УДК 004.7, 004.75, 004,85

ОГЛЯД АКТУАЛЬНИХ ШЛЯХІВ ПРИСКОРЕННЯ РОБОТИ З НАВЧАЛЬНИМИ ДАНИМИ У СИСТЕМАХ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ МОВЛЕННЯ

DOI 10 36994 / 2788- 5518- 2022-02-04-24

Шульга Ю.А. *Відкритий міжнародний університет розвитку людини «Україна»,Київ,Україна. yuhim1308@gmail.com*

Забара С.С., *д.т.н., проф..Відкритий міжнародний університет розвитку людини «Україна», Київ, Україна. staszabara@gmail.com*

Анотація: *В умовах збільшення кількості голосових зразків для обробки основним завданням є пришвидшення роботи із навчальними даними та збереження швидкодії та відмовостійкості систем розпізнавання вцілому, що стає доступним при використанні розподілених та децентралізованих систем. Використання багатохмарних підходів дозволяє отримувати переваги від використання кожної з них та отримуючи масштабованість, більшу кількість обчислень за одиницю часу та зменшення часу очікування на результат.*

Ключові слова: *розпізнавання мовлення, розподілені системи, децентралізовані системи, глибинне навчання.*

REVIEW OF CURRENT WAYS OF ACCELERATING WORK WITH TRAINING DATA IN AUTOMATIC SPEECH RECOGNITION SYSTEMS

Yukhym Shulha. *Open International University of Human Development "Ukraine", Kyiv, Ukraine. yuhim1308@gmail.com*

Stanislav Zabara, *Dr.habil., Prof. Open International University of Human Development "Ukraine", Kyiv, Ukraine. staszabara@gmail.com*

Abstract: *The training data and preserve the speed and fault tolerance of recognition systems as a whole, which becomes available when using distributed and decentralized systems. To process a large number of requests, it is necessary to keep the initial database up-to-date, taking into account trends in system usage and received new or abnormal requests. For the development of such systems, both logical or topological and algorithmic approaches are used. Each of them has its own shortcomings and advantages, which need to be studied for use in each system to obtain optimal results. The use of multi-cloud approaches allows you to get the benefits of using each of them and getting scalability, a greater number of calculations per unit of time, and a reduction in waiting time for the result. For a preliminary analysis of features and potential complex architectural solutions when working with state-of-the-art approaches, already existing studies were reviewed to identify the necessary improvements and the possibility of application in a wide range of systems. The combination of acceleration and availability is one of the most important because the use of the most modern algorithms may require the use of specific hardware. Extensive research uses specialized equipment of the same type connected by a high-bandwidth local network, which in practical designs will require the use of a multidimensional architectural approach, which will require a balanced use of the geographical location of computing centers in order to reduce delays between system elements to balance the obtained acceleration and potentially possible delays in the transmission of information between elements located in different treatment centers.*

Key words: *speech recognition, distributed systems, deep learning.*

Вступ

Розвиток система автоматичного розпізнавання в умовах збільшення кількості голосових зразків вимагає адаптації процесу розпізнавання для зберігання автентичних та оброблених голосових записів. Децентралізація та розподілення є одними із найкращих способів забезпечення

