A High-Level Approach to the Synthesis of High-Performance Codes for Quantum Chemistry

Oak Ridge National Laboratory

<u>David E. Bernholdt</u>, Venkatesh Choppella, David Dean, Robert Harrison, Thomas Papenbrock, Michael Strayer, Trey White

Pacific Northwest National Laboratory

So Hirata

Ohio State University

Gerald Baumgartner, Daniel Cociorva, Russ Pitzer, P Sadayappan, a small army of graduate students

Louisiana State University

J Ramanujam

Princeton University

Marcel Nooijen, Alexander Auer

Research at ORNL supported by the Laboratory Directed Research and Development Program
Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the US Dept. of Energy under contract DE-AC-05-00OR22725.

Research at PNNL supported by the Office of Basic Energy Sciences, U. S. Dept. of Energy
Research at OSU, Princeton, and LSU supported by the National Science Foundation Information Technology Research Program

Problem Domain: High-Accuracy Quantum Chemical Methods

- Coupled cluster methods are widely used for very high quality electronic structure calcs.
- Typical Laplace factorized CCSD(T) term:

Typical methods will have tens to hundreds of such terms

$$A3A = \tfrac{1}{2}\big(X_{ce,af}Y_{ae,cf} + X_{c\bar{e},a\bar{f}}Y_{a\bar{e},c\bar{f}} + X_{c\bar{e},\bar{a}f}Y_{\bar{a}\bar{e},cf}$$
$$+ X_{\bar{c}e,a\bar{f}}Y_{ae,\bar{c}\bar{f}} + X_{\bar{c}e,\bar{a}f}Y_{\bar{a}e,\bar{c}f} + X_{\bar{c}\bar{e},\bar{a}\bar{f}}Y_{\bar{a}\bar{e},\bar{c}\bar{f}}\big)$$

$$X_{ce,af} = t_{ij}^{ce}t_{ij}^{af} \qquad Y_{ae,cf} = \langle ab\|ek\rangle\langle cb\|fk\rangle$$

- Indices i, j, k O(O=100) values, a, b, c, e, f O(V=3000)
- Term costs $O(OV^5) \approx 10^{19}$ FLOPs; Integrals ~ 1000 FLOPs each
- $O(V^4)$ terms ~ 500 TB memory each

Problems

Complexity of Methods

- Implementation takes months
- Experimentation required to develop new methods

Complexity of Computers

- Different architectures have significantly different performance characteristics

Our Solution

"Tensor Contraction Engine"

- Tensor contraction expressions as input
- (Fortran) source code as output

Generated code increases productivity

Optimize generated code for target computer

What's Novel?

Code generation merely for productivity, historically

- Imitate what researcher would do – but quicker

We treat as a computer science problem

- Like a compiler
- Algorithmic choices explored rigorously and exhaustively

A High-Level Language for Tensor Contraction Expressions

```
range V = 3000;
range O = 100;

index a,b,c,d,e,f : V;
index i,j,k : O;

mlimit = 1000000000000;

function F1(V,V,V,O);
function F2(V,V,V,O);

procedure P(in T1[O,O,V,V], in T2[O,O,V,V], out X)=

begin
   X == sum[ sum[F1(a,b,f,k) * F2(c,e,b,k), {b,k}]
                  * sum[T1[i,j,a,e] * T2[i,j,c,f], {i,j}],
                  {a,e,c,f}];
end
```

$$A3A = \tfrac{1}{2}\left(X_{ce,af} Y_{ae,cf} + X_{c\bar{e},a\bar{f}} Y_{a\bar{e},c\bar{f}} + X_{\bar{c}\bar{e},af} Y_{\bar{a}\bar{e},cf} + X_{\bar{c}e,a\bar{f}} Y_{ae,\bar{c}\bar{f}} + X_{\bar{c}e,\bar{a}f} Y_{\bar{a}e,\bar{c}f} + X_{\bar{c}\bar{e},\bar{a}\bar{f}} Y_{\bar{a}\bar{e},\bar{c}\bar{f}} \right)$$

$$X_{ce,af} = t_{ij}^{ce} t_{ij}^{af} \qquad Y_{ae,cf} = \langle ab \| ek \rangle \langle cb \| fk \rangle$$

