УДК 517.958:532.72

doi.org/10.15407/fmmit2023.36.087

Багаторозрядна арифметика у послідовній, паралельній та квантовій моделях обчислень

Валерій Задірака[1], Андрій Терещенко[2], Інна Швідченко[3]

[1] академік НАН України, д. ф.-м. н., Інститут кібернетики імені В.М. Глушкова НАН України, просп. Академіка Глушкова, 40, 03680, Київ, e-mail: zvk140@ukr.net
[2] к. ф.-м. н., докторант, Інститут кібернетики імені В.М. Глушкова НАН України, e-mail: teramidi@ukr.net
[3] к. ф.-м. н., п. н. с., Інститут кібернетики імені В.М. Глушкова НАН України, e-mail: inetsheva@gmail.com

У роботі розглянуто різні моделі обчислень для реалізації операцій багаторозрядної арифметики. Поділ на послідовну, паралельну та квантові моделі обчислень відбувається для врахування особливостей архітектури пристроїв, на яких будуть виконуватися програми. У роботі наведені особливості реалізації алгоритмів для різних моделей обчислень. Наведені основні критерії ефективності при обчисленні складності для різних моделей обчислення. Відмічено обмеження, які необхідно враховувати.

Ключові слова: багаторозрядна арифметика, послідовна модель обчислень, паралельна модель обчислень, квантова модель обчислень

Вступ. Поява різних нових обчислювальних систем пов'язана з вирішенням прикладних задач у різних галузях. Серед таких задач можна виділити задачі обчислення систем лінійних алгебраїчних рівнянь з кількістю невідомих у сотні мільйонів, моделювання фізичних, хімічних процесів, захисту інформації. Їх розв'язання розширює використання багаторозрядної арифметики із-за того, що неврахування похибок приводить до того, що іноді отримуються комп'ютерні рішення, які не відповідають фізичному змісту.

1. Моделі обчислень

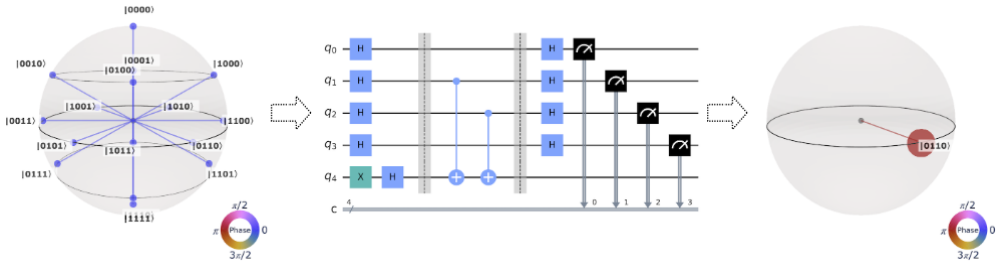
Послідовна, паралельна та квантова моделі обчислень генерують питання розробки нових та вдосконалення існуючих алгоритмів реалізації операцій багаторозрядної арифметики та пошуку критеріїв їх якості за складністю, швидкодією, енергоефективністю і т. п.

Під моделлю послідовного комп'ютера мається на увазі реалізація моделі, в якій алгоритм виконується на одному процесорі (ядрі). Оброблювані дані знаходяться в основній пам'яті комп'ютера. Для аналізу складності відбувається оцінка кількості операцій, які виконуються одна за одною послідовно.

В якості моделі паралельного комп'ютера можна розглядати модель GPU (Graphics Processing Unit) відеоплати, яка є найпростішою «у домашніх умовах». GPU побудована за технологією SIMD (Single Instruction–Multiple Data), де потокові процесори можуть виконувати одну інструкцію одночасно, оперуючи з різними даними, має власну пам'ять, яка значно швидша за оперативну пам'ять.

