УДК 004.8:004.421.2:519.876.5

АЛГОРИТМІЧНІ МЕТОДИ ПАРАЛЕЛЬНОЇ ОБРОБКИ ВЕЛИКИХ ДАНИХ У СИСТЕМАХ ПРОГНОЗНОЇ АНАЛІТИКИ

Назар Заплатинський, Віталій Фіялковський, Святослав Штогрин, Тарас Квасниця, Андрій Татомир

Львівський національний університет природокористування,
вул. Володимира Великого, 1, м. Дубляни, Львівський р-н, Львівська обл., Україна,
e-mail: hayk. ukr@gmail.com; vitalik. fiyalkovsky@i.ua, sviatoslav. shtohryn@gmail.com, taras1kvasnytsya@gmail.com andrew.tatomyr@gmail.com

https://doi.org/10.32718/agroengineering2025.29.232-241

Заплатинський Н., Фіялковський В., Штогрин С., Квасниця Т., Татомир А. Алгоритмічні методи паралельної обробки великих даних у системах прогнозної аналітики

Стрімке зростання обсягів даних та ускладнення прогнозно-аналітичних моделей зумовлюють необхідність ефективного використання паралельної обробки у сучасних інформаційних системах. Обґрунтовано алгоритмічні та архітектурні підходи до паралельної обробки великих даних у системах прогнозної аналітики з урахуванням вимог продуктивності, масштабованості та адаптивності. Аналітично зіставлено моделі паралелізму, програмні фреймворки і обчислювальні архітектури, а також проаналізовано системні обмеження, що впливають на ефективність прогнозно-аналітичних систем у розподілених і гетерогенних середовищах. Показано, що ізольоване масштабування обчислювальних ресурсів не забезпечує пропорційного зростання продуктивності, а найвищої ефективності досягають гібридні конфігурації, які поєднують різні моделі паралелізму та апаратні прискорювачі. Наукова новизна роботи полягає в систематизації алгоритмічних і архітектурних підходів до паралельної обробки великих даних у прогнозній аналітиці з урахуванням адаптивності та системних обмежень, а також у формуванні узагальненого аналітичного підходу до поєднання програмних платформ і спеціалізованих обчислювальних архітектур. Отримані результати можуть бути використані при проєктуванні високопродуктивних прогнозно-аналітичних систем. Окреслено напрями подальшого розвитку паралельних прогнозно-аналітичних систем, зокрема в контексті інтеграції методів машинного навчання, потокової обробки даних і динамічного керування обчислювальними ресурсами. Особливу увагу приділено питанням узгодження алгоритмічних рішень із характеристиками апаратної платформи з метою мінімізації накладних витрат і підвищення енергоефективності обчислень. Запропонований аналітичний підхід може слугувати методологічною основою для побудови адаптивних високопродуктивних систем прогнозної аналітики в умовах змінних навантажень і неоднорідних обчислювальних середовищ.

Ключові слова: алгоритмічні методи, прогнозна аналітика, GPU-обчислення, масштабованість, штучний інтелект, машинне навчання, розподілені обчислення.

Zaplatynskyi N., Fiialkovskyi V., Shtrohryn S., Kvasnytsia T., Tatomyr A. Algorithmic methods of parallel processing of big data in predictive analytics systems

The rapid growth of data volumes and the increasing complexity of predictive and analytical models necessitate the efficient use of parallel processing in modern information systems. The aim of this paper is to substantiate algorithmic and architectural approaches to parallel processing of big data in predictive analytics systems, taking into account performance, scalability, and adaptability requirements. The study provides an analytical comparison of parallelism models, software frameworks, and computing architectures, and examines system-level constraints that affect the efficiency of predictive analytics systems in distributed and heterogeneous environments. It is shown that isolated scaling of computational resources does not ensure proportional performance gains, whereas the highest efficiency is achieved by hybrid configurations that combine multiple parallelism models with hardware accelerators. The scientific novelty of the study lies in the systematization of algorithmic and architectural approaches to parallel big data processing in predictive analytics with consideration of adaptability and system constraints, as well as in the formulation of an integrated analytical approach to combining software platforms and specialized computing architectures. The obtained results can be applied to the design of high-performance predictive analytics systems. The article also outlines directions for the further development of parallel predictive and analytical systems, in particular in the context of integrating machine learning methods, stream data processing, and dynamic management of computational resources. Special attention is paid to aligning algorithmic solutions with the characteristics of the hardware platform to minimize overhead costs and improve the energy efficiency of computations. The proposed analytical approach can serve as a methodological basis for building adaptive high-performance predictive analytics systems under conditions of variable workloads and heterogeneous computational environments.

Keywords: algorithmic methods, predictive analytics, GPU computing, scalability, artificial intelligence, machine learning, distributed computing.

Постановка проблеми. Інтенсивне зростання обсягів даних та ускладнення прогнозно-аналітичних моделей зумовлюють необхідність використання паралельної обробки як базового механізму забезпечення продуктивності сучасних інформаційних систем. Водночас практичне впровадження паралельних алгоритмів у прогнозній аналітиці супроводжується низкою системних обмежень, пов'язаних із масштабованістю, адаптивністю обчислювальних архітектур та неоднорідністю обчислювальних середовищ.

Аналіз сучасних наукових публікацій свідчить, що більшість досліджень зосереджена або на окремих програмних фреймворках обробки великих даних, або на спеціалізованих апаратних прискорювачах, без комплексного урахування їх взаємодії у складі прогнозно-аналітичних систем. У результаті залишається недостатньо формалізованим питання вибору та поєднання алгоритмічних і архітектурних рішень з урахуванням адаптивності, масштабованості та продуктивнісних компромісів у реальних сценаріях прогнозування.

