

Programación paralela

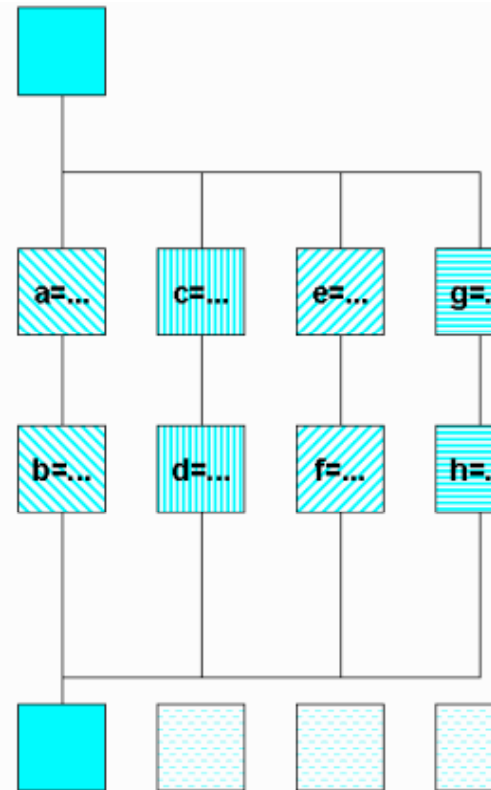
Introducción a OpenMP (II)

José M. Cecilia

OpenMP sections Directives

C / C++:

```
#pragma omp parallel
{
  #pragma omp sections
  {{ a=...;
    b=...; }
  #pragma omp section
  { c=...;
    d=...; }
  #pragma omp section
  { e=...;
    f=...; }
  #pragma omp section
  { g=...;
    h=...; }
} /*omp end sections*/
} /*omp end parallel*/
```



Directiva Master

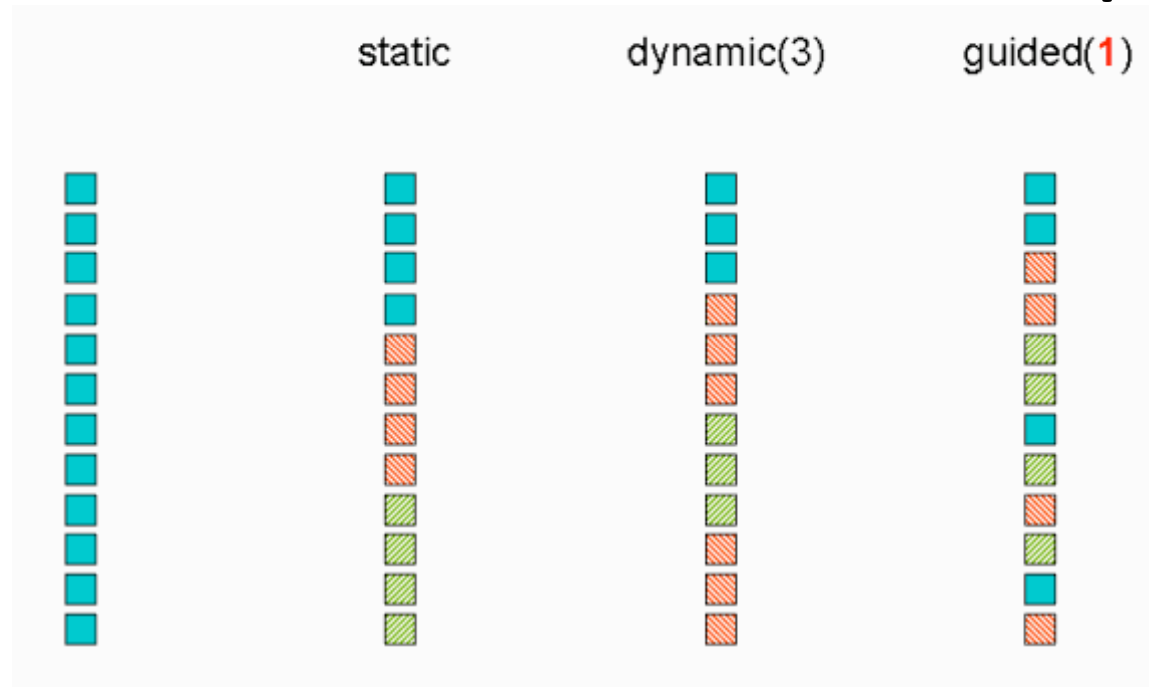
- ✓ La construcción **master** delimita un bloque estructurado que solo es ejecutado por el thread maestro. Los otros *threads* no lo ejecutan.

```
#pragma omp parallel private (tmp)
{
    acciones();
    #pragma omp master
    { acciones_maestro(); }
    #pragma barrier
    acciones();
}
```

Planificación de Tareas. Schedule

- ✓ Diferentes formas de asignar iteraciones a threads:
 - `schedule(static [,chunk])` : “chunk” iteraciones se asignan de manera estática a los threads en round-robin
 - `schedule (dynamic [,chunk])` Cada thread toma “chunk” iteraciones cada vez que está sin trabajo
 - `schedule (guided [,chunk])` Cada thread toma iteraciones dinámicamente y progresivamente va tomando menos iteraciones.

