Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Accelerating Atmospheric Models using GPUs

Christian Guzman Ruiz

Master thesis on Modelling for Science and Engineering

09/09/2019

Index

Introduction

Introduction

- **Atmospheric models** are widely used in meteorological institutions and investigation centers for weather forecast and climate prediction studies

- The Multiscale Online Nonhydrostatic AtmospheRe CHemistry model (NMMB-MONARCH or **MONARCH**) **consumes** thousands of hours on Marenostrum IV supercomputer

- **GPU** potential on supercomputers is higher than CPU

Introduction: Objectives

1. Reduce the execution time of MONARCH

2. Study the potential of GPU implementations on Earth Science models

-> Apply implementations focused on the GPU<-

Problem: Translate code to GPU is hard

Methodology

Methodology: Work

1. Search most time-consuming MONARCH component -> **Chemistry mechanism, *phlex-chem* option**
2. Identify most time-consuming *phlex-chem* functions -> **Derivative**
3. Analyze most relevant function-> **Develop a GPU basic version**
4. Apply GPU Derivative and Deriv optimizations -> **Speedup positive for big mechanisms**
5. Apply optimization to include more data in *phlex-chem* -> ***N cells***

Methodology: Test Environment

Marenostrum IV:

- Peak Performance of 11.15 Petaflops
- 384.75 TB of main memory
- 3,456 nodes:
  - **2x Intel Xeon Platinum 8160 24C at 2.1 GHz**
  - 216 nodes with 12x32 GB DDR4-2667 DIMMS (8GB/core)
  - 3240 nodes with 12x8 GB DDR4-2667 DIMMS (2GB/core)
- Interconnection networks:
  - 100Gb Intel Omni-Path Full-Fat Tree
  - 10Gb Ethernet
- **Compiler: ICC version 17.0.4**

Methodology: Test Environment

CTE-POWER:

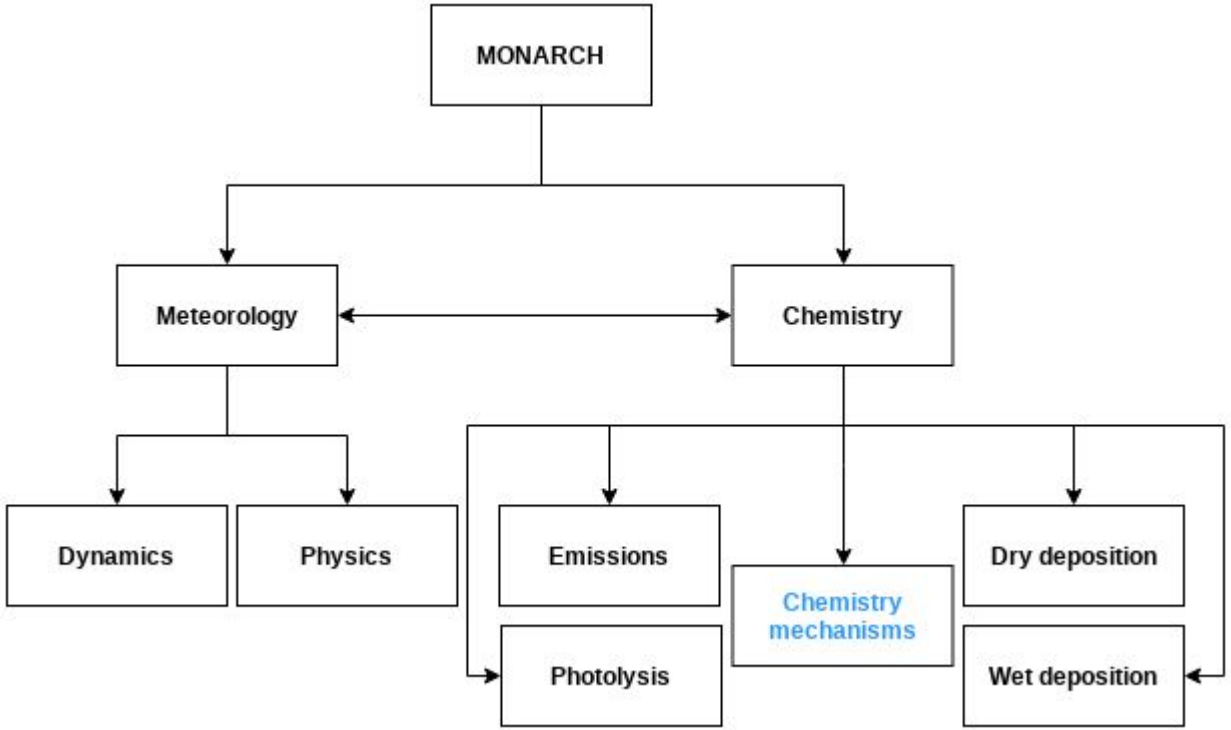
2 login nodes and 52 compute nodes, each of them:

- **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.**
- GPFS via one fiber link 10 GBit
- **Compilers: GCC version 6.4.0 and NVCC version 9.1**

MONARCH introduction

MONARCH introduction: Components

MONARCH components **computes** a region of the map divided in **points** (cells) over a number of **time-steps**



MONARCH introduction: Chemistry mechanisms

Three different solutions:

1. **Euler-Backward-Iterative (EBI):** Efficient and fast solver, fix code with Carbon Bond 2005 (**CB05**) mechanism. **Default solver**
2. **Kinetic PreProcessor (KPP)**: Partially run-time configured mechanism
3. ***The Phlexible Module for Chemistry*** *(Phlex-chem, provisional name)*: Novel option, provide a flexible framework between MONARCH and chemistry mechanisms.

