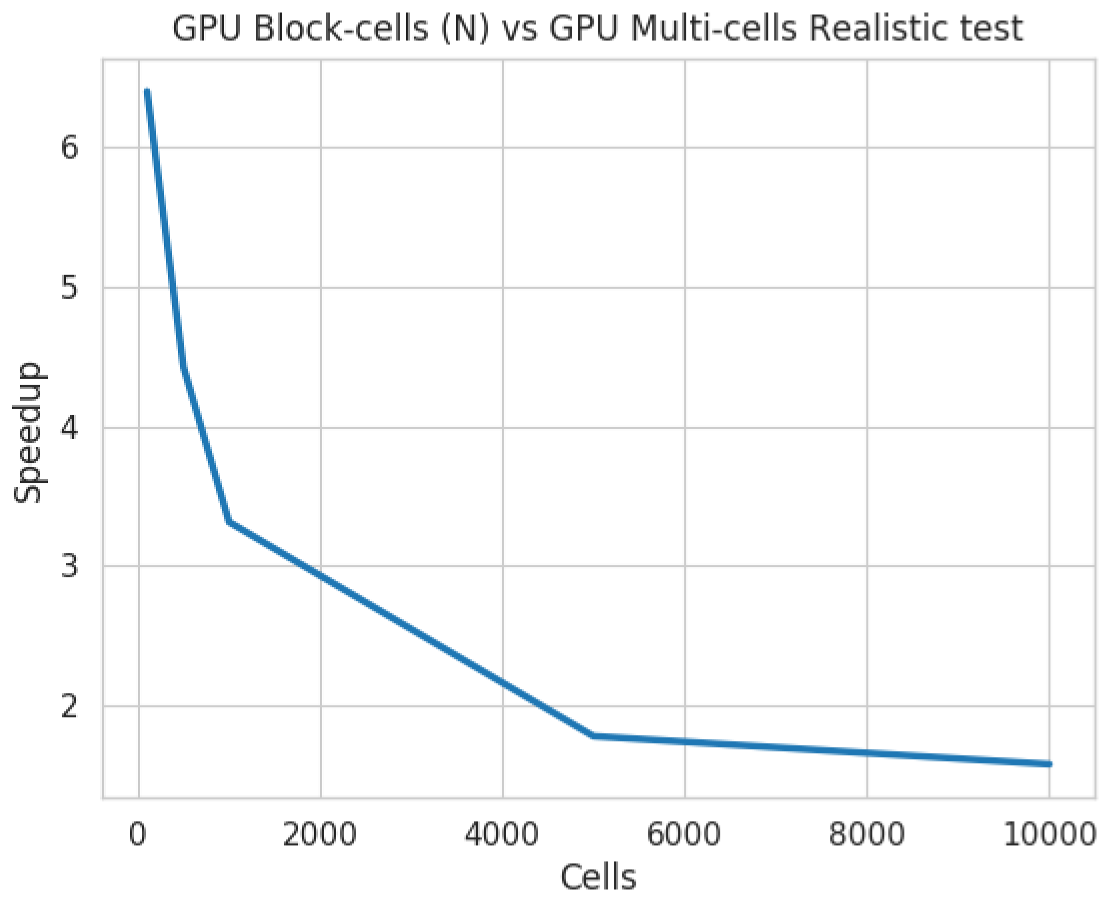
Results

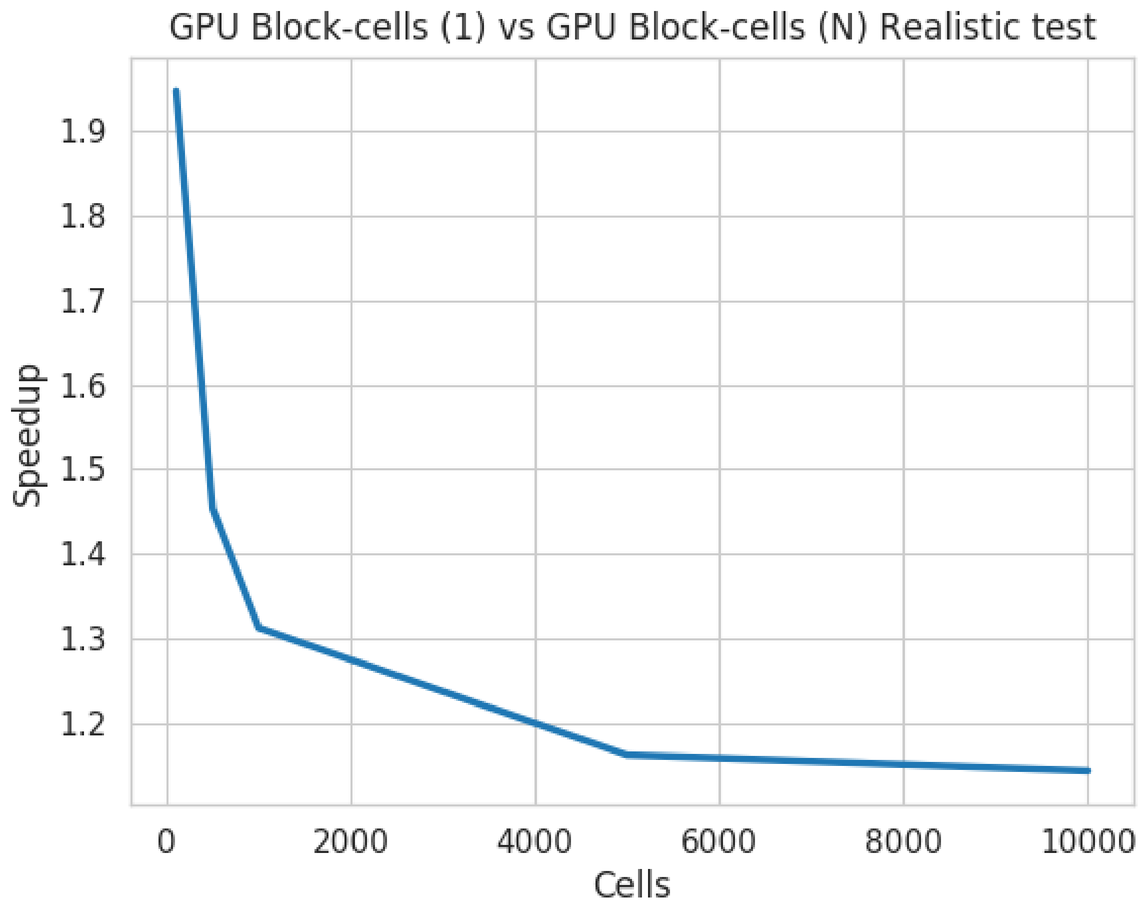