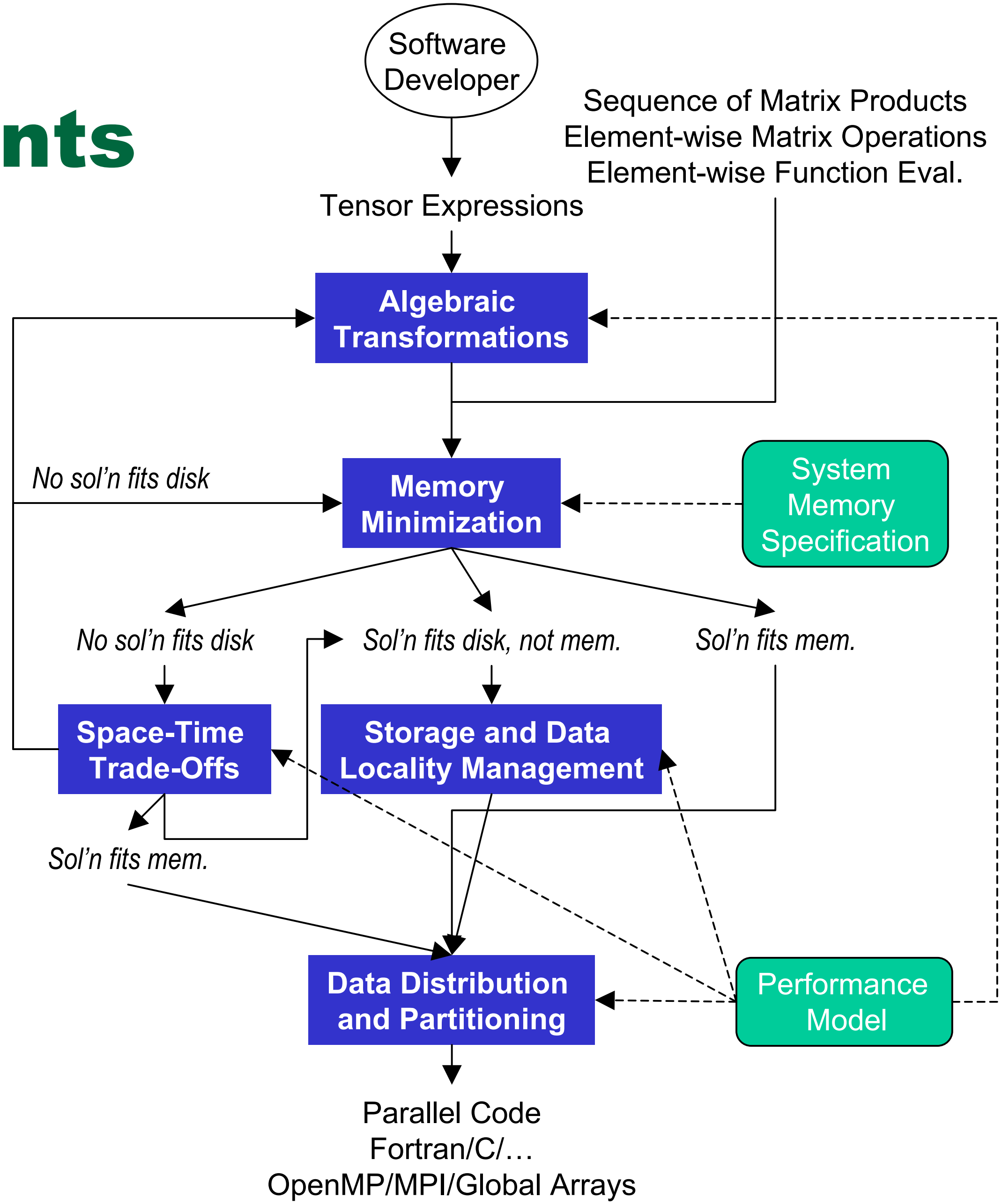
CCSD Doubles Equation

```
hbar[a,b,i,j] == sum[f[b,c]*t[i,j,a,c],{c}] -sum[f[k,c]*t[k,b]*t[i,j,a,c],{k,c}] +sum[f[a,c]*t[i,j,c,b],{c}] -sum[f[k,c]*t[k,a]*t[i,j,c,b],{k,c}] -
    sum[f[k,j]*t[i,k,a,b],{k}] -sum[f[k,c]*t[j,c]*t[i,k,a,b],{k,c}] -sum[f[k,i]*t[j,k,b,a],{k}] -sum[f[k,c]*t[i,c]*t[j,k,b,a],{k,c}]
    +sum[t[i,c]*t[j,d]*v[a,b,c,d],{c,d}] +sum[t[i,j,c,d]*v[a,b,c,d],{c,d}] +sum[t[j,c]*v[a,b,i,c],{c}] -sum[t[k,b]*v[a,k,i,j],{k}]
    +sum[t[i,c]*v[b,a,j,c],{c}] -sum[t[k,a]*v[b,k,j,i],{k}] -sum[t[k,d]*t[i,j,c,b]*v[k,a,c,d],{k,c,d}] -sum[t[i,c]*t[j,k,b,d]*v[k,a,c,d],{k,c,d}] -
    sum[t[j,c]*t[k,b]*v[k,a,c,i],{k,c}] +2*sum[t[j,k,b,c]*v[k,a,c,i],{k,c}] -sum[t[j,k,c,b]*v[k,a,c,i],{k,c}] -sum[t[i,c]*t[j,d]*t[k,b]*v[k,a,d,c],{k,c,d}]
    +2*sum[t[k,d]*t[i,j,c,b]*v[k,a,d,c],{k,c,d}] -sum[t[k,b]*t[i,j,c,d]*v[k,a,d,c],{k,c,d}] -sum[t[j,d]*t[i,k,c,b]*v[k,a,d,c],{k,c,d}]
    +2*sum[t[i,c]*t[j,k,b,d]*v[k,a,d,c],{k,c,d}] -sum[t[i,c]*t[j,k,d,b]*v[k,a,d,c],{k,c,d}] -sum[t[j,k,b,c]*v[k,a,i,c],{k,c}] -
    sum[t[i,c]*t[k,b]*v[k,a,j,c],{k,c}] -sum[t[i,k,c,b]*v[k,a,j,c],{k,c}] -sum[t[i,c]*t[j,d]*t[k,a]*v[k,b,c,d],{k,c,d}] -
    sum[t[k,d]*t[i,j,a,c]*v[k,b,c,d],{k,c,d}] -sum[t[k,a]*t[i,j,c,d]*v[k,b,c,d],{k,c,d}] +2*sum[t[j,d]*t[i,k,a,c]*v[k,b,c,d],{k,c,d}] -
    sum[t[j,d]*t[i,k,c,a]*v[k,b,c,d],{k,c,d}] -sum[t[i,c]*t[j,k,d,a]*v[k,b,c,d],{k,c,d}] -sum[t[i,c]*t[k,a]*v[k,b,c,j],{k,c}]
    +2*sum[t[i,k,a,c]*v[k,b,c,j],{k,c}] -sum[t[i,k,c,a]*v[k,b,c,j],{k,c}] +2*sum[t[k,d]*t[i,j,a,c]*v[k,b,d,c],{k,c,d}] -