MONARCH modules time-impact

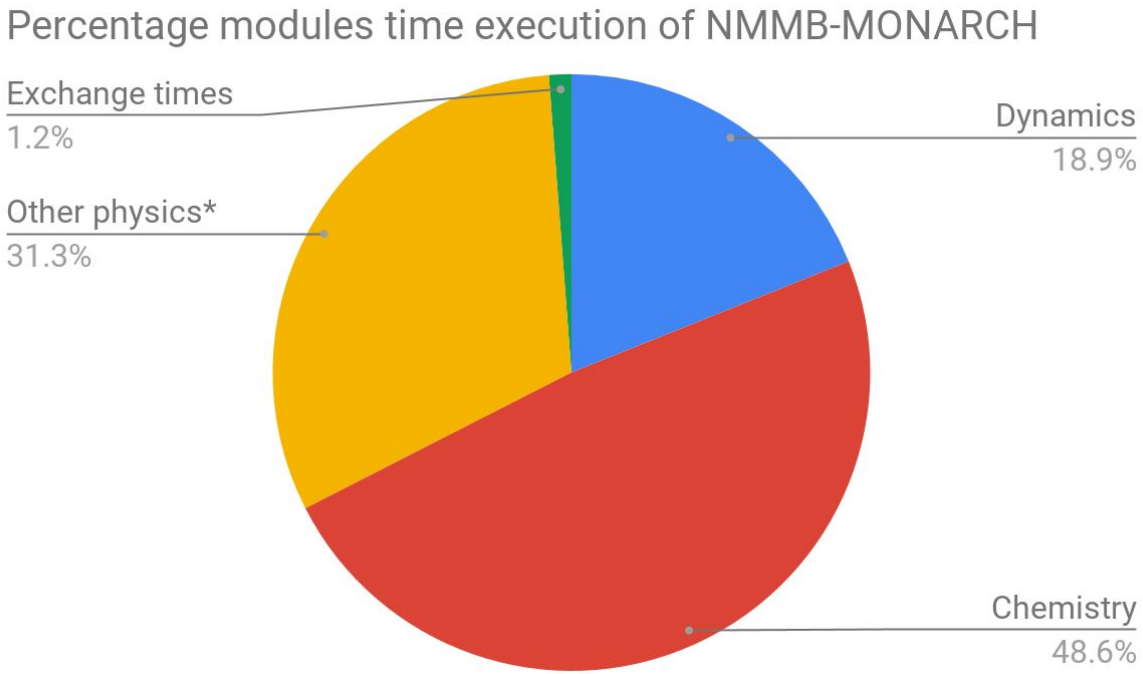
MONARCH measurements

Default configuration

Solver: EBI

Mechanism: Carbon Bond 2005 (**CB05**)

Focus on *phlex-chem*
-> Comfortable and expensive



NMMB-MONARCH v2.0
Multiscale Online Nonhydrostatic AtmospheRe CHemistry model

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Universitat Autònoma de Barcelona

Phlex-chem overview

Phlex-chem overview

- Allows user define chemical mechanisms through JSON input files

- Solve a set of equations:

  ***phlex-chem* ->** y= $r_1X_1 + r_2X_2$ -> r_3X_3**; CVODE ->** y'= f(t,y)

- Predict the **concentrations** of some chemical reactants [*X*]

- Rate constants **[*r*]** are calculated by **chemical reactions:**

$$Ae^{(\frac{-E_a}{k_b T})}(\frac{T}{D})^B(1.0 + E * P)$$

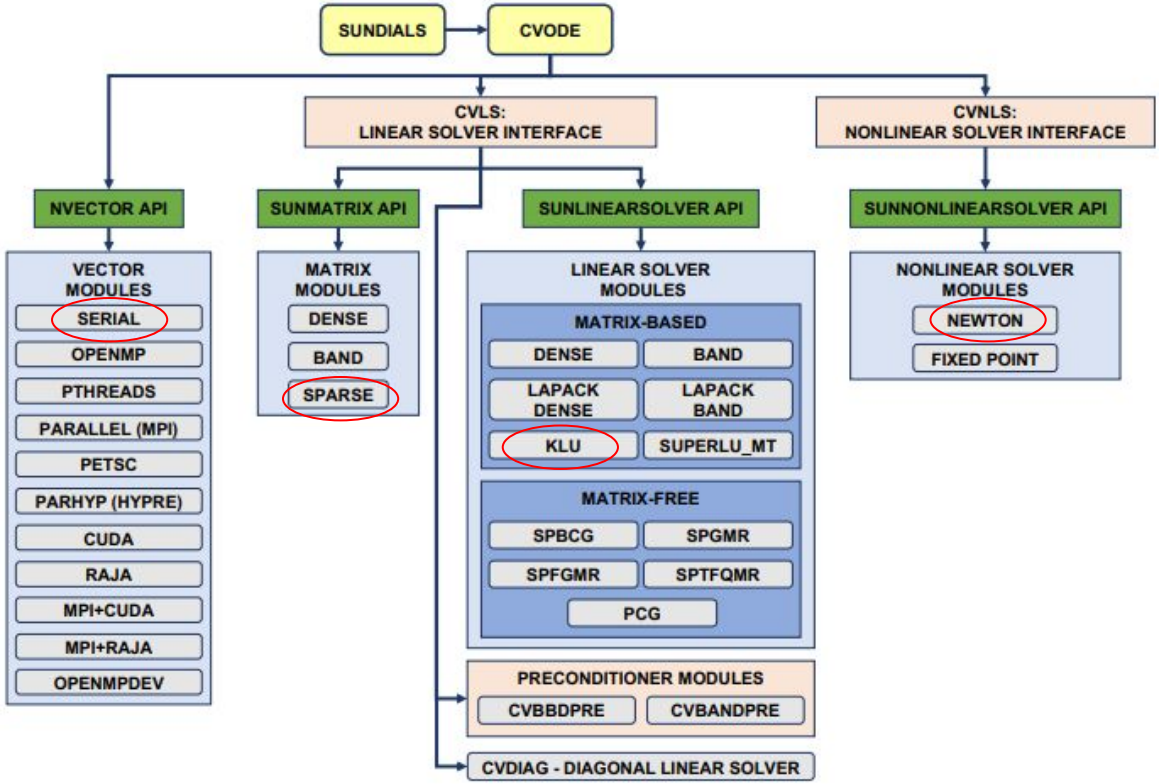
Differential equation solver: CVODE

Differential equation solver: CVODE

- Predict future concentrations **(y'=f(t,y)) -> CVODE**

- Requires from *phlex-chem:*

  f(y) *and* $J = \partial \mathbf{f} / \partial \mathbf{y}$



CVODE: KLU SPARSE

- **KLU SPARSE** store and compute the Jacobian matrix ($\partial f / \partial y$)

- **SPARSE:** Efficient structure for matrix with few nonzeros

$$r_1 X_1 + r_2 X_2 \rightarrow r_3 X_3$$

- **KLU method**
  - **First setup**: Symbolic factorization
  - **Setup**: KLU *refactor* and numerical conditioning (*rcond*)
  - **Solve**: Pivoting and forward and backward substitution

Phlex-chem on MONARCH

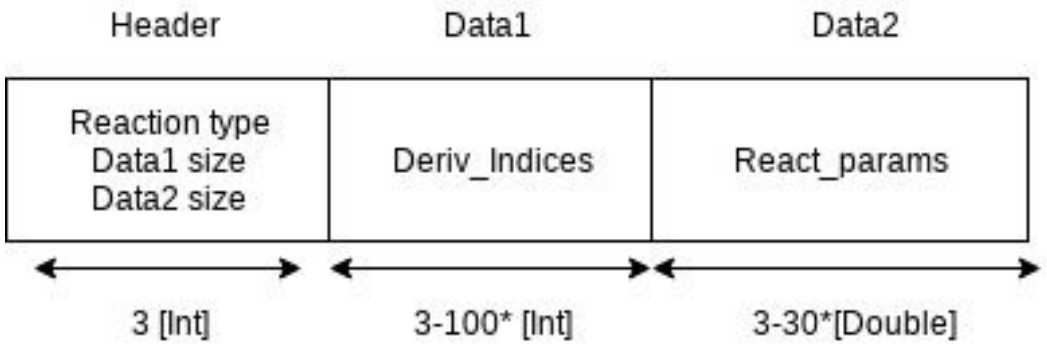
Phlex-chem on MONARCH

- ***Phlex-chem*** executed by tests **independently** from MONARCH, but from a **MONARCH point of view**
- **Concepts summary:**
  - **State** [X_N]*:* Chemical concentrations array
  - **Rates** [r_N]: Reaction rates, results of reaction equations
  - **f(y)**: $r_1X_1 + r_2X_2 \rightarrow r_3X_3$
  - **Derivative:** Solve f(y). CVODE needs it
  - **Jacobian:** Solve $\partial f / \partial y$. CVODE needs it
  - **Cell:** Point of the map for MONARCH. Use all the system variables. Cells are independent from each other

