6th AsHES workshop

May 26th, 2016, Chicago, USA

Efficiency of general Krylov methods on GPUs – An experimental study

H. Anzt, M. Kreutzer, M. Koehler, G. Wellein, J. Dongarra

Piotr Luszczek

Solving large sparse linear systems on GPUs

http://blog.heltontool.com/category/tools/

- Large variety of Iterative methods
- Krylov solvers work good for many problems
- Efficiency depends on problem characteristics
  - eigenvalue distribution
  - diagonal dominance
  - definiteness

- **Black-Box Scenario:** Problem characteristics are not known.

The Shotgun Approach

Run **multiple Krylov solvers** simultaneously as **poly-iterative** method

- **Theoretical benefits**
  - benefit from the fastest convergence
  - drop solvers that break down

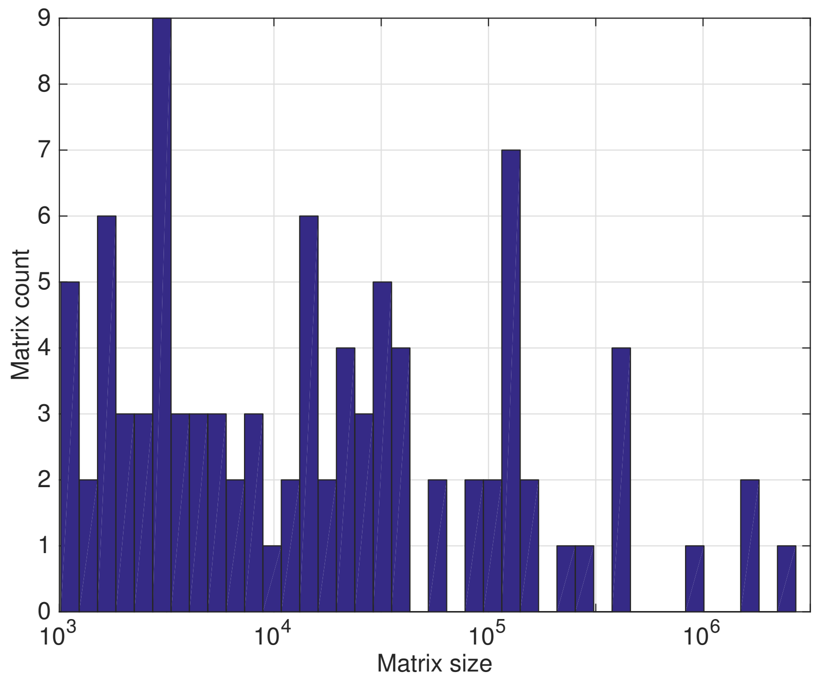
- **Computational benefits**
  - Runtime overhead small for solvers with similar structure
    - SpMM replaces SpMV to generate multiple Krylov subspaces
    - Interleaving global communication for low synchronization count
    - Enhanced fault-tolerance

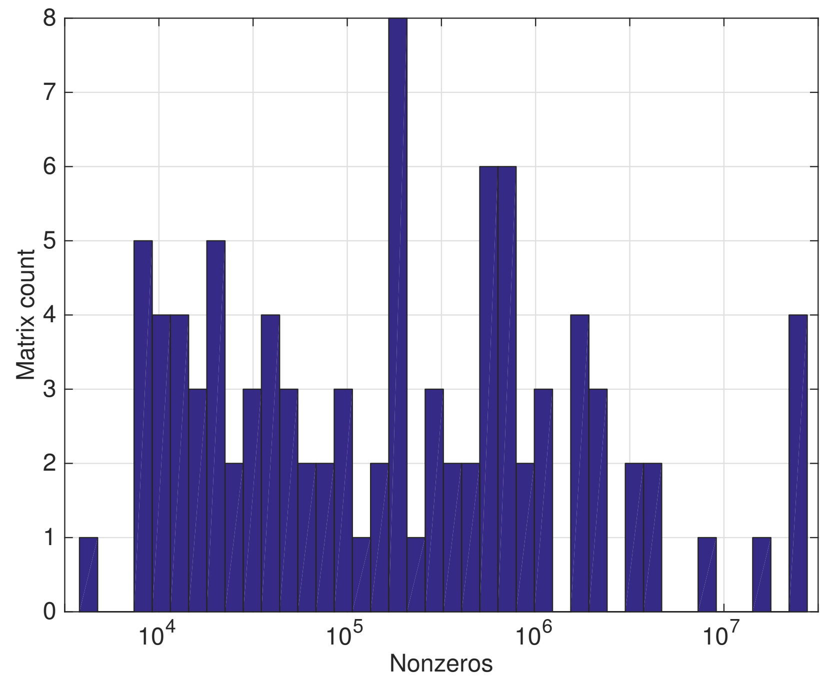
- **Limitation: Solvers are required to have similar structure (SpMV/reduction)**