У цьому контексті наукова проблема полягає у відсутності цілісного аналітичного підходу до оцінювання ефективності паралельної обробки великих даних у прогнозно-аналітичних системах, який би дозволяв узгоджено враховувати алгоритмічні моделі паралелізму, особливості обчислювальних архітектур та системні обмеження розподілених середовищ.

Відповідно, дослідницьке завдання дослідження полягає в обґрунтуванні та систематизації алгоритмічних і архітектурних підходів до паралельної обробки великих даних у прогнозній аналітиці з метою підвищення продуктивності, масштабованості та адаптивності таких систем.

Аналіз останніх досліджень і публікацій. Сучасні дослідження у сфері прогнозної аналітики для великих даних концентруються на поєднанні архітектурних рішень та алгоритмічних методів паралельної обробки. У працях відзначено перехід від суто інфраструктурного масштабування до алгоритм-орієнтованих підходів, спрямованих на підвищення продуктивності без втрати точності [2; 4].

На системному рівні наголошено на ролі хмарних платформ, MapReduce та Spark, що забезпечують ефективне зберігання і паралельну обробку потокових та пакетних даних [1; 9]. На рівні алгоритмів перевагу мають гібридні схеми розпаралелювання та асинхронні методи синхронізації параметрів, які знижують накладні витрати [6; 7].

Дослідження на багатоядерних та GPU-системах підтверджують ефективність CUDA-орієнтованих рішень і стратегій, що поєднують *data* та *task parallelism* [3; 12]. Також відзначається значення алгоритмів, які добре масштабуються (градієнтний бустинг, розподілене SGD) та гібридних ML-підходів [15; 11].

Окремо досліджено питання безпеки і приватності у розподіленому навчанні, де інтегруються *federated learning* і методи диференційної приватності [17]. Водночас невивченими залишаються проблеми стандартів бенчмаркінгу, автоматичного вибору стратегії розпаралелювання та балансування між якістю прогнозу й ресурсними обмеженнями [4; 8].

Аналіз сучасних наукових публікацій свідчить, що проблема паралельної обробки великих даних активно досліджується у контексті окремих програмних фреймворків, алгоритмічних моделей або спеціалізованих апаратних архітектур. Водночас у більшості робіт ці підходи розглядаються ізольовано, без урахування їх взаємодії у складі прогнозно-аналітичних систем. Недостатньо висвітленими залишаються питання узгодженого вибору моделей паралелізму та обчислювальних архітектур з урахуванням адаптивності, системних обмежень і характеру прогнозних задач. Крім того, у наявних дослідженнях обмежено представлено аналітичні зіставлення, що дозволяли б оцінювати компроміси між продуктивністю, масштабованістю та енергоефективністю у гетерогенних і розподілених середовищах.

Зазначені прогалини зумовлюють необхідність комплексного аналітичного підходу до оцінювання паралельної обробки великих даних у прогнозно-аналітичних системах, що й визначає напрям і зміст дослідження.

Постановка завдання. Наше завдання – обґрунтування ефективних алгоритмічних і архітектурних підходів до паралельної обробки великих даних у прогнозно-аналітичних системах з урахуванням вимог продуктивності, масштабованості та адаптивності.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Проаналізувати основні моделі паралелізму (паралелізм даних, паралелізм задач, гібридні моделі) та визначити їх придатність для задач прогнозної аналітики.
2. Зіставити сучасні програмні фреймворки і апаратні архітектури паралельної обробки великих даних із позицій продуктивності, масштабованості та адаптивності.

3. Виявити ключові системні обмеження паралельної обробки у розподілених і гетерогенних обчислювальних середовищах (I/O, комунікаційні накладні витрати, сублінійне масштабування).

4. Проаналізувати вплив вибору обчислювальної архітектури (CPU, GPU, TPU, FPGA, edge / fog, хмарні середовища) на продуктивність і ефективність прогнозно-аналітичних систем на основі узагальнення сучасних досліджень.

5. Сформувати узагальнені рекомендації щодо поєднання алгоритмічних і архітектурних рішень для побудови високопродуктивних прогнозно-аналітичних систем.

Виклад основного матеріалу. Паралельна обробка великих даних ґрунтується на поєднанні концепцій розподілених обчислень, багатопроцесорних архітектур та спеціалізованих алгоритмічних підходів, спрямованих на ефективне використання обчислювальних ресурсів. Її теоретичний фундамент становлять моделі паралелізму, що формалізують принципи розподілу завдань між незалежними виконавчими елементами. Серед ключових моделей вирізняють:

– Модель паралелізму даних (Data Parallelism), що передбачає розподіл великих масивів даних на сегменти з подальшою незалежною обробкою кожного сегмента окремим процесором або вузлом. Такий підхід типовий для задач класифікації, кластеризації та аналізу потокових даних.

– Модель паралелізму задач (Task Parallelism), у межах якої розподіляються не дані, а різні обчислювальні функції, що можуть виконуватися одночасно. Це забезпечує підвищення продуктивності при багатокомпонентних прогнозних моделях.

– Гібридні моделі паралелізму, які поєднують обидва підходи, що особливо важливо для систем прогнозної аналітики з різнорідними даними та багаторівневими алгоритмами.

На рис. 1 представлено узагальнену архітектуру системи прогнозної аналітики з паралельною обробкою великих даних, яка відображає послідовність перетворення даних від джерел надходження до формування прогнозних результатів, а також взаємодію між інфраструктурним, алгоритмічним та прикладним рівнями системи.

