Інформаційні технології

УДК 004.056.55:004.312.2 **doi: 10.26906/SUNZ.2025.4.39-44**

М. Е. Бондаренко, Г. С. Іващенко

Харківський національний університет радіоелектроніки, Харків, Україна

ОРГАНІЗАЦІЯ ПАРАЛЕЛЬНОГО ВИКОНАННЯ МЕТОДІВ ОБРОБКИ ГОЛОСОВИХ СИГНАЛІВ НА БАГАТОЯДЕРНИХ CPU ТА GPU

Анотація. Актуальність. Набули поширення такі системи, як голосові помічники та засоби ідентифікації мовця, які функціонують на основі обробки голосових сигналів. Продуктивність цих систем залежить від обсягів даних і умов функціонування. Обробка великих масивів голосових сигналів або забезпечення роботи в реальному часі вимагає високопродуктивних обчислень. Таку швидкодію можна досягти за допомогою масивно-паралельних систем, включаючи багатопроцесорні кластери або дискретні GPU. **Об'єкт дослідження**: організація паралельних обчислювальних процесів у задачах обробки голосових сигналів із використанням можливостей архітектур сучасних процесорів. **Мета статті:** розробка системи паралельної обробки голосових сигналів з адаптованими алгоритмами для багатоядерних CPU, інтегрованих та дискретних GPU. **Результати дослідження**. Порівняльний аналіз показав, що за невеликого навантаження (голосові помічники, персональні застосунки) достатньо використання CPU, що забезпечує ефективне виконання обчислень із низькими часовими затримками. Натомість для обробки великих масивів голосових даних запропонований паралельний підхід, реалізований на CPU і GPU, що скорочує час виконання на 25-30% порівняно з послідовною реалізацією на CPU. **Висновки**. Дослідження показали, що використання паралелізму на CPU доцільне для етапів обробки голосового сигналу із малим обсягом обчислень, тоді як дискретні GPU можуть бути використані на етапах обчислення MFCC, спектрального віднімання та вейвлет-фільтрації.

Ключові слова: системи голосової ідентифікації, обробка сигналів, виділення характеристик, нормалізація, MFCC, спектральне віднімання та вейвлет-фільтрація, CPU, GPU, iGPU.

Вступ

Постановка проблеми. Потреба у захисті інформації від несанкціонованого доступу сприяє розвитку систем голосової ідентифікації, які застосовуються у біометричній аутентифікації, голосових асистентах, мобільних застосунках, хмарних сервісах, онлайн-банкінгу та системах контролю доступу [1]. Автоматизація цих систем забезпечується використанням методів обробки сигналів, зокрема спектральним аналізом та оцінкою тембрових і ритмічних характеристик голосу [2].

Зі збільшенням масштабів застосування та розвитком технологій обробки даних зростають вимоги до продуктивності обчислювальних платформ, оскільки обробка великих обсягів даних у реальному часі включаючи фільтрацію сигналу, вилучення ознак, обчислення спектральних характеристик і формування моделей голосу потребує значних обчислювальних ресурсів. Реальні умови експлуатації, що характеризуються високим фоновим шумом, наявністю реверберації та низькою якістю записів, створюють значне обчислювальне навантаження на систему через необхідність виконання шумозаглушення, компенсації реверберації, нормалізації сигналу та виділення ознак [3]. Через це актуальним є застосування паралельної обробки на багатоядерних CPU та дискретних GPU, що дозволяє підвищити продуктивність системи та скоротити час виконання обчислень.

Сучасні CPU від Intel, AMD та Apple забезпечують високу продуктивність завдяки багатоядерній архітектурі та підтримки багатопоточності, що робить їх ефективними для складних обчислень у реальному часі, зокрема аналізу даних, машинного навчання та розпізнавання мовлення [4]. Алгоритми з послідовною або обмеженою паралельною обробкою реалізуються на CPU, забезпечуючи контроль над порядком виконання команд та організацію потоків обчислень. На початкових етапах обробки голосових сигналів вони виконують шумозаниження, нормалізацію амплітуди, фреймування та виділення характеристик [3]. Проте обмежена здатність до масово-паралельних обчислень знижує ефективність використання багатоядерних CPU у сучасних системах, що потребують одночасного виконання великої кількості операцій. У таких випадках доцільно застосовувати інтегровані та дискретні GPU, здатні забезпечити обробку великих даних у реальному часі [5].

