Comparative Performance Modeling of Parallel Preconditioned Krylov Methods

Kanika Sood, Boyana Norris

University of Oregon

ABSTRACT

Many scientific and engineering computations rely on the scalable solution of large sparse linear systems. Preconditioned Krylov methods offer many algorithmic choices with varying performance depending on the linear system's properties. We analytically compare the communication costs of parallel Krylov methods [1], offered by PETSc [2] to produce scalability rankings of solution methods. We combine this ranking with a machine learning performance model [3] and then demonstrate the use of the combined model to select solver configurations in an application simulating driven fluid flow in a cavity.

PROBLEM

- Optimal numerical method selection is challenging.
- Preconditioned Krylov methods with the same complexity perform differently for different inputs.

SOLUTION

We analyze the communication overheads of Krylov methods to generate a communication-based solver ranking. For small scales, we use our Machine Learning (ML) model [4] to suggest solvers. For large scales we combine the ML model and the communication-based ranking to produce solver recommendations.

REFERENCES

[1] Saad, Yousef. *Iterative methods for sparse linear systems*. SIAM, 2003.

[2] Balay, Satish, et al. PETSc web page, http://www.mcs.anl.gov/petsc/ 2017.

[3] Jessup, Elizabeth, et al. *Performance-based numerical solver selection in the Lighthouse framework.* SIAM Journal on Scientific Computing, 2016.

[4] Alpaydin, Ethem. *Introduction to machine learning*. MIT 2017.

[5] Motter, Pate, et al. *Lighthouse: An automated solver selection tool.* Software Engineering for High Performance Computing in Computational Science and Engineering (SEHPCCSE), 2015.

[6] Sood, Kanika et al. *Lighthouse: A taxonomy-based solver selection tool.* Proceedings of the Second Workshop on Software Engineering for Parallel Systems (SEPS), 2015.

[7] Lighthouse Project, http://lighthousehpc.github.io/lighthouse/, 2017.

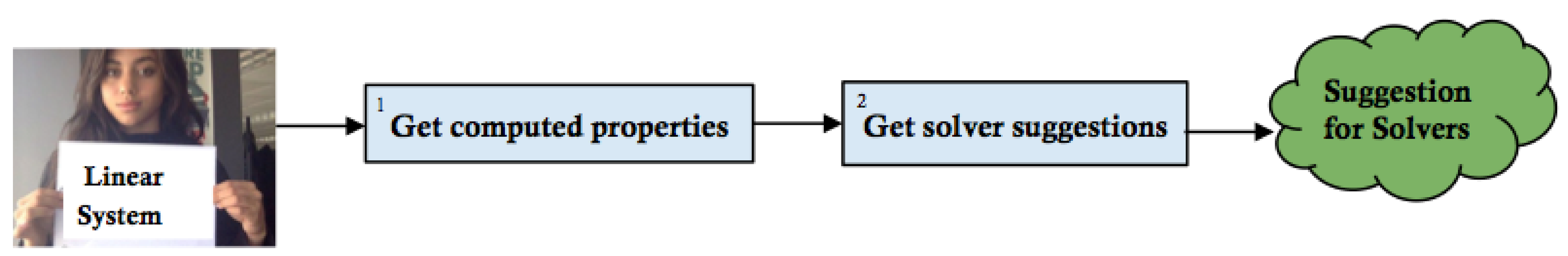
METHODOLOGY

- Model two aspects of the performance:
1. Model convergence using an ML approach (see ML Model Construction)
2. Model communication by considering the operations below.

