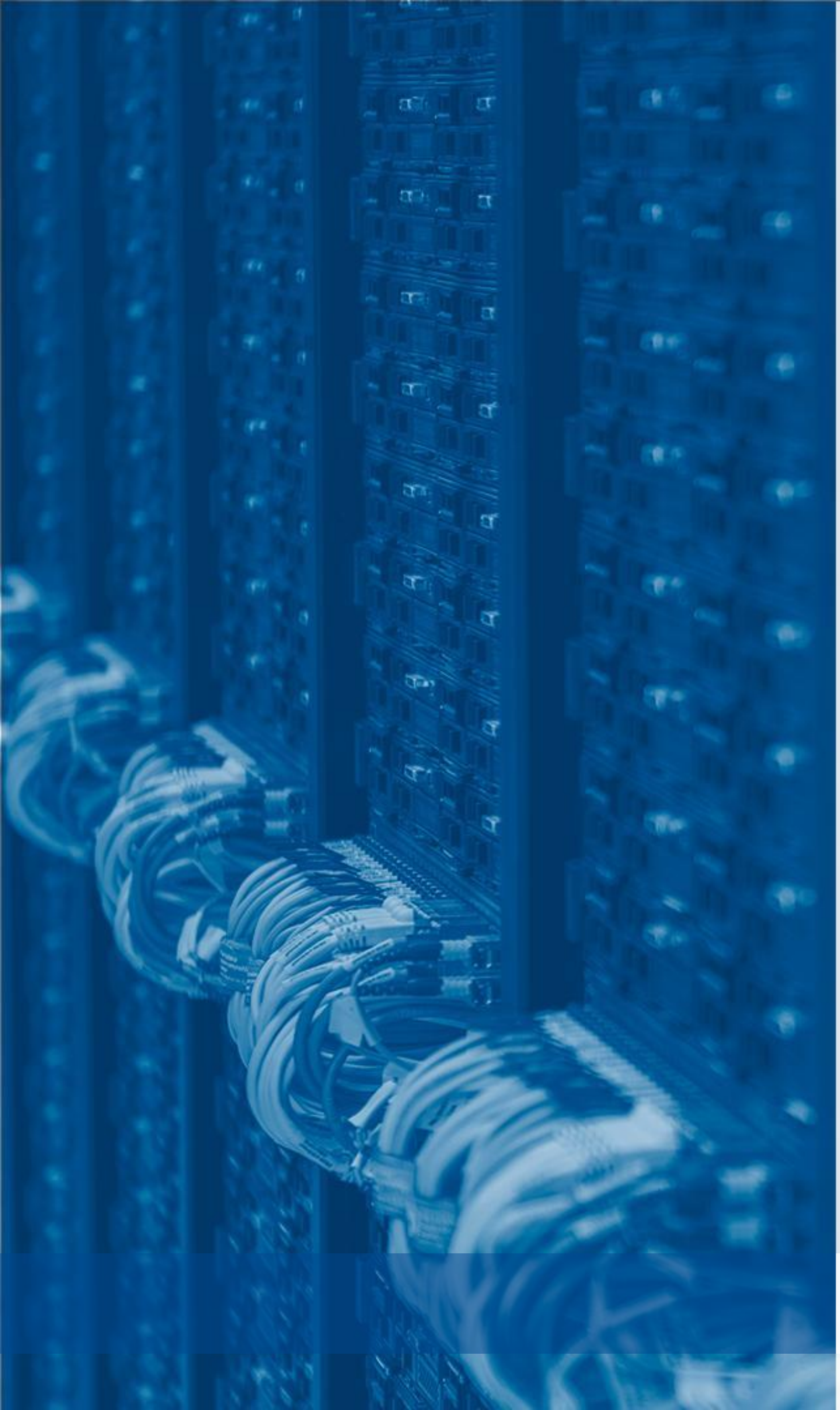


Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

CAMP First GPU Solver: A Solution to Accelerate Chemistry in Atmospheric Models

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

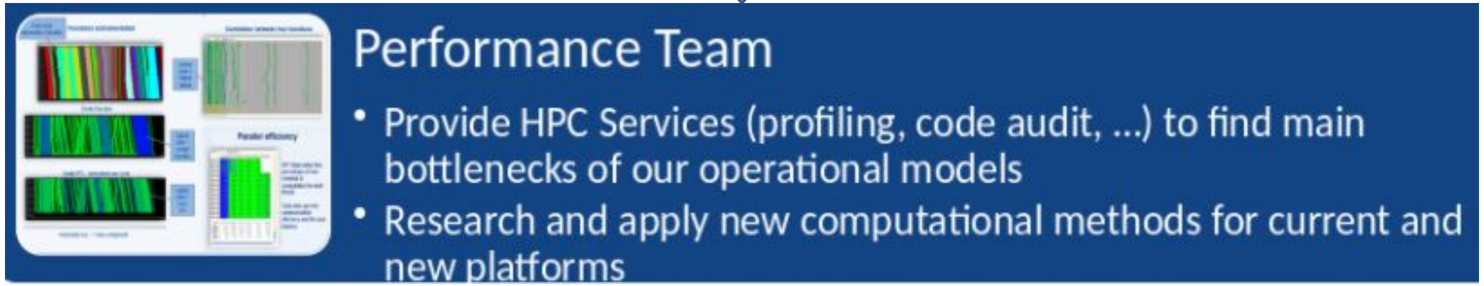
Barcelona SuperComputing Center
***National Center for Atmospheric Research (NCAR)**

7th ENES workshop

BSC Departments

Earth Sciences

To develop and implement global and regional state-of-the-art models for short-term air quality forecast and long-term climate applications



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

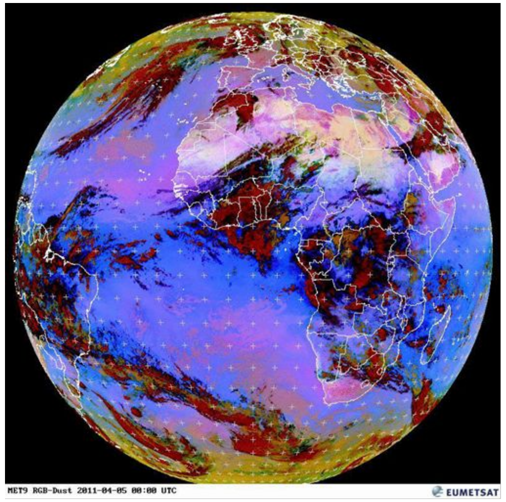
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

