Osztott rendszerek specifikációja és implementációja - dokumentáció

Lombosi Balázs - D3BA80

2016. december 4.

Tartalomjegyzék

1. Feladat

Az egyik bemenet –input_matrices.txt – első sorában tartalmazza az M pozitív egész számot, következő $4xM$ sorában pedig az M db $4x4$es mátrix sorfolytonos reprezentációját.
M
$x_{1,1,1}$ $x_{1,1,2}$ $x_{1,1,3}$ $x_{1,1,4}$ - az 1. mátrix elemei
$x_{1,2,1}$ $x_{1,2,2}$ $x_{1,2,3}$ $x_{1,2,4}$ - ...
$x_{1,3,1}$ $x_{1,3,2}$ $x_{1,3,3}$ $x_{1,3,4}$ - ...
$x_{1,4,1}$ $x_{1,4,2}$ $x_{1,4,3}$ $x_{1,4,4}$ - az 1. mátrix elemei
...
$x_{M,4,1}$ $x_{M,4,2}$ $x_{M,4,3}$ $x_{M,4,4}$ - az M. mátrix utolsó sorának elemei

A másik bemenet –input_points.txt – első sorában tartalmazza az N pozitív egész számot, a következő N sorában pedig az N db 3 hosszú vektorokat. N
$v_{1,1}$ $v_{1,2}$ $v_{1,3}$ - az 1. vektor elemei
...
$v_{N,1}$ $v_{N,2}$ $v_{N,3}$ - az N. vektor elemei

A program olvassa be az adatokat, majd végezze el minden vektorra az M db mátrix szorzást. Az az eredményt írja az – output.txt – kimeneti fájlba. Megkötés, hogy a fájlból olvasást és a fájlba írást egy szál/folyamat végezze, az egyes mátrix szorzások eredményét pedig egy szál továbbítsa adatcsatornán keresztül egy rákövetkező szálnak.

2. Felhasználói dokumentáció

2.1. Környezet

A program több platformon futtatható, nincsen dinamikus függősége. Telepítésre nincs szükség, elegendő a futtatható állományt elhelyezni a számítógépen (Windows operációs rendszeren *.exe* fájl, UNIX operációs rendszeren *.out* fájl).

2.2. Használat

A program használata egyszerű, mert nem vár parancssori paramétereket, így akár parancssoron kívül is lehet futtatni. A fájl mellett kell elhelyezni az *input_matrices.txt* és *input_points.txt* fájlokat melyet a program beolvas, az eredményt pedig az *output.txt* nevű fájlba írja.
A program futásának feltétele, hogy a bemeneti fájlok *(input_matrices.txt,input_points.txt)* felépítése helyes legyen, azaz a feladatban leírtak szerint legyenek benne az adatok.

3. Fejlesztői dokumentáció

A feladat megoldásához a C++11 es szabványt választottam és annak beépített nyelvi elemeit.

3.1. A megoldás módja

A kódot logikailag több részre bonthatjuk, egy fő- és több alfolyamatra. A főfolyamat, a $main()$ függvény lesz felelős az adatok beolvasásáért az input fájlokból, amik most az *(input_matrices.txt és input_points.txt)*. A főfolyamat beolvassa a mátrixokat amikkel a műveleteket fogja elvégezni,majd létrehoz $M+1$ db adatcsatornát és M db szál, az M db szál mindegyike kap egy-egy mátrixot és azzal a mátrixal fogja elvégezni a szorzást majd továbbítja az eredményt a következő szálnak. A *vektorok* beolvasásakor az első adatcsatornát (*pipeline*)-t töltjük fel amivel beolvasás közbe már az első szál eltud kezdeni számolni.

3.2. Implementáció

A párhuzamosságot a C++11 szabvány nyelvi elemeit kihasználva valósítjuk meg. Az eredmény helyben, a beolvasott vektorban lesz.

A C++ beépített adatszerkezeteiből az adatok tárolásához az $std::vector$-t. A beolvasott mátrixokat egy $std::vector<std::vector<std::vector<int>>>$ adatszerkezetben tároljuk. A főfolyamat létrehoz $M+1$ db adatcsatornát $std::vector<Pipeline<Vector>>$ $pipelines(M+1)$ és M db $threadet$ indít az $Operation$ függvénnyel ami paraméterül kapja az i és $i+1$ -ik adatcsatornát, a threadeket szintén egy $std::vector$ban tároljuk. Ez után a főfolyamat beolvassa a vektorokat a $pipelines[0]$ adatcsatornába.

Az $Operation$ függvény fut az M db szálon, ami induláskor paraméterül kapja a *Mátrixot* amivel a mátrix szorzást végzi, két adatcsatornát $(from, to)$, amelyikről olvassa$(from)$ a vektort amivel el kell végeznie a műveletet, és amelyikre az eredményt kell írnia (to), és az N számot, hogy hányszor kell elvégeznie a műveletet azaz hány vektor fog érkezni, a $from$ adatcsatornáról.

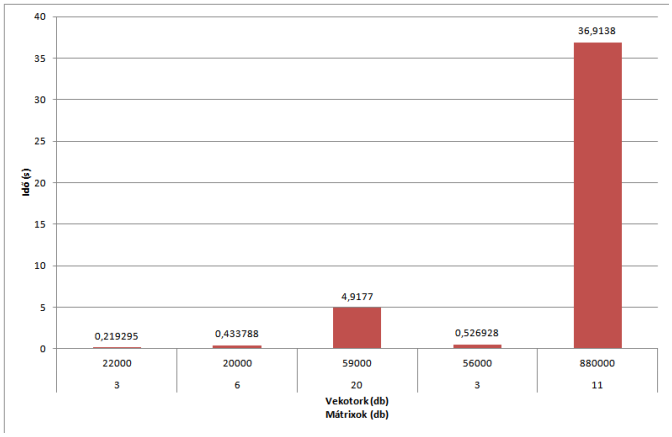
3.3. Visszavezetés

A feladatot vissza vezethetjük az adatcsatorna tételére.

T		F
D	$\leftarrow$	$< v_1 \dots v_N >$
$F(m, v)$	$\leftarrow$	$multiplication(m, v)$
X_i	$\leftarrow$	*változatlan*

$$\forall i \in [0..M]\ f_i(m, v) = (multiplication(m, v))$$

3.3.1. Mérések



3.4. Fordítás

A program forráskódja a *main.cpp* fájlban található. A program fordításához követelmény egy C++11 szabványt támogató fordítóprogram megléte a rendszeren. A legnépszerűbbek az *msvc*, *g++* és *clang*. A fordítás menete (UNIXon g++ 4.9.2-es verziójú fordítóval, Windowson g++ 5.1.0 verziója fordítóval lett tesztelve):

- UNIX környezetben a következő: $'g++ -std = c++11\ main.cpp -lpthread'$,
- Windows környezetben: $'g++ -std = c++11\ main.cpp'$.

Az $-std = c++11$ kapcsoló szükséges, mert alapértelmezetten egy régebbi C++ szabványt támogat a fordító. UNIX környezetben szükséges a -pthread vagy -lpthread kapcsoló, hogy a fordító tudja, hogy többszálú programot fordít.

3.5. Tesztelés

A programot a megadott teszt input fájlokkal teszteltem le. A teszteléshez használt processzor: Intel Core i7-3632QM @ 2.20Ghz .