e-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2019. № 3
p-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2019. № 3

УДК 519.688: 004.75:004.421

ОБЧИСЛЕННЯ ФАКТОРИЗАЦІЇ ЧИСЛА В МУЛЬТИПОТОКОВОМУ РЕЖИМІ НА КРИСТАЛІ

Процько І. О. – канд. техн. наук, доцент кафедри автоматизованих систем управління Національного університету «Львівська політехніка», Львів, Україна.

Грищук О. В. – розробник, ТзОВ «Логіка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Забезпечення високої швидкодії обчислення комп'ютерними системами класичної задачі факторизації цілочисельного значення на прості множники вимагає розробки ефективних алгоритмічних методів з використанням новітніх інформаційних технологій. Швидке обчислення факторизації чисел для забезпечення високої криптостійкості інформаційних даних, для переходу до багатовимірного подання одновимірних послідовностей інформаційних даних та інших застосувань є достатньо затребуваним в багатьох практичних завданнях.

Мета роботи – вдосконалення методу пробних ділень для обчислення факторизації цілочисельного значення з використанням розпарарелелення обчислень та ефективного використання обчислювальних ресурсів комп'ютерних систем, що забезпечить швидше обчислення значень простих множників розкладу.

Метод. Запропоновано використання залишків кожного розряду бінарного представлення числа факторизації з метою перевірки на подільність в підході виконання пробних ділень на прості числа.

Результати. Підсумком дослідження є розроблення програми паралельного виконання факторизації цілочисельного значення в комп'ютерних системах з багатоядерними процесорами.

Висновки. У проведеному дослідженні застосовано метод проведення перевірки на подільність з використанням залишків кожного розряду бінарного представлення числа факторизації, що дозволяє в мультипотоковому режимі виконувати розклад числа на множники. Основна ідея застосування належного математичного апарату полягає у використанні залишків цілого показника степеня числа два від простих чисел. В результаті цього виконується операція накопичення значень залишків, яке перевіряється на рівність з відповідним простим числом та його степенів. Можливість мультипотокової програмної організації факторизації числа забезпечує її паралельне виконання в багатоядерних процесорах комп'ютерних систем.

КЛЮЧОВІ СЛОВА: факторизація числа, прості множники, залишки вагових коефіцієнтів, пул потоків, паралельне обчислення.

АБРЕВІАТУРИ

КС – комп'ютерна система;
ФЦЗ – факторизація цілочисельного значення;
HT – hyper-threading technology;
IDE – integrated development environment;
CMT – chip multi-threading;
TD – trial division.

НОМЕНКЛАТУРА

F_p – просте число Ферма;
M_p – просте число Мерсенна;
N – цілочисельне число факторизації;
n_i – двійковий розряд числа факторизації;
p – просте число;
p_i – прості множники факторизації числа;
T – періодичність повторення залишків;
X – алгебраїчна структура;
Y – алгебраїчна структура.

ВСТУП

Реалізація фундаментальної теореми арифметики про однозначне подання цілого числа у вигляді добутку простих чисел [1] і на сьогоднішній час не залишає багатьох розробників. Факторизація, як процес розкладу натурального числа на прості множники, є достатньо затребувана в багатьох практичних застосуваннях. Наприклад, перехід від одновимірного до багатовимірного подання послідовностей інформаційних даних широко використовується в багатовимірних засобах опрацювання даних. У методі Гуда-Томаса обчислення дискретного перетворення Фур'є складеного обсягу N, що використовує китайську теорему про залишки для цілих чисел, не потрібно виконувати добутки на повертаючі множники, за умовою що N – обсяг послідовності розкладається на взаємно простоті множники для переходу від одно до багатовимірного перетворення. Метод Агарвала-Кулі переводить обчислення одновимірної N-точкової згортки в багатовимірну за умови, що обсяг розкладається на взаємно прості множники [2]. ФЦЗ використовується для знаходження одного з параметрів криптосистем, завдяки якому здійснюється захист значних об'ємів інформаційних даних [3].

Починаючи з появи нових підходів в алгоритмах шифрування на кінець 70-х років, розроблено ряд алгоритмів ФЦЗ. До них належать ρ – метод, (p−1) – метод Полларда, метод еліптичних кривих, метод квадратного решета, метод решета числового поля й інші [4]. На основі цих методів розробляються програмні модулі та модифікації класичних алгоритмів для рішення різноманітних прикладних завдань [5].

