I Don’t Care about BLAS

Devin Matthews

Institute for Computational Engineering and Sciences, UT Austin

#smallmoleculesmatter

From QC to DGEMM

$\bar{H}\hat{R}|\Phi\rangle = E\hat{R}|\Phi\rangle$ “Simple” eigenproblem…

$r^{abef}_{ijkl},\ \bar{H}^{abij}_{cdkl}$ In terms of tensors…

$r^{abef}_{ijkl},\ W^{ai}_{ck},\ F^{i}_{k},\ t^{ab}_{ij},\ \ldots$ In terms of other tensors…

$r^{ab\bar{e}\bar{f}}_{ij\bar{k}\bar{l}}$ or $\check{r}^{abef}_{ijkl}$ With structured sparsity…

$r^{a<b\bar{e}<\bar{f}}_{i<j\bar{k}<\bar{l}}$ or $\check{r}^{abef}_{i\leq j\leq k\leq l}$ With symmetry…

$r^{abef}_{0000},\ r^{abef}_{0001},\ r^{abef}_{0002},\ \ldots$ With slicing (or blocking etc.)…

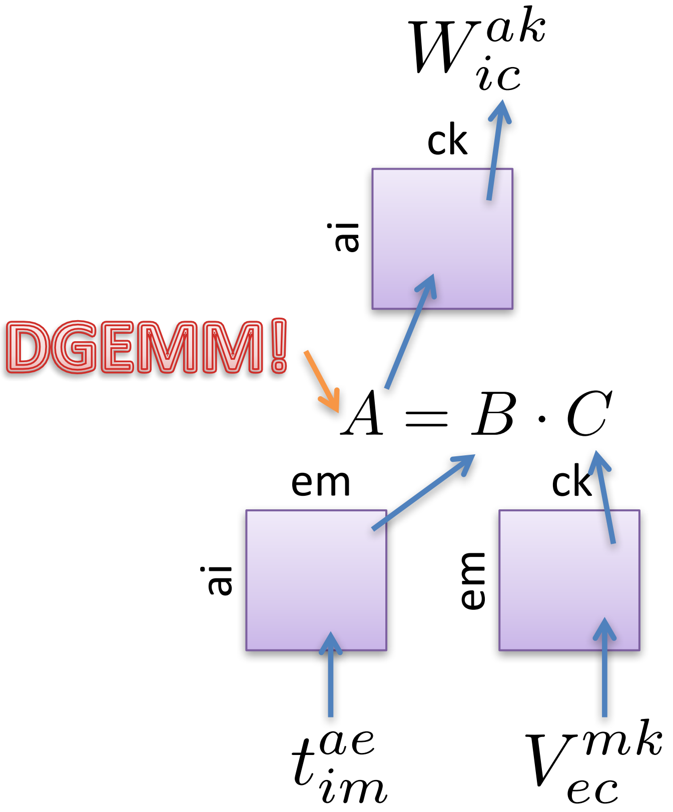
$r^{(ab)_{\gamma_{ab}}}_{(ef)_{\gamma_{ef}}}$ With more sparsity…

$r^{ab}_{ef} \in \mathbb{R}^{n_a \otimes n_b \otimes n_e \otimes n_f}$ In terms of dense tensors…

DGEMM! → $A = B \cdot C$ In terms of matrices.

Tensor Contraction Today

“TTDT”



“LoG”

$$W^{ak}_{ic} = t^{ae}_{im} \bullet V^{mk}_{ec}$$

$$[W_{ak}]_{ic} = \sum_m [t_{am}]_{ei} \bullet [V_{mk}]_{ec} \quad \forall\, i, e, c$$

```
for i
    for e
        for c
            DGEMM
```

DGEMM!