Графічні процесори від NVIDIA та AMD спеціалізуються на масово-паралельних обчисленнях і застосовуються у відеообробці, 3D-рендерингу та глибокому навчанні. Архітектура GPU підтримує одночасне виконання великої кількості однотипних операцій, обробку великих голосових масивів, обчислення MFCC та реалізацію складних моделей розпізнавання [6].

Інтегровані GPU від Intel та AMD використовують системну пам'ять, спільну з CPU, що знижує затримки при обміні даними та підвищує загальну енергоефективність. Їх застосування є доцільним при обмежених ресурсах або обробці невеликих обсягів аудіосигналів [7].

Аналіз останніх досліджень і публікацій. Актуальні напрямки розвитку систем обробки голосових сигналів відображені у сучасних дослідженнях [8–10], що присвячені організації паралельних обчи-

слень та підвищенню продуктивності на різних апаратних платформах для задач аналізу та обробки мовних даних.

Дослідження [8] демонструє особливості виконання мобільних моделей машинного навчання, які широко використовуються в системах голосової ідентифікації, зокрема Deep Neural Networks (DNN), на пристроях з обмеженими ресурсами.

В системах голосової ідентифікації обробка сигналів характеризується високим обчислювальним навантаженням на процесор, обумовленим необхідністю виконання складних алгоритмів попередньої обробки та виділення ознак. Як показано в [8], для невеликих задач, таких як обробка окремих фреймів або швидке вилучення ознак, CPU забезпечує достатню паралельність без залучення GPU. Для складних моделей або великих наборів даних GPU суттєво прискорює обчислення завдяки масово-паралельній обробці.

Дослідження [9] показує застосування розпізнавання мовлення у системах голосового керування, IP-телефонії та персональної ідентифікації. Особлива увага приділяється обчисленню MFCC для виділення фонетичних характеристик, що потребує значних ресурсів. Паралелізація обчислення MFCC виконана на Intel Core i5-2310 за допомогою технології OpenMP, із розбиттям сигналу на незалежні фрейми. Експериментальні результати показали значне скорочення часу обробки.

Дослідження [10] присвячене паралельній обробці прихованих марковських моделей (HMM), що у системах голосової обробки використовуються для моделювання та аналізу послідовності часових ознак мовних сигналів. Навчання та тестування моделей виконувалося на GPU з використанням архітектури паралельних обчислень CUDA, що дозволяє ефективно обробляти великі обсяги даних у багатопоточному режимі. На CPU виконувались послідовні та керуючі операції. За допомогою cuBLAS та механізму HyperQ було реалізовано одночасне виконання кількох обчислювальних черг на GPU, що дозволило скоротити час навчання та обробки моделей до 9 разів порівняно з використанням багатоядерних CPU.

Розглянуті дослідження підтверджують доцільність використання як багатоядерних CPU, так і GPU для реалізації паралельних обчислень у задачах обробки голосових сигналів та ідентифікації мовця.

Метою роботи є розробка та дослідження системи паралельної обробки голосових сигналів із адаптованими алгоритмами для багатоядерних CPU, інтегрованих та дискретних GPU.

Постановка задачі

Методи обробки голосових сигналів, такі як фреймування, виділення ознак, спектральний аналіз та застосування вейвлет-фільтрації або MFCC, повинні підтримувати паралельне виконання на CPU, інтегрованому або дискретному GPU. Для цього використовуються обчислювальні можливості процесорів та спеціалізовані бібліотеки, зокрема OpenMP для багатопоточності на CPU та CUDA для масово-паралельної обробки на GPU. Такий підхід дозволяє розподіляти обчислення між потоками або ядрами, ефективно використовувати ресурси пам'яті та оптимізувати обробку великих обсягів даних, забезпечуючи скорочення часу виконання та підвищення продуктивності системи [11-13].