    sum[t[j,d]*t[i,k,a,c]*v[k,b,d,c],{k,c,d}] -sum[t[j,c]*t[k,a]*v[k,b,i,c],{k,c}] -sum[t[j,k,c,a]*v[k,b,i,c],{k,c}] -sum[t[i,k,a,c]*v[k,b,j,c],{k,c}]
    +sum[t[i,c]*t[j,d]*t[k,a]*t[l,b]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[k,b]*t[l,d]*t[i,j,a,c]*v[k,l,c,d],{k,l,c,d}] -
    2*sum[t[k,a]*t[l,d]*t[i,j,c,b]*v[k,l,c,d],{k,l,c,d}] +sum[t[k,a]*t[l,b]*t[i,j,c,d]*v[k,l,c,d],{k,l,c,d}] -
    2*sum[t[j,c]*t[l,d]*t[i,k,a,b]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[j,d]*t[l,b]*t[i,k,a,c]*v[k,l,c,d],{k,l,c,d}]
    +sum[t[j,d]*t[l,b]*t[i,k,c,a]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,c]*t[l,d]*t[j,k,b,a]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,c]*t[l,a]*t[j,k,b,d]*v[k,l,c,d],{k,l,c,d}]
    +sum[t[i,c]*t[l,b]*t[j,k,d,a]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,k,c,d]*t[j,l,b,a]*v[k,l,c,d],{k,l,c,d}] +4*sum[t[i,k,a,c]*t[j,l,b,d]*v[k,l,c,d],{k,l,c,d}] -
    2*sum[t[i,k,c,a]*t[j,l,b,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,k,a,b]*t[j,l,c,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,k,a,c]*t[j,l,d,b]*v[k,l,c,d],{k,l,c,d}]
    +sum[t[i,k,c,a]*t[j,l,d,b]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,c]*t[j,d]*t[k,l,a,b]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,j,c,d]*t[k,l,a,b]*v[k,l,c,d],{k,l,c,d}] -
    2*sum[t[i,j,c,b]*t[k,l,a,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,j,a,c]*t[k,l,b,d]*v[k,l,c,d],{k,l,c,d}] +sum[t[j,c]*t[k,b]*t[l,a]*v[k,l,c,i],{k,l,c}]