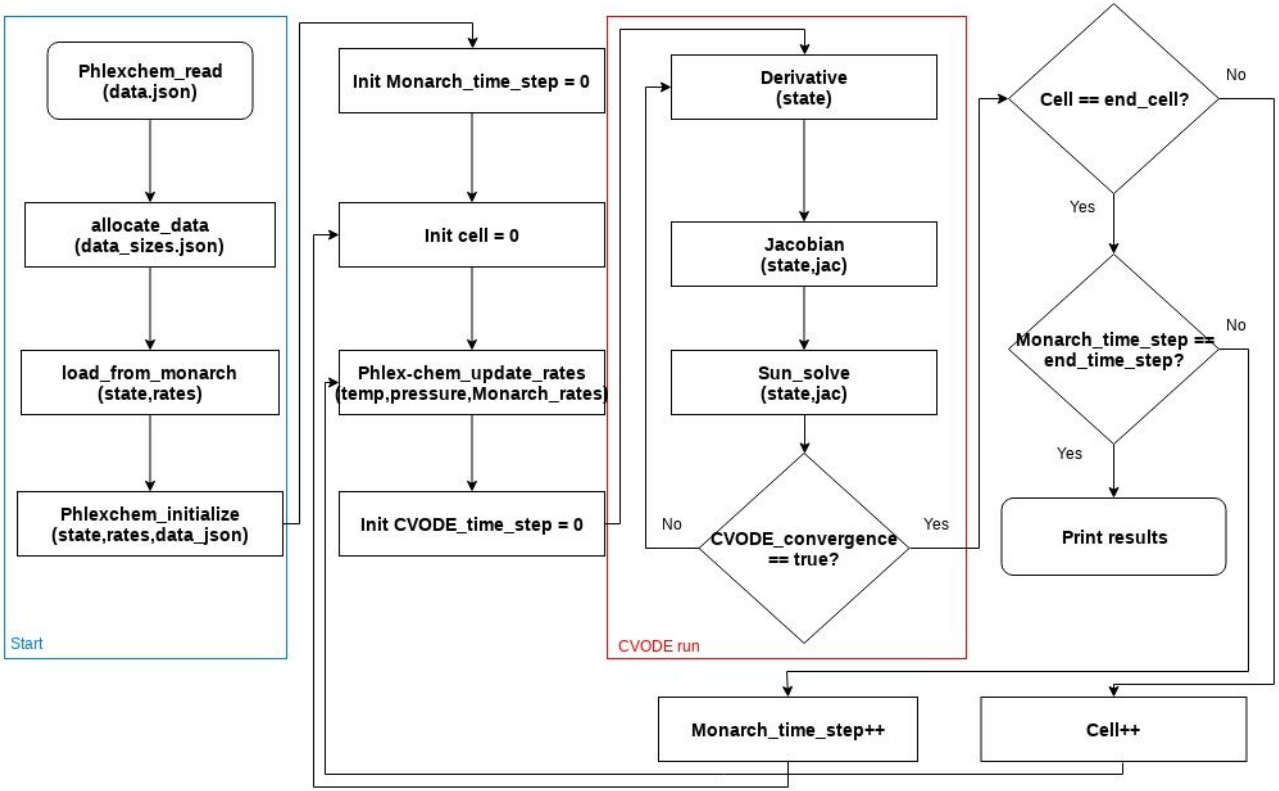
Phlex-chem on MONARCH: Data structure

- **All** the **reaction** related **data** are stored on the **same array structure: RXN**

