INTEGRATING DATA DISTRIBUTION AND LOOP TRANSFORMATIONS

J. Ramanujam
A. Narayan
Louisiana State University

SIAM Conference on Parallel Processing for Scientific Computing
San Francisco, CA
February 15, 1995

Overview

Key optimizations:

- move communication out of loops
- use fewer but larger messages (block transfers: *message vectorization*)

integrating data mapping and program transformations

Features:

- dense and band matrix codes
- only one dimension of an array is distributed
- find combination of data mapping and loop transformation
- use *data reference matrices*

Example

do $i = 1, N_1$
  do $j = 1, N_2$
    do $k = 1, N_3$
      $B(i, j - i) = B(i, j - i) + A(i, j + k)$
    enddo
  enddo
enddo

Data Reference matrix of A, $A_{\mathcal{R}}$: $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} i \\ j + k \end{pmatrix}$

Data Reference matrix of B, $B_{\mathcal{R}}$: $\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} i \\ j - i \end{pmatrix}$

Example – continued

A row is common to all data reference matrices
$\Longrightarrow$ communication-free data mapping is possible

- Distribute the matrices A and B by rows (the second array dimension is not distributed)
- No transformation required here
- Partition the outer loop iterations among processors

$\boxed{\textbf{Strategy}}$

On applying an invertible transformation T, the transformed data reference matrix,

$$A'_{\mathcal{R}} = A_{\mathcal{R}} T^{-1}$$

Source loop nest : $(i_1, i_2, \ldots, i_n)$ Transformed loop nest : $(j_1, j_2, \ldots, j_n)$

$$T : \begin{pmatrix} i_1 \\ \vdots \\ i_n \end{pmatrix} \longrightarrow \begin{pmatrix} j_1 \\ \vdots \\ j_n \end{pmatrix}$$

Columns of T^{-1} determine subscripts of array references in transformed loop nest

$\boxed{\text{Use reference matrices to construct the best } T^{-1}}$

Strategy – LHS reference

Goal:

– Distribute the outermost iterations of the transformed loop nest (and)

– Distribute a corresponding dimension of the LHS array

On applying T, the transformed data reference matrix of an LHS array should have the following structure:

$\boxed{\begin{array}{l} j_1 \text{ must be the only index in some dimension, } r \\ \Longrightarrow \text{Row } r \text{ of } A_{\mathcal{R}}T^{-1} \text{ is } (\alpha, 0, \ldots, 0) \end{array}}$

The LHS array can be distributed only along dimension r
The remaining dimensions of LHS array are not distributed.

The outermost loop iterations are distributed among processors
$\Longrightarrow$ all references to LHS arrays are local

Strategy – Block Transfer for RHS References

Goal: To move communication arising from an RHS reference outside the innermost loop and use block transfers

– j_n must be the only index in some dimension; (and)

– j_n must not appear in any other dimensions

$$\Longrightarrow B_{\mathcal{R}}T^{-1} \text{ is of the form } \begin{pmatrix} \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta \\ \vdots & \vdots & \vdots & 0 \\ \vdots & \ddots & \vdots & 0 \end{pmatrix}$$

Dimension r of array B is not distributed

Communication due to B is moved out of the innermost loop

Use this approach moving outward from the innermost loop

Matrix Multiplication

do $i = 1, N$
  do $j = 1, N$
    do $k = 1, N$
      $C(i,j) \mathrel{+}= A(i,k) * B(k,j)$
    enddo
  enddo
enddo

$$C_{\mathcal{R}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \; A_{\mathcal{R}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \; B_{\mathcal{R}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \; T^{-1} = Q$$

Step 1: Distribute first dimension of C   Set

$$\begin{aligned} C_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,1}) &= \quad 1 \quad \Longrightarrow Q_{11} = 1 \\ C_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,2}) &= \quad 0 \quad \Longrightarrow Q_{12} = 0 \\ C_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,3}) &= \quad 0 \quad \Longrightarrow Q_{13} = 0 \end{aligned}$$

Matrix Multiplication – continued

Step 2: <u>Derive distribution of A</u>
$C_{\mathcal{R}}(1,:) = A_{\mathcal{R}}(1,:) \Longrightarrow$ Distribute A and C identically.

