УДК 004.7 **doi: 10.26906/SUNZ.2019.4.100**

В. С. Харченко, В. Я. Пєвнєв

Національний аерокосмічний університет імені М. Є. Жуковського «ХАІ», Харків, Україна

ТЕХНОЛОГІЯ ПОБУДОВИ ПАРАЛЕЛЬНИХ АЛГОРИТМІВ ФАКТОРИЗАЦІЇ

Розглядається технологія розпаралелювання процесу факторизації великих чисел, заснована на зміні відстані між співмножники на числової осі. Представлений і проаналізовано алгоритм факторизації, який аналізує не співмножники, а складові, що дозволяє швидко вирішувати завдання факторизації як при близьких співмножником, так і при значно відрізняються один від одного. Алгоритм факторизації, заснований на рішенні нерівності, дозволяє відсікати велику кількість варіантів співмножників, які не є рішенням поставленої задачі. Проведені експериментальні дослідження показали хороші результати вирішення завдання факторизації з використанням можливого розпаралелювання цього процесу.

Ключові слова: алгоритми факторизації, технологія розпаралелювання, найбільший спільний дільник, алгоритм квадратичного решета

Вступ

Постановка проблеми. Задача факторизації є однією з самих відомих в теорії чисел. З цим завданням стикаються школярі, коли їм необхідно розкласти число на множники. Незважаючи на гадану простоту цієї задачі, фахівці стикаються з труднощами, які обумовлені трудомісткістю факторизації великих чисел [1–7].

Велика увага до даної задачі виникла завдяки появі асиметричного шифрування, заснованому саме на складності факторизації великих чисел. Під великими числами маються на увазі числа розміром у двісті і більше десяткових знаків.

Аналіз останніх досліджень і публікацій. На сьогодні для практичного застосування існує два великі класи алгоритмів факторизації. Це експоненціальні алгоритми, складність яких залежить від довжини числа (метод Ферма, ρ і (p - 1) алгоритм Полларда) і субекспоненціальні (алгоритм Діксону, методи Ленстра, квадратичного решета і решета числового поля). Ці та деякі інші алгоритми досить детально описані в [1, 2]. Багато дослідників з надією дивляться на алгоритм Шора [3], але в осяжному майбутньому побудови квантового комп'ютера є лише надією

Більшість дослідників говорять про те, що, незважаючи на велику кількість існуючих методів, задача факторизації залишається актуальною темою. Існує декілька шляхів вирішення цієї проблеми - розробка принципово нових підходів до рішення задачі факторизації, розпаралелювання відомих алгоритмів, використання швидких обчислень та ін.

Метою статті є розробка технології розпаралелювання процесу факторизації та її експериментальне дослідження.

Основна частина

Розглядаючи підходи до процесу розпаралелювання, можна дійти наступних висновків.

Існує два підходи. Перший підхід - розпаралелюється сам обчислювальний процес усередині алгоритму, тобто одночасно виконуються ті позиції алгоритму, які незалежні. Другий підхід - одночасно виконується алгоритм на різних вхідних даних. Найчастіше ці підходи виконуються одночасно, що дозволяє значно зменшити час розв'язання задачі.

Розглянемо один з алгоритмів факторизації, представлений в [8]. Цей алгоритм ґрунтується на наступній теоремі: Якщо хоча б в одному з варіантів представлення псевдопростого числа у вигляді двох доданків, ці доданки виявляться не взаємно простими числами, то дане число є складеним. Алгоритм виглядає таким чином:

1. З заданого числа обчислюється квадратний корінь і отриманий результат округляється в меншу сторону
2. Обчислюється різниця між заданим числом і числом, отриманим в п.1
3. Отримані числа розкладаються на співмножники.
4. Якщо в співмножником є хоч би одно однакове число, то перейти до п. 8
5. З меншого числа віднімається одиниця
6. Якщо отриманий результат > 1, то перехід до п. 2.
7. Число просте.
8. Число складене.

