Università degli Studi della Campania *Luigi Vanvitelli*
Scuola Politecnica e delle Scienze di Base
Dipartimento di Matematica e Fisica

Algebraic multigrid for HPC: some results from the EoCoE project

Daniela di Serafino
Università della Campania "L. Vanvitelli", Caserta, Italy
daniela.diserafino@unicampania.it

Inaugurazione del supercalcolatore CRESCO6
Centro Ricerche ENEA, Portici, 30.05.2018

EoCoE Project – www.eocoe.eu (more in M. Celino's talk)

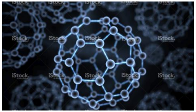
EoCoE Energy oriented Centre of Excellence for computing applications

Horizon 2020 Programme (Project ID: 676629)
21 institutions from 8 European countries, Oct. 2015 - Sept. 2018

Overall goal: exploit the tremendous potential offered by the ever-growing computing infrastructure to foster and accelerate the European transition to a reliable and low-carbon energy supply

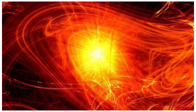
4 thematic application pillars, each addressing a specific research community

meteorology for energy



materials for energy

water for energy



fusion for energy

EoCoE Project (cont'd)

Transversal basis

Transversal multidisciplinary effort in Computational Mathematics and High Performance Computing for tackling algorithmic and software technology challenges across the application pillars

(applied maths, numerical algorithms, programming tools & techniques, parallel I/O)

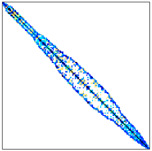
Numerical Linear Algebra

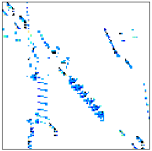
- Ubiquitous in EoCoE applications ... and in Computational Science and Engineering
- Substantial fraction of overall simulation time

$\Longrightarrow$ Specific task in the transversal basis

The ubiquitous large and sparse linear system $Ax = b$

- Large: hundred millions to billions of unknowns
- Sparse: matrices with enough zeros that it pays to take advantage of them (J.H. Wilkinson)





pictures from SuiteSparse Matrix Collection (https://sparse.tamu.edu)

- Must be solved many times in a single application code
- Most used methods: Krylov solvers with (usually mandatory) preconditioners
  $\Longrightarrow$ memory-bound computations, variable memory access pattern, collective operations, high data dependency, ...

Intense and continuous activity for the development of efficient solvers and preconditioners for high-end HPC architectures

Italian Linear Algebra Team in EoCoE

Ambra Abdullahi Hassan, University of Rome “Tor Vergata”, Rome, Italy

Valeria Cardellini, University of Rome “Tor Vergata”, Rome, Italy

Pasqua D’Ambra, Institute for Applied Computing (IAC) of the Italian National Research Council, Naples, Italy

Daniela di Serafino, University of Campania “L. Vanvitelli”, Caserta, Italy

Salvatore Filippone, Canfield University, Cranfield, UK (initially at University of Rome “Tor Vergata”)

- Focus on Algebraic MultiGrid (AMG) preconditioned Krylov solvers
- Work performed and in progress: extension and improvement of PSBLAS and MLD2P4 sparse linear algebra libraries, prompted by needs of modern HPC architectures and EoCoE applications

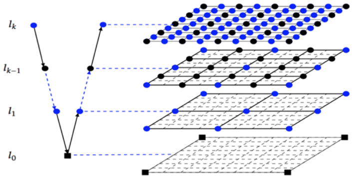
Algebraic MultiGrid (AMG)

- Basic idea of MG methods: using a sequence of coarse grids to accelerate the fine grid solution

pre-smoothing:
$x = x + S(b - Ax)$

residual restriction:
$r_c = P^T(b - Ax)$

coarse-grid solution
$A_c e_c = r_c$



post-smoothing:
$x = x + S^T(b - Ax)$

error interpolation
and solution update
$x = x + Pe_c$

- AMG: the construction of the MG hierarchy is carried out using only information from the matrix and not from the geometry of the problem

- Optimal convergence (linear complexity) and good scaling potential, but exposing high parallelism is not easy at "too coarse" levels

PSBLAS: Parallel Sparse BLAS

A library for sparse matrix computations on parallel machines

(Sw project started by S. Filippone et al. at the end of 90s, prompted by work of I. Duff et al. on Sparse BLAS standard, ACM TOMS '97, '02)

Current version (v. 3.5)

- Parallel sparse matrix operations and data management, Krylov solvers
- General row-block matrix distribution, support infrastructure for mesh handling and sparse matrix I/O
- Data allocation through graph partitioning (METIS, ParMETIS, SCOTCH)
- Object-oriented design in Fortran 2003, C and Octave interfaces in progress
- Message-passing paradigm (MPI), plugin available for NVIDIA Cuda
- Internal matrix representation/storage: distributed sparse matrix in CSR/CSC/COO format in the base library, extension plugins available for ELLPACK, JAD and GPU-enabled formats

Freely available from https://github.com/sfilippone/psblas3

(Filippone & Colajanni, ACM TOMS '00; Filippone & Buttari, ACM TOMS '12; Cardellini et al., Scientific Programming '14; Filippone et al., ACM TOMS '17)

