

Tema 1 – Conway’s Game of life

Responsabil temă: Cristian Chilipirea

1. Enunț **Deadline: 2014-Noiembrie-02 23:59**

Tema constă în simularea în paralel a jocului ”Game of Life” - B3/S23.

Jocul presupune existența unei hărți sub formă de **matrice** de mărimea **W x H**.

Fiecare element din matrice are 2 valori:

a. *Individ* – valoarea **1**

b. *Gol* – valoarea **0**

Jocul are un număr dat de etape, în fiecare etapă fiecare element din matrice își schimbă valoarea după setul de reguli B3/S23 (Se generează un nou individ, dacă are 3 vecini; Un individ supraviețuiește dacă are 2 sau 3 vecini). Pe larg:

a. *Un individ nou este creat dacă are 3 vecini*

b. *Un individ continuă să existe dacă are 2 sau 3 vecini*

c. *Orice individ cu 1, 0 sau mai mult de 3 vecini dispare*

Vecinii unui element A sunt cei colorați cu verde mai jos.

Există 2 metode de a reprezenta harta:

1. Ca un plan

0	0	0	0	0	0
0	A	B	C	D	0
0	E	F	G	H	0
0	I	J	K	L	0
0	0	0	0	0	0

2. Ca un toroid

L	I	J	K	L	I
D	A	B	C	D	A
H	E	F	G	H	E
L	I	J	K	L	I
D	A	B	C	D	A

La un toroid prima linie/coloană e lipită de ultima. Zona gri de mai sus reprezintă întreaga zona SIMULATA. Zona simulată nu va avea o margine.

Toate elementele matricei se vor schimba simultan în funcție doar de etapa anterioară.

2. Detalii implementare

Va trebui să:

1. Implementați un program secvențial care să citească un fișier cu harta, să simuleze până la a N-a etapă și să o salveze în alt fișier.
2. Paralelizați programul secvențial folosind OpenMP.
3. Testați programul cu număr diferit de thread-uri și scrieți rezultatele în README alături de restul de informații precum descrierea implementării.

Fișierul de intrare

Pe prima linie vom avea

- O literă **P** (plan) sau **T** (toroid) ce indică forma hărții.
- Un număr întreg **W_hartă** – lățimea hărții
- Un număr întreg **H_hartă** – lungimea hărții
- Un număr întreg **W** – lățimea hărții ce trebuie simulată
- Un număr întreg **H** – lungimea hărții ce trebuie simulată

Aceasta va fi urmată de **H_hartă** linii cu **W_hartă** valori de 1 (indivd) sau 0 (gol) ce reprezintă o bucată din hartă începând de la coordonatele (0, 0) în simulare.

Exemplu:

```
T 3 3 100 100
0 1 0
0 0 1
1 1 1
```

ATENȚIE: mărimea hărții din fișier poate să nu fie egală cu mărimea hărții ce va trebui să o simulați.

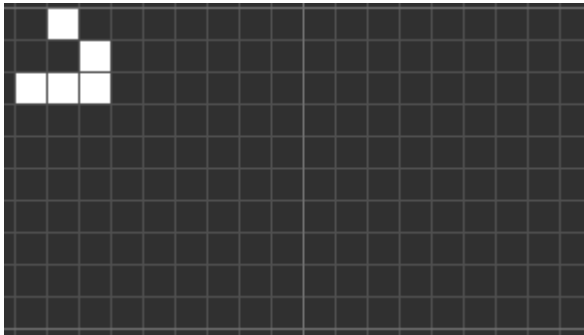
H și W nu se schimbă între fișierele de intrare și ieșire.

În fișierul de ieșire $W_hartă \leq W$ și $H_hartă \leq H$

3. Golly

Pentru a vă verifica tema puteți folosi utilitarul [golly](#).

Exemplul de mai sus va arăta astfel în golly:



Pentru a simula până la un anumit pas:

1. Control - > Set base step
2. Control - > Faster/Slower
3. Repetă până ce valoarea step este cea dorită
4. Apăsați space sau tab

Tipul de simulare este QuickLife

Puteți copia codul de mai jos in Golly ctrl+c ctrl+v

x = 3, y = 3, rule = B3/S23:T100,100

bo\$2bo\$3o!

În arhiva temei există 2 executabile ce le puteți rula pe fep să convertiți din standardul golly în formatul temei și invers. Le puteți apela

`./txtToRle nume_fișier.txt nume_fișier.rle`

`./rleToTxt nume_fișier.rle nume_fișier.txt`

4. Testare

Există 5 teste din care 2 publice. Fiecare test valorează 20 de puncte. Din care:

- 20% - varianta paralelă funcțională (rezultatele dau corect)
- 60% - variantă paralelă scalabilă $S > 1.1$ (doar dacă primiți tot punctajul la **a**)
- 20% - variantă paralelă scalabilă eficient $S > 1.7$ (doar dacă primiți tot punctajul la **a**)

$$S = \frac{\frac{Tp1}{Tp2} + \frac{Tp2}{Tp4} + \frac{Tp4}{Tp8}}{3}$$

Unde Tp_i = Timp execuție i thread-uri

Fișiere arhiva temei:

- `rlToTxt` – convertește între format Golly și cel al temei
- `txtToRle` – convertește între format temei și cel Golly
- `test1.txt` – primul test
- `test1.ref.txt` – rezultatele primului test după 1000 iterații
- `test2.txt` – al doilea test
- `test2.reg.txt` – rezultatele celui de-al doilea test după 1000 iterații

5. Upload temă

Tema se va pune pe [vmChecker](#).

Tema va fi o arhivă `.zip` ce conține fișierul(sau fișierele) `.c`, `README`, alături de un `Makefile` a cărui regulă principală va compila tema cu numele `tema1`.

Apel: `./tema1 num_threads num_steps input_file output_file`

Tema se va testa în prealabil pe cluster pe coada ***ibm-quad.q***

qsub -cwd -q ibm-quad.q script.sh

Unde `script.sh` pornește aplicația voastră. Exemplu:

```
#!/bin/bash
```

```
./tema1 2 1000 test1.txt test1.out.txt
```