Під моделлю ква́нтового комп'ютера розуміють пристрій, функціонування якого ґрунтується на двох принципах квантової механіки: принципі суперпозиції та явищі квантової заплутаності. Якщо класичний комп'ютер оперує двійковими бітами зі значеннями 0 або 1, то квантовий комп'ютер використовує квантові біти (кубіти), які можуть знаходитися у суперпозиції станів. Заплутаність стосується станів більш ніж одного кубіта. Комбінований стан кубітів містить більше інформації, ніж кубіти окремо. Заплутані стани можна використовувати для квантової телепортації – передачі інформації від одного кубіта до іншого незалежно від відносної фізичної близькості кубітів.

Алгоритми та програми для квантових комп’ютерів пишуться у вигляді квантових схем, які складаються з квантових операцій над квантовими даними, що зберігаються у кубітах, та класичних обчислень. Спочатку готується суперпозиція всіх можливих станів квантової системи, які є вхідними даними для квантової схеми. Далі квантова схема змінює суперпозиції компонентів. Те, що залишається після скасування відносних амплітуд і фаз вхідного стану, є результатом квантової схеми. Обчислення за схемою відбувається зліва направо. Лівий кінець має початкові квантові дані, а правий – результат квантової схеми. Квантові схеми дозволяють квантовому комп’ютеру отримувати вихідну інформацію. Моделлю квантового комп’ютера є квантова машина Тюрінга, яка була розроблена Девідом Дойчем у 1985 році, де він припустив, що квантові вентилі можуть функціонувати так само, як і класичні бінарні логічні елементи. Квантові вентилі (або gates) є примітивними операціями над квантовими даними, які є квантово-механічним ядром квантової схеми. Деякі такі вентилі, як $\oplus$ (див. рис. 1) мають класичний аналог операції побітового інвертування *XOR*.



Суперпозиція всіх можливих станів | Обчислення за квантовою схемою зі зміною суперпозиції компонентів | Результат

Рис. 1. Процес квантових обчислень

Вентиль Адамара H разом з параметеризованими обертаннями $rX(\theta)$ та $rY(\theta)$ генерують стани суперпозиції, вентилі Z, $rZ(\theta)$, S, T передають фази, які можна використовувати для інтерференції. Вентиль CX використовується для заплутування станів пари кубітів. Операції «обчислення» (див. рис. 1), показані на схемі як вимірювачі, витягують частину інформації про стан кубіту, часто втрачаючи фазу, можуть представляти класичний біт результату. Деякі класично-квантові комп’ютери розширюють послідовність квантових схем із чергуванням з класичними обчисленнями, наприклад, варіаційні квантові алгоритми.

2. Особливості оцінки складності алгоритмів багаторозрядної арифметики для різних моделей

Перші послідовні комп'ютери опрацьовували біти. Прискорення обчислень відбувалося на рівні бітів за рахунок використання більш оптимальних схем, які, наприклад, давали змогу «передбачити» знак переносу для більшої кількості бітів для операції побітового додавання [1]. Обчислювальну складність операцій багаторозрядної арифметики розраховували, використовуючи довжину числа у бітах. Алгоритм Шенхаге–Штрассена [2] для реалізації множення двох чисел довжиною у n бітів має складність $O(n \cdot \log_2(n) \cdot \log_2(\log_2(n)))$. Поряд з цим почали надавати оцінку складності алгоритму $O(N \cdot \log_2(N))$, де N – довжина числа у машинних словах (8, 16, 32, 64, 128 і т.д.), коли 8-розрядні процесори (серії Z80 та КР580) почали виконувати однобайтові операцій. Ці процесори виконували множення на порядок повільніше ніж операції додавання, тому для реалізації багаторозрядної арифметики кількість операцій множення мала вплив на вибір таких методів, як множення Карацуби [3]. Важливою особливістю обчислення складності алгоритмів у послідовній моделі є те, що у багатослівних операціях додавання врахування знаку переносу відбувається автоматично і не впливає на загальну структуру алгоритму.

Для паралельних систем перші спроби сформулювати критерії прискорення відносять до роботи Джина Амдала [4], у якій він сформулював закон, який стверджує, що невелика частина програми, що не піддається розпаралелюванню, обмежить загальне прискорення від розпаралелювання, $S_p = 1/(\alpha + (1-\alpha)/p)$, де α – частина послідовних обчислень, p – кількість процесорів.

