# Parallel Sparse Computation Toolkit: Towards Exascale Linear Algebra

F. Durastante

Wednesday, September 14$^{rd}$ - ITWSHPC-22

Università di Pisa, fabio.durastante@unipi.it
Istituto per le Applicazioni del Calcolo "M. Picone" – CNR

# Collaborators and Funding

Pasqua D'Ambra,
Consiglio Nazionale delle Ricerche
Istituto per le Applicazioni del
Calcolo "M. Picone"

Salvatore Filippone,
Università degli Studi di Roma
"Tor Vergata"
Dipartimento di Ingegneria Civile
e Ingegneria Informatica
IAC-CNR

EoCoE
Project ID: 824158
European Commission
Horizon 2020
European Union funding
for Research & Innovation
textarossa

# What we want to solve

$$\text{Solve}: \; A\mathbf{x} = \mathbf{b},$$

where

- $A \in \mathbb{R}^{n \times n}$ is a very large and sparse matrix $\text{nnz}(A) = O(n)$,
- $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$,

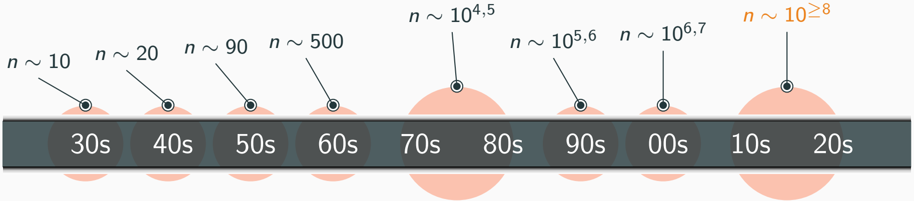
is often the most time consuming computational kernel in many areas of computational science and engineering problems.

# What we want to solve

$$\text{Solve}: \; A\mathbf{x} = \mathbf{b},$$

where

- $A \in \mathbb{R}^{n \times n}$ is a very large and sparse matrix $\text{nnz}(A) = O(n)$,
- $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$.




The exascale challenge, using computer that do $10^{15}$ Flops, targeting next-gen systems doing $10^{18}$ Flops to solve problems with tens of billions of unknowns.

# Parallel Sparse Computation Toolkit – `psctoolkit.github.io`

Two central libraries PSBLAS and AMG4PSBLAS:

- Existing software standards:
  - MPI, OpenMP, CUDA
  - Serial sparse BLAS,
  - (Par)Metis,
  - AMD
- Attention to performance using modern Fortran;
- Research on new preconditioners;
- No need to delve in the data structures for the user;
- Tools for error and mesh handling beyond simple algebraic operations;
- Standard Krylov solvers

# Parallel Sparse Computation Toolkit – psctoolkit.github.io

Two central libraries PSBLAS and AMG4PSBLAS:

- Domain decomposition preconditioners of Schwartz type
- Algebraic multigrid with aggregation schemes
  - Parallel coupled Weighted Matching Based Aggregation
  - Smoothed Aggregation (Vaněk, Mandel, Brezina)
- Parallel Smoothers (Block-Jacobi, DD-Schwartz, Hybrid-GS/SGS/FBGS, $\ell_1$ variants) that can be coupled with specialized block (approximate) solvers MUMPS, SuperLU, incomplete factorizations ((H)AINV, (H)INVK/L, (H)ILU-type)
- V-Cycle, W-Cycle, K-Cycle

# Parallel Sparse Computation Toolkit – `psctoolkit.github.io`

- Opensource code, BSD3 License:

  "free as in free beer and as in free speech."

- **Integrated\interfaced** with

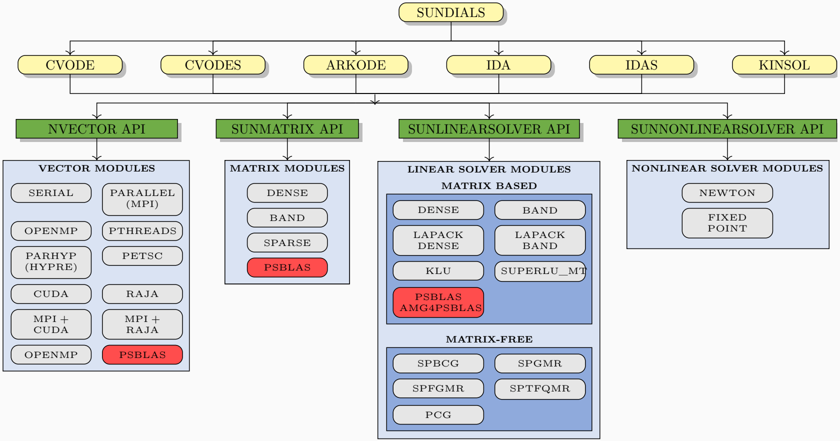
Alya - High Performance Computational Mechanics
Barcelona Supercomputing Center

SUNDIALS
SUite of Nonlinear and DIfferential/ALgebraic equation Solvers.
Lawrence Livermore National Laboratory

- **Available** from GitHub repository, packaged for Fedora and Centos, and *recently* for Spack.
- **Website:** `https://psctoolkit.github.io/`

# The SUNDIALS/KINSOL Software Framework

The SUNDIALS integration has been implemented to have a **Newton solver** using **our Krylov methods** and **preconditioner**.

