Fast and Scalable Eigensolvers for Multicore and Hybrid Architectures

Paolo Bientinesi

AICES, RWTH Aachen
pauldj@aices.rwth-aachen.de

40th SPEEDUP Workshop on High-Performance Computing
February 6–7, 2012
University of Basel, Switzerland

Symmetric Dense Eigenproblem

$AX = X\Lambda$	STDEIG
$AX = XB\Lambda$	GENEIG

- Input:

$A \in \mathcal{C}^{n\times n}$,	$A^H = A$
$B \in \mathcal{C}^{n\times n}$,	SPD
$k,\ 1 \leq k \leq n$	#eigenpairs

- Output:

$X \in \mathcal{C}^{n\times k}$,	eigenvectors
$\Lambda \in \mathcal{R}^{k\times k}$,	eigenvalues

- Accuracy:

$\|AX - X\Lambda\|$,	residual
$\|X^H X - I\|$,	orthogonality

6-stage approach

GENEIG $AX = XB\Lambda$

1	$LL^H = B$	Cholesky factorization	$O(n^3)$
2	$M \leftarrow L^{-1}AL^{-H}$	Reduction to standard form	$O(n^3)$
3	$T = Q^H M Q$	Reduction to tridiagonal form	$O(n^3)$
4	$TZ = Z\Lambda$	Tridiagonal eigenproblem	$O(kn)$–$O(n^3)$
5	$Y = QZ$	Backtransformation #1	$O(kn^2)$
6	$X = L^{-H}Y$	Backtransformation #2	$O(kn^2)$

Nested Eigensolvers

GENEIG $\rightarrow$ STDEIG $\rightarrow$ TRDEIG

1	$LL^H = B$	Cholesky factorization	$O(n^3)$
2	$M \leftarrow L^{-1}AL^{-H}$	Reduction to standard form	$O(n^3)$
3	$T = Q^H M Q$	Reduction to tridiagonal form	$O(n^3)$
4	$TZ = Z\Lambda$	Tridiagonal eigenproblem	$O(kn)$–$O(n^3)$
5	$Y = QZ$	Backtransformation #1	$O(kn^2)$
6	$X = L^{-H}Y$	Backtransformation #2	$O(kn^2)$

Algorithms

Stage 4: TRDEIG

1958	Bisection + Inverse Iteration (BI)	subsets	$O(kn^2)$
1961	QR	high-accuracy	$O(n^3)$
1981	Divide & Conquer (DC)	BLAS3, accurate	$O(n^3)$
1997	MRRR	subsets, no re-orth.	$\mathbf{O(kn)}$

Algorithms

Stage 4: TRDEIG

1958	Bisection + Inverse Iteration (BI)	subsets	$O(kn^2)$
1961	QR	high-accuracy	$O(n^3)$
1981	Divide & Conquer (DC)	BLAS3, accurate	$O(n^3)$
1997	MRRR	subsets, no re-orth.	$\mathbf{O(kn)}$

Stage 3: Reduction to TRDEIG

- 1-stage Householder
- Successive Banded Reduction

Numerical Libs – Development Cycle (?)

(0) New architecture

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance),   FFT

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

(2) factorizations, AX=B

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

(2) factorizations, AX=B

(2) factorizations, AX=B

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM (→ peak performance), FFT

(2) BLAS3, factorizations, AX=B ⟶ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

(2) factorizations, AX=B

(2) factorizations, AX=B

(3) factorizations, AX=B, matrix operations

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

(2) factorizations, AX=B

(2) factorizations, AX=B

(3) factorizations, AX=B, matrix operations

$\vdots$

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

(2) factorizations, AX=B

(2) factorizations, AX=B

(3) factorizations, AX=B, matrix operations

⋮

⋮

(4) Eigenproblems

Numerical Libs – Development Cycle (?)

