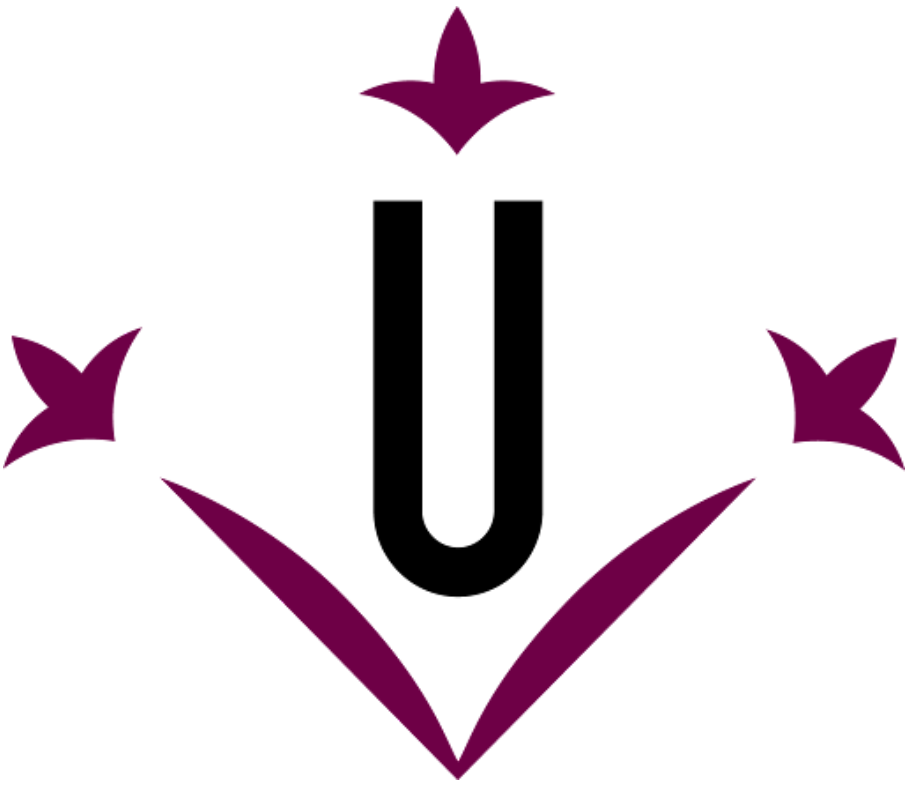
PRÀCTICA MPI

IMPLEMENTACIÓ APLICACIÓ PARAL·LELA



van de Crommert Rodoreda, Òscar
Gonzàlez Saló, Marc

INTRODUCCIÓ

Per paral·lelitzar el mètode de cerca exhaustiu utilitzant MPI (Message Passing Interface), primer cal trobar els"HotSpots" de l'algoritme on es farà ús de la llibreria MPI per paral·lelitzar el programa i baixar el temps d'execució. Seguidament, es farà el disseny dels missatges que s'envien els processos per poder garantir una comunicació estable, global i sense interrupcions.

IMPLEMENTACIÓ

Amb d'aquesta implementació tots els processos treballen per igual així que s'utilitza un patró de disseny SPMD (Single Program Multiple Data), el procés zero treballa més que la resta perquè és l'encarregat de fer les comparacions entre els millors costos dels diferents processos.

Tots els processos reben un conjunt de combinacions les quals han de processar i després comparar per obtenir la solució òptima dins del rang assignat per finalment retornar el cost al procés zero:

```
int CalcularCombinacionOptima(TCombinacionArboles PrimeraCombinacion,
TCombinacionArboles UltimaCombinacion)
{
	//la resta de la funció
	return Optimo.Coste;
}
```