MLD2P4: MultiLevel Domain Decomposition Parallel Preconditioners Package based on PSBLAS

A library of parallel AMG preconditioners

(Sw project started by P. D'Ambra, D. di Serafino and S. Filippone in 2004)

Current version (v. 2.2)

- Initially developed as a package of algebraic multigrid Schwarz preconditioners, now implements more general AMG preconditioners
- Object-oriented design in Fortran 2003, layered sw architecture on top of PSBLAS
- Clear separation between interface and implementation of methods
- A wide variety of cycles, smoothers, coarsest-level iterative and direct solvers, including interfaces to third-party software (MUMPS, UMFPACK, SuperLU, SuperLU_dist)
- Coarsening based on decoupled smoothed aggregation, first experiments with decoupled weighted matching (sequential version by D'Ambra & Vassilevski, '13, '18)
- Plugin for approximate inverses (Bertaccini & Filippone, CAWMA '16)

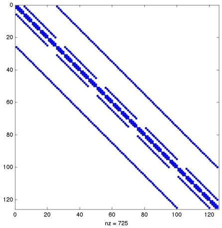
Freely available from https://github.com/sfilippone/psblas3

(D'Ambra, di Serafino & Filippone, APNUM '07, ACM TOMS '10, CAMWA '13; Abdullahi, D'Ambra, di Serafino & Filippone, LNCS '18)

Application to linear systems from EoCoE: an example

- Simulations with Parflow model at Jülich Supercomputing Centre (JSC, S. Kollet): filtration through variably saturated porous media for incompressible flows (3D Richard eqns, finite-difference discretization, Newton-Krylov solver)

- Simplified 3D model: $-\nabla \cdot \mathbf{K} \nabla p = f$
on unit cube with no-flow boundary conditions



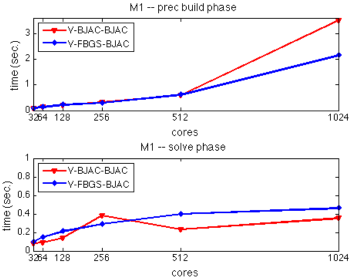
- Anisotropic conductivity tensor randomly generated from lognormal distribution with mean $\mu = 1$ and variable standard deviation $\sigma = 1, 2, 3$

- Cell-centred discretization on Cartesian grid

Some results on CRESCO4 cluster, operated by ENEA

- Row-block distribution of the matrix via Metis graph partitioner, 15500 matrix rows per core, $\sim$16 million rows (DoFs) on 1024 cores
- Conjugate Gradient with AMG V-cycle preconditioner (smoothed aggregation, forward/backward hybrid Gauss-Seidel (FBGS) and Block Jacobi (BJAC) smoothers, 10 BJAC sweeps as coarsest-level solver)

Weak scaling (preconditioner build phase and solve phase)



CRESCO4: 64 nodes available, each consisting of 2 sockets with 8-core Intel Xeon E5-2670 procs @2.6 GHz and 64 GB RAM, connected via Infiniband

Work in progress: MLD2P4 extension to GPUs

Some issues

- Improvement of GPU support to different sparse matrix storage schemes, especially for sparse matrix by vector product (based on design patterns, see Cardellini et al., Scientific Programming '14, and Filippone et al., ACM TOMS '17)

- Exploitation of the Sparse Approximate Inverse plugin as local solver for block Jacobi smoothers and solvers (much larger amount of parallelism exposed by sparse matrix-vector products as compared to sparse triangular solves)

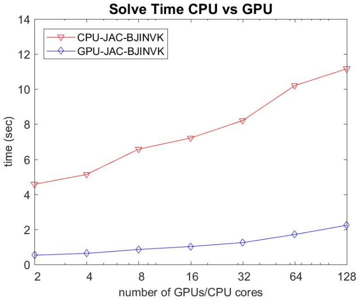
Preliminary implementation of solve phase available

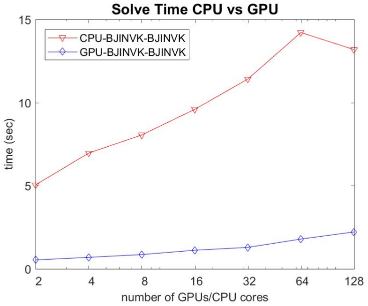
Abdullahi Hassan, D'Ambra, di Serafino & Filippone, Tech. Rep. '18

Very preliminary results on JURECA, operated by JSC

- Row-block distribution of the matrix obtained by 3D decomposition of the grid, 2×10^6 matrix rows per core, up to 2.56×10^8 rows (DoFs) on 128 GPUs
- Conjugate Gradient with AMG V-cycle (smoothed aggregation, point or block Jacobi smoother, 10 block-Jacobi sweeps as coarsest-level solver, with sparse approximate inverse local solver (BJINVK))

Weak scaling (preconditioner build phase and solve phase)





JURECA: 1872 compute nodes with 2 Intel Xeon E5-2680 v3 Haswell CPUs per node, 75 compute nodes equipped with 2 NVIDIA Tesla K80 GPUs, Infiniband connection

Thank you for your attention!