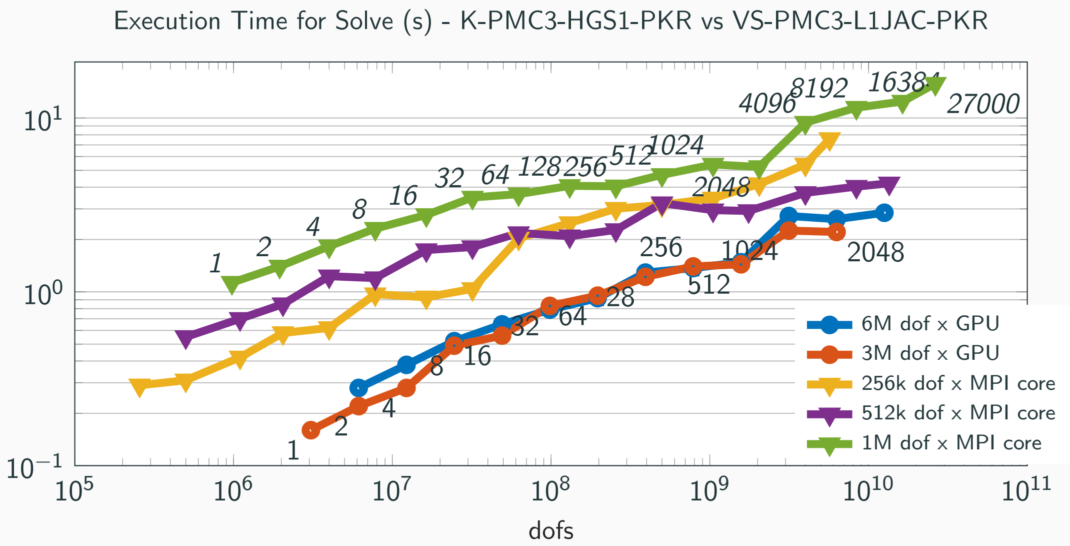


# Weak Scalability - CPU/GPU Runs - Piz Daint

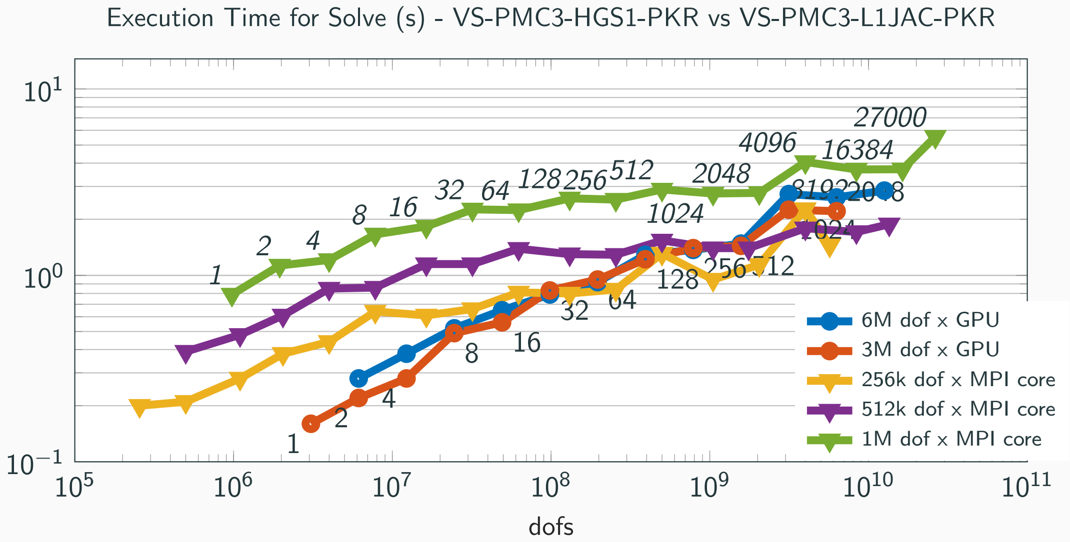
**Weak scaling:** In case of weak scaling, both the **number of processors** and the **problem size** are **increased**. This also results in a *constant workload per processor*.