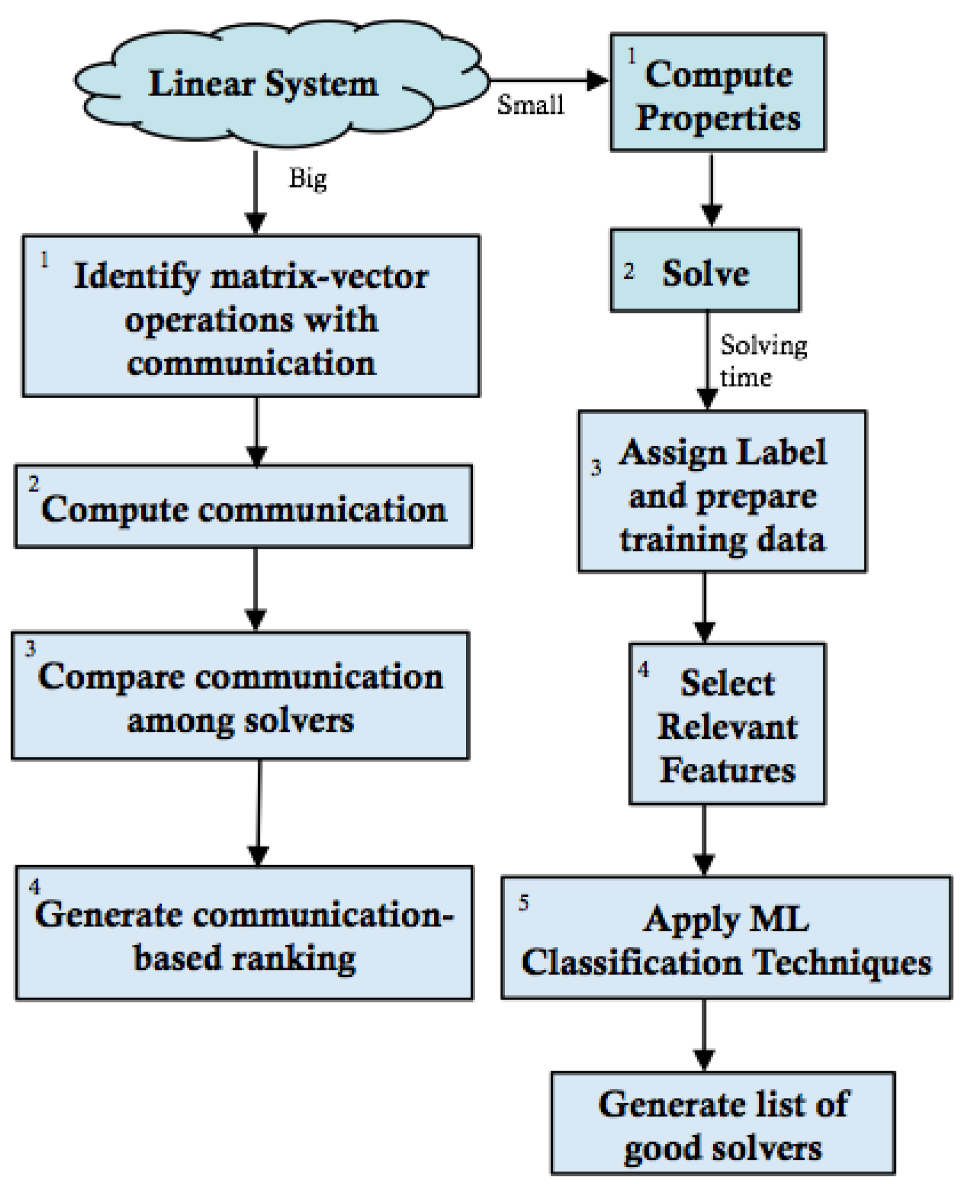
Operation	Description
MatMult	Computes matrix-vector product: $y = Ax$
MatMultTranspose	Computes matrix transpose times a vector $y = A'x$
VecNorm	Computes norm of the vector: $r = \|\|x\|\|$
VecDot	Computes the dot product of the vectors x and y
VecMDot	Computes one or more vector dot products.
VecMDot_MPI	Computes vector multiple dot products and performs reductions
VecTDot	Computes indefinite vector dot product: $y^H x$, where y^H denotes the conjugate transpose of vector y
VecDotNorm2	Computes the inner product of two vectors and the 2-norm squared of the second vector
PCApply	Performs the preconditioning on the vector
PCApplyTranspose	Applies the transpose of preconditioner to a vector
VecScatterBegin	Performs a scatter from one vector to another
MPIU_Allreduce	Determines if the call from all the MPI processes occur from the same location in the code.

Model-based solver selection: Given a new sparse linear system, we

1. Apply the ML model to obtain a list of "good" solver configurations [5-7].
2. If solving large-scale problems (>1000 cores), find the top-ranked methods within that set based on the communication model.