    +sum[t[l,c]*t[j,k,b,a]*v[k,l,c,i],{k,l,c}] -2*sum[t[l,a]*t[j,k,b,c]*v[k,l,c,i],{k,l,c}] +sum[t[l,a]*t[j,k,c,b]*v[k,l,c,i],{k,l,c}] -
    2*sum[t[k,c]*t[j,l,b,a]*v[k,l,c,i],{k,l,c}] +sum[t[k,a]*t[j,l,b,c]*v[k,l,c,i],{k,l,c}] +sum[t[k,b]*t[j,l,c,a]*v[k,l,c,i],{k,l,c}]
    +sum[t[j,c]*t[l,k,a,b]*v[k,l,c,i],{k,l,c}] +sum[t[i,c]*t[k,a]*t[l,b]*v[k,l,c,j],{k,l,c}] +sum[t[l,c]*t[i,k,a,b]*v[k,l,c,j],{k,l,c}] -
    2*sum[t[l,b]*t[i,k,a,c]*v[k,l,c,j],{k,l,c}] +sum[t[l,b]*t[i,k,c,a]*v[k,l,c,j],{k,l,c}] +sum[t[i,c]*t[k,l,a,b]*v[k,l,c,j],{k,l,c}]
    +sum[t[j,c]*t[l,d]*t[i,k,a,b]*v[k,l,d,c],{k,l,c,d}] +sum[t[j,d]*t[l,b]*t[i,k,a,c]*v[k,l,d,c],{k,l,c,d}] +sum[t[j,d]*t[l,a]*t[i,k,c,b]*v[k,l,d,c],{k,l,c,d}] -
    2*sum[t[i,k,c,d]*t[j,l,b,a]*v[k,l,d,c],{k,l,c,d}] -2*sum[t[i,k,a,c]*t[j,l,b,d]*v[k,l,d,c],{k,l,c,d}] +sum[t[i,k,c,a]*t[j,l,b,d]*v[k,l,d,c],{k,l,c,d}]
    +sum[t[i,k,a,b]*t[j,l,c,d]*v[k,l,d,c],{k,l,c,d}] +sum[t[i,k,c,b]*t[j,l,d,a]*v[k,l,d,c],{k,l,c,d}] +sum[t[i,k,a,c]*t[j,l,d,b]*v[k,l,d,c],{k,l,c,d}]
    +sum[t[k,a]*t[l,b]*v[k,l,i,j],{k,l}] +sum[t[k,l,a,b]*v[k,l,i,j],{k,l}] +sum[t[k,b]*t[l,d]*t[i,j,a,c]*v[l,k,c,d],{k,l,c,d}]
    +sum[t[k,a]*t[l,d]*t[i,j,c,b]*v[l,k,c,d],{k,l,c,d}] +sum[t[i,c]*t[l,d]*t[j,k,b,a]*v[l,k,c,d],{k,l,c,d}] -2*sum[t[i,c]*t[l,a]*t[j,k,b,d]*v[l,k,c,d],{k,l,c,d}]
    +sum[t[i,c]*t[l,a]*t[j,k,d,b]*v[l,k,c,d],{k,l,c,d}] +sum[t[i,j,c,b]*t[k,l,a,d]*v[l,k,c,d],{k,l,c,d}] +sum[t[i,j,a,c]*t[k,l,b,d]*v[l,k,c,d],{k,l,c,d}] -
    2*sum[t[l,c]*t[i,k,a,b]*v[l,k,c,j],{k,l,c}] +sum[t[l,b]*t[i,k,a,c]*v[l,k,c,j],{k,l,c}] +sum[t[l,a]*t[i,k,c,b]*v[l,k,c,j],{k,l,c}] +v[a,b,i,j]
```

