

Plan de Lucru (Conway's Game Of Life)

Enunt

Game of Life, cunoscut și sub numele de Life, este un automat celular conceput de matematicianul britanic John Horton Conway în 1970. Este un joc cu zero jucători, ceea ce înseamnă că evoluția sa este determinată de starea sa inițială, care nu necesită intrări suplimentare.

Se interacționează cu Jocul Vieții prin crearea unei configurații inițiale și observarea modului în care evoluează.

Variante

Varianta secvențială va fi creată folosind C#.

Variantele paralele vor folosi: Fire de execuție și MPI.

Pentru firele de execuție voi încerca să împart liniile și/sau coloanele pe anumite thread-uri pentru a mări viteza simulării.

Pentru MPI va trebui să învăț.

Informatii platforma

- Windows 10
- Procesor: Ryzen 5

Numar nuclee	6
Numar thread-uri	12
Frecventa	3,7 GHz
Frecventa turbo	4,6 GHz
Cache level 2	3 MB
Cache level 3	32 MB

- RAM: 16GB
- Placa video: (nu este utilizată) NVidia RTX 3060ti

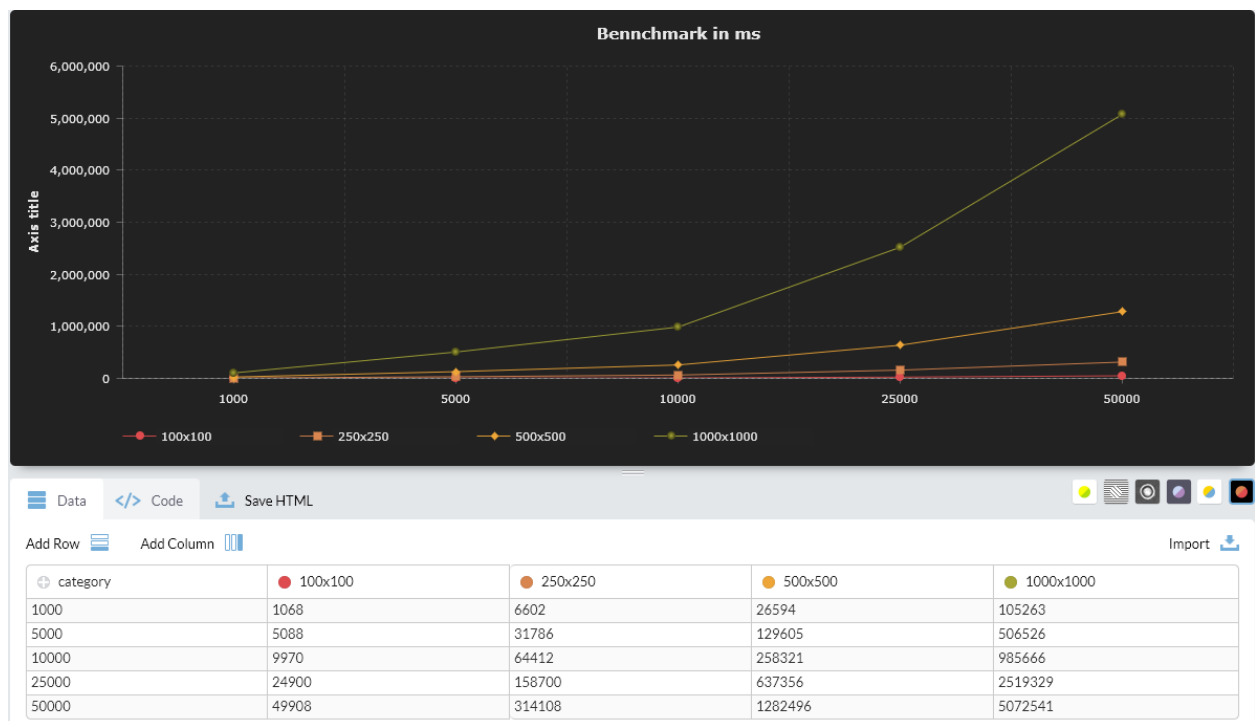
GITHUB

<https://github.com/alexpopescu021/Conway.git>

Benchmark pentru varianta secvențială

<https://live.amcharts.com/5MDkx/>

SAU

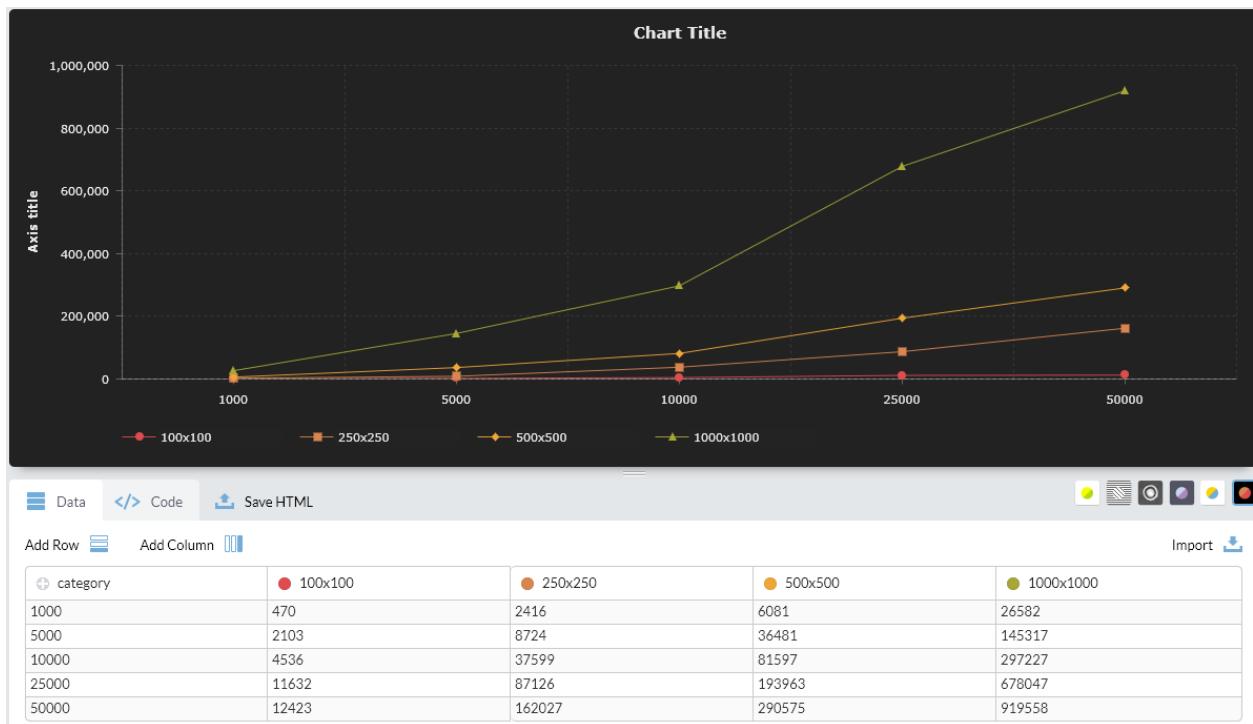


Cel mai mare input a durat 84 minute. Se poate observa o creștere semnificativă a timpului de execuție atât când creștem numărul de iterații dar și când creștem dimensiunile planșei de joc.

