Scheduling I

Por defecto: se usa el scheduling por defecto, definido por la *ICV* (*internal control variable*) ***def-sched-var***

```
#pragma omp parallel for
for (...)
{ ... }
```

Explícito: con cláusula *schedule* en el *for*

```
#pragma omp parallel for schedule(scheduling-type)
for (...)
{ ... }
```

Scheduling II

Runtime: El scheduling se determina por el valor de la *ICV* (*internal control variable*) ***run-sched-var***.

Podemos establecer su valor mediante la variable de entorno OMP_SCHEDULE

O bien, mediante la función *omp_set_schedule(kind,chunk)*

```
fran@nagrella:~$ export OMP_SCHEDULE="static,2"
```

In omp.h

```
  typedef enum omp_sched_t{

    omp_sched_static=1,
    omp_sched_dynamic=2,
    omp_sched_guided=3,
    omp_sched_auto=4

  }omp_sched_t;
```

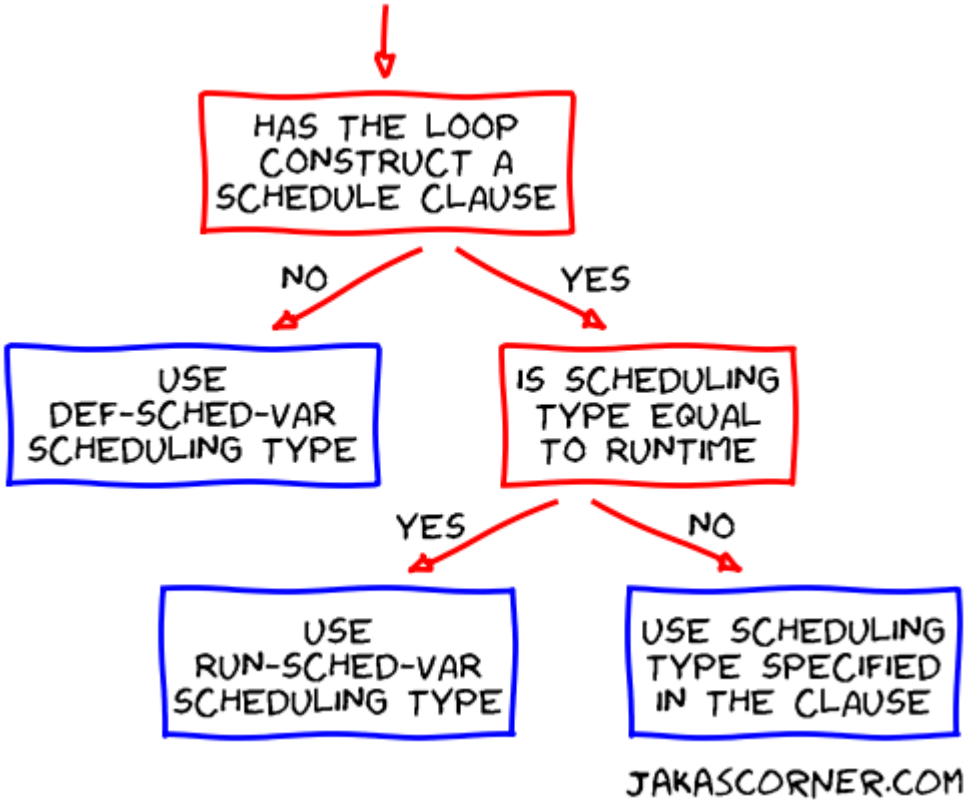
```
#include <stdio.h>
#include <omp.h>
int main(void) {
  omp_sched_t kind;
  int chunk;
  omp_get_schedule(&kind, &chunk);
  printf("%d %d\n", kind, chunk);

...
  omp_set_schedule(omp_sched_static,1);
...

}
```

Scheduling III

Flujo de control para scheduling *(http://jakascorner.com/blog/2016/06/omp-for-scheduling.html)*