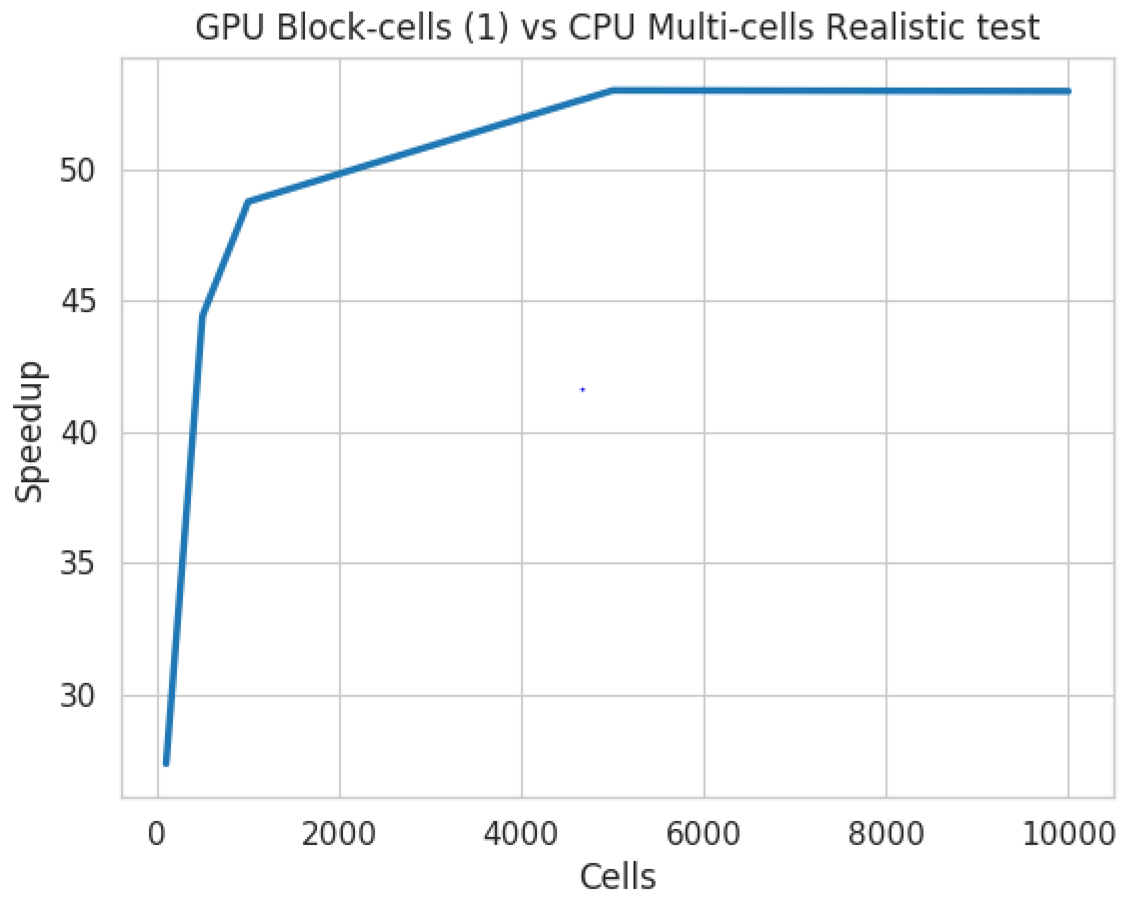
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