$$Ae^{(\frac{-E_a}{k_b T})}(\frac{T}{D})^B(1.0 + E * P)$$

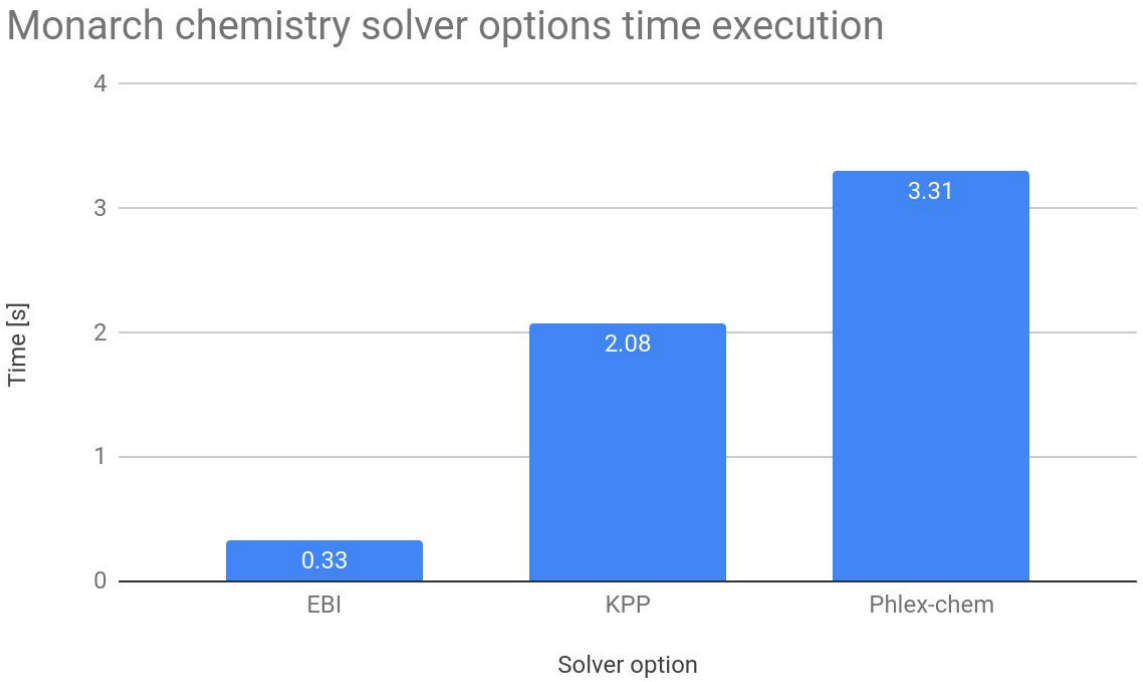


Phlex-chem on MONARCH: Workflow



Phlex-chem on MONARCH: Measurements

- CB05 mechanism
- 100 Time-steps * 100 Cells -> 10,000 solver iterations



NMMB-MONARCH v2.0
Multiscale Online Nonhydrostatic AtmospheRe CHemistry model

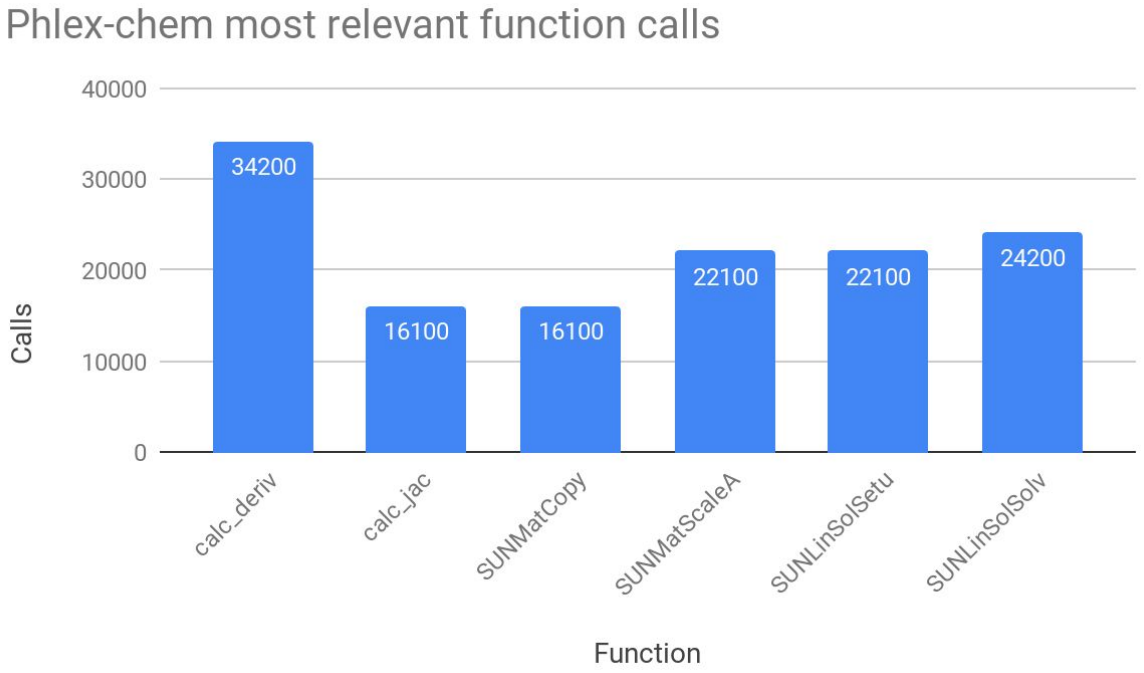
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Universitat Autònoma de Barcelona

Phlex-chem on MONARCH: Measurements

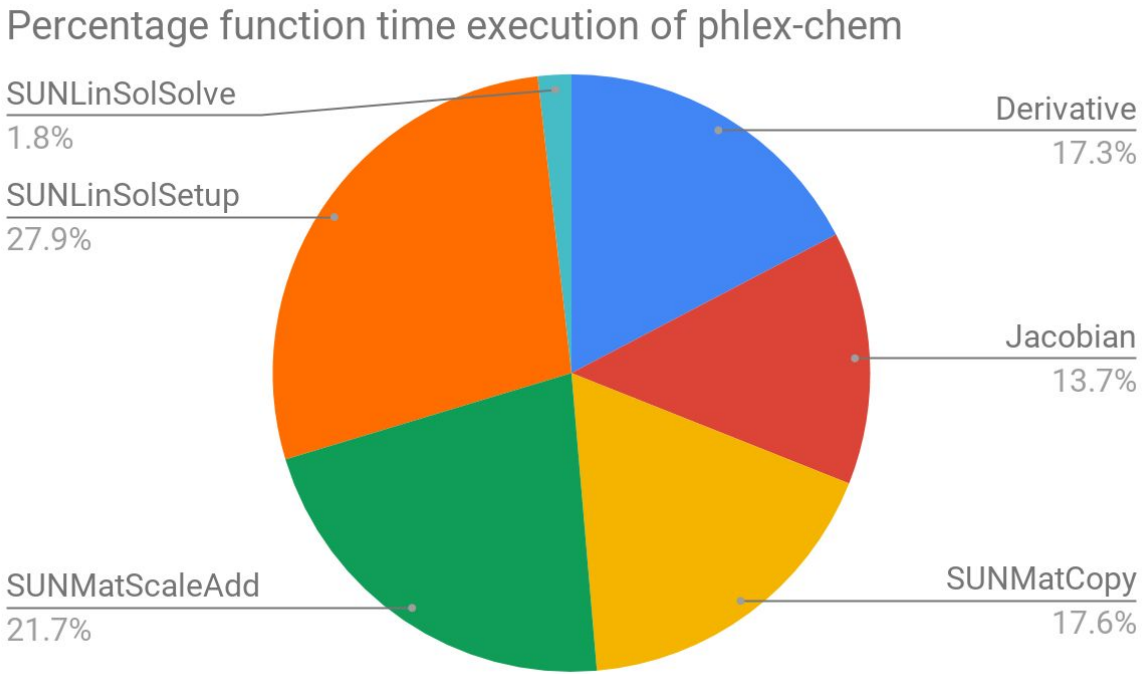
- CVODE calls:
  - Derivative: 3-4
  - Jacobian: 1-2

- Mean **function time** per iteration is in the order of **µs**



Phlex-chem on MONARCH: Measurements

- Flat profile among all functions
- Derivative and Jacobian = 30% Both are very similar -> **Improve Derivative**

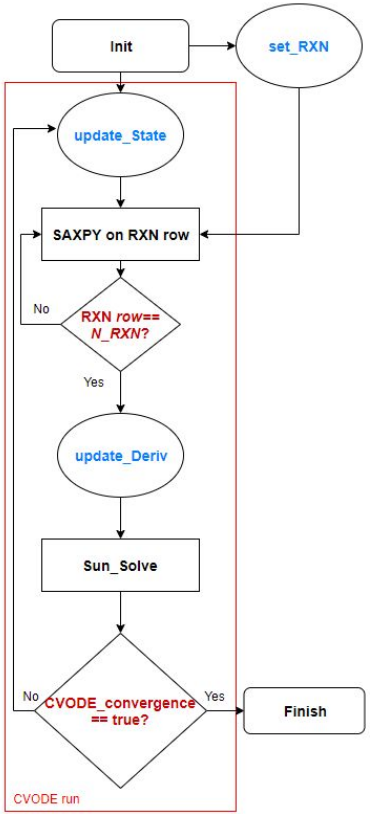


Phlex-chem: Derivative analysis

Phlex-chem: Derivative workflow

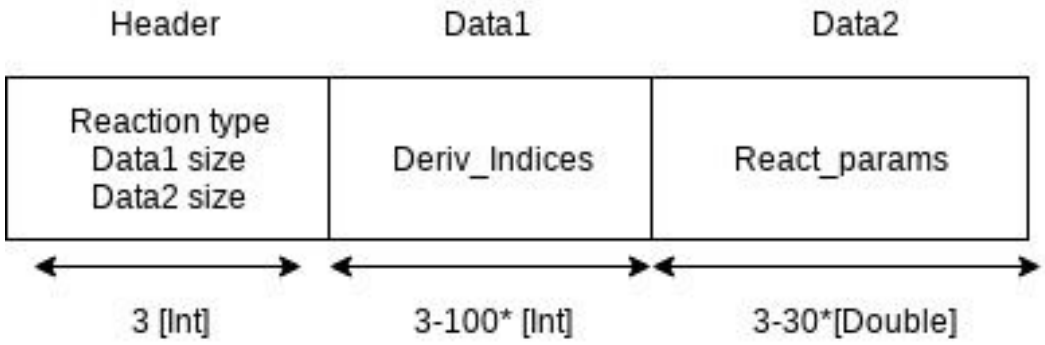
- $r_1X_1 + r_2X_2 \rightarrow r_3X_3$ translates to SAXPY operations in Derivative & Jacobian:

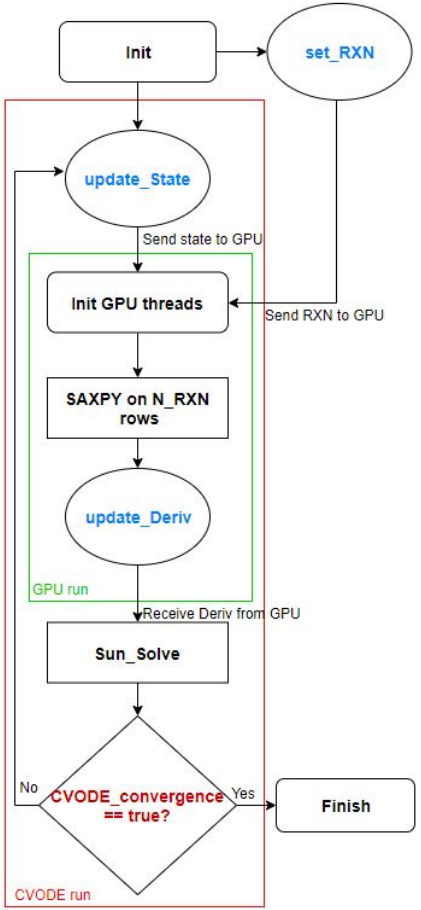
$$y_{i+1} \leftarrow \alpha x + y_i \; ; \quad i=0, 1 \ldots N\_RXN$$



GPU Derivative

- **Parallelize reactions loop**
- **RXN loop deleted**: Work distributed between threads
- **Added data transfers** between CPU & GPU
- **Atomic** SAXPY operation





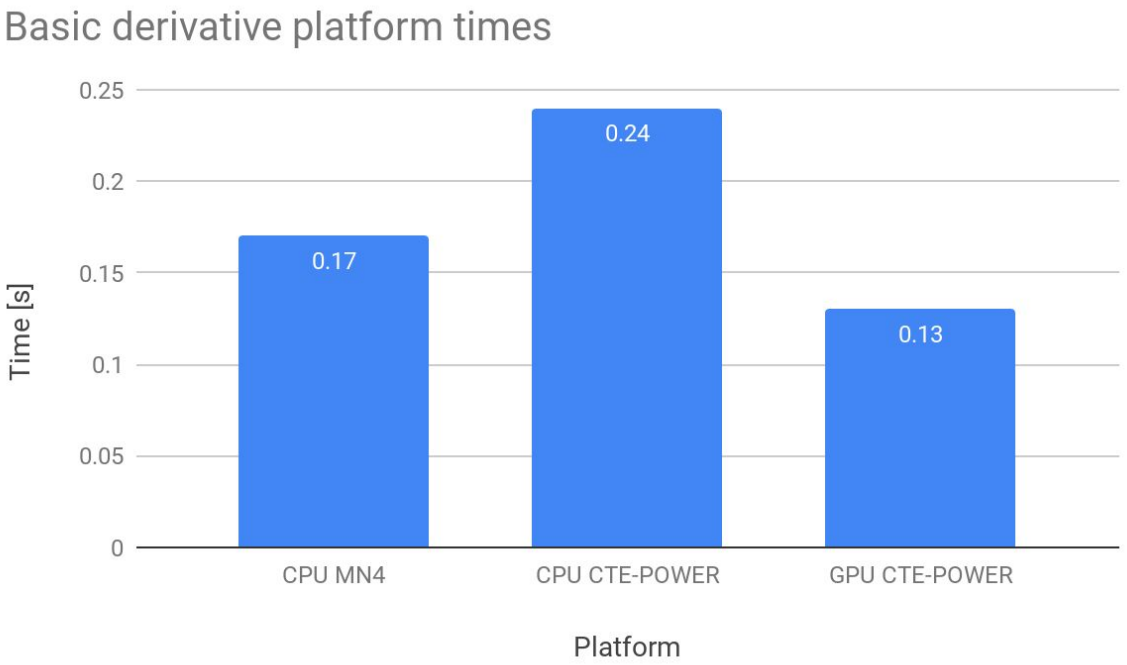
NMMB-MONARCH v2.0
Multiscale Online Nonhydrostatic AtmospheRe CHemistry model

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

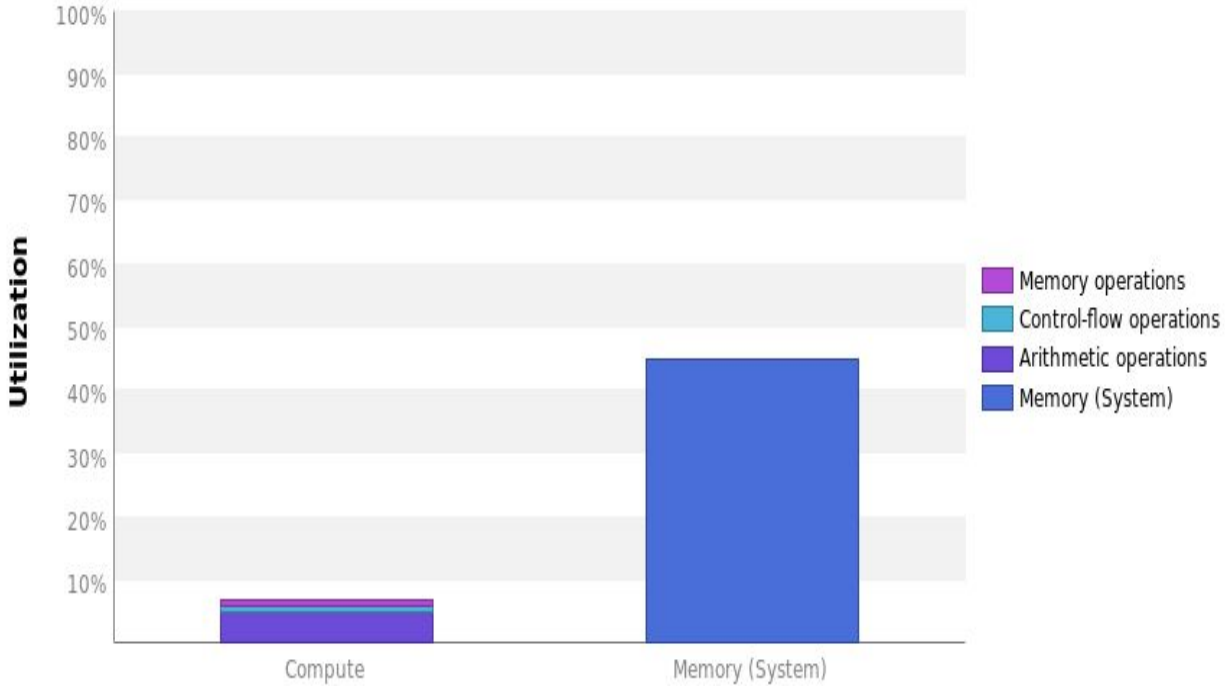
Universitat Autònoma de Barcelona

Basic Derivative

- **Basic Derivative:** Simple script, Similar function without *phlex-chem* or CVODE
- **Configuration:**
  - CALLS: 1000
  - SPECIES: 100
  - N_RXN: 5000