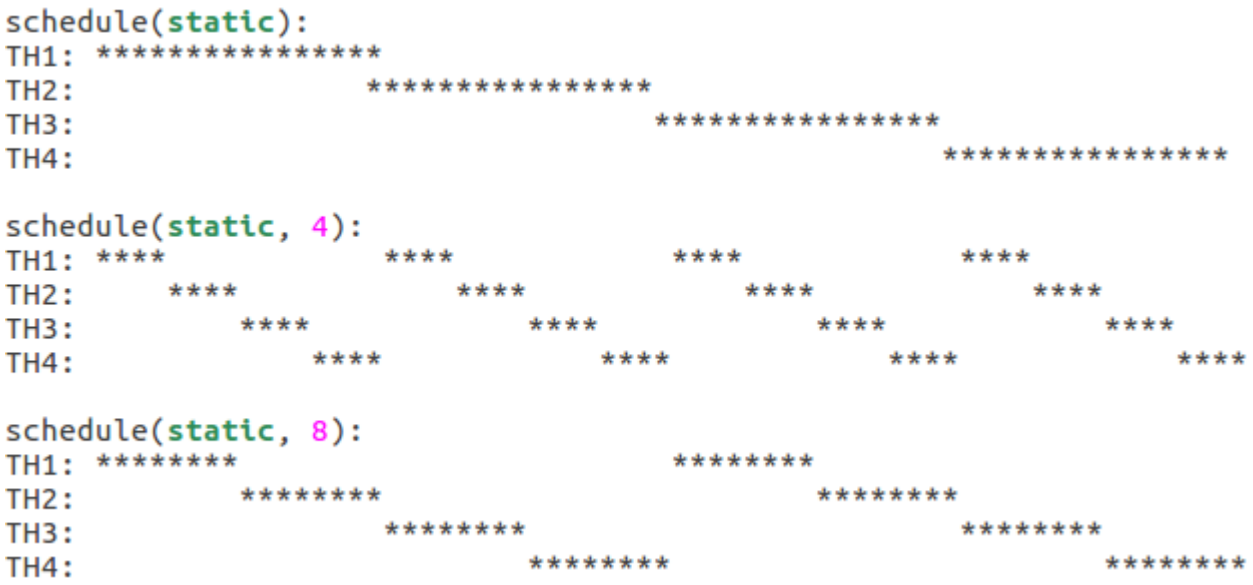


Tipos de Scheduling I

schedule(static, chunk-size)

OpenMP divide las iteraciones en lotes de tamaño *chunk-size* y los asigna a las hebras en *round-robin*

Si no hay *chunk-size*, divide las iteraciones en partes lo más parecidas posible y las asigna a cada hebra