Другим фундаментальним блоком є архітектури обчислювальних систем, серед яких найпоширеніші:

– SIMD-архітектури (Single Instruction, Multiple Data), орієнтовані на одночасне виконання однієї інструкції над численними елементами даних. Цей підхід застосовують у графічних процесорах (GPU) для масивних паралельних обчислень.

Рис. 1. Узагальнена архітектура системи прогнозної аналітики з паралельною обробкою великих даних
Fig. 1. Generalized architecture of a predictive analytics system with parallel processing of big data

– MIMD-архітектури (Multiple Instruction, Multiple Data), що дозволяють одночасне виконання різних інструкцій над різними наборами даних. Такі архітектури реалізуються у високопродуктивних кластерних та хмарних середовищах.

– SMP (Symmetric Multiprocessing) та MPP (Massively Parallel Processing) системи, які відрізняються рівнем інтеграції процесорів і масштабованістю апаратної інфраструктури.

Особливе місце у розвитку паралельної обробки посідає концепція MapReduce, запропонована компанією Google як універсальна парадигма для розподілених обчислень над великими даними. MapReduce формалізує обчислювальний процес у вигляді двох ключових функцій:

– *Map*, що розбиває початковий набір даних на пари «ключ-значення» і розподіляє їх між вузлами системи;

– *Reduce*, що агрегує результати проміжної обробки та формує кінцевий результат.

Ця концепція забезпечила підґрунтя для побудови сучасних фреймворків, зокрема Hadoop та Spark, які поєднують теоретичну простоту моделі з високим рівнем масштабованості на практиці.

Тож теоретичні основи паралельної обробки великих даних інтегрують математичні моделі паралелізму, архітектурні принципи багатопро-

цесорних систем та алгоритмічні парадигми. Вони створюють науково-методологічну базу для розробки інноваційних рішень у сфері прогнозної аналітики, де ефективність обробки безпосередньо визначає якість прогнозів і практичну значущість результатів [2; 5].

Теоретичні підходи описують архітектурні можливості паралельних систем, тоді як їхнє реальне функціонування залежить від алгоритмів, що керують розподілом даних і ресурсів. У цьому контексті важливим є аналіз алгоритмічних методів та стратегій, які дозволяють адаптувати обчислювальні процеси до специфіки великих даних і завдань прогнозної аналітики.

Ефективність сучасних систем прогнозної аналітики значною мірою визначається тим, наскільки оптимально організовано алгоритмічне керування обчислювальними процесами. Одним із ключових завдань є оптимізація розподілу навантаження, яка передбачає рівномірне розподілення обчислювальних операцій між усіма вузлами або потоками. Неефективне розподілення може призвести до дисбалансу: одні вузли простоюють, тоді як інші перевантажуються, що негативно впливає на загальну продуктивність.

Для запобігання цьому застосовують методи балансування ресурсів, які реалізуються як у вигляді статичних стратегій (де завдання попередньо фіксуються за конкретними вузлами), так і динамічних (де розподіл адаптивно змінюється залежно від стану системи та інтенсивності надходження даних). Динамічні підходи демонструють вищу ефективність у випадках із нерегулярними або непередбачуваними потоками даних.

Окрему групу становлять алгоритми для обробки потокових і нерегулярних даних. Потокова аналітика потребує механізмів обробки даних у режимі реального часу з мінімальною затримкою, що досягається завдяки методам мікропакетної обробки (micro-batching) та спеціалізованим структурам даних, здатним до швидкого оновлення. Нерегулярні дані, що характеризуються високою варіативністю і непередбачуваними схемами доступу, потребують алгоритмів з адаптивною складністю та можливістю багатоваріантного розгалуження обчислювальних процедур [1; 7].

У табл. 1 систематизовано основні алгоритмічні методи та стратегії, що застосовуються у паралельній обробці великих даних у системах прогнозної аналітики. Вона демонструє взаємозв'язок між типом алгоритмічного підходу, його характеристиками та сферами практичного застосування, а також висвітлює ключові переваги й обмеження.

Аналіз свідчить, що жодна зі стратегій не є універсальною: кожна має як сильні сторони, так і обмеження, які необхідно враховувати у виборі методів для конкретних завдань прогнозної аналітики. Оптимальним підходом є комбіноване використання методів – поєднання динамічного балансування з потоковими алгоритмами та адаптивними механізмами обробки нерегулярних даних. Це дозволяє забезпечити стабільність, масштабованість і точність прогнозів у складних інформаційних середовищах [8].

Таблиця 1. Алгоритмічні методи та стратегії паралельної обробки
Table 1. Algorithmic methods and parallel processing strategies

Напрям	Характеристика	Приклади застосування	Переваги	Обмеження
Оптимізація розподілу навантаження	Рівномірний розподіл обчислювальних задач між вузлами	Розподілені обчислювальні кластери	Зменшення часу виконання, зниження простоїв	Складність у випадку непередбачуваних потоків
Методи балансування ресурсів	Статичні та динамічні стратегії розподілу ресурсів	Хмарні сервіси, Big Data-платформи	Адаптивність до змін середовища	Потреба у складних механізмах моніторингу
Алгоритми потокової обробки	Обробка даних у реальному часі з мінімальною затримкою	FinTech, IoT, моніторинг мережевого трафіку	Висока оперативність, релевантність прогнозів	Високі вимоги до апаратних ресурсів
Алгоритми обробки нерегулярних даних	Робота з непередбачуваними структурами даних	Соціальні мережі, біоінформатика	Гнучкість, адаптивність	Зниження продуктивності при великих обсягах

Алгоритмічні методи задають логіку й ефективність паралельної обробки, однак їхня практична реалізація можлива лише за наявності відповідних платформ і систем. Саме вони створюють середовище, у якому обчислювальні процеси набувають масштабованості, стійкості та здатності працювати з гетерогенними потоками даних [4, с. 194].

