УДК 681.3

АНАЛІЗ МОДЕЛЕЙ ПАРАЛЕЛЬНОГО ПІДСУМОВУВАННЯ ЕЛЕМЕНТІВ ЧИСЛОВОГО МАСИВУ

К.т.н., доц. Мартинюк Т. Б., ас. Хом'юк В. В.,
студ. Куперштейн Л.Н., студ. Матвєєв Є. С.

Вступ

При моделюванні найпростіших нейроподібних структур існує проблема адекватного відтворення основних функцій нейрона, а саме, накопичення та порівняння із порогом множини вхідних сигналів [1,2]. У технічних засобах ці функції реалізуються як підсумовування масиву чисел у вигляді зважених величин і порівняння отриманої суми із порогом [3,4].

Отже, для досягнення більшої подібності з біологічними аналогами бажано реалізувати такі принципи обробки великих числових масивів, як максимальний паралелізм під час підсумовування і суміщення операцій підсумовування і порівняння із порогом[5,6]. Серед відомих способів розпаралелювання операції підсумовування найбільш придатним з цієї точки зору є алгоритм паралельного підсумовування масиву чисел із застосуванням різницевих зрізів (РЗ) [7].

В основі цього алгоритму використовується метод паралельного додавання чисел [8], який замість традиційного підсумовування чисел виконує накопичення кратних мінімальної складової всіх доданків на кожному кроці оброблення. Таким чином, замість фіксованого часу, що характерно для традиційного методу підсумовування, де у найкращому випадку цей час дорівнює 0 $(\log_2 n)$ (n кількість чисел у вхідному масиві), для алгоритму РЗ середній час оброблення можна визначити як випадкову величину, яка залежить не тільки від розмірності числового масиву, але й від кількості чисел, що повторюються , а також їх розподілу у масиві [7]. В результаті значний ефект можна отримати при підсумовуванні даним методом великих масивів чисел, розташованих навколо величин, що повторюються, а саме, при визначенні рельєфу кореляційної функції двовимірних зображень [9], при обчисленні цифрової двовимірної згортки [10] або для просторового додавання у нейронних системах [1,4].

Математична постановка задачі

Визначення середнього часу обробки або збіжності певного алгоритму набуває актуального значення при реалізації цього алгоритму у конкретному елементному базисі. Особливо, якщо використовується така специфічна елементна база як програмовані логічні інтегральні схеми (ПЛІС), оскільки в більшості випадків весь пристрій можна розташувати в одному чіпі, швидкодія якого відома[11]. Тоді значення середнього часу оброблення спрощує процес визначення швидкодії конкретного пристрою під час його комп'ютерного моделювання.

Тому, метою даної роботи є аналіз моделі паралельного підсумовування масиву чисел та отримання графіку залежності середньої кількості циклів підсумовування чисел в масиві від розмірності масиву.

Більшість випадкових величин, що зустрічаються на практиці, можуть бути представлені як суми досить великого числа достатньо малих доданків – елементарних помилок. Яким би законам розподілу не відповідали окремі елементарні помилки, сума виявляється підпорядкованою закону, який досить близький до нормального. Фактично нормальний закон розподілу являється граничним законом, до якого наближаються інші закони розподілу. Цей закон характеризується щільністю ймовірності виду:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}, \qquad (1)$$

де $m = M[X]$ – математичне сподівання[12], яке ще часто називають центром розсіювання; σ - середнє квадратичне відхилення. З даної формули безпосередньо випливає, що центром симетрії розподілу являється центр розсіювання *m*, тобто вираз (1) не змінюється на обернений при зміні знака різниці (*x*-*m*). Таким чином, в нормальному законі розподілу математичним сподіванням випадкової величини є її медіана. Щодо середнього квадратичного відхилення, то з умови $f_{\max} = \frac{1}{\sigma\sqrt{2\pi}}$ видно, що із збільшенням дисперсії ймовірність значень, віддалених від центру розсіювання, зменшується. Тому в дослідах, де інтервал елементів числового масиву обмежується величинами 0 та 1000, максимальне значення дисперсії D[X]=500 можна розглядати як граничне значення [12,13], що обумовлено також і відповідними формулами [12].

