

# T5 DATA-AWARE TASK DECOM.

## Improving locality

```
void matmul (double C[MATSIZE][MATSIZE],
             double A[MATSIZE][MATSIZE],
             double B[MATSIZE][MATSIZE]) {
    int i, j, k;

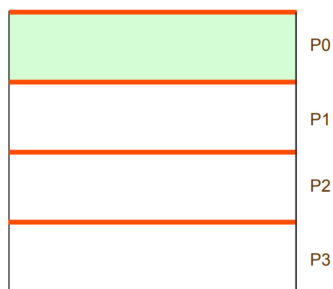
    #pragma omp parallel
    {
        int myid = omp_get_thread_num();
        int howmany = omp_get_num_threads();
        int i_start = myid * (MATSIZE/howmany);
        int i_end = i_start + (MATSIZE/howmany);
        if (myid == howmany-1) i_end = MATSIZE;

        for (int i=i_start; i<i_end; i++)
            for (int j=0; j<MATSIZE; j++)
                for (int k=0; k<MATSIZE; k++)
                    C[i][j] += A[i][k]*B[k][j];
    }
}
```

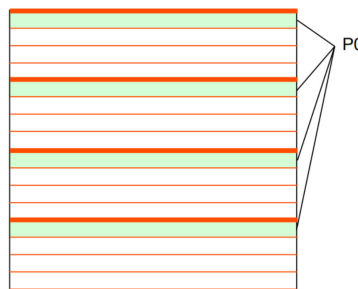
- Tareas implícitas en el parallel
- Usar el identificador del thread para decidir qué ejecutar y quién

→ Balanceo de carga: **PROBLEMA** la última tarea implícita puede hacer un núm. diferente de iter.

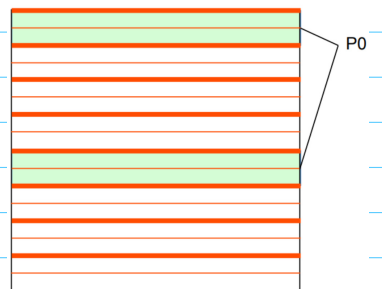
### BLOCK



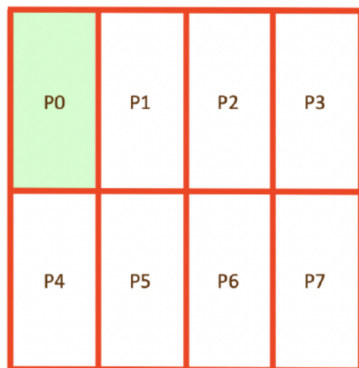
### CYCLIC



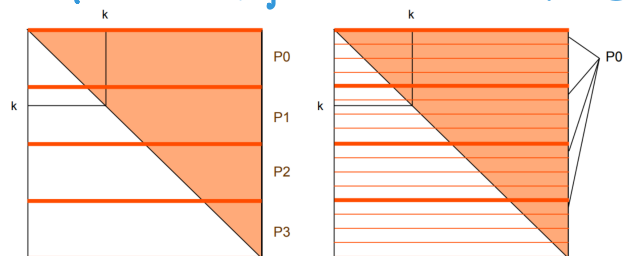
### BLOCK-CYCLIC



### 2D BLOCK / BLOCK DATA DECOMPOSITION



### B & C in triangular iteration space.



Transp / 4

## Type of Data Decompositions:

- **Input:** parte las estructuras de datos de entrada entre tareas.  
la regla del Owner Computes implica que **TODAS** las computaciones que usan datos de la entrada se ejecutarán por la tarea que tiene asignada el input.
- **Output:** parte las estructuras de datos de salida entre tareas.  
la Owner Compute es igual.  
→ Se pueden combinar.

## False sharing:

Block  
Cyclic  
Siempre  
KK?  
Tp. 21/22