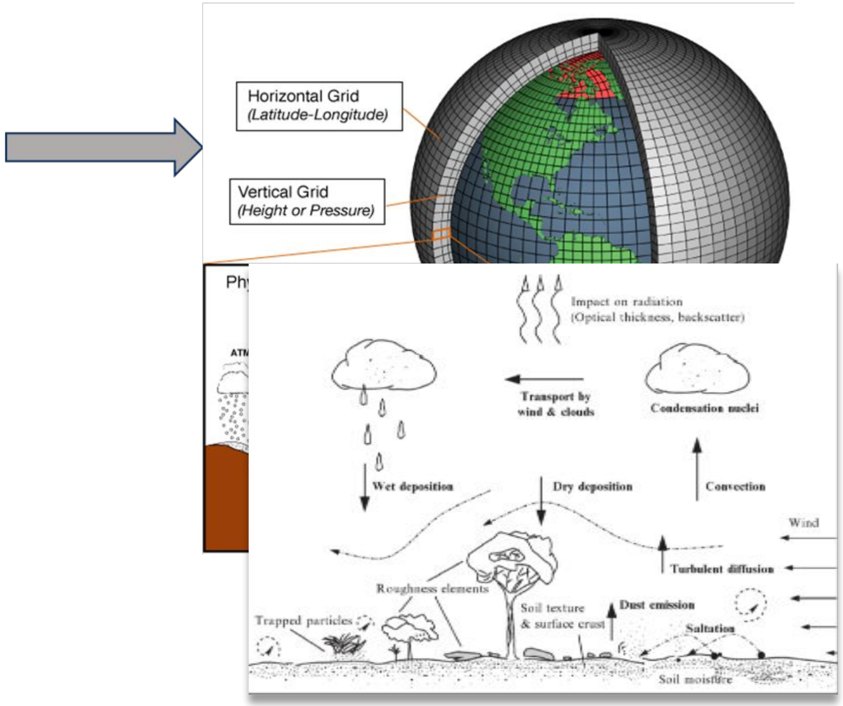
Background

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Atmospheric models

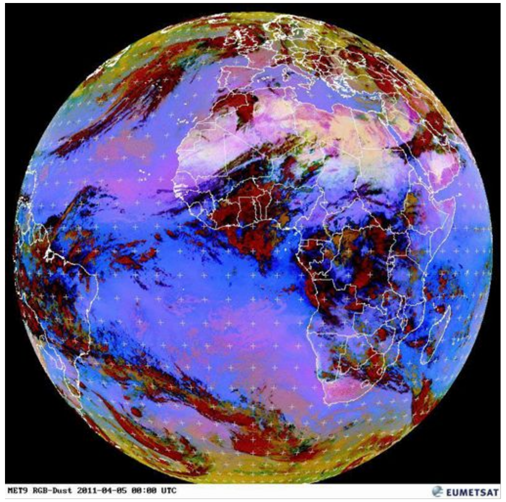
Atmospheric models are a **mathematical representation** of atmospheric water, gas, and aerosol cycles.

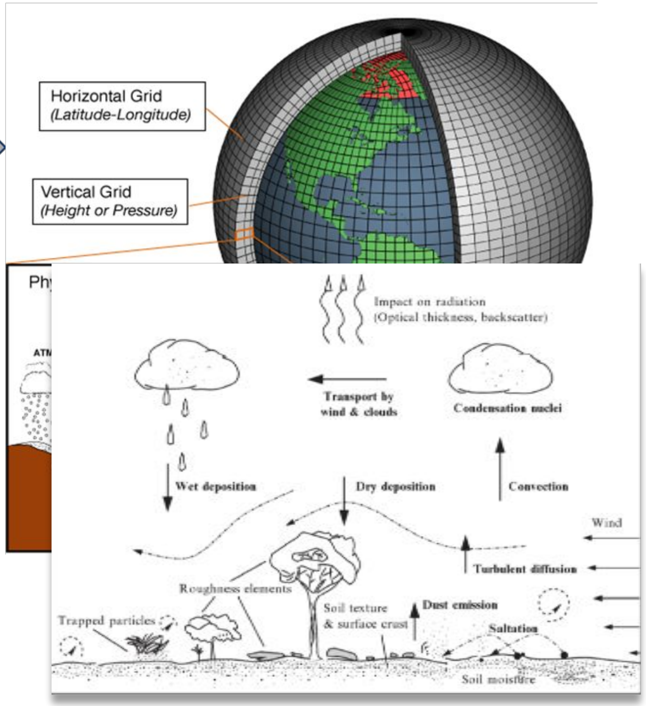




Atmospheric models

Atmospheric models are a **mathematical representation** of atmospheric water, gas, and aerosol cycles.

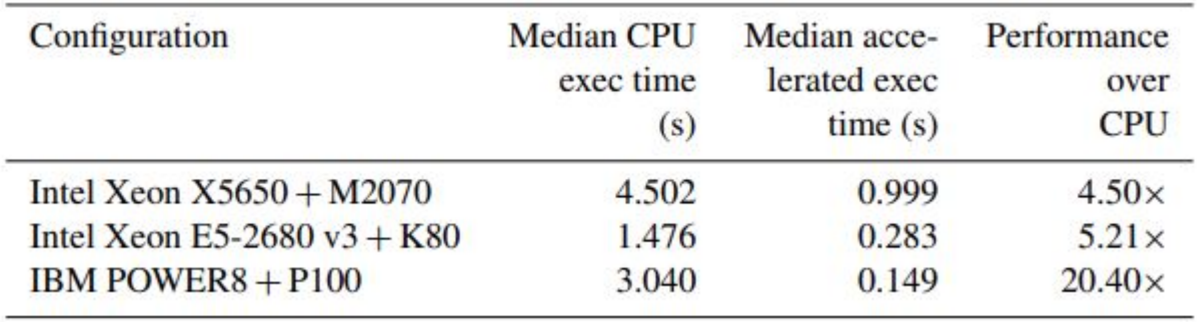




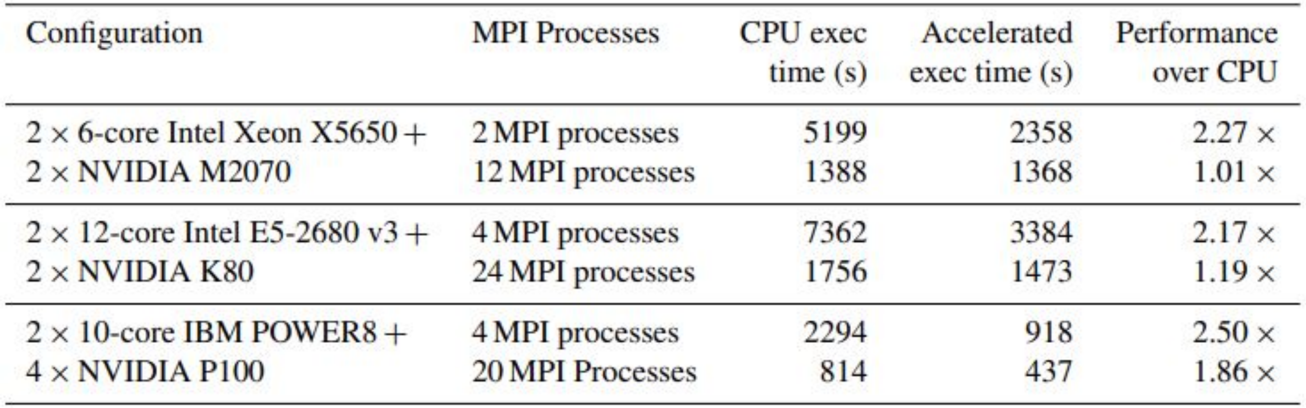
Resolution of chemical processes can take up to 80% of the time execution!