Se poate observa că timpul este mai afectat de dublarea dimensiunii matricii decât de dublarea numărului de iterații.

Benchmark pentru varianta paralelizata

<https://live.amcharts.com/lkODd/>



Se poate observa ca cel mai mare input, care a durat 83 de minute pe varianta secvențială, a fost redus la aproximativ 15 minute.

Varianta paralelizata folosește un `Parallel.For` pentru a înlocui un nested for din funcția `UpdateState()`, astfel fiecare proces v-a lucra pe o alta linie din matrice simultan.

Referințe gasite momentan

<https://www.geeksforgeeks.org/program-for-conways-game-of-life/>

<http://www.jeremybytes.com/Downloads.aspx#ConwayTDD>

https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

https://www.youtube.com/watch?v=FWSR_7kZuYg&ab_channel=TheCodingTrain

<https://stackoverflow.com/questions/12405938/save-time-with-parallel-for-loop>

<https://devblogs.microsoft.com/pfxteam/is-it-ok-to-use-nested-parallel-for-loops/>

<https://en.delphipraxis.net/topic/1296-nested-parallel-for-loops-bad-idea/>

```
2 references
public void UpdateState()
{
    Parallel.For(0, gridHeight, i =>
    {
        for (int j = 0; j < gridWidth; j++)
        {
            var liveNeighbors = GetLiveNeighbors(i, j);
            nextState[i, j] =
                LifeRules.GetNewState(CurrentState[i, j], liveNeighbors);
        }
    });

    CurrentState = nextState;
    nextState = new CellState[gridHeight, gridWidth];
}
```