За своєю ідеєю представлений метод схожий з квадратичним решетом. Відмінність складає те, що число, отримане в результаті обчислення квадратного кореня, в запропонованому методі зменшується. Це обумовлено тим, що більше кореня квадратного може бути максимум один співмножник, а решта будуть менше отриманого кореня.

Як приклад розглянемо число 996533 [8]. Квадратний корінь з цього числа, округлений в меншу сторону буде дорівнює 998. У табл. 1 в колонках один і два знаходяться рівні і менше числа 998 і їх розкладання, згідно з основною теоремою арифметики. У третій колонці - числа, отримані як різниця між числом, що факторизується, і числами в першій колонці. У четвертій колонці розкладання чисел з третьої колонки. У п'ятій колонці - найбільший спільний дільник.

На підставі отриманого результату одне з чисел, яке є з'єднанням числа 996533, становить 89, а друге - 11 + 2 * 7 * 17 * 47 = 11197.

Наскільки ефективний такий спосіб факторизації числа?

Таблиця 1 – **Розкладання числа 996533**

I	II	III	IV	V
998	2*499	995535	3*3*5*22123	1
997	PN	995536	2^4*43*1447	1
996	2^2*3*83	995537	17*157*373	1
995	5*199	995538	2*3*277*599	1
994	2*7*71	995539	PN	1
993	3*331	995540	2^2*5*7*13*547	1
992	2^5*31	995541	3*29*11443	1
991	PN	995542	2*497771	1
990	2*3^2*5*11	995543	19*151*347	1
989	23*43	995544	2^3*3^3*11*419	1
988	2^2*13*19	995545	5*199109	1
987	3*7*47	995546	2*497773	1
986	2*17*29	995547	3*7*47407	1
985	5*197	995548	2*2*248887	1
984	2^3*3*41	995549	PN	1
983	PN	995550	2*3*5^2*6637	1
982	2*491	995551	PN	1
981	3^2*109	995552	2^5*53*587	1
980	2^2*5*7^2	995553	3^2*13*67*127	1
979	11*89	995554	2*7*17*47*89	89

Очевидно, що досить великий час витрачатиметься на розкладання великого числа на співмножники. Виходячи з поставленою задачі по знаходженню спільної співмножника, має сенс використовувати усім відомий алгоритм Евкліда для знаходження найбільшого спільного дільника.

Перш ніж перейти до використання алгоритму Евкліда слід зазначити ще одну закономірність. Більшість розглянутих пар мають в своєму складі перші десять простих чисел від 2 до 29. Якщо перемножити ці числа, то в результаті вийде число 6469693230.

Визначимо НСД чисел 6469693230 та 996533. НСД (6469693230, 996533) = 1. Це означає, що наше досліджуване число не ділиться без залишку ні на одне просте число, яке менше або дорівнює 29. Це означає, що як тільки залишок від ділення в будь-якої ітерації стане рівним або менше числа 29, то можна однозначно стверджувати, що НСД цих чисел буде рівним 1. Використовуючи добуток десяти наступних простих чисел від 31 до 71, можна обчислити НСД цієї пари чисел. При цьому обчислення можна закінчувати, коли залишок від ділення в будь-якої ітерації стане рівним або менше числа 71.

Аналогічно можна визначити максимальне число, на якому можна закінчувати процес факторизації. Цим числом буде максимальне просте число, яке використовувалося при для побудови тестової послідовність для обчислення НСД.

Якщо розглядати табл. 1, то НСД необхідно шукати між I і II числами, представленими в першій і третій колонками відповідно. Результат представлений в п'ятій колонці.

Можливість застосування методів відшукання НСД досить продуктивна. Хоча це є не що інше, як метод пробного ділення, але його швидкодія набагато вище. Це можна досягти за рахунок попереднього обчислення добутку простих чисел, причому їх кількість в ньому може бути досить великим. Наприклад, якщо перемножити усі прості числа, які менше 1000, а таких чисел 169, і НСД = 1, то ймовірність того, що число, яке перевіряється, складене, буде дорівнює 2^{-169} [9]. Слід зазначити і той факт, що результати попереднього обчислення можна зберігати в базі даних, яка буде постійно поповнюватися.

