Programación Paralela
2020-II

Prof. José Fiestas

1). Describa la diferencia entre ambas barreras para el constructor task

a)

```
#pragma omp task {}      // T1
#pragma omp task         // T2
{
    #pragma omp task {}  // T3
}
#pragma omp task {}      // T4

#pragma omp taskwait
```

b)

```
#pragma omp task {}          // T1
#pragma omp taskgroup
{
    #pragma omp task         // T2
    {
        #pragma omp task {}  // T3
    }
    #pragma omp task {}      // T4
}
```

Solución: En a) solo T1, T2, T4 han culminado al llegar a taskwait, en b) solo T2, T3 y T4 han culminado en taskgroup
(4 pt)

2) Escriba un código en paralelo (en OMP) que permita hayar la raiz de una función. Considere el uso de los constructores presentados en clase.
(3 pt)

3) ¿En qué orden recorrerá esta función en paralelo un árbol predefinido?

```
struct node {
  struct node *left;
  struct node *right;
};
extern void process(struct node *);
void traverse( struct node *p ) {
  if (p->left)
#pragma omp task   // p is firstprivate by default
      traverse(p->left);
  if (p->right)
#pragma omp task    // p is firstprivate by default
      traverse(p->right);
  process(p);
}
```

Solución: no está definido el orden, ya que no existe una barrera taskwait para la ejecución de **process(p)**
(3 pt)

4) Utilice las siguientes fracciones de código para enunciar 3 principales diferencias y/o similitudes entre task y section, respecto a la ejecución en paralelo de las funciones indicadas

Solución: task no da exclusividad a la tarea por hilo, mientras section si lo

```
#pragma omp parallel sections
    {
      #pragma omp section
      {
          funcion_a();
      }
      #pragma omp section
     {
          funcion_b();
        }
    }


#pragma omp single nowait
    {
      #pragma omp task
          funcion_a();
      #pragma omp task
          funcion_b();
    }
#pragma omp taskwait
```

hace. Section tiene una barrera implícita, mientras que task necesita el taskwait. Task y section obtendrán en el ejemplo el mismo resultado
(3 pt)

5) Utilice el ejercicio 03 para esbozar un código en C+OMP para recorrer un árbol **inorder**

```
void printInorder(struct node* node)
{
        if (node == NULL)
        return;

/* first recur on left child */
#pragma omp task
        printInorder(node->left);

/* then print the data of node */
#pragma omp taskwait
        printf("%d ", node->data);

/* now recur on right child */
#pragma omp task
        printInorder(node->right);
}
```

```
int main()
{
#ifdef _OPENMP
        omp_set_num_threads(4);
#endif
        struct node *root = newNode(1);
        root->left              = newNode(2);
        root->right             = newNode(3);
        root->left->left        = newNode(4);
        root->left->right       = newNode(5);
//      root->left->right->left         = newNode(6);
//      root->left->right->right        = newNode(7);

//      printf("\n Preorder traversal of binary tree is \n");
//      printPreorder(root);

#pragma omp parallel
#pragma omp single
//      printf("\n Inorder traversal of binary tree is \n");
//      printInorder(root);
        printf("\n Postorder traversal of binary tree is \n");
        printPostorder(root);

        return 0;
}
```

(4 pt)

6) Se utiliza una tarjeta gráfica NVidia GeForce GT 650M, con 1024 threads per block y hasta 65535 blocks (en 1 dimensión) para modelos de dinámica molecular

- si cada thread puede procesar una operacion de coma flotante (FLOP), ¿Cuantos átomos podría procesar teóricamente esta tarjeta en forma simultánea, si cada uno ejecuta 200 FLOPs por iteración?
  67107840 hilos / 200 = 335539 atomos en forma simultánea

- ¿Si el tiempo de cálculo por operación es $5\text{x}10^{-7}$ segundos, cuál sería el tiempo de procesamiento luego de un millón de iteraciones ? ¿Cual sería el tiempo si el código se ejecutaría en forma secuencial?
  $T_p = 5 \cdot 10^{-7} * 1 \cdot 10^6 = 0.5$ sec
  $T_s = 0.5 * 1 \cdot 10^6 = 5 \cdot 10^5$ sec

- En otro modelo, se quiere simular el número de células del cuerpo humano (1 trillon) ¿Cuál sería el tiempo total de procesamiento, si cada célula necesita un promedio de 100 FLOPs para modelar la interacción dinámica, con un tiempo de cálculo por operación de $5\text{x}10^{-12}$ segundos, y sabiendo que se necesita un mínimo de un millon de iteraciones para completar la simulación? ¿Cual sería el tiempo de un código secuencial?

- ¿Cual sería una combinación óptima de threads x blocks para declarar el kernel de un modelo de n moléculas de 100,000 átomos cada una?
  Fuerza <<blocks_per_grid, threads_per_block, >>
  Condicion: $100000 \cdot n \ mod(65535) = 0$
  Cantidad de hilos por bloque = $100000 \cdot n/65535$

(**4 puntos**)