

Projekt č. 2 (OpenMP a MPI)

Konvolučný filter Obrazu

I. Popis riešenia

- a. Inicializácia MPI prostredia
- b. Načítanie obrázka so súboru do poľa – len v root MPI procese
 - Obrázok musí byť vo formáte .bmp
 - Podmienkou je aby jeho rozmery boli deliteľné 4
- c. Zdieľanie rozmerov obrázka a src_size medzi MPI procesmi
 - Implementované ako MPI_Bcast výšky, šírky a src_proc
 - src_proc je rozmer poľa, ktoré príme každé vlákno a rovná sa výška/počet OMP procesov
- d. Rozdelenie obrázka=poľa medzi MPI procesy
 - MPI_Scatter pošle každému časť poľa veľkosti src_proc
- e. Každý OMP proces filtruje svoju časť
 - Potreba prechádzať každý pixel danej časti obrázka – využitie OMP, paralelizácia for cyklu do viacerých vlákien
 - Používa konvolučný kernel 5x5
- f. Spojenie obrázka=poľa do root MPI procesu
 - MPI_Gather spojí časti veľkosti src_proc do jedného poľa
- g. Uloženie obrázka do súboru – len v root MPI procese

II. Experimenty pre rôzne konfigurácie paralelného systému

a. Jeden uzol

| <i>časy s rôznym počtom OMP vlákien*</i> | | |
|--|------------------|---------------------|
| Count of threads | convolution time | convolution speedup |
| 1 | 0.113 | 0% |
| 2 | 0.101 | 11% |
| 4 | 0.061 | 46% |
| 8 | 0.058 | 49% |
| 12 | 0.065 | 42% |
| 16 | 0.067 | 41% |

*spustený 1 MPI proces

Z danej tabuľky vyplýva, že najväčšie zrýchlenie prináša použitie 8 OMP vlákien.

| <i>časy s rôznym počtom MPI procesov (1 uzol)*</i> | | | | |
|---|-------|-------|-------|--------|
| Count of processes | 1 | 2 | 4 | 8 |
| MPI initiation time | 0.03 | 0.094 | 0.208 | 0.556 |
| <i>init speedup</i> | 0% | -213% | -593% | -1753% |
| Picture loading time | 0.002 | 0.001 | 0.001 | 0.002 |
| Picture convolution time + collective communication routines | 0.114 | 0.069 | 0.063 | 0.075 |
| <i>convolution speedup</i> | 0% | 39% | 45% | 34% |
| Picture saving time | 0.002 | 0.002 | 0.002 | 0.002 |
| Total execution time | 4.18 | 5.085 | 6.923 | 10.912 |

* spustené 1 OMP vlákno

Celkový čas behu programu rastie so zvyšujúcim sa počtom MPI procesov. Ale pri meraní času samotnej konvolúcie sa časy zlepšujú. Takže pri kovolúcii veľkého obrázka sa paralelizáciu oplatí použiť. Najväčšie zrýchlenie prináša použitie 4 MPI procesov.

b. Dva uzly

| <i>časy s rôznym počtom MPI procesov (2 uzly)*</i> | | | | |
|---|-------|-------|--------|--------|
| Uzol_1 + Uzol_2 | 1+1 | 2+2 | 1+3 | 3+1 |
| MPI initiation time | 0.328 | 0.848 | 4.388 | 1.671 |
| Picture loading time | 0.016 | 0 | 0 | 0.016 |
| Picture convolution time + collective communication routines | 0.469 | 0.484 | 0.709 | 0.868 |
| Picture saving time | 0 | 0 | 0 | 0 |
| Total execution time | 6.205 | 9.39 | 11.371 | 11.449 |

*spustených 8 OMP vlákien

V tomto prípade veľa času trvali kolektívne operácie - prenos informácie po sieti. Takže čím viac uzlov a procesov, tým horší čas.

III. Zhodnotenie

Riešenie využíva oba paralelné programátorské modely, aj OpenMP aj MPI a je schopné pracovať na viacerých uzloch.

Testovanie ukázalo, že paralelizácia zrýchľuje samotnú konvolúciu.

Avšak pri malom rozmere obrázka čas potrebný na réžiu paralelizácie prevyšuje čas na konvolúciu, .