Barrett et al.: Algorithmic bombardment for the iterative solution of linear systems: A poly-iterative approach, *Journal of Computational and Applied Mathematics 74, 1996.*

Contribution

- Run different **Krylov methods** on large number of test matrices
- Analyze with different target metrics: **Convergence, SpMV, Runtime**
- Non-symmetric test matrices from University of Florida Matrix Collection
  - 1,000 < n < 5,000,000; nnz<100,000,000
  - At least one of the considered methods converges within 2n SpMV
  - 94 non-symmetric test matrices in total

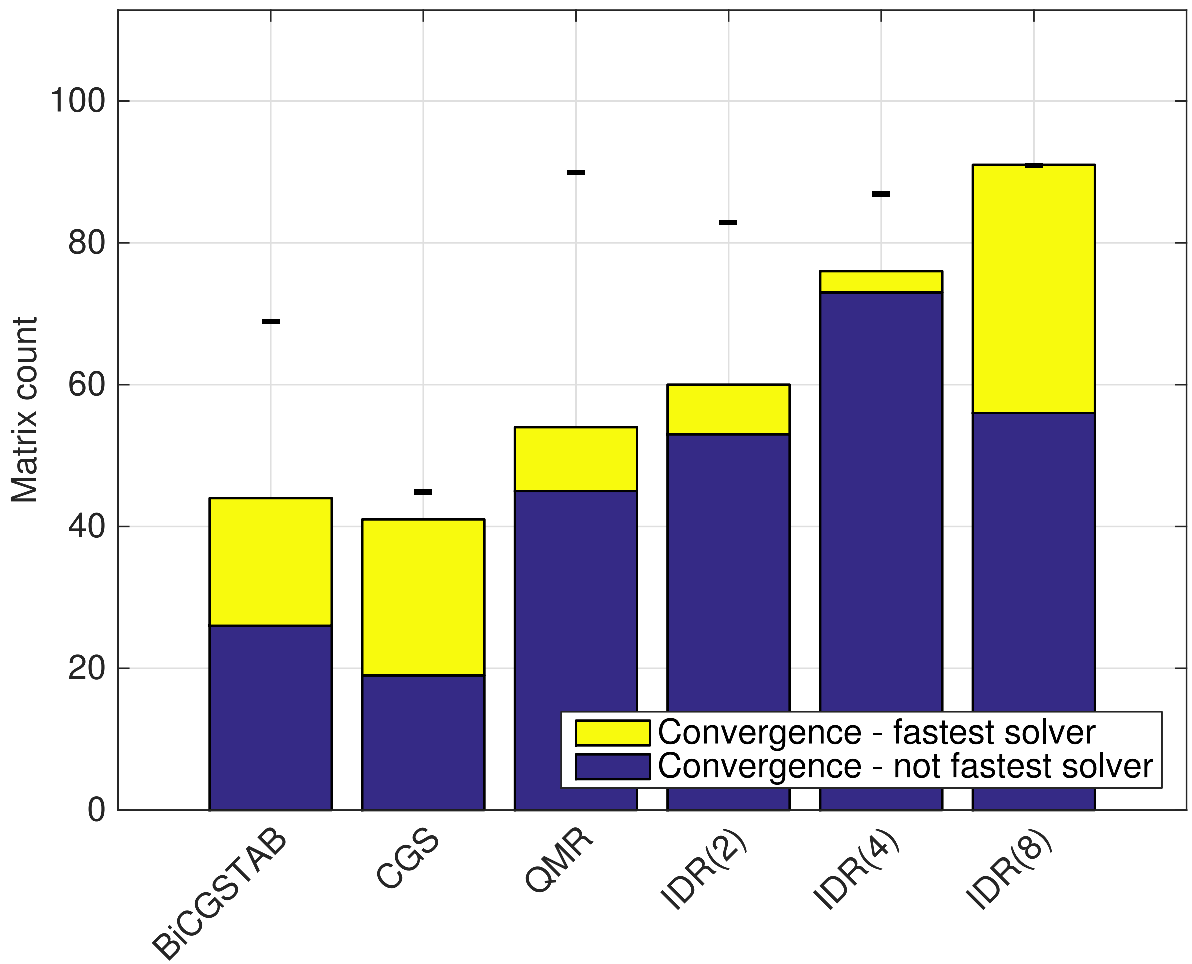




Experiment setup

- libufget
  - C - interface to access matrices at UFMC