(0) New architecture

(1) GEMM ($\rightarrow$ peak performance), FFT

(2) BLAS3, factorizations, AX=B $\longrightarrow$ LINPACK benchmark

(2) BLAS3, factorizations, AX=B

(2) factorizations, AX=B

(2) factorizations, AX=B

(3) factorizations, AX=B, matrix operations

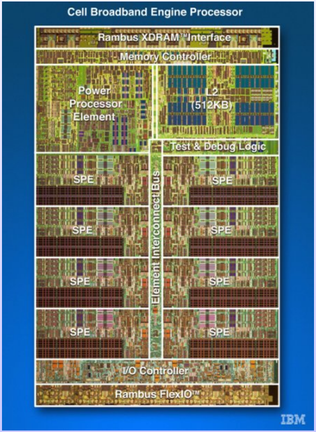
⋮

⋮

(4) Eigenproblems

HPC

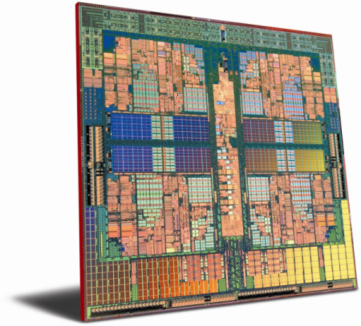
Linear solvers $\neq$ Eigensolvers



- GEMM: 99%
- FFT
- Linear systems: HPL
  2008: Roadrunner >1 PetaFLOP
- 2009: discontinued

2005: GPGPUs

- CUBLAS (*)
- HPL, Top500
- CULA
- FLAME, MAGMA



- GEMM
- mt BLAS
- HPL, Top500
- FLAME, PLASMA

Our contributions

MR^3-SMP multithreaded

- Matthias Petschow — RWTH Aachen
  `http://code.google.com/p/mr3smp`

PMRRR, EleMRRR hybrid MPI + MT

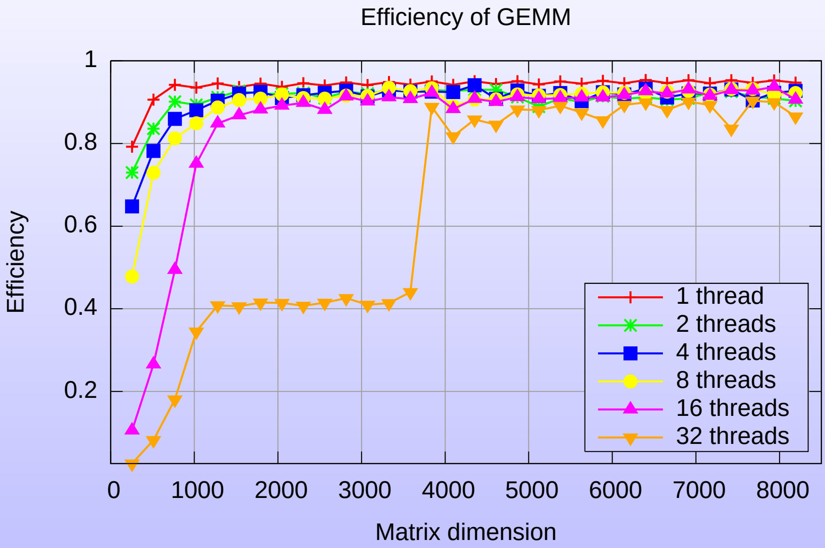
- Matthias Petschow — RWTH Aachen
  `http://code.google.com/p/pmrrr`
- Jack Poulson — UT Austin
  `http://code.google.com/p/elemental`

... GPUs

- Christian Lessig — University of Toronto
- Enrique Quintana-Ortí — Universidad Jaume I
- Francisco Igual — Universidad Jaume I

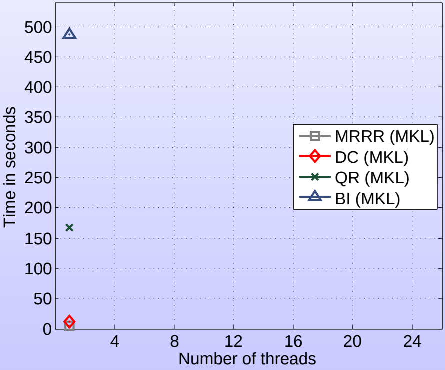
Multi-threaded BLAS

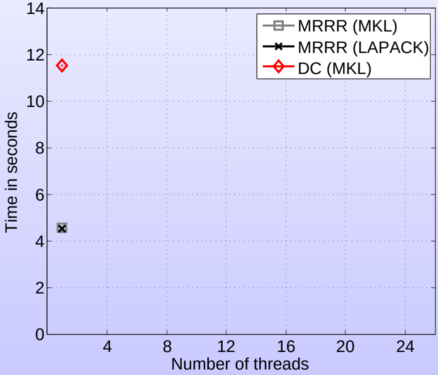
Xeon, 32 physical cores



Multi-threaded BLAS for TRDEIG?

