Programación con memoria distribuida

Introducción

Entrega correspondiente al trabajo de ***Programación con memoria distribuida***.

Alumnos

- Ulises Jeremias Cornejo Fandos 13566/7
- Federico Ramón Gasquez 13598/6

Ejercicio 1

El juego de las N-Reinas consiste en ubicar sobre un tablero de ajedrez N reinas sin que estas se amenacen entre ellas. Una reina amenaza a aquellas reinas que se encuentren en su misma fila, columna o diagonal. La solución al problema de las N-Reinas consiste en encontrar todas las posibles soluciones para un tablero de tamaño NxN.

Idea General

El almacenamiento de las posiciones de las reinas se dispone en el `int * positions`. Dadas las reglas impuestas, solo puede haber una reina por fila. Luego, se utiliza el índice del arreglo de enteros como indicador de fila y el valor como indicador de columna. De este modo, `positions[i] = j` indica que la reina *i* está ubicada en la posición `(i, j)`.

La solución del problema utiliza el algoritmo de **backtracking**.

La idea es colocar reinas una por una en diferentes columnas, comenzando desde la columna más a la izquierda. Cuando colocamos una reina en una columna, buscamos los enfrentamientos con las reinas ya colocadas. En la columna actual, si encontramos una fila para la cual no hay conflicto, marcamos esta fila y columna como parte de la solución. Si no encontramos esa fila debido a los enfrentamientos, volvemos atrás y devolveremos el resultado falso.

Finalmente se imprimen el resultado, el tiempo de ejecución y se libera el espacio de memoria alocado.

MPI

La resolución con MPI está basada en la solución secuencial planteada. Se distribuye el trabajo en distintos procesos que colaboran entre si para lograr encontrar el resultado en el menor tiempo posible utilizando el modelo `Master-Worker`.

El proceso 0 es el **master**, y se encarga de repartir los trabajos, *bajo demanda*, a los procesos restantes, **workers/slaves**. Cada proceso worker solicita trabajo al *master*, el cual responde con un valor de columna para la primer reina de una solución. Así, cada proceso trabaja con uno de los

árboles que representaría todas las soluciones posibles para un determinado valor de la primer reina. Adicionalmente, el master busca parte de las soluciones.

Los *slaves* realizan un procesamiento similar al secuencial para un valor de columna para la primer reina y al finalizar el procesamiento solicitan al *master* otro valor de columna para la primer reina.

Una vez terminado el procesamiento se realiza la reducción de la cantidad de soluciones encontradas, local a cada proceso, a una sola variable.

Finalmente se imprimen el resultado, el tiempo de ejecución local a cada proceso y el total, liberando el espacio de memoria alocado.

Métricas

Las métricas mostradas corresponden a promedios de un conjunto de 5 mediciones. Los tiempos dispuestos en las siguientes tablas están medidos en segundos. Las computadoras utilizadas para tomar los tiempos de ejecución de este ejercicio son computadoras de la sala de PC de postgrado.

Se toman mediciones con valores de N en [2, 16], observando que el tiempo de ejecución de la solución utilizando MPI es mejor que la secuencial cuando N > 6 dado el overhead causado por el pasaje de mensajes y la espera sincrónica de algunos de estos.

Luego, se descarta en el analisis los tiempos de ejecución tomados para todo N < 10, dado que la diferencia de los tiempos era despreciable.

A continuación se muestra el analisis de los tiempos de ejecución para N en [11, 16].

---

4 procesos (1 maquina)

Tiempos

Tamaño de entrada	Tiempo secuencial	Tiempo MPI
11	0.072539	0.025025
12	0.265769	0.116403
13	1.517316	0.727294
14	9.683095	4.611232
15	66.442554	32.229390
16	488.179246	235.053206

Desbalance de carga

Tamaño de entrada	MPI
11	1.027419
12	1.082382
13	1.108826
14	1.063742

Tamaño de entrada	MPI
15	1.081011
16	1.086238

Speedup

Tamaño de entrada	MPI
11	2.898661
12	2.283179
13	2.086248
14	2.099893
15	2.061551
16	2.076888

Eficiencia

Tamaño de entrada	MPI
11	0.724665
12	0.570795
13	0.521562
14	0.524973
15	0.515387
16	0.519222

---

4 procesos (2 maquina)

Tiempos

Tamaño de entrada	Tiempo secuencial	Tiempo MPI
11	0.072539	0.031485
12	0.265769	0.116569
13	1.517316	0.675230
14	9.683095	4.341968
15	66.442554	30.336491
16	488.179246	221.225224

Desbalance de carga

Tamaño de entrada	MPI
11	1.178317
12	1.027199
13	1.090435
14	1.070537
15	1.095356
16	1.084888

Speedup

Tamaño de entrada	MPI
11	2.303923
12	2.279928
13	2.247109
14	2.230117
15	2.190186
16	2.206707

Eficiencia

Tamaño de entrada	MPI
11	0.575981
12	0.569982
13	0.561777
14	0.557529
15	0.547546
16	0.551676

---

8 procesos (2 maquina)

Tiempos

Tamaño de entrada	Tiempo secuencial	Tiempo MPI
11	0.072539	0.031905
12	0.265769	0.096253
13	1.517316	0.419270
14	9.683095	2.722888
15	66.442554	18.377330
16	488.179246	136.249856

Desbalance de carga

Tamaño de entrada	MPI
11	0.560630
12	0.593408
13	0.608157
14	0.648605
15	0.618368
16	0.631707

Speedup

Tamaño de entrada	MPI
11	2.273593

Tamaño de entrada	MPI
12	2.761150
13	3.618947
14	3.556185
15	3.615462
16	3.582971

Eficiencia

Tamaño de entrada	MPI
11	0.284199
12	0.345144
13	0.452368
14	0.444523
15	0.451933
16	0.447871