



Бургаски Държавен Университет "Проф. д-р Асен
Златаров"

Факултет по технически науки
Катедра "Компютърни системи и технологии"

АВТОРЕФЕРАТ

за присъждане на образователна и научна степен "Доктор"
на маг. инж. Милица Тошкова Костадинова- Гочева

Докторска програма: "Компютърни системи и технологии"

Професионално направление:

5.3 Комуникационна и компютърна техника

Област от висшето образование: 5. Технически науки

ТЕМА:

**Изчислителни методи за импулсно изследване на генетични
невронни мрежи**

Научни ръководители:

Проф. дмн Гани Трендафилов Стамов

Проф. д-р Станислав Денчев Симеонов

Бургас
2026 г

Дисертационният труд е обсъден и допуснат до защита на разширено заседание на катедра “Компютърни системи и технологии”, проведено на 7.5.2026 г. в Бургаски държавен университет “Проф. д-р Асен Златаров” - Бургас.

Дисертационният труд съдържа 149 страници, от които 36 фигури и 8 таблици. Използвани са 128 литературни източници. Резултатите са публикувани в 6 статии.

Защитата на дисертационния труд ще се състои на 2026 г. от ч. в зала в Бургаски държавен университет “Проф. д-р Асен Златаров” - Бургас на научно жури в състав:

Материалите за защитата са на разположение на интересувашите се в на Бургаски държавен университет “Проф. д-р Асен Златаров” - Бургас.

Автор: маг. инж. Милица Тошкова Костадинова- Гочева

Тема: Изчислителни методи за импулсно изследване на генетични невронни мрежи

Използвани съкращения

ИНМ - Изкуствени невронни мрежи
иРНК - Информационната рибонуклеинова киселина
AMD - Advanced Micro Devices
API - Application Programming Interface
CUDA - Compute Unified Device Architecture
CNN - Convolutional Neural Networks
GPU - Graphics Processing Unit
GRU - Gated Recurrent Unit
GRN - Gene Regulatory Networks
IBM - International Business Machines Corporation
IGRNs - Impulsive Gene Regulatory Networks
MLP - Multilayer Perceptron
MPI - Message Passing Interface
OpenCL - Open Computing Language
RNN - Recurrent Neural Networks
DNN - Deep Neural Network
SNNs - Spiking Neural Networks
ROCm - Radeon Open Compute
HIP - Heterogeneous-computing Interface for Portability
HPC - High-Performance Computing

Глава 1

Увод. Цел и задачи.

Областта на интелигентните системи напоследък демонстрира значителни пробиви с няколко постижения, базирани на биологични невронни мрежи, които предоставят естествени модели за разработване на нови системи. Изкуствените невронни мрежи (ИНМ) са създадени благодарение на съвместните усилия на изследователи от различни научни области, за да изпълняват различни задачи, като разпознаване и прогнозиране на образи, оптимизация, асоциативна памет и управление на процеси.

Целта на настоящия дисертационен труд е да се моделират устойчиви процеси при невронни мрежи от тип импулсни генетични регулаторни невронни мрежи. Да се разработи софтуерно приложение, което улеснява въвеждането на входни параметри, извличане на резултатни данни, съставяне на графики на получените резултати и изготвяне на доклади съгласно разглежданите необходими зависимости за устойчивост на модела.

Изпълнение на следните задачи:

- Задача 1. Да се проучат съществуващи софтуерни решения за Импулсни генетични регулаторни невронни мрежи или сходен модел
- Задача 2. Да се създаде математически модел и да се изследват условията за устойчивост при мрежа от три неврона

- Задача 3. Да се моделират условията за устойчивост на разглеждания модел невронни мрежи
- Задача 4. Да се създаде и реализира класически алгоритъм за невронни мрежи на разглеждания модел
- Задача 5. Да се реализират и изследват паралелни алгоритми за условия за устойчивост на съществуваща конфигурация
- Задача 6. Да се реализира паралелен алгоритъм на база OpenMPI
- Задача 7. Да се реализира паралелен алгоритъм на база ROCm
- Задача 8. Да се реализира паралелен алгоритъм на база хибриден модел OpenMPI и ROCm
- Задача 9. Да се направи сравнение между трите технологии
- Задача 10. Да се изследват резултатите и тяхното представяне.

Предмет на дисертационният труд е получаване, изследване и анализиране на устойчиви процеси при невронни мрежи от тип импулсни генетични регулаторни невронни мрежи с променливи импулсни смущения чрез софтуерни модели, библиотеки. Използването на паралелна техника дава предимство за скалируемост за получаване на резултати, което води до ускоряване на процесите.

Глава 2

Неврон и Невронни мрежи.

В настоящата глава се анализира еволюционното развитие на невронните мрежи през последните десетилетия. Представена е ретроспекция на прехода от използването на единичен неврон за решаването на специфични казуси към внедряването на еднослойни и многослойни архитектури с цел разширяване на техния апроксимационен и класификационен капацитет.

2.1 Биологични невронни мрежи

Човешкият мозък функционира като масивно паралелна система, съставена от приблизително 100 милиарда неврона, подпомагани от множество невроглиални клетки. Високата изчислителна мощ на биологичната система се дължи на огромната мрежа от връзки – всеки неврон може да взаимодейства с до 10 000 други клетки. Общият брой на синаптичните връзки в мозъка достига 100 трилиона, което позволява обработка на информация с интензитет, оценяван на около 1 трилион бита в секунда.

2.2 Изкуствени невронни мрежи

В този раздел се разглеждат теоретичните основи и архитектурните принципи на изкуствените невронни мрежи (ИНМ). Те са представени като математически модели, вдъхновени от биологичните невронни структури, предназначени за решаване на сложни задачи чрез апроксимация, класификация и разпознаване на образи.

2.2.1 История и развитие на невронните мрежи

Направен е ретроспективен анализ на еволюцията на ИНМ — от концептуалните идеи през 40-те години на XX век до съвременната ера на дълбокото обучение (Deep Learning).

2.2.2 Първия формален модел - невронът на Мак-Кълък и Пийтс

Детайлно е описан моделът на Уорън Маккълък и Уолтър Питс от 1943 г., който поставя основите на изчислителната невронаука. Анализирани са принципът му на работа като логически прагов елемент, който демонстрира, че мрежа от такива елементи може да изпълнява произволни логически функции, поставяйки началото на бинарното представяне на невронната активност.

2.2.3 Мрежови архитектури видове изкуствени невронни мрежи

Представена е класификация на основните видове мрежови топологии.

2.2.4 Компоненти на изкуствена невронна мрежа

Описани са фундаменталните градивни елементи на ИНМ:

- Входи и тегла: представящи силата на връзките между елементите.
- Сумираща функция: изчисляваща претеглената сума на входните сигнали.
- Активационна функция: определяща изходното състояние на

неврона (Sigmoid, ReLU, Tanh и др.).

- Слоеви: входен, скрити и изходен слоеви, определящи дълбочината на модела.