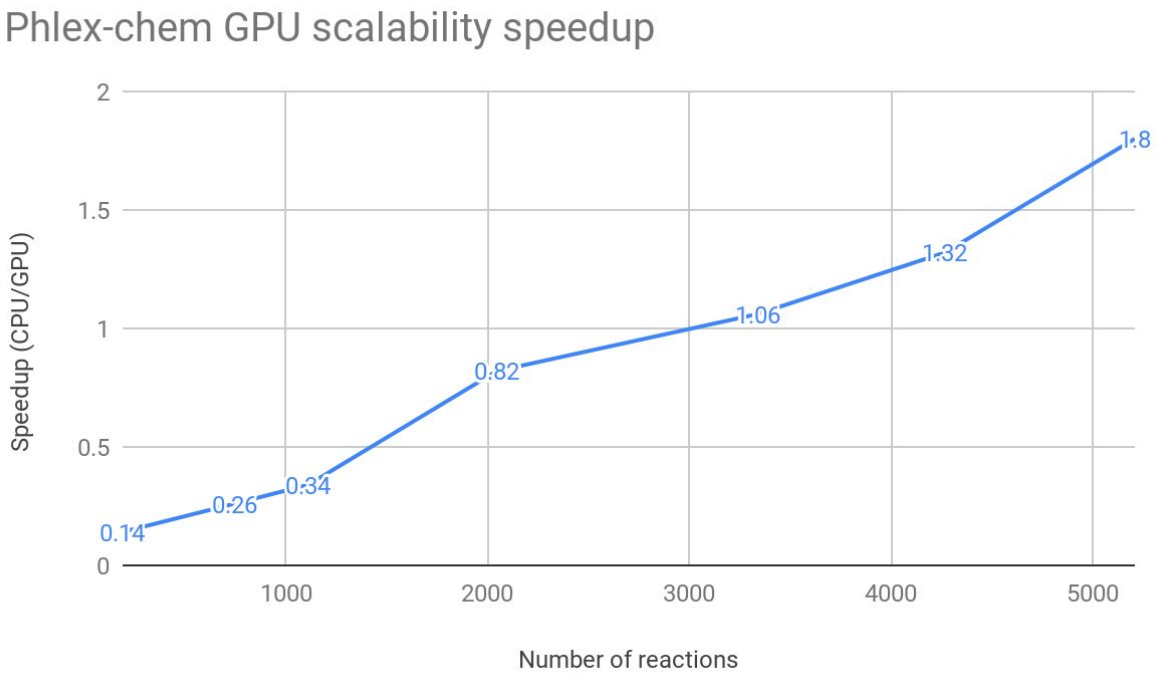


Basic Derivative



Derivative GPU

- **Configuration:** CB05 repeated, 10,000 calls
- **Big mechanisms** are worth to compute on **GPU**
- We can still improve memory access



