Synthesizing Highly Optimized Code for Correlated Electronic Structure Calculations

Oak Ridge National Laboratory
<u>David E. Bernholdt</u>, Venkatesh Choppella, ***Robert Harrison***

Pacific Northwest National Laboratory
So Hirata

Louisiana State University
J Ramanujam

Ohio State University
Gerald Baumgartner, Alina Bibireata, Daniel Cociorva, Xiaoyang Gao, Sriram Krishnamoorthy, Sandhya Krishnan, Chi-Chung Lam, Quingda Lu, ***Russell M. Pitzer, P Sadayappan***, Alexander Sibiryakov

University of Waterloo
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, Waterloo, 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

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

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

- 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

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

The “Tensor Contraction Engine” Addresses Programming Challenges

- User describes computational problem (tensor contractions, a la many-body methods) in a simple, high-level language
  - Similar to what might be written in papers
- Compiler-like tools translate high-level language into traditional Fortran (or C, or…) code
- Generated code is compiled and linked to libraries providing computational infrastructure

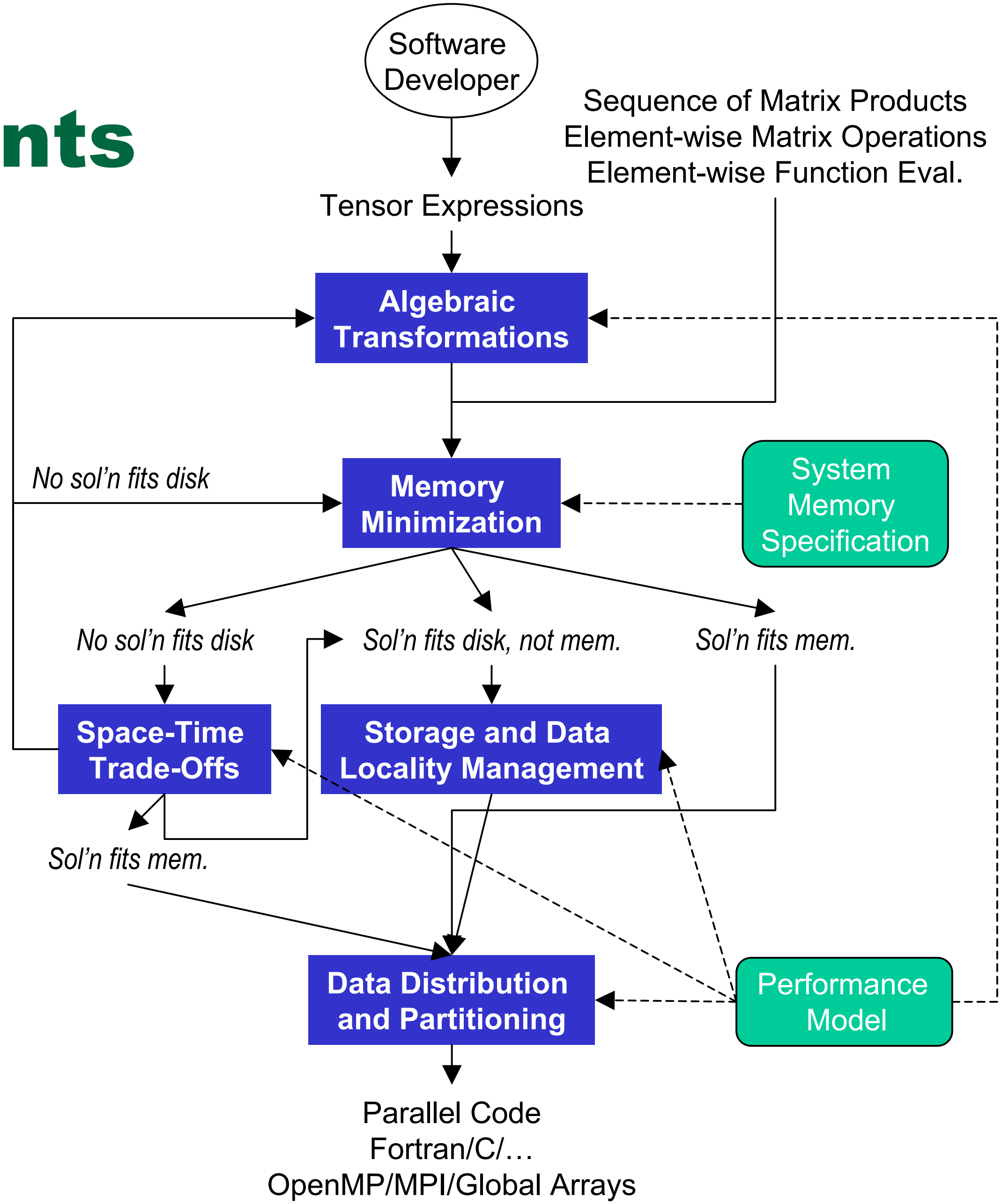
- **Productivity**
  - User writes simple, high-level code
  - Code generation tools do the tedious work
- **Complexity**
  - Significantly reduces complexity visible to programmer
- **Performance**
  - Perform optimizations prior to code generation
  - Automate many decisions humans make empirically
  - Tailor generated code to target computer
  - Tailor generated code to specific problem

So What's New About This Project?

