Performance of the block Jacobi-Davidson method for the solution of large eigenvalue problems on modern clusters

Melven Röhrig-Zöllner, Jonas Thies, Achim Basermann
Simulation and Software Technology, DLR

Funded by the German SPPEXA programme

26. Oct. '15

Motivation

Aim

- Find some exterior eigenpairs

$$Av_i = \lambda_i v_i$$

of a large, sparse matrix A.

- Subtask block orthogonalization
- SPPEXA project ESSEX

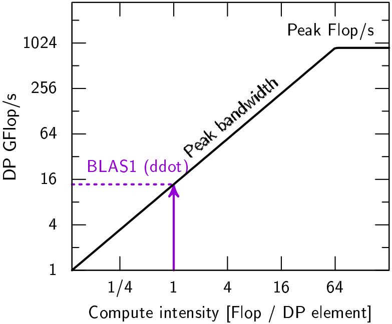
Memory gap

- Small memory bandwidth vs. high Peak Flop rate

→ Increase the compute intensity

Roofline performance model

(2x 12 core Haswell EP)



Block JDQR method

Block Jacobi-Davidson correction equation

- n_b current approximations: $A\tilde{v}_i - \tilde{\lambda}_i\tilde{v}_i = r_i$, $i = 1, \dots, n_b$
- Previously converged Schur vectors $(q_1, \dots, q_k) = Q$
- Solve approximately (with $\tilde{Q} = \begin{pmatrix} Q & \tilde{v}_1 & \dots & \tilde{v}_{n_b} \end{pmatrix}$):

$$(I - \tilde{Q}\tilde{Q}^T)(A - \tilde{\lambda}_i I)(I - \tilde{Q}\tilde{Q}^T)x_i = -r_i \qquad i = 1, \dots, n_b$$

- Use some steps of a block(ed) iterative solver
- Orthogonalize new directions $x_1, \dots, x_{n_b}$ (outer subspace iteration)

Numerical properties

- Usually needs more operations $\rightarrow$ avoided in practice
- Slightly more robust

Performance of basic operations

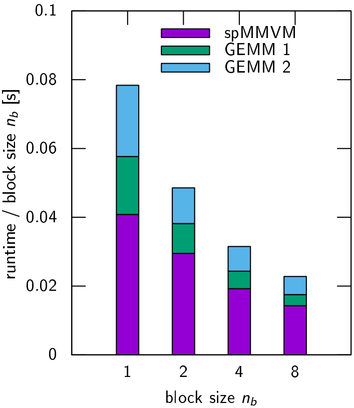
Jacobi-Davidson Operator

- Block spMVM + 2× GEMM:

$$y_i \leftarrow (I - QQ^T)(A - \tilde{\lambda}_i I)x_i$$

with $Q \in \mathbb{R}^{n\times 8}$, $i = 1, \dots, n_b$

- Matrix with $n \approx 10^7$, $n_{nz} \approx 15 \cdot 10^7$
- 10-core Intel Ivy Bridge CPU

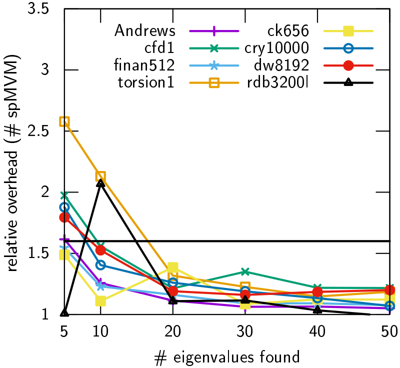


Implementation details

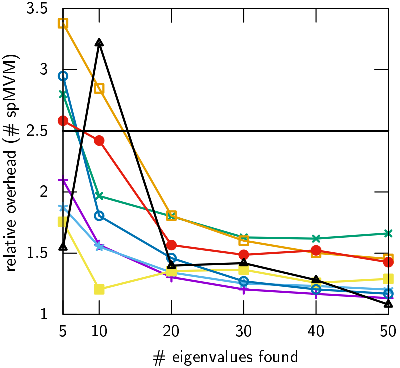
- Row-major blocks of vectors
- Hand-written BLAS operations

Numerical behavior

Block size 2



Block size 4



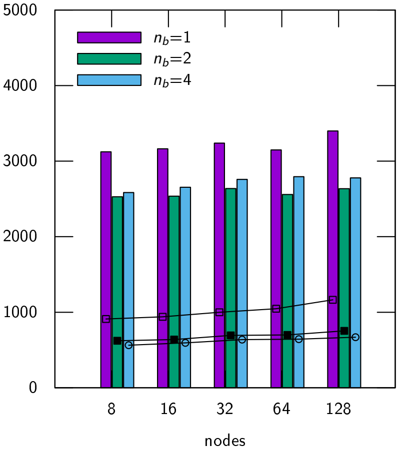
Strong scaling performance

Setup

- Non-symmetric matrix from 7-point 3D PDE discretization ($n \approx 1.3 \cdot 10^8$, $n_{nz} \approx 9.4 \cdot 10^8$)
- Seeking 20 eigenvalues
- Ivy Bridge Cluster