Об'єктом дослідження є процес розробки алгоритмічного та програмного забезпечення для ФЦЗ на

© Процько І. О., Грищук О. В., 2019
DOI 10.15588/1607-3274-2019-3-13

прості множники з використанням мультипотокових обчислень.

Предметом дослідження є алгоритм ФЦЗ, що виконує проведення його перевірки на подільність з використанням залишків кожного розряду бінарного представлення числа факторизації за простим числом.

Метою роботи є вдосконалення методу пробних ділень для обчислення ФЦЗ з використанням розпаралелення обчислень.

1 ПОСТАНОВКА ЗАДАЧІ

Нехай задано ціле число N, яке факторизуємо на прості множники, подається у вигляді

$$N = p_1^{s1} p_2^{s2} \dots p_n^{sn}, \quad (1)$$

де p_i прості множники; sk степінь їх повторюваності, i=1, 2,…n.

Найпростішим рішенням виконання факторизації є метод пробних ділень. Метод добре розпаралелюється, використовуючи перевірки на ділення без залишку числа N на набір дільників з множини простих чисел 2, 3, 5, 7, 9, 11,… . Однак, послідовне виконання операцій ділення для перевірки на кратність в сучасних універсальних КС виконується за допомогою мікропрограми ділення, що має найбільший ваговий коефіцієнт серед арифметичних операцій [6].

Задача підвищення швидкодії обчислення ФЦЗ N вирішується в напрямі заміни операції ділення на сукупність обчислень, що використовують арифметичні операції з меншим ваговим коефіцієнтом.

Для одержання ефективного результату ФЦЗ може бути використана можливість не тільки розпаралелення обчислень перевірки на подільність з множини простих чисел, але й на рівнях визначення залишку в розробленому програмному забезпеченні для сучасних КС.

2 ОГЛЯД ЛІТЕРАТУРИ

Сучасні інформаційні технології ставлять вимогу перед алгоритмічними засобами, що полягають у регулярності, простоті та інформаційній незалежності їхніх складових з метою ефективного використання обчислювальних ресурсів. Багато алгоритмів ФЦЗ аналізуються саме з позицій можливості багатопотокової програмної організації та, відповідно, паралельного виконання [7, 8, 9]. Виконання багатьох потоків у багатоядерному процесорі називають CMT багатопотоковістю на кристалі.

Дослідження та розробка засобів факторизації на основі ρ та (p−1) методів Полларда (Pollard) [10], методу Шермана-Лемана (Sherman-Lehman) [11], методу Шенкса (Shanks) [12], методу еліптичних кривих Ленстри (Lenstra) [13] в напрямку розпаралелювання виконання обчислень показує, що основна ідея полягає у виборі випадкового значення наближеного до елементу розкладу для окремого паралельного процесу, який буде виконувати алгоритм відповідного методу. Однак, виконання алгоритму декількома одночасними значними за складністю потоками для знаходження множників числа розкладу не завжди дає очікуваний результат.

У відомому методі Ферма основна ідея факторизації числа $N=p_1 p_2$ полягає в пошуку таких пар натуральних чисел, які задовольняють певній умові. Однак, у випадку, коли множники p_1 і p_2 близькі за значенням до 1 та N алгоритм буде виконуватись гірше, ніж метод пробних ділень [14].

Метою роботи є вдосконалення методу пробних ділень для обчислення ФЦЗ через розпаралелення обчислень та ефективне використання обчислювальних ресурсів КС, що забезпечить швидке обчислення простих множників.

3 МАТЕРІАЛИ І МЕТОДИ

Найбільш відомий алгоритм факторизації – це перевірка діленням TD, що полягає в перевірці на подільність числа N для кожного простого числа, меншого або рівного квадратному кореню з числа розкладу N. Операція ділення реалізується з послідовного набору арифметичних операцій на основі додавання/віднімання, незалежно від того, який з алгоритмів використовується.