Planificación de un loop



Exclusión Mutua. Sección crítica

```
#pragma omp parallel shared(x,y)
```

```
...
```

```
#pragma omp critical (section1)  
    actualiza(x);  
#pragma omp end critical(section1)
```

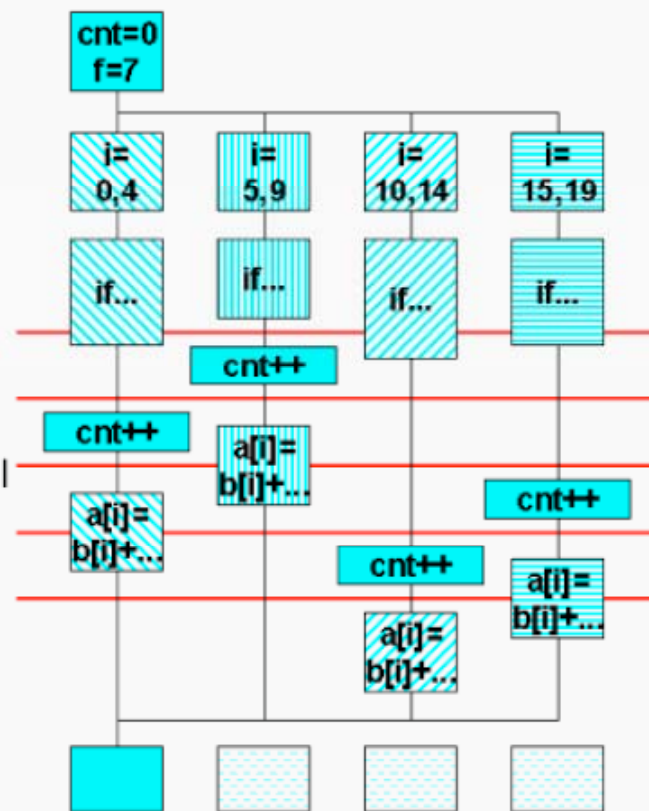
```
...
```

```
#pragma omp critical(section2)  
    actualiza(y);  
#pragma omp end critical(section2)  
#pragma omp end parallel
```

Exclusión Mutua. Sección crítica

```
C++: cnt = 0;
      f=7;
      #pragma omp parallel
      {
        #pragma omp for
        for (i=0; i<20; i++) {
          if (b[i] == 0) {
            #pragma omp critical
            cnt ++;

            /* endif */
            a[i] = b[i] + f * (i+1);
          } /* end for */
        } /*omp end parallel */
```



Barreras

- ✓ Los *threads* se detienen hasta que todos alcancen la barrera
- ✓ Sintaxis `#pragma omp barrier`

Ejemplo

```
#pragma omp parallel
#pragma omp for
for(i=0;i<n;i++) {
<acciones>
#pragma omp barrier
<acciones> }
```


Paralelismo anidado

- ✓ Definir diferentes niveles de paralelismo (Ej. bucles for anidados).
- ✓ `omp_set_nested(1);`

```
omp_set_num_threads(2);  
#pragma omp parallel  
{  
    <Acciones nivel 1>  
    omp_set_num_threads(4);  
    #pragma omp parallel  
    {  
        <Acciones nivel 2>  
        omp_set_num_threads(2);  
        ....  
        ....  
    }....  
}
```

Variables de entorno

- ✓ OMP_SCHEDULE. Tipo de scheduling para for y parallel for.
 - ✓ OMP_NUM_THREADS. Número de threads a usar, aunque se puede cambiar internamente con la función de librería.
 - ✓ OMP_DYNAMIC – Activa o desactiva el ajuste dinámico del número de threads.
 - ✓ OMP_NESTED – Activa o desactiva el anidamiento.
-
- ✓ `$>export OMP_NUM_THREADS = 4`
 - ✓ `$>export OMP_NESTED = 1`

bibliografía

- OpenMP Official Website: www.openmp.org
- OpenMP 4.0 Specifications and Examples.
- An OpenMP book – Rohit Chandra, “Parallel Programming in OpenMP”. Morgan Kaufmann Publishers.
- Compunity: The community of OpenMP researchers and developers in academia and industry – <http://www.compunity.org/>
- Conference papers: – WOMPAT, EWOMP, WOMPEI, IWOMP
<http://www.nic.uoregon.edu/iwomp2005/index.html#program>

GRACIAS