не тільки для збереження автентичності й недопущення підміни голосових зразків, а ще і для прискорення роботи системи вцілому. Важливо створити умови для ефективного розподіленого навчання, а також щоби алгоритми навчання могли збігатися з великим розміром пакета. Стратегія розвитку систем такого роду має містити декілька основних етапів. Перший етап повинен дослідити основні гіперпараметри (напр. швидкість навчання, об'єм навчальної бази) це необхідно для забезпечення належної точності, використовуючи, у роботі пакети різних розмірів. Наступний етап більш практичний і потребує практичного тестування стратегій (синхронну, асинхронну, асинхронну децентралізовану паралельну стохастичного градієнтного спуску й гібридну). Під час роботи з децентралізованими системами розпізнавання мовлення експерименти показали, що використання асинхронного децентралізованого паралельного стохастичного градієнтного спуску дає можливість працювати з розміром пакету, що втричі більший за розмір пакету для синхронного стохастичного градієнтного спуску. Дослідження показують, що цей розвиток досить перспективний і показує добру масштабованість і суттєве пришвидшення роботи моделі, однак таке прискорення можливе в разі використання однотипного спеціалізованого обладнання, що, може, накласти певні обмеження на практичне використання таких методів. Нівелюють таку потребу потенційно набагато продуктивніші квантові обчислювальні кластери, а також модульний підхід до створення обчислювального кластеру, який, може, бути легко масштабований шляхом залучення нових обчислювальнихх клієнтів, на які, може, бути розподілені задачі з обчислення. Хоча останній підхід не є повністю децентралізованим, а розподіленим його можна зробити децентралізованим, використовуючи, блокчейн підхід у розробленні.

Виконання досліджень

Системи автоматичного розпізнавання мовлення з моменту своєї появи демонстрували суттєвий та стабільне зростання та впровадження у велику кількість галузей. Особливо привабливі системи автоматичного розпізнавання стали після початку їх доступності у форматі ПЗ, як сервіс, а в часи пандемії ріст впровадження систем автоматичного розпізнавання мовлення склав 23 %. Розпізнавання мовлення — це одна з перших галузей, де методи глибинного навчання дають змогу одержувати суттєво кращі результати за менший термін необхідний для виконання. Для того, аби отримати репрезентативні результати важливо комбінувати підходи та стратегії роботи із системами автоматичного розпізнавання мовлення для забезпечення необхідної продуктивності. Найвагомішим елементом, що забезпечує продуктивну роботу системи є навчальні дані. Обсяг тренувальних даних для систем розпізнавання мовлення для типових задач складає від тисячі до десятків тисяч годин, а вузькоспеціалізовані задачі використовують до мільйона годин. Такий великий об'єм даних потрібно результативно та безпечно обробити. Обробка такого масиву даних на одному обчислювальному елементі, може, зайняти роки тож логічним кроком є застосування підходів, за допомогою, яких час на обробку скоротиться. Підходи розподіляються на алгоритмічні та структурні. Найчастішим та ефективним підходом є розподілене навчання. Розподілене навчання широко використовується у сферах машинного зору, адже має подібні до систем автоматичного розпізнавання мовлення алгоритмічні та структурні схеми. Це важливе уточнення адже ці сфери мають подібну структуру тому схеми пришвидшення роботи не завжди суміжні, а найчастіше не взаємозамінні, тобто, пряме запозичення технік, може, не дати бажаного результату.

Логічні підходи розподіляються відповідно до топології. їх є три: централізована, кооперативна й некооперативна (рис.1).

Централізована топологія містить у собі центр обробки або ж злиття даних, що виконує задачі з отримання даних, розподілу завдань та збору результатів. Завдяки наявності в структурі такого центру прямого спілкування між агентами не відбувається, що робить таку схему досить простою, розповсюдженою та досить просто масштабованою. Однак наявність централізованого вузла керування вносить у систему вразливість та створює вузьке місце в пропускній здатності системи, що й доводять дослідження різних груп дослідників. Але зважаючи на широку розповсюдженість розроблені практичні підходи, які дають змогу максимально нівелювати негативні впливи. Однак для роботи із розподіленим, а тим паче із децентралізованими підходами ця схема не може бути застосована. Через це для оптимізації систем застосовують кооперативна й некооперативна. У

некооперативній топології всі агенти незалежні один від одного, що дозволяє їм реалізувати власні очікування. Кожен агент самостійно ділиться даними та своєю поведінкою. А в кооперативній кожен з агентів має додатковий зв'язок, що дозволяє отримувати більше інформації про завдання та стани в програмному вигляді реалізований у вигляді балансувального алгоритму для кращого розподілу завдань. Крім, того, у такий спосіб збільшується стійкість до несправностей та відмов, збільшити конфіденційність даних, збільшити потужність розподіленої обробки інформації в мережі.