Таким чином, достатньо розглянути, для порівняння, нормальний та рівномірний закони розподілу випадкових величин. Для моделювання таких задач використаємо універсальний пакет прикладних програм (ППП) MathCAD 2000 Pro. При цьому згенеруємо 100 масивів чисел за нормальним і рівномірним законами розподілу випадкових величин по *p* елементів в кожному, де *p* = 6,8,10,…,24, а інтервал елементів масиву обмежимо величинами 0 та 1000.

У літературі [7] наведена емпірична формула для визначення середнього часу у тактах обробки числового масиву за методом РЗ, а саме:

$$N_{cep} = n - \sum_{r=1}^{R}(m_r - 1), \qquad (2)$$

де R – кількість груп з кількістю m_r чисел, що повторюються, причому m_r і R – випадкові величини. Доведемо це, для чого, спочатку, покажемо справедливість такого твердження:

$$N_{(n,\, m)} = n - m + 1, \qquad (3)$$

де $N_{(n,\, m)}$ - час обробки у тактах масиву n чисел, з яких m однакових. Нехай $n = 2$ і при цьому елементи рівні, тобто $m = 2$. Тоді алгоритм РЗ виконає 1 такт, внаслідок якого елементи обнуляться , тобто $N_{(2,\, 2)} = 1$. Нехай для $n = k$, серед яких l однакових, тобто $m = l$, виконується співвідношення $N_{(k,\, l)} = n - m + 1 = k - l + 1$. Покажемо, що дане співвідношення також виконується і для $n = k + 1$, серед яких є l однакових. Маємо $N_{(k+1,\, l)} = (k+1) - l + 1 = k - l + 1 + 1 = (k - l + 1) + 1 = N_{(k,\, l)} + 1$. На основі принципу математичної індукції можна зробити висновок, що при наявності n елементів числового масиву серед яких є m однакових, алгоритм виконає $t_{(n,\, m)} = n - m + 1$ тактів, що і потрібно було довести.

Далі покажемо справедливість співвідношення (2). Нехай $R = 1$, це означає, що $N = n - m_1 + 1$, що вірно, оскільки справедливе твердження 1. Припустимо, що співвідношення (2) виконується і для $R = k$, тобто $N = n - \sum_{r=1}^{k}(m_r - 1)$. Покажемо, що воно має місце і для $R = k + 1$. Маємо

$$N = n - \sum_{r=1}^{k+1}(m_r - 1) = n - (m_1 - 1 + \sum_{r=2}^{k+1}(m_r - 1)) = n - \sum_{r=2}^{k+1}(m_r - 1) - m_1 + 1 =$$

$= n_k - m_1 + 1$, що виконується за формулою (3). Таким чином, рівність (2) виконується для довільної кількості груп R з m_r числами, що повторюються.

Для практичного підтвердження формул (2) і (3) проведемо комп'ютерне моделювання паралельного підсумовування елементів масиву чисел.

Моделі алгоритмів та їх варіантний аналіз

В ході вирішення задачі було створено декілька різних алгоритмів для її розв'язку. Із математичної моделі методу різницевих зрізів (МРЗ)[7] випливає, що якщо в масиві немає однакових елементів, то кількість ітерацій (num) буде дорівнювати кількості елементів (n). Якщо в масиві є два однакових елементи, то num = n - 1, якщо 3 або 2 по 2 однакових, то num = n - 2 і так далі у відповідності з виразом (2).