2.2.5 Обучение

В този раздел невронните мрежи се класифицират в две основни категории въз основа на тяхната структура и начин на обработка на информацията:

- **Статични мрежи:** Характеризират се с директно изчисляване на изхода от входа чрез предварителни връзки (feedforward). Типичен пример е многослойният перцептрон.
- **Динамични мрежи:** Те съдържат закъснения и рекурентни (обратни) връзки. За разлика от статичните, изходът тук зависи не само от текущия вход, но и от предходните състояния или входове на мрежата (напр. мрежа на Хопфийлд).

2.2.6 Устойчивост

Рекурентните невронни мрежи (RNN) се отличават от мрежите с предварителна връзка (feedforward) по способността си да обработват времеви и пространствени зависимости. Докато при стандартните мрежи изходът зависи единствено от текущия входен вектор, при рекурентните модели той е функция на времето. Това води до по-сложно поведение: при фиксиран вход изходът може да се стабилизира, но също така може да осцилира, да нараства неограничено или да проявява хаотична динамика.

Глава 3

Математически модел на импулсни генетични регулаторни невронни мрежи

3.1 Импулсни генетични регулаторни невронни мрежи

В настоящия раздел се представя развитието на импулсните генетични регулаторни невронни мрежи като интердисциплинарна научна област. Направен е ретроспективен преглед на ключовите етапи от тяхното възникване и са систематизирани значимите научни приноси в литературата. Въз основа на прегледа са идентифицирани недостатъчно изследваните аспекти и нерешените проблеми, които определят насоките на настоящата работа.

3.2 Теоретичен модел

Дефинира се теорема и доказателствена част за равновесието на системата на импулсна генетична невронна мрежа по посочения

модел (3.3) за глобална експоненциална устойчивост по отношение на функцията h .

Представен е следния (IGRNs) модел, зададен чрез:

$$\left\{ \begin{array}{l} \dot{m}_i(t) = -a_i m_i(t) + \sum_{j=1}^n w_{ij}(t) f_j(p_j(t)) + B_i(t), \quad t \neq t_k, \\ \dot{p}_i(t) = -c_i p_i(t) + d_i(t) m_i(t), \quad t \neq t_k, \\ \Delta m_i(t_k) = \alpha_{ik} m_i(t_k) + \nu_{ik}, \\ \Delta p_i(t_k) = \gamma_{ik} p_i(t_k) + \chi_{ik}, \end{array} \right. \quad (3.3)$$

които регулират концентрациите на иРНК $m_i(t)$ и протеин $p_i(t)$ във времето t , където:

$i/ i, i = 1, 2, \dots, n$ е броят на възлите, a_i, c_i са положителни константи, представляват степените на разреждане, като се използва безразмерната транскрипция ограничена скорост $q_{ij}(t)$ в момент t на транскрипционния фактор j до i , $w_{ij}(t)$ се определят като:

$$w_{ij}(t) = \left\{ \begin{array}{ll} q_{ij}(t), & \text{когато } j \text{ е активатор на гена } i, \\ -q_{ij}(t), & \text{когато } j \text{ е репресор на гена } j, \\ 0, & \text{когато няма връзка от възела } j \text{ към гена } i, \end{array} \right.$$

$d_i(t) \in \mathbb{R}$ обозначава скоростта на транслационния процес, f_j представлява регулаторната (активираща) функция на протеина и има вида

$$f_j(p_j) = \frac{(p_j/\beta_j)^{H_j}}{1 + (p_j/\beta_j)^{H_j}},$$

където H_j обозначават коефициентите на Хил, а β_j са положителни скалярни величини, $q_i(t)$ се определя като $q_i(t) = \sum_{\mu \in I_i} q_{i\mu}(t)$, където I_i е множеството, съдържащо всички репресори на гена i ;

Демонстрират се валидността на получените резултати от Теорема 3.3 - критерии за за глобална експоненциална устойчивост по отношение на множествата.

Глава 4

Паралелни алгоритми за изследване на импулсни генетични невронни мрежи

Човешкият мозък функционира чрез сложни мрежи от неврони и синапси, които позволяват паралелна обработка на информация и самообучение. Изкуствените невронни мрежи се стремят да емулират този биологичен модел чрез архитектура от взаимно свързани възли за разпределени изчисления. Въпреки че съвременните компютри предлагат огромна изчислителна скорост, те все още не могат напълно да пресъздадат мащаба и функционалната сложност на човешкия мозък [114].

4.1 Основни понятия

4.1.1 Последователно време на изпълнение

Последователното време на изпълнение T_1 е времето, необходимо за изпълнение на дадена програма върху един процесор, без паралелизация.

4.1.2 Паралелно време на изпълнение

Паралелното време на изпълнение T_p е времето, необходимо за изпълнение на същата програма върху p процесора (ядра), използвайки паралелни алгоритми.

4.1.3 Закони за скалируемост

Формулата на Амдал за общо ускорение (Speedup) при паралелизация:

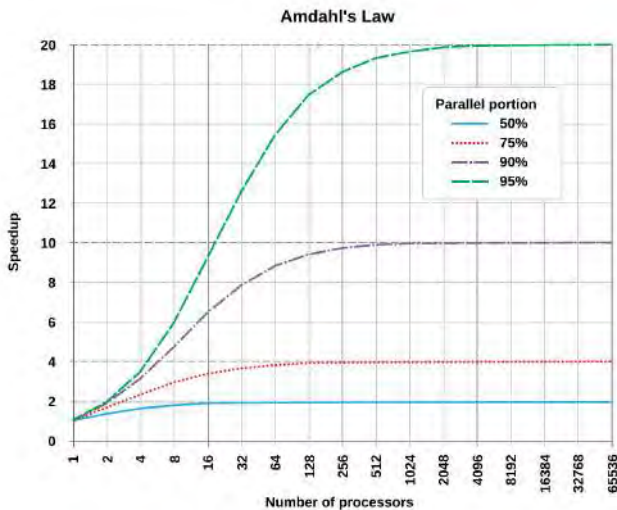
$$S = \frac{1}{(1 - p) + \frac{p}{s}} \quad (4.1)$$

където:

p – броят на процесорите,

s – е ускорението на паралелизируемата част (често означавано като брой процесори или ускоряващ фактор),

$(1 - p)$ – неизбежната последователна част [115].



Фигура 4.1: Параметрично влияние на скалируемостта според закон на Амдал

4.1.4 Закони за паралелна ефективност

Паралелната ефективност E_n измерва каква част от времето изчислителните ресурси се използват полезно. Тя се дефинира като отношението на ускорението към броя на използваните процеси [116]:

$$E_n = \frac{S_n}{n} \times 100 \quad (4.2)$$

където:

- E_n е паралелната ефективност на системата;
- S_n е ускорението при използване на n процесора;
- n е броят на използваните процесори;

В идеалния случай $E_n = 100\%$, но в реални условия тя намалява поради т.нар. Паралелен овърхед (Parallel Overhead), който включва: Време за латентност на PCIe шината при трансфер на данни между Host и Device. Синхронизационни закъснения при операцията MPI_Allgather. Конфликти при достъпа до общата системна памет (Memory Socket contention)[116].