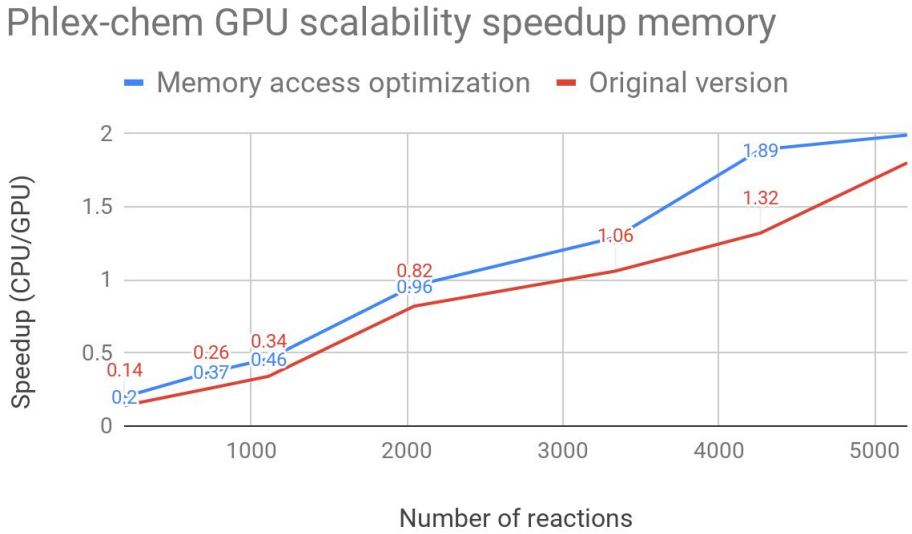
Derivative GPU: Memory optimization

Rxn old structure

Value 1	Value 4	Value 7
Value 2	Value 5	
Value 3	Value 6	Value 9

Rxn optimized structure

Value 1	Value 2	Value 3
Value 4	Value 5	Value 6
Value 7		Value 9



NMMB-MONARCH v2.0
Multiscale Online Nonhydrostatic AtmospheRe CHemistry model

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

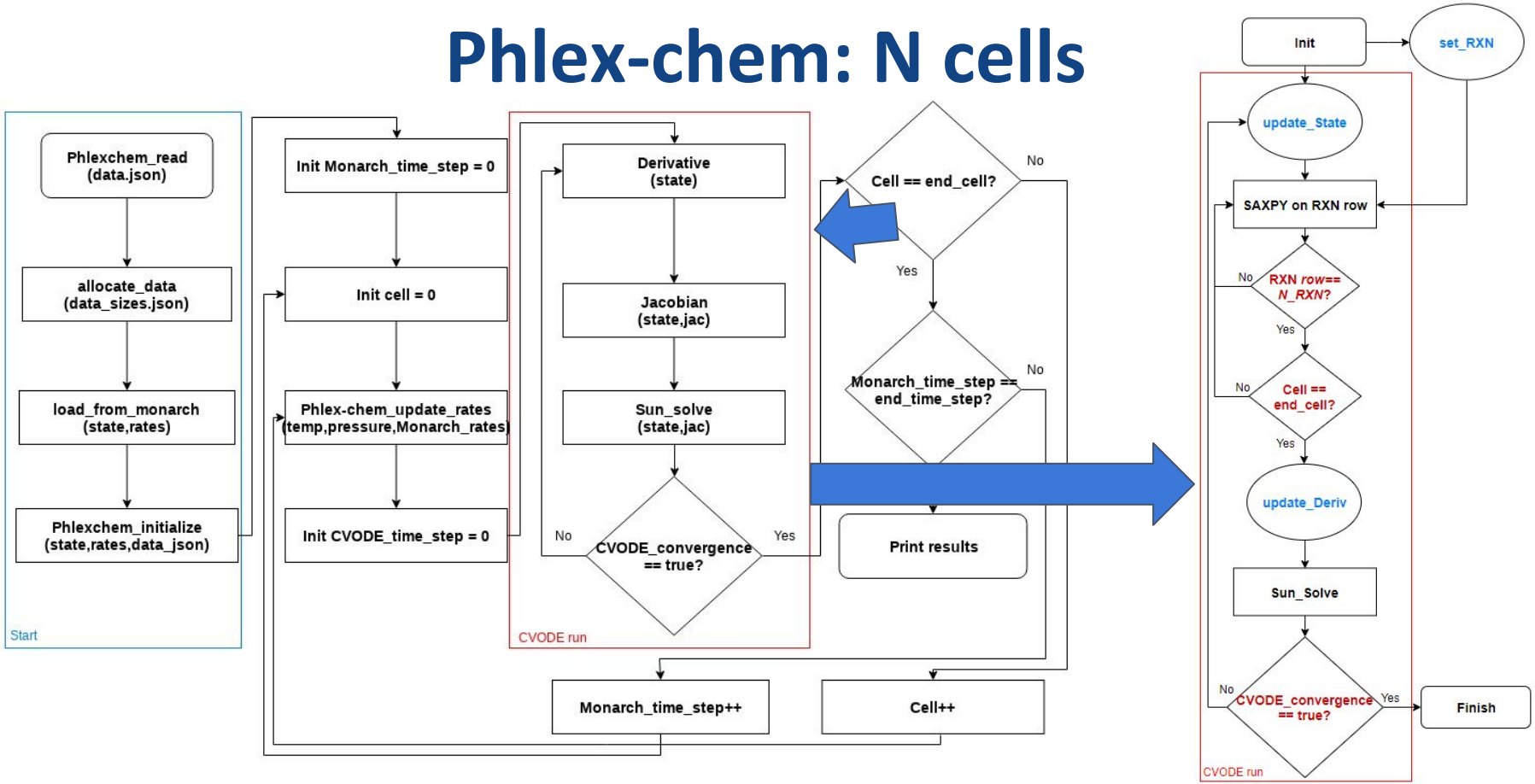
Universitat Autònoma de Barcelona

Phlex-chem: Multiple cells optimization

Phlex-chem: N cells

- **Keys:**
  - **Cells** data has **no-dependance** between them
  - GPU bottleneck is on data transfers
  - **GPU works better with bigger data sizes**
  - **Monarch** computes **10,800 cells** per core by default

Phlex-chem: N cells



NMMB-MONARCH v2.0
Multiscale Online Nonhydrostatic AtmospheRe CHemistry model

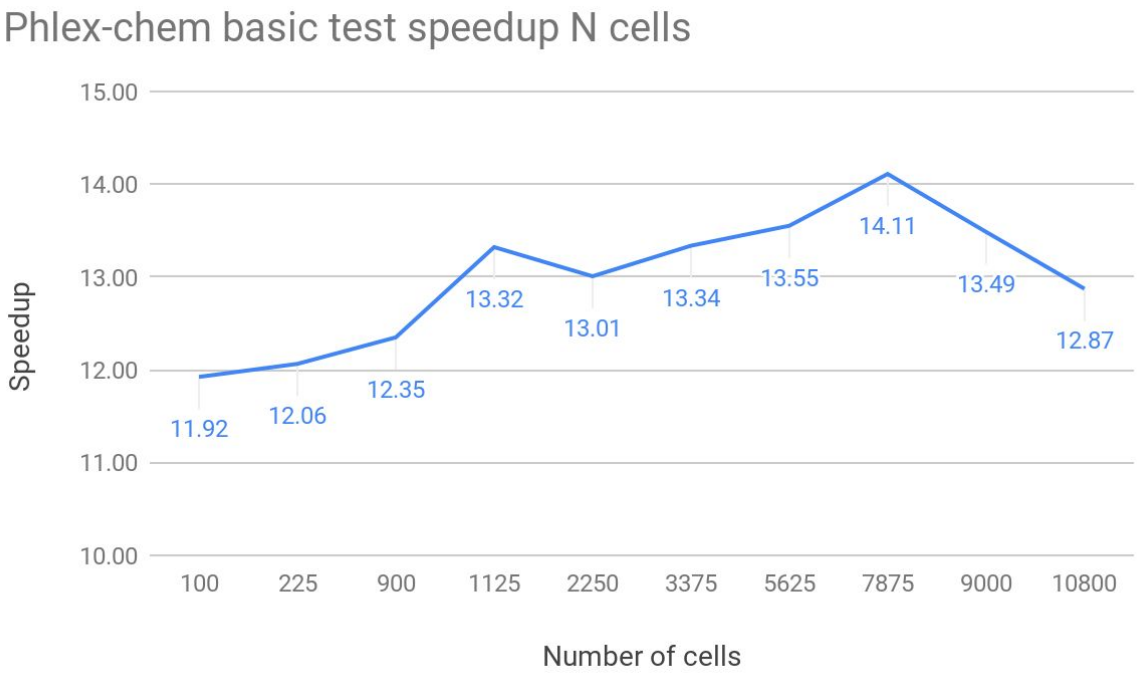
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Universitat Autònoma de Barcelona

Set up N cells

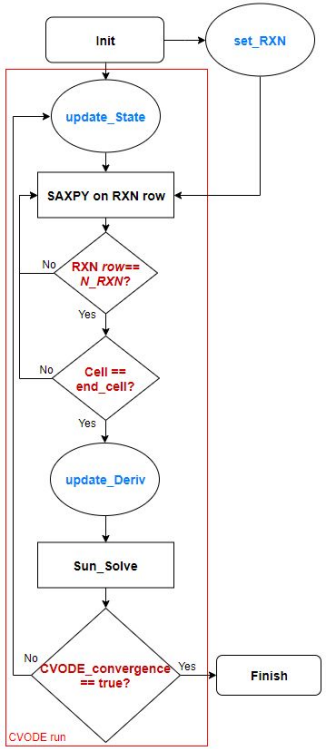
- **Few changes** on the code
- **Simple MONARCH test**: Only arrhenius reaction, few species
- A lot of improvements:
  - **Avoid reset** variables each iteration
  - **Reduce cache misses**
  - **Benefit from vectorization** on RXN
  - **Reduce calls** of Derivative and Jacobian
  - **Benefit from KLU SPARSE**