На основі цього міркування був створений алгоритм, в якому елементи масиву порівнюються самі з собою (рис. 1). Введення елементів масиву відбувається окремо. Алгоритм має таку структуру:

1. Кількості ітерацій num надається значення кількості елементів у вхідному масиві n.
2. У двох вкладених циклах виконуються наступні дії:
   a) кожен елемент масиву порівнюється з іншим;
   b) відбувається перевірка на неоднаковість індексів елементів масивів для того, щоб елемент не порівнювався сам з собою.

   Якщо елементи однакові, то кількість ітерацій зменшується на одиницю, якщо ні - відразу перевіряється наступний елемент.
3. Після перевірки всіх елементів до кількості ітерацій додається половина її різниці з кількістю елементів масиву, оскільки для однакових елементів з різними індексами декремент кількості ітерацій буде відбуватися і для того випадку, коли вони поміняються індексами.

Цей алгоритм достатньо простий і наочний, але його основним недоліком є перебір всіх елементів масиву двічі для порівняння. З цього і випливає необхідність робити після перевірки всіх елементів поправку кількості ітерацій, щоб уникнути її подвійного декрементування. Такого недоліку позбавлений другий варіант алгоритму, який створений на основі першого і значною мірою схожий на нього. Його відмінність полягає в поєднанні введення елементів масиву з їх безпосереднім порівнянням (рис. 2):

1. В циклі послідовно вводяться всі елементи масиву.
2. Для кожного з них виконується ще один вкладений цикл, в якому елемент масиву порівнюється зі всіма елементами, що були введені до нього. Якщо він дорівнює якомусь з них, то кількість ітерацій зменшується на одиницю, а якщо ні, то перевіряється наступний елемент.
3. Завдяки тому, що кожний елемент масиву порівнюється не зі всіма іншими, а тільки з тими, що мають менший індекс, число порівнянь (а відповідно і час роботи) зменшується в два рази. Тим самим вирішується і проблема подвійного декрементування.

Але, незважаючи на простоту, обидва дані алгоритми мають один вагомий недолік. Він полягає в тому, що у разі, коли кількість однакових елементів перевищує два, то відбувається декрементування з порядком більшим ніж два. Це приводить до додаткового зменшення кількості ітерацій. Тому розроблені алгоритми можна використовувати тільки для числових послідовностей, сформованих, наприклад, за нормальним законом розподілу з великою дисперсією, в яких однакові елементи зустрічаються достатньо рідко. Для інших послідовностей ці алгоритми будуть давати наближений результат.

Щоб уникнути цієї проблеми був розроблений третій, остаточний варіант алгоритму, який повністю відповідає математичній моделі підсумовування за МРЗ (рис. 3) і має такий вигляд:

1. Відбувається обнулення значень seq (сумарна кількість однакових елементів), num (кількість ітерацій), eq (кількість однакових елементів для даної ітерації), причому eq обнулюється вже в блоці з післяумовою.
2. Викликається процедура Min знаходження найменшого елемента масиву m (елементи масиву вводяться в окремій процедурі попередньо), блок-схема якої не наводиться за несуттєвістю.
3. Запускається цикл, який виконується для всіх елементів масиву. В ньому елемент масиву порівнюється з мінімальним. Якщо вони дорівнюють один одному, то елемент масиву обнулюється і до кількості елементів в даній ітерації додається одиниця, якщо ж елемент більший за мінімум, то йому присвоюється значення їх різниці.
4. Після закінчення цього циклу кількість ітерацій збільшується на одиницю, і до значення seq додається значення eq.
5. Перевіряється умова рівності сумарної кількості рівних елементів загальній кількості елементів масиву. Якщо вона не виконується, то

процедура повторюється з відповідного місця (див. рис. 3); в протилежному випадку процедура завершується.