Серед провідних технологій у цій сфері варто виокремити *Hadoop*, який заклав основу сучасної екосистеми Big Data завдяки реалізації парадигми *MapReduce* та розподіленого файлового сховища (HDFS). Відповідно до результатів практичних досліджень застосування Hadoop у розподілених середовищах [9], його головною перевагою є здатність до зберігання та обробки величезних обсягів даних у кластерах із тисяч вузлів, тоді як продуктивність у режимах реального часу залишається обмеженою.

Згідно з результатами експериментальних досліджень продуктивності розподілених систем [16], Apache Spark поєднує розподілену обробку з механізмом in-memory computing, що значно зменшує затримки та робить його ефективним інструментом для потокової аналітики, машинного навчання і побудови прогнозних моделей.

Як показано в експериментальних порівняльних дослідженнях потокових фреймворків [14; 16], Apache Flink здатний забезпечувати безперервний аналіз потоків завдяки низькій латентності та підтримці складних обчислювальних графіків, що є критично важливим для прогнозної аналітики у реальному часі.

Окремий напрям становлять GPU-обчислення, де графічні процесори виконують масивні паралельні операції, особливо ефективні для навчання нейронних мереж та глибинної аналітики. Їхнє використання значно прискорює роботу з багатовимірними прогнозними моделями, однак потребує спеціалізованих бібліотек і високих енергетичних витрат.

Нарешті, *хмарні сервіси (AWS, Azure, Google Cloud)* забезпечують динамічну масштабованість і доступність інструментів для прогнозної аналітики, інтегруючи Hadoop-, Spark- і Flink-кластери з готовими сервісами машинного навчання та аналітичними модулями.

У табл. 2 відображено найбільш поширені платформи та технології, що застосовуються для реалізації паралельної обробки у прогнозній аналітиці. Вона дозволяє порівняти їхні ключові характеристики, переваги та обмеження, а також визначити оптимальні сфери використання.

Таблиця 2. Системи та платформи прогнозної аналітики
Table 2. Predictive analytics systems and platforms

Платформа / технологія	Ключові характеристики	Переваги	Обмеження (адаптивність)	Типові сфери застосування
Hadoop	MapReduce + HDFS, пакетна обробка	Масштабованість, відмовостійкість	Висока латентність, низька адаптивність до real-time	Архіви даних, пакетна аналітика
Apache Spark	In-memory computing, підтримка MLlib	Висока швидкодія, ефективність для ML	Вимоги до RAM, адаптивність залежить від конфігурації	Потокова аналітика, прогнозування
Apache Flink	Потокова обробка у реальному часі	Низька латентність	Висока системна адаптивність, складність налаштування	IoT, фінансовий моніторинг
GPU-обчислення	Масивний паралелізм (SIMD)	Значне прискорення ML / DL	Обмежена алгоритмічна адаптивність, високі енергозатрати	Глибинне навчання
TPU	Апаратна оптимізація тензорних операцій	Висока продуктивність та енергоефективність	Низька універсальність, жорстка спеціалізація	Масштабне навчання нейромереж
FPGA	Реконфігурована апаратна логіка	Висока енергоефективність, апаратна адаптивність	Складність програмування, високі інженерні вимоги	Спеціалізовані прогностичні системи
Edge / Fog computing	Обробка даних на периферії мережі	Мінімальна латентність, висока адаптивність до контексту	Обмежені обчислювальні ресурси	IoT, промислова аналітика, кіберфізичні системи
Хмарні сервіси (AWS, Azure, GCP)	Інтеграція Big Data та ML	Гнучка масштабованість	Залежність від провайдера	Бізнес-аналітика, SaaS

Зіставлення характеристик, наведених у табл. 2, свідчить, що платформи та апаратні архітектури, які застосовуються у системах прогнозної аналітики, суттєво відрізняються за рівнем адаптивності та характером паралельної обробки. Програмні платформи типу Hadoop, Spark і Flink забезпечують адаптивність переважно на інфраструктурному та системному рівнях за рахунок динамічного розподілу ресурсів і масштабування, тоді як апаратні прискорювачі реалізують адаптивність на рівні обчислювальної архітектури. Зокрема TPU орієнтовані на високопродуктивне виконання тензорних операцій у задачах глибинного навчання, що забезпечує високу швидкодію за умови обмеженої алгоритмічної гнучкості. Натомість FPGA характерні можливістю апаратної реконфігурації, що дозволяє адаптувати обчислювальні структури під конкретні алгоритми прогнозної аналітики та вимоги до енергоефективності. Отож, результати зіставлення підтверджують доцільність комбінування програмних платформ і спеціалізованих апаратних архітектур для досягнення балансу між продуктивністю, масштабованістю та адаптивністю систем прогнозної аналітики [9; 16].

Аналіз продуктивності сучасних фреймворків паралельної обробки великих даних показує суттєві відмінності у часових та обчислювальних характеристиках залежно від обраної архітектури та типу навантаження. Зокрема результати опублікованих досліджень свідчать, що Apache Spark за рахунок in-memory обробки забезпечує прискорення виконання ітеративних аналітичних задач у 3–10 разів порівняно з класичною реалізацією MapReduce у Hadoop, особливо в задачах машинного навчання та прогнозування [9; 14]. Для потокових сценаріїв Apache Flink демонструє середню латентність на рівні десятків мілісекунд, що є критично важливим для фінансового моніторингу та IoT-аналітики, тоді як Spark Streaming працює з латентністю близько сотень мілісекунд.