Якщо проаналізувати табл.1, то можна виявити ще одну закономірність, яка не відразу впадає в око. При зменшенні числа з першої колонки до 890 знову НСД стане рівним 89. І такий результат буде з'являтися з кроком 89.

Розглянемо дані, представлені в табл. 2 [10]. У першому і другому стовпчику таблиці знаходяться співмножники числа, яке потрібне факторизувати. У третьому стовпці – квадратний корінь з добутку представлених чисел, округлений до цілого в меншу сторону. У четвертому стовпці величина числа, відповідна колонки 1 табл.2, на рядку якого був знайдений НСД $\neq$ 1. У п'ятому стовпці представлено кількість кроків до знаходження НСД. Незважаючи на малий розмір чисел, які знаходяться в табл. 2, їх аналіз дозволяє зробити дуже цікаві висновки. Як в більшості алгоритмів знайти «близькі» співмножники не складає особливих труднощів. Якщо різниця в величині співмножників більш ніж в чотири рази, то розмір кроку може коливатися від одиниці до величини меншого співмножника.

Таблиця 2 – **Результати факторизації**

I	II	III	IV	V
263	283	272	263	10
263	547	379	263	117
101	601	246	202	45
53	307	127	106	22
53	331	132	106	27
53	367	139	106	34
53	397	145	106	40
53	487	160	159	2

Це видно з табл. 2. Максимальна кількість кроків в представленому методі до отримання результату дорівнюватиме $\lceil 0{,}5\ N^{0{,}5} \rceil$, а мінімальне - один. Це буде значно менше, ніж в будь-якому з широко відомих алгоритмів факторизації.

Виходячи з отриманих результатів можна запропонувати стандартне розпаралелювання процесу факторизації, за допомогою ділення частини числової осі, на якій знаходяться числа, що перевіряються нами. Розмір одержуваних відрізків буде залежати від кількості пристроїв, на яких буде відбуватися процес факторизації. При необмежені можливості під кожне число можна виділити свій процесор, і для вирішення задачі потрібно тільки один крок. Якщо розглянути сумарну кількість кроків при використанні такого методу, то воно буде дорівнює кількості усіх перевірених чисел, тобто використовується метод простого перебору.

Для організації обчислювального процесу необхідно виробити критерій ефективності, на підставі якого можна буде планувати цей процес. Залежно від вирішуваної задачі можна запропонувати кілька варіантів критеріїв. Це може бути мінімізація часу

вирішення задачі при обмеженнях на кількість процесорів або мінімізація процесорів при обмеженні на час вирішення завдання.

У роботі розглядається розпаралелювання задачі факторизації на трьох процесорах. У табл. 3 [11] представлені результати визначення НСД двох чисел, одне з яких дорівнює 29, а друге знаходиться в першому стовпці. У другому і третьому стовпчиках знаходяться числа, що відповідають першому і третьому стовпцях табл. 3, яким відповідає результату факторизації. У четвертому, шостому та восьмому стовпцях поміщені величини НСД, при яких було знайдено рішення. У п'ятому, сьомому і дев'ятому стовпцях знаходиться числа, які визначають кількість кроків, потрібних для знаходження рішення.

Таблиця 3 – **Результати розпаралелювання алгоритму факторизації**

*1					*2		*3	
I	II	III	IV	V	VI	VII	VIII	IX
53	39	1498	29	11	53	3	29	10
61	42	1727	29	14	58	2	61	12
71	45	2014	29	17	58	7	71	8
79	47	2244	29	19	58	10	79	4
89	50	2531	29	22	58	14	87	1
97	53	2760	29	25	58	18	87	5
107	55	3048	29	27	58	21	87	10
113	57	3220	29	29	58	23	87	13
127	60	3623	58	3	58	28	87	19
149	65	4256	58	8	87	6	87	27
157	67	4486	58	10	87	9	29	1
179	72	5119	58	15	87	15	29	9
199	75	5696	58	18	87	21	29	16
211	78	6041	58	21	87	24	29	20
229	81	6560	58	24	87	29	29	26
241	83	6906	58	26	116	3	29	29
257	86	7367	58	29	116	7	29	5
263	87	7540	87	1	116	8	29	7
277	89	7944	87	3	116	11	29	11
293	92	8405	87	6	116	15	29	15