- The creation of "little languages" and code generation tools has a long history in chemistry and other domains
- **Usually viewed only as productivity tools**
  – Imitate what researcher would do – but quicker
- **We treat it as a computer science problem**
  – Similar to (not identical to) an optimizing compiler
  – Algorithmic choices are explored and evaluated rigorously and (in most cases) exhaustively
  – Make use of machine architecture & performance models to specialize generated code to target system
- **Target applications**
  – Rapid experimentation with new many-body methods
  – Implementation of high-complexity methods
  – Improving computational efficiency on parallel machines
  – Also for nuclear physics…

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



A High-Level Language for Tensor Contraction Expressions

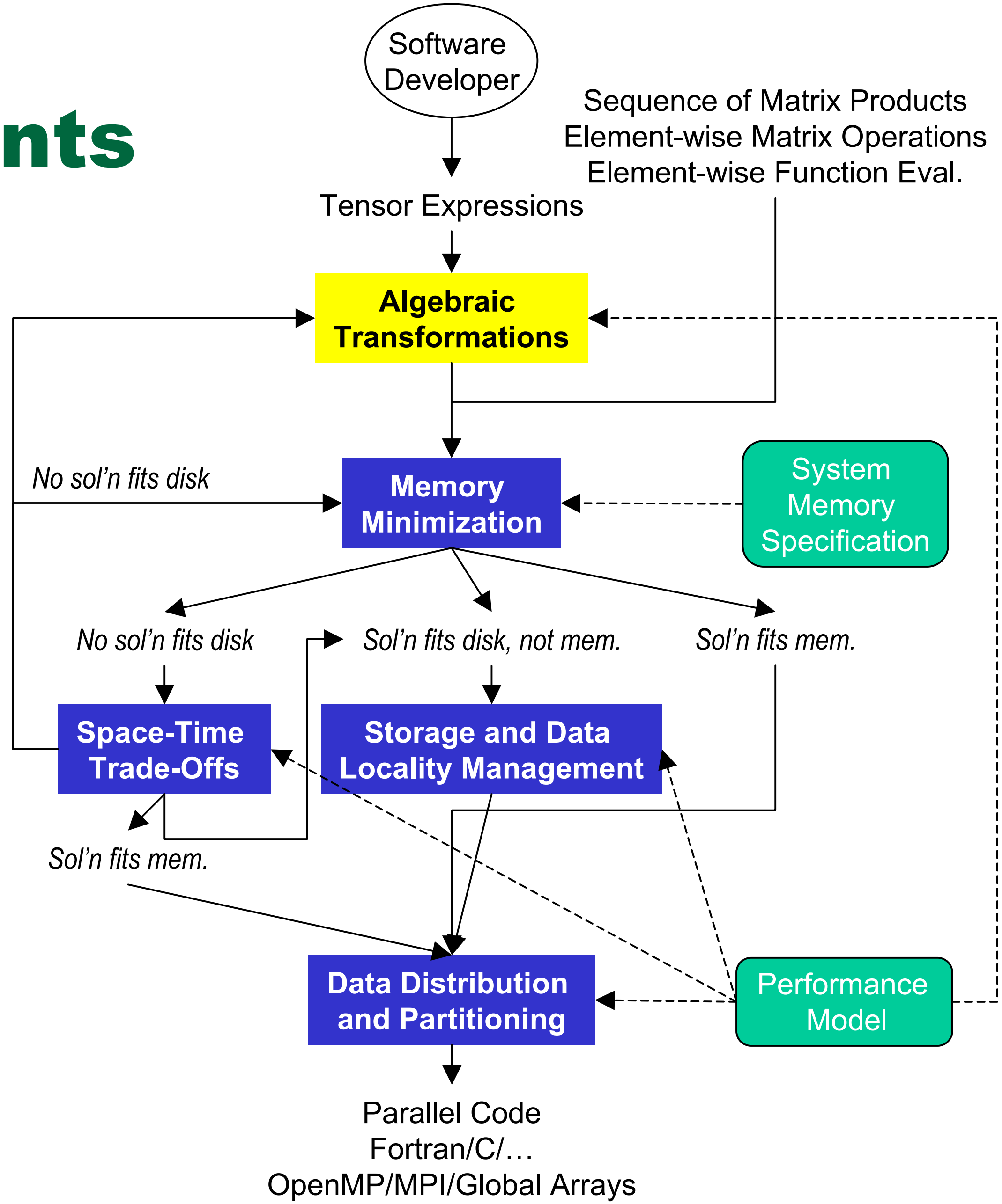
```
range V = 3000;
range O = 100;

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

mlimit = 10000000000000;

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
```