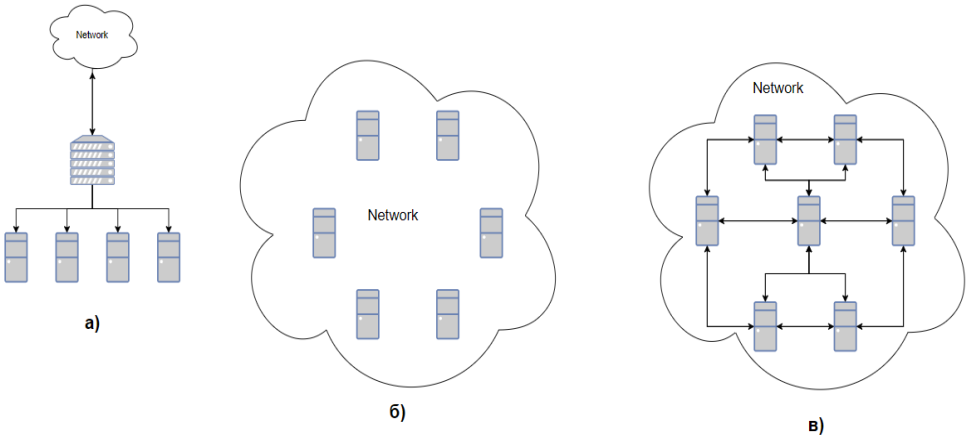

Рис.1. Різновид топологій: а) Централізована; б) Децентралізована не кооперативна; в) Децентралізована кооперативна

Структурні підходи містять у собі логічні підходи, однак, спрямовані на впровадження в практичну реалізацію для отримання найкращих показників. Усі структурні стратегії, що можуть бути застосовані до систем автоматичного розпізнавання можна виокремити в такі групи :

- *Паралельний доступ до даних* та *паралельне підключення до моделі*:

Паралелізм даних розподіляє мініпакети даних між кількома учнями (наприклад, GPU або CPU), причому кожен учень має копію моделі. Обчислення виконується для кожного учня паралельно з використанням даних перед тим, як система певним чином агрегує статистичні дані. Іноді модель занадто велика, щоби поміститися в пам'ять одного учня. За цієї умови паралелізм моделі використовується для розподілу моделі між учнями водночас кожен учень має, лише часткову модель. Вихідні дані одного учня використовуються, як вхідні дані іншого учня для проведення обчислень повної моделі. Хоча паралелізм моделей використовується в деяких сценаріях, паралелізм даних є домінуючою розподіленою стратегією на практиці в сучасній спільноті розподіленого навчання.

- *Одно вузлова* та *багато вузлова структура*:

Під вузлом розуміється фізичний пристрій (наприклад, сервер) у комп'ютерній мережі. Розподілене навчання на одному вузлі — це коли всі учні (наприклад, графічні процесори) залишаються на одній машині зі спільною пам'яттю. Спілкування між учнями є надійним і, може, забезпечити помірне прискорення. Тобто такий варіант усе ще розумний підхід для розподіленого навчання нейронних мереж. Очевидно, що він має обмеження щодо можливості масштабування, а кількість учнів обмежена апаратною конфігурацією машин. Розподілене навчання з декількома вузлами має кілька машин (вузлів) для формування хмари або кластера, де учні використовують спільну пам'ять для локального зв'язку всередині вузла та передачу повідомлень через мережу для зв'язку між вузлами. Це стало нормою для останніх масштабних розподілених тренувань.

