Optimizing PLASMA Eigensolver on Large SGI UV Shared Memory Systems

Cheng Liao
SR Software Engineer
November, 2016

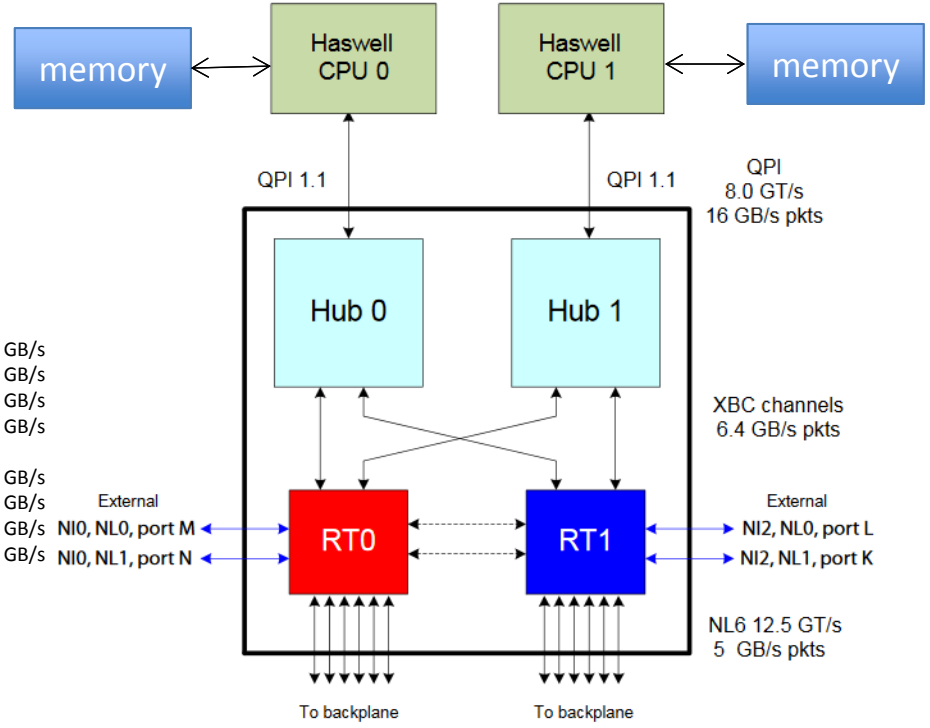
List of Eigensolvers

SCALAPACK	ELPA	EIGENEXA	PLASMA
PDSYEVD, PDSYEVX, PDSYEV, PDSYEVR	ELPA1, EPLA2	EIGEN_S, EIGEN_SX	DSYEVR (SMP only, no MPI, dataflow scheduler)
1-stage tridiagonalization, 50% memory bound kernel	1-stage & 2-stage, the latter tridiagonalizes much faster	improved 1-stage penta-diagonalization with sym2v	2-stage tridiagonalization with tile storage layout
tridiagonal system solvers: D&C, BI, QR, MRRR	D&C tridiagonal solver	D&C pentadiagonal solver that takes longer to execute	LAPACK MRRR tridiagonal solver dstemr
compute bound 1-stage back-transformation	compute bound 1/2-stage back-transformation	compute bound 1-stage back-transformation	compute bound 2-stage back-transformation
de facto Industrial standards but slow	likes high FLOPS systems	likes high memory bandwidth systems	likes high FLOPS systems

SGI UV3000 – Commodity CPUs + Proprietary Node Controllers with Routing Options for Cache Coherence

copy with GRU, THP on

latency bet node 0 and node 0: 87.6 ns
latency bet node 0 and node 1: 399.4 ns
latency bet node 0 and node 2: 508.8ns
latency bet node 0 and node 3: 508.2 ns
...
grucopy on node 0 dest 0 src 0 BW: 3.764 GB/s
grucopy on node 0 dest 0 src 1 BW: 3.658 GB/s
grucopy on node 0 dest 0 src 2 BW: 3.487 GB/s
grucopy on node 0 dest 0 src 3 BW: 3.503 GB/s
...
grucopy on node 0 dest 0 src 0 BW: 3.779 GB/s
grucopy on node 0 dest 1 src 0 BW: 3.778 GB/s
grucopy on node 0 dest 2 src 0 BW: 3.547 GB/s
grucopy on node 0 dest 3 src 0 BW: 3.532 GB/s
...