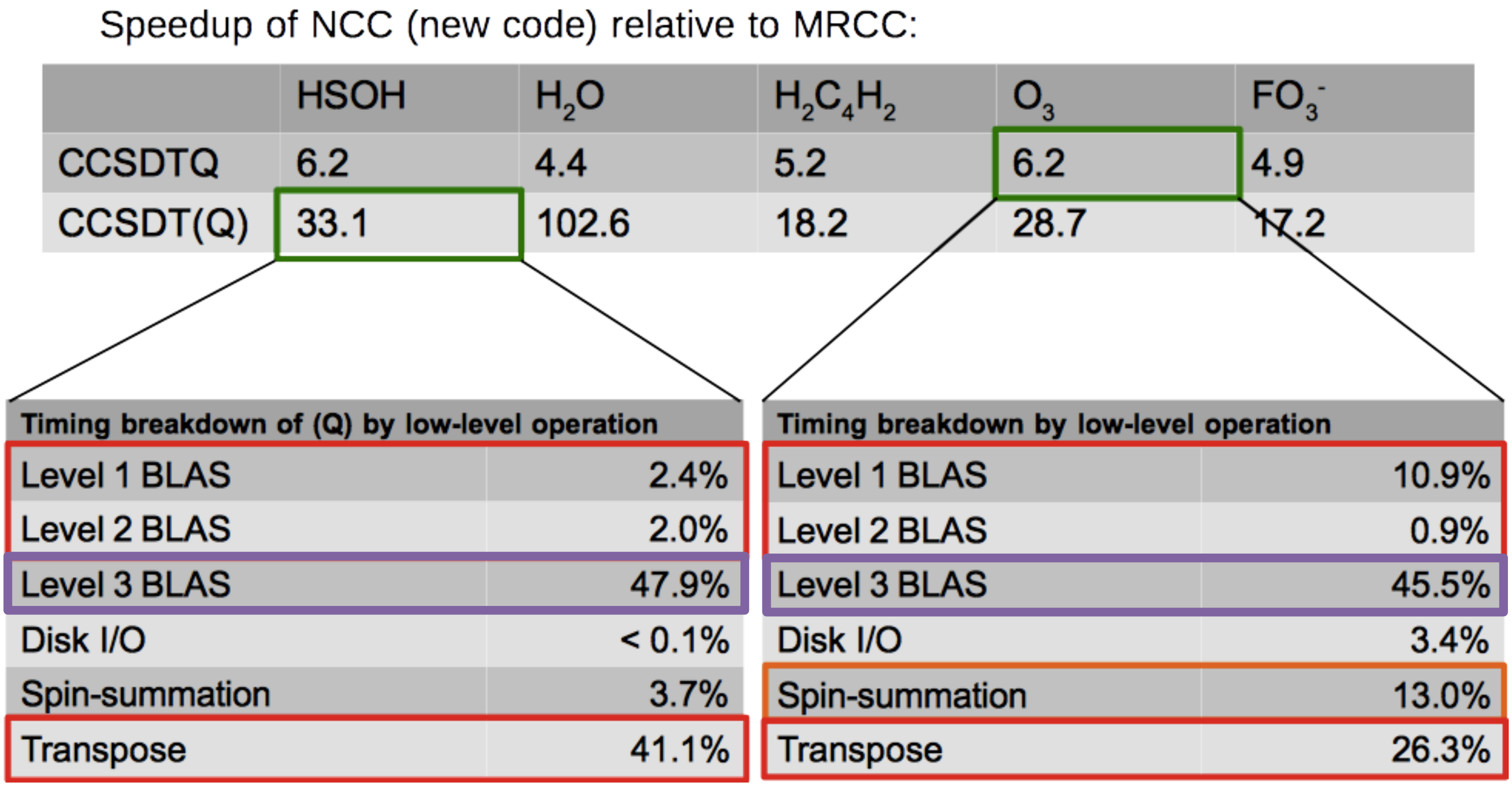
DGEMM
intel
inside

DGEMM Considered Harmful

- Tensors have to be **transposed** in order to use DGEMM.
- DGEMM needs **dense** matrices. If our tensors have **structure** (permutational symmetry, point group symmetry, sparsity, etc.) we have to **expand** or **block** them.
- Point group symmetry is efficiently handled with the **Direct Product Decomposition** (DPD), but we want to *automate* and *optimize* it.
- **Blocking** reduces the size of individual DGEMM calls. Can we **aggregate** these into more efficient operations?

DPD: Stanton, J.F.; Gauss, J.; Watts, J.D.; Bartlett, R.J. *J. Chem. Phys.* **1991**, *94*, 4334.

How Much Does Transpose Cost?



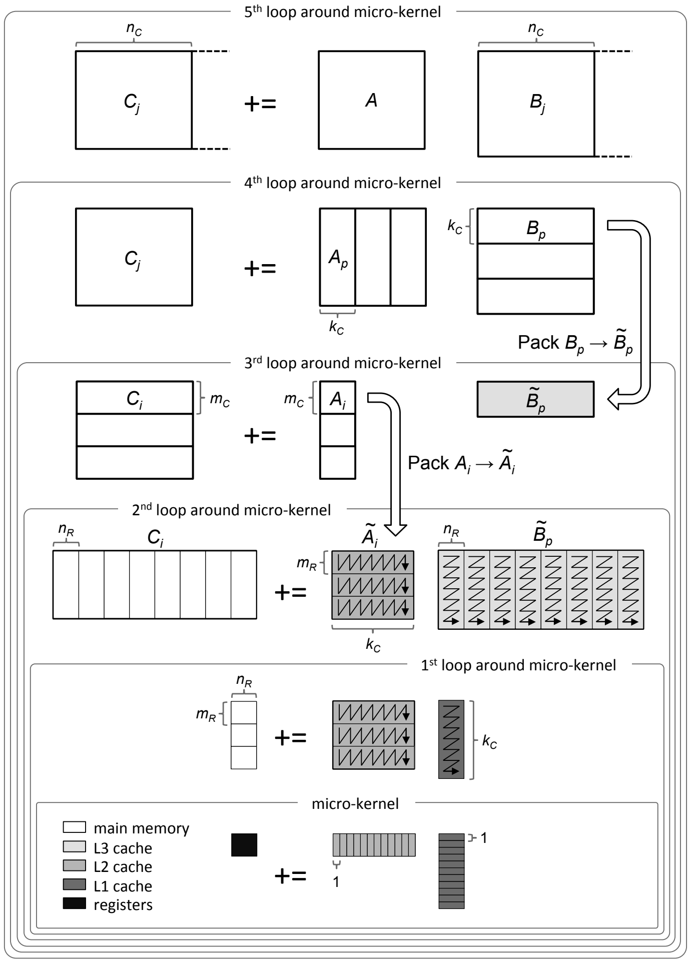
Speedup of NCC (new code) relative to MRCC:

	HSOH	H_2O	$H_2C_4H_2$	O_3	FO_3^-
CCSDTQ	6.2	4.4	5.2	6.2	4.9
CCSDT(Q)	33.1	102.6	18.2	28.7	17.2

Timing breakdown of (Q) by low-level operation	
Level 1 BLAS	2.4%
Level 2 BLAS	2.0%
Level 3 BLAS	47.9%
Disk I/O	< 0.1%
Spin-summation	3.7%
Transpose	41.1%