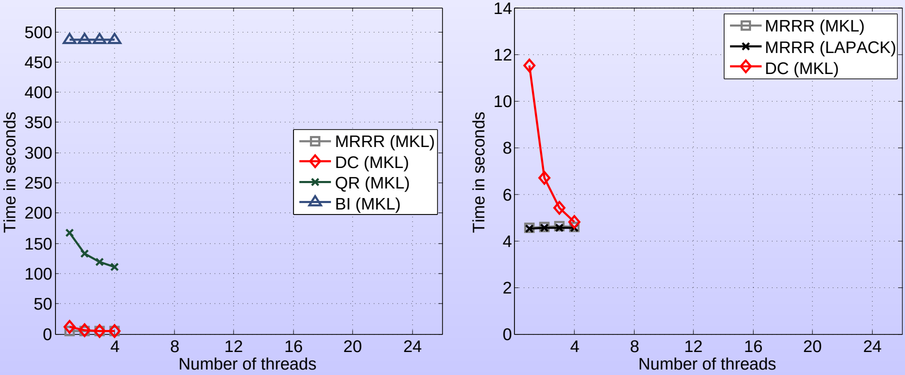
Tridiagonal eigensolvers. Matrix size=4289, from DFT.





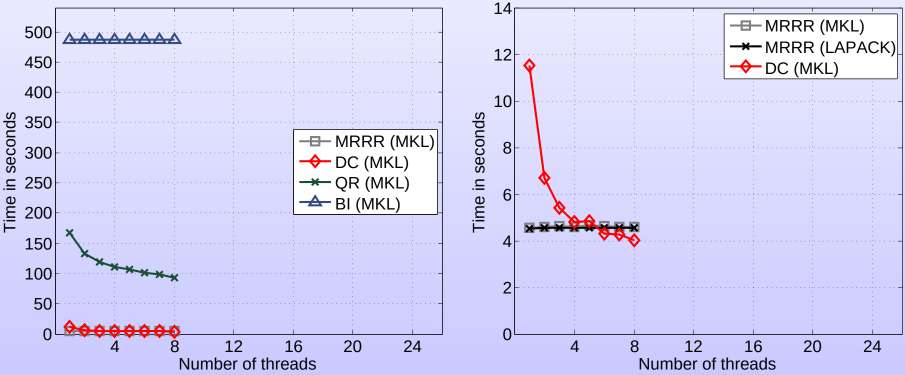
Multi-threaded BLAS for TRDEIG?

Tridiagonal eigensolvers. Matrix size=4289, from DFT.



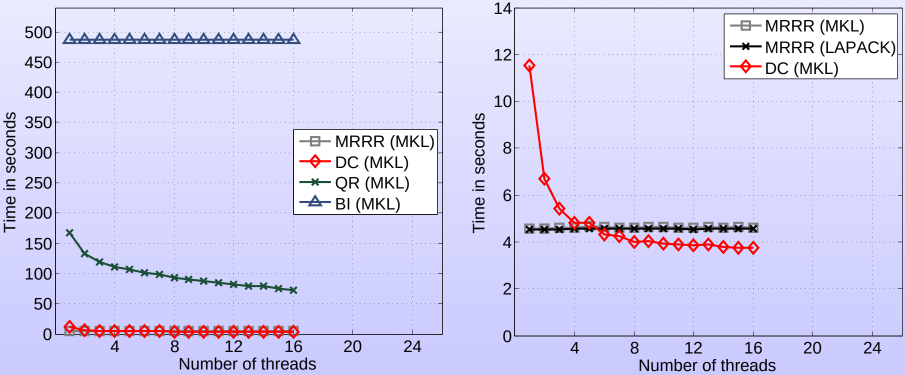
Multi-threaded BLAS for TRDEIG?