Per enviar fem ús de la funció MPI_Pack i MPI_Send. Per fer la correcta implementació del mètode MPI_Pack hem hagut d'investigar exhaustivament, ja que els continguts a empaquetar són estructures que contenen estructures:

```
MPI_Pack(&solucion.Combinacion, 1, MPI_UNSIGNED_LONG, buf, bufsize, &position,
MPI_COMM_WORLD);
MPI_Pack(&solucion.ArbolesTalados.NumArboles, 1, MPI_INT, buf, bufsize,
&position, MPI_COMM_WORLD);
MPI_Pack(solucion.ArbolesTalados.Arboles, solucion.ArbolesTalados.NumArboles,
MPI_INT, buf, bufsize, &position, MPI_COMM_WORLD);
MPI_Pack(&solucion.Coste, 1, MPI_INT, buf, bufsize, &position, MPI_COMM_WORLD);
MPI_Pack(&solucion.CosteArbolesRestantes, 1, MPI_INT, buf, bufsize, &position,
MPI_COMM_WORLD);
MPI_Pack(&solucion.LongitudCerca, 1, MPI_FLOAT, buf, bufsize, &position,
MPI_COMM_WORLD);
MPI_Pack(&solucion.MaderaSobrante, 1, MPI_FLOAT, buf, bufsize, &position,
MPI_COMM_WORLD);

MPI_Send(buf, position, MPI_PACKED, 0, 0, MPI_COMM_WORLD);
```

Després que el procés zero finalitzi els càlculs del seu apartat, assigna el millor resultat de la seva part com a Òptim, això ens permetrà fer comparacions en un futur. Tot seguit per rebre un paquet de cada procés fem ús del mètode MPI_Recv i del MPI_Unpack:

```
MPI_Recv(buf, bufsize, MPI_PACKED, MPI_ANY_SOURCE, 0, MPI_COMM_WORLD, &status);

MPI_Unpack(buf, bufsize, &position, &received_solucion.Combinacion, 1,
MPI_UNSIGNED_LONG, MPI_COMM_WORLD);
MPI_Unpack(buf, bufsize, &position, &received_solucion.ArbolesTalados.NumArboles,
1, MPI_INT, MPI_COMM_WORLD);
MPI_Unpack(buf, bufsize, &position, &received_solucion.ArbolesTalados.Arboles,
received_solucion.ArbolesTalados.NumArboles, MPI_INT, MPI_COMM_WORLD);
MPI_Unpack(buf, bufsize, &position, &received_solucion.Coste, 1, MPI_INT,
MPI_COMM_WORLD);
MPI_Unpack(buf, bufsize, &position, &received_solucion.CosteArbolesRestantes, 1,
MPI_INT, MPI_COMM_WORLD);
MPI_Unpack(buf, bufsize, &position, &received_solucion.LongitudCerca, 1,
MPI_FLOAT, MPI_COMM_WORLD);
MPI_Unpack(buf, bufsize, &position, &received_solucion.MaderaSobrante, 1,
MPI_FLOAT, MPI_COMM_WORLD);
```

A mesura que es rep els paquets es desempaqueten i es tornen a formar les estructures. Seguidament, es compara la solució que s'ha rebut amb la que actualment està marcada com a Òptima i, en cas que

tingui un cost menor, se sobreescriu la combinació Òptima per la qual ens acaba d'arribar.
Repetim el procés tants cops com processos-1 tenim.
Un cop finalitzades les comprovacions, imprimim per pantalla la combinació Òptima i el seu cost, juntament amb el temps que hem tardat amb fer tots els càlculs:

```
Datos Entrada:
      Arboles: 3.
      Arbol 1→ (3,0) Coste:10, Long:2.
      Arbol 2→ (5,5) Coste:20, Long:25.
      Arbol 3→ (7,-3) Coste:30, Long:32.

[CalcularCercaOptimaExhaustiva] Evaluación Combinaciones posibles:
[     4] OptimoParcial          4→ Coste 30,  1 Arbol/es talado/s: 3   :
[     6] OptimoParcial          6→ Coste 50,  2 Arbol/es talado/s: 2 3
[     2] OptimoParcial          2→ Coste 20,  1 Arbol/es talado/s: 2
[CalcularCercaOptimaExhaustiva] Tiempo requerido cálculo cerca optima: 0.000
segs. 8 combinaciones evaluadas
      OptimoEX 2→ Coste 20, 1 Arboles talados:2
                              Id:  2  Coord: (5,5)      Cost:  20    Long:  25
```

Per visualitzar millor les estadístiques de temps, de tant el mètode exhaustiu, com del mètode Branch And Bound. Hem modificat el programa perquè no es calculin al mateix moment sinó de forma seqüencial.