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

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

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

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

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

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

Memory Minimization: Loop Fusion

$$T1_{bcdf} = \sum_{e,l} B_{befl} D_{cdel}$$

$$T2_{bcjk} = \sum_{d,f} T1_{bcdf} C_{dfjk}$$

$$S_{abij} = \sum_{c,k} T2_{bcjk} A_{acik}$$

Formula sequence

```
T1 = 0; T2 = 0; S = 0
for b, c, d, e, f, l
  [ T1bcdf += Befl Dcdel
for b, c, d, f, j, k
  [ T2bcjk += T1bcdf Cdfjk
for a, b, c, i, j, k
  [ Sabij += T2bcjk Aacik
```

Unfused code

```
S = 0
for b, c
  [ T1f = 0; T2f = 0
    for d, f
      [ for e, l
          [ T1f += Befl Dcdel
        for j, k
          [ T2fjk += T1f Cdfjk
    for a, i, j, k
      [ Sabij += T2fjk Aacik
```

Fused code

Operation Minimal Form

```
for a, e, c, f
  for i, j
    X_aecf += 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 += T1_cebk T2_afbk
for c, e, a, f
  E += X_aecf Y_ceaf
```

Inputs: T_{ijae}, T_{ijcf}

External function calls: f1, f2

Output: E

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

