УДК 519.15:621.372

О. Різник, Є. Вдовенко, В. Буцик
Національний університет “Львівська політехніка”,

ВИКОРИСТАННЯ ЛІНІЙОК ГОЛОМБА ТА ІДЕАЛЬНИХ КІЛЬЦЕВИХ В’ЯЗАНОК ДЛЯ ОПТИМАЛЬНИХ ВІДНОВЛЮЮЧИХ СИСТЕМ У РОЗПОДІЛЕНИХ ОБЧИСЛЕННЯХ

© Різник О., Вдовенко Є., Буцик В., 2011

Розглянуто оптимальні схеми відновлення для розподілених обчислень на основі ідеальних кільцевих в’язанок. Розроблено методику синтезу схем відновлення на основі теорії числових в’язанок, що дає можливість у випадку несправності одного чи більше комп’ютерів рівномірно перерозподіляти навантаження на інші робочі комп’ютери.

Ключові слова: ідеальна кільцева в’язанка, розподілені обчислення, схема відновлення.

In the article the optimal charts of renewal are examined for the up-diffused calculations on the basis of ideal ring bundles. The worked out methods of synthesis of charts of renewal are on the basis of theory of numerical bundles, which enables in the case of disrepair one or more computers evenly to redistribute loading on other working computers.

Keywords: ideal ring bundle, up-diffused calculations, chart of renewal.

Вступ

Розподілені обчислення – це один із способів розв’язання трудомістких завдань за допомогою використання декількох комп'ютерів, об'єднаних у мережу. Розподілені обчислення являють собою частковий випадок паралельних обчислень, тобто розв’язують різні частини однієї задачі декількома процесами одного або більше комп'ютерів [2].

Існує безліч способів виконання паралельних обчислень. Наприклад, це вільно зв’язана паралельна система, що полягає у поєднанні автономних комп'ютерів, сполучених мережею, де процес виконується процесором, на якому він був запущений. Цей випадок називається статичним розподілом. Але вільно зв’язані системи мають ряд недоліків, основний з яких – це планування процесів між процесорами для досягнення продуктивності, а також зменшення часової затримки при комунікації і час виконання. Інший приклад паралельних обчислень – це жорстка в’язанка або симетрична багатопроцесорна система. Ця система побудована на множині подібних процесорів у межах одного комп'ютера, зв'язаних шиною або іншою мережею взаємозв'язку. Вважається, що процес може виконуватись різними процесорами. Такий спосіб обчислення називається динамічним розподілом. Він гарантує високу продуктивність, але не є придатним, адже за відмови одного процесора обчислення приречені на невдачу [6].

Єдиний шлях отримання високої надійності і незалежності від помилок – виконувати додаток на кластерній або розподіленій системі. Є головний перший комп'ютер, який виконує додаток за нормальних умов і другий комп'ютер, який приймає задачу, коли перший комп'ютер вимикається. Також є й третій комп'ютер, який приймає задачу, коли перший і другий комп'ютери вимкнені і так далі. Порядок, в якому комп'ютери використано, називається порядком відновлення, в якому отримані – списком відновлення.

Перевага користування кластерами, окрім меншої залежності від помилок, − навантаження, розподілене між комп'ютерами. Коли усі комп'ютери працюють, бажано, щоб навантаження було розподілене порівну. Проте навантаження на деяких комп'ютерах зростатиме, коли один або більше комп'ютерів виходять з ладу, але і за цих умов необхідно розподілити навантаження якомога порівну на комп'ютери, що залишилися робочими.

Поширення навантаження, коли комп'ютер вимкнений, вирішене порядком відновлення процесів, що проходять на непрацюючому комп'ютері. Безліч усіх порядків відновлення є так званою схемою відновлення, тобто поширення навантаження у разі, коли один або більше непрацюючих комп’ютерів визначається схемою відновлення. Задача виявлення оптимальних (або навіть кращих) схем відновлення раніше не була повною мірою вивчена іншими дослідниками.

Постановка проблеми

Мета цієї статті − знайти оптимальні схеми відновлення, які повинні швидко обчислюватись для великої кількості n комп’ютерів і працювати краще, ніж уже відомі схеми.

Розглянемо кластер з n ідентичних комп'ютерів. Є один процес на кожному комп'ютері. Робота рівномірно розділена на ці n процеси. Є список відновлень, асоційований з кожним процесом. Цей список визначає, де процес треба знову почати, якщо поточний комп'ютер вимкнений. На рис. 1 показано таку систему для $n = 4$. Процес переміщається назад, як тільки непрацюючий комп'ютер повертається до робочого стану.