State of the art - KPP GPU

- Kinetic PreProcessor (KPP) is a analysis tool to solve chemical mechanisms using Rosenbrock methods
- KPP is widely used in the atmospheric community
- The GPU version for the EMAC climate model achieves up to 20x speedup against CPU single-core and 1.86x against 2 CPUs



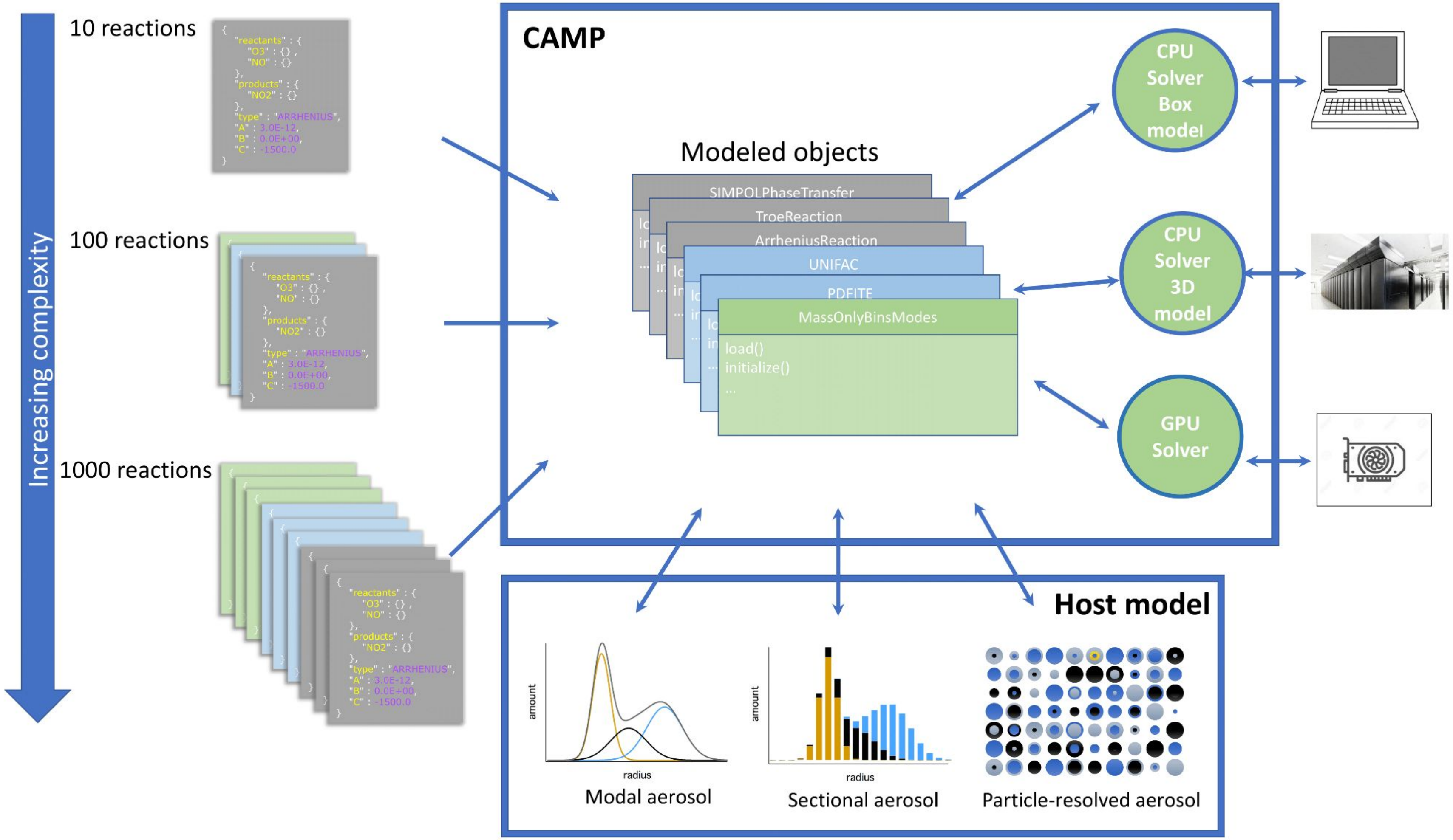
Configuration	Median CPU exec time (s)	Median accelerated exec time (s)	Performance over CPU
Intel Xeon X5650 + M2070	4.502	0.999	4.50×
Intel Xeon E5-2680 v3 + K80	1.476	0.283	5.21×
IBM POWER8 + P100	3.040	0.149	20.40×