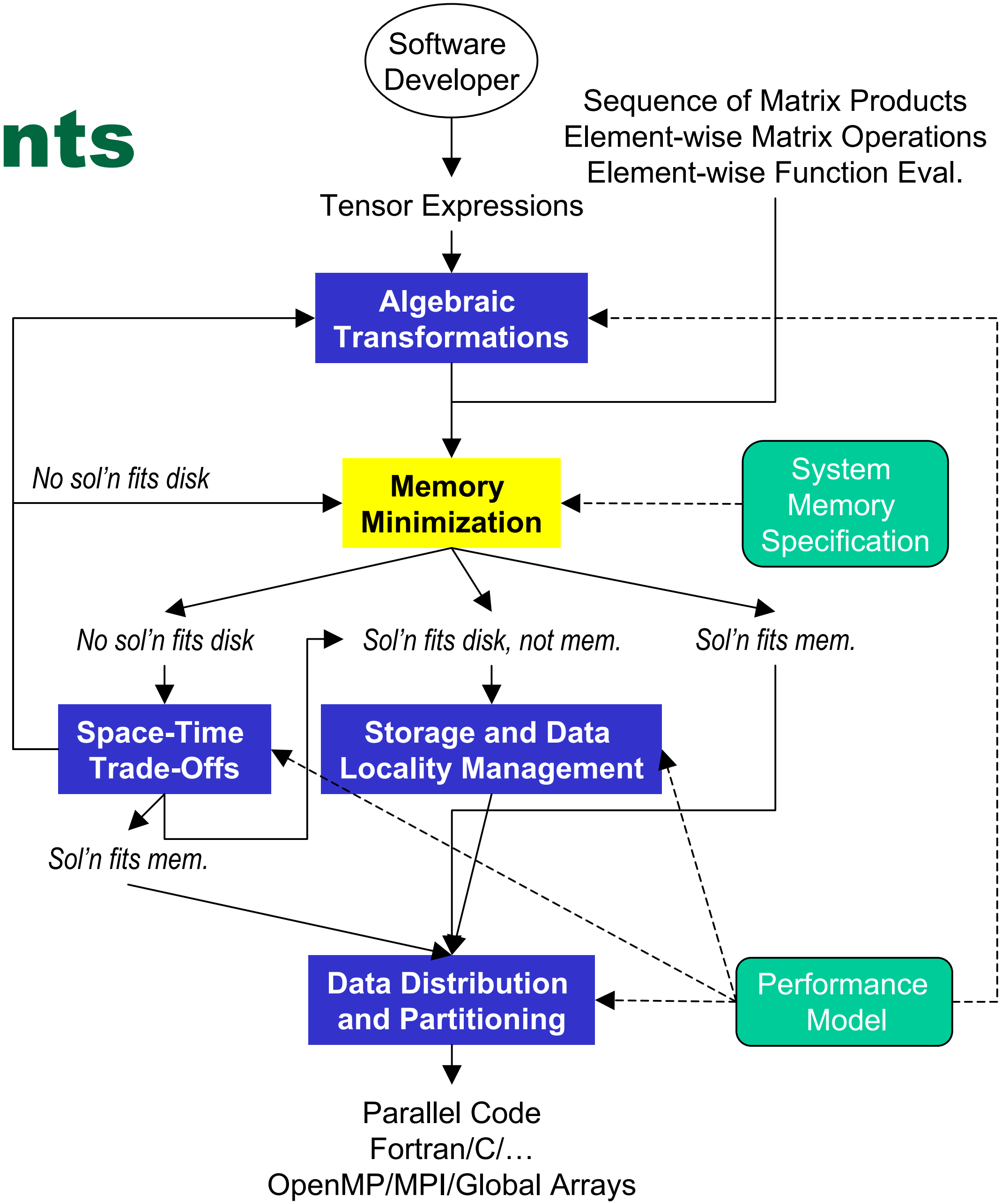
i .. k: range O = 30 .. 100

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

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

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



Operation Minimal Form

```
for a, e, c, f
    for i, j
        X_aecf += 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 += T1_cebk T2_afbk
for c, e, a, f
    E += X_aecf Y_ceaf
```

Inputs (arrows point to T_{ijae} and T_{ijcf})

External function calls (arrows point to f1(c, e, b, k) and f2(a, f, b, k))

Output (arrow points to E)

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

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

$$\left. + X_{\bar{c}e,a\bar{f}}Y_{ae,\bar{c}\bar{f}} + X_{\bar{c}e,\bar{a}f}Y_{\bar{a}e,\bar{c}f} + X_{\bar{c}\bar{e},\bar{a}\bar{f}}Y_{\bar{a}\bar{e},\bar{c}\bar{f}}\right)$$

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

Memory-Minimal Form

Fusion of loops allows reduction of rank of arrays

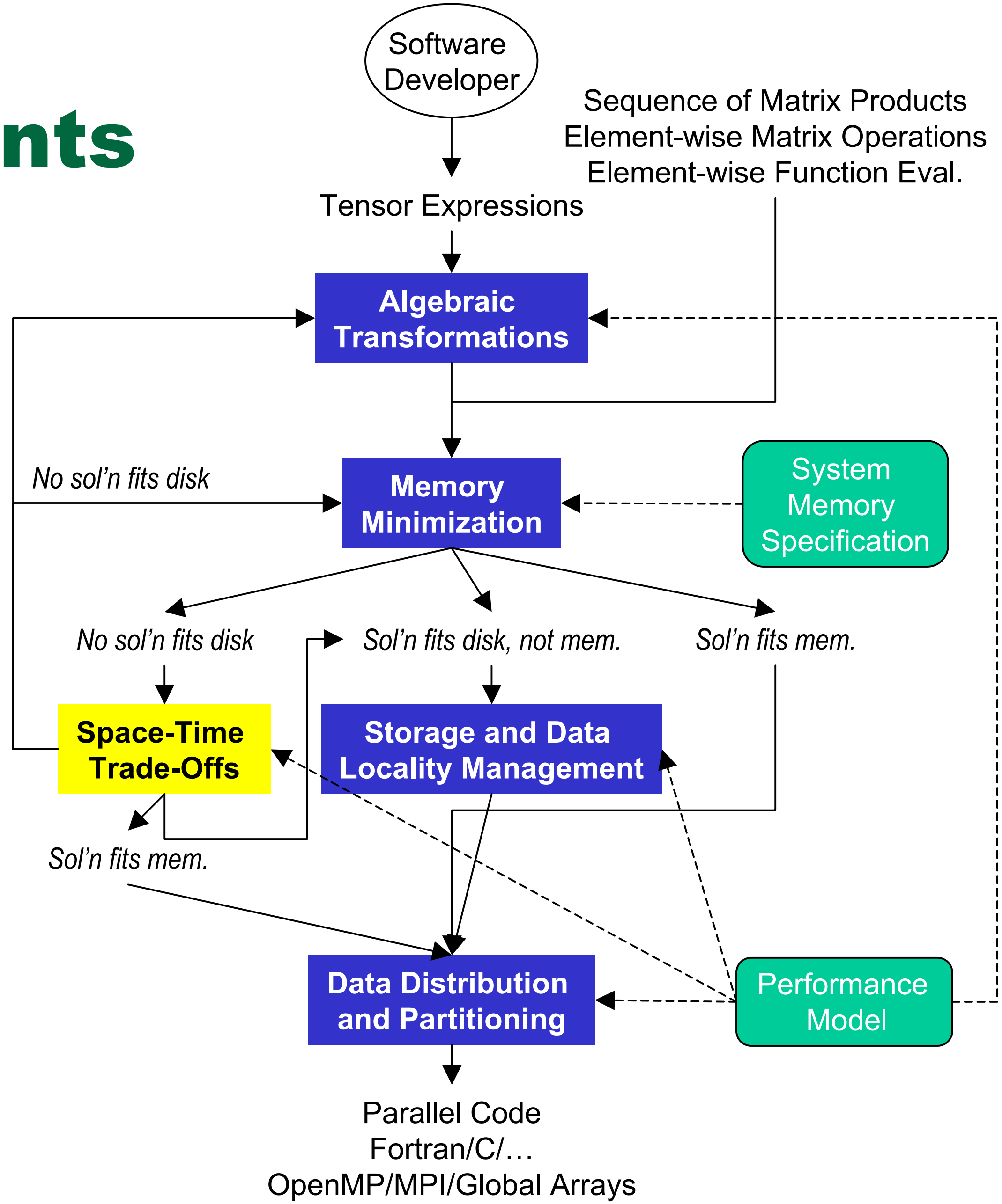
```
for a, f, b, k
  [ T2_afbk = f2(a, f, b, k)
for c, e
  [ for b, k
      [ T1_bk = f1(c, e, b, k)
    for a, f
      [ for i, j
          [ X += T_ijae T_ijcf
        for b, k
          [ Y += T1_bk T2_afbk
        E += X Y
```