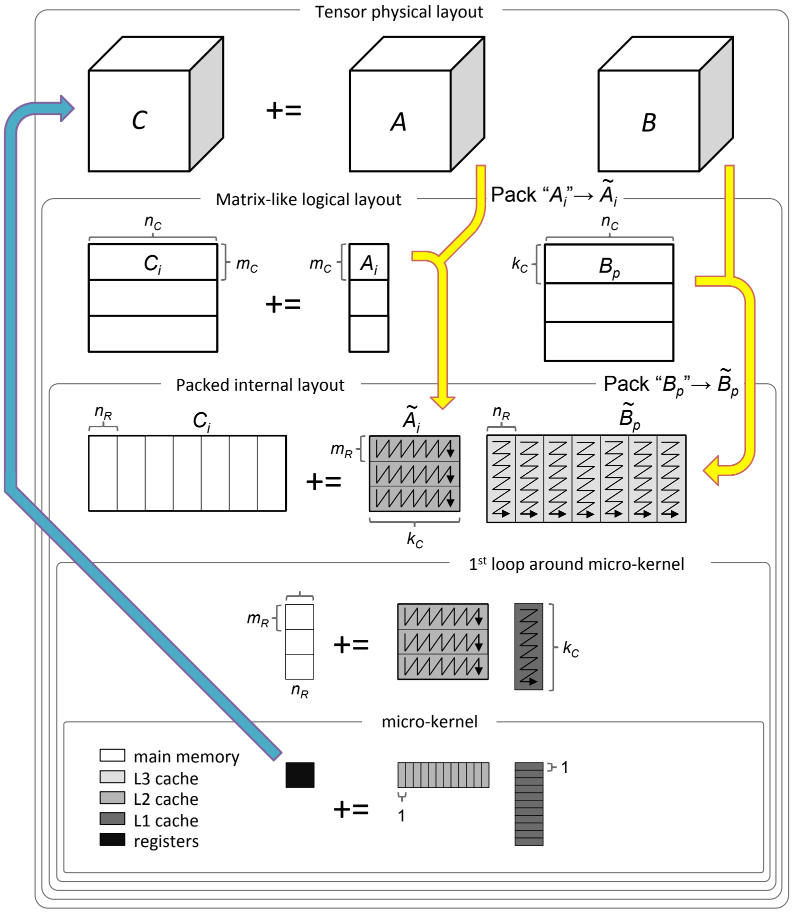
Timing breakdown by low-level operation	
Level 1 BLAS	10.9%
Level 2 BLAS	0.9%
Level 3 BLAS	45.5%
Disk I/O	3.4%
Spin-summation	13.0%
Transpose	26.3%

BLIS → TBLIS

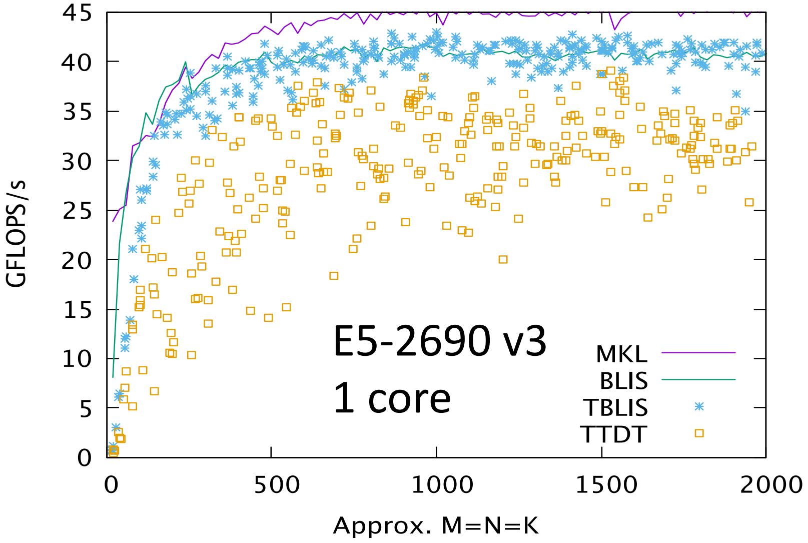

5th loop around micro-kernel
n_C
C_j
+=
A
B_j
4th loop around micro-kernel
C_j
+=
A_p
k_C
B_p
Pack $B_p \rightarrow \tilde{B}_p$
3rd loop around micro-kernel
C_i
m_C
+=
A_i
$\tilde{B}_p$
Pack $A_i \rightarrow \tilde{A}_i$
2nd loop around micro-kernel
n_R
C_i
m_R
+=
$\tilde{A}_i$
$\tilde{B}_p$
1st loop around micro-kernel
micro-kernel
1
main memory
L3 cache
L2 cache
L1 cache
registers

