RACE: Speeding Up Sparse Iterative Solvers Using Cache Blocked Matrix Power Kernels

Christie Alappat[1], Georg Hager[1], Jonas Thies[2], Gerhard Wellein[1]

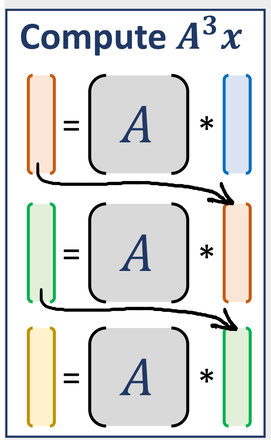
[1]Friedrich-Alexander-Universität, Erlangen-Nürnberg

[2]Delft University of Technology, Netherlands

Can we accelerate MPK?

MPK computes $A^p x$, where A is a sparse matrix. It is a hotspot in many iterative solvers.

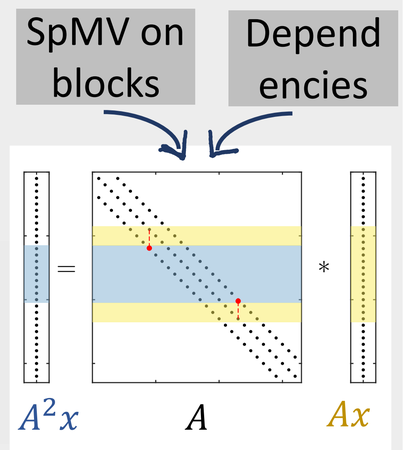
State-of-the-art MPK



- Back-to-back SpMVs (Ax operation).
- SpMV is memory-bound.
- Load matrix A from memory p times.

Can I cache matrix A?

Same matrix A used for all SpMVs.



Existing techniques

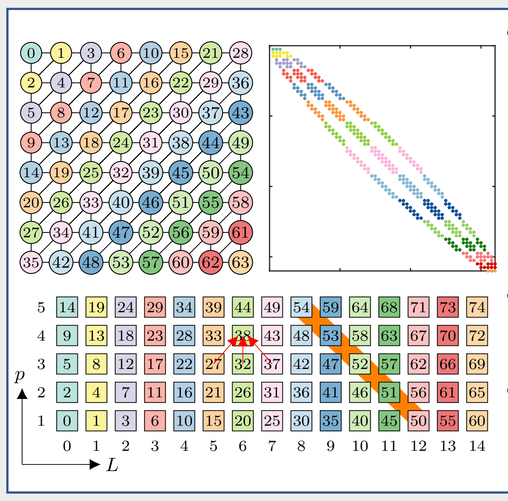
- Challenging on general matrix [1].
- Halo overhead ∝ parallelism [2].

RACE can resolve these problems [3].

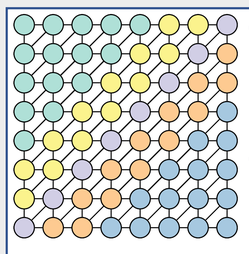
Highlights

- Algebraic
- No halo overhead
- Hardware-aware
- High efficiency

Cache blocking MPK using RACE



- Blocks for caching determined using breadth first search (BFS) levels.
- High data locality.
- Levels are self-aware of SpMV dependencies.



- Level groups (LG) formed from BFS levels to reduce synchronization.
- Point-to-point synchronization instead of global barriers.

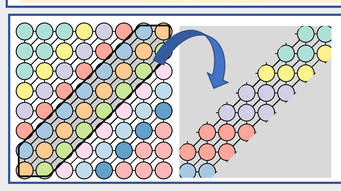
Analytical formula for LG:

$12\times(p+1)\times N_{nz}(T(i))\ bytes < C$

$A^p x$ Size of LG Cache size

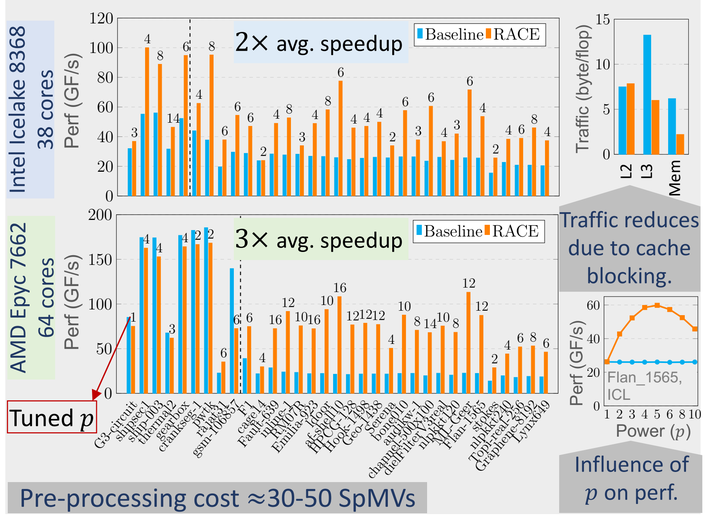
Application

Hardware



- Recursion, if required, to fit levels into cache.