В основу вдосконалення методу пробних ділень для обчислення ФЦЗ, що використовує заміну арифметичної операції ділення на сукупність обчислень на основі властивості допустимості й стабільності розбиття алгебраїчних структур X, Y:

$$\overline{X * Y} = \overline{X} * \overline{Y}. \quad (2)$$

Тобто, у випадку подання цілочисельного значення $N=(n_{k-1} 2^{k-1} + n_{k-2} 2^{k-1} + \dots + n_1 2^1 + n_0 2^0)$ в двійковій системі числення ця властивість матиме вигляд

$$\begin{aligned} N \bmod p &= (n_{k-1} 2^{k-1} + n_{k-1} 2^{k-2} + \dots + n_1 2^1 + n_0 2^0) \\ &\bmod p = n_{k-1} (2^{k-1} \bmod p) + n_{k-1} (2^{k-2} \bmod p) + \\ &\dots + n_1 (2^1 \bmod p) + n_0 (2^0 \bmod p). \end{aligned} \quad (3)$$

Отже, перевірка на подільність простого p зводиться до визначення за модулем p кожного вагового коефіцієнта 2^i (i=0,1,…k–1) числа N, що у бінарній формі має значення розрядів n_i (i=0,1,…k–1), що дорівнюють 0 або 1.

Значення залишків кожного вагового коефіцієнта $2^i \bmod p$ (i=0,1,…k–1) ефективно, в порівнянні з арифметичною операцією ділення, запам'ятовується або визначаються в обчислювальному середовищі. Послідовність залишків періодично повторюється зі зростанням кількості розрядів бінарного подання числа N. Наприклад, період повторення T кількості залишків для початкових простих p подано в табл. 1.

Таблиця 1 – Періодичність повторення залишків

p	3	5	7	11	13	17	19	23	29	31
T	2	4	3	10	12	8	18	11	28	5

© Процько І. О., Грищук О. В., 2019
DOI 10.15588/1607-3274-2019-3-13

Найбільше значення періоду повторення T залишків дорівнює (p–1). Найменше значення T , що періодично повторюється зі збільшенням кількості бінарних розрядів k числа N, є в простих числах Мерсенна

$$M_p = 2^j - 1, \quad (4)$$

де періодичність T, тобто кількість залишків, дорівнює показнику степені j (j = 0, 1, 2, 3,…). Наприклад, для M_p =2^5–1 =31, j=5 і, відповідно, послідовність залишків дорівнює 1, 2, 4, 8, 16. Для простого числа Ферма F_p=17 кількість залишків дорівнює j=(17–1)/2=8 і, відповідно, послідовність залишків дорівнює 1, 2, 4, 8, 16, 15, 13, 9. Для інших конкретних простих значень послідовність залишків може мати періодичність (p–1)/4, (p–1)/8,…і т. д.

Отже, в процесі реалізації в обчислювальному середовищі значення залишків кожного вагового коефіцієнта за модулем простого числа p менші p і визначаються в кількості T для відповідного значення числа розкладу N.

Виконавши накопичення значень залишків за вибраними ваговими коефіцієнтами (2^i mod p) (i=0,1,…k–1) для n_i (i=0,1,…k–1), що дорівнюють 1, порівнюємо накопичену суму з простим числом p. У випадку порівняння, коли накопичене значення залишків більше p – знову проводиться за формулою (3) накопичення залишків від попереднього одержаного накопиченого значення, а у випадку рівності числу p – виводиться елемент розкладу p та виконується перехід на повторну перевірку подільності цього простого. У випадку порівняння, коли накопичене значення залишків менше p, переходимо до наступного значення з послідовності простих чисел. Отже, в результаті отримуємо набір простих множників розкладу (1) для числа N [15].

4 ЕКСПЕРИМЕНТИ

Обчислення факторизації числа N на прості множники $p_1 < p_2 \ldots < p_i$ на основі методу пробних ділень реалізовано в IDE Visual C++ 2017 і може бути скомпільоване для іншого середовища, наприклад для тестування використовувався також компілятор GCC.

Програма виконує наступні дії:

1. Введення числа розкладу N та кількості потоків для паралельного виконання обчислення факторизації.

2. Побудова послідовності в кількості T залишків за простим числом p_i .

3. Аналіз на подільність числа N простому числу p_i.

4. Вивід в файл елемента розкладу p_i для N.

5. Перехід на наступне просте число p_{i+1} з обмеженням множини пошуку простих чисел.

В програмі написаній на мові C++ використовуються функції:

find_prime_factors(), що розподіляє роботу між заданою кількістю потоків;

create_mod_prime_table(), що створює послідовність залишків від ділення на кожне просте число;

is_next_prime(), що визначає чи є число наступним простим числом, використовуючи відомі попередні прості числа;

is_divider(), шо виконує за формулою (2) операції накопичення значень залишків та їх перевірку на рівність зі значенням з множини простих чисел та їх степенів. Накопичення значень залишків проводиться, коли двійкові розряди числа n_i =1.