4.2 Апаратна реализация на високопроизводителни и паралелни системи

4.2.1 Класификация

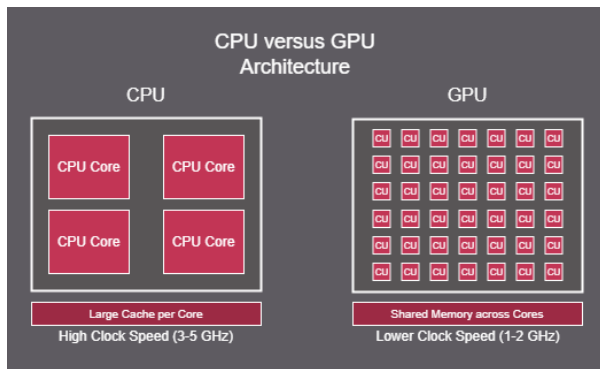
High Performance Computing (HPC)

Настоящият раздел разглежда програмния модел HIP (Heterogeneous-compute Interface for Portability) като инструмент за реализиране на паралелни C/C++ алгоритми върху масово паралелни SIMD архитектури. Анализира се фундаменталната разлика между централните и графичните процесори, като се акцентира върху необходимостта от специфичен подход при програмирането за постигане на оптимална производителност. В исторически план е проследена еволюцията на софтуерния стек на AMD – от инициативата Boltzmann до утвърждаването на отворената платформа ROCm. Специално внимание е отделено на ролята на езика HIP като средство за постигане

на преносимост, позволяващо компилация на програмен код както за AMD, така и за NVIDIA хардуер чрез синтаксис, идентичен с този на CUDA. По този начин HIP се утвърждава като ключов елемент в съвременните хетерогенни изчислителни системи.

Разлики в хардуера: CPU срещу GPU

Централните процесори (CPU) и графичните процесори (GPU) са проектирани за различни цели. Централните процесори (CPU) бързо изпълняват една нишка, намалявайки времето за една операция, като същевременно увеличават броя на последователните инструкции, които могат да бъдат изпълнени. Това включва извличане на данни и намаляване на застойте в конвейера, където АЛУ (асистентско-логическите единици) трябва да чака завършването на предишните инструкции.



Фигура 4.2: Разлики в процесорите и графичните процесори [117].

Докато централните процесори (CPU) са оптимизирани за серийна обработка с ниска латентност (бързо изпълнение на единична задача), графичните процесори (GPU) се фокусират върху висока производителност чрез паралелно изпълнение на множество нишки. Латентността измерва времето за завършване на една операция, докато производителността отчита общия брой операции за единица време.

Архитектурата на GPU се базира на изчислителни единици (CU), които функционират като основни градивни елементи за обработка на програмни ядра. Всяка изчислителна единица разполага със собствени ресурси, включително аритметично-логически единици (ALU), регистри и кеш памет, осигуряващи ефективна паралелна ко-

муникация [117].

Програмни модели и софтуер

Ключов елемент в HPC е софтуерният стек, който включва:

- MPI (Message Passing Interface) – стандарт за комуникация между процеси в разпределена памет
- OpenMP – модел за паралелизъм със споделена памет
- OpenCL - отворен, стандартизиран програмен модел за паралелни изчисления върху хетерогенни системи
- CUDA / OpenACC – програмни модели за GPU ускорение
- Числени библиотеки – BLAS, LAPACK, ScaLAPACK

Ефективното използване на HPC изисква оптимизация на алгоритмите с оглед на памет, комуникационни разходи и мащабируемост.

4.2.2 Високопроизводителна изчислителна система на база AMD GPU

Като предимно екосистема с отворен код, ROCm™ предоставя възможност за проверка, персонализиране и адаптиране на софтуерния стек към специфични изисквания, подкрепена от съвместна общност от разработчици. ROCm е софтуерен стек, съставен предимно от софтуер с отворен код, който предоставя инструментите за програмиране на графични процесори (GPU) на AMD, от ниско ниво ядра до високо ниво приложения за крайни потребители. ROCm предоставя инструментите за HIP (Heterogeneous-computing Interface for Portability - Хетерогенен изчислителен интерфейс за преносимост) , OpenCL и OpenMP. Те включват компилатори, библиотеки за функции на високо ниво, дебъгери, профилири и среди за изпълнение [117].

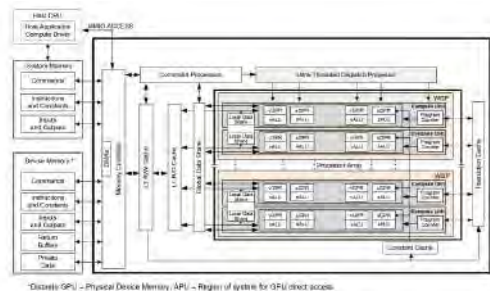


Фигура 4.3: Архитектура на ROCm [117].

AMD-базираната система може да бъде трансформирана в високопроизводителна и персонализирана работна станция за машинно обучение. ROCm™ 7.1.1 поддържа най-новите графични процесори Radeon™ 9000 Series (RDNA 4) и избрани графични процесори 7000 Series (RDNA 3), както и въвежда първоначална поддръжка за Ryzen™ APU, което позволява рентабилно локално разработване и изводи за изследователи и инженери, използващи Pytorch.

Процесорът AMD RDNA3 [118] реализира паралелна микроархитектура, която осигурява платформа за компютърни графични приложения, както и за паралелни приложения за данни с общо предназначение.

Фигурата по-долу показва блокова схема на процесорите от серията AMD RDNA3 Поколение:



Фигура 4.4: AMD RDNA3 Блокова схема на серията поколения [118].

Архитектурата RDNA3 се състои от паралелни процесори, команден процесор и контролер на паметта. Командният процесор управлява комуникацията с хоста чрез регистри и прекъсвания, докато контролерът на паметта осигурява директен достъп (DMA) до локалната и системната памет. Тази структура позволява автоматизирано изчисляване на адресни отмествания и ефективна обработка на данни в реално време. В средата на RDNA3, едно пълно приложение включва две части:

- програма, работеща на хост процесора
- програми, наречени шейдър програми или ядра, работещи на процесора RDNA3.

4.2.3 OpenMPI

MPI е стандарт, дефиниращ модела за обмен на съобщения в паралелни системи, докато OpenMPI е неговата практическа софтуерна реализация. Връзката между тях е като между архитектурен план и завършена сграда – MPI задава абстрактните правила и интерфейси, а OpenMPI осигурява библиотеките и средата за реалното им изпълнение върху хардуера. По този начин стандартът гарантира съвместимост, а конкретната реализация позволява практическото разработване на разпределени приложения.