Performance of RACE MPK



Traffic reduces due to cache blocking.

Influence of p on perf.

Pre-processing cost ≈30-50 SpMVs

TRILINOS Integration to solvers and preconditioners.

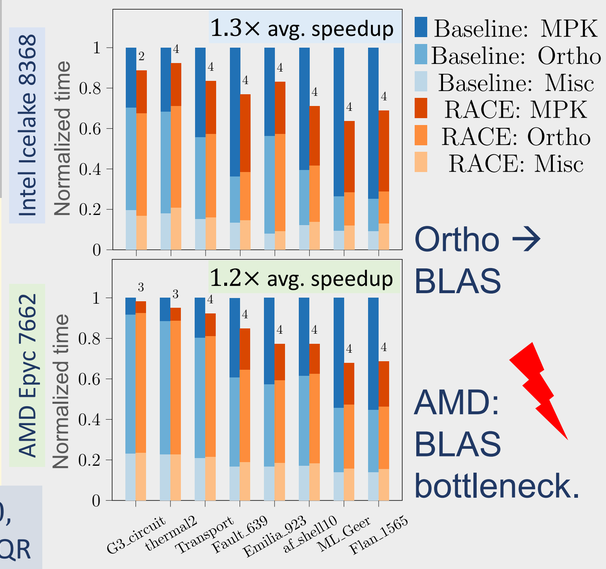
Use of RACE does not change convergence. Bitwise accurate.

s-step GMRES solver

Generate Krylov subspace

$\mathcal{K}(A,b) = span\{b, Ab, A^2b, \dots, A^m b\}$

```
//setup
for j=0:s:m-1
  MPK; //A^s x
  Ortho;
end for
//cleanup
```



Ortho → BLAS

AMD: BLAS bottleneck.

Belos library [4]

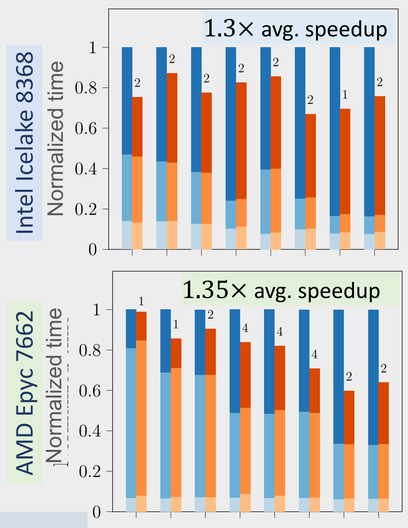
$s = 4, m = 50$, Ortho: ICGS+TSQR

Combining preconditioners with MPK

Consider dependencies between multiple operations for cache blocking.

$M^{-1} \to AM^{-1} \to M^{-1}AM^{-1} \to \cdots$

Ortho cost reduces.



Belos s-step GMRES

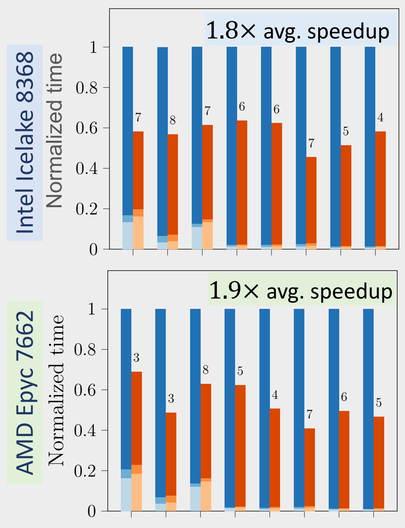
Ifpack2 [4] GS2 ($\gamma = 2$) precon

Two-stage Gauss-Seidel [5] (GS2)

Polynomial preconditioners are perfect for MPK.

$M^{-1}x = \wp(A) = \lambda_0 + \lambda_1 Ax + \lambda_2 A^2 x + \cdots$

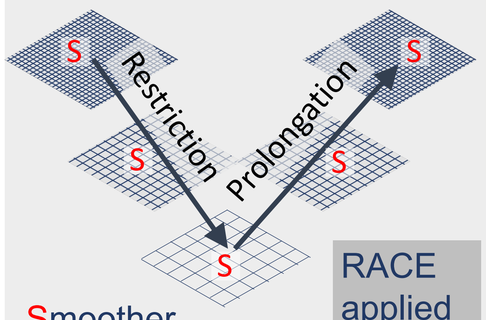
It requires high power in MPK ➔ RACE extremely effective.