На початку алгоритму побудова послідовності залишків за простими числами обмежується пошуком дільників, дорівнює кореню квадратному з N [1]. З кожним знайденим елементом розкладу, підраховується їх добуток і записується в структуру Products. Цей добуток використовується для обмеження пошуку з множини простих чисел. Тобто, якщо добуток помножений на наступне просте число p_i більший за число для якого шукаються дільники, то пошук припиняється. Цей критерій не призводить до зупинки обчислень.

Паралельне обчислення ФЦЗ досягається створенням потоків і розподіленням їх виконання між ядрами мікропроцесора в КС. В плані функціональної декомпозиції обчислення факторизації між потоками здійснюється пропорційно. Для ефективного використання обчислювальних ресурсів використано пул потоків і організацію взаємодії між ними покладено на функцію бібліотеки CTPL [16], що є надбудовою над стандартною бібліотекою STL і має можливість працювати з системними потоками. Всі потоки працюють без зупинок. Для передавання простого числа розкладу, результат записується в атомарне поле Product в структурі ComputationContext. В кінці виконання програми результати зчитуються зі структур і записуються у вихідний файл.

Метою роботи є вдосконалення методу пробних ділень для обчислення ФЦЗ через використання багатопотокових обчислень. Тому багато потокове виконання пробних ділень аналізується для операцій ділення, що реалізовано за:

− накопленим значенням залишків кожного вагового коефіцієнта бінарного числа факторизації (3);

− швидкою програмною реалізацією [17];

− апаратним діленням в блоці цілочисельної арифметики ядер мікропроцесора [18].

5 РЕЗУЛЬТАТИ

Тестування швидкодії виконання ФЦЗ в залежності від кількості потоків визначається за часом обчислення. Розроблена паралельна програма ФЧЗ тестувалась в КС з багатоядерними процесорами зі спільною пам’яттю лінійки Intel Core i3 у середовищі Windows та Intel Core i7 – Linux. Створені потоки паралельно реалізуються повноцінними обчислювальними ядрами в 32 або 64-бітному режимі із застосуванням HT, що суттєво підвищує продуктивність обчислення. Результати тестування подано у таблицях, що містять дані середнього часу (мікросекунди) виконання ФЦЗ в 32-бітному режимі, яке обчислюється 5000 разів для простих чисел N = 131071=(2^{17}–1), 131073 =(2^{17}+1) та

© Процько І. О., Грищук О. В., 2019
DOI 10.15588/1607-3274-2019-3-13

1048573 = (2^{20}–2). Тестування ФЦЗ проводилось в універсальних КС з процесором Intel Core i3–6100, який має 2 фізичних ядра з тактовою частотою 3.7ГГц і кожне фізичне ядро складається з двох віртуальних відповідно HT, та в КС з процесором Intel Core i7–7820HQ, який має 4 фізичних ядра з тактовою частотою 2.9ГГц і кожне фізичне ядро складається з двох віртуальних.

У таблиці 2 показано залежність часу від введеного значення кількості потоків, який витрачено в КС з процесором Intel Core i3–6100 на обчислення ФЦЗ N = 131071 = $(1111111111111111)_2$ та N = 1048573 = $(11111111111111111101)_2$ в середовищі Windows. У таблиці 3 показано залежність часу від введеного значення кількості потоків, який витрачено на обчислення ФЦЗ N =131071 в середовищі Linux. У таблиці 4 показано залежність часу від введеного значення кількості потоків витраченого в КС з процесором Intel Core i3–6100 на обчислення ФЦЗ 131073 в середовищі Windows.

6 ОБГОВОРЕННЯ

Порівняння результатів з таблиць 2 і 4 показує зменшення часу виконання в алгоритмі накопичення значень залишків в порівнянні з алгоритмом, що використовує програмне швидке ділення. Це пояснюється тим, що число розкладу N = 131073= $(00100000000000000001)_2$ має малу кількість встановлених одиничних бітів, і відповідно алгоритм залишків робить менше обчислень і має кращий час виконання.