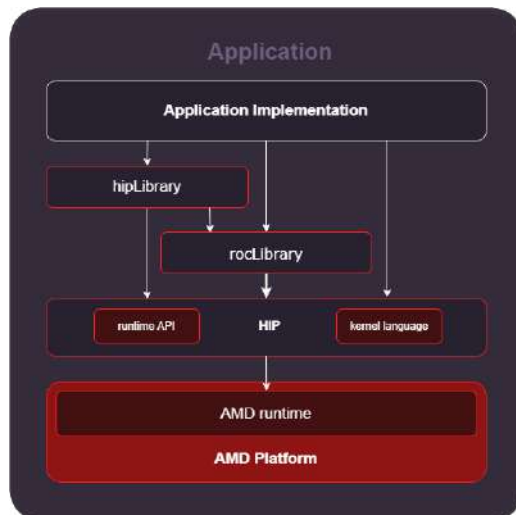
Стандарт MPI

Настоящият раздел разглежда Message-Passing Interface (MPI) като фундаментален стандарт за паралелно програмиране в среди с разпределена памет. MPI е дефиниран като библиотечна спецификация, а не като език, което позволява неговото имплементиране чрез различни функции и методи в езици като C, C++ и Fortran. Проследена е историческата еволюция на стандарта от създаването на MPI Forum през 1992 г. до съвременните версии MPI-3 и MPI-4, които са оптимизирани за работа с хетерогенни архитектури и ускорители (GPU). Специално внимание е отделено на механизма на предаване на съобщения между адресните пространства на отделните процеси и ролята на операционната система. В тази връзка е дефинирано значението на ядрото (Kernel) като привилигиран посредник между приложния софтуер и хардуерните ресурси, осигуряващ необходимата абстракция и контрол при изпълнението на паралелни процеси.

4.2.4 Хибридна паралелна архитектура с ROCm и OpenMPI

За изграждането на ексафлопс суперкомпютри е критична интеграцията между ROCm, комуникационната рамка UCX и „GPU-aware“ MPI. Тези технологии позволяват директен достъп до паметта на ускорителите чрез RDMA, което елиминира междинното копиране на данни и намалява латентността. Практическата ефективност на този модел е доказана при суперкомпютъра Frontier, базиран на хардуер от AMD. В тази архитектура езикът HIP се утвърждава като универсално решение за поддръжка на единна и преносима кодова база. Така се осигурява висока производителност върху разнообразни хетерогенни системи.

Един основен елемент е Unified Communication X (UCX). От версия 5.0.5 на OpenMPI и по-нови версии, включително настоящите компилации на ROCm 7.x за 2025/2026 г., UCX се предлага по подразбиране като транспортен слой, който разпознава ROCm указатели, позволявайки RDMA (Remote Direct Memory Access) и подобен на GPUDirect интерфейс чрез PeerDirect, което драстично намалява латентността на комуникацията между възлите [117], [121], [122], [123], [124].



Фигура 4.5: Архитектурата на HIP (Heterogeneous-Compute Interface for Portability) в екосистемата на AMD [117].

ROCm е платформа за хетерогенни изчисления, която предоставя инструменти, библиотеки и HIP API за разработка върху CPU и GPU. Чрез библиотеки като hipFFT и hipBLAS се осигурява съвместимост с CUDA, което улеснява интегрирането им в съществуващи приложения. Програмният модел HIP позволява поддържането на единна кодова база за хардуер на AMD и NVIDIA, като предлага инструменти за автоматизирано портиране (HIPify). Системата разделя изпълнението на хост (CPU) код за управление на данните и ядра (kernels) за масово паралелна обработка върху графичния процесор. Въпреки високата степен на автоматизация, постигането на оптимална производителност често изисква допълнителна ръчна настройка.

Глава 5

Алгоритми и имплементация на модел устойчивост при импулсни генетични невронни мрежи

Софтуерната реализация на метаматическия модел е разработена на програмния език C++. Разработката премина през четири етапа. Първоначално бе реализиран проекта без използване на паралелна техника, за да бъде разгледан и подобрен метода по който се правят изчисленията.

Във втория етап от разработката, софтуерната реализация бе постигната посредством технологията наречена OpenMPI, върху машина процесор с AMD Ryzen 9 7950X 16-Core Processor (4.50 GHz).

Третата стъпка се извърши чрез използването на технологията ROCm върху машина графичен процесор с AMD Radeon RX 7800 XT (2438 MHz). И четвъртата стъпка се осъществи с хибридната технология MPI + ROCm.

5.1 Обобщение на модела

За направа на различните софтуерни реализации се използва общ модел на невронна мрежа, който се базира на обобщения модел на Импулсни генетични невронни мрежи имащ вида на уравнение (3.3). Използвания модел има вида:

$$\left\{ \begin{array}{l} \dot{M}_i(t) = -a_i M_i(t) + \sum_{j=1}^n w_{ij}(t) f_j(P_j(t - \sigma_j(t))) + B_i(t), \quad t \neq t_k, \\ \dot{P}_i(t) = -c_i P_i(t) + d_i(t) M_i(t - \tau_i(t)), \quad t \neq t_k, \\ \Delta M_i(t_k) = \alpha_{ik} M_i(t_k), \\ \Delta P_i(t_k) = \gamma_{ik} P_i(t_k), \end{array} \right. \quad (3.3)$$

с импулсни смущения от типа:

$$\sum_{i=1}^n |B_i(t) + \tilde{B}_i(t)| < \rho, \quad t \geq 0, \quad \frac{\rho}{\mu} < A$$

където

(5.1)

Моделът (3.3) може да се разглежда като импулсно управляван модел, съответстващ на задвижващата система

$$\left\{ \begin{array}{l} \dot{M}_i(t) = -a_i M_i(t) + \sum_{j=1}^n w_{ij}(t) f_j(P_j(t - \sigma_j(t))) + B_i(t), \\ \dot{P}_i(t) = -c_i p_i(t) + d_i(t) M_i(t - \tau_i(t)), \end{array} \right. \quad (5.2)$$

$i = 1, 2, 3$.

За модела (3.3) можем да проверим дали всички условия на Теорема 3.3 са изпълнени за

$$l_j = 1, \quad j = 1, 2, 3;$$

$$|w_{ij}(t)| \leq 2.3, \quad |d_i(t)| \leq 0.7, \quad i, j = 1, 2, 3;$$

$$\alpha_{ik} = -\frac{2}{7}, \quad \gamma_{ik} = -\frac{3}{8}, \quad i = 1, 2, 3, \quad k = 1, 2, \dots$$

Фиксирани стойности на масивите a_i, c_i, d, f_i, i има само при процедурната реализация, като те са както следва (5.3). Реализациите използват паралелна техника с начални стойности на масивите се задават с минималната стойност подадена като параметър към програмата.

$$w_{ij}(t) = \begin{pmatrix} w_{11}(t) & w_{12}(t) & w_{13}(t) \\ w_{21}(t) & w_{22}(t) & w_{23}(t) \\ w_{31}(t) & w_{32}(t) & w_{33}(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -2.3 \cos(t) \\ -2.3 \sin(t) & 0 & 0 \\ 0 & -2.3 \cos(t) & 0 \end{pmatrix} \quad (5.3)$$

$t \geq 0, 0 < t_1 < t_2 < \dots < t_k < t_{k+1} < \dots, t_k \rightarrow \infty$ като $k \rightarrow \infty$.