- Run on the Piz Daint machine up to 27000 cores and 2048 GPUs
- Test: 3D Constant coefficient Poisson Problem with Flexible Conjugate Gradient
- DoF: 256k/512k/1M unknown $\times$ MPI core and 3M/6M per GPUs
- Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect, NVIDIA Tesla P100 (23rd TOP500 June'22)
- Measures: execution time for solve

# Weak Scalability - CPU/GPU Runs - Piz Daint



# Weak Scalability - CPU/GPU Runs - Piz Daint



# Two applications: wind and water

Herbert Owen

Barcelona Supercomputing Center

- Solution of 3D incompressible unsteady **Navier-Stokes equations** for the **L**arge **E**ddy **S**imulations of **turbulent flows**,
- Alya
- MareNostrum - Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path, Lenovo (82nd TOP500 June'22)

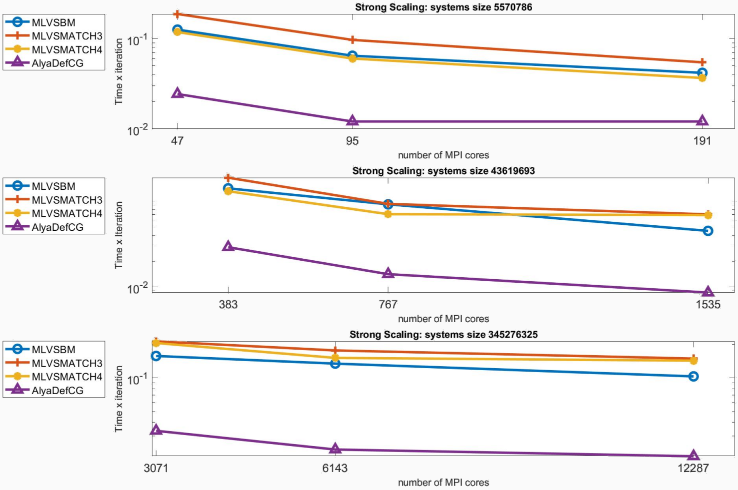
Stefan Kollet

Research Centre Jülich

- Simulation of three-dimensional **groundwater flow** with overland flow via **Richards equation**,
- SUNDIALS/KINSOL & ParFlow
- Marconi-100 - IBM Power System AC922, IBM POWER9 16C 3GHz, Nvidia Volta V100, Dual-rail Mellanox EDR Infiniband, IBM (21st TOP500 June'22)

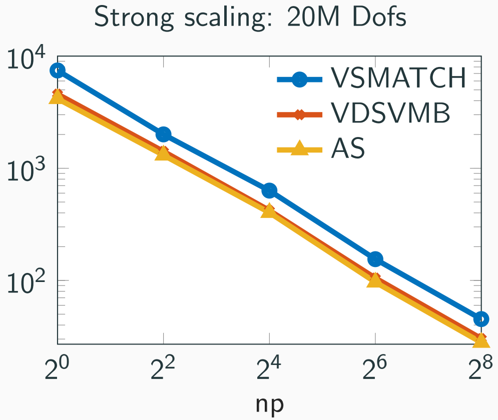
Getting **scalability** on thousand of processors and **better global solution timings**.

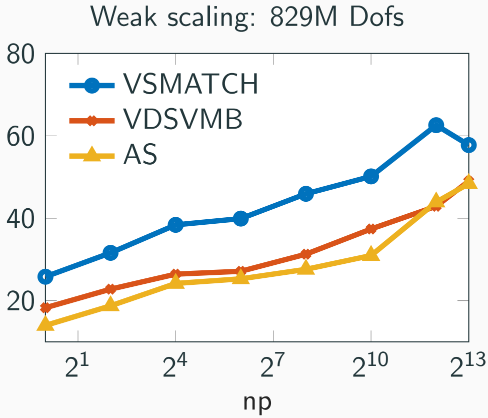
# Strong scaling: solution of the pressure equation




- Measuring **time per single iteration**.
- The trade-off between cost-per-iteration and number of iterations advantages PSCToolkit over Alya old solvers.
- 47 to 12887 cores, different global loads.

# Strong and weak scaling: global solution




1 to 256 cores

1 to 8192 cores

**Total time to solution** (seconds) using different preconditioning strategies.

# Concluding remarks

- ✔ A *suite* of libraries already working on *large* and *real life* test cases,
- ✔ a proved ease of integration and interfacing within other scientific libraries,
- ✔ working on different architectures and software environments,
- ✔ a tool for developing new algorithms.

Currently working on:

- Hybrid OpenMP/MPI parallelism inside preconditioner assembly routines,
- communication avoiding algorithms.

Latest references (detailed bibliography on `psctoolkit.github.io`):


- P. D'Ambra, F. D. and S. Filippone, AMG preconditioners for linear solvers towards extreme scale, SIAM J. Sci. Comput. **43** (2021), no. 5, S679–S703.
- D. Bertaccini, P. D'Ambra, F. D. and S. Filippone, Why diffusion-based preconditioning of Richards equation works: spectral analysis and computational experiments at very large scale (2022), arXiv:2112.05051.
- H. Owen, G. Houzeaux, F. D., S. Filippone, P. D'Ambra, AMG4PSBLAS Linear Algebra Package brings Alya one step closer to Exascale. *In preparation*