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

CAMP: Chemistry Across Multiple Phases

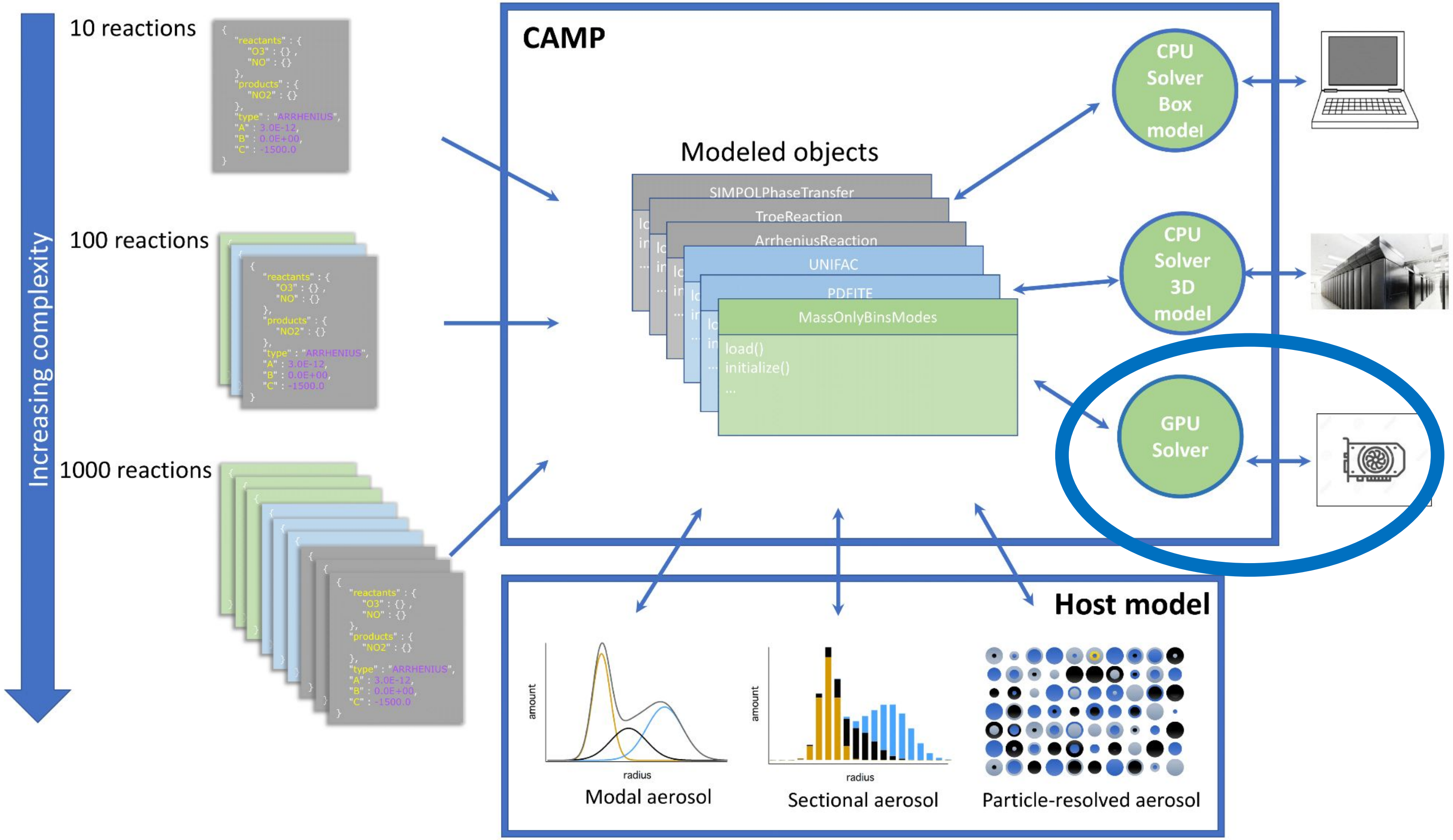


```
{
  "reactants" : {
    "O3" : {} ,
    "NO" : {}
  },
  "products" : {
    "NO2" : {}
  },
  "type" : "ARRHENIUS",
  "A" : 3.0E-12,
  "B" : 0.0E+00,
  "C" : -1500.0
}
```

Dawson, Guzman, Curtis, Acosta, et. al., Chemistry Across Multiple Phases (CAMP) version 1.0, GMD 2022

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CAMP: Chemistry Across Multiple Phases



```
{
   "reactants" : {
      "O3" : {} ,
      "NO" : {}
   },
   "products" : {
      "NO2" : {}
   },
   "type" : "ARRHENIUS",
   "A" : 3.0E-12,
   "B" : 0.0E+00,
   "C" : -1500.0
}
```

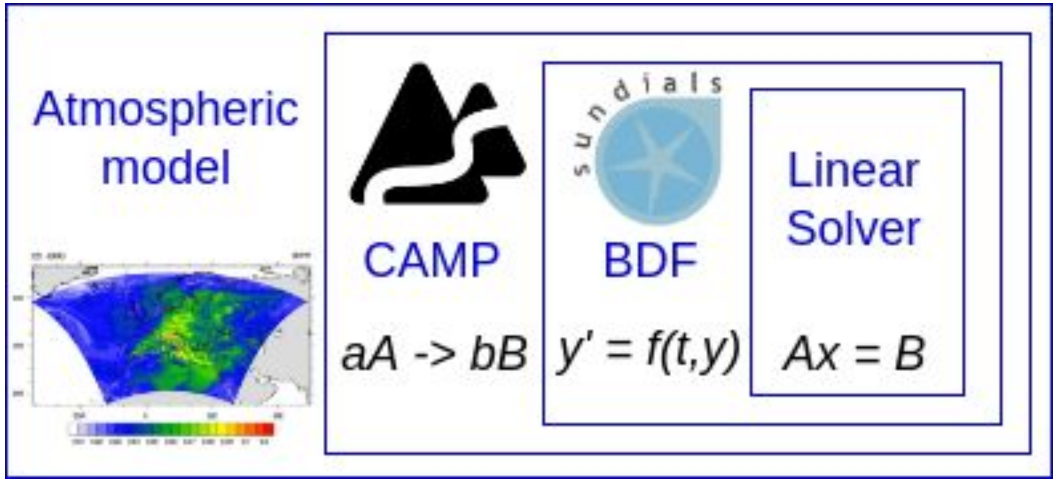
Our objective!

Dawson, Guzman, Curtis, Acosta, et. al., Chemistry Across Multiple Phases (CAMP) version 1.0, GMD 2022

CAMP CPU Solver

- CAMP uses the Backward Differentiation Formula (BDF) from CVODE, which is a solver for ordinary differential equation (ODE) systems.

- BDF requires a linear solver package. The default option is the KLU algorithm for the CPU execution, while it also has a CUDA version of the Biconjugate Gradient (BCG) algorithm.