In the coupled cluster method with single and double excitations (CCSD) the "singles" and "doubles" equations are iterated until convergence and that solution is used to evaluate the molecular energy

TCE Components

- Algebraic Transformations
  - Minimize operation count
- Memory Minimization
  - Reduce intermediate storage
- Space-Time Transformation
  - Trade-offs btw storage and recomputation
- Storage Management and Data Locality Optimization
  - Optimize use of storage hierarchy
- Data Distribution and Partitioning
  - Optimize parallel layout



Algebraic Transformations: Operation Minimization

$$S(a,b,i,j) = \sum_{c,d,e,f,k,l} A(a,c,i,k)B(b,e,f,l)C(d,f,j,k)D(c,d,e,l)$$

- Requires 4 * N^{10} operations if indices a-l have range N

- Using associative, commutative, distributive laws acceptable
- Optimal formula sequence requires only 6 * N^6 operations

$$T1(b,c,d,f) = \sum_{e,l} B(b,e,f,l)D(c,d,e,l)$$

$$T2(b,c,j,k) = \sum_{d,f} T1(b,c,d,f)C(d,f,j,k)$$

$$S(a,b,i,j) = \sum_{c,k} T2(b,c,j,k)A(a,c,i,k)$$

Operation Minimal Form

for a, e, c, f

  for i, j

    $X_{aecf} \mathrel{+}= T_{ijae} T_{ijcf}$

for c, e, b, k

  $T1_{cebk} = f1(c, e, b, k)$

for a, f, b, k

  $T2_{afbk} = f2(a, f, b, k)$

for c, e, a, f

  for b, k

    $Y_{ceaf} \mathrel{+}= T1_{cebk} T2_{afbk}$

for c, e, a, f

  $E \mathrel{+}= X_{aecf} Y_{ceaf}$

Inputs

External function calls

Output

array	space	time
X	V^4	V^4O^2
T1	V^3O	$C_{f1}V^3O$
T2	V^3O	$C_{f2}V^3O$
Y	V^4	V^5O
E	1	V^4

a .. f: range V = 1000 .. 3000

i .. k: range O = 30 .. 100

Memory-Minimal Form

```
for a, f, b, k
  [ T2_afbk = f2(a, f, b, k)
for c, e
  [ for b, k
  |   [ T1_bk = f1(c, e, b, k)
  | for a, f
  |   [ for i, j
  |   |   [ X += T_ijae T_ijcf
  |   | for b, k
  |   |   [ Y += T1_bk T2_afbk
  |   | E += X Y
```

Fusion of loops allows reduction of rank of arrays

array	space	time
X	1	V^4O^2
T1	VO	$C_{f1}V^3O$
T2	V^3O	$C_{f2}V^3O$
Y	1	V^5O
E	1	V^4

a .. f: range V = 3000
i .. k: range O = 100

Redundant Computation Allows Full Fusion

```
for a, e, c, f
    for i, j
        X += T_ijae T_ijcf
    for b, k
        T1 = f1(c, e, b, k)
        T2 = f2(a, f, b, k)
        Y += T1 T2
    E += X Y
```