Закон Амдала визначає верхню межу корисності від збільшення кількості процесорів в обчислювальній системі. Обґрунтовано, що додаткові зусилля не мають ніякого ефекту, якщо задача не може бути розпаралелена через обмеження послідовної частини. Треба враховувати час, необхідний для передачі даних між вузлами обчислювальної системи. Це означає, що залежність часу обчислень від числа вузлів матиме максимум, та з певного моменту додавання нових вузлів в систему буде збільшувати час роботи програми. На рис. 2 видно, що у випадку загальної кількості у 5 відсотків послідовних кроків, програма не може бути прискорена більше ніж у 20 разів, незважаючи на кількість задіяних процесорів.

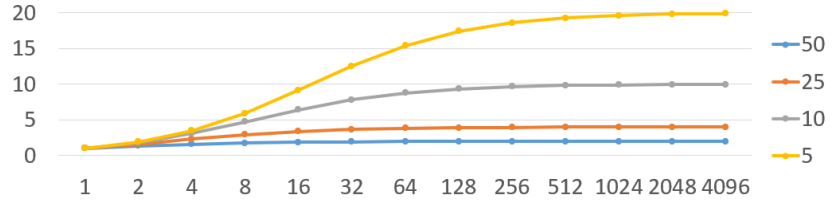


Рис. 2. Залежність S_p коефіцієнту прискорення від відсотку послідовних обчислень та кількості задіяних процесорів

Квантовий алгоритм Шора та алгоритм Саймона з'явилися до появи перших квантових комп'ютерів. У разі обчислення складності квантового алгоритму для практичної реалізації враховують три основні оцінки: кількість

задіяних кубітів, складність квантової схеми та глибина квантової схеми. Якщо одна з цих оцінок перевищує параметри наявних існуючих квантових комп‘ютерів, то такий квантовий алгоритм не може бути реалізовано практично.

Класичний метод множення «у стовпчик» потребує $4n + 1$ кубітів для множення двох чисел довжиною n бітів, але глибина квантової схеми обчислення має квадратичну залежність від довжині n, що для великих n дуже впливає на час виконання та на рівень помилок. Метод Тоом–3 має кращі значення за кількістю задіяних кубітів (для невеликої кількості кубітів) та має меншу глибину квантової схеми (див. табл. 1). Операція ділення є вузьким місцем у оптимізації методу Тоом–3. У роботі [5] запропонована оптимізація, яка дає змогу замінити чотири схеми ділення однією схемою ділення за рахунок використання множення на константу у повторюваній схемі та одну операцію обміну. Починаючи з n=52 класичний метод множення «у стовпчик» починає вигравати за кількістю кубітів та вентилів Тоффолі, хоча програє за глибиною Тоффолі.

Таблиця 1

Складність обчислення операції множення у квантовій моделі обчислення

Метод множення	Кількість вентилів Тоффолі	Кількість кубітів	Глибина Тоффолі
Класичний	$4n^2 - 3n$	$4n + 1$	$4n^2 - 4n$ +1
Карацуби	$42n^{1.585}$	$n^{1.427}$	$n^{1.158}$
Тоом–2.5	$49n^{1.547}$	$n^{1.404}$	$n^{1.143}$
Тоом–3	$8n^2 + 66n^{1.465} - 72n$	$n^{1.353}$	$n^{1.112}$

3. Обмеження реалізації алгоритмів у різних моделях

У послідовній моделі обчислень у разі реалізації алгоритмів необхідно звертати увагу на те, що операції множення повинні виконуватися почергово з операціями додавання. У цьому разі мікрокоманди операції множення можуть виконуватися паралельно з мікрокомандами додавання, що значно пришвидшує виконання. Необхідно враховувати, що технічно існують обмеження на розмір оперативної пам’яті та різні рівні кешу (1-го та 2-го рівня), доступ до яких впливає на швидкодію програми.