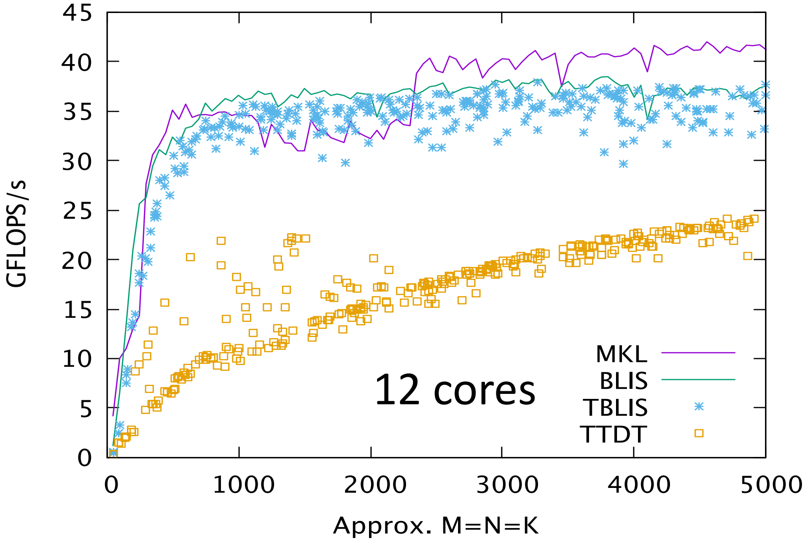
Tensor to matrix packing kernel

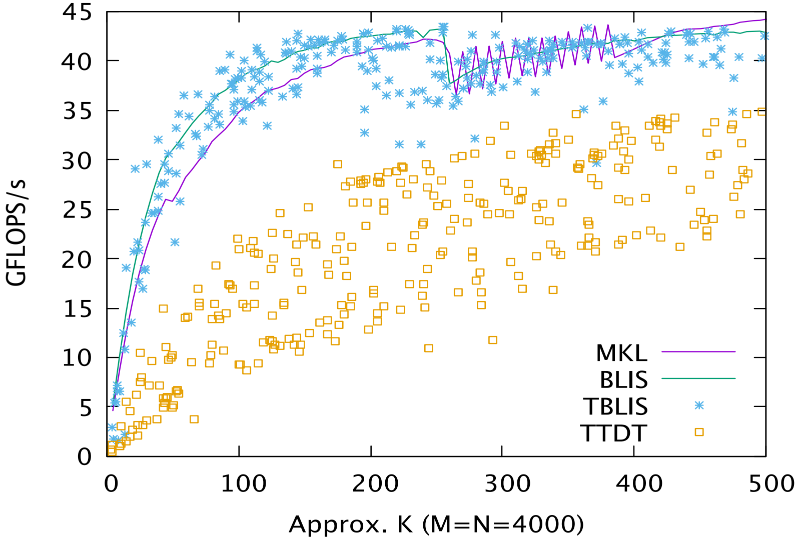
(Possibly) scattered update

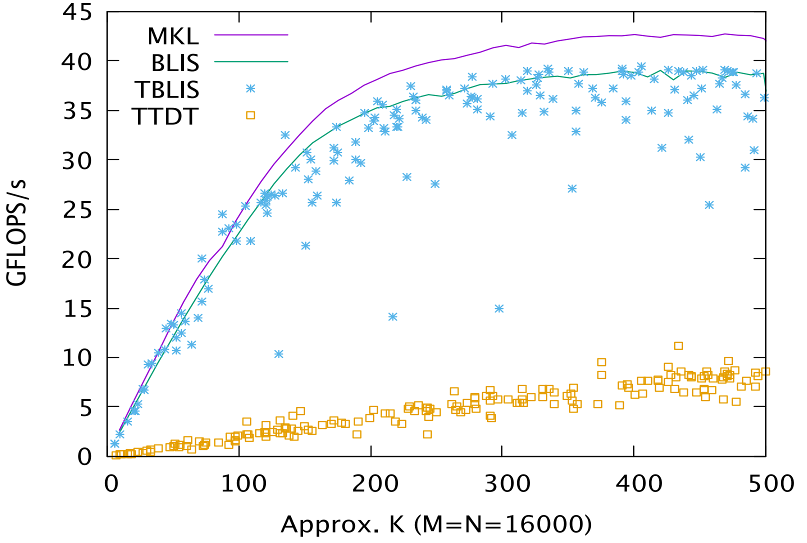


Results for Dense Tensors



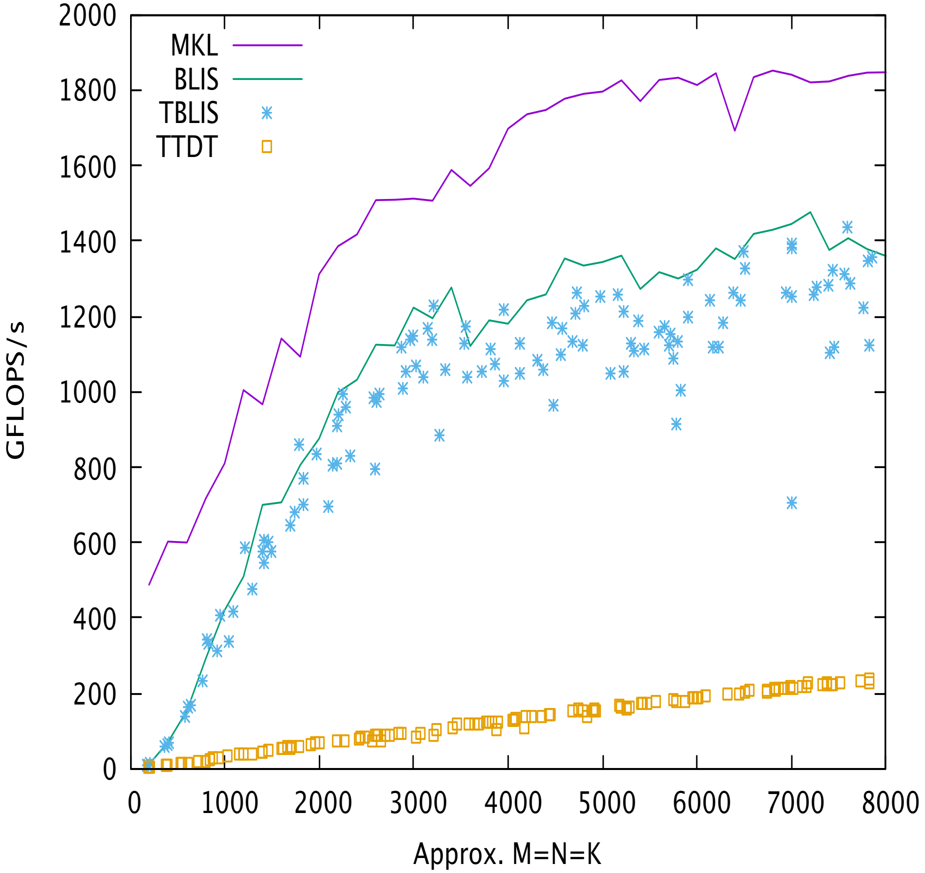




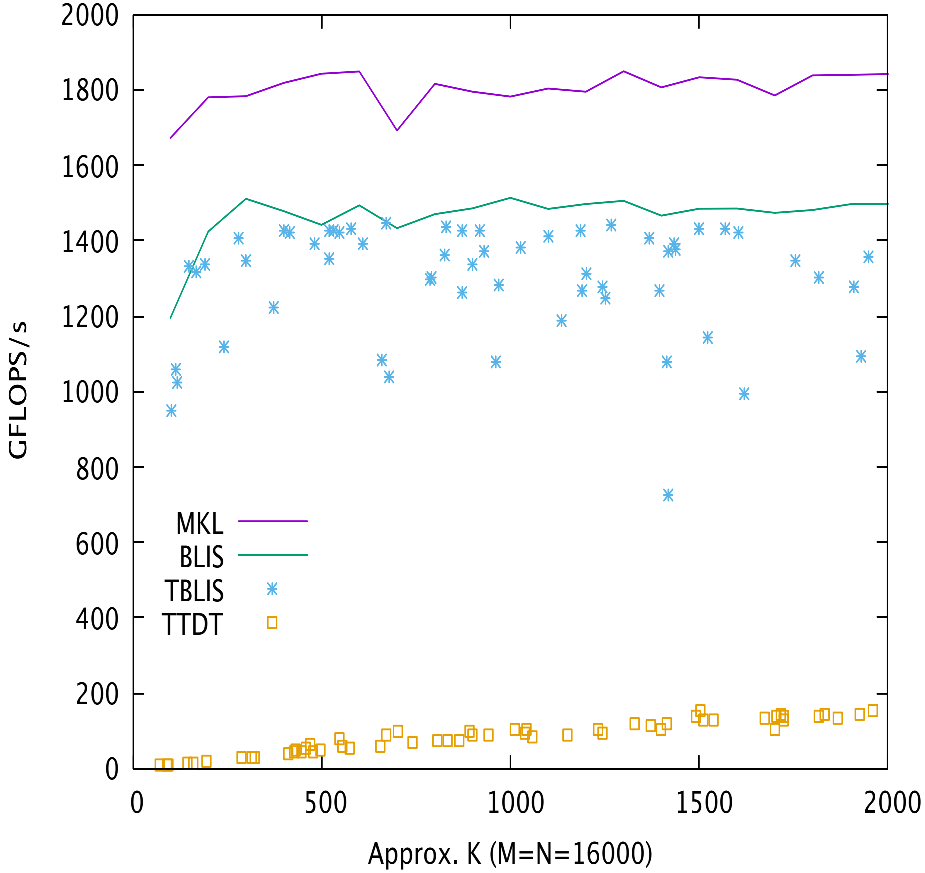


Works Great on Xeon Phi Too

“Square” MM and TC on Xeon Phi 7210

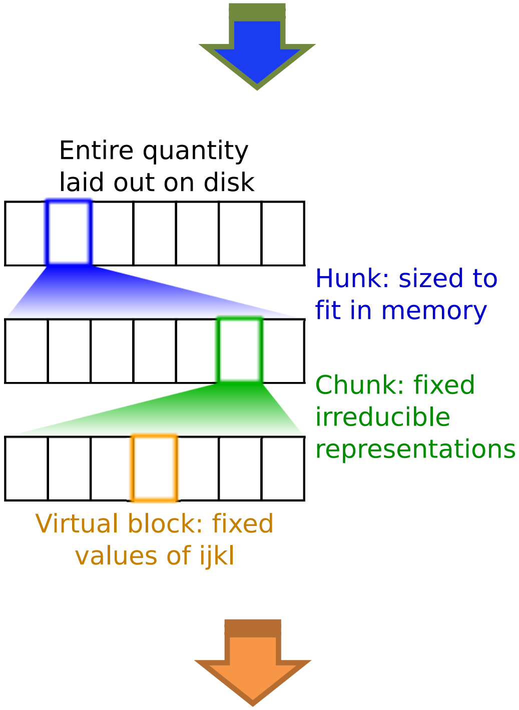


“Rank-k” MM and TC on Xeon Phi 7210



Quasi-Sparse Tensor Contractions

$T^{abc}_{i\leq j\leq k}$, $R^{abcd}_{i\leq j\leq k\leq l}$, $W^{abc}_{i\leq je}$, etc.



Entire quantity laid out on disk