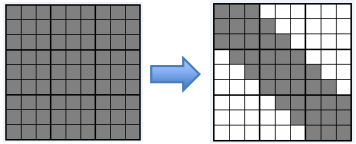
copy with shared memory, THP on

latency bet node 0 and node 0: 87.5ns
latency bet node 0 and node 1: 400.5 ns
latency bet node 0 and node 2: 507.8 ns
latency bet node 0 and node 3: 507.7ns
...
bcopy on node 0 dest 0 src 0 BW: 6.407 GB/s
bcopy on node 0 dest 0 src 1 BW: 2.465 GB/s
bcopy on node 0 dest 0 src 2 BW: 1.977 GB/s
bcopy on node 0 dest 0 src 3 BW: 1.981 GB/s
...
bcopy on node 0 dest 0 src 0 BW: 6.316 GB/s
bcopy on node 0 dest 1 src 0 BW: 5.906 GB/s
bcopy on node 0 dest 2 src 0 BW: 5.275 GB/s
bcopy on node 0 dest 3 src 0 BW: 5.380 GB/s
...

Five Computational Phases of PLASMA Eigensolver

Phase 1

Dense to Band Reduction



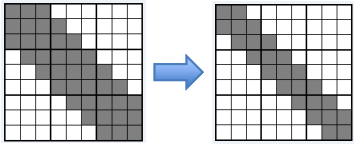
output: **B, V_1, T_1**

FP op count: ~5/3●N^3

Tile

Phase 2

Bulge Chasing



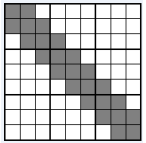
output: **T, V_2, T_2**

FP op count: ~6●NB●N^2

LAPACK

Phase 3

Tridiagonal Eigensolution



output: **ω, E**

dst~~e~~mr mr^3-smp

Phase 4

Eigenvector back-transformation

$$E = (I - V_2 \bullet T_2 \bullet V_2^t) \bullet E$$

FP op count: ~5/2●N^3

LAPACK

Phase 5

Eigenvector back-transformation

$$E = (I - V_1 \bullet T_1 \bullet V_1^t) \bullet E$$

FP op count: ~5/2●N^3

Tile

NUMA Considerations

Data Arrays

- input matrix A
- banded matrix B
- block Householder vectors V & scalars T (2 sets)
- eigenvectors E
- scratch buffers (MKL + PLASMA, etc.)

Interleaved Memory

- Needs to be evenly distributed among participating nodes to avoid network hotspots
- A, V & T are in this category

Local Memory

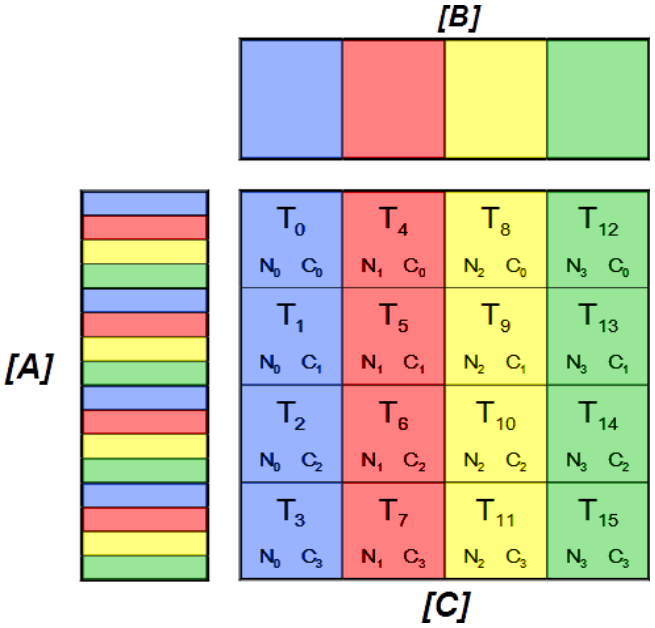
- B, E and scratch buffers are in this category

Thread Pinning

- Threads need to be pinned to establish data locality
- pthread+OpenMP model to improve communication to computation ratio
- Watch out for “OpenMP ambush”

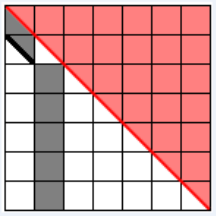
Arrange data and pin threads to minimize communication and avoid network hotspots!