Търси се устойчива точка, така че да съществува положително число μ , което да удовлетворява следното уравнение (3.3).

$$\min_{1 \leq i \leq n} \{(a_i + \tilde{a}_i), (c_i + \tilde{c}_i)\} - \max_{1 \leq i \leq n} \{(K_i^{(2)} + \tilde{K}_i^{(2)}), \sum_{j=1}^n (K_{ji}^{(1)} + \tilde{K}_{ji}^{(1)}) l_i\} \geq \mu$$

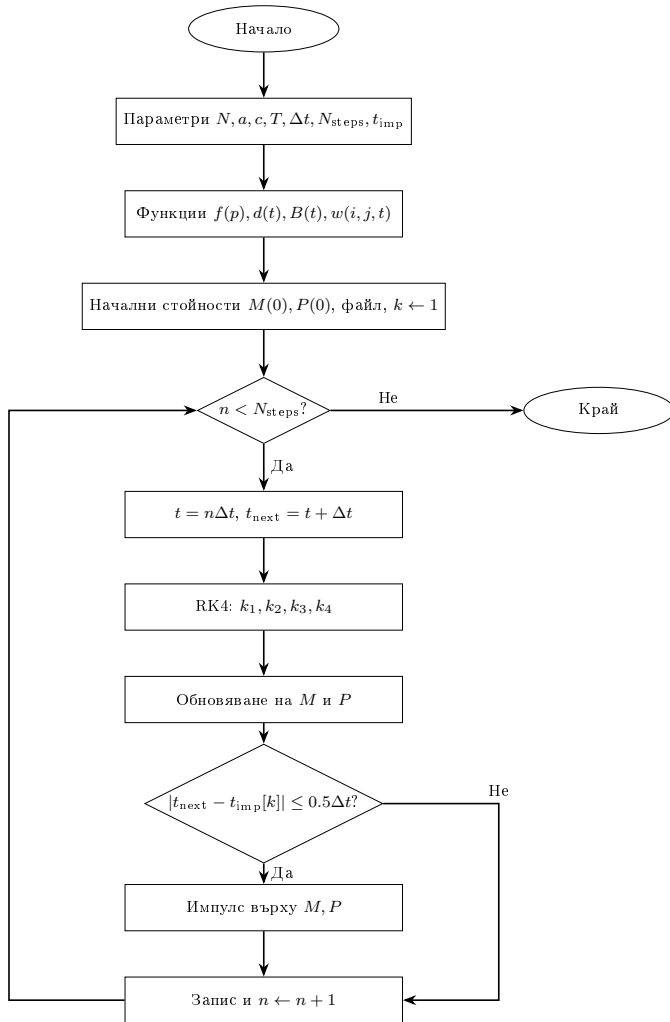
Предполага се, че всички предположения в Теорема 6 са удовлетворени, хипотези от 1 до 5 са удовлетворени. Задаваме съответните константи както следва:

$$a_1 = a_2 = a_3 = 2.9, \quad c_1 = c_2 = c_3 = 2.3, \quad B_1(t) = B_2(t) = B_3(t) = 0, \\ d_1(t) = d_2(t) = d_3(t) = 0.7 \sin(t). \quad (5.4)$$

5.2 Процедурна реализация на импулсния модел (IGRNs)

За целите на численото изследване на импулсни генетични регулаторни невронни мрежи (IGRNs) се конструира процедурна реализация на модела, основана на метода на Рунге–Кута от четвърти ред (RK4), изпълнявана последователно върху един процесор (CPU). Моделът описва свързана система от n гени, за които се въвеждат

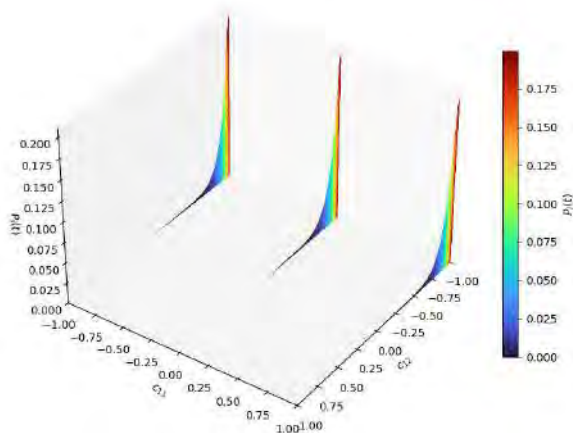
динамики на mRNA концентрация $m_i(t)$ и протеинова концентрация $p_i(t)$.



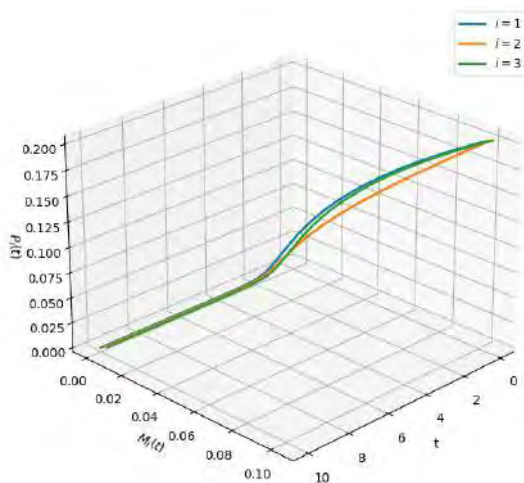
Фигура 5.1: Блок-схема на процедурната реализация

Полученият текстов файл съдържа траекториите (t, M, P) и се използва за визуализация в MATLAB или Python.

Графично обобщение на данните



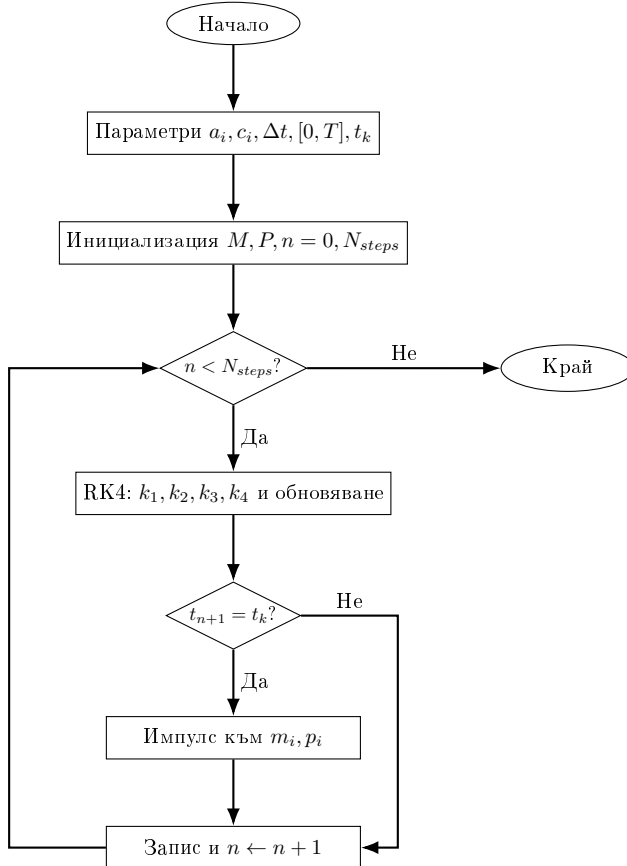
Фигура 5.2: Параметрична зависимост на функцията $P_i(t)$ от параметрите c_1 и c_2 .



Фигура 5.3: Фазова динамика на системата в пространството (t, M_i, P_i) .

