

Tareas (tasks) en OpenMP

Carlos E. Alvarez¹.

¹Dep. de Matemáticas aplicadas y Ciencias de la Computación, Universidad del Rosario

2020-I

Dos problemas

- Cuando el grupo de tareas a realizar es no homogéneo, algunos procesadores van a tardar mas que otros
- ¿Que pasa si queremos paralelizar un ciclo pero no sabemos de antemano cuantas iteraciones son? (while)

Tareas (Tasks)

Idea: Una UE se encarga de empacar y repartir tareas. Las demás van pasando a recojer y ejecutar estas tareas. En cunato una UE termina su tarea regresa por otra, hasta que no queden mas.

Tareas (Tasks)

Implementación:

```
1  #pragma omp parallel
2  {
3      #pragma omp single //thread that creates the tasks
4      {
5          .
6          .
7          #pragma omp task
8          {
9              //Task to undertake
10         }
11         .
12         .
13     }
14 }
```



Ejemplo: Fibonacci

```
1  int fibonacci(int n) {  
2      if (n <= 1)  
3          return n;  
4      else {  
5          int x = fibonacci(n-1);  
6          int y = fibonacci(n-2);  
7          return x+y;  
8      }  
9  }
```

```

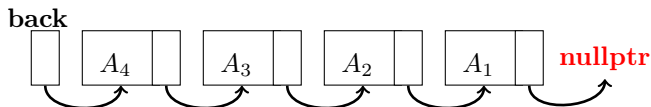
1  int t_fibonacci(int n, int &res) {
2      if(n <= 1){
3          res = n;
4      }else{
5          int x, y;
6          #pragma omp task shared(x)
7          {
8              t_fibonacci(n-1, x);
9          }
10         #pragma omp task shared(y)
11         {
12             t_fibonacci(n-2, y);
13         }
14         #pragma omp taskwait //barrier to wait for next task
15         res = x+y;
16     }
17 }

```

```
1  int fibonacci(int n) {  
2      int res;  
3      #pragma omp parallel  
4      #pragma omp single  
5      {  
6          t_fibonacci(n, res);  
7      }  
8      return res;  
9  }
```

El algoritmo anterior no genera un hilo cada vez que se llama a la función. En lugar de eso, un número fijo de hilos se van turnando la ejecución de las funciones recursivas.

Ejemplo: Recorrido de una lista



```
1 node* p = back
2 while (p):
3     process(p)
4     p = p->next
```

- La manera de recorrer una lista es estrictamente secuencial
- Hay que realizar una serie de operaciones sobre cada elemento
- Mientras una UE trabaja sobre el elemento i , otra puede continuar al elemento $i + 1$

Ej: El profesor proveerá una versión secuencial de un programa que realiza una tarea dada sobre cada elemento de una lista. Cree una versión paralela usando tasks de openMP.