GMRES polynomial precon [6] -Belos

Jacobi + Polynomial preconditioner (deg 80)

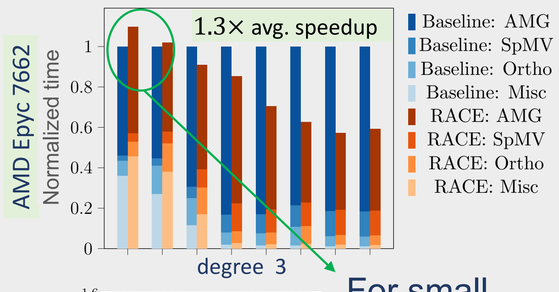
Algebraic multigrid (AMG)

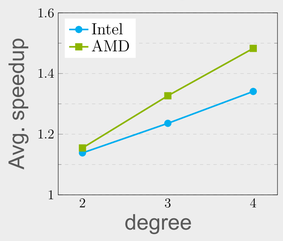


Smoother involves repeated application of the same operation.

➔ RACE can block the operation in cache.

RACE applied only to finest level.





For small matrices, only few iterations.

➔ RACE's pre-processing cost cannot be amortized.

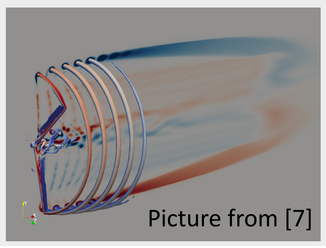
Muelu library [4]

SA-AMG with Chebyshev smoother

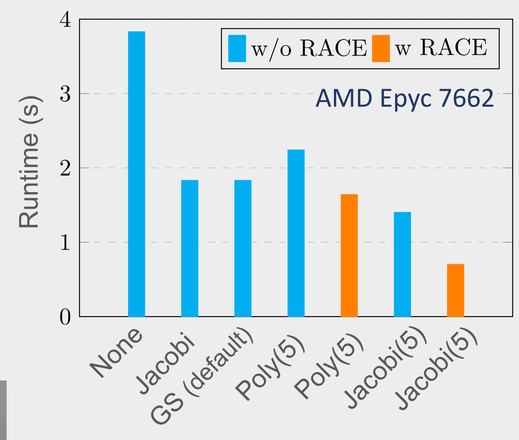
Showcase: Nalu-Wind

Nalu-Wind [7] is a CFD code used for simulation of wind turbines.

Nalu-Wind solves Navier-Stokes equation on an unstructured grid.



Picture from [7]



Solving largest linear system of equation in Nalu-Wind, i.e., momentum equation, with various preconditioners.

Outlook

- RACE MPK very effective: up to 4x speedup.
- Benefits various iterative solvers.
- Can be combined with preconditioners.
- Adds another dimension to solver tuning.

Future work

- Distributed MPI parallel version coming soon (Q3 2023).
- Feasibility study on GPUs under progress.

References

[1] D. Huber, M. Schreiber, and M. Schulz, ``Graph-based multi-core higher-order time integration of linear autonomous partial differential equations'', Elsevier JCS, 2021, 10.1016/j.jocs.2021.101349

[2] M. Mohiyuddin, M. Hoemmen, J. Demmel, K. Yelick, ``Minimizing Communication in Sparse Matrix Solvers'', SC '09, 2009, 10.1145/1654059.1654096

[3] C. Alappat, G. Hager, O. Schenk and G. Wellein, ``Level-Based Blocking for Sparse Matrices: Sparse Matrix-Power-Vector Multiplication'', IEEE TPDS, 2023, 10.1109/TPDS.2022.3223512

[4] The Trilinos Project Website, https://trilinos.github.io

[5] Thomas et al., ``Two-Stage Gauss-Seidel Preconditioners and Smoothers for Krylov Solvers on a GPU Cluster'', 2021, https://arxiv.org/pdf/2104.01196.pdf

[6] J.A. Loe, H.K. Thornquist, and E.G. Boman, ``Polynomial Preconditioned GMRES in Trilinos: Practical Considerations for High-Performance Computing'', SIAM PP, 2020, 10.1137/1.9781611976137.4

[7] M A Sprague, S Ananthan, G Vijayakumar, and M Robinson, ``ExaWind: A multifidelity modeling and simulation environment for wind energy'', Journal of Physics, 2020, 10.1088/1742-6596/1452/1/012071

Acknowledgement

Bundesministerium für Bildung und Forschung

STROEMUNGSRAUM SCALEXA HPC DEEPTECH

NHR FAU

https://github.com/RRZE-HPC/RACE