Кожен потік опрацьовує сформовану таблицю залишків зі зміщенням і кроком, який кратний кількості потоків. Чим більша кількість потоків, тим ширший пошук в таблиці залишків. Чим ширший пошук, тим швидше знаходиться елемент розкладу. Тобто, це пов'язано з особливістю алгоритму. Тому зростання кількості потоків, що створюються програмно, зменшують час обчислення і оптимальним для КС для даних з таблиць 3 і 4 є кількість потоків, що дорівнює 7.

Подвоєння кількості ядер в процесорі Intel Core i7–7820HQ з підтримкою HT у кількості потоків не показало значно меншого часу в порівнянні з процесором Intel Core i3–6100. Можливо це пов'язано з меншою тактовою частотою ядер або поганою оптимізацією на рівні компілятора GCC.

Отже, ФЦЗ з використання багатопотокових обчислень на основі пробних ділень з використанням різної реалізації ділення для невеликих чисел менших 2^{64} має найкращі показники при виконанні апаратного ділення. Алгоритм залишків кожного вагового коефіцієнта бінарного представлення числа розкладу може мати співвимірні показники для конкретних значень, що мають невелику кількість одиничних бітів у бінарному поданні числа розкладу.

ВИСНОВКИ

У роботі розглянуто ФЦЗ з використанням багатопотокового виконання пробних ділень на прості числа для операції ділення, що реалізована за алгоритмами накопичення значень залишків, швидкого програмного виконання та апаратним діленням. Програмне рішення реалізовано в IDE Visual C++ 2017. Завдяки високій степені розпаралелення ФЦЗ в плані функціональної декомпозиції та відповідної незалежності потоків даних програмне рішення виконує розподілення обчислень між паралельними потоками пропорційно забезпечуючи високу ефективність використання обчислювальних ресурсів.

Наукова новизна полягає у вдосконалені методу пробних ділень ФЦЗ з використанням залишків кожного вагового коефіцієнта бінарного представлення числа, що дає змогу значно ефективніше виконувати операцію накопичення значень залишків в порівнянні з виконанням операції ділення на апаратному або програмному рівнях.

Таблиця 2 – Залежності часу (мксек.) обчислення ФЦЗ 131071 та 1048573 від кількості потоків на Intel i3–6100

К-ть потоків	1	2	3	4	5	6	7	8
Алгоритм залишків	534	268	213	191	149	204	105	**102**
Програмне ділення	326	166	135	139	111	141	80	**73**
Апаратне ділення	53	31	26	26	21	28	17	29
Алгоритм залишків	4099	2299	1805	1581	1719	1799	1774	**956**
Програмне ділення	2703	1649	1269	**1150**	1202	1222	1221	**624**
Апаратне ділення	500	513	476	493	610	616	626	**340**

Таблиця 3 – Залежності часу (мксек.) обчислення ФЦЗ 131071 від кількості потоків на Intel i7–7820HQ

К-ть потоків	1	2	3	4	5	6	7	8
Алгоритм залишків	709	382	267	208	167	184	**124**	166
Програмне ділення	273	147	101	80	64	72	**47**	66
Апаратне ділення	49	31	22	18	14	16	**11**	16

Таблиця 4 – Залежності часу (мксек.) обчислення ФЦЗ 131073 від кількості потоків на Intel i3–6100

К-ть потоків	1	2	3	4	5	6	7	8
Алгоритм залишків	59	31	20	26	16	26	**14**	30
Програмне ділення	123	67	42	52	41	40	**29**	54

© Процько І. О., Грищук О. В., 2019
DOI 10.15588/1607-3274-2019-3-13

Практичне значення застосування програмного коду для ФЦЗ в КС з багатоядерними процесорами лінійки Intel Core i3, i7 показало залежність часу обчислення від кількості сформованих потоків, що відповідно пов'язані з кількістю фізичних та віртуальних НТ ядер процесора. Отримані результати важливі для швидкого обчислення ФЦЗ під час забезпечення високого рівня криптостійкості інформаційних даних, для переходу до багатовимірного подання одновимірних послідовностей інформаційних даних та інших застосувань.

Напрям подальших досліджень полягатиме в розробці програмного забезпечення з розширенням розрядної сітки бінарного представлення числа розкладу для ФЦЗ з використанням залишків кожного вагового коефіцієнта бінарного представлення числа для високопродуктивного апаратного інструментарію.

ПОДЯКИ

Робота виконана в рамках держбюджетної науково-дослідної роботи ДБ/Нейрозахист (номер держ. реєстрації № ДР 0119U002256) (2019–20 р.) національного університету «Львівська політехніка» .