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

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



Memory-Minimal Form

Fusion of loops allows reduction of rank of arrays

```
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
```

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
  - $X \mathrel{+}= T_{ijae}\, T_{ijcf}$
- for b, k
  - $T1 = f1(c, e, b, k)$
  - $T2 = f2(a, f, b, k)$
  - $Y \mathrel{+}= T1\ T2$
- $E \mathrel{+}= X\ Y$

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

Full fusion comes at the cost of increased computation

Tiling to Reduce Recomputation

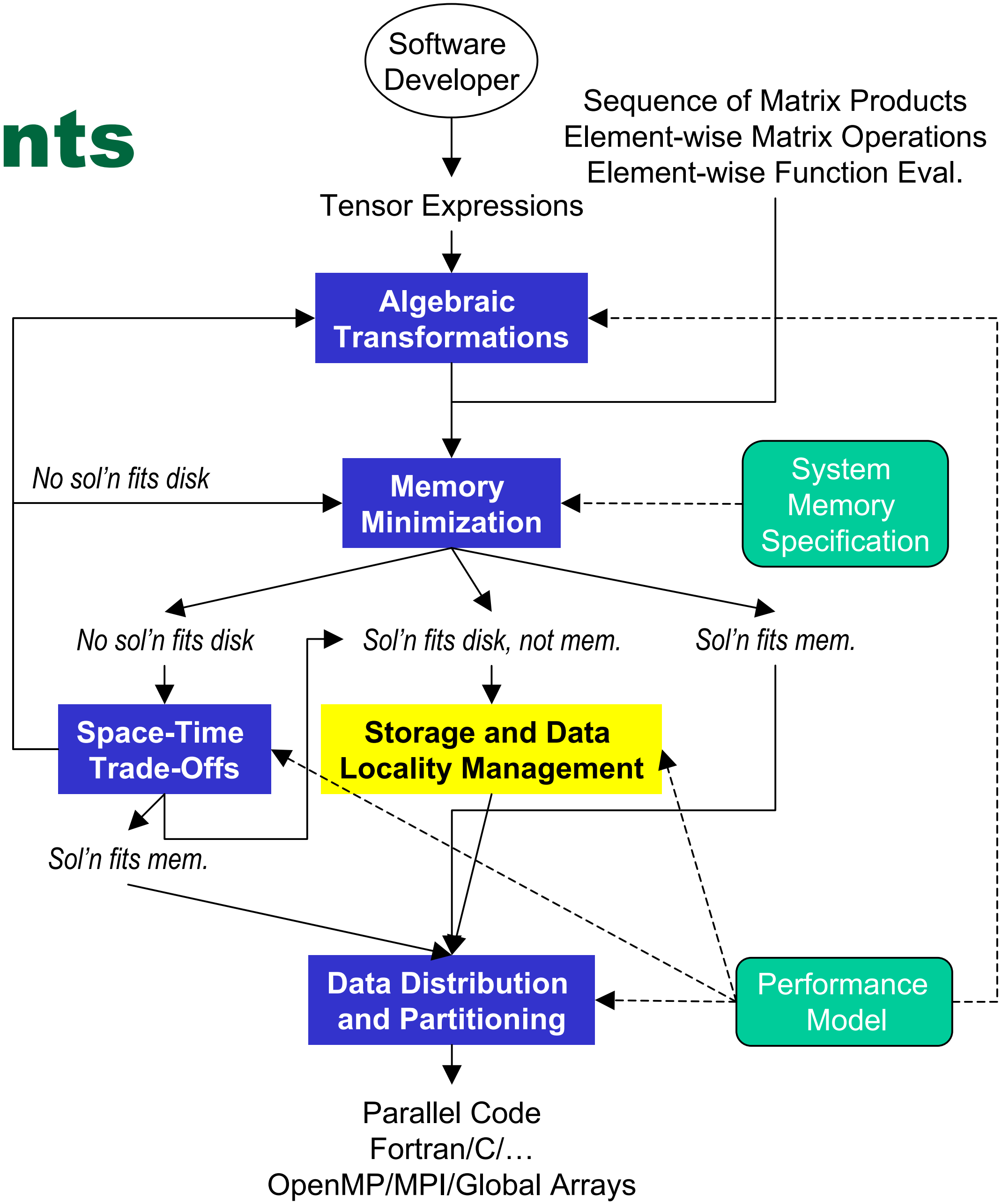
for a^t, e^t, c^t, f^t
- for a, e, c, f
  - for i, j
    - $X_{aecf} \mathrel{+}= T_{ijae} T_{ijcf}$
- for b, k
  - for c, e
    - $T1_{ce} = f1(c, e, b, k)$
  - for a, f
    - $T2_{af} = f2(a, f, b, k)$
  - for c, e, a, f
    - $Y_{ceaf} \mathrel{+}= T1_{ce} T2_{af}$
- for c, e, a, f
  - $E \mathrel{+}=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