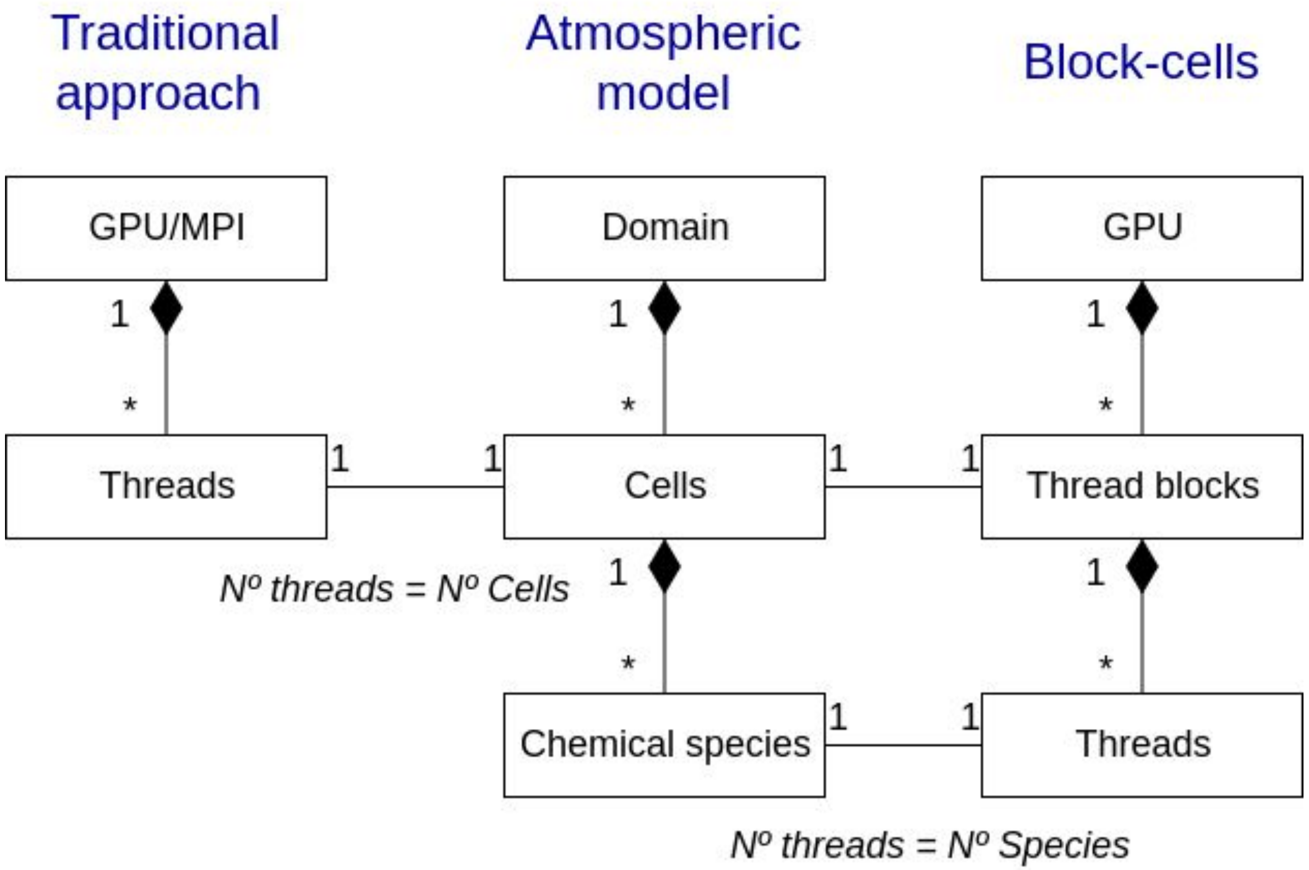


Implementation

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Block-cells (GPU parallelization strategy)

- Block-cells assigns each atmospheric cell to a GPU thread block
- Uses as many threads as chemical species

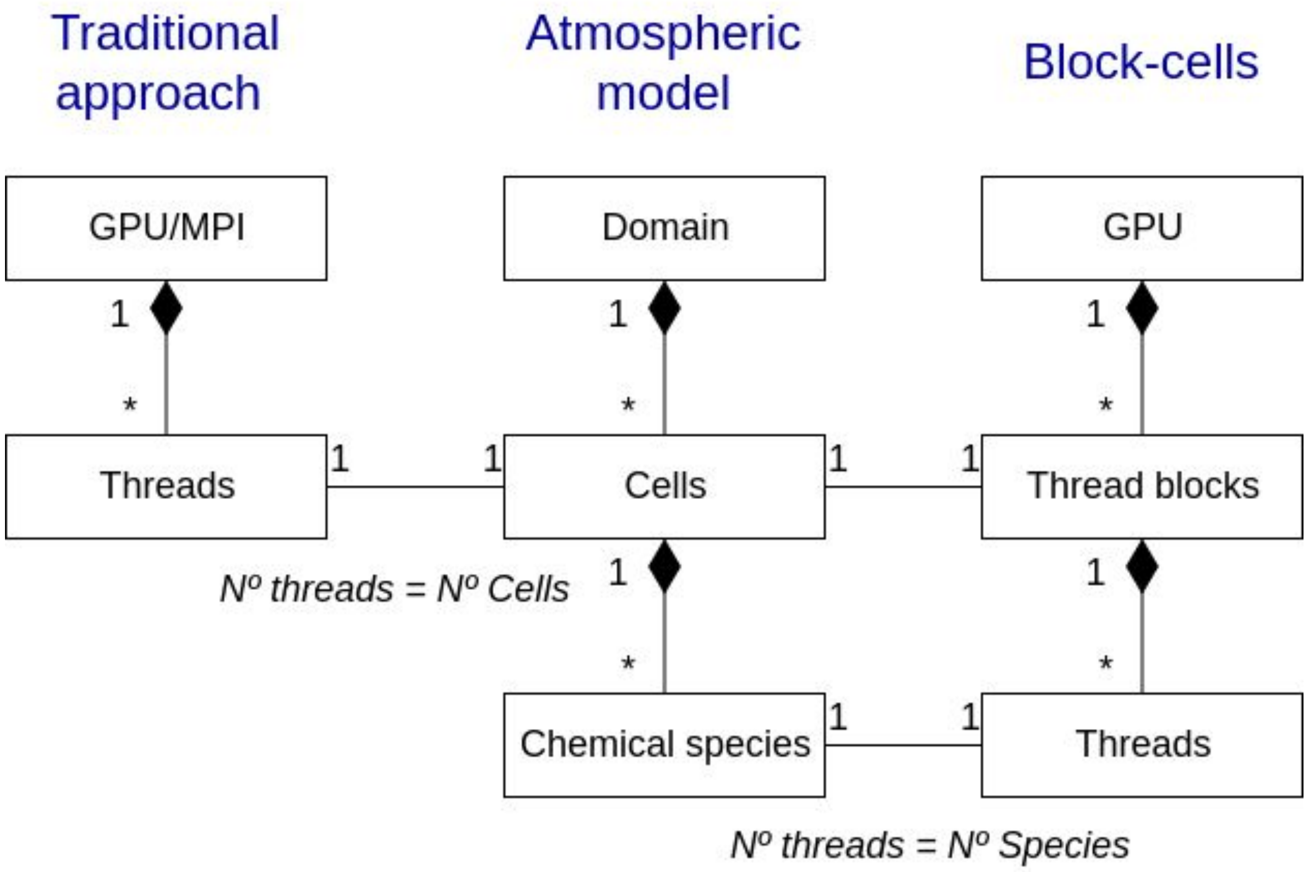


Guzman et. al. Studying a new GPU treatment for chemical modules inside CAMP, 19th ECMWF Workshop

Block-cells (GPU parallelization strategy)

