13. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний в информатике.- М.: Радио и связь, 1990.- 288с.;
14. Минаев Ю.Н., Филимонова О.Ю., Бенамеур Лиес. Методы и алгоритмы решения задач идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе.- М.: Горячая линия-Телеком, 2003.- 205с.;
15. Ивахненко А.Г. Моделирование сложных систем: (информационный подход).- К.:Вища шк. Головное изд-во, 1987.- 63с.;
16. Борисов А.Н., Алексеев А.В., Меркурьев Г.В., Слядзь Н.Н., Глушков В.И. Обработка нечеткой информации в системах принятия решений.- М.: Радио и связь, 1989.- 304с.;
17. Кофман А. Введение в теорию нечетких множеств.- М.: Радио и связь, 1982.- 432с.
18. Эйкхофф П. Основы идентификации систем управления. Оценивание параметров и состояния.- М.: Мир, 1975.- 676с.;
19. Ротштейн О.П., Ракитянська Г.Б. Діагностика на базі нечітких відношень в умовах невизначеності.- Вінниця: УНІВЕРСУМ-Вінниця, 2006.- 275с.;
20. Ротштейн А.П., Штовба С.Д. Нечеткая надежность алгоритмических процессов.- Винница: Континент-Прим, 1997.- 142с.;
21. Митюшкин Ю.И., Мокин Б.И., Ротштейн А.П. Soft Computing: идентификация закономерностей нечеткими базами знаний.- Винница: УНИВЕРСУМ-Винница, 2002.- 145с.;
22. Общая теория систем /Иванов А.М., Петров В.П., Сидоров И.С., Козлов К.А. - СПб.: Научная мысль, 2005. – 480;
23. Бубліченко С.В. Модель оптимізації і прогнозування електромагнітних систем з кільцевим ротором // Вісник Східноукраїнського національного університету ім.В.Даля, 2007. - №5(111). - С. 174 – 179;
24. И. Букур, А.Деляну, Введение в теорию категорий и функторов. М.: Мир, 1972.

УДК 691.3.06

Карпінський М.П., Якименко І.З., Хомінчук А.А.

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ РАХУВАННЯ КІЛЬКОСТІ ТОЧОК ЕЛІПТИЧНОЇ КРИВОЇ

В даній статті розглядається проблематика підрахунку кількості точок на еліптичній кривій. Розроблено метод з використанням технології паралельних обчислень операції піднесення числа до степеня по модулю $f_1(x)$, дозволяє ефективно визначати порядок еліптичної кривої з використанням алгоритму Шуфа. В статті наведено результати роботи програми, розробленої на основі даного методу.

Обчислення порядку групи точок ЕК над кінцевим простим полем має важливі застосування як в криптографії, так і в алгоритмах перевірки простих чисел.

Нехай p - просте число, $p>3$ еліптична крива $E=E_{a,b}$ над полем Z/pZ задана рівнянням $y^2=x^3+ax+b$. Для знаходження порядку $|E(Z/pZ)|$ Р.Шуф в 1985 році в роботі [6] запропонував алгоритм, який має поліноміальну складність $O(\log^2 p)$ бітових операцій. Подальше вдосконалення алгоритму Шуфа були запропоновані Алкіном, Ескізом, Мюллером і іншими авторами [1, 2, 3, 4, 5]. Це дозволило на практиці обчислювати порядки груп точок для простих полів, число елементів яких записується кількома сотнями десяткових цифр, найбільше значення $p=10^{499}+153$ [7]. Основна трудомісткість алгоритму Шуфа заключається в обчисленні високих степенів $x^p, y^p, x^{p^2}, y^{p^2}$ по модулю $f_l(x)$. Обчислення високих степенів має складність $O((\log p)M(n))$ і $O((\log p^2)M(n))$ відповідно бітових операцій.

Тому для прискорення роботи алгоритму Шуфа запропоновано використати технологію паралельних обчислень піднесення числа до степеня по модулю $f_l(x)$.

Отже, піднесення числа до степеня подається у вигляді:

$$x^p = \prod_n x^{\frac{p}{n}} \bmod f_l(x), \qquad (1)$$

де x – число, яке підноситься до степеня; n – кількість потоків (процесорів); p – степінь, до якого підноситься x; $f_l(x)$ – поліном степеня l.

З цієї формули неважко побачити завдання для потоку. Дійсно, кожен потік має обчислити $x^{\frac{p}{n}} \bmod f_l(x)$. Після обчислення всіх часткових степенів їх потрібно перемножити за модулем $f_l(x)$. Звичайно, цю операцію теж можна розпаралелити, але чи доцільно це? Адже кожен з часткових степенів не перевищуватиме модуля $f_l(x)$, тому множення часткових степенів відбудеться швидше в порівнянні з основною задачею. Для прикладу, якщо степінь $p = 8000$, але в дійсності цей степінь має біля 80-90 десяткових цифр, і обчислення розбивається на 8 потоків, то кожен потік має обчислити x^{800}. Тобто, кожен потік має здійснити 800 множень. Множення часткових степенів займе тільки вісім операцій множення, тобто в 100 разів менше. Тому розпаралелення множення часткових степенів є недоцільним.