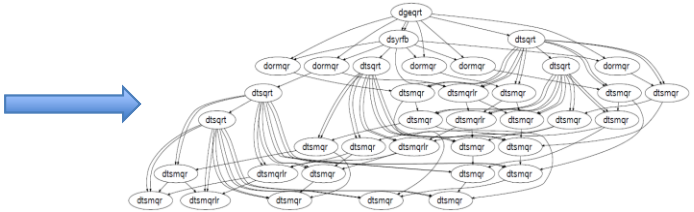
Sample ‘Best’ NUMA Placement – Matrix Multiply on 4 Quad Core Nodes



memory placement color coded!

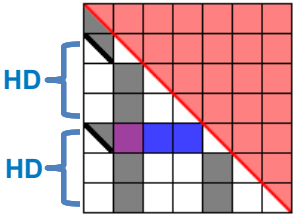
Algorithmic Issue(1) – DBR Degree of Parallelism

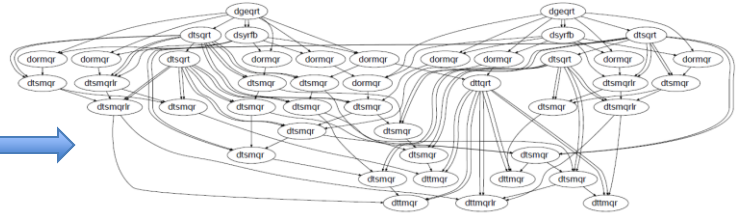




Sequential QR based DBR

DAG for sequential QR based DBR

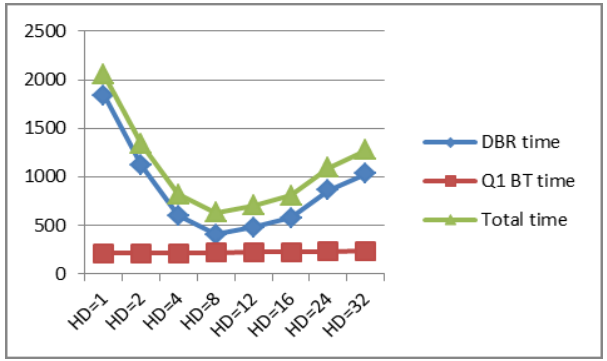




Tree parallel QR based DBR

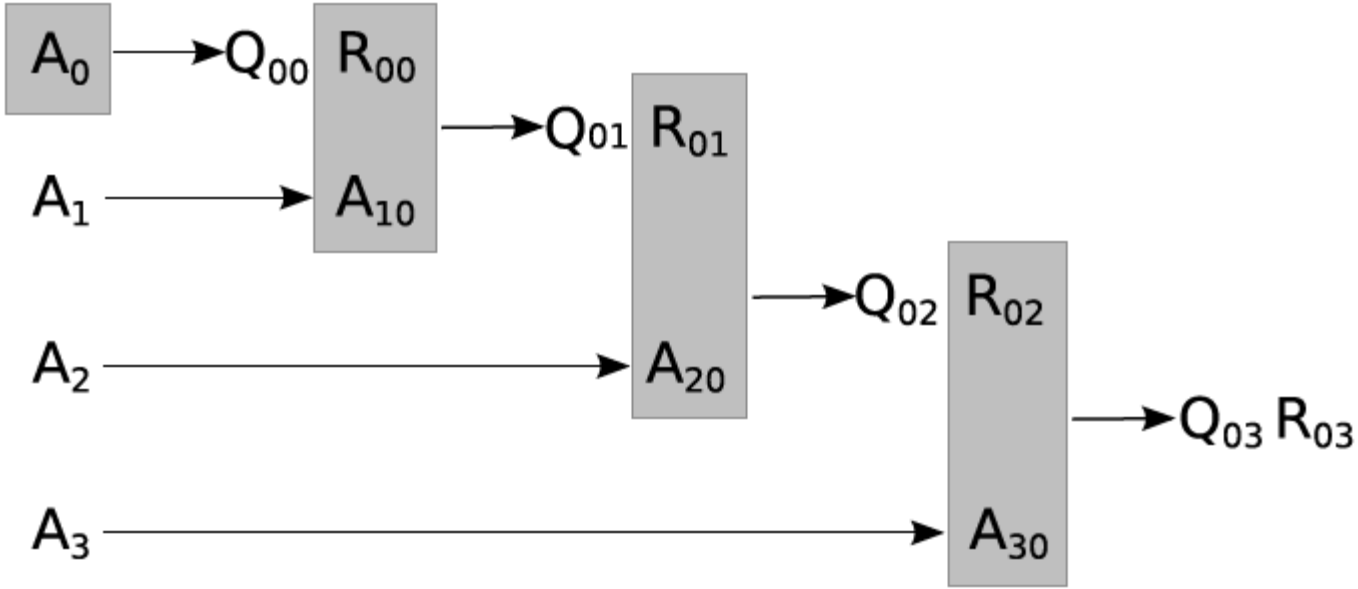
DAG for tree parallel QR based DBR

Employing tree parallel QR with multiple Householder domains can improve dense to band reduction performance very significantly:



Number of HDs vs run time (secs)

Incremental Sequential QR Factorization As Illustrated in LAWN204



A Note on FP OP Counts

```
for (k = 0; k < NT-1; k++){ // NT(==N/Nb): #tiles/row, BS: subdomain size
  BS = (NT – k - 2)/HD + 1; //HD: Number of Householder domains
//local QR factorizations on leading tiles of subdomains
  for (m = k+1; m < NT; m += BS) DGEQRT;
  // Apply the local reflectors
  // LEFT and RIGHT on the diagonal blocks
  for (m = k+1; m < NT; m += BS) DSYRFB;
  // RIGHT on tile column until the bottom
  for (m = k+1; m < NT; m += BS) {  for (n = m+1; n < NT ; n++) DORMQR; }
  // LEFT on tile row until the diagonal
  for (m = k+1; m < NT; m += BS) { for (n = k+1; n < m ; n++) DORMQR; }
  // include other tiles in the subdomains
  for (M = k+1; M < NT; M += BS) {