Окремо слід пояснити зміст першого рядка. У нім знаходяться коефіцієнти множення даних чисел. Це означає, що перші п'ять осередків третього рядка працюють з числом 29 * 53 = 1537. Дані в шостий, сьомий осередках цього рядка відповідатимуть числу 1537*2 =3074, а осередки восьма і дев'ята - числу 1537*3 = 4611.Як видно з табл. 3, у більшості випадків НСД дорівнює 29, або добутку 29 на якесь число (58, 87, 116). Окремо виділяються числа 53, 61, 71, 79, які є другим співмножником числа, що факторизовано. Кількість кроків знаходження рішення в усіх випадках коливається від одного до двадцяти дев'яти. Це досить наочно видно на рис. 1, де представлена залежність розміру кроку визначення НСД від розміру другого співмножника. Слід звернути увагу на те, що в таблиці відсутні деякі з простих чисел на представленому інтервалі. Це пояснюється тим, що їх відсутність не впливає на ілюстрацію роботи запропонованого алгоритму, але значно зменшує розмір таблиці.

На рис. 1 в графічному виді представлені порівняльні розміри інтервалів визначення величини кроку. Слід зазначити деякі особливості побудови цього графіка. Початок координат по осі X доводиться на координати 30, 115 і 260 для coef.1, 2 і 3 відповідно. Це дозволило будувати графіки з однієї точки. На рисунку видно, що отримані значення добре апроксимуються поліномами другого порядку. Дещо гірше апроксимація відбувається за допомогою прямих. В цьому випадку можна говорити про різних кутах нахилу цих прямих. Відповідно виходять інтервали різної довжини, яка росте при зменшенні кута нахилу розглянутих прямих. Виходячи з отриманих візуальних результатів, можна зробити висновок про можливість розбиття зон знаходження НСД одночасно на різних інтервалах числової осі. Іншими словами, можливості розпаралелювання запропонованого алгоритму.

Проаналізуємо V, VII і IX стовпці табл. 3. Як було сказано вище, в V стовпці знаходяться числа, що відповідають НСД кореня квадратного від добутку простих чисел від 53 на просте число 29 і різниці від цього добутку і квадратного кореня.

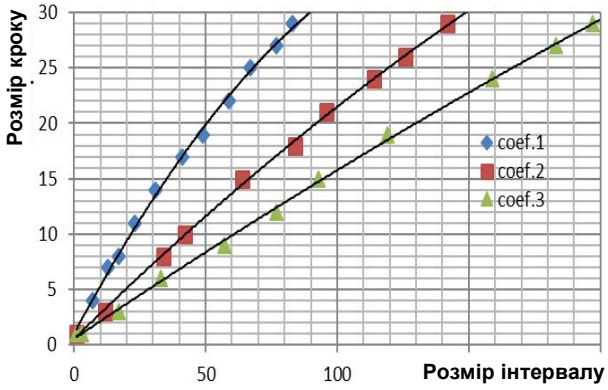

Рис. 1. Порівняльні розміри інтервалів визначення величини кроку

У VII і IX стовпцях отриманий добуток множиться на 2 і 3 відповідно. На рис. 2 під легендою Ряд 1 поміщена гістограма, що відповідає першим 20 значенням величин кроку визначення результату факторизації (стовпець V), Ряд 3 - VII стовпець, Ряд 5 - IX стовпець.