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Виноградов И. М. Основы теории чисел / И. М. Вино- градов. – М.: Наука, 1981. – 167 с.
2. McClellan J. H. Number Theory in Digital Signal Processing / J. H. McClellan, C. M. Rader. – Englewood Clis, NJ: Prentice-Hall, 1979. – 276 p.
3. Brent R. P. Some parallel algorithms for integer factorisation / R. P. Brent // European Conference on Parallel Processing, Toulouse, 1999: Part of the Lecture Notes in Computer Science book series, Berlin : Springing-Verlag, 1999. – Vol. 1685. – P. 1–22.
4. Теория чисел в криптографии : учеб. пособие / [В. А. Орлов, Н. В. Медведев, Н. А. Шимко, А. Б. Домрачева]. – М.: Изд-во МГТУ, 2011. – 223 с.
5. Кнут Д. Искусство программирования. / Д. Кнут. – Т. 2: Получисленные алгоритмы. 3-е издание. – М. : Изд. Дом «Вильямс», 2000. – 390 с.
6. Hindriksen V. How expensive is an operation on a CPU [Electronic resource] / V. Hindriksen. – Access mode: https://streamcomputing.eu/blog/2012-07-16/how-expen- sive-is-an-operation-on-a-cpu
7. A multithreaded bound varying chaotic firefly algorithm for prime factorization / [M. Mishra, U. Chaturvedi, K. Saibal, S. K. Pal] // Advance Computing Conference: 4th IEEE International conference, Gurgaon, India. February 2014: proceedings. – Published by IEEE Computer Society, 2014. – P. 1322–1325. DOI: 10.1109/IAdCC. 2014.6779518
8. Макаренко А. В. Параллельная реализация и сравнительный анализ алгоритмов факторизации с распределенной памятью [Електронний ресурс] / А. В. Макаренко, А. В. Пыхтеев, С. С. Ефимов. – Режим доступу: http://cyberleninka.ru/article/n/parallelnaya-realizatsiya-i-sravnitelnyy-analiz-algoritmov-faktori-zatsii-v-sistemah-s-raspredelyonnoy-pamyatyu
9. Huang L. New prime factorization algorithm and its parallel computing strategy / L. Huang // Information Technologies in Education and Learning: International Conference (ICITEL 2015), Atlantis, 2015: proceedings. – Published by Atlantis Press, 2015. – P. 1–4.
10. Performance Analysis of parallel Pollard's Rho algorithm / [A. K. Koundinya, G. Harish, N. K. Srinath et al.] // International Journal of Computer Science & Information Technology (IJCSIT). – 2011. – Vol. 5, No. 2. – P.157–163. DOI: 10.5121/ijcsit.2013.5214
11. Design and analysis of efficient parallel hardware prime generators / [D. K. Kim, P. Choi, M.-K. Lee et al.] // Journal of semiconductor technology and science. – 2016. – Vol. 16, No. 5. – P. 564–581. DOI: 10.5573/ JSTS. 2016.16.5.564
12. Gower J. E. Square form factorization / J. E. Gower, S. S. Wagstaff // Mathematics of Computation. – 2008. – Vol. 77, No. 261. – P. 551–588.
13. Integer factorization based on elliptic curve method: towards better exploitation of reconfigurable hardware / G. de Meulenaer, F. Gosset, G. M. de Dormale, J. J. Quisquater // Field-Programmable Custom Computing Machines: IEEE Symposium FCCM, Napa, California, 23–25 April 2007: proceedings. – Published by IEEE Computer Society, 2007. – P. 197–206.
14. Ишмухаметов Ш. Т. Методы факторизации натуральных чисел: учебное пособие / Ш. Т. Ишмухаметов. – Казань : Казанький университет, 2011. – 190 с.
15. Пат. 116912 Україна, G06F7/04(2006.01), G06F17/10(2006.01). Пристрій канонічного розкладу числа на множники / І. О. Процько, В. М. Теслюк (Україна); заявник і патентовласник Національний університет «Львівська політехніка». – № a201604083; заявл. 14.04.2016; опубл. 25.05.2018, Бюл. №10. – 4 с.
16. Бібліотека CTPL [Електронний ресурс]. – Режим доступу: https://github.com/vit-vit/CTPL.
17. Division algorithm [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Division_algorithm #Restoring_division
18. Architectures-optimization [Electronic resource] Appendix C. – Access mode: https://www.intel.com/ content/dam/doc/manual/64-ia-32-architectures-optimi- zation-manual.pdf

Стаття надійшла до редакції 27.03.2019.
Після доробки 27.06.2019.