    for (m =M+1; m < min(M+BS,NT); m++) {
      DTSQRT;
      for (i = k+1; i < m; i++) DTSMQR; // LEFT, excluding i=M
      for (j = m+1; j < NT; j++) DTSMQR; // RIGHT
      DTSMQRLR; // LEFT or RIGHT
    }
  }
  // tree-based merge of the local factors
  for (RD = BS; RD < NT-k-1; RD *= 2) {
    for (M = k+1; M+RD < NT; M += 2*RD) {
      DTTQRT;
      for (i=k+1; I<M+RD-1; i++) DTTMQR; // LEFT, excluding i=M
      for (j=M+RD+1; j <NT; j++) DTTMQR; // RIGHT
      DTTMQRLR; // LEFT or RIGHT
    }
  }
}
```

Compute Kernel	Number of calls	
dgeqrt	1	2
dsyrfb	1	2
dormqr	5	10
dtsqrt	5	4
dtsmqr	20	16
dtsmqrlr	5	4
dttqrt	n/a	1
dttmqr	n.a	4
dttmqrlr	n/a	1

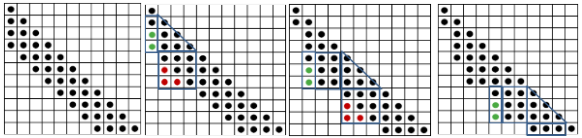
Modest numbers of Householder domains do not change the complexity of D2B reduction:

$$\sum_{k=0}^{NT-2}\sum_{m=k+2}^{NT-1}\sum_{j=k+1}^{NT-1} 5Nb^3 \sim= \left(\frac{5}{3}\right) N^3$$

The 1st stage back-transformation complexity also does not change.

dormqr OPs need to be considered if the number of Householder domains is significant relative to number of tiles/row.

Algorithmic Issue(2)- Communication and Thread Placement in Bulge Chasing



PLASMA implementation:

- All work is done by 3 types of compute kernels/tasks.
- $T_{?y(z-1)}$ and $T_{?(y-1)(z+2)}$ done before $T_{?yz}$.

Shortcomings:

- Left and right applications of Householder reflectors need to load matrix blocks 4 times.
- No thread placement strategy.