- Higher occupancy than traditional approaches (more threads computing data)

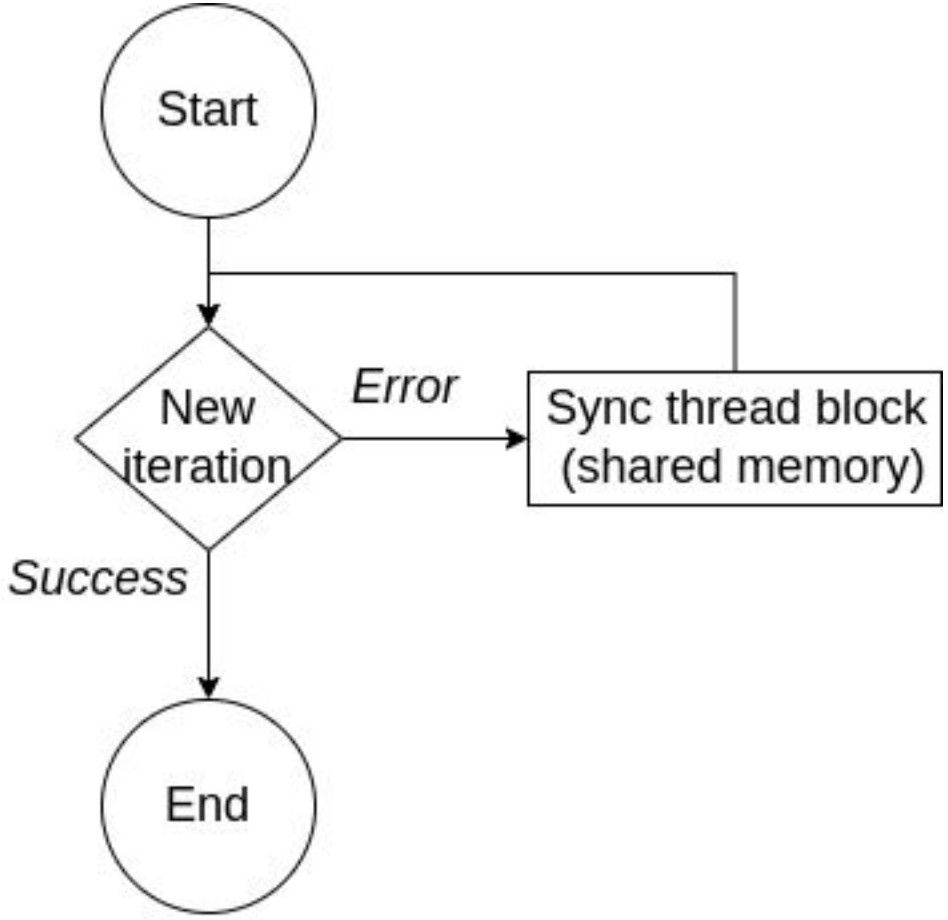
- 34x speedup against CPU single-thread for the CAMP BCG linear solver



Guzman et. al. Studying a new GPU treatment for chemical modules inside CAMP, 19th ECMWF Workshop

Communicating data between threads

- All **communications** are performed at **thread block level**
- An example: A thread triggers an error due to a negative concentration
- The error is shared between the other threads in the block by using shared memory



GPU BDF convergence loop

Test environment

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 **40 cores per node)**
  - **4 x GPU NVIDIA V100 (Volta)** with 16GB HBM2**.**
  - Compilers: GCC version 6.4.0 and NVCC version 10.2

Software configuration

Architecture	Parallel resources	Parallelization language
CPU	1, 40	MPI
GPU	Nº of different chemical concentrations (species x cells)	CUDA

- The evaluation is performed over the code included in the most external loop in BDF. The code related to previous initializations is excluded.

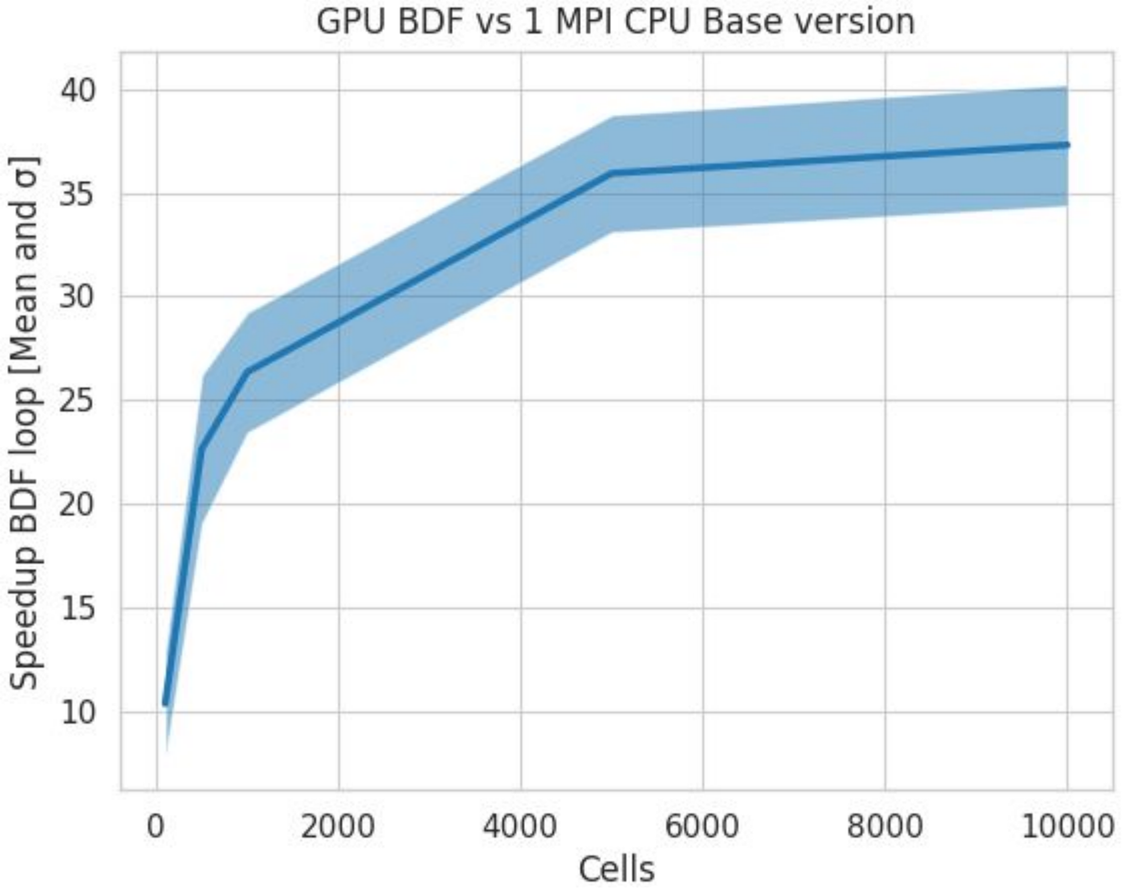
- Chemical mechanism: Gas phase chemistry from Carbon bond 2005 (CB05) | Chemical species: 156 | Cells (ODE systems): 100-10,000 | GPU Shared memory size per block: 256 | CVODE absolute tolerance: 0.01% | BCG tolerance: 1.0e-30

