[07B]- Implementace CA pomocí Python + MPI

Milan Munzar
xmunza00@stud.fit.vutbr.cz

1 Úvod

Tento text popisuje paralelní řešení celulárního automatu realizující hru Life. Program je implementován v jazyce Python s využitím knihovny `mpi4py`. V sekci 4 se nachází porovnání doby běhu programu s různým stupněm paralelizace a sekvenční verze.

2 Conwayova hra Life

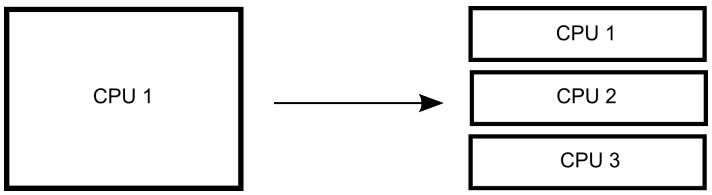
Conwayova hra Life je uniformní dvourozměrný celulární automat. Mřížka hry obsahuje buňky, které mohou být ve stavu živá/mrtvá. Stav buňky závisí na jejím osmiokolím. Pravidla hry jsou následující: živá buňka s méně než dvěma živými sousedy zemře, živá buňka s více než třemi živými sousedy zemře a mrtvá buňka se třemi živými sousedy ožije

Výpočet mřížky jsem urychlil sledováním aktivity buněk. Pokud buňka a její sousedé zůstali v minulém kroku výpočtu mřížky nezměněni, pak ani v tomto kroku se stav buňky nemění. Jako okrajovou podmínku jsem použil pevné přiřazení stavu mrtvá krajním buňkám.

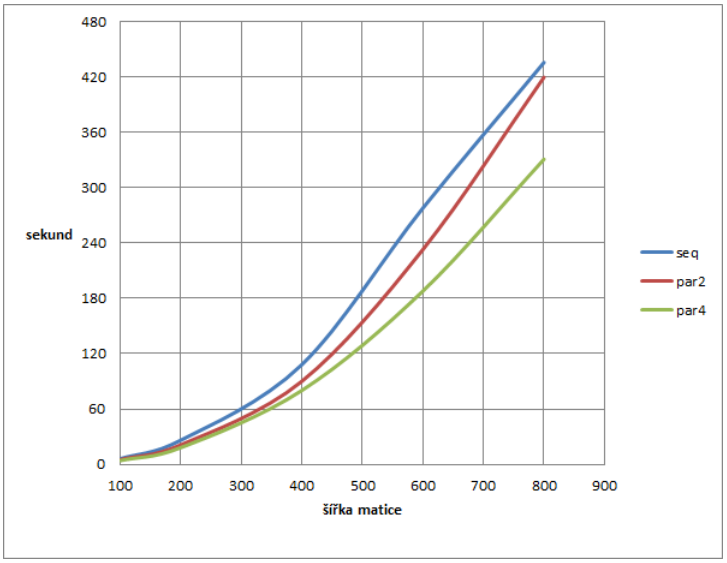
3 Paralelizace Conwayovy hry Life

Výpočet mřížky mezi procesory jsem rozdělil způsobem zobrazeným na obrázku 1. Každý proces dostane *radku_matice/pocet_procesoru* řádků a nad nimi poté provádí výpočet. Tato operace je zajištěna funkcí `scatter()`. Sběr výsledků se prováde funkcí `gather()`.

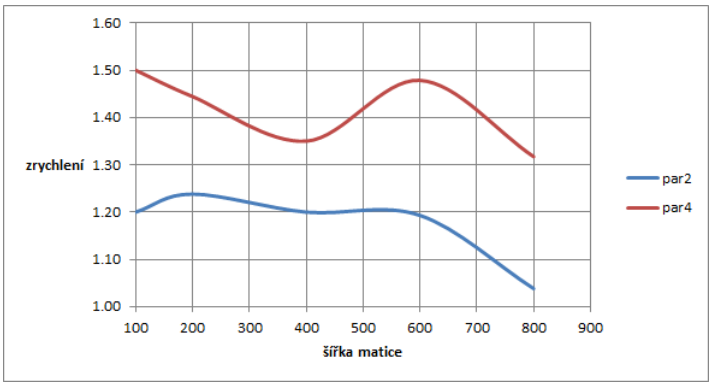
Aby byl výsledek výpočtu validní musí si procesory vyměňovat informace o počtu živých buněk na sousedicích hranách. K tomu jsem použil dodatečné pole reprezentující mezisoučet na hranách matice. Toto pole si procesy vymění pomocí neblokujících operací `Isend()` a `Irecv()`. Použití neblokující komunikace mi dovoluje překrýt výpočet vnitřní matice s výměnou hraničních polí. Po výpočtu vnitřní matice se dokončí výpočet hran (funkcí `Wait()` čekám na dokončení komunikace) a uzavře se jedna iterace algoritmu.



Obrázek 1: Paralelizace výpočtu mřížky



Obrázek 2: Doba běhu programu při různých stupních paralelizace



Obrázek 3: Dosažená zrychlení

4 Zhodnocení paralelní verzí programu

Měření byla prováděna na stroji s dvoujádrovým procesorem `Intel Core(TM) i5-3210M CPU @ 2.50GHz` a 8GB operační paměti. Porovnání doby běhu programu paralelních a sekvenční verzí je vidět na obrázku 2. Dosažená zrychlení jsou vidět na obr. 3. Hodnoty vynesené v grafu jsou brány jako průměrná hodnota ze 3 běhů programu nad aktuálním nastavení mřížky. Počet epoch byl ve všech měřeních 100.

5 Závěr

Podařilo se mi naimplementovat sekvenční i paralelní variantu Conwayovy hry Life. K dodaným skriptům jsem vytvořil jenoduché GUI. Bohužel dosažená zrychlení nejsou veliká. Chybou může být neefektivní implementace a nevhodnost použitého stroje pro měření paralelních programů. Zadání jsem splnil pouze částečně, protože jsem nestihl implementovat alespoň jeden problém vícestavovém prostoru.

Appendix

- Příklad spuštění paralelního programu se třemi procesy:

```
mpiexec -n 3 python prl_life.py
```