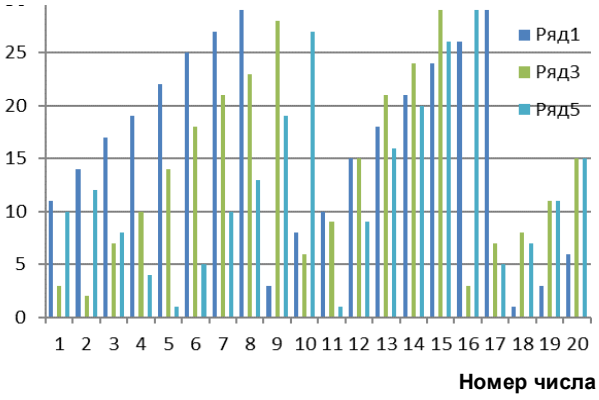

Рис. 2. Залежності величин кроку

На гістограмі добре видно, що в одинадцяти результатах кількість кроків не більше 5, в шістна-

дцяти - не більше 10, і лише в одному випадку кількість кроків більша двадцяти.

Якщо розглянути результати по кожному стовпцю, то п'ятому стовпцю відповідає шість значень не великих 10, сьомому - 9, а дев'ятому - 10. При цьому підрахунку вибирався менший результат з трьох, що відповідає цьому числу. Для наочності виграшу у кількості кроків при розпаралелюванні процесу факторизації чисел побудуємо графік залежності кількості мінімальних кроків від номера числа в табл. 3 (рис. 3). Кількість мінімальних кроків вибиралася як мінімальне серед відповідних чисел в табл. 3 (стовпці V, VII і IX). Отримані дані знаходяться під графіком в другому рядку. У першому рядку знаходиться кількість кроків, відповідне V стовпцю табл. 3. На рисунку вони представлені у вигляді гістограми.

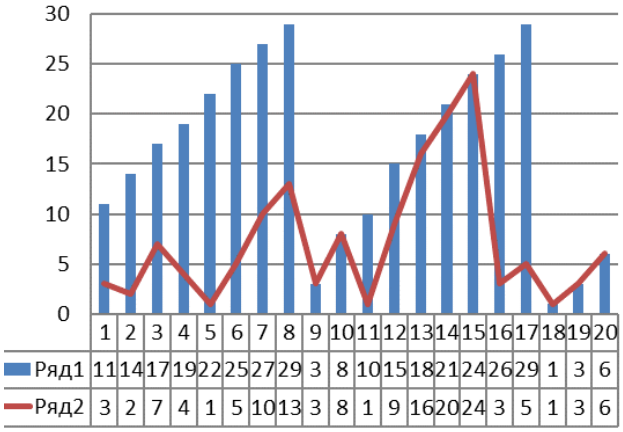

Рис. 3. Мінімальна кількість кроків

Розглянемо алгоритм факторизації, представлений в [11]. Ідея цього алгоритму полягає в складанні нерівності, де аргументом виступає кількість кроків, на яких немає рішення поставленої задачі:

$$j^2 + j(Q - P) + (N - QP) \geq 0,$$

де j – розмір кроку, Q, P – поточні співмножники, N – число, що факторизується.

Порівняємо між собою час роботи алгоритму за рішенням задачі факторизації декількох чисел з різним співвідношенням співмножників (табл. 4).

Таблиця 4 – **Порівняльний аналіз часу роботи алгоритму на трьох процесорах**

I	II			III	IV
	1	2	3		
15943	4	7	20	4	12
21293	16	2	17	2	6
28141	44	4	4	4	12
33109	56	6	3	3	9
38293	16	3	23	3	9
43657	35	2	18	2	6

У колонці I табл. 4 представлені числа, що факторизовані, в трьох колонках під номером II представлена кількість кроків алгоритму, необхідних для вирішення задачі. У колонці III представлена мінімальна кількість кроків, необхідна для вирішення задачі при використанні трьох процесорів. У колонці IV представлена сумарна кількість кроків, необхідна для вирішення задачі при використанні трьох процесорів. Розглянемо деякі особливості представлених результатів в табл. 4. У колонці II знаходиться три стовпчики. Це номери процесорів, які працюють з числами, рівними числу, вказаному в колонці I, помноженому на номер відповідного процесора. Таким чином перший процесор виконував факторизацію числа 15943, другий - числа 15943*2, третій - 15943*3.