Реалізація повинна підтримувати обробку коротких фрагментів із невеликим навантаженням і великих масивів однотипних обчислень, що відтворюють реальні умови експлуатації з варіаціями рівня шуму, спектральної насиченості та частоти дискретизації. Така конфігурація тестових даних дозволить оцінити здатність обчислювальних вузлів підтримувати одночасну обробку численних завдань та визначити їхні переваги й обмеження щодо продуктивності, використання ресурсів і точності розпізнавання.

Основний матеріал

У системах голосової ідентифікації обробка сигналів поділяється на кілька етапів: попередня підготовка сигналу, виділення ознак, обробка та нормалізація ознак, порівняння отриманих характеристик із шаблонами, а також прийняття рішення про ідентичність або відмінність голосів.

На етапі попередньої обробки сигналів здійснюється нормалізація, фільтрація та, за потреби, зменшення шумових компонентів [14]. На етапі виділення ознак проводяться спектральний аналіз, обчислення мел-частотних кепстральних коефіцієнтів (MFCC) та інші обчислювальні операції, необхідні для подальшої класифікації. Запропонований підхід до організації паралельних обчислень розкрито на прикладі обчислення MFCC.

Обчислення MFCC полягає у перетворенні часових сигналів у спектральну форму, що відображає частотну структуру звуку відповідно до сприйняття людським вухом [14-15]. В класичній реалізації першим етапом є фреймування, тобто розбиття сигналу на короткі кадри тривалістю 15-30 мс з перекриттям в 50%, що дозволяє зберегти сталість статистичних характеристик у межах кадру. Після цього на кожен кадр застосовується віконна функція Хеммінга, для зменшення спектрального витоку:

$$x_m[n] = x[n + m \times R] \cdot w[n], \quad n = 0,1,\ldots,N-1, \tag{1}$$

де $w[n]$ – віконна функція, R – крок перекриття кадрів.

Далі обчислюється спектр за допомогою швидкого перетворення Фур'є (FFT):

$$X_m[k] = \sum_{n=0}^{N-1} x_m[n] \log e^{-j2\pi kn/N}, \quad k = 0,\ldots,N-1, \tag{2}$$

Виконується перехід до мел-шкали з подальшим логарифмуванням $log(E)$, де E – енергія спектрального компонента, та дискретним косинусним перетворенням (DCT) для отримання фінальних MFCC-коефіцієнтів. Розрахунок MFCC виконується відповідно до:

$$c_m[n] = \alpha_n \sum_{i=1}^{K} \log E_m[i] \times \cos\left(\frac{\pi n}{K}(i-0.5)\right), \quad (3)$$
$$n = 0, .., L-1,$$

де L – кількість коефіцієнтів MFCC, що зберігають основну спектральну інформацію кадру.

Послідовне виконання описаних операцій для великого обсягу даних створює значні часові затримки, обмежує можливість обробки потокового сигналу та ускладнює інтеграцію алгоритму у високопродуктивні системи реального часу.

Природною властивістю аудіосигналів є відносна незалежність коротких кадрів, що дозволяє організувати паралельне виконання алгоритмів попередньої обробки. Кожен кадр містить спектральну інформацію, обмежену коротким інтервалом часу, що відображає локальні властивості сигналу і може оброблятися незалежно, що дозволяє виконувати FFT, обчислення мел-енергії, логарифмування та DCT одночасно для декількох кадрів. Завдяки незалежності обчислень між кадрами, такий підхід дозволяє організувати паралельну роботу. Проте просте перенесення класичного алгоритму на паралельні архітектури не є достатнім, потрібна адаптація методу, включно з організацією кадрів у батчі або тензори, оптимізацією доступу до пам'яті та синхронізацією результатів для збереження коректної послідовності обчислень. На рис. 1 наведено приклад формування батча розмірністю 3×3×4 із голосових сигналів, представлених у спектральному вигляді.