Improved communication:

$$(\mathbf{I} - \acute{\alpha} \bullet \boldsymbol{u} \bullet \boldsymbol{u}^t) \bullet \mathbf{B} \bullet (\mathbf{I} - \tau \bullet \mathbf{v} \bullet \mathbf{v}^t)$$

(4x loads of **B**)

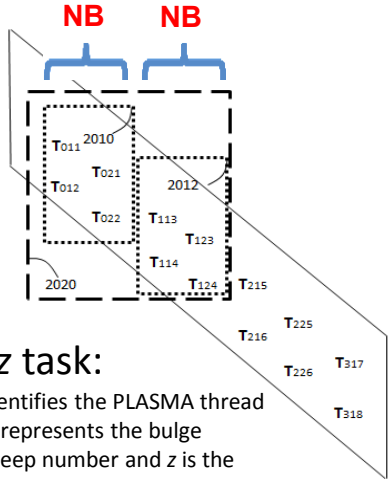
$$\mathbf{B} - \acute{\alpha} \bullet \boldsymbol{u} \bullet \boldsymbol{u}^t \bullet \mathbf{B} + \text{WORK} \bullet \mathbf{v}^t$$

Where $\boldsymbol{w} = \mathbf{B} \bullet \mathbf{v}$, $b = \boldsymbol{u}^t \bullet \boldsymbol{w}$, $s = \acute{\alpha} \bullet \tau \bullet b$,
and $\text{WORK} = -\tau \bullet \boldsymbol{w} + s \bullet \boldsymbol{u}$. (column-wise, 2x loads of **B**)

Improved performance:

Problem/ Tile Size	Bulge Chasing performance on 60 e5-4627 v3 sockets					
	Original code		Improved+linear threads		Improved+round-robin threads	
	sec	gflops	sec	gflops	second	gflops
N=115200, NB=480	106	360	74	518	78	490
N=115200,NB=640	173	295	139	366	109	468
N=115200,NB=960	684	112	458	167	240	318
N=288000,NB=960	7676	62	8917	54	1387	345

Round-Robin Thread Placement:



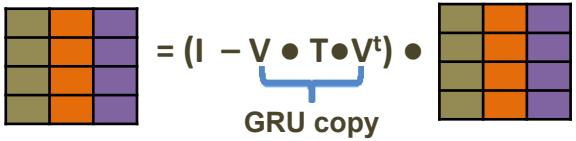
A T*xyz* task:

where *x* identifies the PLASMA thread number, *y* represents the bulge chasing sweep number and *z* is the taskid of the sweep.

Socket #0	Socket #1	Socket #2	Socket #3
Thread 0	Thread 1	Thread 2	Thread 3
Thread 4	Thread 5	Thread 6	Thread 7
…	…	…	…

Algorithmic Issue(3)- 1D Parallel Decomposition in Eigenvector Back-transformations

It is straightforward to implement 1D parallel decomposition for eigenvector back-transformation. However it also is well known that 1D parallel decomposition has a high communication to computation ratio. The problem size needs be sufficient large to ensure good performance.



Total FP Ops for Q2 BT:

$$\sum_{j=vblksiz}^{N,vblksiz} \frac{j}{NB}\left(4N(NB+vblksiz)vblksiz + Nvblksiz^2\right)$$

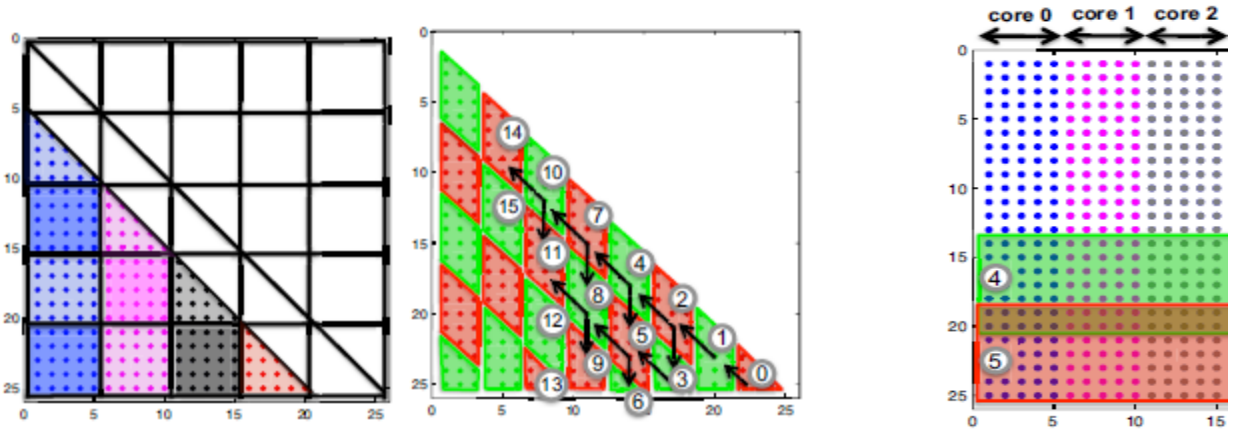
$$\sim = 2N^3\left(1+1.25\frac{vblksiz}{NB}\right)$$