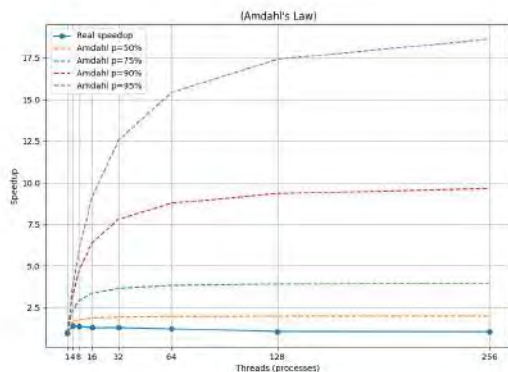
5.4 Процедурна реализация с OpenMPI

За численото изследване на системата от тип IGRNs е разработена процедурна реализация на модел с импулси и непрекъсната динамика между тях. Решението е изпълнено на езика C++ с използване на OpenMPI и метод на Рунге–Кута от четвърти ред (RK4). Реализацията е изпълнена в няколко етапа.



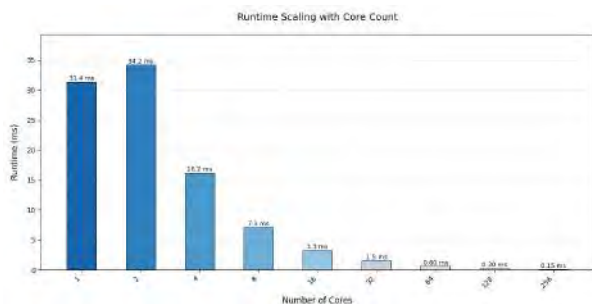
Фигура 5.4: Блок–схема на OpenMPI реализацията

Параметрично влияние на скалируемостта според закон на Амдал



Фигура 5.5: Параметрично влияние на скалируемостта според закон на Амдал

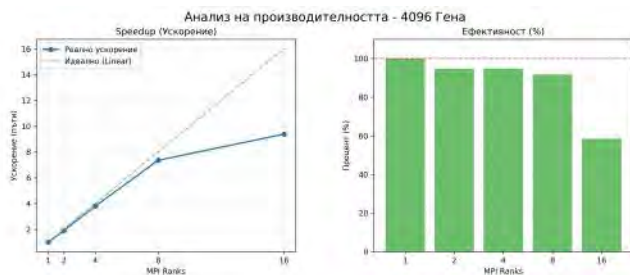
На Фигура 5.5 е показана графиката на функцията, базирана на проверената конфигурация. Оценката на мащабируемостта ще бъде извършена според времето за обработка на mRNA и концентрациите на протеини. Алгоритмизацията на проблема се концентрира върху сложността и оптимизирането на комуникацията между различните процесори.



Фигура 5.6: Време за изпълнение на задачите

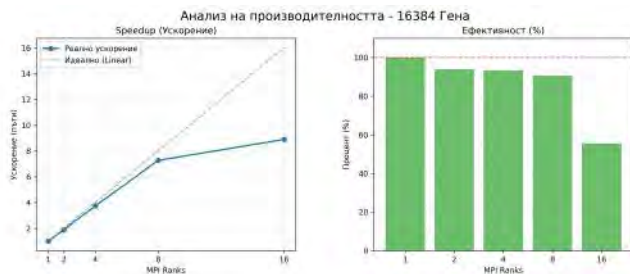
Фигура. 5.6 показва времето за изпълнение на задачата като брой процесори. В началото има ограничение за ускорението на нарастване, но след определен брой процесори тази бариера едва се достига. Наблюдавана е промяна относно по-ниско време за обработка. Състоянието започва от удълженото време за инициализация и достъп до RAM паметта.

5.4.1 Сравнителен анализ



Фигура 5.7: Бенчмарк анализ при 4096 гена

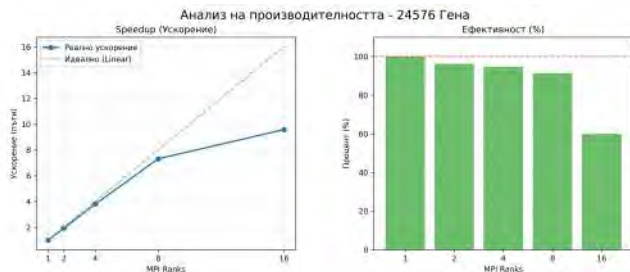
Графиката показва почти идеално линейно ускорение до 8 ранга. Синята линия (реалното ускорение) се движи плътно до пунктираната сива линия (идеалното ускорение). При 8 ранга се постига ускорение от 7.35x. Това означава, че задачата се изпълнява над 7 пъти по-бързо, отколкото с един единствен процес.



Фигура 5.8: Бенчмарк анализ при 16 384 гена

На Фигура 5.8 с 16384 гена се забелязва, че ускорението до 8 ранга остава отлично (7.3x), почти идентично с това при 4096 гена.

Това показва, че алгоритъмът е устойчив на мащабиране. Критичната точка е при 16 ранга. Тук се вижда ясно изразен спад. Ускорението нараства съвсем малко (до около 8.8x), вместо да се доближи до идеалните 16x. Това е типичен пример за Закона на Амдал.

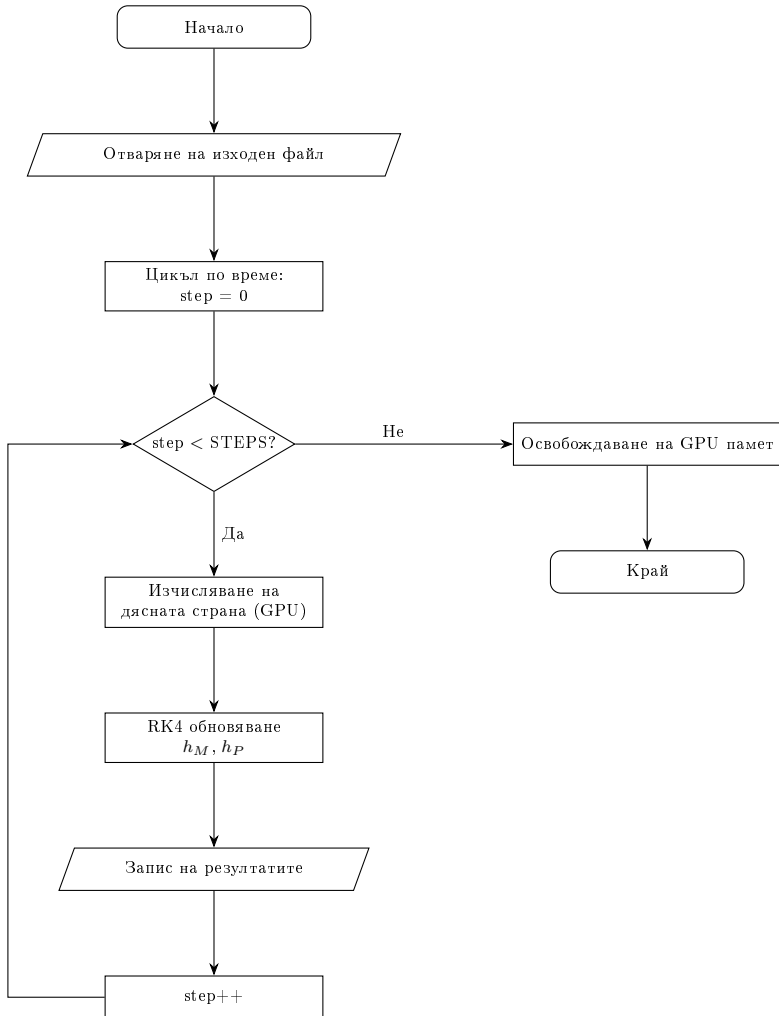


Фигура 5.9: Бенчмарк анализ при 24 576 гена

При тези изследвания се достига най-високото абсолютно ускорение — 9.59x при 16 ранга. Това е подобрение спрямо 16384 гена (където е около 8.8x). По-големият обем работа (повече гени) позволява на системата да оползотвори по-добре наличните ресурси, дори когато ефективността пада. Това е класическо доказателство, че паралелното програмиране е най-изгодно при големи задачи.