Використання апаратних прискорювачів суттєво підвищує продуктивність навчання прогнозних моделей. Зокрема застосування GPU-обчислень у задачах глибинного навчання дозволяє скоротити час тренування моделей у 5–20 разів порівняно з CPU-кластерами аналогічної конфігурації, тоді як TPU забезпечують додаткове підвищення продуктивності для тензорних операцій за рахунок апаратної спеціалізації [3; 12]. У свою чергу, використання FPGA демонструє виграш у продуктивності та енергоефективності у спеціалізованих сценаріях прогнозної аналітики, де обчислювальні графи можуть бути адаптовані під конкретні алгоритми.

Отже, наведені числові приклади підтверджують, що вибір фреймворку та апаратної архітектури безпосередньо впливає на продуктивність прогнозно-аналітичних систем. Найвищі показники досягаються у гібридних конфігураціях, що поєднують програмні платформи обробки великих даних із спеціалізованими апаратними прискорювачами, забезпечуючи баланс між швидкодією, масштабованістю та економічною доцільністю.

Важливо зазначити, що зі зростанням обсягів даних у системах прогнозної аналітики виникає фундаментальна проблема: масштабування не гарантує лінійного зростання продуктивності. Теоретично розподілена обробка передбачає, що збільшення кількості вузлів у кластері має пропорційно зменшувати час виконання обчислень. Подібний ефект сублінійної масштабованості зафіксовано в експериментальних дослідженнях розподілених обчислювальних систем [13; 15], де додавання ресурсів призводить до зростання накладних витрат на комунікацію, синхронізацію та управління даними.

При переході від гігабайтного до петабайтного масштабу обробки даних з’являються такі ефекти:

6. Дисбаланс між обчисленнями та I/O-операціями. Для великих обсягів даних ключовим вузьким місцем стає не процесорна потужність, а швидкодія дискових підсистем та мережевих каналів. Так, у кластерах Hadoop чи Spark значна частина часу витрачається на переміщення даних між вузлами, а не на їхню безпосередню обробку.

7. Зростання латентності при потоковій аналітиці. У системах реального часу (Apache Flink, Kafka Streams) навіть мілісекундні затримки накопичуються і призводять до зсуву прогнозних моделей, що особливо критично у фінансових або медичних застосуваннях.

8. Ефект «data skew» (нерівномірний розподіл даних). У разі нерівномірної сегментації даних частина вузлів обробляє значно більший обсяг, що знижує ефективність паралельності та подовжує загальний час виконання завдання.

Для ілюстрації впливу масштабування на продуктивність наведено залежність прискорення обробки S (N) від кількості вузлів кластера (рис. 2). Графік демонструє сублінійний характер масштабування: зі зростанням N приріст продуктивності зберігається, однак ефективність використання

додаткових ресурсів поступово знижується через накладні витрати на комунікацію, синхронізацію та переміщення даних між вузлами. Це узгоджується з практикою побудови прогнозно-аналітичних систем, де після досягнення певного масштабу інфраструктури вузькими місцями стають мережеві та I/O-ресурси, а також дисбаланс розподілу даних (data skew). Отже, досягнення приросту продуктивності зі збільшенням кількості вузлів потребує не лише додавання обчислювальних ресурсів, а й оптимізації алгоритмів планування, забезпечення data locality та зменшення синхронізаційних бар'єрів.

Навіть при використанні гібридних інфраструктур (CPU+GPU, локальні кластери + хмари) існують фізичні межі масштабованості:

1. Пропускна здатність мережі не зростає пропорційно кількості вузлів. У масштабах понад 1000 серверів мережеві колізії та перевантаження комутаторів стають критичними.
2. Обмеження оперативної пам'яті призводить до вимушеного використання swap-операцій, що кратно знижує продуктивність у задачах in-memory computing (наприклад, у Spark MLlib).
3. Енергетичні бар'єри: для GPU-ферм характерне надмірне споживання енергії (сотні кВт/год на кластер), що робить безконтрольне масштабування економічно та екологічно нераціональним [12; 13].

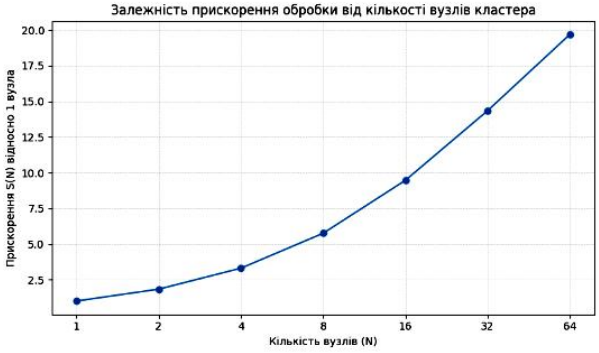

Рис. 2. Залежність прискорення обробки від кількості вузлів кластера

Fig. 2. Dependence of processing speedup on the number of cluster nodes

Примітка: S (N) – прискорення відносно одного вузла

Подолання зазначених проблем можливе через багаторівневі оптимізаційні стратегії, що охоплюють алгоритмічний, інфраструктурний та системний рівні. На алгоритмічному рівні важливим напрямом є використання асинхронних методів обчислень, зокрема підходів на основі asynchronous SGD та parameter server architecture, які знижують залежність від глобальної синхронізації та забезпечують стійкість до затримок у розподілених середовищах. Не менш перспективним є застосування апроксимаційних алгоритмів та методів sketching, що дозволяють аналізувати великі масиви даних із прийнятним рівнем статистичної точності без необхідності повного їхнього опрацювання, що суттєво скорочує часові витрати.