Results

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Speed-up

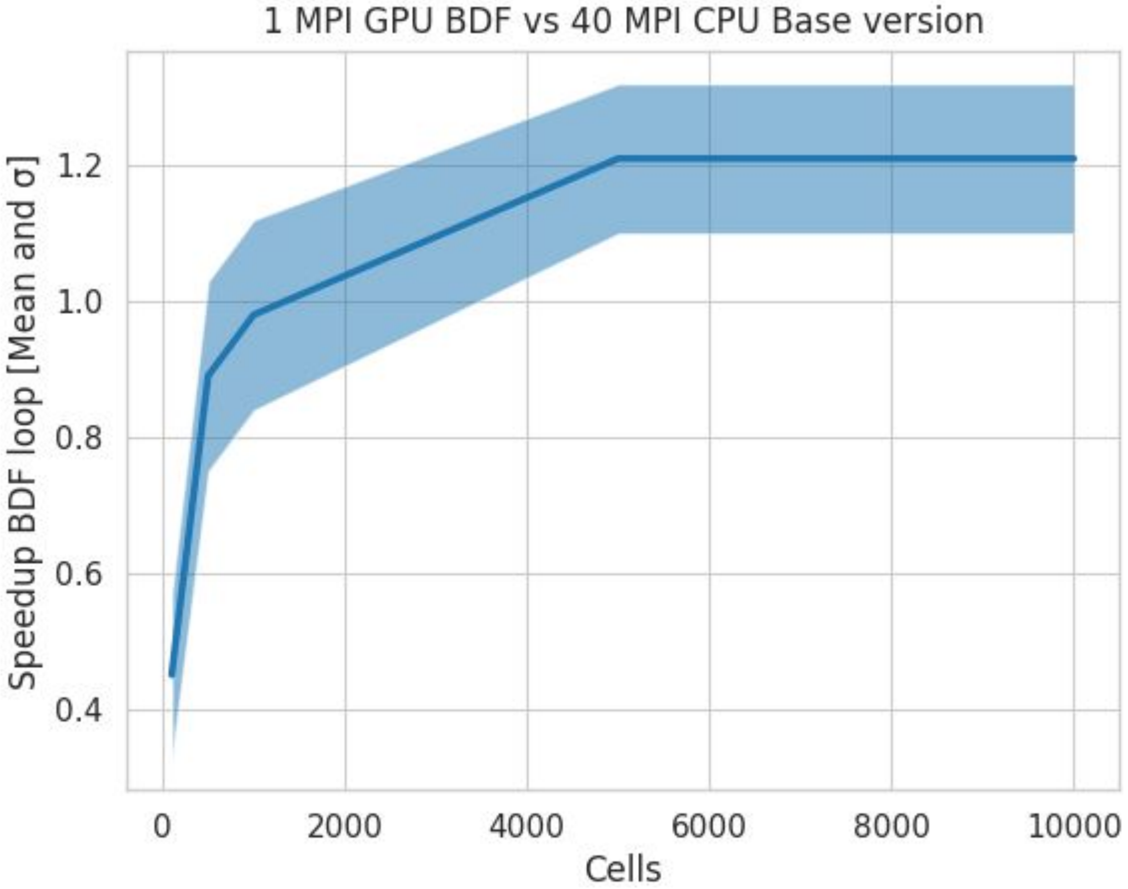
- Up to 35x speedup in average vs single-thread
- Standard deviation around 2



Speed-up against 40 processes

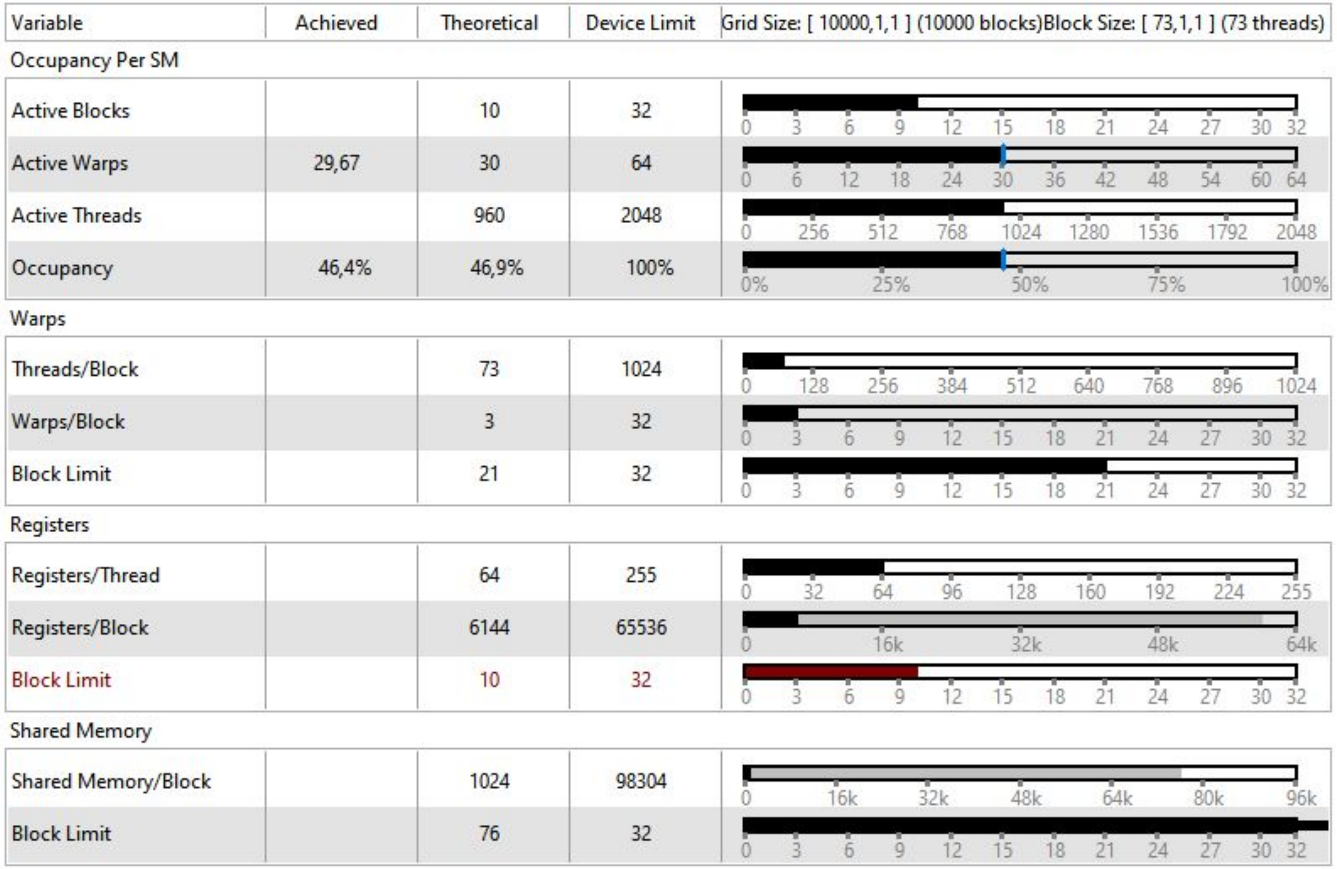
- 1.2x speed-up against a fully CPU node (40 MPI processes)