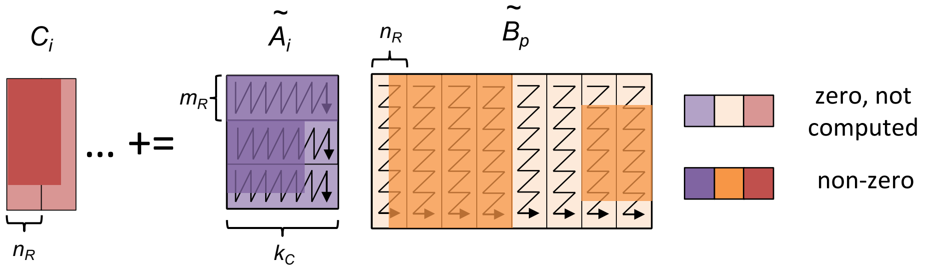
Hunk: sized to fit in memory

Chunk: fixed irreducible representations

Virtual block: fixed values of ijkl

$$Z^{abcd}_{0,13,2,4} += T^{abc}_{0,5,21} W^{5,21;d}_{13,2,4}$$

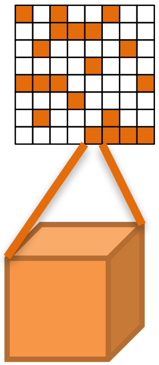
Option #1: Batch within TBLIS framework



zero, not computed

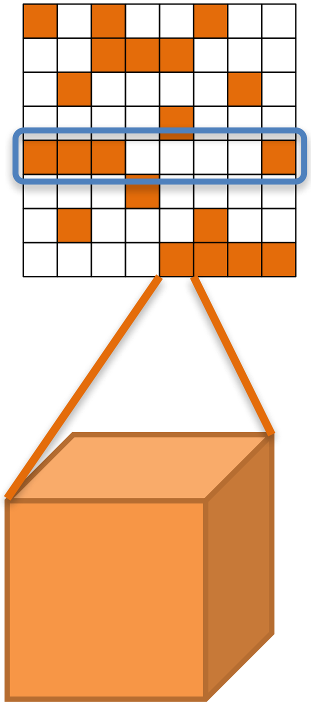
non-zero

Option #2: Batch outside of TBLIS framework



```
for k'
  pfor n'
    pfor m'
      if $\exists A(m',k') \land$
         $\exists B(k',n') \land$
         $\exists C(m',n')$
        contract(…)
      endif
    done
  done
done
```