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

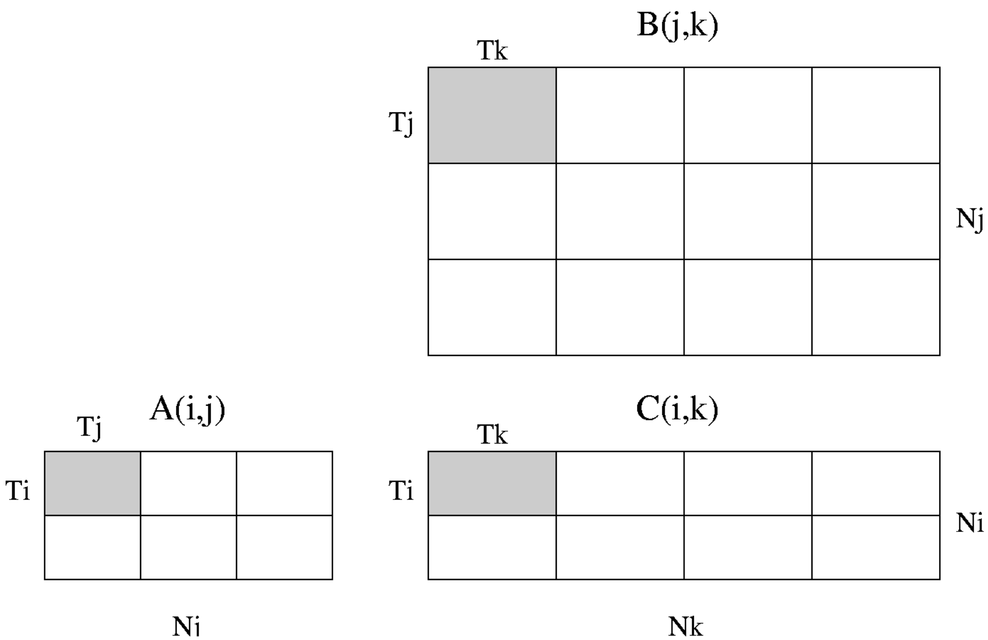
Tiling provides a ***controlled*** compromise between minimal operations and minimal memory (full fusion)

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



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

Current TCE Capabilities

Capability	Prototype TCE	Optimizing TCE
Basic sequential code generation for CC-based methods	Yes	Yes
QC Packages Interfaced:		
• File based	NWChem, UTChem	NWChem
• General (file, memory, direct)		Under development
Symmetry Support:		
• Spin	Spin orbitals	General, in progress
• Spatial	Abelian	Abelian, in progress
• Permutational	Fermions	General, in progress
Optimizations:		
• Operation Minimization	Partial	Yes
• Memory Minimization	Partial	Yes
• Space-Time Transformation	No	Yes
• Data Locality	Partial	Yes
Parallel code generation	Limited general	General, in progress