Per Socket Communication for Q2 BT:

$$\underbrace{\frac{N^2}{2} * sizeof(double)}_{V} + \underbrace{\frac{N^2 vblksiz}{2*NB} * sizeof(double)}_{T} = \sim 5N^2\ bytes$$

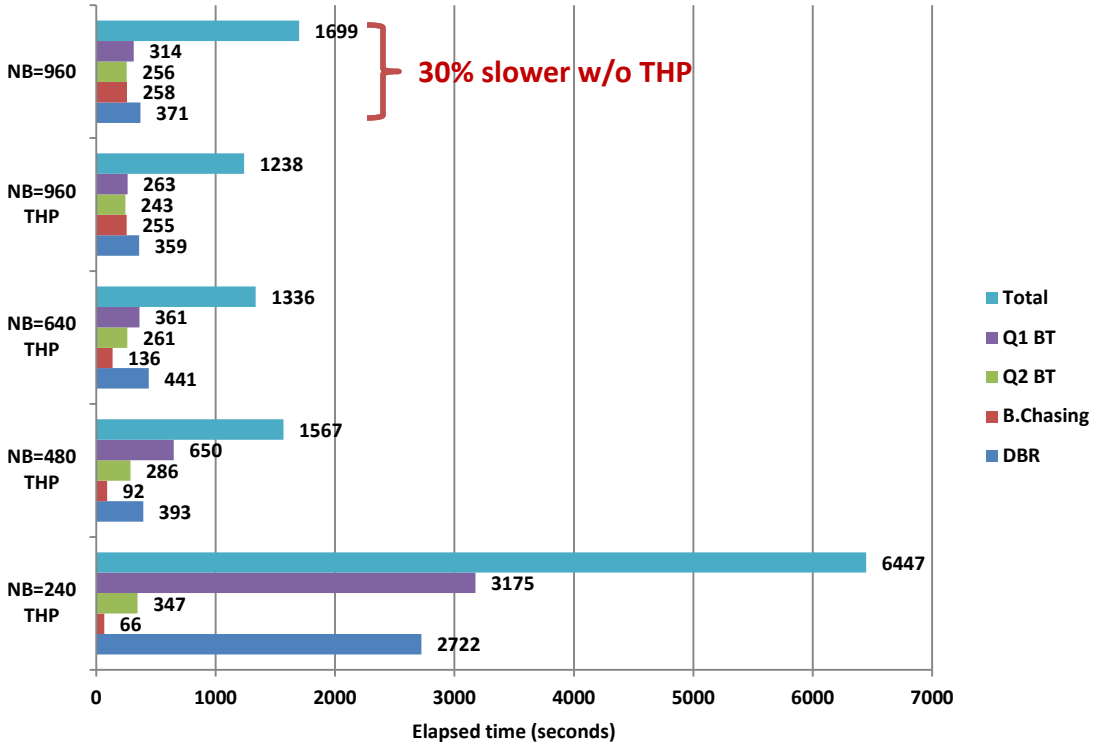
Communication/computation ratio is roughly 2/β: $\frac{\frac{N}{\alpha}}{NP}$, where α, β, NP are the BW, FP speed and number of sockets