CPU N cells: Results

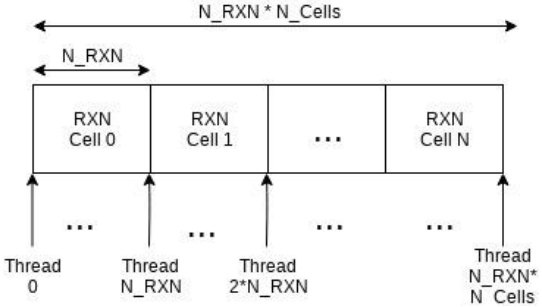


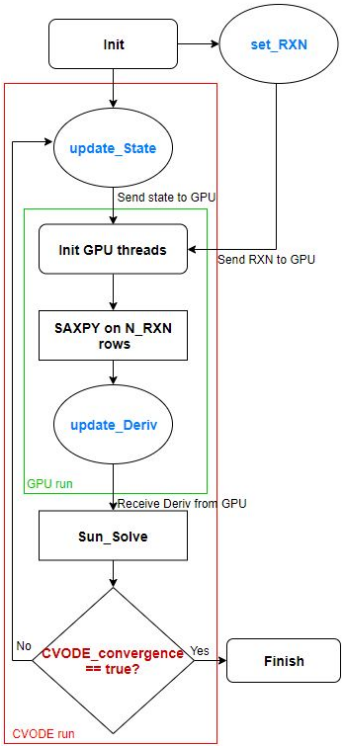
GPU N cells

CPU



Reaction & Cell parallelization

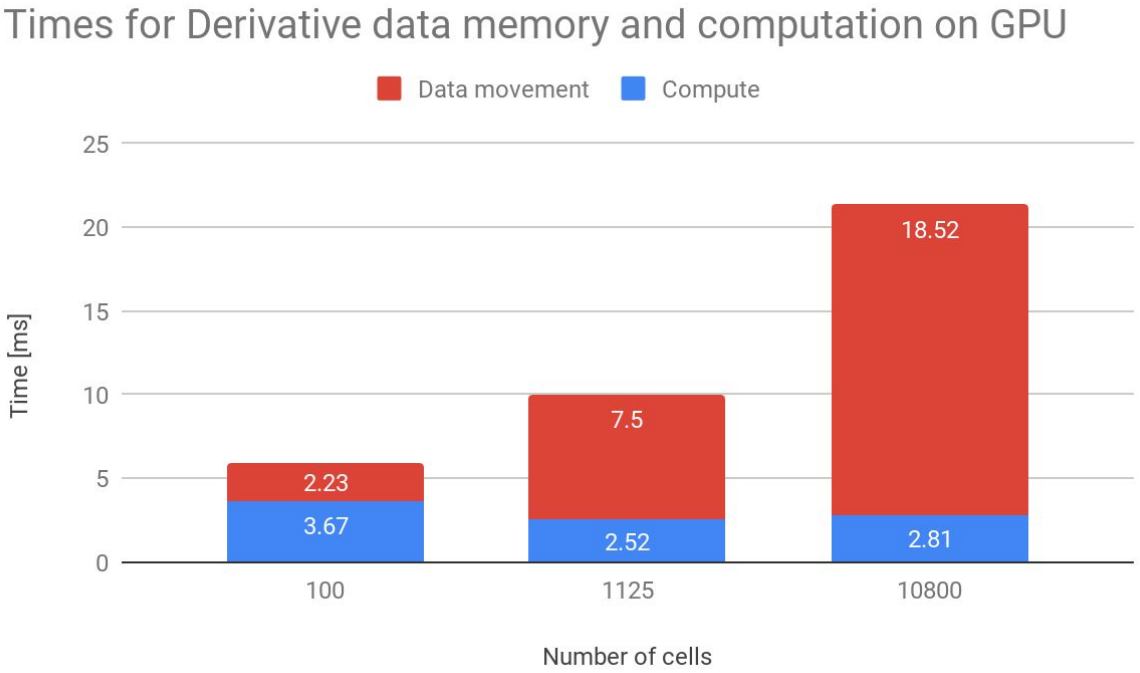




GPU

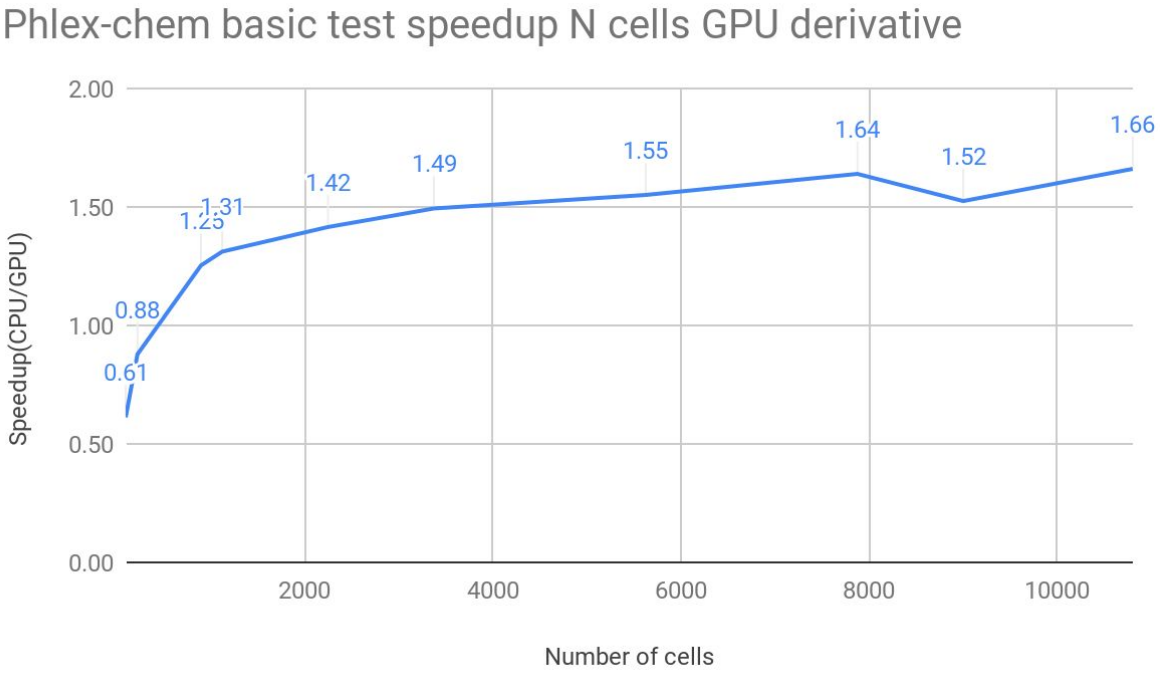
GPU N cells: Results

- **Compute**: Fix time
- **Data**: Time increases slowly respect number of cells
- **Bottleneck:** Data movement



GPU N cells: Results

- **Notable speedup** for translating only Derivative to GPU
- **Working with little GPU capacity**: MB of data in front GB and 4 GPU's available



Conclusions and future work

Conclusions

- MONARCH Chemistry solver consumes 48.6% of the total execution time
- CVODE consumes ~70% of time execution; Derivative and Jacobian ~30%
- GPU parallelization on RXN improves time from 2000 reactions
  - We speedup a bit optimizing memory access (~30%)
  - GPU works better with bigger data sizes

Conclusions

- Computing multiple cells data inside *phlex-chem* improves simple MONARCH test by 12-14 times faster

- GPU with multiple cells gives 1.5x speedup only parallelizing Derivative

-> Improve the general workflow is important as improve the functions

-> GPU optimizes greatly complex modules but need a lot of work

Future work

- Implement optimizations in complex tests
- Implement optimized *phlex-chem* on MONARCH
- Heterogeneous optimization computing CPU & GPU simultaneously
- Improve memory access on RXN: Reduce memory jumps & SPARSE

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thank you