- *Централізоване* та *децентралізоване навчання*:

Централізоване розподілене навчання покладається на центральний вузол, і всі учні спілкуються з центральним вузлом, лише для оновлення моделі та передачі результатів. Децентралізоване налаштування не має центрального вузла, і всі учні утворюють мережу певної топології (наприклад, кільце). Усі учні перебувають у рівному становищі щодо обчислень і

спілкування. Централізоване розподілене навчання зазвичай використовується на практиці в багатьох програмах машинного навчання. Однак це створює велике вузьке місце у вигляді центрального вузла, коли кількість учнів велика. З іншого боку, децентралізоване розподілене навчання має перевагу в пропускній здатності зв'язку та стає все більш популярним, хоча і проєктування та адаптація системи займає набагато більше ресурсів.

- *Синхронне* та *асинхронне навчання*:

Алгоритми розподіленого навчання можуть працювати в синхронному або асинхронному режимі. У синхронному режимі обчислення та оновлення моделі синхронізуються одне з одним. Оновлення моделі виконується після того, як усі учні завершать обчислення локального завдання. Однак в асинхронному режимі синхронізація між обчисленням і оновленням моделі видаляється, й учні отримують свої мініпакети за потреби залежно від статусу своїх обчислень і швидкості зв'язку, що значно скорочує час простою, порівнюючи, із синхронним режимом.

З практичної точки зору створення повністю децентралізованої системи вимагає значних зусиль, а певні процеси, що притаманні роботі зі звуковими сигналами повноцінно децентралізувати неможливо тому найоптимальнішим є використання розподілених систем. Основні зусилля такої системи спрямовані на розподіл процесу розпізнавання на процеси за можливістю паралельного виконання. У кінцевому результаті, виходить, система, що складається з таких частин:

- Паралельна частина, яка, може, виконуватися паралельно, наприклад, градієнтне обчислення.
- Послідовна частина, яку не можна розпаралелити. Наприклад, підсумовування градієнтів або моделей, може, бути виконано, лише після того, як градієнти або моделі будуть на місці.
- Комунікаційна частина, яка передає інформацію між обчислювальними ресурсами, наприклад, передача градієнта/ваги між учнями.

Для оцінювання потенційного прискорення такої системи використовують закон Амдала застосований до найбільш використовуваного алгоритму при розподіленому навчанні, а саме стохастичного градієнтного спуску(англ. Stochastic gradient descent, SGD). Припускаючи, що вартість зв'язку дорівнює нулю, то згідно з законом Амдала прискорення паралельної програми обмежене$\frac{1}{1-p}$, де p є частиною розпаралелюваної частини програми. Для алгоритму SGD паралельна частина домінує над послідовною. Отже, p дуже близьке до 1, і в принципі можна досягти дуже високих прискорень. Як наслідок, основним обмежуючим чинником для досягнення лінійного прискорення в цьому випадку є вартість зв'язку.

Типова система розподіленого навчання, як зображено на рис. 1, має такі апаратні компоненти (рис.2):

- сховище даних
- пам'ять, процесори (CPU/GPU)
- мережу.

Передача даних відбувається за такий спосіб: кожен учень завантажує вхідні дані зі сховища даних з обмеженням пропускної здатності сховища в основну пам'ять. Він проводить попередню обробку даних із допомогою центрального процесора перед тим, як відправити їх на графічний процесор через шину передачі даних для навчання моделі. Коли обчислення градієнта завершено, градієнти надсилаються до центрального процесора або іншим учням згідно з обмеженням пропускної здатності мережі. Типова пропускна здатність систем зберігання коливається залежно від типу

- 1–10 МБ/с (мережева файлова система) до 100 МБ/с (жорсткі диски, HDD),
- 300–500 МБ/с (твердотільний накопичувач, SSD), до кількох ГБ/с (SSD з енергонезалежною пам'яттю Express (NVMe)).
- Типова пропускна здатність основної пам'яті становить кілька десятків ГБ/с.
- Типова пропускна здатність шини даних CPU-GPU коливається від 16 ГБ/с в одну сторону (наприклад, PCI-e 3-го покоління) до 50 ГБ/с в одну сторону (наприклад, IBM Power 9 Nvlink).