На інфраструктурному рівні ключову роль відіграє data locality-aware scheduling, коли завдання спрямовуються на ті обчислювальні вузли, які зберігають локальні копії даних. Це дає змогу мінімізувати обсяг мережевих комунікацій і підвищити ефективність розподіленої обробки. Додаткові можливості відкриває використання гетерогенних обчислювальних середовищ, що поєднують CPU, GPU та FPGA. Такий підхід дозволяє адаптивно розподіляти завдання відповідно до архітектурної оптимальності, наприклад, передаючи матричні операції на GPU, тоді як нерегулярні обчислення ефективніше виконуються на CPU.

На системному рівні доцільне застосування методів кешування проміжних результатів, наприклад, у середовищі Apache Spark (через механізми persist / cache), що дає змогу уникати повторних обчислень при багатократному зверненні до одних і тих самих даних. Важливим засобом оптимізації також є використання колонарних форматів збереження (Parquet, ORC), які знижують I/O-навантаження при вибірковому доступі до даних завдяки більш ефективному стисненню та організації структур. У хмарних середовищах до цього додається можливість автоматизованого масштабування ресурсів, включно з autoscaling та serverless computing, що дозволяє системі адаптувати кількість обчислювальних потужностей до поточного навантаження та забезпечує економічну ефективність [15, с. 779].

Отже, проблема масштабованості у прогнозній аналітиці не є виключно технічним викликом. Вона вимагає комплексного підходу, що поєднує оптимізацію алгоритмів, удосконалення архітектури інфраструктури та адаптивні моделі використання ресурсів. Ефективність таких систем визначається не так обсягом доступної апаратної потужності, як якістю інженерних рішень у сфері розподілених обчислень.

Попри значні зусилля, спрямовані на оптимізацію масштабованості та продуктивності систем

прогнозної аналітики, ключовим індикатором їхньої практичної цінності залишається не швидкодія сама по собі, а *якість одержуваних прогнозів*. Ефективність алгоритмів паралельної обробки даних безпосередньо корелює з точністю побудованих моделей, оскільки саме спосіб організації обчислень визначає рівень втрат інформації, швидкість ітераційного навчання та стійкість моделей до шумів у даних. Отож, між продуктивністю інфраструктури та результативністю прогнозної аналітики – складний взаємозв'язок, який не зводиться до лінійної залежності.

З одного боку, збільшення пропускної здатності та зменшення латентності системи забезпечує можливість багаторазового тренування моделей на великих обсягах даних, що підвищує статистичну значущість і робастність результатів. Наприклад, використання in-memory обчислень у Spark чи GPU-акселерації дозволяє проводити ітеративне навчання моделей у реальному часі, зменшуючи ризик недоадаптації або втрати релевантності прогнозу. З іншого боку, надмірна оптимізація у бік швидкодії може призвести до застосування апроксимативних методів обробки (sketching, sampling, approximate query processing), що, хоча й знижує часові витрати, водночас спричиняє деградацію точності прогнозних моделей, особливо в контексті складних нелінійних залежностей [6, с. 581].

Критичним аспектом є також стійкість моделей до розподілених похибок, що виникають у процесі паралельних обчислень. Синхронізаційні збої, втрати пакетів у мережі чи асинхронне оновлення параметрів у нейронних мережах можуть призвести до виникнення *parameter staleness*, який знижує збіжність алгоритмів оптимізації. У цьому контексті продуктивність системи набуває не лише кількісного, а й якісного виміру: правильна архітектура обчислювального процесу прямо впливає на метричні показники моделей (MAE, RMSE, AUC-ROC тощо) [10, с. 1525].

Отже, у прогнозній аналітиці алгоритмічна ефективність паралельної обробки даних постає не самоціллю, а інструментом підвищення точності прогнозів. Справжня оптимізація полягає у досягненні балансу між продуктивністю інфраструктури та достовірністю результатів моделювання. Це передбачає інтеграцію методів апаратного прискорення, алгоритмічних оптимізацій і статистично обґрунтованих процедур перевірки моделей, які спільно забезпечують стійкість і надійність прогнозної аналітики в умовах обробки великих даних.

Встановлений взаємозв'язок між продуктивністю паралельних обчислень та точністю прогнозних моделей набуває особливого значення тоді, коли результати аналітики застосовуються у конкретних прикладних сферах. Адже саме від стабільності обчислювальної інфраструктури та алгоритмічної ефективності залежить здатність системи забезпечити своєчасні та достовірні прогнози, що у практичних умовах трансформуються у фінансові виграші, зниження ризиків або оптимізацію управлінських рішень. Логічним наступним кроком є розгляд практичних сценаріїв застосування алгоритмічних методів паралельної обробки у прогнозній аналітиці, які демонструють реальну цінність цих технологій.

Зокрема у сфері фінансової аналітики високопродуктивні системи дозволяють здійснювати *high-frequency trading* та моніторинг ринкових аномалій у реальному часі, забезпечуючи конкурентні переваги банків і біржових платформ. Як показано в прикладних дослідженнях прогнозування попиту та поведінки користувачів [14; 17], паралельні алгоритми дають змогу оперативно аналізувати великі обсяги транзакційних і сенсорних даних, що є критичним для ритейлу, логістики та енергетичного сектору.

У контексті аналізу поведінки користувачів масштабовані моделі дозволяють виявляти приховані закономірності у великих потоках цифрових слідів (clickstream data), що застосовується у маркетинговій персоналізації та побудові рекомендаційних систем [14]. Нарешті, у сфері управління ризиками паралельна обробка даних забезпечує виявлення системних загроз та прогнозування кризових сценаріїв у фінансових і виробничих системах, де часова затримка між обробкою та прийняттям рішень може мати критичні наслідки [17].