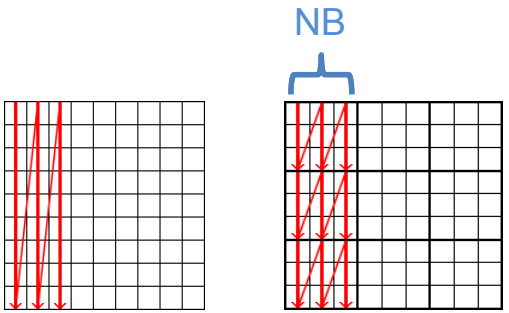
PLASMA Eigenvector Back-transformations



Illustrations from Haidar, Luszczek and Dongarra 2014 paper

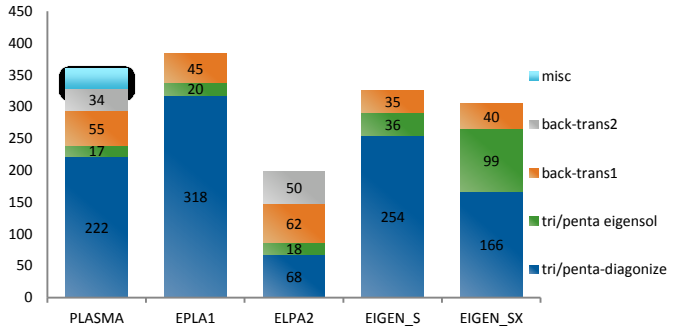
Effects of Tile & Memory Page Sizes



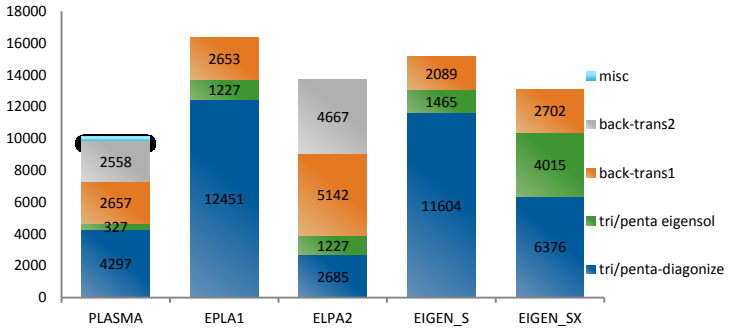


LAPACK & Tile layout

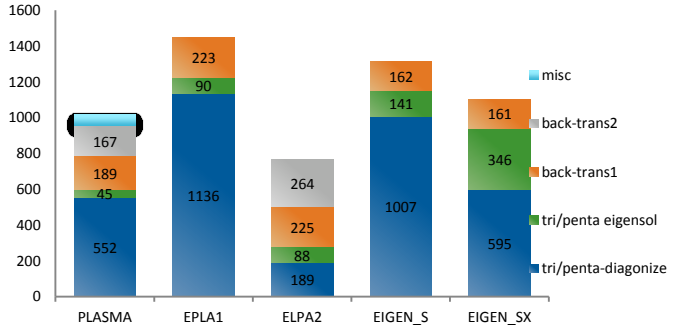
Performance Comparisons



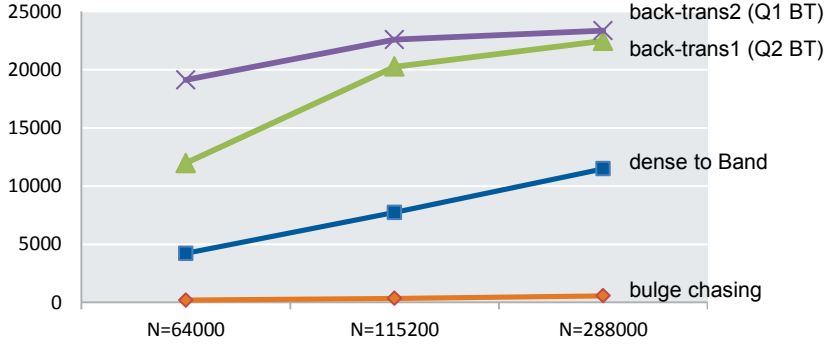
Elapsed times of N=64000 on 60 sockets



Elapsed times of N=288000 on 60 sockets



Elapsed times of N=115200 on 60 sockets



PLASMA GFLOPS improvement w.r.t. problem size

Conclusion

The PLASMA Eigensolver works well with large problem sizes. Can the strengths of PLASMA and ELPA be combined?