  - Max Planck Institute for Dynamics of Complex Technical Systems
- MAGMA
  - Accelerator-focused linear algebra software library
  - Dense and sparse linear algebra routines, solvers, eigensolvers
  - We choose: BiCGSTAB, CGS, QMR, IDR(2), IDR(4), IDR(8)
  - University of Tennessee
- NVIDIA K40 GPU
  - 1,682 GFlop/s (double precision).
  - 12 GB; 288 GB/s (theoretical) – 193 GB/s (experimentally)
  - CUDA v. 7.5
- **Solver setting**
  - Solve: $A\,x = b$ for $b \equiv 1$ starting with $x \equiv 0$
  - Relative residual stopping criterion: $10^{-10}\,|b|$

Solver Robustness – The Convergence Metric



The Shotgun Approach

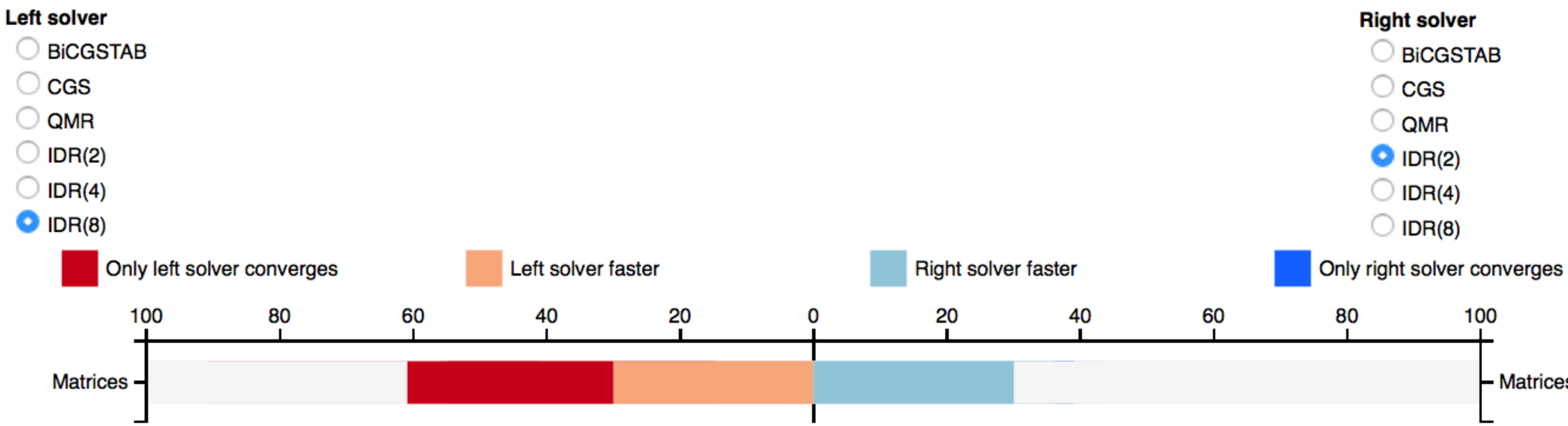
Run multiple Krylov solvers simultaneously
as poly-iterative method