array	space	time
X	1	V^4O^2
T1	1	$C_{f1}V^5O$
T2	1	$C_{f2}V^5O$
Y	1	V^5O
E	1	V^4

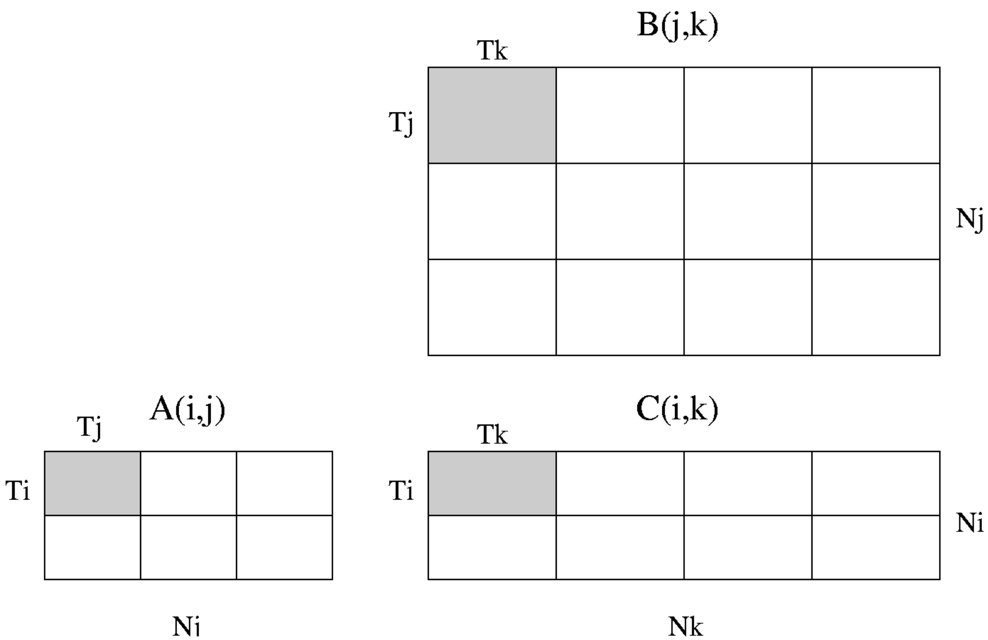
Tiling to Reduce Recomputation

for a^t, e^t, c^t, f^t

- for a, e, c, f
  - for i, j
    - $X_{aecf} \mathrel{+}= T_{ijae} T_{ijcf}$
- for b, k
  - for c, e
    - $T1_{ce} = f1(c, e, b, k)$
  - for a, f
    - $T2_{af} = f2(a, f, b, k)$
  - for c, e, a, f
    - $Y_{ceaf} \mathrel{+}= T1_{ce} T2_{af}$
- for c, e, a, f
  - $E \mathrel{+}= X_{aecf} Y_{ceaf}$

array	space	time
X	B^4	V^4O^2
T1	B^2	$C_{f1}(V/B)^2V^3O$
T2	B^2	$C_{f2}(V/B)^2V^3O$
Y	B^4	V^5O
E	1	V^4

Tiling to Minimize Memory Access Time




```
FOR ii = 1, Ni, Ti
    FOR jj = 1, Nj, Tj
        FOR kk = 1, Nk, Tk
            FOR i = ii, ii + Ti - 1
                FOR j = jj, jj + Tj - 1
                    FOR k = kk, kk + Tk - 1
                        C(i,k) = C(i,k) + A(i,j) * B(j,k)
END FOR k, j, i, kk, jj, ii
```