Що лежить в основі запропонованого методу розпаралелювання? Більшість методів факторизації в своїй основі мають метод Ферма, який якимось чином модифікується [1,n2]. У пропонованому варіанті розпаралелювання проводиться спроба змінити відстань між числами на числової осі. Якщо число 15943 = 107 * 149, то на другому процесорі факторизується число 31886 = 214 * 149. В даному випадку відстань між числа збільшується, і це відображено в табл.4 у вигляді збільшення кількості кроків на другому і третьому процесорах.

Висновки

У роботі розглядаються підходи до розпаралелювання задачі факторизації, які застосовуються в сучасних обчислювальних системах. Пропонуються критерії ефективності, на підставі яких можна висувати вимоги до способів побудови і організації обчислювального процесу паралельною системою. Пропонована технологія розпаралелювання процесу факторизації ґрунтується на можливості зменшення відстані між співмножниками на числовій осі. Це досягається шляхом множення числа, що факторизується, на невеликі співмножники, які враховуватимуться при остаточному рішенні задачі. Представлений і проаналізований алгоритм факторизації, який здійснює пошук не співмножників, а доданків, що дозволяє швидко вирішувати задачу факторизації як при близьких співмножниках, так і при тих, що значно відрізняються один від одного. Алгоритм факторизації, що розглядається в статті, ґрунтований на рішенні нерівності, дозволяє відсікати велику кількість варіантів співмножників, які не є рішенням поставленої задачі. Розглянута в роботі можливість використання запропонованих алгоритмів на трьох процесорах. Проведений експеримент підтверджує гіпотезу про можливість збільшення швидкості рішення задачі факторизації збільшенням розміру числа, яке треба факторизувати. Результати експериментальних досліджень проілюстровані діаграмами, на яких наочно показаний виграш в часі при використанні запропонованої технології.

СПИСОК ЛІТЕРАТУРИ

1. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. М.: МЦНМО, 2003. 328 с.
2. Ишмухаметов Ш. Т. Методы факторизации натуральных чисел. Казань : Казан. ун-т, 2011, 190 с.
3. Shor P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer // Foundations of Computer Science : Conference Publications. 1997. P.1484–1509.

4. Manikandan V, Porkodi V, Mohammed AS, Sivaram M, "Privacy Preserving Data Mining Using Threshold Based Fuzzy cmeans Clustering", ICTACT Journal on Soft Computing, Volume 9, Issue 1, 2018, pp.1813-1816. **DOI:** 10.21917/ijsc.2018.0252
5. Porkodi V., Sivaram M., Mohammed A.S., Manikandan V. Survey on White-Box Attacks and Solutions. Asian Journal of Computer Science and Technology. Vol. 7, Is. 3. pp. 28–32.
6. Amin Salih Mohammed, Saravana Balaji B., Hiwa Abdulkarim Mawlood. Conceptual analysis of Iris Recognition Systems. Advanced Information Systems. 2019. Vol. 3, No. 2. P. 86-90. DOI : https://doi.org/10.20998/2522-9052.2019.2.15
7. Saravanan S., Hailu M., Gouse G.M., Lavanya M., Vijaysai R. Optimized Secure Scan Flip Flop to Thwart Side Channel Attack in Crypto-Chip. *International Conference on Advances of Science and Technology*, ICAST 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Vol 274. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-15357-1_34
8. Pevnev V. Pseudoprime Numbers: Basic Concepts And The Problem Of Security, 13th International Conference on Information and Communication Technology (ICT) in Education Research and Industrial Applications, 2017. P. 583-593.
9. Solovay R., V. Strassen. A fast Monte-carlo test for primality. In: SIAM J. Comput., V. 6, 1977. P.84-85.
10. Pevnev V. Investigation of the algorithm for the numbers primality determining Publication Year: 2018, P. 243 – 247
11. Певнев В. Я., Логвиненко Н.Ф., Шостак А.В. Алгоритм факторизации на основе решения квадратного неравенства Системи обробки інформації . 2001. №.3(13). С.13-16.