- Original work: poly-iterative solver with BiCGSTAB, QMR, CGS
- IDR(s) structurally different, hard to combine in simultaneous fashion

Barrett et al.: Algorithmic bombardment for the iterative solution of linear systems: A poly-iterative approach, *Journal of Computational and Applied Mathematics 74, 1996.*

Solver Orthogonality w.r.t. Problem Suitability

- *Which methods to include in Multi-Iterative solver?*



Combination solves 91 of 94 systems.

http://www.icl.utk.edu/~hanzt/solver_ortho/

The Shotgun Approach

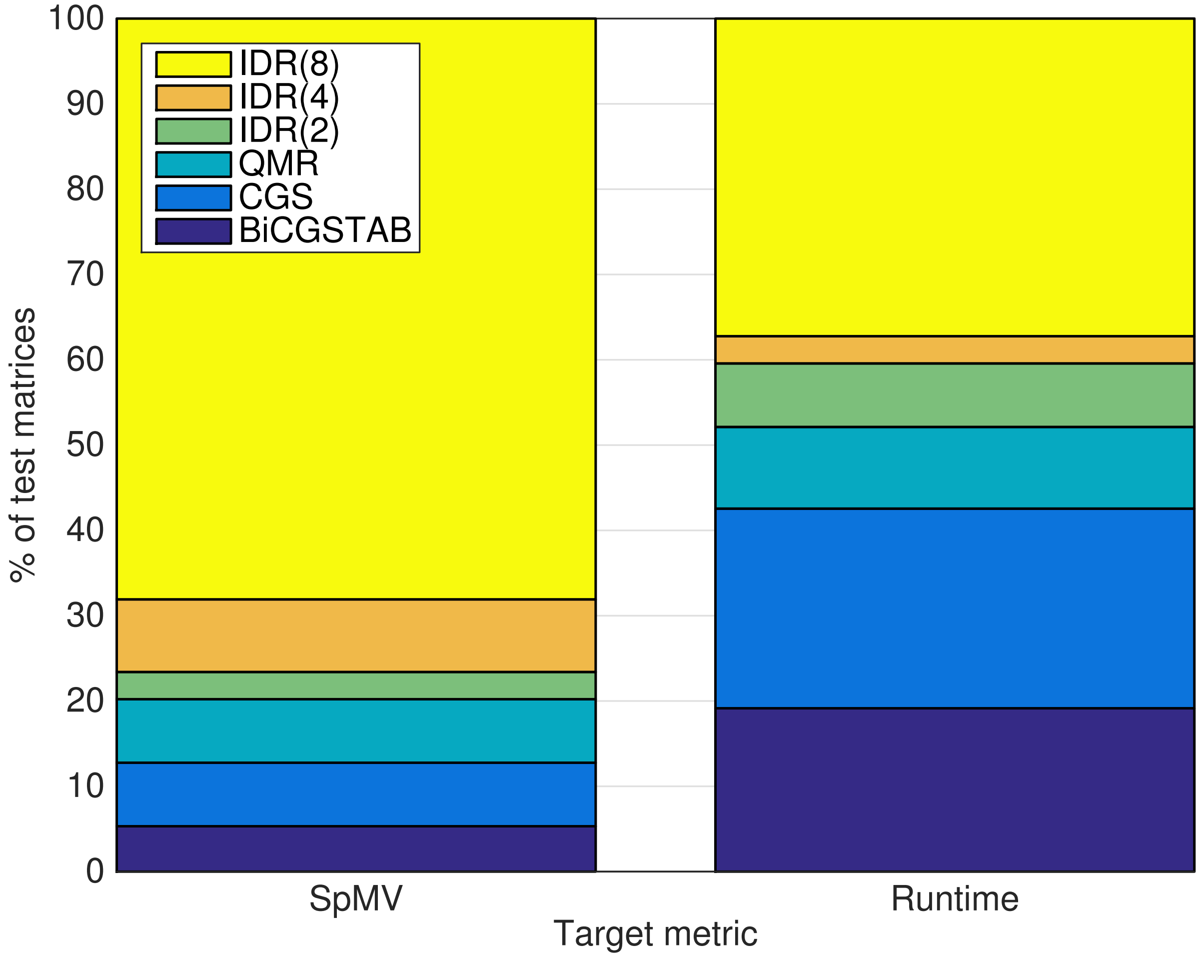
Run multiple Krylov solvers simultaneously
as poly-iterative method