УДК 519.688: 004.75:004.421

ВЫЧИСЛЕНИЕ ФАКТОРИЗАЦИИ ЧИСЛА В МУЛЬТИПОТОЧНОМ РЕЖИМЕ НА КРИСТАЛЛЕ

Процько И. Е. – канд. техн. наук, доцент кафедры автоматизированных систем управления Национального университета «Львивська политэхника», Львов, Украина.

Грищук О. В. – разработчик, ТзОВ "Логика", Львов, Украина.

АННОТАЦИЯ

Актуальность. Обеспечение высокого быстродействия вычисления компьютерными системами классической задачи факторизации целочисельного значения на простые множители требует разработки эффективных алгоритмических методов с использованием современных информационных технологий. Быстрое вычисление факторизации чисел для обеспечения высокой криптостойкости информационных данных, для перехода к многомерному представлению одномерных последовательностей информационных данных та других применений есть достаточно востребованным во многих практических задачах.

Цель работы. Усовершенствование метода пробных делений для вычисления факторизации целочисельного значения с использованием распараллеливания вычислений и эффективного использования вычислительных ресурсов компьютерных систем, что обеспечит быстрое вычисление значений простых множителей разложения.

Метод. Предложено использование остатков каждого разряда бинарного представления числа факторизации с целью проверки на делимость в методе использования пробных делений на простые числа.

© Процько І. О., Грищук О. В., 2019
DOI 10.15588/1607-3274-2019-3-13

Результаты. Итогом исследования есть разработка программы параллельного выполнения факторизации целочисельного значения в компьютерных системах с многоядерными процессорами.

Выводы. В проведенном исследовании применен метод проведения проверки на делимость с использованием остатков каждого разряда бинарного представления числа факторизации, что позволяет в мультипоточном режиме выполнять разложение числа на множители. Основная идея применения соответствующего математического аппарата состоит в использовании остатков целого показателя степени числа два от простых чисел. В результате этого исполняется операция накопления значений остатков, которое проверяется на равенство с соответствующим простым числом та его степеней. Возможность мультипоточной программной организации вычисления факторизации числа обеспечивает ее параллельное выполнение в многоядерных процессорах компьютерных систем**.**

КЛЮЧЕВЫЕ СЛОВА: факторизация числа, простые множители, остатки весовых коэффициентов, пул потоков, параллельные вычисления.

UDC 519.688: 004.75:004.421

COMPUTATION FACTORIZATION OF NUMBER AT CHIP MULTITHREADING MODE

Prots'ko I. O. – PhD, Associate Professor of Information Systems and Technologies Department, Lviv National Polytechnic University, Lviv, Ukraine.

Gryschuk O. V. – Software Developer, LtdC "Lohika", Lviv, Ukraine.

ABSTRACT

Context. Ensuring high-speed calculation by computer systems of the classical task of factorization of integer value on simple factors requires the development of effective algorithmic methods using the latest information technologies. Fast computation of factorization of numbers to provide high cryptocapability of information data, using multidimensional representation of one-dimensional sequences of information data and other applications is sufficiently in demand in many practical tasks.

Objective. The purpose of the work is to improve the method of trial divisions to compute the factorization of integer value with using parallelization of computations and efficient use of computing resources of computer systems, which ensures faster computation of the values of prime factors of the decomposition.

Method. It is proposed to use the residuals of each digit of the binary representation of the factorization number in order to check for divisibility in the method performing of trial divisions into prime numbers.

Results. The result of the study is to develop of a program of parallel execution of the factorization of integer value in computer systems with multi-core processors.

Conclusions. In the research, a method of checking for divisibility using the residuals of each digit of the binary representation of the factorization number was applied, which allows for multi-threaded mode to execute the decomposition of the number into the factors. The basic idea of applying the corresponding mathematical apparatus is to use the residuals of the integer exponent of the number two from prime numbers. As a result, the accumulation of residuals is performed, which is checked for equality with the corresponding prime number and its degrees. The possibility of a multithreaded software organization for computing the number factorization ensures its parallel execution in multi-core processors of computer systems.

KEYWORDS: factorization of numbers, prime factors, residuals of weight coefficients, threads pool, parallel computation.