Quasi-Sparse Tensor Contractions

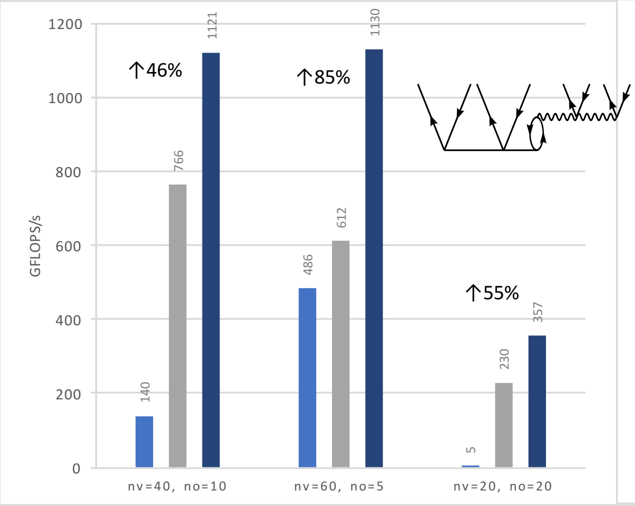


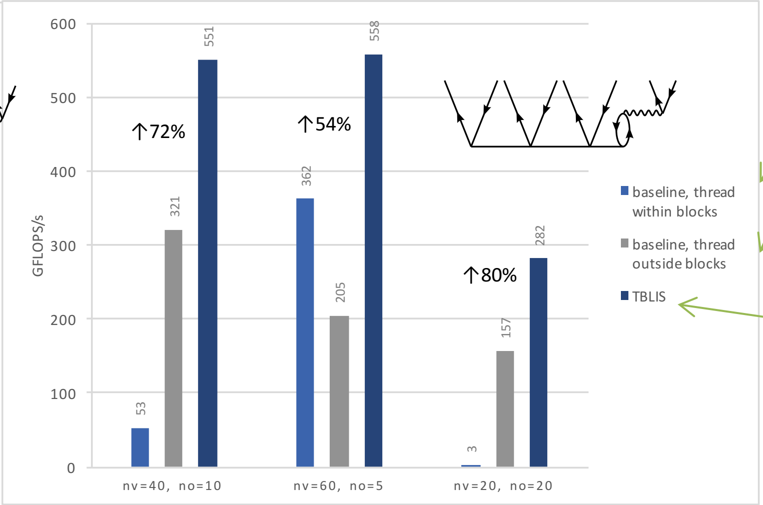
```
for k’
  pfor n’
    pfor m’
      if ∃A(m’,k’) ∧
         ∃B(k’,n’) ∧
         ∃C(m’,n’)
        contract(…)
      endif
    done
  done
done
```

Use hierarchical dynamic+static parallelism and aggregate blocks when possible.

```
Split communicator into c_in & c_out
pfor n’ over c_out
  pfor m’ over c_out
    ks = {}
    for k’
      if ∃A(m’,k’) ∧
         ∃B(k’,n’) ∧
         ∃C(m’,n’)
        append k’ to ks
      endif
    done
    pcontract(ks,…) over c_in
  done
done
```

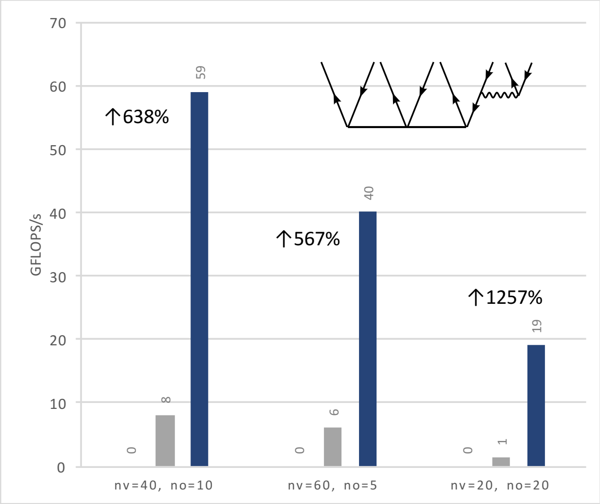
Quasi-Sparse Tensor Contractions

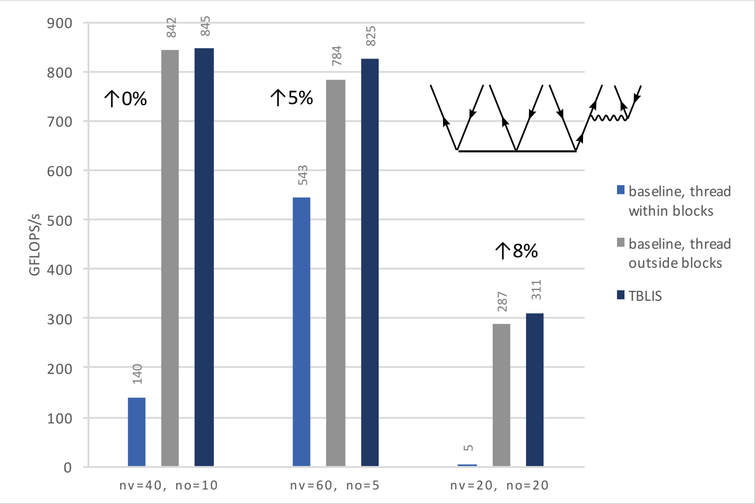




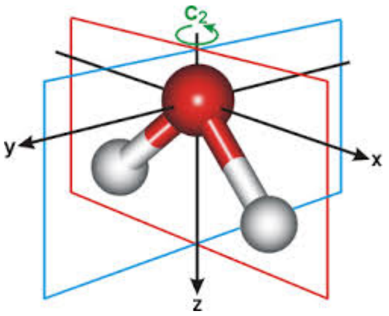
(uses TBLIS for inner tensor contraction)

(adds hierarchical multithreading and block aggregation)



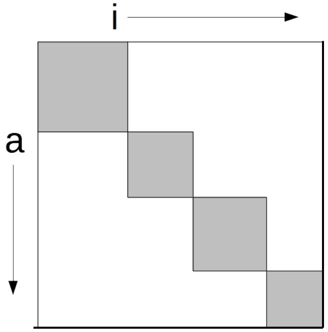


Taking Advantage of Structure



$$T_i^a \longrightarrow$$

$$\Gamma_a \otimes \Gamma_i = \Gamma_T$$



Point Group Symmetry

Cost savings proportional to g^2 (g = number of irreducible representations/blocks).