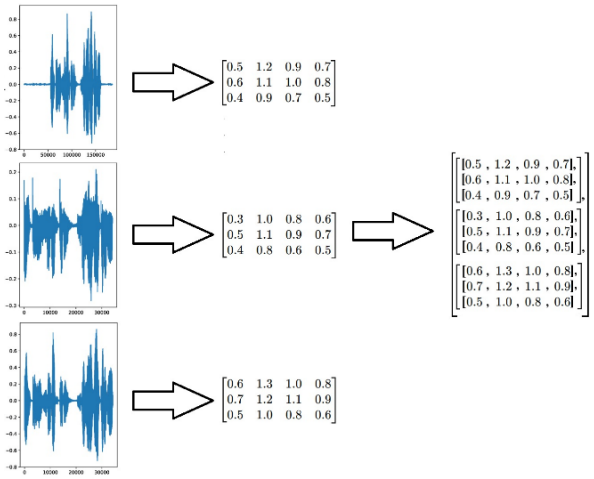

Рис. 1. Об'єднання масивів спектральних ознак голосових сигналів у батч

В процесі адаптації алгоритму кадри формуються у батчі таким чином, щоб забезпечити вирівняний доступ до пам'яті та можливість одночасної обробки кількох кадрів. Це дозволяє організувати дані у вигляді багатовимірних тензорів, де перша вісь відповідає номеру прикладу в батчі, а наступні часовим і частотним вимірам сигналу. Кожен кадр обробляється автономно, включно з обчисленням FFT, мел-енергії, логарифму та DCT, що запобігає блокуванням у потоках.

Результати обробки кадрів синхронізуються у вихідній структурі даних, забезпечуючи правильний порядок MFCC-коефіцієнтів для подальшого аналізу. Така організація дозволяє ефективно використовувати кеш-пам'ять і прискорювачі (GPU, iGPU, TPU), оскільки операції виконуються над цілими блоками даних, а не над окремими зразками, що зменшує накладні витрати на обмін даними. Розмір батчу є важливим фактором: малі батчі (8–16 кадрів) забезпечують швидшу реакцію системи та частіше оновлення моделі, тоді як великі (64–256 і більше) краще завантажують GPU та ефективніше використовують ресурси, але потребують більше пам'яті.

В даній роботі підготовка методу обчислення MFCC до ефективного виконання на CPU здійснювалася з урахуванням можливостей сучасних багатоядерних архітектур та особливостей організації пам'яті, таких як ієрархія кешів, розподіл пам'яті у NUMA-системах, вирівнювання даних і використання регістрового зберігання проміжних результатів. Використання багатопоточності та багатоядерності через бібліотеки Multiprocessing та Joblib дозволило уникнути гонок даних і забезпечити синхронізацію результатів. При цьому методи обчислення були адаптовані для використання можливостей SIMD-векторизації (Single Instruction, Multiple Data): дані організовувалися у суміжні блоки, що дозволяло виконувати однотипні операції над кількома елементами масиву одночасно за допомогою векторних інструкцій CPU, що значно підвищує продуктивність обчислень. Локалізація даних у кешах забезпечувала мінімізацію доступів до повільної основної пам'яті та дозволила ефективно задіяти високопропускні регістрові шини процесора для обробки багатьох кадрів одночасно.

Додатково враховувалася організація пам'яті у багатопроцесорних системах із NUMA для зменшення затримок доступу до віддалених банків пам'яті та підвищення ефективності паралельних обчислень.

Бібліотеки CuPy та PyTorch із підтримкою CUDA забезпечують низькорівневий доступ до обчислювальних блоків, де кожен кадр обробляється багатьма потоками, які групуються у блоки. Така організація дозволяє одночасно виконувати велику кількість операцій над різними кадрами та їхніми частотними компонентами.

Для оптимізації швидкодії особливу увагу приділено використанню пам'яті. Проміжні результати, такі як FFT, мел-енергія та DCT-коефіцієнти, зберігаються у регістрах обчислювальних потоків, що мінімізує затримки доступу до даних. Дані, до яких здійснюється частий доступ, розташовуються у спільній пам'яті, що дозволяє уникнути численних звернень до глобальної пам'яті та зменшує накладні витрати на обмін між потоками. Таке розміщення даних забезпечує ефективне використання кеш-пам'яті та обчислювальних ресурсів під час одночасної обробки кадрів і батчів сигналів, скорочує загальний час обробки та підтримує масштабованість методів для великих наборів голосових даних.