І, хоча цей алгоритм достатньо громіздкий, він точно виконує поставлену задачу. Програма побудована таким чином, що процедура підрахування кількості ітерацій викликається по 100 разів (за кількістю масивів) для кожного файлу з однієї групи (групи сформовані за нормальним та рівномірним законами розподілу) в процедурі CalcFile, в якій потім підраховується середня кількість ітерацій і виводиться на екран та у вихідний файл. Процедурі CalcFile передається параметр - назва групи файлів (n., n1. чи r.), тобто вона тричі викликається в основній програмі.

Програма, що реалізує алгоритм, написана на мові програмування Turbo Pascal 7.0. Ця мова дозволяє достатньо легко і швидко виконати поставлену задачу, не користуючись мовами низького рівня або спеціалізованими мовами для моделювання.

Результатом роботи програми є файли для масивів з нормальним (2 типи, які відрізняються величиною σ) та рівномірним розподілами, тобто три файли з іменами n.dat, n1.dat, r.dat відповідно до кожної групи вхідних файлів. В них містяться 10 рядків, в кожному з яких вказана середня кількість ітерацій для відповідного масиву чисел. Число в кожному рядку має дійсний вигляд. Такий формат вихідних файлів найбільш зручний для подальшого їх аналізу за допомогою математичного пакету MathCAD 2000 Pro.

Аналіз моделі паралельного підсумовування масиву чисел

При генерування масивів чисел розподілених за нормальним та рівномірним законами розподілу розглянуто відповідно два випадки.

1. Коли математичне сподівання m=500, дисперсія D=500, а середнє квадратичне відхилення $\sigma=\sqrt{D}=22.36068$. Цей випадок розглядається, виходячи з аналізу літератури [12,13], оскільки дисперсія, як і середнє

квадратичне відхилення, є мірою розсіяння випадкової величини навколо її математичного сподівання.

2. Коли математичне сподівання m=500, а середнє квадратичне відхилення σ=150. Цей випадок розглядається виходячи з варіантного аналізу середнього квадратичного відхилення. Тобто, одержано, що σ=150 – значення середнього квадратичного відхилення, при якому згенеровані елементи масивів попадають в проміжок від a = 0 до b =1000.

В результаті отримано середню кількість ітерацій підсумовування масиву чисел, розподілених за рівномірним і нормальним законами, із p елементів, де p=6,8,10,…,24, які теж записані у файли. Зчитавши їх у системі MathCAD 2000 PRO, отримаємо такий результат :

$$yn_i := READ(\text{"N.dat"}) \quad yn1_i := READ(\text{"N1.dat"}) \quad yr_i := READ(\text{"R.dat"})$$

yn =

	0
0	5.99
1	7.95
2	9.91
3	11.91
4	13.87
5	15.7
6	17.79
7	19.56
8	21.43
9	23.45

yn1 =

	0
0	5.83
1	7.65
2	9.43
3	11.24
4	12.81
5	14.53
6	16.32
7	17.64
8	19.5
9	21

yr =

	0
0	5.99
1	7.99
2	9.99
3	11.87
4	13.9
5	15.85
6	17.86
7	19.79
8	21.85
9	23.69

де: yn – масив, який зчитаний з файлу N.dat і містить середні кількості ітерацій підсумовування масивів чисел із p елементів, де p=6,8,…24, розподілених за рівномірним законом з σ=150. Тобто, для 6-ти елементів масиву середня кількість ітерацій підсумовування виражається першим елементом масиву yn (yn1, yr), для 8-ми елементів – другим і т. д.;

yn1 – масив, який зчитаний з файлу N1.dat і містить середні кількості ітерацій підсумовування масивів чисел із p елементів, де p=6,8,…24, розподілених за нормальним законом з σ=22.36;

yr – масив, який зчитаний з файлу R.dat і містить середні кількості ітерацій підсумовування масивів чисел із p елементів, де p=6,8,…24, розподілених за рівномірним законом.