Methods Implemented to Date using TCE

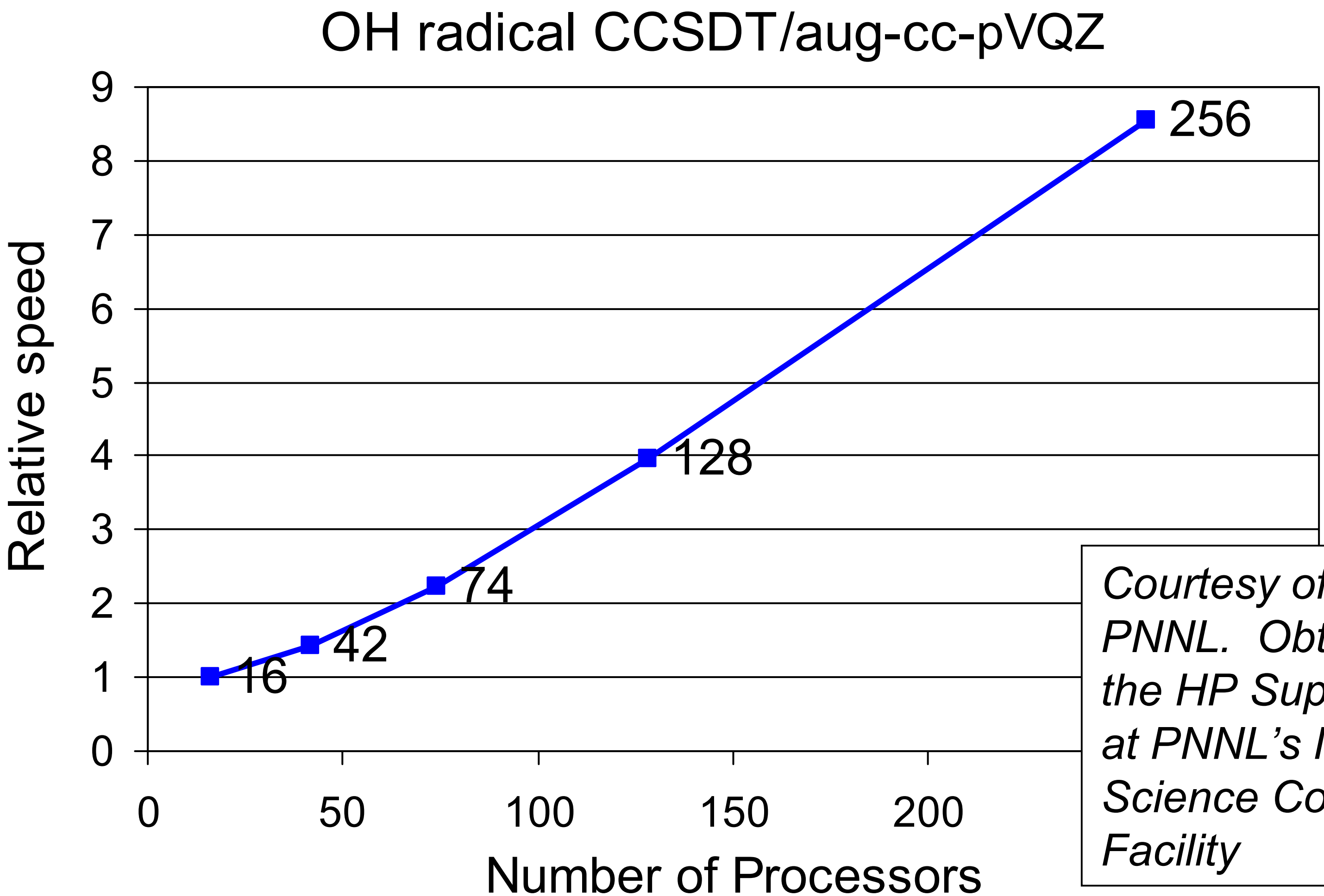
Prototype TCE

- Parallel
  - MBPT(2), MBPT(3), MBPT(4)
  - CCD, CCSD, CCSD(T), QCISD, CCSDT, CCSDTQ
  - CISD, CISDT, CISDTQ
  - CCSD, CCSDT, CCSDTQ lambda equations
  - CCSD, CCSDT, CCSDTQ dipole moments
  - Parallel EOM-CCSD
  - Local/AO-based CCSD
- Sequential
  - Relativistic 2/4-component CI, CC, & MBPT

Optimizing TCE

- *Parallel (limited)*
  - *CCD, CCSD*
- Code availability
  - Prototype TCE to be released with NWChem 4.5
  - Optimizing TCE still under intense development
  - Some prototype-generated implementations to appear in …
    - NWChem 4.5
    - UTChem 2003