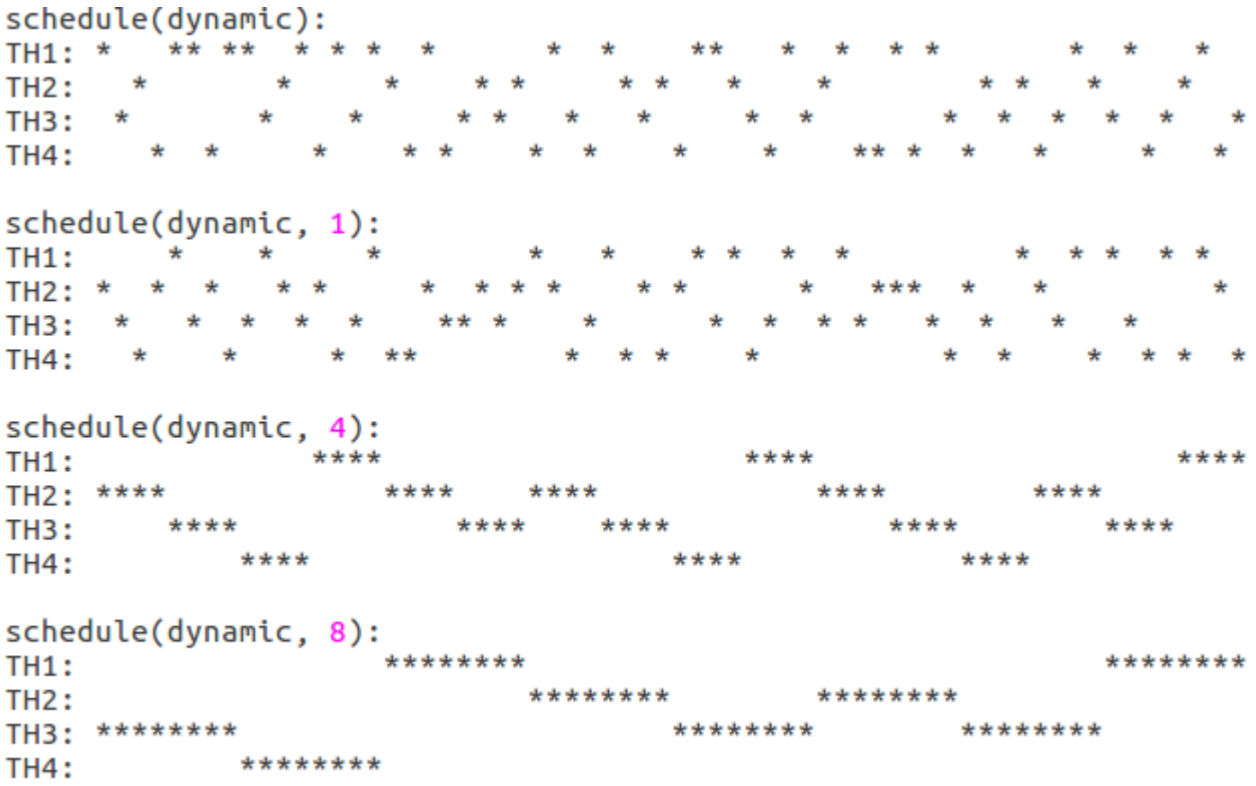


Tipos de Scheduling II

schedule(dynamic, chunk-size)

OpenMP divide las iteraciones en lotes de tamaño *chunk-size* y los asigna en función de la hebra que va acabando.

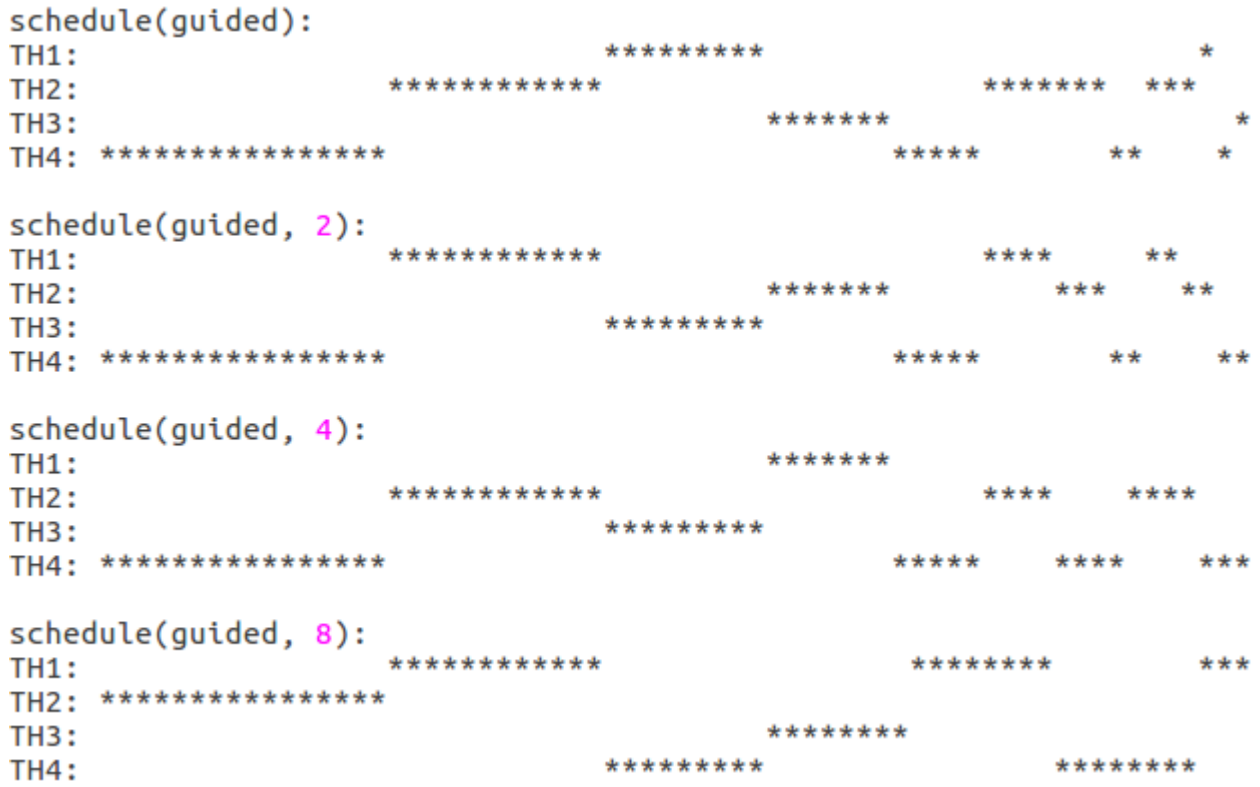
Por defecto *chunk-size* es 1.



Tipos de Scheduling III

schedule(guided, chunk-size)

Es como *dynamic*, pero aquí el tamaño de los lotes es proporcional a las iteraciones que aún no se han asignado. El mínimo tamaño a repartir es *chunk-size*.



Tipos de Scheduling IV

schedule(runtime)

El tipo de scheduling se fija en tiempo de ejecución. Depende de la ICV *run-sched-var*