Рецензент: д-р техн. наук, проф. С. Г. Семенов,
Національний технічний університет «ХПІ», Харків
Received (Надійшла) 20.06.2019
Accepted for publication (Прийнята до друку) 14.08.2019

Технология построения параллельных алгоритмов факторизации

В. С. Харченко, В. Я. Певнев

Предметом изучения статьи являются алгоритмы факторизации больших чисел. Целью статьи является разработка технологии распараллеливания процесса факторизации и ее экспериментальное исследование. Задачи: повысить быстродействие предлагаемых алгоритмов за счет распараллеливания процесса факторизации больших чисел. Рассматриваются подходы для распараллеливания задачи факторизации, которые применяются в современных системах. Предлагаются критерии эффективности, на основании которых можно выдвигать требования к способам построения и организации вычислительного процесса параллельной системой. Предлагаемая технология основывается на возможности уменьшения расстояния между сомножителями на числовой оси. Это достигается путем умножения факторизуемого числа на небольшие сомножители, которые будут учитываться при окончательном решении задачи. Представлен и проанализирован алгоритм факторизации, который производит поиск не сомножителей, а слагаемых, что позволяет быстро решать задачу факторизации как при близких сомножителях, так и при значительно отличающихся друг от друга. Рассмотрена возможность использования предложенного алгоритма на трех процессорах. Проведенный эксперимент подтверждает гипотезу о возможности увеличения скорости решения задачи факторизации путем увеличения размера факторизуемого числа. Результаты экспериментальных исследований проиллюстрированы рисунками, на которых наглядно показан выигрыш во времени при использовании предложенной технологии. Рассматриваемый в статье алгоритм факторизации, основанный на решении неравенства, позволяет отсекать большое количество вариантов сомножителей, которые не являются решением поставленной задачи. Проведенные экспериментальные исследования на трехпроцессорной системе показали хорошие результаты решения задачи факторизации с использованием возможного распараллеливания этого процесса. Проведенные исследования по распараллеливанию задачи факторизации имеют большое значение как в теоретической плоскости, так и практическое значение при использовании систем шифрования с открытым ключом.

Ключевые слова: алгоритмы факторизации, технология распараллеливания, наибольший общий делитель, алгоритм квадратичного решета.

Technology of construction of parallel algorithms of factorization

V. Kharchenko, V. Pevnev

The subject of study of the article are algorithms for factoring large numbers. The aim of the article is to develop a technology for parallelization of the factorization process and its experimental research. Tasks: to increase the speed of the proposed algorithms by parallelizing the process of factoring large numbers. The approaches for parallelization of the factorization problem, which are used in modern systems, are considered. Efficiency criteria are proposed, on the basis of which it is possible to put forward requirements for the methods of constructing and organizing the computational process by a parallel system. The proposed technology is based on the possibility of reducing the distance between the factors on the numerical axis. This is achieved by multiplying the factoring number by small factors, which will be taken into account in the final solution of the problem. The factorization algorithm is presented and analyzed, which searches for non-factors, and terms, which allows you to quickly solve the factorization problem, both with close factors, and with significantly different from each other. The possibility of using the proposed algorithm on three processors is considered. The experiment confirms the hypothesis about the possibility of increasing the speed of solving the factorization problem by increasing the size of the factoring number. The results of experimental studies are illustrated by drawings, which clearly show the time gain when using the proposed technology. The factorization algorithm considered in the article, based on the solution of inequality, makes it possible to cut off a large number of variants of factors that are not the solution of the problem. Experimental studies conducted on a three-processor system have shown good results in solving the factorization problem using the possible parallelization of this process. The conducted studies on parallelization of the factorization task are of great importance both in the theoretical plane and also of practical importance when using public key encryption systems.

Keywords: factorization algorithms, paralleling technology, the greatest common divisor, quadratic sieve algorithm.