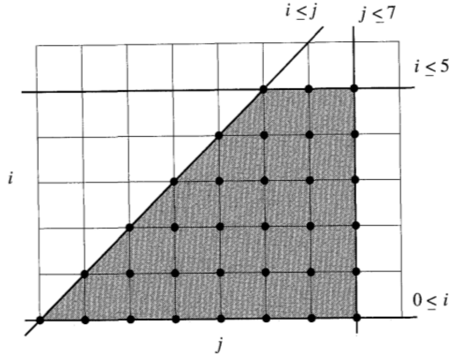
Optimization for Locality and Parallelism

Unnikrishan C

September 24, 2019


$i \le j$
$j \le 7$
$i \le 5$
i
$0 \le i$
j

Access Pattern required Vs Happening

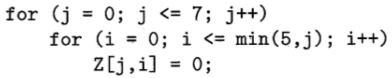
0 <= i <= 5, i <= j <= 7

[0,0] [0,1] [0,2] [0,3] [0,4] [0,5] [0,6] [0,7]

Bounds for Array Access

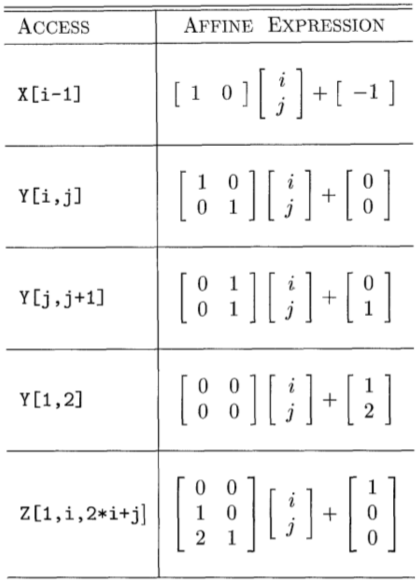
$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 0 \\ 5 \\ 0 \\ 7 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Modified Loop



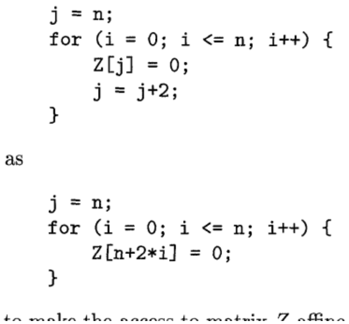
```
for (j = 0; j <= 7; j++)
    for (i = 0; i <= min(5,j); i++)
        Z[j,i] = 0;
```

Affine Accesses



ACCESS	AFFINE EXPRESSION
`X[i-1]`	$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix}$
`Y[i,j]`	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
`Y[j,j+1]`	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
`Y[1,2]`	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
`Z[1,i,2*i+j]`	$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Conversion to Affine Access - An Example



```
    j = n;
    for (i = 0; i <= n; i++) {
        Z[j] = 0;
        j = j+2;
    }
```

as

```
    j = n;
    for (i = 0; i <= n; i++) {
        Z[n+2*i] = 0;
    }
```

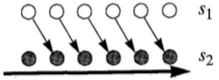
to make the access to matrix Z affine

Reuse- Spatial, temporal, group

```
float Z[n];
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        Z[j+1] = (Z[j] + Z[j+1] + Z[j+2])/3;
```

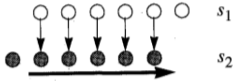
Affine Transformations - Reindexing

```
for (i=1; i<=N; i++) {
  Y[i] = Z[i];    /*s1*/
  X[i] = Y[i-1]; /*s2*/
}
```



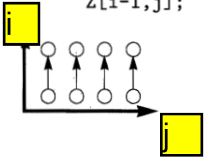
Re-indexing
$s_1 : p = i$
$s_2 : p = i - 1$

```
if (N>=1) X[1]=Y[0];
for (p=1; p<=N-1; p++){
  Y[p]=Z[p];
  X[p+1]=Y[p];
}
if (N>=1) Y[N]=Z[N];
```



Affine Transformations - Permutation

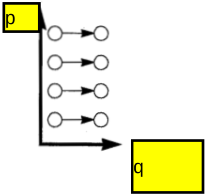
```
for (i=1; i<=N; i++)
  for (j=0; j<=M; j++)
    Z[i,j] =
          Z[i-1,j];
```



Permutation

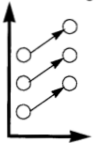
$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix}$$

```
for (p=0; p<=M; p++)
  for (q=1; q<=N; i++)
     Z[q,p] = Z[q-1,p]
```



Affine Transformations - Skewing

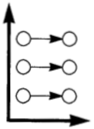
```
for (i=1; i<=N+M-1; i++)
  for (j=max(1,i+N);
       j<=min(i,M); j++)
     Z[i,j] =
       Z[i-1,j-1];
```



Skewing

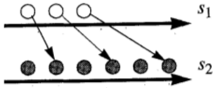
$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

```
for (p=1; p<=N; p++)
  for (q=1; q<=M; q++)
     Z[p,q-p] =
        Z[p-1,q-p-1]
```



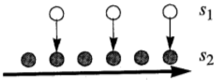
Affine Transformations - Scaling

```
for (i=1; i<=N; i++)
  Y[2*i] = Z[2*i]; /*s1*/
for (j=1; j<=2N; j++)
  X[j]=Y[j];        /*s2*/
```



Scaling
$s_1 : p = 2 * i$
$(s_2 : p = j)$

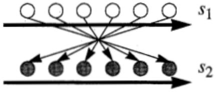
```
for (p=1; p<=2*N; p++){
  if (p mod 2 == 0)
    Y[p] = Z[p];
  X[p] = Y[p];
}
```



Affine Transformations - Reversal

<

```
for (i=0; i>=N; i++)
  Y[N-i] = Z[i];   /*s1*/
for (j=0; j<=N; j++)
  X[j] = Y[j];    /*s2*/
```



Reversal
$s_1 : p = N - i$
$(s_2 : p = j)$

```
for (p=0; p<=N; p++){
  Y[p] = Z[N-p];
  X[p] = Y[p];
}
```