Tridiagonal eigensolvers. Matrix size=4289, from DFT.



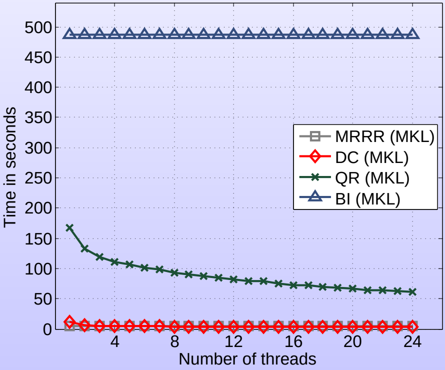
Multi-threaded BLAS for TRDEIG?

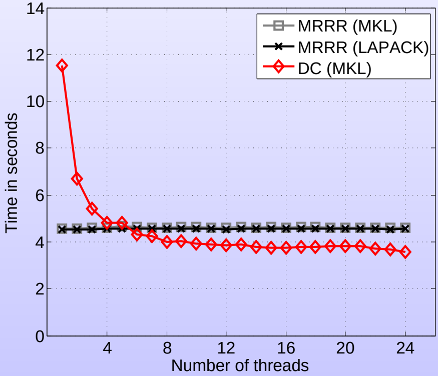
Tridiagonal eigensolvers. Matrix size=4289, from DFT.



Multi-threaded BLAS for TRDEIG?

Tridiagonal eigensolvers. Matrix size=4289, from DFT.



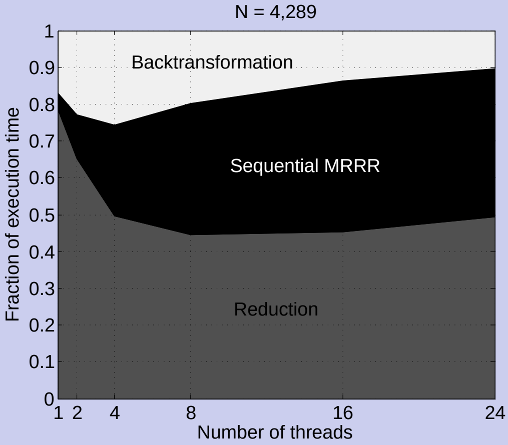


More motivation?

"MR3 is $O(n^2)$ anyway. . . "

More motivation?

"MR3 is $O(n^2)$ anyway…"



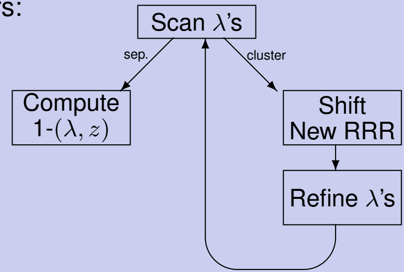
MRRR – Dhillon & Parlett (1998)

Multiple Relatively Robust Representations

- first stable algorithm to compute k eigenpairs in $O(nk)$ ops
- no reorthogonalization

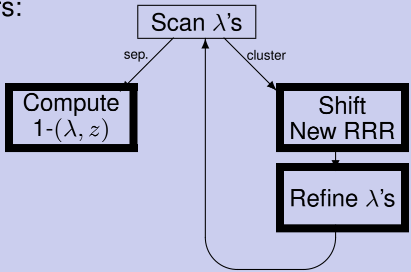
Multiple Relatively Robust Representations

- first stable algorithm to compute k eigenpairs in $O(nk)$ ops
- no reorthogonalization
- 1) eigenvalues $\rightarrow$ 2) eigenvectors + eigenvalues
- eigenvalues: *dqds* or *Bisection*
- eigenvectors:

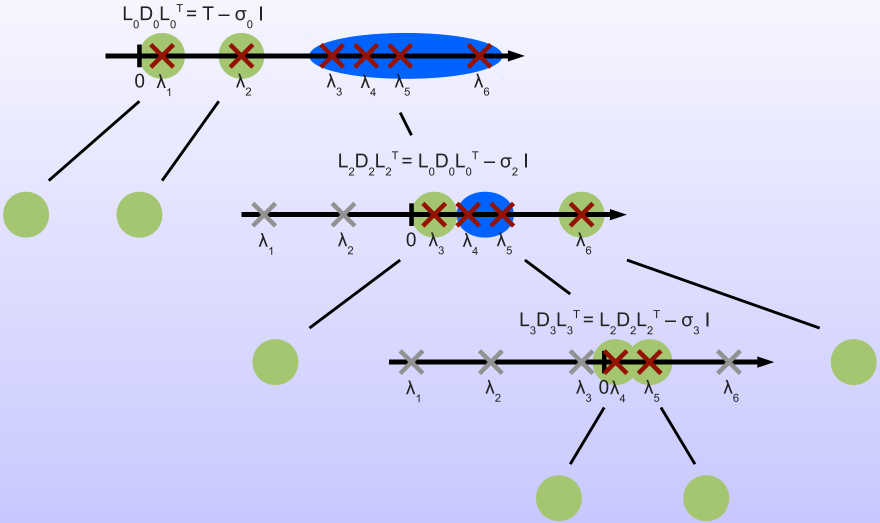


Multiple Relatively Robust Representations

- first stable algorithm to compute k eigenpairs in $O(nk)$ ops
- no reorthogonalization
- 1) eigenvalues $\rightarrow$ 2) eigenvectors + eigenvalues
- eigenvalues: *dqds* or *Bisection*
- eigenvectors:



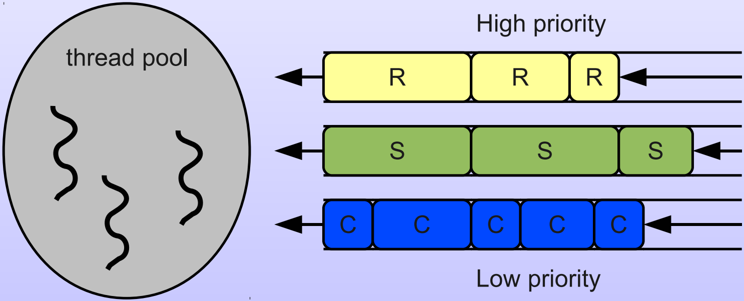
Representation Tree



MR^3-SMP: the work queue

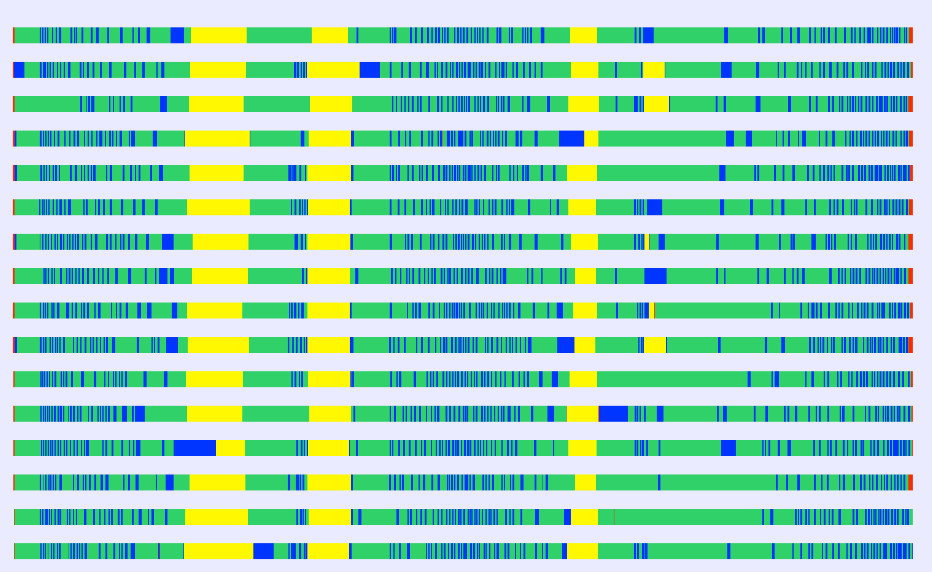
Tasks:

- a) Singleton ⇒ **S**: Eigenvector computation
- b) Cluster ⇒ **C**: Shift + new representation (RRR)
- c) New RRR ⇒ **R**: Eigenvalues refinement



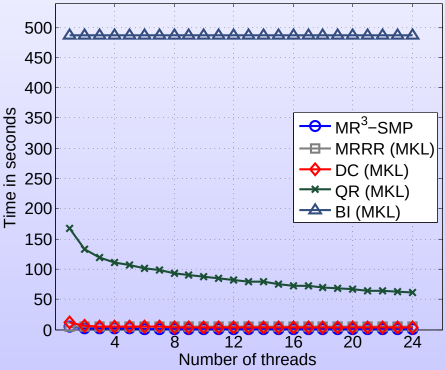
Example trace: 16 cores—eigenvectors

Matrix size: 12387   Execution time: 3.3s   Sequential: 49.3s (LAPACK)



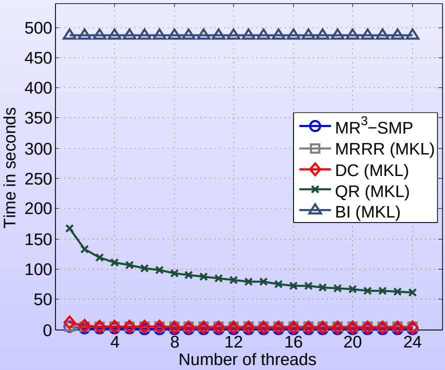
MR^3-SMP: Timings

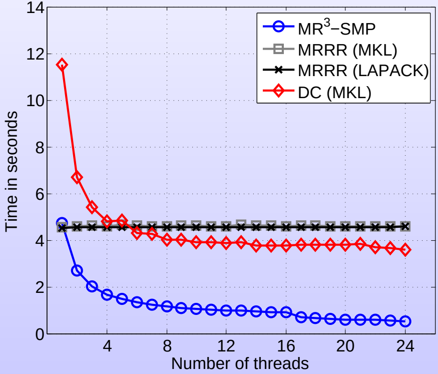
Matrix size=4289, from DFT.



MR^3-SMP: Timings

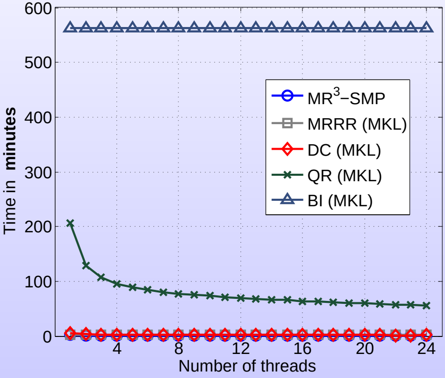
Matrix size=4289, from DFT.





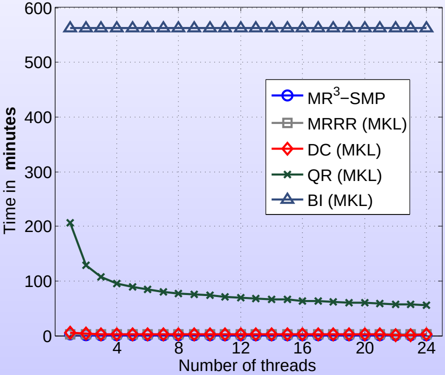
A larger example

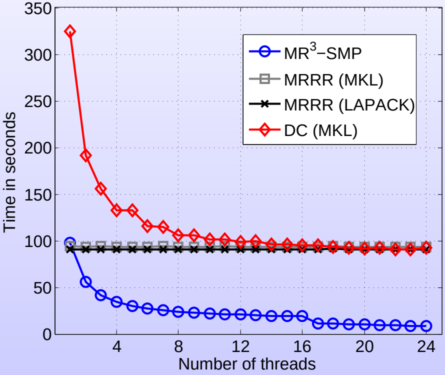
Matrix size=16023; frequency response analysis of automobiles.