REFERENCES

1. Vinogradov I. M. Osnovy teorii chisel. Moscow, Nauka, 1981, 167 p.
2. McClellan J. H., Rader C. M. Number Theory in Digital Signal Processing. Englewood Clis, NJ, Prentice-Hall, 1979, 276 p.
3. Brent R. P. Some parallel algorithms for integer factorisation, *European Conference on Parallel Processing, Toulouse, 1999: Part of the Lecture Notes in Computer Science book series.* Berlin, Springing-Verlag, 1999, Vol. 1685, pp. 1–22.
4. Orlov V. A., Medvedev N. V., Shimko N. A., Domracheva A. B. Teoriya chisel v kriptografii: ucheb. Posobiye. Moscow, Izd-vo MGTU, 2011, 223 p.
5. Knuth D. E. The art of computer programming. 3- ed., Vol. 1, Seminumerical Algorithms. Menlo Park, California, Addison-Wesley, 1998, 762 p.
6. Hindriksen V. How expensive is an operation on a CPU [Electronic resource]. Access mode: https://streamcomputing.eu/blog/2012-07-16/how-expen- sive-is-an-operation-on-a-cpu
7. Mishra M., Chaturvedi U., Saibal K., Pal S. K. A multithreaded bound varying chaotic firefly algorithm for prime factorization, *Advance Computing Conference: 4th IEEE International conference.* Gurgaon, India. February 2014, proceedings, Published by IEEE Computer Society, 2014, pp. 1322–1325. DOI: 10.1109/IAdCC.2014.6779518
8. Makarenko A. V., Pykhteyev A. V., Yefimov S. S. Parallel'naya realizatsiya i sravnitel'nyy analiz algoritmov faktorizatsii s raspredelennoy pamyat'yu [Electronic resource]. Access mode: http://cyberleninka.ru/article/n/ parallelnaya-realizatsiya-i-sravnitelnyy-analiz-algoritmov-faktorizatsii-v-sistemah-s-raspredelyonnoy-pamyatyu
9. Huang L. New prime factorization algorithm and its parallel computing strategy, *Information Technologies in Education and Learning: International Conference (ICITEL 2015).* Atlantis, 2015, proceedings, Published by Atlantis Press, 2015, pp. 1–4.
10. Koundinya A. K., Harish G., Srinath N. K. et al. Performance analysis of parallel Pollard's Rho algorithm, *International Journal of Computer Science & Information Technology (IJCSIT),* 2011, Vol. 5, No. 2, pp. 157–163. DOI: 10.5121/ijcsit.2013.5214
11. Kim D. K., Choi P., Lee M.-K. et al. Design and analysis of efficient parallel hardware prime generators, *Journal of semiconductor technology and science*, 2016, Vol. 16, No. 5, pp. 564–581. DOI: 10.5573/ JSTS. 2016.16.5.564
12. Gower J. E., Wagstaff S. S. Square form factorization, *Mathematics of Computation,* 2008, Vol. 77, No. 261, pp. 551–588.
13. de Meulenaer G., Gosset F., de Dormale G. M., Quisquater J. J. Integer factorization based on elliptic curve method: towards better exploitation of reconfigurable hardware, *Field-Programmable Custom Computing Machines: IEEE Symposium FCCM.* Napa, California, 23–25 April 2007, proceedings, Published by IEEE Computer Society, 2007, pp. 197–206.
14. Ishmukhametov SH. T. Metody faktorizatsii natural'nykh chisel: uchebnoye posobiye. Kazan', Kazan'skiy universitet, 2011, 190 p.
15. Protsko I., Teslyuk V. (Ukraine)Pat. 116912 Ukraine, G06F7/04(2006.01), G06F17/10(2006.01). Device of canonical factorization a number on factors; applicant Lviv National Polytechnic University, № a201604083; update. 14.04.2016; pubdate. 25.05.2018, bul. №10, 4 p.
16. Library CTPL [Electronic resource]. Access mode: https://github.com/vit-vit/CTPL.
17. Division algorithm [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Division_algorithm#Restoring_division
18. Architectures-optimization [Electronic resource] Appendix C. Access mode: https://www.intel.com/ content/dam/doc/manual/64-ia-32-architectures-optimi- zation-manual.pdf

© Процько І. О., Грищук О. В., 2019
DOI 10.15588/1607-3274-2019-3-13