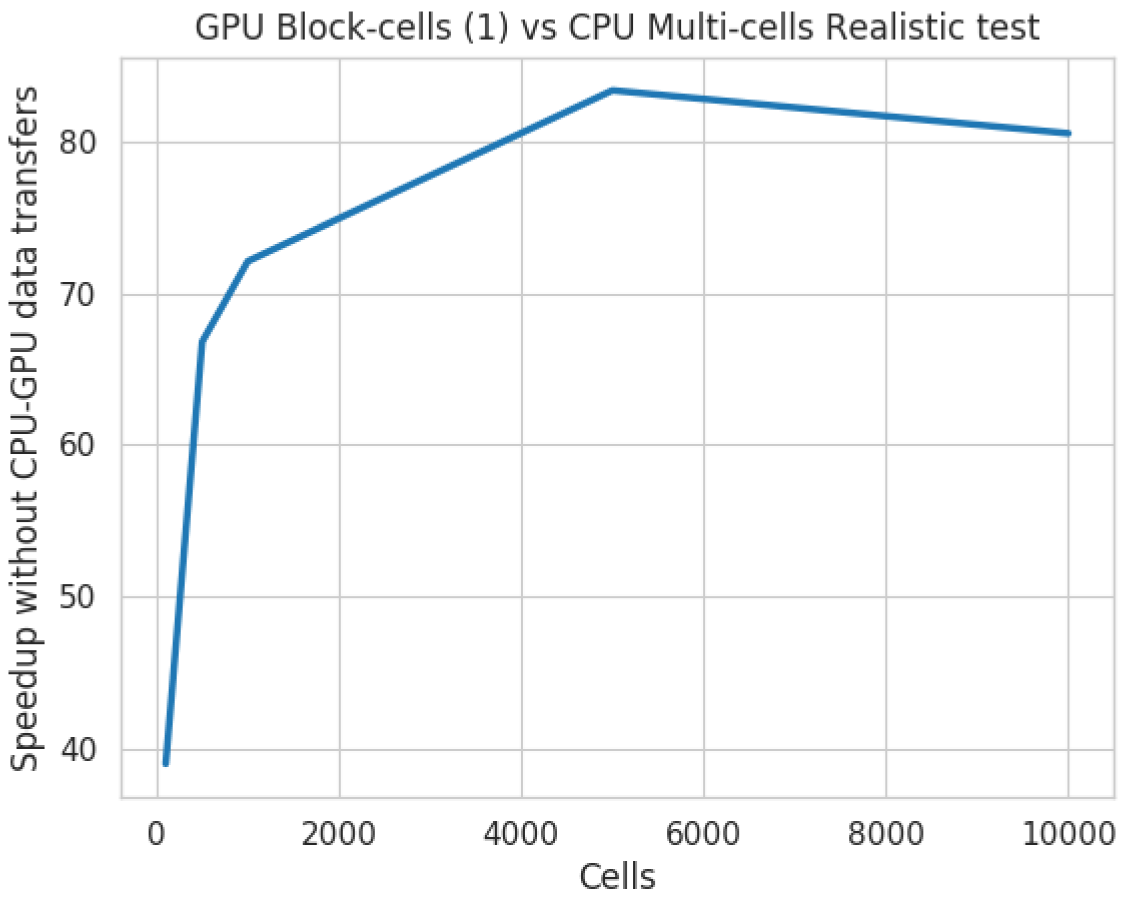
Results


GPU Block-cells (N) vs GPU Multi-cells Realistic test
Speedup
Cells
2
3
4
5
6
0
2000
4000
6000
8000
10000