Results

- $n_b = 2$: significantly faster
- $n_b = 4$: no further improvement



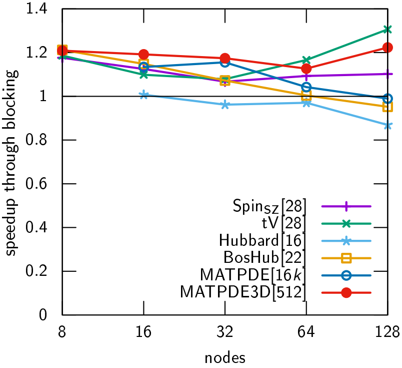
Block speedup

Setup

- Different large matrices from
  - Quantum physics
  - PDE discretization
- Seeking 20 eigenvalues
- Block size $n_b = 2$ (similar for 4)
- Ivy Bridge Cluster

Results

- Faster by a factor 1.2
- Higher communication volume



Block orthogonalization schemes

Problem definition

- Given orthogonal vectors $(w_1, \dots, w_k) = W$
- For $X \in \mathbb{R}^{n \times n_b}$ find orthogonal $Y \in \mathbb{R}^{n \times \tilde{n}_b}$ with

$$YR_1 = X - WR_2, \qquad \text{and} \qquad W^T Y = 0$$

Two phase algorithms

Phase 1 Project: $\bar{X} \leftarrow (I - WW^T)X$

Phase 2 Orthogonalize: $Y \leftarrow f(\bar{X})$

- suitable f:
  - SVQB (Stathopoulos and Wu, SISC 2002)
  - TSQR (Demmel et al., SISC 2012)
- Each phase messes with the accuracy of the other. $\rightarrow$ iterate

Basic operations

Kernel fusion

Fuse operations to increase the compute intensity:

Phase 2 $\bar{X} \leftarrow X\bar{M}, \quad N \leftarrow W^T\bar{X}$

Phase 1 $\bar{X} \leftarrow X - WN, \quad M \leftarrow \bar{X}^T\bar{X}$

Phase 3 $\bar{X} \leftarrow X\bar{M}, \quad M \leftarrow \bar{X}^T\bar{X}$

$\Rightarrow$ use SVQB

Increased precision

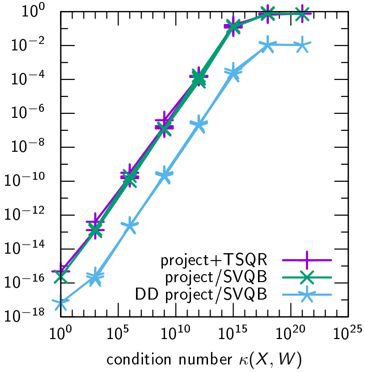
Idea Calculate value and error of each arithmetic operation

- Store intermediate results as double-double (DD) numbers
- Based on arithmetic building blocks (2Sum, 2Mult)
  Muller et al.: Handbook of Floating-Point Arithmetic, Springer 2010
- Exploit FMA operations (AVX2)

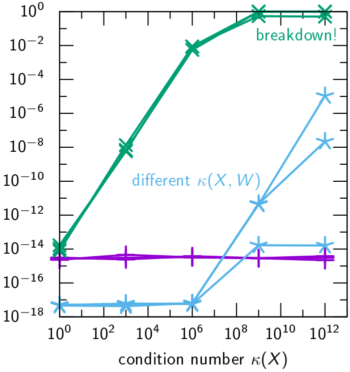
Results: accuracy after one iteration

$n = 1000$, $n_{proj} = 20$, $n_b = 4$

Error in $W^T Y$

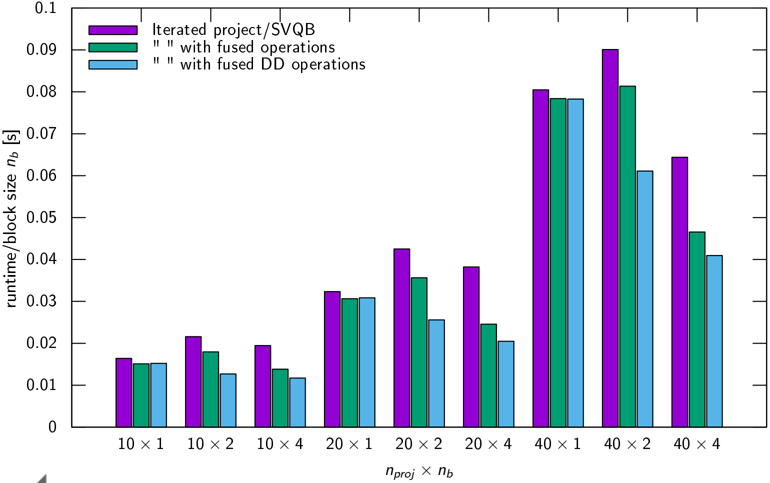


Error in $Y^T Y$



Results: runtime to convergence

$n = 8 \cdot 10^6$, $\kappa(X) = 10^{-6}$, $\kappa(X, W) = 10^{-12}$, $\epsilon = 10^{-10}$, Intel Haswell Workstation



Conclusion

Block JDQR algorithm

- Slightly more operations vs. better performance
- Only small block sizes useful ($n_b = 2$ or $n_b = 4$)
- Requires a careful implementation
  (hand-written BLAS, row-major memory layout)

→ Faster for more than 10 eigenvalues (factor 1.2)
See Röhrig-Zöllner et al.: Increasing the performance of Jacobi-Davidson by blocking. Accepted for publication in SISC.

Block orthogonalization

- Improved performance through kernel fusion
- Better accuracy through double-double operations (for free)

→ Faster when less iterations are required

Thank you for your attention!

Faster through

- block algorithms
- kernel fusion
- increased precision