



UNIVERSITAT^{DE}
BARCELONA

Pràctica 4. Mapat a memòria i algorismes paral·lels

Noah Márquez Vara
Alejandro Guzman Requena

27 maig 2022

1 INTRODUCCIÓ

La present pràctica, tal i com indica el seu títol, es centra en el mapat a memòria i algorismes paral·lels.

Concretament, en aquesta pràctica s'utilitzarà el mapat a memòria per tal de mapejar matrius, carregar-les fent un desmapat a memòria i posteriorment poder realitzar una multiplicació matricial mitjançant un algorisme amb paral·lelisme.

Hem vist a les classes de teoria i teoricopràctica el concepte de mapejat a memòria i funcions dissenyades especialment per a fer-ho, com és el cas de *mmap*, la qual utilitzarem en els exercicis proposats per tal de mapejar matrius a arxius situats a la memòria de l'ordinador.

També dissenyarem un algorisme per tal de reduir el temps de execució de la multiplicació matricial mitjançant el concepte de paral·lelisme. L'algorisme tractarà d'assignar a diversos processos (tants com processadors tingui l'ordinador; o bé 4 com se'ns demana per l'exercici 3) un nombre de files en concret, aprofitant que les files de la primera matriu en una multiplicació matricial són independents unes d'altres.

Finalment compararem els temps d'execució de les multiplicacions de matrius amb i sense paral·lelisme, per extreure conclusions en relació a la reducció de temps que implica fer servir un algorisme amb paral·lelització i un que no.

2 PROVES REALITZADES

1. Exercici 1

Per tal de comprovar el correcte funcionament del codi, hem fet diverses proves amb diferents mides de matrius per tal de crear-les i mostrar-les per pantalla.

Hem realitzat varies execucions del codi, amb mides diferents, per tal de mostrar per pantalla les matrius inicialitzades amb els coeficients de Vandermonde corresponents, per posteriorment carregar-les de memòria i comprovar que són iguals.

La mida de les matrius les hem canviat internament en el codi, cridant a les dues funcions (una per crear les matrius i l'altra per carregar-les de memòria) amb diferents valors en els paràmetres.

A continuació mostrem el que retorna el programa per l'exercici 1 (no mostrem totes les proves realitzades ja que ocuparia massa espai en el document):

Exemple 1

Vandermonde matrix A with rows = 3 and columns = 3:

1.000000	0.000000	0.000000
1.000000	1.000000	1.000000
1.000000	2.000000	4.000000

Vandermonde matrix B with rows = 3 and columns = 2:

1.000000	0.000000
1.000000	1.000000

```
1.000000    2.000000
```

```
Saved Matrix A:
```

```
1.000000    0.000000    0.000000
1.000000    1.000000    1.000000
1.000000    2.000000    4.000000
```

```
Saved Matrix B
```

```
1.000000    0.000000
1.000000    1.000000
1.000000    2.000000
```

Amb les proves realitzades hem pogut comprovar que els resultats coincideixen amb el que podíem d'esperar.

2. Exercici 2

En aquest exercici es realitza la multiplicació de matrius amb i sense paral·lelisme.

Per avaluar el correcte funcionament del codi, hem realitzat varies proves tant amb mides diferents de matrius com amb diferent número de processos. Aquí, però, mostrem només l'exemple que se'ns demanava a l'enunciat per no allargar massa el document:

Exemple 1

```
Vandermonde matrix A with rows = 3 and columns = 3:
```

```
1.000000    0.000000    0.000000
1.000000    1.000000    1.000000
1.000000    2.000000    4.000000
```

```
Vandermonde matrix B with rows = 3 and columns = 2:
```

```
1.000000    0.000000
1.000000    1.000000
1.000000    2.000000
```

```
A*B (NO parallelism):
```

```
1.000000    0.000000
3.000000    3.000000
7.000000    10.000000
```

```
A*B (Parallelism):
```

```
1.000000    0.000000
3.000000    3.000000
7.000000    10.000000
```

Per tant, es pot veure com independentment del nombre de processos seleccionats la multiplicació matricial amb paral·lelisme es realitza correctament i dona el mateix resultat que sense paral·lelisme.

Comentar que ens surten dos *Warnings* per *expression result unused* però no els considerem importants per la pràctica.

3 PREGUNTES EXERCICI 3

Aprofitem aquest apartat per comentar les proves realitzades en l'exercici 3 ja que ens vindrà bé ajuntar-ho tot en un apartat per mostrar-ho correctament.

En aquest exercici se'ns demanava comprar les implementacions amb i sense paral·lisme implementades a l'exercici 2. Tal i com s'especifica a l'enunciat, les proves s'han realitzat fora de la màquina virtual en un ordinador *MacBook* per aprofitar que està basat en *Unix*.

A continuació mostrem una taula dels temps parcials que ha trigat cadascuna de les implementacions:

No Parallelism	Paralellism
3.482942	0.000983
3.920693	0.002188
3.370554	0.001025
3.394492	0.000923
3.386310	0.001028
3.386336	0.001012
3.361881	0.001059
3.373595	0.000929
3.399457	0.001053
3.507688	0.001063

De mitjana per la implementació sense paral·lisme obtenim: **3.458410** i per la implementació amb paral·lisme obtenim una mitjana de: **0.001201**, per tant, contestant a la pregunta de si la implementació amb paral·lisme millora el temps d'execució, podem veure clarament que sí.

Per tal de contestar a la pregunta de com canvien aquestes mitjanes en funció del nombre de processos i si augmentar aquests sempre implicarà obtenir millors mitjanes, hem fet proves per tal d'arribar a les següents conclusions. Hem fet proves amb 2, 8 i 100 processos i a mesura que augmentàvem el nombre de processos veiem com la mitjana en la implementació amb paral·lisme augmentava. Les mitjanes obtingudes pels nombres de processos indicats anteriorment són les següents: **0.000941** (2 processos), **0.001838** (8 processos), **0.015868** (100 processos).

Podem veure així que augmentar el nombre de processos no té sentit si aquest nombre supera els processadors que té l'ordinador que executa el programa.

4 CONCLUSIONS

Aquesta última pràctica ens ha ajudat en gran mesura a entendre d'una manera molt més completa i estesa el concepte de mapat a memòria i paral·lisme. Les classes teòriques van servir per introduir-nos en el tema, però les hores dedicades a la implementació dels diferents exercicis i a la recerca d'informació per tal d'entendre correctament com funcionava el mapat de memòria i sobretot com implementar-ho, ens han estat molt profitoses.