

I - S - 2021 - RRF - Programación Paralela y Concurrente - 002

Tarea04: goldbach_omp_mpi

En las tareas anteriores usted escribió programas que reciben números enteros en la entrada estándar y calculan las sumas de Goldbach de forma serial (tarea01), concurrente con Pthreads (tarea02), y comparó su rendimiento (tarea03). Estas soluciones aprovechan los recursos concurrentes de una máquina, pero no escalan, ni aprovechan varias computadoras disponibles en una organización o un clúster.

En esta tarea su objetivo es distribuir el cálculo de sumas de Goldbach entre varias máquinas usando dos tecnologías: OpenMP y MPI. Su programa recibirá los números en la entrada estándar en el mismo formato, y deberá producir los resultados en el mismo orden, de tal forma que pase los casos de prueba.

Concurrencia declarativa [30%]

Utilice la tecnología de paralelismo de datos OpenMP para paralelizar el cálculo de sumas. Dadas las facilidades de esta tecnología, utilice una unidad de descomposición más granular que los números. Asegúrese de que su implementación pasa los casos de prueba, y no genera diagnósticos del linter.

El cálculo de las sumas con los for.

Nota: Si toma como código base la tarea02 o tarea03, deberá reemplazar el código de Pthreads por OpenMP. Si usa como código base su solución serial (tarea01), asegúrese de aplicarle las optimizaciones seriales que realizó en tareas posteriores y de que éstas pasen los casos de prueba.

Comparación Pthreads-OpenMP [20%]

Como hubo un cambio en la implementación de la concurrencia, compare el rendimiento de su implementación con OpenMP contra la de Pthreads siguiendo el mismo procedimiento que aplicó en la tarea03, utilizando tantos hilos como núcleos de CPU hay en el sistema con el caso de prueba 023. Calcule el Speedup y la eficiencia. Cree un gráfico combinado con esas dos secuencias. Agréguelo a un documento de análisis de resultados donde incluya una discusión no mayor a 500 palabras.

Distribución [30%]

Para hacer su solución escalable y aprovechar las computadoras de un clúster, distribuya el cálculo de sumas de Goldbach utilizando la tecnología MPI. Tome en cuenta que su programa debe leer los datos de la entrada estándar y no de archivos ni argumentos de línea de comandos. Descomponga la unidad de trabajo a distribuir, ésta puede tener mayor granularidad que la unidad usada para OpenMP. Idee un mapeo eficiente entre las unidades de trabajo y los procesos, de tal forma que los recursos del clúster se aprovechen lo más equitativamente posible. Asegúrese de que su solución distribuida con MPI pase los casos de prueba usando todos los núcleos con OpenMP, y que no genere diagnósticos del linter.

Comparación OpenMP-MPI [20%]

Compare el rendimiento de su solución distribuida usando el caso de prueba 023 en el clúster de Arenal. Cree un proceso por cada nodo secundario del clúster, el cual cree tantos hilos de ejecución como núcleos de CPU hay disponibles en la máquina donde corre.

Mida la duración de la versión distribuida, calcule el incremento de velocidad

y la eficiencia. Agregue los resultados a su gráfico, de tal forma que en el eje-x se pueda comparar la versión concurrente con Pthreads, con OpenMP, y finalmente la distribuida con MPI/OpenMP. Agregue a su discusión un análisis de los resultados que obtuvo con la versión distribuida.