The Build to Order Compiler for Matrix Algebra Optimization

Liz Jessup
University of Colorado at Boulder

Ian Karlin, Geoff Belter, Jeremy Siek, Erik Silkensen, Pavel Zelinsky, Tom Nelson
Library use increases memory access cost

\[ q = Ap \]
\[ s = A^T r \]

DGEMV('n', m, n, alpha, a, lda, p, l, beta, q, l);

DGEMV('y', m, n, alpha, a, lda, r, l, beta, s, l);

for (i=0;i<n;++i)
  for (j=0;j<m;++j) {
    q[j] += p[i]*A[i][j];
  }

for (i=0;i<n;++i)
  for (j=0;j<m;++j) {
    s[i] += r[j]*A[i][j];
  }
Subset of MATLAB

\[
\begin{align*}
q &= A^*p \\
s &= A'^*r
\end{align*}
\]

BTO

Generate Loops

Enumerate Fusion Opportunities

Refine

Generate C Code

\[
\begin{align*}
&\text{for } (i=0; i<N; ++i) \\
&\quad \text{for } (j=0; j<M; ++j) \\
&\qquad q[j] \ += p[i]*A[i][j]; \\
&\qquad s[i] \ += r[j]*A[i][j];
\end{align*}
\]
Hybrid Refinement

V1  V2  V3

Analytic Model

Empirical Testing

analytic model predicts runtimes for enumerated versions

only best versions are tested

Analytic model based on reuse distances

Code Generation

Friday, January 7, 2011
Powerful optimization possible

But what is the right amount of fusion?
An experiment: \( n\text{vecs} \times \text{DGEMV} \)

**fusion options for \( n\text{vecs} = 2 \)**

**no fusion**

```plaintext
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        v0[i] += A[i][j] * u0[j]

for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        v1[i] += A[i][j] * u1[j]
```

**outer loops fused**

```plaintext
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        v0[i] += A[i][j] * u0[j]

for (j = 0; j < n; j++)
    v1[i] += A[i][j] * u1[j]
```

**all loops fused**

```plaintext
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        v0[i] += A[i][j] * u0[j]

for (j = 0; j < n; j++)
    v1[i] += A[i][j] * u1[j]
```
nvecs = 4

nvecs = 5

nvecs = 6

Inner Loops: register spill

Friday, January 7, 2011
All Outer Loops Fused

Matrix order

Mflops

nvecs = 4
nvecs = 8
nvecs = 12

Outer Loops: cache misses

Outer Loops:
cache misses

nvecs = 8

L2 misses per flop

0.05
0.04
0.03
0.02
0.01
0.00

0.00

0 2000 4000 6000 8000 10000 12000 14000

matrix order

0 2000 4000 6000 8000 10000 12000 14000

matrix order

0 2000 4000 6000 8000 10000 12000 14000

matrix order

0 2000 4000 6000 8000 10000 12000 14000