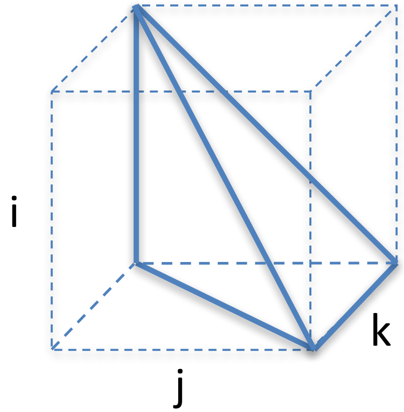
Permutational Symmetry

Factorial cost savings for increasing dimensionality.

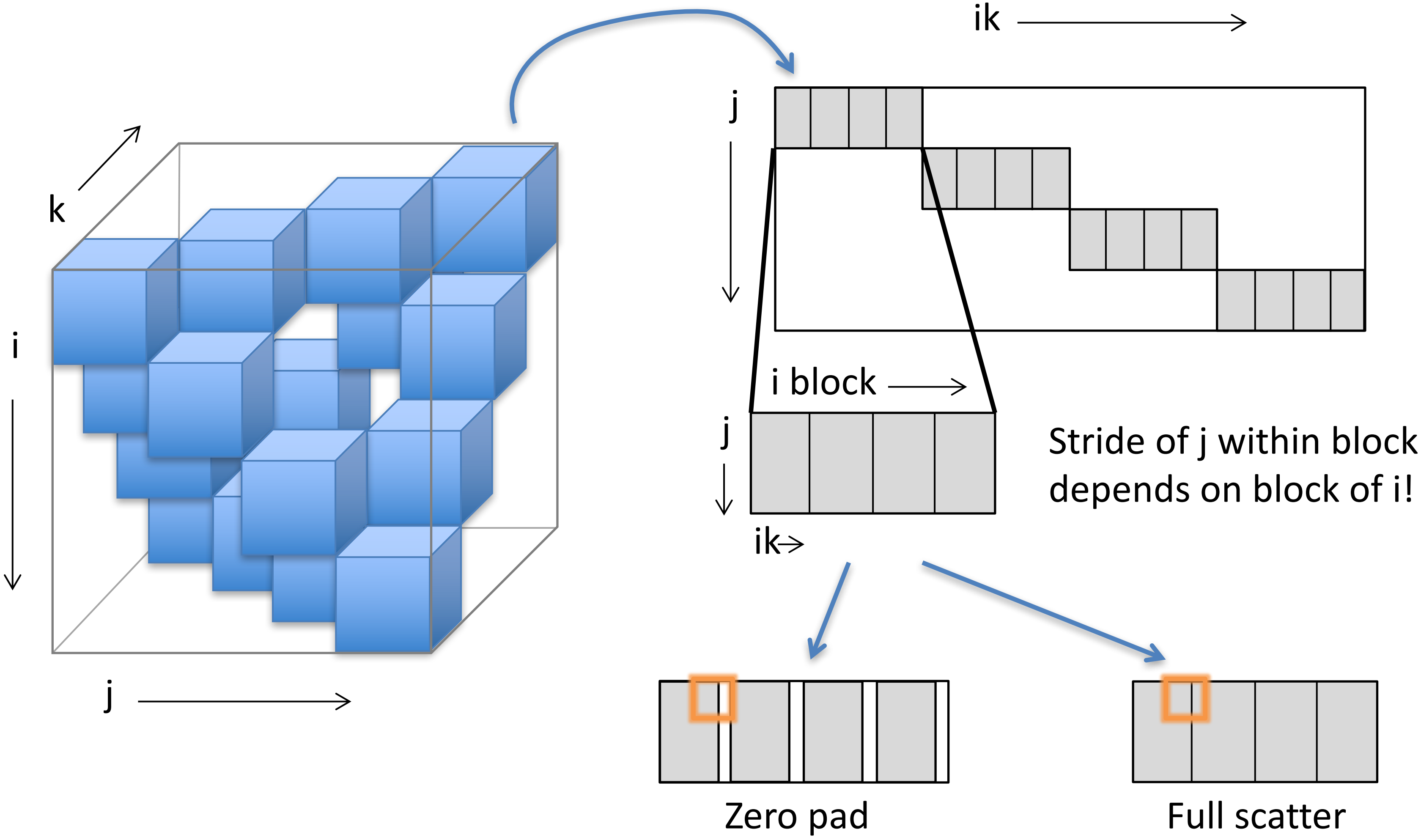
$$A_{ijk} = -A_{jik} = A_{jki} = -A_{kji} = A_{kij} = -A_{ikj}$$