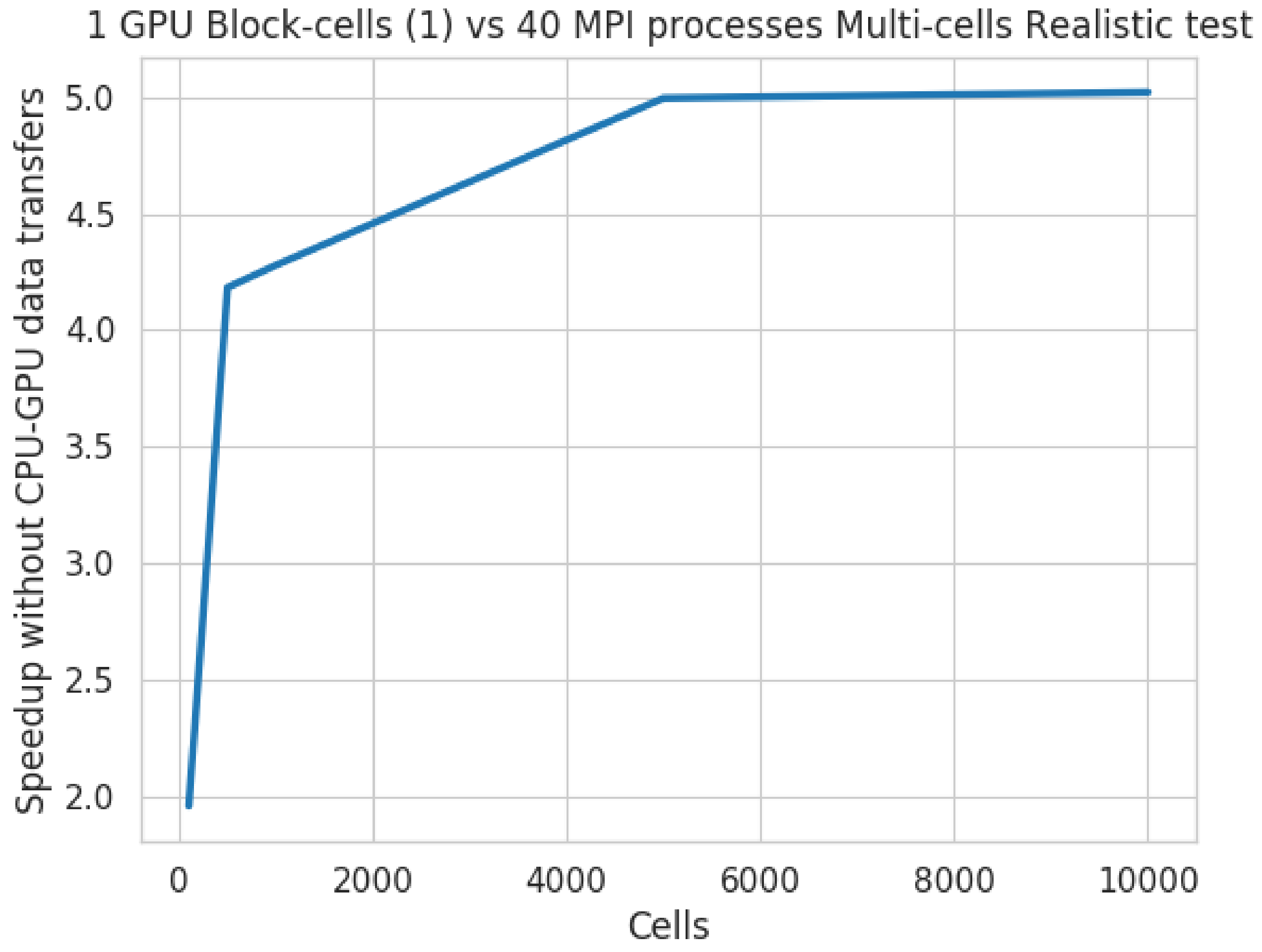

GPU Block-cells (1) vs GPU Block-cells (N) Realistic test
Speedup
Cells
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
0
2000
4000
6000
8000
10000