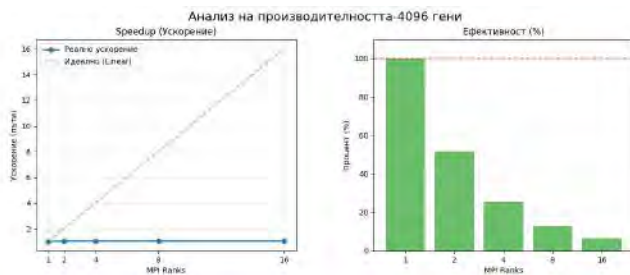
5.5 ROCm реализация

С цел ускоряване на численото интегриране на импулсния модел (IGRNs) е разработена GPU-ускорена версия на алгоритъма, използваща AMD ROCm/HIP.

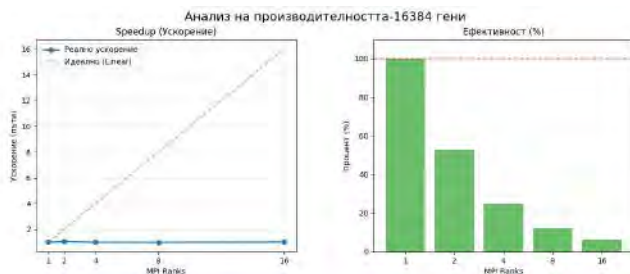


Фигура 5.10: Блок-схема на ROCm реализацията

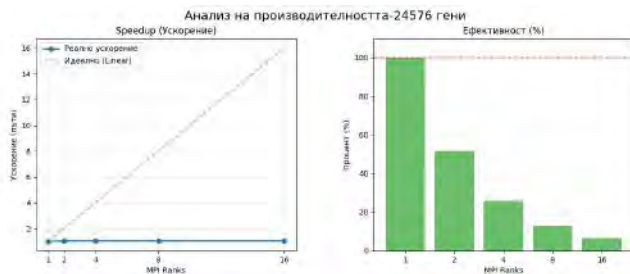
5.5.1 Сравнителен анализ



Фигура 5.11: Бенчмарк анализ при 4096 гена



Фигура 5.12: Бенчмарк анализ при 16384 гена



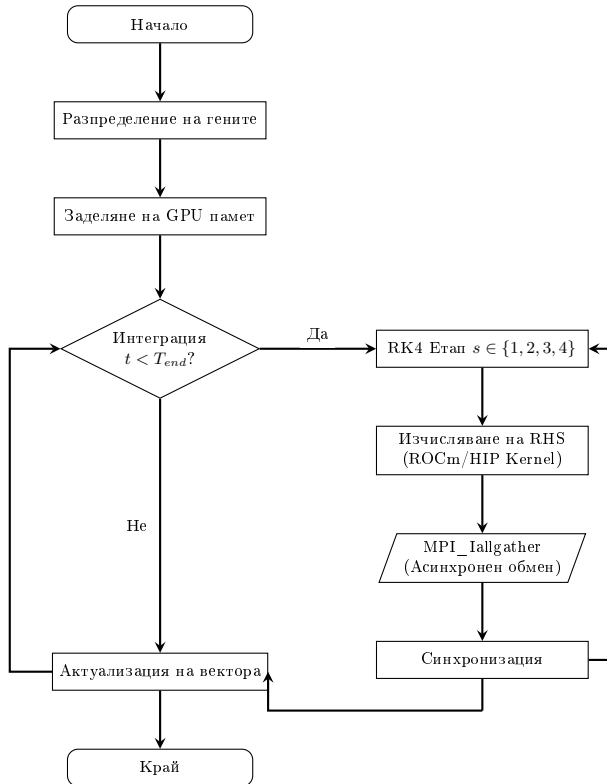
Фигура 5.13: Бенчмарк анализ при 24576 гена

Експерименталните резултати показват отсъствие на значима мащабируемост, като кривата на реалното ускорение (Speedup)

остава близо до единица, независимо от увеличаването на броя на паралелните рангове. Този феномен е пряко следствие от ниската натовареност на изчислителните ядра, при която обемът на обработваните данни не е достатъчен, за да компенсира системните разходи за синхронизация и управление на нишките. Наблюдава се инверсна зависимост между паралелната ефективност и броя на изчислителните единици, като при 16 ранга ефективността редуцира до стойности под 10. Тези данни потвърждават, че текущата изчислителна задача не достига критичната маса, необходима за пълното оползотворяване на изчислителния капацитет и пропускателната способност на графичния процесор.

5.6 Хибридна реализация OpenMPI + ROCm

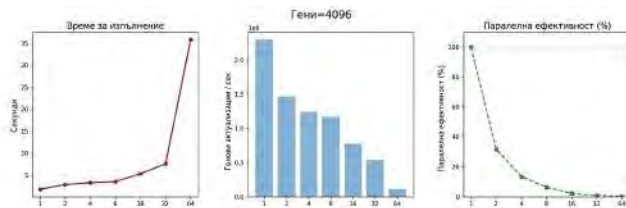
The program code implements a numerical solution of a system of differential equations modeling an integrated gene regulatory network (IGRNs). A **4th-order Runge–Kutta (RK4)** method is used, combined with parallel processing through **MPI** for distributed memory and **HIP** for GPU computations.



Фигура 5.14: Блок-схема на Хибридната реализация.

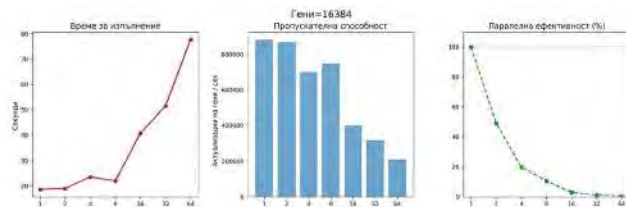
5.6.1 Сравнителен анализ

Най-високата пропускателна способност от 2,38 милиона обновявания/сек се постига при 1 MPI ранг. Това е така, защото 4096



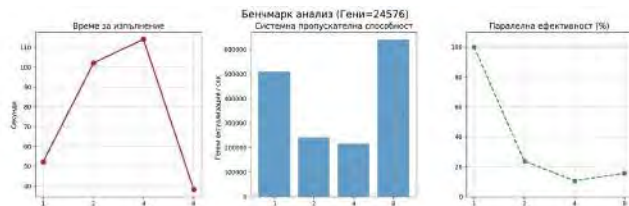
Фигура 5.15: Бенчмарк анализ при 4096 гена

гена са твърде малко натоварване за RX 7800 XT, за да оправдаят разделянето им между много процеси. Времето за изпълнение се увеличава с добавянето на нови процеси. При 64 процеса симулацията е близо 20 пъти по-бавна, заради огромното закъснение при *MPI_Allgather* и претоварването на командната опашка на графичния процесор. Наблюдава се леко подобрене при прехода от 4 към 8 процеса, но след това производителността рязко се понижава.



