

# [07B]- Implementace CA pomocí Python + MPI

Milan Munzar

Fakulta informačních technologií VUT v Brně

10. května 2014

# Zadání projektu

- Implementovat Conwayovu hru Life a alespoň jeden problém ve vícestavovém CA.
- Povolené prostředky: Python + mpi4py.
- Provést měření nad různými stupni paralelizace.
- Porovnat se sekvenční verzí.
- Vyhodnotit výsledky.

# Conwayova hra Life

- dvourozměrný uniformní celulární automat
- buňka může být ve stavu živá/mrtvá
- pravidla hry:
  - živá buňka s méně než dvěma živými sousedy zemře
  - živá buňka s více než třemi živými sousedy zemře
  - mrtvá buňka se třemi živými sousedy ožije

# Hra Life pseudokód

---

## Algoritmus 1 sequential Life

---

```
1: curr_epoch  $\leftarrow$  0
2: while curr_epoch < no_epoch do
3:   for x = 1 to matrix_size - 1 do
4:     for y = 1 to matrix_size - 1 do
5:       sum  $\leftarrow$  noNeighbor(x, y, matrix1)
6:       matrix2[x][y]  $\leftarrow$  applyRules(sum, matrix1[x][y])
7:     end for
8:   end for
9:   swap(matrix1, matrix2)
10:  curr_epoch  $\leftarrow$  curr_epoch + 1
11: end while
```

---

# Okrajové podmínky

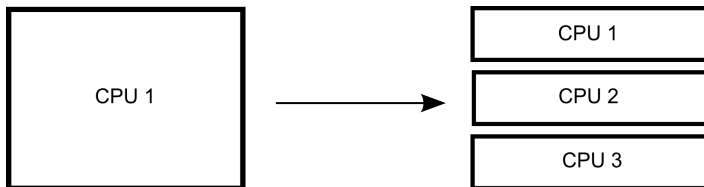
- Okraji matice jsou napevno přiděleny mrtvé buňky.
- Jiná možnost je svázat okraje.

# Použitá optimalizace

- Platí, že pokud buňka ani žádný její soused nezměnil v minulém kroku stav, pak v tomto kroku buňka svůj stav měnit nebude.
- Přidána matice aktivních buňek.

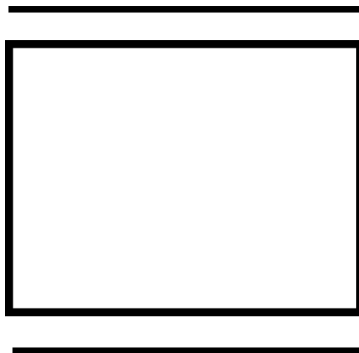
# Paralelizace hry Life

## ■ MPI\_scatter



# Počítání na hranách matic

- Získání součtu z osmiokolí.
- Sousedé si vyměňují mezisoučty na hranách.





# Paralelní hra Life pseudokód

---

## Algoritmus 2 parallel Life

---

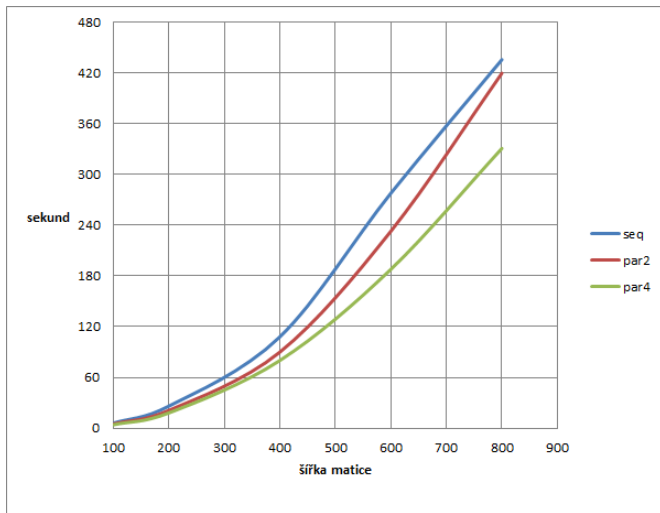
```
1: curr_epoch  $\leftarrow$  0
2: while curr_epoch < no_epoch do
3:   request  $\leftarrow$  MPI_Irecv(borders)
4:   borders  $\leftarrow$  computeBorders(matrix1)
5:   MPI_Isend(borders)
6:   matrix2  $\leftarrow$  computeInnerMatrix(matrix1)
7:   MPI_Wait(request)
8:   updateBorders(borders, matrix2)
9:   computeCells(matrix1, matrix2)
10:  swapPointers(matrix1, matrix2)
11:  curr_epoch  $\leftarrow$  curr_epoch + 1
12: end while
```

---

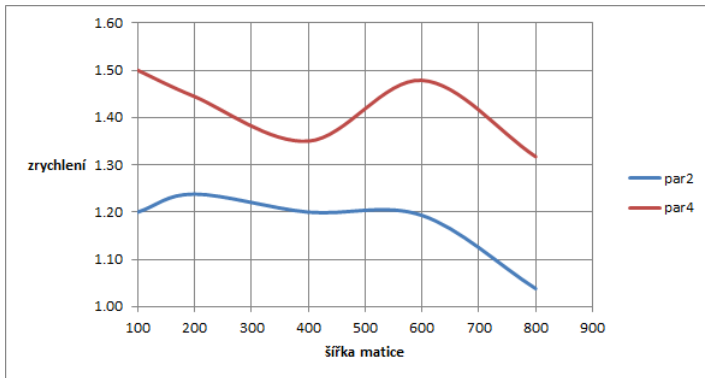
# Experimenty

Testováno na stroji s Intel Core(TM) i5-3210M CPU @ 2.50GHz a 8GB operační paměti.

# Doba výpočtu při různých stupních paralelizace



# Dosažené zrychlení



# Použité prostředky

- Python 2.7
  - Tkinter, threading, Queue
  - numpy
  - mpi4py
  - timeit