

---

# High Performance Computing

## Lecture 30

Matthew Jacob

Indian Institute of Science

---

# Example 2 with Array Merging

What if we re-declare the arrays as

```
struct {double A, B;} array[2048];
```

```
for (i=0; i<2048, i++) sum += array[i].A*array[i].B;
```

Hit ratio: 75%

---

## Example 3: DAXPY

$$\frac{1536 + 1536 + 2048}{6144}$$

- Double precision  $Y = aX + Y$ , where  $X$  and  $Y$  are vectors and  $a$  is a scalar

double  $X[2048]$ ,  $Y[2048]$ ,  $a$ ;

for ( $i=0$ ;  $i<2048$ ;  $i++$ )  $Y[i] = a*X[i]+Y[i]$ ;

- Reference sequence

load  $X[0]$  load  $Y[0]$  store  $Y[0]$

load  $X[1]$  load  $Y[1]$  store  $Y[1]$  ... etc

- Hits and misses: Assuming that base addresses of  $X$  and  $Y$  don't conflict in cache, hit ratio of 83.3%

---

## Example 4: 2-d Matrix Sum

```
double A[1024][1024], B[1024][1024];  
for (j=0;j<1024;j++)  
for (i=0;i<1024;i++)  
    B[i][j] = A[i][j] + B[i][j];
```

- Reference Sequence:

load A[0,0] load B[0,0] store B[0,0]

load A[1,0] load B[1,0] store B[1,0] ...

- Question: In what order are the elements of a multidimensional array stored in memory?

---

# Storage of Multi-dimensional Arrays

## ■ Row major order

- ❑ Example: for a 2-dimensional array, the elements of the first row of the array are followed by those of the 2<sup>nd</sup> row of the array, the 3<sup>rd</sup> row, and so on
- ❑ This is what is used in C

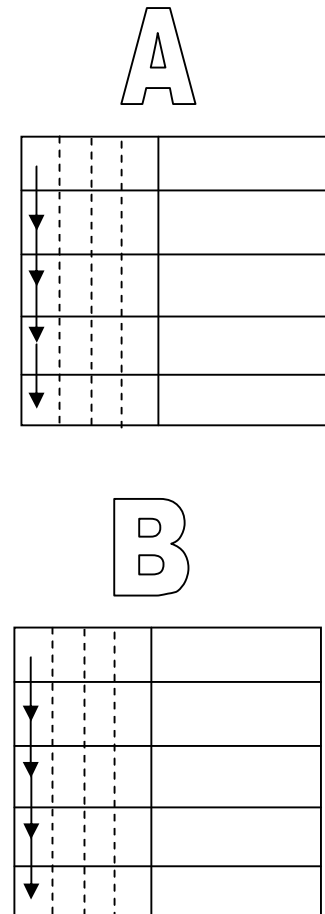
## ■ Column major order

- ❑ A 2-dimensional array is stored column by column in memory
- ❑ Used in FORTRAN

# Example 4: Hits and Misses

load A[0,0] load B[0,0] store B[0,0]  
load A[1,0] load B[1,0] store B[1,0] ...

- Reference order is different from storage order for arrays are not the same
- Our loop will show no spatial locality
  - Assume that packing has been to eliminate conflict misses due to base addresses
  - Miss(cold), Miss(cold), Hit for each array element
  - Hit ratio: 33.3%
  - Question: Will A[0,1] be in the cache when required?



---

# Example 4 with Loop Interchange

```
double A[1024][1024], B[1024][1024];  
for (j=0; j<1024; j++)  
for (i=0; i<1024; i++)  
    B[i][j] = A[i][j] + B[i][j];
```

# Example 4 with Loop Interchange.

```
double A[1024][1024], B[1024][1024];  
for (i=0; i<1024; i++)  
for (j=0; j<1024; j++)  
    B[i][j] = A[i][j] + B[i][j];
```

- Reference Sequence:

```
load A[0,0] load B[0,0] store B[0,0]  
load A[0,1] load B[0,1] store B[0,1] ...
```

- Hit ratio: 83.3%