A larger example

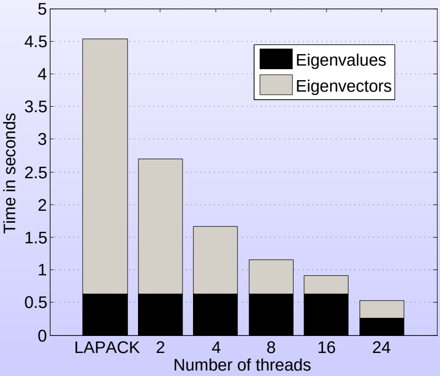
Matrix size=16023; frequency response analysis of automobiles.



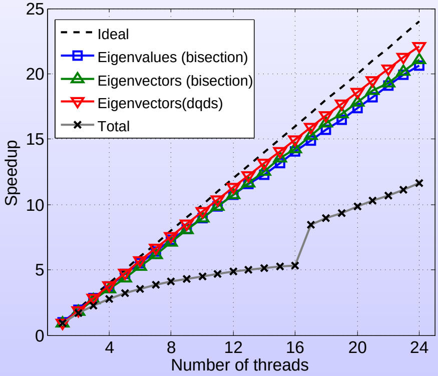


From 9+ hours to 8.3 seconds.

Speedups

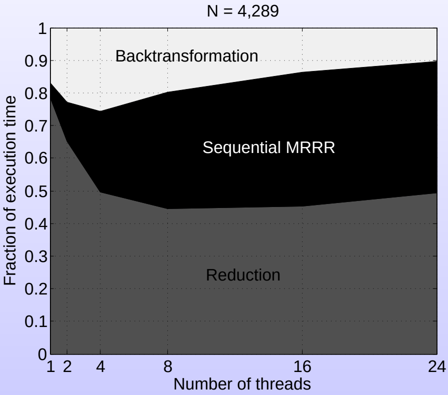


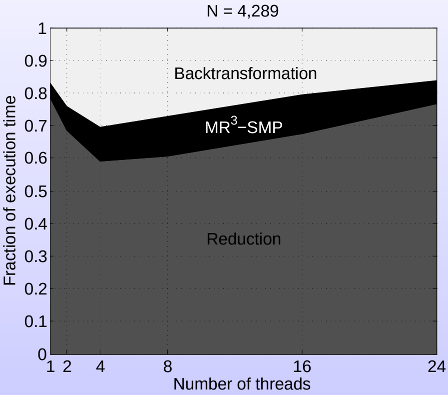
Time in seconds
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
LAPACK
2
4
8
16
24
Number of threads
Eigenvalues
Eigenvectors



Speedup
25
20
15
10
5
0
4
8
12
16
20
24
Number of threads
Ideal
Eigenvalues (bisection)
Eigenvectors (bisection)
Eigenvectors(dqds)
Total

3 stages: before and after





Distributed memory

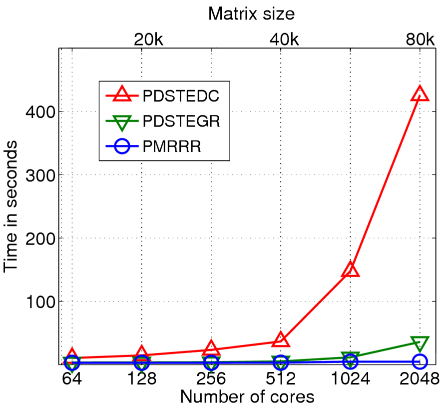
PMRRR, EleMRRR

- Static assignment of eigenpairs to nodes
- Multithreading
- Node-node communication: only eigenvalues
- PMRRR + Elemental $\Rightarrow$ EleMRRR

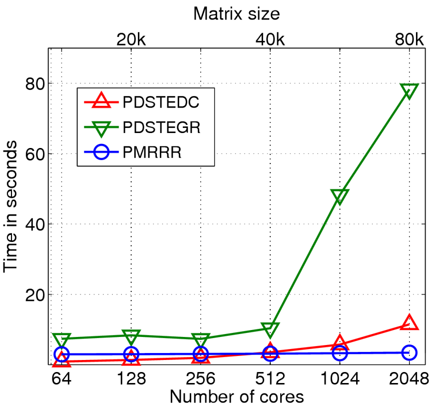
  Generalized, standard and tridiagonal
  hybrid eigensolvers