Для виконання розпаралелення піднесення числа до степеня по модулю було створено спеціальний програмний продукт для операційних систем Windows та Linux з використанням бібліотеки LIP Ар'єна Ленстри. Оскільки ці програми відрізняються лише функціями створення та знищення потоків і методами синхронізації, то буде описано варіант продукту для операційної системи Windows. Варто зазначити, що в ОС Windows в якості механізму синхронізації використано семафори, а в ОС Linux – мутекси.

Задачу піднесення числа до степеня за модулем можна розбити на наступні елементи:

- обчислення поліномів;
- визначення степеня $\frac{p}{n}$ для кожного потоку;
- створення потоків.

Відповідно кожен потік має виконати наступні кроки:

- заблокувати доступ до глобальних змінних та скопіювати їх в свою пам'ять - використовується механізм семафорів;
- звільнити блокуючу змінну (семафор) та приступити до обчислення свого часткового результату;
- заблокувати доступ до глобальних змінних та занести в масив часткових результатів знайдене значення;
- перевірити, чи не виконалися решта потоків. Якщо так, то знайти кінцевий результат методом перемноження часткових результатів всіх потоків;
- завершити своє виконання.

Отже, в якості механізму синхронізації вибрано семафори та мутекси. Семафор – це змінна спеціального типу, яка є доступною будь-яким процесам для виконання двох операцій – «зайнято» та «відкрито». Проте в ОС Windows семафор є звичайним лічильником від нуля до будь-якого максимального значення. При розробці програмного продукту не виникло потреби використовувати семафори саме як лічильники, використовується сам факт їх існування, так званий «двійковий семафор». Отже, важливими є функції створення, знищення та перевірки існування семафору. API-функції ОС Windows реалізують ці задачі. Зупинимося на цих функціях детальніше:

- CreateSemaphore – функція створення семафору. Параметри: вказівник на структуру властивостей (можна задати 0), початкове значення лічильника, максимальне значення лічильника, ім'я створюваного семафору. Результат: handle семафору.
- ReleaseSemaphore – функція для збільшення лічильника. Параметри: handle се-

мафору, значення, на яке потрібно збільшити семафор та адреса попереднього значення. Результат – логічний признак успішності виконання операції (0 – виникла помилка, інакше повертається якесь число).

- OpenSemaphore – вертає handle семафору по вказаному імені. Параметри: тип доступу, логічний признак наслідування, ім'я семафору. Результат – handle семафору або 0, якщо його не існує.

Функція очікування WaitForSingleObject має два вхідних параметри – handle об'єкту, на який здійснюється очікування та час таймауту очікування. Семафор має статус «встановлено» при значенні лічильника більше нуля та стан «зайнято» при значенні лічильника 0. Функція очікує, поки семафор не набуде стану «встановлено», зменшує його лічильник на одиницю та завершується. Таким чином, після захоплення доступу до глобальних змінних семафор автоматично скидається в стан «зайнято» і потік виконує всі операції з глобальною пам'яттю. Після виконання всіх операцій потік викликає функцію ReleaseSemaphore, яка збільшує лічильник семафору на одиницю, після чого семафор «захоплюється» наступним потоком [8].

Мутекси використовуються як механізм синхронізації в ОС Linux. Створюється мутекс в момент його оголошення макросом DEFINE_MUTEX. Для звільнення мутексу використовується функція mutex_unlock, для очікування і блокування – функція mutex_lock. В якості параметру цим функціям передається адреса мутексу [9].

Саме по такому алгоритму працює розроблений програмний продукт. На рисунку 1 наведено робоче вікно програми.

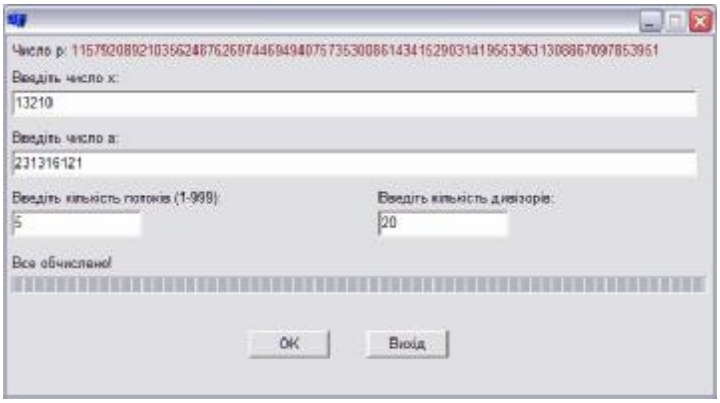

Рис. 1. Робоче вікно програми

При запуску програма автоматично визначає кількість процесорів, встановлених в системі і встановлює в це значення кількість потоків для того, щоб досягти максимального приросту продуктивності.