Узагальненням усієї виконаної роботи є графік (рис. 4), де по осі абсцис відкладена кількість елементів у масиві, а по осі ординат – середні значення ітерацій підсумовування елементів цих масивів чисел. Варіювання середнім квадратичним відхиленням σ нормального закону розподілу дає можливість отримати деяку функцію, яка залежить від трьох змінних, елементами якої є кількість елементів у вхідному масиві P, середня кількість циклів підсумовування цього масиву чисел $N_{сер}$ та середнє квадратичне відхилення σ. Ця залежність зображена на рисунку 5.

Висновки

1. На основі роботи програми написаної за алгоритмом підсумовування чисел в масиві за методом різницевих зрізів були отримані графіки залежності середньої кількості циклів від розмірності масиву. Виходячи з графіків можна легко визначити для будь-якої кількості операндів середню кількість ітерацій їх підсумовування. Причому кількість ітерацій для масивів з меншим середньоквадратичним відхиленням σ = 22,36 при нормальному розподілі є меншою, ніж для масивів з σ = 150, оскільки розкид значень для чисел з меншим σ буде меншим. Тому при однаковій для обох масивів кількості елементів, у масиві з більш пологим "колоколом" розподілу буде більша кількість однакових елементів. Рівномірний розподіл при заданих умовах дає незначну кількість однакових елементів у масиві.

2. Побудовані графіки дають можливість визначити при відомих мінімальному і максимальному значеннях середній час підсумовування масиву чисел виду: Т = $N_{сер} \cdot T_ц$, де $N_{сер}$ – середня кількість циклів операції підсумовування масиву чисел, $T_ц$ – час виконання одного циклу. Подальше використання цих результатів необхідне для дослідження роботи, параметрів

та властивостей паралельного конвеєрного процесора, на якому реалізується досліджений алгоритм підсумовування масиву чисел.

Література

1. The neural and neural-like networks: synthesis, realization, application and future / V.V. Hrytsyk at el. // Інформаційні технології і системи. – 1998.- т.1. - №1/2. – с.15-55.
2. Нейрокомпьютеры и интеллектуальные роботы / Под ред. Н.М.Амосова. – К.: Наукова думка, 1991. – 272 с.
3. Лорьер Ж. – Л. Системы искусственного интеллекта: Пер. с франц. – М.: Мир, 1991. – 568 с.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. Пер. с англ. – М.: Мир, 1992. – 240 с.
5. Перцентрон – система распознавания образов / Под ред. А.Г.Ивахненко. – К.: Наукова думка, 1975. – 431 с.
6. Хьюбел Д. Глаз, мозг, зрение: Пер. с англ. – М.: Мир, 1990. – 239 с.
7. Мартинюк Т.Б. Рекурсивні алгоритми багатооперандної обробки інформації. Монографія. – Вінниця: “Версум – Вінниця”, 2000. – 216 с.
8. Свечников С.В., Кожемяко В.П., Тимченко Л.И. Квазиимпульсно – потенцеальные оптоэлектронные элементы и устройства логико – временного типа. – К.: Наукова думка, 1987. – 256 с.
9. Martyniuk T., Kozhemjako A., Homchuk M. Relief Determination of Correlation Function in Image Processing // Праці 3-ї Всеукраїнської МНТК “Обробка сигналів і зображень та розпізнавання образів УКРОБРАЗ’96.”- Київ, 1996.- С. 90-91.
10. Очин В.Ф. Вычислительные системы обработки изображений.- Л.:Энергоатомиздат, 1989.- 136 с.
11. Антонов А.П., Мелехин В.Ф., Филиппов А.С. Обзор электронной базы фирмы ALTERA.- С.-Петербург: “Файнстрит”, 1997.- 144 с.

12. Гнеденко Б.В. Курс теории вероятностей. - 6-е изд. - М.: Наука, 1988. - 450 с.
13. Смирнов Н.В., Дунин-Барковский Н.В. Курс теории вероятностей и математической статистики для технических приложений. - 3-е изд. - М.: Наука, 1969.- 320 с.