Відмінності в адаптації алгоритму для CPU та GPU зумовлені потребою в організації пам'яті, формуванны батчів і керування потоками для досягнення максимальної продуктивності. На CPU накладні

витрати пов'язані з плануванням процесів і потоків, тоді як на GPU основними джерелами накладних витрат є пересилання даних між процесорами та підготовка багатовимірних масивів для ефективного паралельного виконання обчислень.

Приблизний час виконання алгоритму у паралельному режимі розраховується згідно:

$$T_{пар.} \approx \frac{T_{послід.}}{P} + T_{затрим.}, \quad (4)$$

де $T_{послід.}$ – час послідовного виконання, P – кількість паралельних обчислювальних одиниць, $T_{затрим.}$ – час накладних витрат на синхронізацію та управління пам'яттю.

Для оцінки продуктивності системи необхідно проаналізувати роботу при різних обсягах даних і розмірів батчів, їх вплив на час виконання та ефективність використання ресурсів.

Результати експериментальних досліджень

Експериментальна модель системи обробки голосових сигналів була розгорнута в середовищі Jupyter Notebook. Модель передбачає використання як CPU, так і GPU, включно з інтегрованими GPU, для виконання різних етапів обробки. Це дозволяє контролювати розподіл обчислювальних завдань між апаратними ресурсами та оцінювати ефективність паралельної обробки етапів голосових сигналів залежно від складності та обсягу даних.

У дослідженні для порівняння ефективності паралельних обчислень використовувалися різні обчислювальні вузли. CPU Intel Core i7-13700H (6 продуктивних і 8 ефективних ядер, 20 потоків) забезпечує високу продуктивність на малих та середніх обсягах даних. Інтегрований GPU Intel Iris Xe (96 виконавчих блоків, 768 шейдерних ядер) ефективний для середніх обсягів сигналів, тоді як дискретний GPU на архітектурі Ada Lovelace (3072 CUDA-ядер) обробляє великі набори даних у ресурсномістких задачах ідентифікації користувача.

Для оцінки ефективності обробки голосових сигналів використано набір CHiME [13], що містить понад 50 годин аудіо реальних розмов чотирьох учасників у межах 20 сесій. На його основі сформовано піднабори різного розміру для порівняння продуктивності обчислювальних вузлів і впливу паралельних обчислень.

Кадри аудіосигналу формувалися у батчі та організовувалися у багатовимірні тензори з узгодженим вирівнюванням у пам'яті, що забезпечувало одночасну обробку кількох кадрів та ефективне використання обчислювальних ресурсів. Кожен кадр проходив автономно повний цикл перетворень, включно з FFT, обчисленням мел-енергії, логарифмуванням та DCT, зменшуючи накладні витрати на синхронізацію та підвищуючи продуктивність CPU.

На GPU паралельне виконання алгоритму реалізовувалося через масове однотипне обчислення на великій кількості кадрів, що робило його ефективною платформою для обробки MFCC у батчах. Кадри формувалися у багатовимірні тензори, де перша вісь відповідала номеру прикладу, а наступні часовим і частотним вимірам сигналу, що дозволяло максимально використовувати апаратну структуру GPU. Для досліджуваної системи з розпаралелюванням на GPU експериментально обрано батч розміром 32 кадра.

Таблиця 1 – **Час отримання MFCC**

Кількість записів	Час обробки, с		
	CPU	iGPU	Дискретний GPU
100	9,2	5,1	4,3
500	48,5	22,4	20,7
2500	250,3	102,5	98,2
5000	520,1	198,4	195,3
10000	1048,5	395,2	384,7
100000	10390,6	3712,8	3820,4

В ході першого експерименту оцінено час виконання розрахунку MFCC, що застосовуються на етапі виділення ознак у коротких голосових фрагментах. Експеримент дозволив оцінити здатність CPU обробляти невеликі обсяги даних без залучення GPU та порівняти ефективність обчислень у задачах низького і високого навантаження при обробці голосових сигналів.