ML MODEL CONSTRUCTION



Training dataset: 1836 square matrices (University of Florida Sparse Matrix Collection), up to 484,481 rows, up to 79,936,000 nonzeros.

Solver timing results: 97,117; labeled "good": 56,464; labeled "bad": 40,651

RESULTS

Application Description

- Driven cavity flow simulation.
- Combination of lid-driven flow and buoyancy-driven flow.
- Non-linear PDEs:

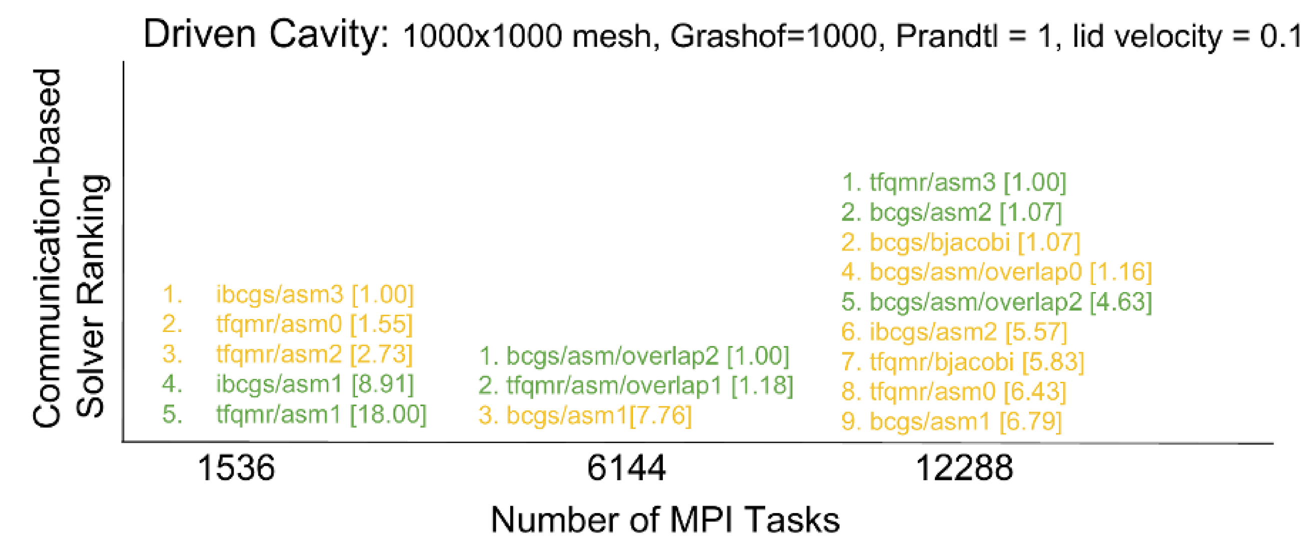
$-\Delta U - \nabla_y \Omega = 0$

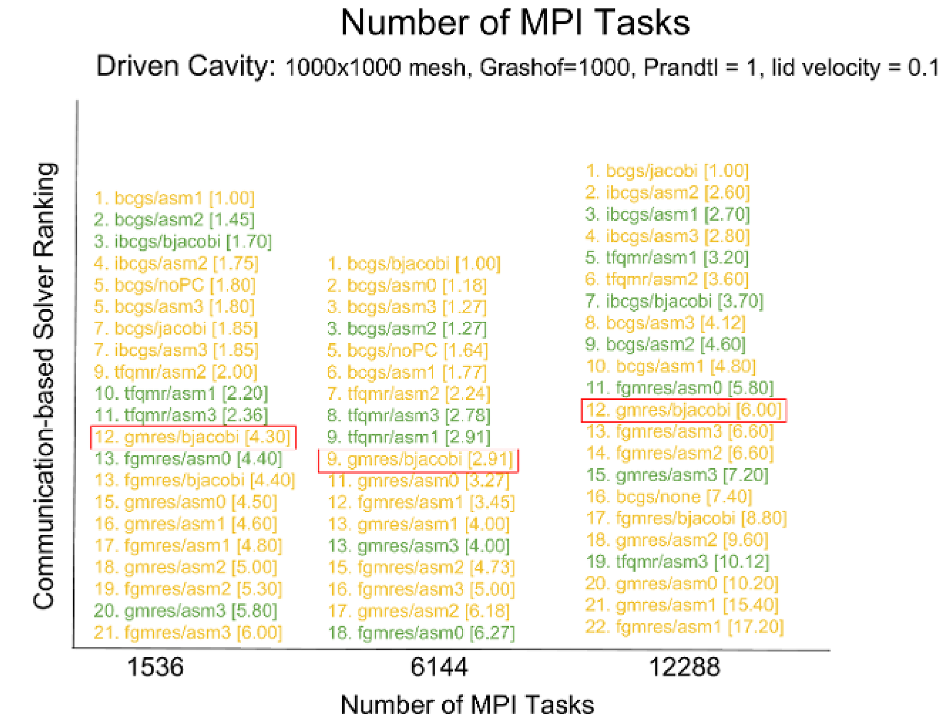
$-\Delta V + \nabla_x \Omega = 0$

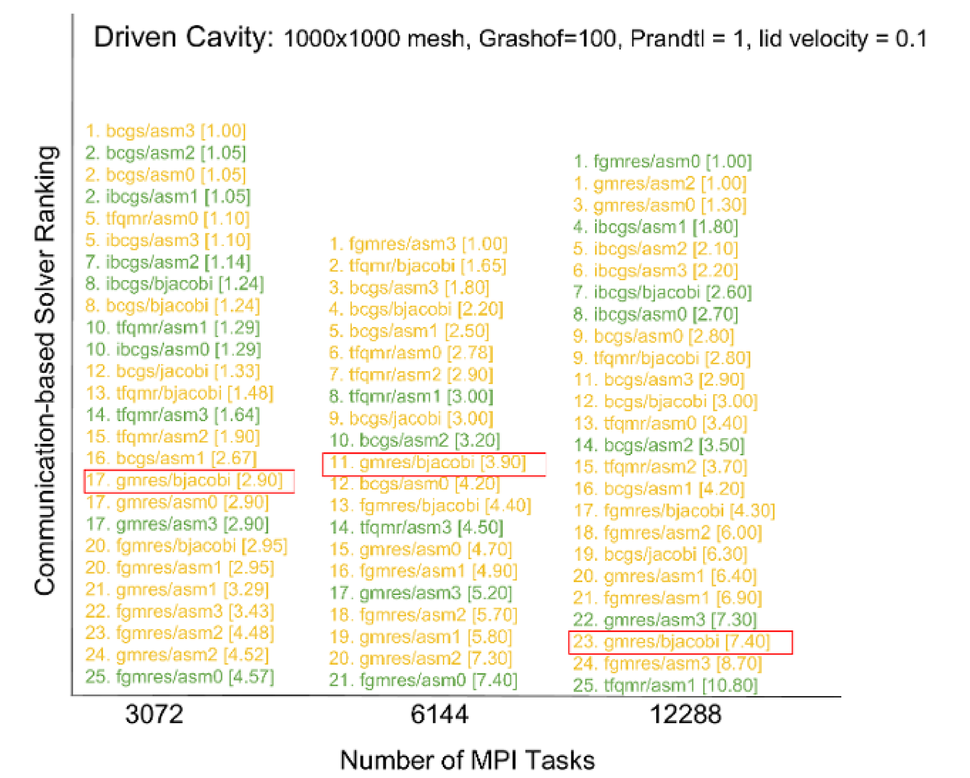
$-\Delta\Omega + \nabla.([U*\Omega, V*\Omega]) - GR * \nabla_x T = 0$

$-\Delta T + PR*\nabla.([U * T, V * T]) = 0$

- The system is discretized by using finite differences with a 5-point stencil on a uniform 2D Cartesian mesh, with four unknowns per mesh point (2D velocity, vorticity, temperature).







CONCLUSIONS

- Performance-based parallel preconditioned Krylov method selection through a new comparative modeling approach for parallel Krylov methods.
- Demonstrated solver selection on a PDE-based numerical simulation.

Acknowledgment

CCF-1219089, CNS-0821794

U.S. DEPARTMENT OF ENERGY

DE-AC02-06CH11357, DE-AC02-05CH11231