

PARALELNÍ A DISTRIBUOVANÉ ALGORITMY

DOKUMENTÁCIA PROJEKTU Č. 3

Ján Jusko, Vysoké Učení Technické v Brně

25/04/2017

1 Zadanie

Cieľom tohto projektu je implementovať algoritmus *Mesh Multiplication* za pomoci knihovny *Open MPI*. Program je riadený testovacím skriptom *test.sh* ktorý preloží strojový súbor *mm.cpp*, vypočíta počet potrebných procesorov a následne volá program *mm*.

Program neprijíma žiadne argumenty, pracuje výlučne s dvoma textovými súbormi *mat1* a *mat2* umiestnenými v rovnakom priečinku. Program bol implementovaný podľa vzoru pôvodného algoritmu *Mesh multiplication* preberaný na prednáškach predmetu *PRL* a nebol nijak modifikovaný.

2 Algoritmus

Mesh multiplication je paralelný algoritmus ktorý vykonáva operáciu násobenia nad dvoma maticami A a B . Pri svojom výpočte používa rovnaký počet procesorov ako počet prvkov vo výslednej matici C .

Rozmer výslednej matice C sa odvíja od rozmerov vstupných matíc a je definovaný ako $A_{rows} \times B_{columns}$. Každý procesor P_i udržiava v svojich registroch 3 hodnoty:

- **a** - príslušný prvok matice A
- **b** - príslušný prvok matice B
- **c** - produkt násobenia prvkov a a b

2.1 Princíp

Listing 1: Algoritmus mesh multiplication.

```
1 // mesh multiplication
2
3 void mesh_multiplication(matrix A, matrix B, matrix C):
4     for i=1 to m do in parallel
5         for j=1 to k do in parallel
6             cij = 0
7             for s=1 to n do
8                 cij += ais * bsj
```

2.2 Analýza

Zložitosť tohto algoritmu je kubická $\Theta(n^3)$, čo z neho robí neoptimálny algoritmus.

Optimálny algoritmus pre násobenie dvoch matíc síce zatiaľ nie je známy, ale vieme že určite sa nachádza v intervale:

$$(\Theta(n^2), \Theta(n^3))$$

3 Implementácia

Po načítaní vstupných matíc a zistení ich oboch rozmerov sa pripraví výsledná matica príslušných rozmerov podľa definície. Následne sa v okrajových procesoroch v hornom riadku a ľavom stĺpci inicializujú vstupné matice aby bolo možné propagovať hodnoty a, b cez mriežku procesorov. Hodnoty matice mat1 (a) smerom doprava horizontálne a hodnoty matice mat2 (b) smerom dole vertikálne. Pri tomto posúvaní hodnôt sa zároveň v každom procesore, ktorý už má obe hodnoty počíta produkt týchto dvoch hodnôt. Nakoniec procesor ID 0, ktorý je v ľavom hornom rohu naplní výslednú maticu z hodnôt c, ktoré prijme od všetkých ostatných procesorov.

4 Výsledok

Implementovaný program pracuje správne podľa zadania. Pre meranie výpočetného času behu programu nad rôznym počtom vstupných hodnôt som použil bash funkciu *time*. Priemerné hodnoty sú uvedené v tabuľke 1.

Rozmer matice A	Rozmer matice B	Priemerný systémový čas [s]
3 x 2	2 x 4	0.252
5 x 3	4 x 4	0.340
6 x 3	3 x 9	1.004
10 x 3	3 x 10	3.028
15 x 3	15 x 5	limit exceeded

Table 1: Tabuľka priemerných časových hodnôt

5 Záver

Z tabulky je očividné, že zo vzrastajúcimi rozmermi vstupných matíc, rastie aj čas a to veľmi rýchlo. To odpovedá teoretickej časovej zložitosti algoritmu. Ostáva len konštatovať, že pre štúdijné účely bol algoritmus úspešne implementovaný.