Для паралельної моделі обчислення діють такі самі обмеження, як і для послідовної моделі. На практиці, чим більше паралельних процесорів задіються, тим більше обмежень потрібно враховувати. Якщо розглядати GPU, то одне з перших обмежень є розмір кеш-пам‘яті робочої групи процесорів. Обчислення на GPU розбиваються на робочі групи паралельних процесорів. Кожна така група має свою кеш-пам‘ять. Кеш-пам‘ять робочої групи значно швидше локальної або глобальної пам’яті GPU. Перевагою кеш-пам‘яті є те, що якщо хоча б один процесор групи читає дані з локальної або глобальної пам’яті, то решті процесорів немає потреби виконувати операцію зчитування з тієї ж комірки пам’яті. Дані зчитуються вже з кеш-пам‘яті. Наступним обмеженням є кількість паралельних процесорів у робочі групі. Особливістю виконання операцій у робочій групі є те, що не потрібно синхронізувати усі процесори групи, вони виконують операції синхронно. Перевагою GPU є те, що GPU підтримує

векторні операції. Але векторні операції мають обмеження за довжиною 2, 4, 8 та 16.

Для квантової моделі обчислень необхідно враховувати, що кількість кубітів кожного року збільшується, що збільшує складність отримання стану «заплутаності». Неможливо створити ідеальні умови, коли зовнішнє середовище не впливає на стани кубітів, тому необхідно виконувати калібрування. Швидкість виконання схеми залежить від часу спрацьовування вентиля (gate). Наприклад, кількість обробки вентилів за секунду (CLOPS) для ibm_oslo дорівнює 2.6K, що відповідає середньому часу у 0.38 мілісекнди для одного вентиля. Враховуючи нестабільність станів кубітів, є обмеження на кількість вентилів, які може опрацювати квантовий комп'ютер. Кількість вентилів також є обмеженою для кожного кубіта, що також впливає на реалізацію алгоритму у квантовій схемі.

Висновки. У роботі надано загальний опис різних моделей обчислень, розглянуто особливості реалізації алгоритмів багаторозрядної арифметики у послідовній, паралельній та квантовій моделях обчислень та проаналізовано особливості обчислення складності за різними критеріями якості для знаходження ефективних алгоритмів. Алгоритмичне та програмне забезпечення виконання операцій багаторозрядної арифметики для послідовної та паралельної моделей впроваджено у СБ України та Люблінському політехнічному інституті (Польща).

Література

[1] Задірака В.К., Терещенко А.М. Комп'ютерна арифметика багаторозрядних чисел у послідовній та паралельній моделях обчислень. – Київ: Наук. думка, 2021. 136 с.
[2] Schonhage A., Straßen V. Schnelle Multiplikation großen Zahlen. *Computing*. 1971. 7, N 3–4. P. 281–292. DOI: 10.1007/BF02242355.
[3] Карацуба А.А., Офман Ю.П. Умножение многоразрядных чисел на автоматах. *ДАН СССР*, 145 (1962). С. 293–294.
[4] Amdahl G.M. Validity of the single processor approach to achieving large-scale computing capabilities. *AFIPS Conf. Proc*. 30. 1967. P. 483–485.
[5] Larasati H.T. Awaludin A.M., Ji J. Kim H. Quantum Circuit Design of Toom 3-Way Multiplication. *Appl. Sci*. 2021. 11, 3752. DOI: 10.3390/app11093752.

Multi-digit arithmetic in sequential, parallel and quantum computational models

Valeriy Zadiraka, Andrii Tereshchenko, Inna Shvidchenko

Various computational models for the implementation of multi–digit arithmetic operations are considered. The division into sequential, parallel and quantum computing models is to consider the features of the architecture of the devices on which the programs will be executed. The work presents the features of the implementation of algorithms for different computational models. The main criteria to get the complexity for different computational models are presented. It is said about restrictions that should be considered in the case of the implementation.

Отримано 09.03.23