Results


GPU Block-cells (1) vs CPU Multi-cells Realistic test
Speedup
30
35
40
45
50
0
2000
4000
6000
8000
10000
Cells


GPU Block-cells (1) vs CPU Multi-cells Realistic test
Speedup without CPU-GPU data transfers
40
50
60
70
80
0
2000
4000
6000
8000
10000
Cells

Results


1 GPU Block-cells (1) vs 40 MPI processes Multi-cells Realistic test
Speedup without CPU-GPU data transfers
5.0
4.5
4.0
3.5
3.0
2.5
2.0
0
2000
4000
6000
8000
10000
Cells

Conclusions

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Conclusions

- Adapted CAMP to integrate GPU libraries, facilitating the development of a GPU chemical solver in the future

- Added a GPU linear solver developed in house, improving the performance significantly from the CPU single-thread version (up to 50x speedup, 80x without CPU-GPU data transfers) and 5x times faster than using all the MPI process available

- The Multi-cells strategy allows the integration of GPU modules inside a CPU solver

- The best kernel settings for the Block-cells strategy is using 1 cell per GPU block, achieving from 1x to 3x speedup from computing multiple cells in a block.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Next steps

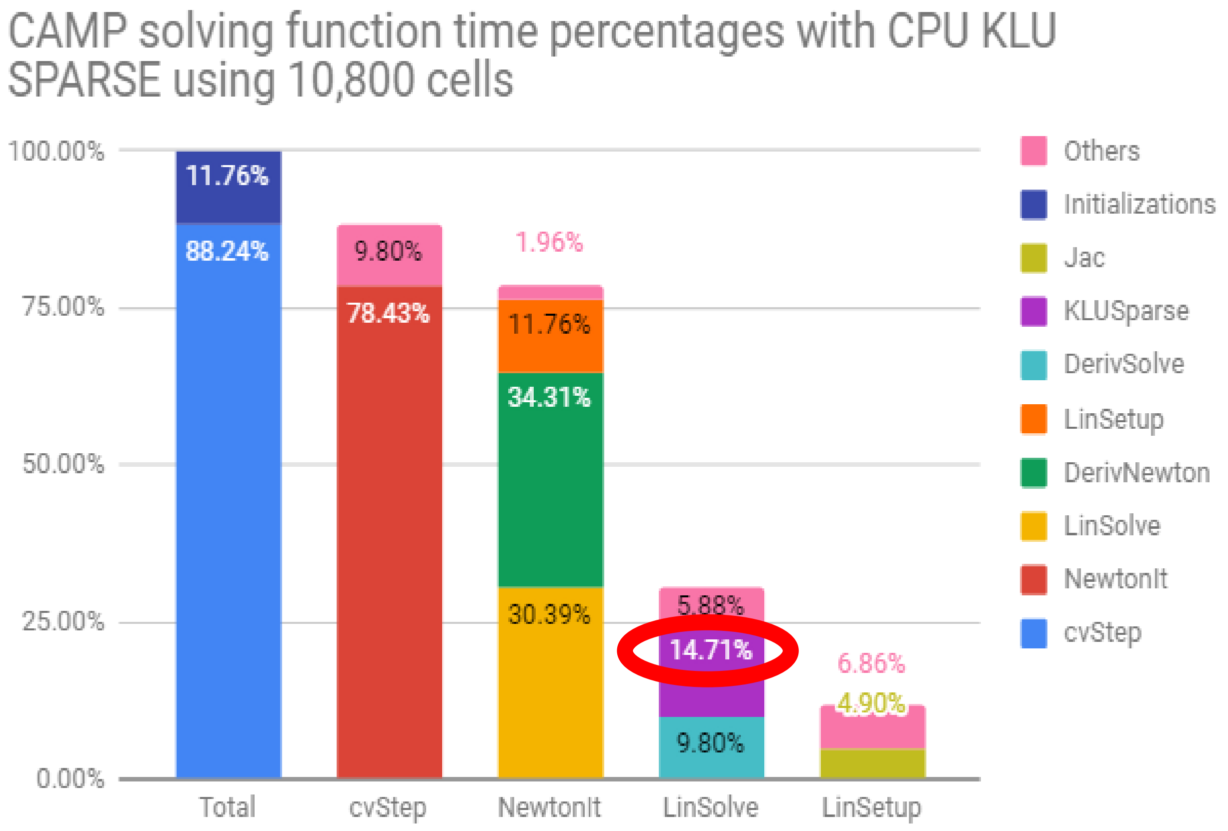
- Use the 4 GPUs available
- Translate the rest of the chemical solving to GPU (minimizing communications CPU-GPU)
- Evaluate more chemical configurations
- Integrate into MONARCH (exploit an heterogeneous approach)