Фигура 5.16: Бенчмарк анализ при 16 384 гена

В проведените бенчмарк тестове върху мрежа от 16 384 гена, хибридната MPI+ROCm архитектура поддържа висока ефективност (над 98%) с ниско ниво на паралелизъм между хостовете. Тъй като обаче има споделяне на ресурси на един графичен процесор (AMD RX 7800 XT), увеличаването на MPI ранга над 8 води до големи разходи за комуникация и планиране на задачи, което в следствие на това намалява пропускателната способност.



Фигура 5.17: Бенчмарк анализ при 24 576 гена

Осем нишки означават осем паралелни опашки от команди - натовареността на изчислителните модули. Разделянето на много голямата W матрица на осем части/парчета изглежда е в съответствие с L3 кеша/Infinity Cache на RX 7800 XT. За големи системи ($N > 20\,000$), това е вярно и за архитектурата gfx1101, където осем MPI ранга се изпълняват по-бързо от един последователен поток на изпълнение.

Оптималната работна точка се намира при 8 MPI ранга, където се постига баланс между натоварването на изчислителните единици и комуникационните разходи.

Настоящата работа допринася с това, че оптималната точка на работа е при 8 MPI ранга, където хибридният метод може да постигне максимална производителност от 639 297 генни актуализации в секунда. Този капацитет е критично важен при извършването на мащабни анализи на устойчивостта, както се изисква от синтетичната биология и генното инженерство.

Глава 6

Заклучение

В съответствие с поставената цел в настоящата дисертация се моделират устойчиви процеси при импулсни генетични регулаторни невронни мрежи с импулсни смущения, чрез използването на високопроизводителна и паралелна техника, бяха изпълнени следните задачи:

- Задача 1. Да се проучат съществуващи софтуерни решения за импулсни генетични регулаторни невронни мрежи или сходен модел
- Задача 2. Да се създаде математически модел и да се изследват условията за устойчивост при мрежа от три неврона
- Задача 3. Да се моделират условията за устойчивост на разглеждания модел невронни мрежи
- Задача 4. Да се създаде и реализира класически алгоритъм за невронни мрежи на разглеждания модел
- Задача 5. Да се реализират и изследват паралелни алгоритми за условия за устойчивост на съществуваща конфигурация
- Задача 6. Да се реализира паралелен алгоритъм на база OpenMPI
- Задача 7. Да се реализира паралелен алгоритъм на база ROCm

- Задача 8. Да се реализира паралелен алгоритъм на база хибриден модел OpenMPI и ROCm
- Задача 9. Да се направи сравнение между двете технологии
- Задача 10. Да се изследват резултатите и тяхното представяне.

Настоящата разработка постига целите си чрез изпълнение на поставените задачи.

Въз основа на експерименталните данни, за постигане на максимална производителност при мащабни симулации се препоръчват следните стратегии:

Препоръчва се хибридната технология OpenMPI + ROCm. Тя позволява на системата да използва най-доброто от двата технологии: MPI за междупроцесна комуникация и огромната изчислителна мощ на ROCm за обработка на данни.

Системата се препоръчва за задачи с над 20 000 гена. При по-малки задачи (например 4096), административните разходи на MPI и подготовката на GPU опашките превишават времето за самото изчисление.

За конкретната конфигурация се препоръчва използването на 8 MPI ранга. Това осигурява оптимален баланс – достатъчно процеси за хранване на видеокартата с данни, без да се предизвиква ограничената пропускателна способност на системната шина за шината на паметта или претоварване на L3 кеша.

За съвременните нужди на биоинформатиката, чистият OpenMPI е остарял подход. Бъдещето и високата производителност се крият в хетерогенните изчисления, които резултати бяха демонстрира с 9.59x ускорение.

Приноси на дисертационния труд

Научно-приложни приноси на дисертационния труд

1. Литературен обзор
2. Създаване на математически модел на импулсна генетична регулаторна невронна мрежа (IGRNs), състояща се от три неврона.
3. Изследване на устойчивост: Дефинирани са и са изследвани условията за глобална асиоптична експоненциална устойчивост на решенията по отношение на множество за разглеждания модел с импулсни смущения.
4. Теоретична обосновка на паралелното изпълнение: Изследвани са зависимостите между софтуерната реализация и скалируемостта на хардуерни конфигурации при високопроизводителни системи ROCm и OpenMPI.

Приложни приноси

Тези приноси се отнасят до практическата софтуерна реализация, алгоритмите и експерименталната част:

1. Софтуерно проучване: Установена е липсата на съществуващи готови софтуерни решения за специфичния тип импулсни генетични регулаторни невронни мрежи към момента на разработката.

2. Софтуерна реализация на език C++: Разработване на процедурен метод за изчисляване на математическия модел за доказване на критерия за устойчивост.

Разработване на паралелни алгоритми:

3. Реализация на база OpenMPI.
4. Реализация на база ROCm.
5. Реализация на хибриден модел OpenMPI + ROCm.
6. Провеждане на сравнителен анализ: Направено е сравнение между трите технологии върху конкретна хардуерна конфигурация.

Публикации

- [1] Stamov, G., Stamova, I., Kostadinova M., 2024, Impulsive Gene Regulatory Networks: Stability of Sets, 2024, BioInfoMed'2024 [под печат]
- [2] Stamov, G., Stamova, I., Kostadinova M., 2025, Impulsive Delayed Gene Regulatory Networks and Practical Stability of h-manifolds, Continuous and Discrete Dynamics: Theory and Applications, Springer [под печат]
- [3] Kostadinova-Gocheva, M. On the Impulsive Gene Regulatory Networks. Announcements of the Union of Scientists – Sliven, Vol. 38, No. 2, 2023, p. 11 - 16
- [4] Stamov, G., & Kostadinova-Gocheva, M., 2024, Robust stability of sets for uncertain impulsive gene regulatory networks, Proceedings of the Technical University of Sofia, Vol. 74, No. 3, doi: 10.47978/TUS.2024.74.03.009
- [5] КОСТАДИНОВА-ГОЧЕВА, М. Нови тенденции в изучаването на импулсни генетични регулаторни невронни мрежи. В: Сборник с доклади от Студентска научна конференция – 2024. Бургас: Бургаски свободен университет, 2024, ISSN: 1311-221X, p. 91–95.
- [6] Kostadinova-Gocheva, M. The implementation of Impulsive Gene Regulatory Networks Applying the Message Passing Interface, Proceedings of the Technical University of Sofia, Vol. 76, No. 1, 2026. doi: 10.47978/TUS.2025.76.01.007.