Після введення всіх необхідних значень користувач натискає кнопку «ОК», що запускає потік генерації дивізорів, обчислення модулів відбувається в окремому потоці для того, щоб коректно відображалися зміни на головному вікні програми.

Визначаються часткові степені $\frac{p}{n}$, створюються потоки та звільняється блокуючий семафор. Варто зазначити, що якщо кількість потоків відповідає кількості процесорів, встановлених в системі, програма здійснює прив'язку потоків до процесорів, це робиться для того, щоб два потоки не виконувалися на одному процесорі в той час як інший процесор буде вільним.

Після завершення обчислень програма видає вікно з інформацією про затрачений час. Результат обчислень не виводиться на екран, оскільки ним може бути дуже велике число, при експериментах результати досягали 60 тисяч десяткових знаків, тому воно виводиться в лог-файл Log.txt. В цей лог-файл записується інформація про кількість

процесорів, тільки в ОС Windows, повідомлення потоків про запуск та завершення роботи, виділення та звільнення пам'яті, також в лог-файл виводяться повідомлення про спроби виділити вже виділену пам'ять та звільнити невиділену пам'ять, результат обчислення, затрачений час та повідомлення про завершення роботи програмного продукту. Розроблений програмний продукт, який дозволяє розбити цю операцію на 1000 паралельних віток, після модифікації програми можна і більше, але варто зазначити, що найкращим варіантом кількості паралельних потоків є кількість процесорів, або процесорних ядер, встановлених в системі. При цій умові буде досягнуто максимальний приріст продуктивності та прискорення розпаралелення.

Висновки

У даній роботі авторами запропоновано технологію паралельного обчислення операції піднесення числа до степеня за модулем, що дозволяє ефективно визначати порядок еліптичної кривої з використанням алгоритму Шуфа. Дана методика дозволяє пришвидшити та досягнути максимального приросту продуктивності пошуку порядку на ЕК. Отже доцільно використовувати розроблений метод в криптосистемах, що базуються на еліптичних кривих для захисту інформації на практиці.

Література

1. Карпінський М., Ботюк А., Якименко І. "Підвищення ефективності обчислення точок на еліптичних кривих над обмеженими полями"// Вісник Тернопільського державного технічного університету імені Ів.Пулюя, - Том 8, - №4 - 2003, ст.67-73;
2. Blake I. F., Seroussi G., Smart N. P. Elliptic curves in cryptography. Cambridge University Press, 1999;
3. Elkies N.D. Elliptic and modular curves over .nite .elds and related computaional issues // Computational perspectives in number theory: Proc. of a Conf. in Honor of A.O. L.Atkin / J. T. Teitelbaum and D.A.Buell, editors. 1998. (Amer. Math. Soc. Inf. Press; V. 7). P. 21—76;
4. Muller V. Ein Algorithmus zur Bestimmung der Punktzahl elliptischer Kurvenuber endlichen Korpern der Characteristic grossen drei / PhD thesis, Universitat des Saarlandes, 1995;
5. Lercier R. Algorithmique des courbes dans les corps .nis / These. L'Ecole Polytechnique, Laboratoire D'Informatique, CNRC, Paris, 1997;
6. Schoof R. Counting points on elliptic curves over .nite .elds // J. Theorie des Nombres des Bordeaux. 1995. V. 7. P. 219—254;
7. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. — М.:МЦНМО, 2003.—328 с.;
8. http://www.firststeps.ru/mfc/msdn/r.php?81;
9. http://www.iti.cs.tu-bs.de/cgi-bin/UNIXhelp/man-cgi?mutex_lock+3THR.

УДК 519.711

Тишин П.М., Гайворонская Г.С., Ботнарь К.В.

НЕЧЕТКАЯ МНОГОКРИТЕРИАЛЬНАЯ ОЦЕНКА ПРОЕКТНЫХ РЕШЕНИЙ В МНОГОУРОВНЕВЫХ ИЕРАРХИЧЕСКИХ СИСТЕМАХ

В статье рассмотрен вопрос многокритериальной оценки эффективности проектных решений, которые принимаются для различных подсистем на разных уровнях иерархической системы в нечетких условиях. Построены лексикографические нечеткие отношения предпочтения. Определено множество парето-оптимальных вариантов проектных решений.

При решении технических задач, связанных с оптимизацией многоуровневых иерархических систем, часто недостаточно оценивать некоторые изменения в системе только по одному критерию. В тоже время, введение дополнительных критериев оценки может привести к ситуации, когда повышение показателей по одной группе критериев, сопровождается понижением показателей по другой группе критериев оценки. Кроме того, в сложных системах нередко возникает проблема с достоверностью или полнотой исходных