- Типова пропускна здатність мережі коливається від 100 МБ/с (наприклад, 1 Гбіт Ethernet) до 10 ГБ/с (наприклад, 100 Гбіт Ethernet, RDMA).

Кластер HPC високого класу (наприклад, суперкомп’ютер SUMMIT) зазвичай оснащений NVMe SSD, NVlinks і 100Gb Ethernet (або вище) для забезпечення швидкого зв’язку. З боку обчислень ключовим вирішальним чинником є FLOPS (англ. Floating Point Operations Per Second, FLOPS), Типові процесори високого класу працюють зі швидкістю від сотень FLOPS до 1 FLOPS, а типові процесори високого класу

Графічні процесори працюють на 10 TFLOPS або вище. Системні програмісти використовують паралельне програмування, яке зазвичай досягається з допомогою багатопоточності, щоби перекривати зв’язок з обчисленнями. У синхронному режимі,завантаження та обробка даних можуть накладатися на градієнтне обчислення. В асинхронному режимі всі шляхи зв’язку (тобто завантаження даних, попередня обробка даних, передача даних CPU-GPU і передача градієнтів/ваг) можуть перекриватися градієнтними обчисленнями. У такі спроби покращення мають на меті досягти ідеального перекриття між комунікацією та обчисленнями. На практиці спілкування, які не можуть бути перекриті обчисленнями, є обмежуючим чинником у досягненні лінійного прискорення. Одним із важливих завдань у розробленні розподіленого навчання є максимізація перекриття між ними.

Для розподіленого навчання використовують алгоритм градієнтного спуск та різні його варіації, а саме градієнтний спуск, синхронний градієнтний спуск, асинхронний градієнтний спуск та гібридний варіант градієнтного спуску, а також різні набори навчальних даних. Аналізуючи дослідження можна зробити висновок, що для роботи із розподіленими системами оптимальним вибором буде асинхронний децентралізований паралельний стохастичний градієнтний спуск (рис.3). Експерименти з алгоритмами навчання дають змогу виявити вузькі місця системи адже менша кількість навчальних даних не дозволить навчити систему так, щоби виявити вузьке місце, а більший дозволить пристосувати роботу до вузьких місць, а інша алгоритмічна структура не буде використовувати апаратну частину системи в об’ємах достатніх для виявлення проблем та вираховування потенційного приросту. Станом на зараз у літературі є ґрунтовні дослідження, які використовують асинхронний децентралізований паралельний стохастичний градієнтний спуск та показують суттєвий приріст швидкодії за умови використання градієнтного спуску.

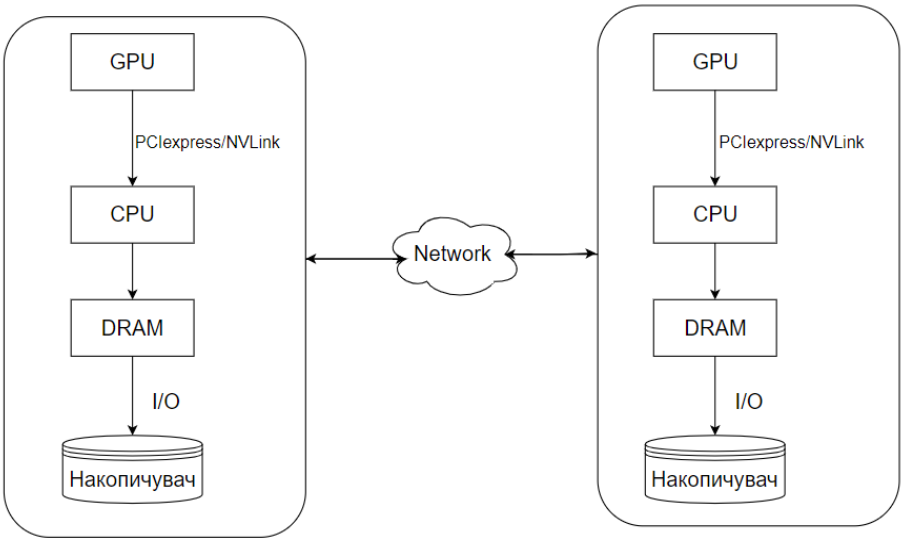

Рис.2. Компоненти та потоки даних у високопродуктивному обчислювальному середовищі для розподіл еного навчання