- Since there’s no communication between threads, we estimate 4.8x speed-up using the full GPU resources in a node (4 GPUs)   - *Ongoing work*



Kernel profiling

- Some optimizations already performed (like adjusting the number of registers per thread)
- The register usage is likely preventing the kernel from fully utilizing the GPU
- This usage is mostly produced by the algorithm definition, which computes big data like the Jacobian matrix



Variable	Achieved	Theoretical	Device Limit	Grid Size: [10000,1,1] (10000 blocks)Block Size: [73,1,1] (73 threads)
Occupancy Per SM				
Active Blocks		10	32	
Active Warps	29,67	30	64	
Active Threads		960	2048	
Occupancy	46,4%	46,9%	100%	
Warps				
Threads/Block		73	1024	
Warps/Block		3	32	
Block Limit		21	32	
Registers				
Registers/Thread		64	255	
Registers/Block		6144	65536	
Block Limit		10	32	
Shared Memory				
Shared Memory/Block		1024	98304	
Block Limit		76	32	

Conclusions

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Conclusions

- Our Block-cells strategy increase the GPU parallel threads against traditional implementations (Nº Threads = Nº Cells)

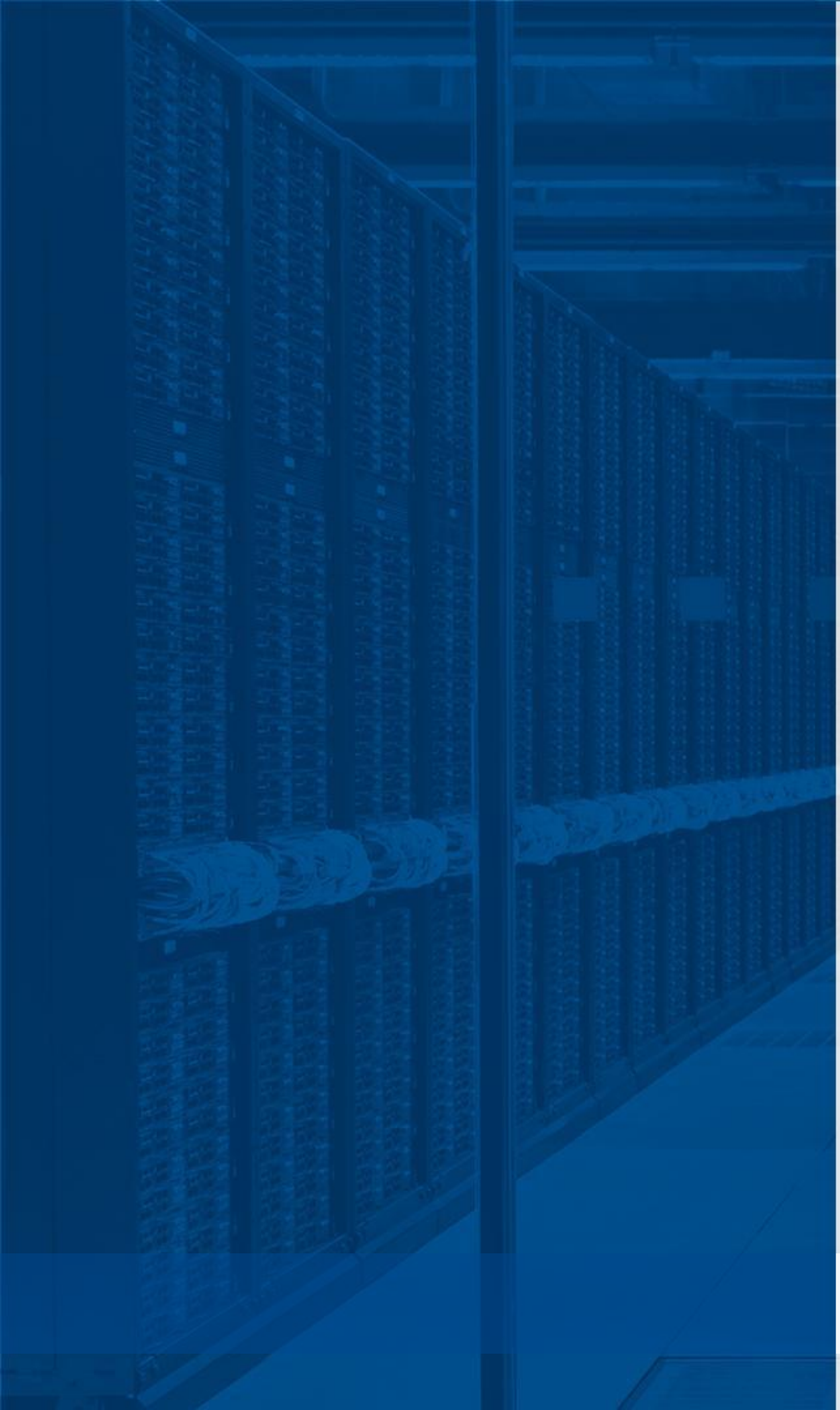
- The CUDA BDF loop performs up to 35x times faster than CPU single-thread
  - 1.2x speed-up against CPU using the fully resources node.
  - Since the load is independent between threads, we estimate up to 4.8x speedup using 4 GPUs per node.

- The kernel profiling suggests a limitation on the performance by memory

- Our approach can be used in more chemical applications thanks to the versatility of the CAMP module.

Future work

- Add multi-device functionality to compute up to 4 GPus per node.
- Balance load between CPU and GPU architectures.
- Integrate our implementation inside an atmospheric model.



BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you

christian.guzman@bsc.es