

# Simulacija života umjetnih organizama uz CUDA (grafičko šta već da napišemo)

Dominik Brdar

bacc.ing.rac. Fakultet elektrotehnike i računarstva  
dominik.brdar@fer.hr

Matija Holik

bacc.ing.rac. Fakultet elektrotehnike i računarstva  
matija.holik@fer.hr

Ivan Žugaj

bacc.ing.rac. Fakultet elektrotehnike i računarstva  
ivan.zugaj@fer.hr

January 14, 2022

# 1 Uvod

Izvođenje složenih izračuna s velikim brojem podataka kao što je simulacija međudjelovanja populacija mikroorganizama različitih svojstava zahtjeva iznimno puno računalnih resursa. Grafička procesorska jedinica zbog svojstvene sklopovske arhitekture, koja omogućuje paralelno računanje složenih operacija nad velikim skupom podataka, može se koristiti u svrhu ubrzavanja izvođenja procesa takvih zadataka. Za programiranje grafičke kartice s ciljem kvalitetnije simulacije interakcija organizama u sklopu ovog seminara, odabrana je CUDA platforma.

? koje ćemo simulirati u tekući sa Ph brijednoću jednakoj određenoj broju različitih tipova organizama.

Key words: simulation, paralelism, CUDA, CPU, arrays, PyCuda, PyCharm

## 2 Motivacija

Jedna od klasičnih prikaza molekula i mikroorganizama je prikaz računalnom simulacijom. Proučavanje i liječenje bolesti za koje nije dovoljno promatrati živote organizama uz pomoć mikroskopa nebi bilo moguće bez takvih simulacija. Mnoge tvrtke, poduzeća i istraživački centri značajno doprinose razvoju i kvaliteti lijekova koristeći računalne simulacije. Kretanje i ponašanje velikog broja molekula i organizama iziskuje značajnu računalnu moć budući da je potrebno obrađivati stotine tisuća objekata koji se prikazuju u simulaciji. Osim njihovog postojanja, simuliraju se i brojne interakcije između samih organizama ili molekula. U takvom sustavu, elementi imaju lokalne međuovisnosti, ali i neke globalne uvijete, te se za ovakve zadatke prikazuje da korištenje grafičke procesorske jedinice može znatno ubrzati proces simulacije. Nvidia CUDA nudi pristupačno programsko okruženje za programiranje i korištenje grafičke kartice i njenog sklopovlja za paralelizaciju neovisnih računanja i obradu velikog skupa podataka istovremeno. Proizvođač Nvidia, koji je sam i razvijao svoje grafičke kartice, njihovu arhitekturu i programe na najnižoj razini, najbolje može ostvariti i programsku potporu pomoću koje korisnici njihovih grafičkih kartica mogu svoje programe optimirati tako da za ključne dijelove implementiraju izvođenje na grafičkoj procesnoj jedinici. Na taj način, programer ne treba razumjeti svoju grafičku karticu na niskoj razini, već koristi CUDA okruženje koje

jamči najbolje iskorištavanje dostupnih resursa. S druge strane, takvi programi su optimirani samo za korisnike Nvidia grafičkih kartica, dok grafičke kartice drugih proizvođača imaju drugačija arhitekturna rješenja i neće moći interpretirati kod pisan pomoću CUDA platforme. Iako je prenosivost mana, CUDA je ipak vrlo zastupljena u raznim područjima interesa kao što je strojno učenje ili block-chain tehnologije zbog nenadmašive preciznosti u smislu apsolutnog i optimalnog iskorištavanja grafičke kartice.

### 3 Cilj rada

Produkt ovog rada je simulacija suživota 4 vrste istog zamišljenog organizma nazvanog Quid na određenoj površini unutar vode pri neutralnoj pH vrijednosti (7). Aktivnost tih organizama mijenja pH vrijednosti tekućine, a uz to se organizmi unutar promatranog područja gibaju, rastu i među njima se događaju razne interakcije koje dovode do promijena njihovih stanja. Za potrebe ilustracije učinkovitosti različitih programskih rješenja prikazana je i razlika izvršavanja simulacije korištenjem samo procesora i radne memorije, te uz pomoć grafičke kartice. Pritom je cilj rada dobiti uvid u mogućnosti programiranja grafičke kartice za razne primjene, a ne samo za iscertavanje. Ideja je da konačan ishod rada bude dodatna motivacija za nastavak proučavanja i uvođenja paralelizma u razne procese.

### 4 Opis programskog rješenja

Prije početka simulacije korisnik odabire broj iteracija koje će se izvesti i koliko maksimalno quidova simulacija može generirati prije završetka. Potom odabire za svaku vrstu quidova koliko će ih se generirati na početku, veličinu područja koje se promatra i početnu temperaturu koja utječe na gibanje quidova u tom području. Nakon unosa parametara pokreće se simulacija. Svaki od quidova tijekom simulacije kreće se u različitim smjerovima koje su nasumične u x i y osi ovisno o inicijalno postavljenoj temperaturi sustava čime dobivamo kaotičnost sustava.

QUID - Quidovi su podjeljeni na 4 različita tipa:

1. tip je crvene boje sa pH vrijednošću 3.

2. tip je žute boje sa pH vrijednošću 5.
3. tip je zelene boje sa pH vrijednošću 9.
4. tip je plave boje sa pH vrijednošću 12.

Interakcije između različitih tipova opisane su u tablici 1.

Vrste interakcija: X (množe se), - (ubiju se), 0 (nema učinka). Tipovi quidova se prikazuju brojevima (1, 2, 3, 4)

-	1	2	3	4
1	x	-	0	-
2	-	x	-	0
3	0	-	x	-
4	-	0	-	x

Table 1: primjer matrice interakcija

## 4.1 Tijek simulacije

Na ekranu se pojavljuje simulacija uz mogućnosti pauziranja, prekida simulacije i mogućnosti mijenjanja perioda svake iteracije radi lakšeg praćenja promijena i temperature u našem sustavu.

Ispisuje se broj Quidova u simulaciji, kao i pH svakog kvadranta i cijelog sustava. Slika 3.1 prikazuje simulaciju u trenutku izvođenja.

Simulacija se prekida u nekoliko slučajeva:

- kraj broja iteracija koje smo unijeli,
- nepostojanje quidova unutar simulacije budući da se quidovi kreću te prilikom izlaska izvan područja promatranja prestanu postojati,
- prekoračenje maksimalnog broja quidova koje smo unijeli.

## 4.2 Promjene stanja quidova i izračun pH vrijednosti

Reakcija quidova događa se ovisno o njihovim tipovima kao što prikazuje tablica 1. Quidovi međusobno moraju biti na udaljenosti manjoj od 5 kako bi se interackija između njih dogodila.

Kako bi saznali udaljenost izmmeđu svakog od quidova, računamo preko obične liste. Prvi quid unutar liste iterira po ostatku liste i traži quid sa najmanjoj mogućom udaljenosti Potom pri pronalasku najbližeg susjeda stvaramo objekt tipa Neighbours koji sadrži dva quida i njihovu udaljenost. Izvođenje opisanog procesa ima vremensku složenost  $O(n^2)$ .

Nakon izračuna dolazi do iteracije po stogu koji potom ovisno o vrsti quidova izaziva interakcije. Nakon iteracija, prolazom po listi quidova u kompleksnosti  $O(n)$ , ovisno o lokaciji quidova, računa se pH vrijednost svakog kvadranta.

## 5 Osvrt na rezultat

## 6 Zaključak