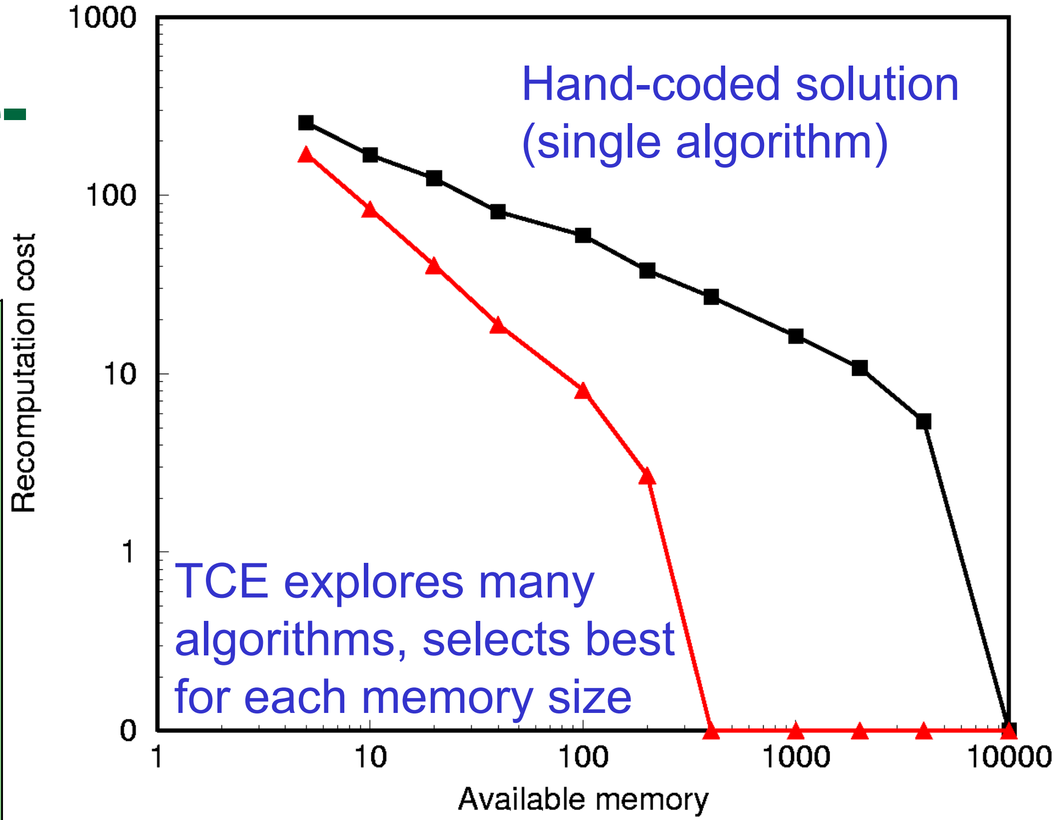
Parallel Scalability of Prototype TCE-Generated Code



Courtesy of So Hirata, PNNL. Obtained on the HP Supercluster at PNNL's Molecular Science Computing Facility

The Value of Optimization (Space-Time Trade-Offs)

```
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
```



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

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

On the Drawing Board…

- More flexibility in sequencing and controlling optimizations
- Common sub-expression elimination
- Global factorization (across equations)
  - Complex problem
- Improving parallel code generation
  - Multi-level parallelism
    - Threads
    - Multiple loosely coupled tasks
- More sophisticated performance models
- Develop approximate algorithms for opt.
  - Address situations where exhaustive search too expensive
    - i.e. Deliver best result spending at most 3 min on code gen.
    - … or 60 min … or 3 days …
- Generalizations beyond electronic structure

Ways to Use the TCE

Mode	Prototype TCE	Optimizing TCE
Explore/develop new methods rapidly	Now	Soon
Develop parallel implementations	Now	Soon
Develop large-scale high-performance parallel implementations	Under consideration	Under development

Summary

- Automatic generation of code from high-level algebraic expressions
  - Approach problem like a compiler
  - Use of "high-level language" allows automation of design decisions usually made by human software developer
  - Produce robust, reliable code
- Addresses productivity, complexity, and performance
  - Compiler-like optimizations key to full utility of code generation approaches
- Strong interdisciplinary collaboration between chemists and computer scientists
  - Problem from chemists, solutions from computer scientists (w/ significant help from chemists)

For More Information…

- At the ACS meeting:

Monday	8-10pm	Poster PHYS 224 (Sci-Mix)	North Pavillion
Tuesday	2:10-2:50pm	Nooijen talk PHYS 144	1E08
Tuesday	6-8pm	Poster COMP 105	North Pavillion
Wednesday	7:30-10pm	Poster PHYS 224	North Pavillion
Thursday	9:10-9:50am	Hirata talk PHYS 438	1E08

- Workshop being planned for Sanibel Symposium, 28 February – 6 March 2004 (workshop probably Wednesday 3 March)

- Web: http://www.cis.ohio-state.edu/~gb/TCE
- Email: bernholdtde@ornl.gov