- Original work: poly-iterative solver with BiCGSTAB, QMR, CGS
- IDR(s) structurally different, hard to combine in simultaneous fashion

- poly-iterative solver converges in 63 of 94 test cases (67%)
- IDR(2) converges for 60 of 94 test cases (64%)
- IDR(4) converges for 67 of 94 test cases (71%)
- IDR(8) converges for 91 of 94 test cases (96%)

Barrett et al.: **Algorithmic bombardment for the iterative solution of linear systems: A poly-iterative approach**, *Journal of Computational and Applied Mathematics 74, 1996.*

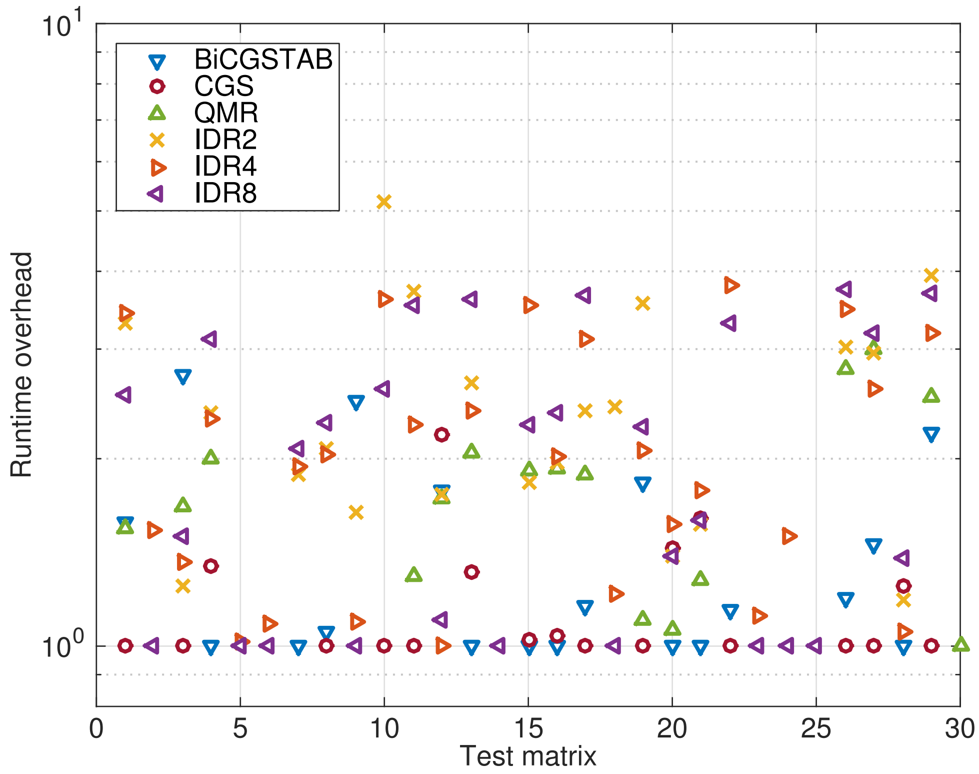
Performance– SpMV count and Runtime



- SpMV count indicative for performance when using preconditioners
- **IDR(8)** wins most cases in SpMV metric