Варто зазначити, що дослідження проводилися на спеціальному обладнанні, що не потребує додаткових операцій із пристосування до роботи з обладнанням, а цей чинник позитивно впливає, як на швидкодію так і на точність результатів. У дослідженнях використовувалися навчальні бази

тривалістю до 2000 годин та 128 графічних ядер Nvidia V100. Результати досліджень показують результати прискорень до 50 раз (табл.1), однак у такому варіанті важливо забезпечити високу швидкодію між елементами адже відставання доставки пакету до одного із компонентів означатиме не повне або ж взагалі неможливе виконання операції (табл.1).

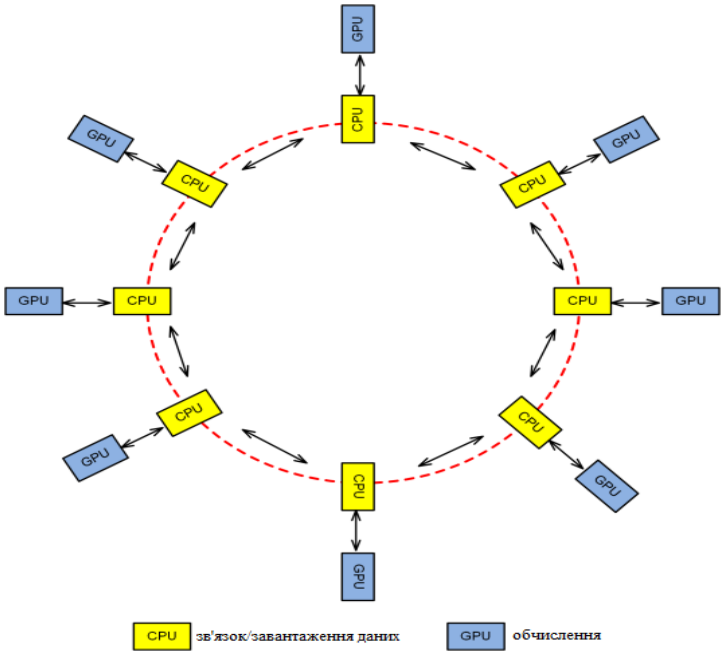

Рис.3. Ілюстрація впровадження ADPSGD, усі учні утворюють кільце

Кожен учень має обчислення локального градієнта, що виконується на графічному процесорі, усереднення моделі з іншими учнями та завантаження даних на центральний процесор. Два процеси здійснюються одночасно.

Для побудови практичних систем у перелічених дослідницьких роботах цікавлять показники: час прискорення та показник помилок на слово. Ці показники цікавлять через прямий вплив на побудову архітектури системи вцілому адже час прискорення показує, скільки потенційно часу можна одержати для виконання інших операцій (рис.4) та дати прискорення в роботі кінцевому користувачу також важливий цей показник тим, що в разі виникнення несправностей дасть змогу перевести на роботу на резервний елемент системи (в разі наявності такого) або ж відновити роботу в штатному режимі та дати користувачу результат виконання, а показник помилок на слово показник важливий тим, що саме його мінімальне значення дає змогу відрізків, тож кінцевий користувач одержить бажаний результат без тривалого очікування та небажаного спотворення аудіозапису. Ці дослідження показують певні особливості проєктування систем розпізнавання. Під час впровадженні таких обчислювальних елементів у гібридну або ж багатохмарну інфраструктурну схему варто враховувати необхідність забезпечення каналів комунікації з високою швидкодією для отримання прискорення від використання таких обчислювальних елементів. Потрібно звернути увагу на критичну необхідність резервування, як мережевих з’єднань так і всіх елементів системи адже стійкість системи до поломок дасть змогу гарантовано одержувати бажані результати.