Step 3: <u>Derive distribution of B</u> Set

$$\boxed{\begin{array}{c} B_{\mathcal{R}}(\vec{1},:) \cdot Q(\vec{:},1) = 0 \Longrightarrow Q_{31} = 0 \\ B_{\mathcal{R}}(\vec{1},:) \cdot Q(\vec{:},2) = 0 \Longrightarrow Q_{32} = 0 \\ B_{\mathcal{R}}(\vec{1},:) \cdot Q(\vec{:},3) = 1 \Longrightarrow Q_{33} = 1 \\ B_{\mathcal{R}}(\vec{2},:) \cdot Q(\vec{:},3) = 0 \Longrightarrow Q_{23} = 0 \end{array}} \qquad B_{\mathcal{R}}Q = \begin{pmatrix} 0 & 0 & 1 \\ ? & ? & 0 \end{pmatrix}$$

The first dimension of B is not distributed.

Finally we have, $Q = T^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ Q_{21} & Q_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

Matrix Multiplication – Final code

Choose Q as identity matrix. No transformation needed

Distribute A and C by rows, and B by columns.

```
do i = 1 to N
    do j = 1 to N
        receive B(:, j)
        do k = 1 to N
            C(i, j) = C(i, j) + A(i, k) * B(k, j)
        enddo
    enddo
enddo
```

Iterations of i loop distributed among processors!

SYR2K code

do $i = 1, N$
  do $j = i, \min(i + 2b - 2, N)$
    do $k = \max(i - b + 1, j - b + 1, 1), \min(i + b - 1, j + b - 1, N)$
      $C[i, j - i + 1] \mathrel{+}= A[k, i - k + b] * B[k, j - k + b]$
        $+ A[k, j - k + b] * B[k, i - k + b]$
    enddo
  enddo
enddo

$$C_{\mathcal{R}} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \end{pmatrix}$$

$$A^1_{\mathcal{R}} = B^2_{\mathcal{R}} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}$$

$$A^2_{\mathcal{R}} = B^1_{\mathcal{R}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

SYR2K – LHS row distribution

Step 1: Distribute first dimension of C   Set

$$
\begin{aligned}
C_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,1}) = & \quad 1 \quad \Longrightarrow Q_{11} = 1 \\
C_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,2}) = & \quad 0 \quad \Longrightarrow Q_{12} = 0 \\
C_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,3}) = & \quad 0 \quad \Longrightarrow Q_{13} = 0
\end{aligned}
$$

Step 2: Derive distribution of A

$$
\boxed{
\begin{array}{c}
A^1_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,1}) = 0 \Longrightarrow Q_{31} = 0 \\
A^1_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,2}) = 0 \Longrightarrow Q_{32} = 0 \\
A^1_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,3}) = 1 \Longrightarrow Q_{33} = 1 \\
A^1_{\mathcal{R}}(\vec{2,:}) \cdot Q(\vec{:,3}) = 0 \Longrightarrow Q_{13} - Q_{33} = 0
\end{array}
}
\qquad
A^1_{\mathcal{R}} Q = \begin{pmatrix} 0 & 0 & 1 \\ ? & ? & 0 \end{pmatrix}
$$

$Q_{13} - Q_{33} = 0$ impossible since $Q_{13} = 0$ and $Q_{33} = 1$.