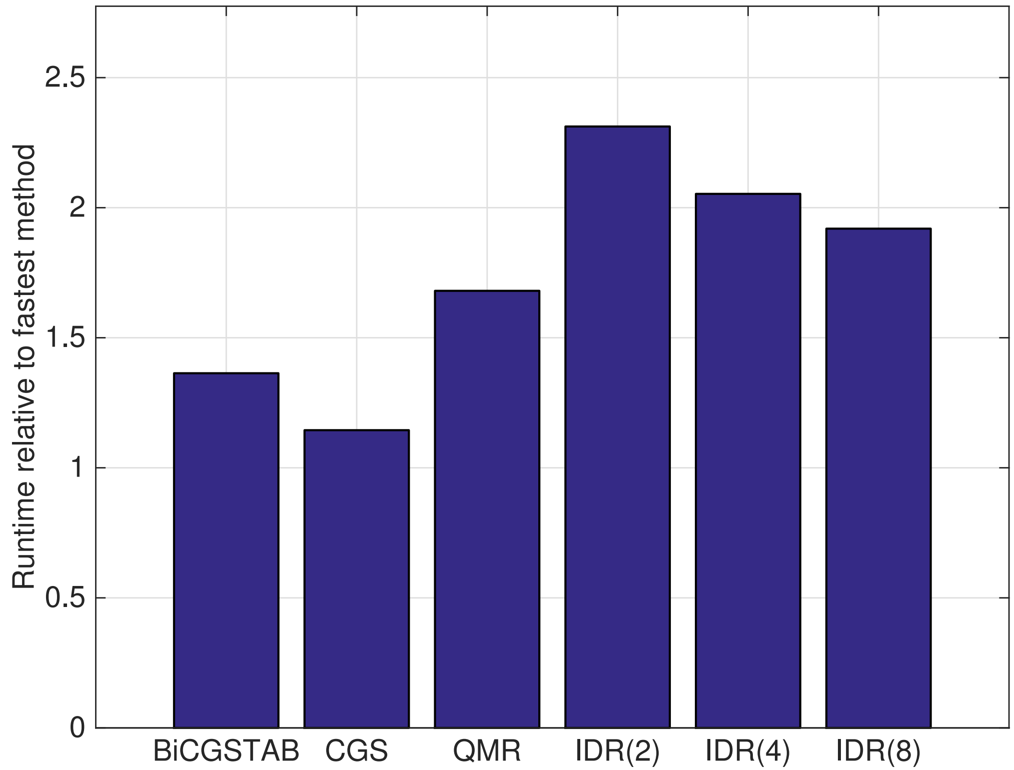
The Price of Robustness

- IDR(8) solves many systems – but often there is a faster solver
- Normalize execution times for each matrix to fastest solver



The Price of Robustness

- IDR(8) solves many systems – but often there is a faster solver
- Normalize execution times for each matrix to fastest solver
- Take average over all converging configurations



Summary

- IDR(s) is in a very robust solver.
- Robustness increases with shadow space dimension s.
- IDR(8) solves 91 of 94 test problems (96% success).
- For converging combinations, CGS, MQR, or BiCGSTAB often faster.
- On average, IDR(8) less than twice slower than the fastest method.

Future work

- Relate solver success to the problem origins.
- Enhance solvers with preconditioning.
- Target other architectures (Xeon Phi, low-power & embedded devices).

The authors would like to acknowledge support from the U.S. Department of Energy, the German Research Foundation (DFG) through the Priority Program 1648, and NVIDIA.
The authors would also like to thank Daniel B. Szyld for sharing his knowledge of Krylov methods.

U.S. DEPARTMENT OF ENERGY

DFG Deutsche Forschungsgemeinschaft

ICL THE UNIVERSITY OF TENNESSEE KNOXVILLE