⇓

$$A_{i<j<k}$$

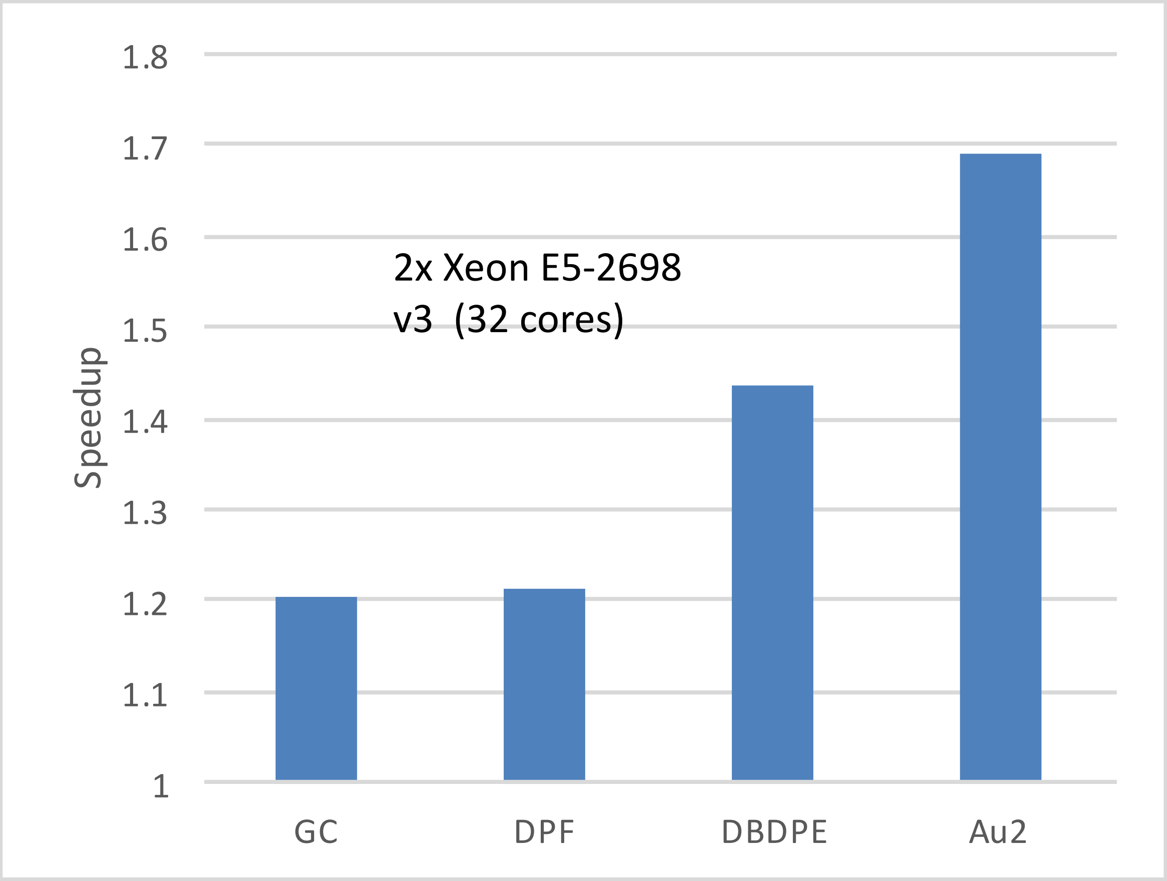


Taking Advantage of Structure



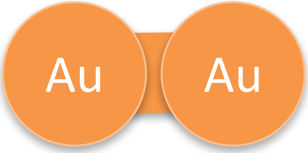
Stride of j within block depends on block of i!

Speedup in computation of coupled cluster singles and doubles (CCSD) ground state energy when using TBLIS

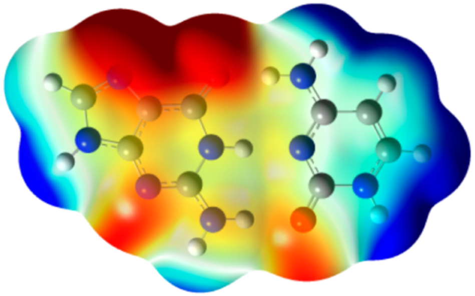


2x Xeon E5-2698 v3 (32 cores)

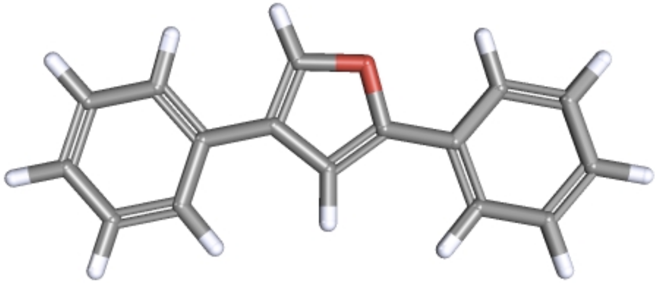
Less symmetry ↔ More symmetry



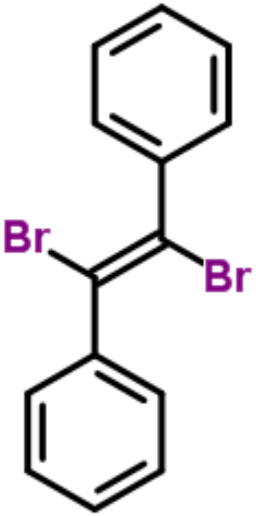
Gold dimer (**Au_2**)
all-electron, D_{2h} symmetry



Guanine-cytosine dimer (**GC**), no symmetry
Krepl et al., J. Phys. Chem. B 2013, 117, 1872



2,4,-diphenylfuran (**DPF**), C_s symmetry



(E) 1,2-dibromo-1,2-diphenylethene (**DBDPE**) planar, C_{2h} symmetry

Summary

- Novel algorithms leveraging the **BLIS methodology** can significantly outperform DGEMM-based algorithms for **tensor contraction**.
- Breaking through the DGEMM barrier allows **new algorithms** to be implemented with high efficiency.

Thanks!

THE UNIVERSITY OF TEXAS AT AUSTIN

ICES

TACC
TEXAS ADVANCED COMPUTING CENTER

NSF

intel

Arnold and Mabel BECKMAN FOUNDATION

SHPC

Robert van de Geijn

Jianyu Huang

Field Van Zee

Tyler Smith

Devangi Parikh

⟨CC|CC⟩
www.cfour.de

#smallmoleculesmatter

TBLIS on GitHub