SYR2K – LHS row distribution (2)

$$A^1_{\mathcal{R}} Q = \begin{pmatrix} ? & ? & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ is impossible

$$A^2_{\mathcal{R}} Q = \begin{pmatrix} 0 & 0 & 1 \\ ? & ? & 0 \end{pmatrix}$$ is impossible

$$A^2_{\mathcal{R}} Q = \begin{pmatrix} ? & ? & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ is impossible

Step 3: Derive distribution of B Similar analysis

There is no way to distribute arrays A and B (when the first dimension of C is distributed) to allow block transfers outside innermost loop

SYR2K – LHS column distribution

Step 1: Distribute second dimension of C   Set

$$
\begin{aligned}
C_{\mathcal{R}}(\vec{2,:}) \cdot Q(\vec{:,1}) &= \quad 1 & \Longrightarrow Q_{11} = Q_{21} - 1 \\
C_{\mathcal{R}}(\vec{2,:}) \cdot Q(\vec{:,2}) &= \quad 0 & \Longrightarrow Q_{12} = Q_{22} \\
C_{\mathcal{R}}(\vec{2,:}) \cdot Q(\vec{:,3}) &= \quad 0 & \Longrightarrow Q_{13} = Q_{23}
\end{aligned}
$$

Step 2: Derive distribution of A First reference to A

$$
\boxed{
\begin{array}{c}
A^1_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,1}) = 0 \Longrightarrow Q_{31} = 0 \\
A^1_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,2}) = 0 \Longrightarrow Q_{32} = 0 \\
A^1_{\mathcal{R}}(\vec{1,:}) \cdot Q(\vec{:,3}) = 1 \Longrightarrow Q_{33} = 1 \\
A^1_{\mathcal{R}}(\vec{2,:}) \cdot Q(\vec{:,3}) = 0 \Longrightarrow Q_{13} - Q_{33} = 0
\end{array}
}
\qquad
A^1_{\mathcal{R}} Q = \begin{pmatrix} 0 & 0 & 1 \\ ? & ? & 0 \end{pmatrix}
$$

$Q_{13} = 1$ and $Q_{23} = 1$. First dimension of A not distributed

SYR2K – LHS column distribution (2)

Second reference to A:

$$\boxed{\begin{array}{c} \vec{A^2_{\mathcal{R}}(1,:)} \cdot \vec{Q(:,1)} = 0 \Longrightarrow Q_{31} = 0 \\ \vec{A^2_{\mathcal{R}}(1,:)} \cdot \vec{Q(:,2)} = 0 \Longrightarrow Q_{32} = 0 \\ \vec{A^2_{\mathcal{R}}(1,:)} \cdot \vec{Q(:,3)} = 1 \Longrightarrow Q_{33} = 1 \\ \vec{A^2_{\mathcal{R}}(2,:)} \cdot \vec{Q(:,3)} = 0 \Longrightarrow Q_{23} - Q_{33} = 0 \end{array}}$$

$$A^2_{\mathcal{R}} Q = \begin{pmatrix} 0 & 0 & 1 \\ ? & ? & 0 \end{pmatrix}$$

First dimension of A not distributed. Similar analysis for array B.

Second dimensions of A and B (and C) are distributed

One possible solution $Q = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$. $T = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix}$.

SYR2K – Final code

do $u = 1, N$
  do $v = u, \min(u + 2b - 2, N)$
    receive $A(:, v + b)$
    receive $A(:, u + v + b)$
    receive $B(:, u + v + b)$
    receive $B(:, v + b)$
    do $w = \max(u - b + 1, v - b + 1, 1), \min(u + b - 1, v + b - 1, N)$
      $C[v + w + 1, u] \mathrel{+}= A[w, v + b] * B[w, u + v + b]$
        $+ A[w, u + v + b] * B[w, v + b]$
    enddo
  enddo
enddo

Iterations of u loop distributed among processors!

Summary

Results

- dense and band matrix codes
- one dimension of an array is distributed
- find best combination of data mapping and loop transformation that
    - moves communication out of loops
    - results in *message vectorization*
- uses *data reference matrices* and inverse of transformation matrix

In progress

- more than one distributed dimension
- reduce complexity of search
- relax owner-compute