Тож саме прикладні кейси підтверджують, що алгоритмічна оптимізація паралельних обчислень є не лише теоретичною проблемою, а й практичною необхідністю, без якої сучасні системи прогнозної аналітики втрачають свою ефективність у реальних умовах використання.

Перспективність паралельних алгоритмічних підходів у прогнозній аналітиці визначається не лише їхньою здатністю обробляти великі масиви даних у режимі реального часу, а й потенціалом до подальшої інтеграції з методами машинного навчання (ML) та штучного інтелекту (AI). Сучасні виклики, пов'язані зі складністю моделей та динамічністю даних, висувають вимогу переходу від суто інфраструктурних оптимізацій до когнітивно орієнтованих систем, де алгоритмічна ефективність прямо підпорядковується якості прогнозів і здатності моделей до адаптації.

Одним із ключових напрямів є впровадження *інтегративних підходів ML / AI* у розподілені системи. Використання *глибинних нейронних мереж* (DNN) у поєднанні з високопродуктивними обчислювальними платформами (GPU-кластери, TPU, FPGA) дозволяє реалізовувати масштабоване навчання моделей у режимі *distributed deep learning*. Прикладом є архітектури на основі parameter server чи ring-allreduce, які забезпечують ефективне узгодження параметрів між тисячами обчислювальних вузлів. Це відкриває можливості побудови прогнозних моделей нового покоління, здатних виявляти приховані нелінійні закономірності у даних високої розмірності [3, с. 1849].

Іншим перспективним вектором є розвиток гібридних підходів до обробки, що поєднують різні парадигми паралельних обчислень. Наприклад, інтеграція batch-орієнтованих технологій (Spark, Hadoop) із системами streaming-аналітики (Flink, Kafka Streams) формує середовище *lambda- ma kappa-архітектур*, де ретроспективний аналіз поєднується з оперативним прогнозуванням. У такий спосіб система зберігає баланс між високою точністю та низькою латентністю. Додаткові можливості забезпечує federated learning, що дозволяє здійснювати колективне навчання моделей без централізованого збирання даних, підвищуючи як масштабованість, так і безпекові аспекти прогнозної аналітики.

Отже, отримані результати та проведений аналіз дозволяють окреслити конкретні напрями подальшого розвитку алгоритмічних і архітектурних рішень у системах прогнозної аналітики. Зокрема актуальне дослідження адаптивних алгоритмів паралельної обробки, здатних динамічно змінювати стратегії розподілу обчислень залежно від характеристик даних і поточного навантаження. Перспективним напрямом є також поглиблений аналіз гетерогенних обчислювальних середовищ із поєднанням CPU, GPU, FPGA та спеціалізованих прискорювачів з метою підвищення енергоефективності й масштабованості прогнозно-аналітичних систем. Окремої уваги потребують питання оптимізації масштабування у хмарних і edge / fog середовищах з урахуванням обмежень латентності та пропускної здатності мережі. Подальший розвиток тематики пов’язаний із експериментальною валідацією запропонованих підходів на реальних наборах даних та аналізом компромісів між точністю прогнозування й обчислювальними витратами, що дозволить сформувати обґрунтовані рекомендації щодо проєктування високопродуктивних прогнозно-аналітичних систем, а також уточнити межі їх ефективного застосування в умовах масштабних і динамічних обчислювальних середовищ.

Висновки. У дослідженні розглянуто проблему підвищення ефективності паралельної обробки великих даних у прогнозно-аналітичних системах в умовах зростання обсягів інформації, ускладнення алгоритмічних моделей і неоднорідності обчислювальних середовищ. Дослідження спрямоване на узгоджений аналіз алгоритмічних і архітектурних підходів, що визначають продуктивність, масштабованість та адаптивність таких систем.

У межах виконання поставлених завдань проаналізовано основні моделі паралелізму (паралелізм даних, паралелізм задач і гібридні моделі) та обґрунтовано їх придатність для завдань прогнозної аналітики залежно від типу даних і характеру обчислень. Зіставлено сучасні програмні фреймворки і апаратні архітектури паралельної обробки великих даних, що дозволило виявити їхні функціональні відмінності, рівні адаптивності та обмеження у контексті прогнозно-аналітичних застосувань.

На основі узагальнення результатів сучасних досліджень проаналізовано вплив вибору обчислювальної архітектури (CPU, GPU, TPU, FPGA, edge / fog і хмарні середовища) на продуктивність і ефективність прогнозно-аналітичних систем. Показано, що використання гетерогенних та гібридних конфігурацій дозволяє досягти кращого балансу між швидкодією, масштабованістю та енергоефективністю порівняно з ізольованим застосуванням окремих архітектур. Окрему увагу приділено аналізу системних обмежень паралельної обробки у розподілених середовищах, зокрема сублінійній масштабованості, накладним витратам на комунікацію, I/O-операціям та ефекту нерівномірного розподілу даних. Наведена емпірична ілюстрація залежності продуктивності від кількості обчислювальних вузлів підтверджує, що нарощування ресурсів без урахування системних факторів не забезпечує пропорційного зростання швидкодії.

Узагальнення наведених у статті числових показників свідчить, що застосування in-memory обробки в Apache Spark забезпечує прискорення ітеративних аналітичних та прогнозних задач у середньому у 3–10 разів порівняно з класичною реалізацією MapReduce у Hadoop. Використання апаратних прискорювачів у задачах глибинного навчання дозволяє скоротити час тренування моделей у 5–20 разів при переході від CPU-орієнтованих конфігурацій до GPU-рішень, тоді як TPU демонструють додатковий виграш у тензор-

них операціях за рахунок апаратної спеціалізації. Водночас збільшення кількості обчислювальних вузлів у кластерах супроводжується сублінійним зростанням продуктивності, що підтверджує необхідність алгоритмічної оптимізації та адаптивного керування ресурсами замість простого нарощування інфраструктури.