Результати, наведені у табл. 1, демонструють, що паралельне виконання на центральному процесорі забезпечує продуктивність лише на невеликих обсягах даних (100-500 записів). При обробці великих обсягів даних (10000-1000000 записів) паралельне виконання на дискретному GPU та інтегрованому GPU забезпечує трикратну перевагу за часом виконання у порівнянні з обчисленнями на центральному процесорі.

У другому експерименті розглянуто застосування спектрального віднімання для пригнічення шумових компонентів голосового сигналу. Даний підхід передбачає аналіз частотних спектрів для кожного фрейму, що суттєво підвищує обчислювальне навантаження навіть при роботі з відносно невеликими наборами даних.

Результати експерименту показують, що паралелізація на базі CPU демонструє задовільну продуктивність для обробки невеликих наборів даних, тоді як збільшення обсягу понад 5000 записів призводить до істотного зростання часу обробки (табл. 2).

Таблиця 2 – **Час обробки даних методом спектрального віднімання**

Кількість записів	Час обробки, с		
	CPU	iGPU	Дискретний GPU
100	10,3	5,5	4,3
500	51,5	23,0	21,3
2500	254,1	105,4	100,2
5000	523,7	205,6	197,8
10000	1079,2	405,7	394,9
100000	10564,3	3780,1	3850,3

Використання інтегрованого та дискретного GPU забезпечує суттєве прискорення обробки спектральних фреймів, що дозволяє ефективно працювати з великими обсягами голосових сигналів та скорочує загальний час виконання ідентифікації користувачів.

Таблиця 3 – **Час обробки даних методом вейвлет-фільтрації**

Кількість записів	Час обробки, с		
	CPU	iGPU	Дискретний GPU
100	41,1	22,5	21,0
500	210,3	103,2	100,5
2500	1050,6	520,3	510,2
5000	2102,6	1032,6	1015,8
10000	4205,7	2060,4	2025,1
100000	42062,3	20780,1	20140,2

При застосуванні вейвлет-фільтрації, яка забезпечує багаторівневий аналіз сигналу в часочастотній області, обчислювальна складність значно зростає через великий обсяг операцій на кожному рівні декомпозиції.

Наведені в табл. 3 результати демонструють, що зростання обчислювального навантаження при обробці голосового сигналу обмежує ефективність використання паралелізації на CPU. Застосування ресурсоємних методів, таких як вейвлет-фільтрація, призводить до значного збільшення часу виконання на CPU та інтегрованих GPU, що доводить перевагу використання дискретного GPU.

Застосування паралельної обробки не впливає на точність роботи системи (відсоток успішної голосової ідентифікації користувача). Це свідчить про те, що підвищення продуктивності системи за рахунок розподілу обчислень між потоками або ядрами не призводить до зниження якості розпізнавання, що є важливим для забезпечення надійності систем голосової ідентифікації при роботі з великими обсягами даних. Формування кадрів у батчі забезпечує вирівняний доступ до пам'яті та впорядковане виконання обчислень, що зменшує час очікування даних і до-зволяє ефективно використовувати ресурси, такі як завантаження ядер CPU або потоків GPU та скоротити загальний час обробки. Важливим фактором є правильний вибір розміру батчу: занадто малий обсяг не забезпечує ефективної паралельності, тоді як надто великий може призводити до перевантаження пам'яті та зниження продуктивності.

Завдяки описаному підходу методи попередньої обробки голосових сигналів можуть виконуватись у реальних умовах ідентифікації користувачів, з одночасним збереженням точності результатів. Це дозволяє обробляти великі набори голосових сигналів та масштабувати систему залежно від апаратних можливостей.

