

Curso PASA - OpenMP

Calcular π mediante *integración numérica*.

Considerando que π es el área del círculo unidad, podemos aproximarlo calculando el área de un cuarto de círculo mediante una *suma de Riemann*.

- Sea $f(x) = \sqrt{1 - x^2}$ la función que describe el cuarto de círculo para $x \in [0, 1]$;
- Entonces

$$\pi/4 \approx \sum_{i=0}^{N-1} \Delta x f(x_i) \quad \text{donde } x_i = i\Delta x \text{ y } \Delta x = 1/N \quad (1)$$

Escribir un programa para esto y paralelizarlo mediante OpenMP.

1. Poner una directiva `parallel` al rededor del loop. El resultado sigue siendo correcto? Ha bajado el tiempo de ejecución con la cantidad de threads? (La respuesta debería ser no y no.)
2. Cambiar la directiva `parallel` a `parallel for` (o `parallel do`). El resultado pasó a ser correcto? Mejoró el tiempo de ejecución? (La respuesta debería ser no y si.)
3. Poner una directiva `critical` junto al update de la variable que acumula el valor de π . Contestar las mismas preguntas. (La respuesta debería ser si y "por Dios no!".)
4. Quitar el `critical` y agregar `reduction(+:cuartopi)` (donde `cuartopi` es la variable que acumula el valor de π) a la directiva `parallel for`. Ahora debería ser correcta y eficiente.

Utilizar distinta cantidad de hilos y registrar la aceleración obtenida respecto al código secuencial. Puede observar una diferencia de desempeño entre el código paralelo y el secuencial?