К-ть учнів	Розмір пакету	Загальний розмір	WER%		Час (год)	Прискорення
			SWB	CH		
1	128	128	7,5	13,0	121,96	
1	256	256	7,5	13,0	99,0	1.0x
1	512	512	7,5	13,1	82,06	1.2x
16	512	8,192	7,4	13,3	5,88	16.8x
32	256	8,192	7,6	13,1	3,60	27.5x
64	128	8,192	7,5	13,3	2,28	43.4x
128	64	8,192	7,7	13,3	1,98	50.x

Для більш повного розуміння прискорення потрібно проаналізувати і втрати, які можуть виникати під час роботи із даними. Як було зазначення для максимального прискорення втрати мусять бути відсутні або ж мінімальні. Стійкість до втрат важливий параметр адже в реальних умовах повна відсутність втрат, може, вимагати складних архітектурних рішень. Для порівняння втрат ADPSGD під час роботи з даними в дослідженнях використовують SSGD тобто синхронний стохастичний градієнтний спуск .Таке порівняння показує перевагу асинхронного виконання перед синхронним у швидкості, адже в разі використання SSGD учні обчислюють градієнт і чекають, поки буде обчислено всі градієнти, потім модель оновлюється та знову розповсюджується всім учням, чого не відбувається в асинхронному режимі адже кожен учень отримує новий градієнт після завершення обчислення, але синхронний варіант має більшу лояльність до втрат, хоча й поступається в прискоренні .

Також маю зазначити, що ці вищезгадані дослідження доводять раціональність використання платформних підходів під час розробки чи проєктування систем розпізнавання, що забезпечуватиме масштабованість і адаптивність та позитивно впливатиме на безпеці. Такі варіанти обчислювальних систем уже реалізовані практично, основна особливість їх є не використання спеціальних обчислювальних нод, а створення таких, використовуючи, певну кількісті обчислювальних елементів об’єднаних в одну ноду, що сумарно дають таку ж обчислювальну потужність, що й системи, які використовувались у дослідженнях. У такому підході основними плюсами є суттєво нижчий рівень вартості та простота в масштабуванні, однак, основними питанням є забезпечення швидкодії між елементами та час виконання операції. У разі роботи з простою обробкою попередньо збереженої навчальної бази теоретично можна знехтувати незначним збільшенням часу на виконання з огляду на меншу вартість. Однак робота в режимі реального часу або ж із великими об’ємами використання такого підходу повинно бути досліджене.

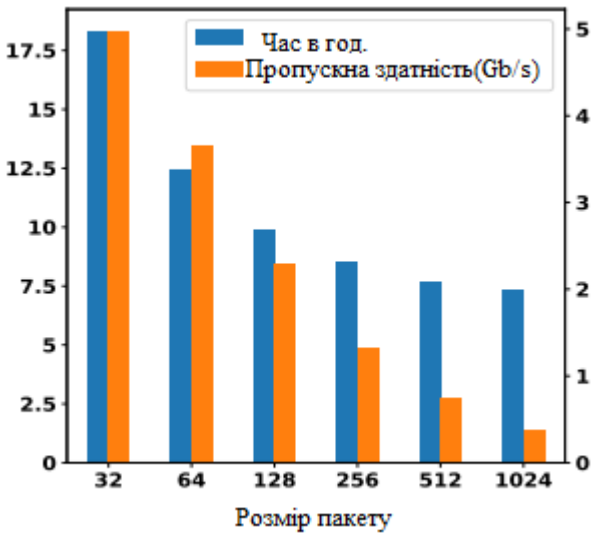

Рис.4. Мінімальна пропускна здатність залежно від розміру пакету.

