

Explicación de la Estrategia de Paralelización

1. Breve descripción del problema

El problema consiste en la simulación de un ecosistema compuesto por tres tipos de entidades: plantas, herbívoros y depredadores. El entorno está representado por una cuadrícula bidimensional de tamaño configurable, donde las plantas crecen de forma aleatoria, los herbívoros se desplazan para alimentarse de las plantas y reproducirse, y los depredadores cazan herbívoros para sobrevivir y reproducirse. Cada entidad tiene energía que se consume con el movimiento y se recupera al alimentarse. La simulación avanza en pasos discretos de tiempo, actualizando las posiciones, el estado de energía y la población de cada tipo de entidad hasta completar el número de iteraciones especificado. El objetivo es comparar el rendimiento de la simulación en una versión secuencial y una versión paralela, midiendo tiempos de ejecución y observando el comportamiento poblacional.

2. Estrategia de paralelización utilizada

Para acelerar la simulación, se utilizó una estrategia de paralelización con OpenMP que distribuye el trabajo entre múltiples hilos. Se identificaron las partes más costosas en tiempo de ejecución y que son fácilmente paralelizables debido a que operan sobre elementos independientes de la cuadrícula o listas de entidades:

- Crecimiento de plantas: se paralelizó con `'#pragma omp parallel for collapse(2)'` para recorrer la cuadrícula en dos dimensiones y permitir que múltiples hilos procesen diferentes celdas simultáneamente.
- Movimiento y alimentación de herbívoros: se utilizó `'#pragma omp for schedule(dynamic,128)'` para balancear la carga de trabajo, ya que algunos herbívoros pueden morir o reproducirse en diferentes momentos.
- Movimiento y alimentación de depredadores: de forma similar a los herbívoros, se paralelizó el procesamiento con reparto dinámico y uso de locks para manejar conflictos al acceder a posiciones compartidas.
- Cálculo de estadísticas: se empleó `'reduction(+:variable)'` para sumar conteos en paralelo sin condiciones de carrera.

Para evitar condiciones de carrera al acceder a celdas compartidas, se implementaron arreglos de ocupación y locks por celda (`'omp_lock_t'`) que garantizan que solo un hilo modifique el estado de una celda a la vez. Además,

se usaron funciones auxiliares como 'try_repro_locked' para manejar la reproducción de animales asegurando la consistencia de datos.

3. Justificación del uso de OpenMP

OpenMP fue elegido debido a su facilidad de integración con código C existente y su eficiencia en la paralelización de bucles, que constituyen la mayor parte de la carga computacional de esta simulación. La simulación involucra grandes cantidades de iteraciones independientes sobre datos distribuidos en memoria, lo que se adapta perfectamente al modelo de paralelismo de memoria compartida de OpenMP.

Se utilizó un esquema de paralelismo principalmente estático para el crecimiento de plantas, y dinámico ('schedule(dynamic,128)') para el movimiento y alimentación de animales, con el fin de equilibrar la carga de trabajo entre hilos cuando el tiempo de procesamiento por entidad es irregular. No fue necesario activar paralelismo anidado, ya que la estructura de la simulación no requiere procesar sub-bucles dentro de bucles ya paralelizados.

OpenMP también permite un control fino del número de hilos mediante la variable de entorno 'OMP_NUM_THREADS', lo que facilita ajustar el rendimiento a la arquitectura disponible sin modificar el código fuente.