Висновки

Реалізовано паралельне виконання методів обробки голосових сигналів, зокрема з обчисленням мел-частотних кепстральних коефіцієнтів (MFCC), спектральним відніманням для пригнічення шуму та вейвлет-фільтрацією. Проведено порівняльний аналіз продуктивності у системах голосової ідентифікації з використанням технологій паралельних обчислень на багатоядерному центральному процесорі (CPU), інтегрованому графічному процесорі (iGPU) та дискретному графічному процесорі (GPU). Експериментальні сценарії охоплювали широкий спектр обсягів вхідних даних та різних алгоритмів обробки.

Результати досліджень показали, що CPU забезпечує високу продуктивність для обробки невеликих наборів даних і виконання алгоритмів помірної складності, таких як вейвлет-фільтрація, спектральне віднімання та обчислення MFCC, завдяки багатоядерній архітектурі та можливості паралельної обробки без значних накладних витрат на передачу даних. Інтегрований GPU (iGPU) забезпечує можливість паралельної обробки, проте продуктивність при збільшенні обсягів даних зростає обмежено через його архітектурні особливості. Дискретний GPU дозволяє суттєво прискорити обробку великих обсягів даних завдяки масовій паралельній архітектурі та високій пропускній здатності пам'яті, проте при малих наборах даних накладні витрати на синхронізацію потоків і передачу знижують загальну ефективність.

Подальшим розвитком є застосування гібридної моделі, де обчислювальні завдання динамічно розподіляються між CPU та GPU залежно від обсягу та складності обробки. Такий підхід дозволить поєднувати високу швидкодію дискретного GPU при роботі з великими обсягами даних із ефективністю CPU на малих обсягах, зменшуючи накладні витрати на передачу даних та синхронізацію потоків.

СПИСОК ЛІТЕРАТУРИ

1. A. Kumar, S. Jain and M. Kumar, "Deep Learning based Fusion for a Multi-Biometric Identification Using LSTM", 2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET), Ghaziabad, India, 2024, pp. 1-6. https://doi.org/10.1109/ACET61898.2024.10730213.
2. Mykhailichenko I., Ivashchenko H., Barkovska O., Liashenko O., "Application of Deep Neural Network for Real-Time Voice Command Recognition", IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, pp. 1-4. https://doi.org/10.1109/KhPIWeek57572.2022.9916473.
3. Бондаренко М. Е., Іващенко Г. С. Використання послідовності методів попередньої обробки в системах голосової ідентифікації. Системи управління, навігації та зв'язку. Полтава: ПНТУ, 2025. Т. 2 (80). С. 90-96. https://doi.org/10.26906/SUNZ.2025.2.090.
4. B. Gawrych and P. Czarnul, "Performance Assessment of OpenMP Constructs and Benchmarks Using Modern Compilers and Multi-Core CPUs", 2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS), Warsaw, Poland, 2023, pp. 973-978. https://doi.org/10.15439/2023F7822.