Наукова новизна роботи полягає в систематизації алгоритмічних і архітектурних підходів до паралельної обробки великих даних у прогнозній аналітиці з урахуванням адаптивності, масштабованості та системних обмежень, а також у формуванні узагальненого аналітичного підходу до поєднання програмних платформ і спеціалізованих апаратних прискорювачів у складі прогнозно-аналітичних систем.

Отримані результати підтверджують виконання поставленого у статті дослідницького завдання та можуть бути використані як теоретико-методична основа для проєктування і оптимізації високопродуктивних прогнозно-аналітичних систем, орієнтованих на роботу з великими та динамічними наборами даних.

Бібліографічний список

1. Bu L., Zhang H., Xing H., Wu L. Research on parallel data processing of data mining platform in the background of cloud computing. *Journal of Intelligent Systems*. 2021. Vol. 30. P. 479–486. DOI: 10.1515 / jisys – 2020–0113.

2. Dritsas E., Trigka M. Exploring the Intersection of Machine Learning and Big Data: A Survey. *Machine Learning and Knowledge Extraction*. 2025. Vol. 7, No. 1. P. 13. DOI: 10.3390 / make7010013.

3. Ghimire A., Amsaad F. A Parallel Approach to Enhance the Performance of Supervised Machine Learning Realized in a Multicore Environment. *Machine Learning and Knowledge Extraction*. 2024. Vol. 6, No. 3. P. 1840–1856. DOI: 10.3390 / make6030090.

4. Jamarani A., Haddadi S., Sarvizadeh R. et al. Big data and predictive analytics: A systematic review of applications. *Artificial Intelligence Review*. 2024. Vol. 57. P. 176–253. DOI: 10.1007 / s10462–024–10811–5.

5. Kolisetty V. V., Rajput, D. S. A review on the significance of machine learning for data analysis in big data. *Jordanian Journal of Computer and Information Technology (JJCIT)*. 2020. Vol. 6. P. 155–171.

6. Laouni D. Dynamic Distributed and Parallel Machine Learning algorithms for big data mining processing. *Data Technologies and Applications*. 2021. No. 4. P. 558–601. DOI: 10.1108 / dta – 06–2021–0153.

7. Laouni D., Bensaber D. A., Adjoudj R. Big Data analytics for prediction: parallel processing of the big learning base with the possibility of improving the final result of the prediction. *Information Discovery and Delivery*. 2018. Vol. 46, No. 2. P. 147–160. DOI: 10.1108 / IDD–02–2018–0002.

8. Naeem M., Jamal T., Diaz-Martinez J., Butt S. A., Montesano N., Tariq M. I., De-la Hoz-Franco E., De-La-Hoz-Valdiris E. Trends and future perspective challenges in big data. *Advances in Intelligent Data Analysis and Applications: Proceedings of the Sixth Euro-China Conference on Intelligent Data Analysis and Applications* (Arad, Romania, 15–18 October 2019). Berlin / Heidelberg: Springer, 2022. P. 309–325.

9. Natesan P., Sathishkumar V. E., Mathivanan S., Venkatasen V., Maheshwari J., Jayagopal P., Shaikh Muhammad A. A Distributed Framework for Predictive Analytics Using Big Data and MapReduce Parallel Programming. *Mathematical Problems in Engineering*. 2023. Article ID 6048891. 10 p. DOI: 10.1155 / 2023 / 604889.

10. Oo M. C. M., Thein T. An efficient predictive analytics system for high dimensional big data. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34. P. 1521–1532. DOI: 10.1016 / j. jksuci. 2019.09.001.

11. Raghavendra S. Scalability of Data Science Algorithms: Empowering Big Data Analytics. *Journal of Artificial Intelligence and Computing Techniques*. 2024. Vol. 1. P. 1–9.

12. Rakhimov M., Ochilov M., Javliev S., Nasimov R. Analysis and Possibilities of Parallel Approach in Big Data Processing. *ICFNDS '24: Proceedings of the 8th International Conference on Future Networks & Distributed Systems*. 2024. P. 20–25. DOI: 10.1145 / 3726122.3726126.

13. Rakhimov M., Zaripova D., Javliev S., Karimberdiyev J. Deep learning parallel approach using CUDA technology. *AIP Conference Proceedings*. 2024. Vol. 3244, No. 1. 11 p. DOI: 10.1063 / 5.0241439.

14. Silva J., Hernández Palma H., Niebles Núñez W., Ovallos-Gazabon D. Parallel Algorithm for Reduction of Data Processing Time in Big Data. *Journal of Physics: Conference Series*. 2020. Vol. 1432, No. 1. 10 p. DOI: 10.1088 / 1742–6596 / 1432 / 1 / 012095.

15. Tamilselvan K., M. N. S., Saranya A., Abdul Jaleel D., Rajani Kanth T. V., Govardhan S. D. Optimizing data processing in big data systems using hybrid machine learning techniques. *International Journal of Computational and Experimental Science and Engineering*. 2025. Vol. 11, No. 1. P. 775–782. DOI: 10.22399 / ijcesen. 9361.

16. Tyagi A. K., G, R. Machine learning with big data. *Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM*) (Amity University Rajasthan, Jaipur, India, 26–28 February 2019). Jaipur: Amity University Rajasthan, 2019.

17. Xu R., Baracaldo N., Joshi J. Privacy-preserving machine learning: Methods, challenges and directions. *arXiv preprint*. 2021. arXiv: 2108.04417.

Стаття надійшла 20.02.2025