TRDEIG: PMRRR

1-2-1 matrix

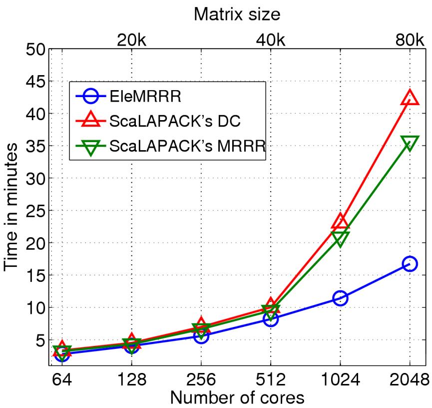


Wilkinson matrix

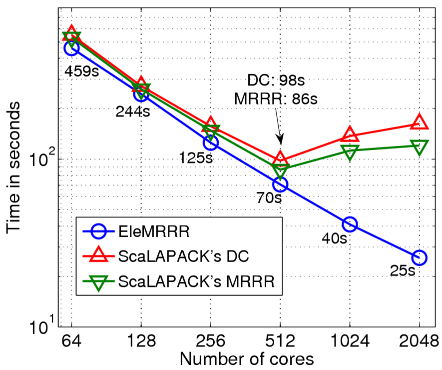


GENEIG: Weak & strong scaling

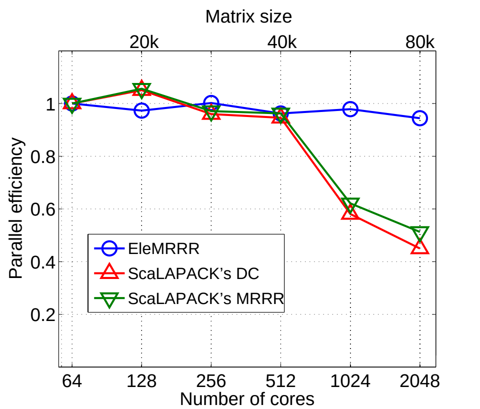
Weak scalability

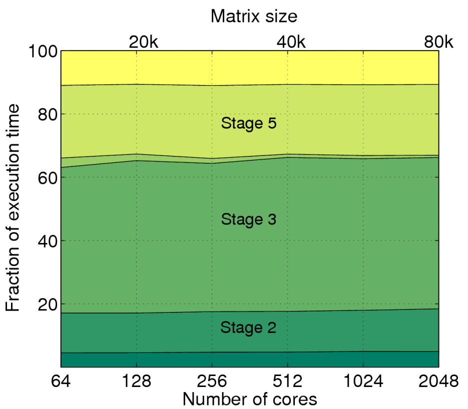


Strong scalability, n=20000



GENEIG: Efficiency





mrrr_dp = data-parallel MRRR

	rand(0,1)		rand(-1,1)	
n	LAPACK	mrrr_dp	LAPACK	mrrr_dp
128	6.98	6.26	6.79	3.84
256	32.1	13.0	31.86	8.34
512	154.9	28.7	152.7	19.2
1024	656.1	60.2	647.6	54.0

STDEIG

Nehalem, 8 cores

Reduction to tridiagonal form

n	LAPACK	SBR	SBR + GPU
2048	0.23	0.6	0.58
6144	8.4	8.58	6.26
10240	40.5	30.4	20.32
24576	582.4	308.4	166.8

Reduction + backtransformation

n	LAPACK	SBR	SBR + GPU
2048	0.50	1.77	1.12
6144	13.5	29.0	12.7
10240	61.6	116.8	43.8
24576	845.1	1416.7	403.3

Conclusions

Multi-threaded BLAS for eigensolvers: not THAT good

MR^3-SMP, PMRRR, EleMRRR

- eigensolvers tailored for multi-core, distributed, hybrid architectures
- faster than LAPACK, MKL, ScaLAPACK
- almost perfect speedups
- software is available

Deutsche Forschungsgemeinschaft
DFG

Financial support from the Deutsche Forschungsgemeinschaft (German Research Association) through grant GSC 111 is gratefully acknowledged.