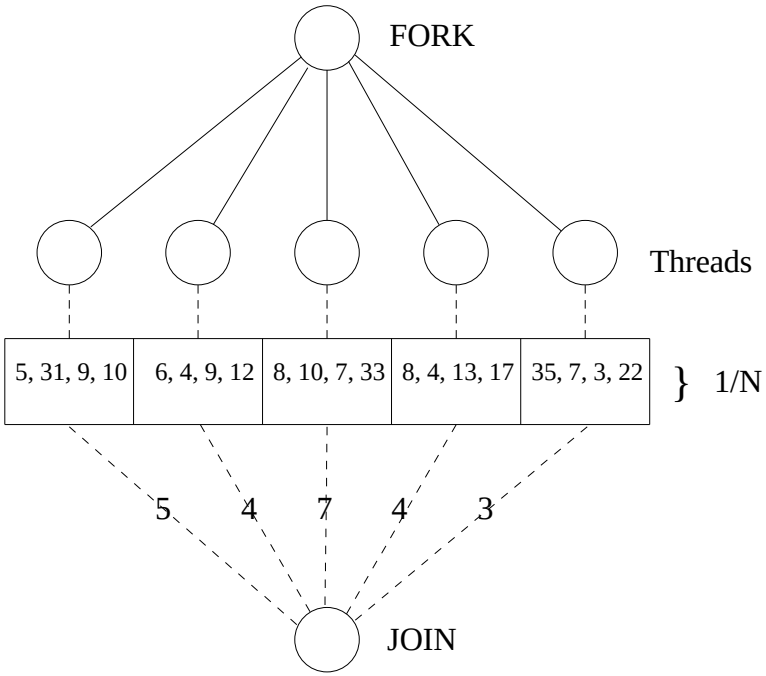
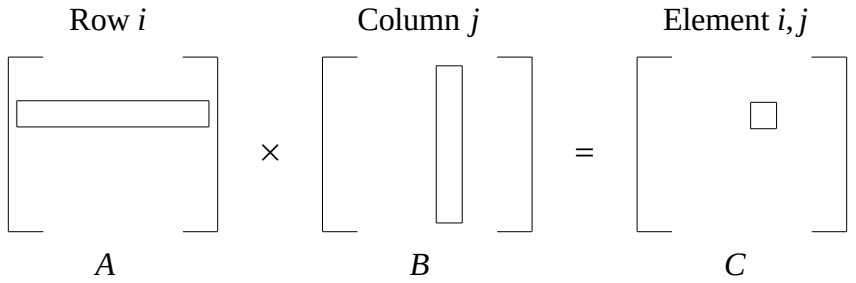
CONCURRENT PROGRAMS

- Try for *N* times faster on *N* processors.
- Decompose the problem into conceptual tasks.



Example of a Concurrent Program

MATRIX MULTIPLICATION

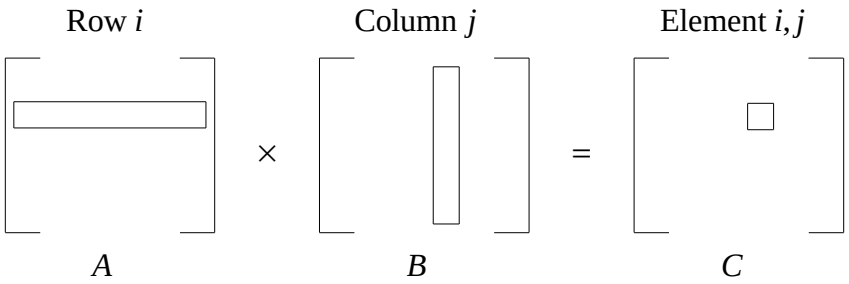


Conceptual Unit of Computation is Element of *C*.

```
FOR i = 1 TO n
    FOR j = 1 to n
        threads[t] = FORK (ComputeElement, i, j)
        t = t + 1
    ENDFOR
ENDFOR
JOIN (threads)
```

```
ComputeElement (i, j)
FOR k = 1 to n
    C[i,j] = C[i,j] + (A[i,k] * B[k,j]);
ENDFOR
```

LIMITING/CONTROLLING CONCURRENCY

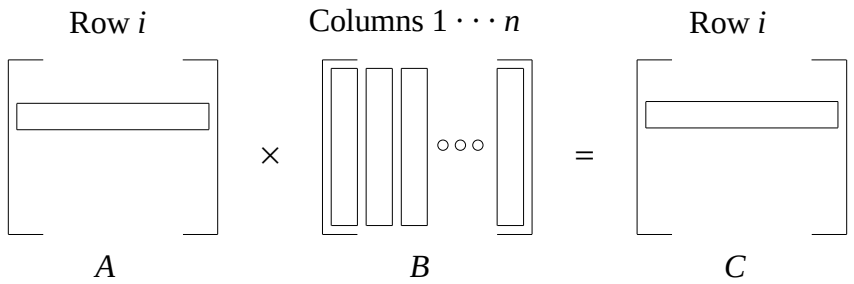


Conceptual Unit of Computation is Element of *C*.

- One thread per element of the resulting matrix.
- Consider 1000×1000 matrix $\Rightarrow$ 1 Million threads.

$\Rightarrow$ 1 Million threads is likely to overwhelm the system.

REDUCING THE NUMBER OF THREADS

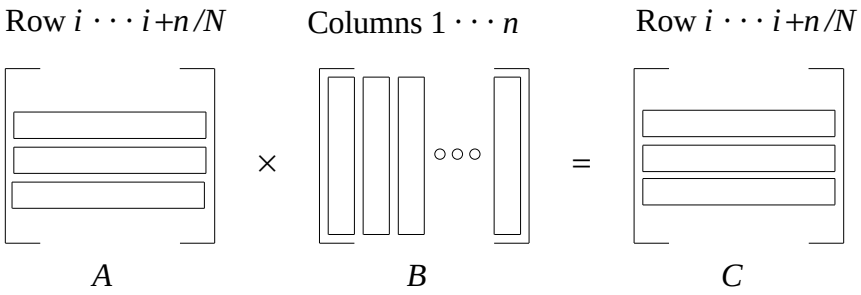


Conceptual Unit of Computation is Row of *C*.

```
FOR i = 1 TO n
    threads[t] = FORK (ComputeRow, i)
    t = t + 1
ENDFOR
JOIN (threads)
```

```
ComputeRow (i)
FOR j = 1 to n
    FOR k = 1 to n
        C[i,j] = C[i,j] + (A[i,k] * B[k,j]);
    ENDFOR
ENDFOR
```

ADAPTING TO THE NUMBER OF PROCESSORS



Conceptual Unit of Computation is Group of Rows of *C*.

```
FOR i = 1 TO n BY n/N
    threads[t] = FORK (ComputeRow, i, i + n/N)
    t = t + 1
ENDFOR
JOIN (threads)
```

```
ComputeRow (start, end)
FOR i = start to end
    FOR j = 1 to n
        FOR k = 1 to n
            C[i,j] = C[i,j] + (A[i,k] * B[k,j]);
        ENDFOR
    ENDFOR
ENDFOR
```