Choose Ti, Tj, and Tk such that $Ti * Tj + Ti * Tk + Tj * Tk < $ cache size

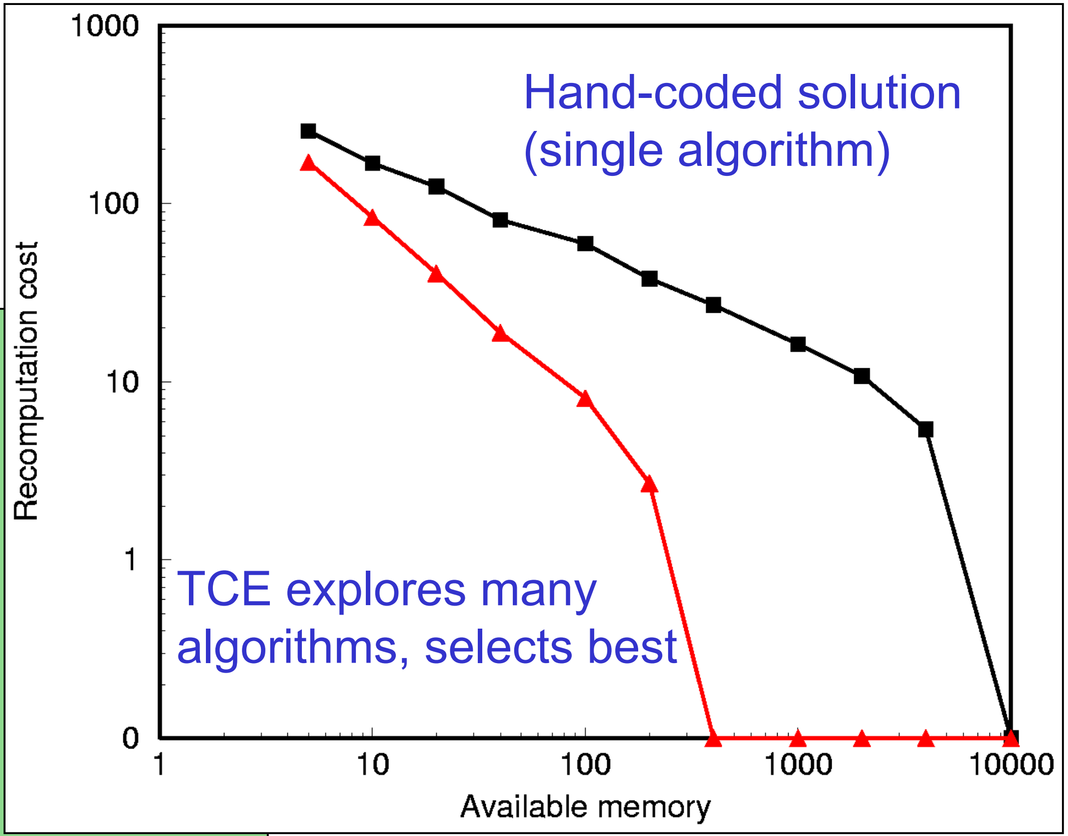
Number of cache misses:

- A(i,j): Ni * Nj
- B(j,k): Nj * Nk * Ni/Ti
- C(i,k): Ni * Nk * Nj/Tj

Same algorithm used to manage locality in disk-based algorithms

The TCE in Operation

```
range V = 3000;
range O = 100;

index a,b,c,d,e,f : V;
index i,j,k : O;

mlimit = 100000000000;

function F1(V,V,V,O);
function F2(V,V,V,O);

procedure P(in T1[O,O,V,V], in T2[O,O,V,V], out X)=

begin
    X == sum[ sum[F1(a,b,f,k) * F2(c,e,b,k), {b,k}]
                  * sum[T1[i,j,a,e] * T2[i,j,c,f], {i,j}],
                  {a,e,c,f}];
end
```



$$A3A = \tfrac{1}{2}\left(X_{ce,af}Y_{ae,cf} + X_{c\bar{e},a\bar{f}}Y_{a\bar{e},c\bar{f}} + X_{c\bar{e},\bar{a}f}Y_{\bar{a}\bar{e},cf} + X_{\bar{c}e,a\bar{f}}Y_{ae,\bar{c}\bar{f}} + X_{\bar{c}e,\bar{a}f}Y_{\bar{a}e,\bar{c}f} + X_{\bar{c}\bar{e},\bar{a}\bar{f}}Y_{\bar{a}\bar{e},\bar{c}\bar{f}}\right)$$

$$X_{ce,af} = t_{ij}^{ce} t_{ij}^{af} \qquad Y_{ae,cf} = \langle ab\|ek\rangle\langle cb\|fk\rangle$$

Work in Progress and Planned

- Parallel code generation
  - Data distribution interacts w/ memory minimization and are being combined
  - Multi-level parallelism
- More sophisticated performance models
- Common sub-expression elimination
  - Greatly increases complexity of operation min.
- Chemistry-specific optimizations
- Develop approximate algorithms for opt.
  - Address situations where exhaustive search too expensive
  - Deliver best result spending at most 15 min on code gen.
  - Deliver best result spending at most 3 days on code gen.

Summary

- Automatic generation of code from high-level algebraic expressions
  - Approach problem like a compiler
  - Use of HLL allows automation of design decisions usually made by human software developer
- Addresses productivity, complexity, and performance
- Strong interdisciplinary collaboration between chemists and computer scientists
  - Problem from chemists, solutions from computer scientists (w/ significant help from chemists)

For more information

http://www.cis.ohio-state.edu/~gb/TCE

Email: *bernholdtde@ornl.gov*