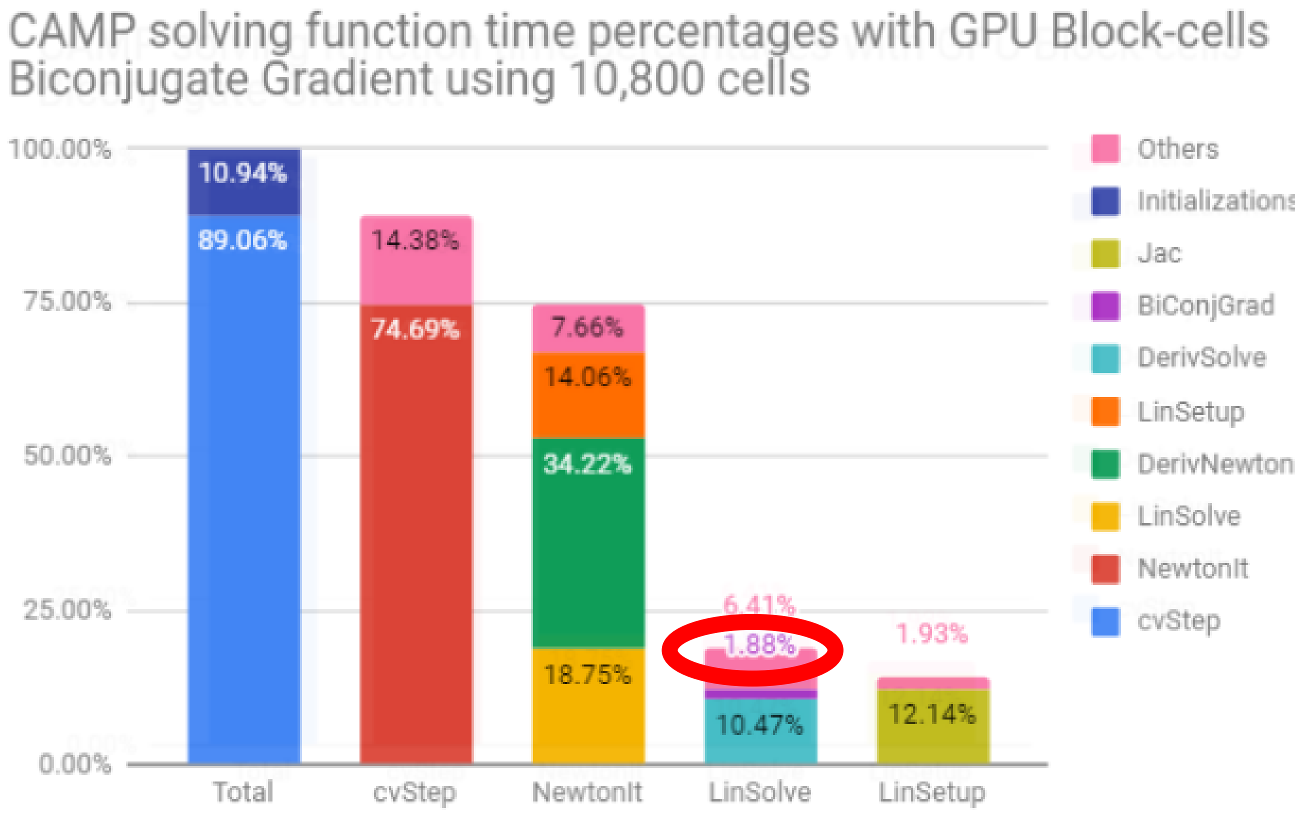
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you

christian.guzman@bsc.es

Performance improvement over CAMP


CAMP solving function time percentages with CPU KLU SPARSE using 10,800 cells
Total
cvStep
NewtonIt
LinSolve
LinSetup
Others
Initializations
Jac
KLUSparse
DerivSolve
LinSetup
DerivNewton
LinSolve
NewtonIt
cvStep


CAMP solving function time percentages with GPU Block-cells Biconjugate Gradient using 10,800 cells
Total
cvStep
NewtonIt
LinSolve
LinSetup
Others
Initializations
Jac
BiConjGrad
DerivSolve
LinSetup
DerivNewton
LinSolve
NewtonIt
cvStep