5. I. Vasileska, P. Tomšič, L. Kos and L. Bogdanović, "Unveiling Performance Insights and Portability Achievements Between CUDA and SYCL for Particle-in-Cell Codes on Different GPU Architectures", 2024 47th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2024, pp. 1115-1120, DOI: https://doi.org/10.1109/MIPRO60963.2024.10569866.
6. F. Lumpp, H. D. Patel and N. Bombieri, "A Framework for Optimizing CPU-iGPU Communication on Embedded Platforms", 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021, pp. 685-690. https://doi.org/10.1109/DAC18074.2021.9586304.
7. H. Li, J. K. Ng and T. Abdelzaher, "Enabling Real-time AI Inference on Mobile Devices via GPU-CPU Collaborative Execution", 2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Taipei, Taiwan, 2022, pp. 195-204. https://doi.org/10.1109/RTCSA55878.2022.00027.
8. R. M. Fazliddinovich and B. U. Abdumurodovich, "Parallel processing capabilities in the process of speech recognition", 2017 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2017, pp. 1-3. https://doi.org/10.1109/ICISCT.2017.8188585.
9. L. Yu, Y. Ukidave and D. Kaeli, "GPU-Accelerated HMM for Speech Recognition", 2014 43rd International Conference on Parallel Processing Workshops, Minneapolis, MN, USA, 2014, pp. 395-402. https://doi.org/10.1109/ICPPW.2014.59.
10. S. M. Hussain, B. Saritha, B. S. Reddy, C. Srikar, B. Suchitra and G. Purnachandrarao, "DeepVoice: An End-to-End Speaker Recognition System Leveraging Convolutional and Recurrent Neural Networks for Robust Voice Identification", 2025 International Conference on Electronics, AI and Computing (EAIC), Jalandhar, India, 2025, pp. 1-5. https://doi.org/10.1109/EAIC66483.2025.11101389.
11. M. Dyvak and O. Kindzerskyi, "Implementation of Parallel Computation for Identification of Interval Models based on Multi-core Parallelism and CUDA Technology", 2024 14th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2024, pp. 72-76. https://doi.org/10.1109/ACIT62333.2024.10712545.
12. R. Kouatly and T. A. Khan, "Performance of Text-Independent Automatic Speaker Recognition on a Multicore System", in Tsinghua Science and Technology, vol. 29, no. 2, pp. 447-456, April 2024. https://doi.org/10.26599/TST.2023.9010018.
13. P. Foster, S. Sigtia, S. Krstulovic, J. Barker, M. D. Plumbley. "CHiME-Home: A Dataset for Sound Source Recognition in a Domestic Environment", in Proceedings of the 11th Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015, pp. 1-5. https://doi.org/10.1109/WASPAA.2015.7336899.
14. A. Moondra and P. Chahal, "Voice Feature Extraction Method Analysis for Speaker Recognition with Degraded Human Voice", 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 385-388. https://doi.org/10.1109/ICAC3N60023.2023.10541716.
15. Y. A. Wubet and K. -Y. Lian, "Speaker Anonymization for Voice Biometrics Protection Using Voice Conversion and Multi-Target Speaker Voice Fusion", in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 6046-6057, 2025. https://doi.org/10.1109/TIFS.2025.3577023.

Received (Надійшла) 11.07.2025
Accepted for publication (Прийнята до друку) 15.10.2025

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Іващенко Георгій Станіславович – кандидат технічних наук, доцент, доцент кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Heorhii Ivashchenko – Candidate of Technical Sciences, Associate Professor at the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: heorhii.ivashchenko@nure.ua; ORCID Author ID: http://orcid.org/0000-0003-1027-5262;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57217030807.

Бондаренко Максим Едуардович – аспірант кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Maksym Bondarenko – PhD student at the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: maksym.bondarenko@nure.ua; ORCID Author ID: http://orcid.org/0000-0002-2500-7626;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57216159508.

Parallel implementation of voice signal processing methods on multicore CPU and GPU

Maksym Bondarenko, Heorhii Ivashchenko

Abstract. Relevance. Systems such as voice assistants and speaker identification tools, which operate based on voice signal processing, have become increasingly widespread. The performance of such systems depends on the volume of data and operating conditions. Processing large collections of voice signals or ensuring real-time operation requires high-performance computing. Such performance can be achieved with massively parallel systems, including multiprocessor clusters or discrete GPUs. **Object of research** is the organisation of parallel computing processes in voice signal processing tasks using the capabilities of modern processor architectures. **Purpose of the article** is to develop a parallel voice signal processing system with adapted algorithms for a multi-core CPU and an integrated and discrete GPU. **Research results**. Comparative analysis revealed that for small loads (such as voice assistants and personal applications), CPU usage is sufficient to ensure efficient computation with low latency. However, when processing large datasets and performing streaming analytics, the proposed parallel approach, implemented on both the CPU and GPU, reduces execution time by 25-30% compared to a sequential implementation on the CPU. **Conclusions**. Research has shown that parallelism on the CPU is suitable for stages of voice signal processing that require a small amount of computation. At the same time, discrete GPUs can be utilised in stages with intensive computational tasks such as MFCC calculation, spectral subtraction, and wavelet filtering.

Keywords: voice identification, signal processing, feature extraction, normalisation, MFCC, spectral subtraction and wavelet filtering, CPU, GPU, iGPU.