Висновок

Описані дослідження показують суттєвий приріст у продуктивності роботи із навчальними даними в лабораторних умовах, що теоретично із невеликою похибкою, може, бути отриманий і від реальної системи розпізнавання. Однак є певні необхідні умови, які роблять можливим отримання прискорення. Перенесення такого способу роботи на обчислювальні елементи менш продуктивні, але об'єднані між собою можуть показувати приріст за умови розподілення задачі відповідно до потужності кожного з елементів. Такий підхід може вимагати окремого балансувального модуля та має бути вивчений так само, як і доцільність такого варіанту роботи адже центральний балансувальний компонент системи є елементом притаманним для централізованої топології, хоча й не впливає безпосередньо на спосіб спілкування, отримання та передачі учнями, а більш коректне розподілення завдань, потенційно, може, дати додатковий приріст у швидкодії системи.

References

1. G Hinton, L Deng, D Yu, G Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, T. N Sainath, and B Kingsbury, «Deep neural networks for acoustic modeling in speech recognition,» IEEE Signal Processing Maganize, pp. 82–97, November 2012.
2. G Saon, G Kurata, T Sercu, K Audhkhasi, S Thomas, D Dimitriadis, X Cui, B Ramabhadran, M Picheny, L.-L Lim, B Roomi, and P Hall, «English conversational telephone speech recognition by humans and machines,» in Interspeech, 2017.
3. W Xiong, J Droppo, X Huang, F Seide, M. L Seltzer, A Stolcke, D Yu, and G Zweig, «Toward human parity in conversational speech recognition,» IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 12, pp. 2410–2423, Dec 2017.
4. P Goyal, P Dollar, R. B Girshick, P Noordhuis, L Wesolowski, ´ A Kyrola, A Tulloch, Y Jia, and K He, «Accurate, large minibatch SGD: training imagenet in 1 hour,» CoRR, vol. abs/1706.02677, 2017.
5. M Cho, U Finkler, S Kumar, D. S Kung, V Saxena, and D Sreedhar, «Powerai DDL,» CoRR, vol. abs/1708.02188, 2017.
6. Y You, I Gitman, and B Ginsburg, «Scaling SGD batch size to 32k for imagenet training,» CoRR, vol. abs/1708.03888, 2017.
7. X Jia, S Song, W He, Y Wang, H Rong, F Zhou, L Xie, Z Guo, Y Yang, L Yu, T Chen, G Hu, S Shi, and X Chu, «Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes,» CoRR, vol. abs/1807.11205, 2018.
8. W Wen, C Xu, F Yan, C Wu, Y Wang, Y Chen, and H Li, «Terngrad: Ternary gradients to reduce communication in distributed deep learning,» in NIPS'2017, pp. 1509–1519. Curran Associates, Inc., 2017.
9. F Seide, H Fu, J Droppo, G Li, and D Yu, «1-bit stochastic gradient descent and application to data-parallel distributed training of speech dnns,» in Interspeech 2014, September 2014.
10. D Amodei(et.al.), «Deep speech 2: End-to-end speech recognition in english and mandarin,» in ICML'16. 2016, pp. 173–182, PMLR.

11. X Lian, W Zhang, C Zhang, and J Liu, «Asynchronous decentralized parallel stochastic gradient descent,» in ICML, 2018.
12. W Zhang, S Gupta, X Lian, and J Liu, «Staleness-aware async-sgd for distributed deep learning,» in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, 2016, pp. 2350–2356.
13. P Patarasuk and X Yuan, «Bandwidth optimal all-reduce algorithms for clusters of workstations,» J. Parallel Distrib. Comput., vol. 69, pp. 117–124, 2009.
14. Baidu, Effectively Scaling Deep Learning Frameworks, Available at https://github.com/baidu-research/ baidu-allreduce.
15. Nvidia, NCCL: Optimized primitives for collective multiGPU communication, Available at https://github.com/ NVIDIA/nccl.