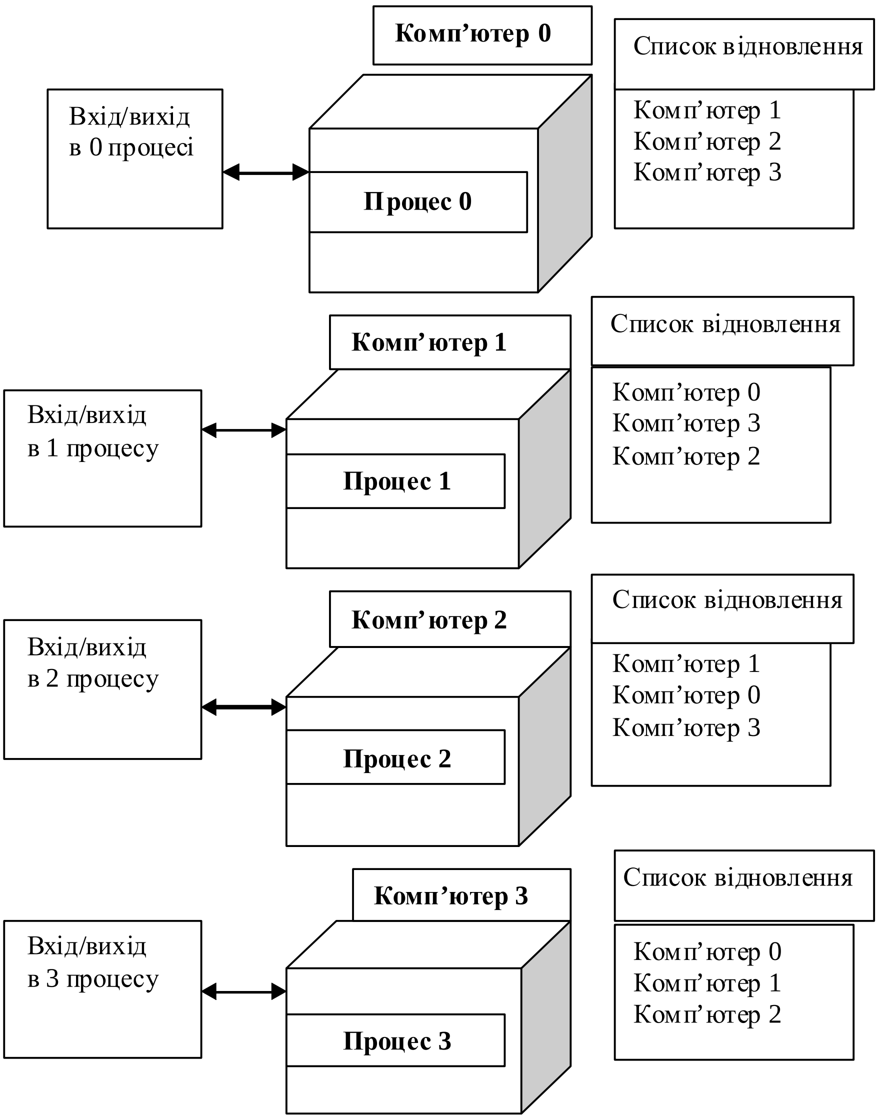

Рис. 1. Схема відновлення для $n = 4$

У більшості кластерних систем це може бути встановлено користувачем або здійснюється автоматичний перерозподіл процесів, коли непрацюючий комп'ютер відновлює свою роботу. Наведений один список відновлення для кожного процесу. Сукупність усіх списків відновлень називається схемою відновлення. Права сторона рис. 1 показує випадок, коли перший комп'ютер вимкнений. Список відновлення для нульового процесу показує, що він повинен бути запущений на комп’ютері один, доки нульовий комп’ютер вимкнений. Якщо комп'ютер один також ламається, нульовий процес виконується на комп'ютері два, який є другим комп'ютером у списку відновлень.

Перший комп'ютер в списку відновлень процесу один – нульовий комп’ютер. Проте, коли нульовий комп’ютер вимкнений, процес один буде знову розпочатий на комп'ютері три. Тому, якщо комп'ютери нуль і один не працюють, є два процеси на комп'ютері два (процес нуль і два) і два процеси на комп'ютері три (процеси один і три).

Якщо комп’ютери нуль і один вимкнені, максимальне навантаження на кожному з тих, що залишилися − вдвічі більше за нормальне навантаження. Це добрий результат, навантаження оптимально розподілене, як тільки це можливо. Проте, якщо комп'ютер нуль і два вимкнені, є три процеси на комп'ютері один (процеси нуль, один і два), тобто максимальне навантаження втричі більше за нормальне на найбільш завантаженому комп'ютері.

Тому, для схеми відновлення на рис. 1, комбінація вимкнених комп’ютерів нуль і два несприятливіша, ніж комбінація вимкнених комп’ютерів нуль і один. Результати також дійсні, коли є n систем даних усередині кластера, наприклад, один телекомунікаційний центр постачає дані для кожного комп'ютера у кластері. Якщо комп'ютер відключається, центр повинен направити свої дані на деякий інший комп'ютер у кластері, тобто повинен бути "список відновлень", що асоціюється з кожним центром [2]. Знайти оптимальні схеми відновлення − фундаментальна задача в розподілених і кластерних системах. Припустимо, що робота, яка виконується кожним з n комп'ютерів, має бути переміщена як одна атомна одиниця. Приклади − системи, де уся робота, що виконується комп'ютером, робиться від однієї зовнішньої системи або коли усю роботу виконує один процес системи, де зовнішньою комунікацією управляє IP протокол [1] (у цьому випадку увесь зовнішній рух за однією мережевою адресою спрямовується за мережевою адресою як одна атомна одиниця).

Розв’язання поставленої задачі

Переформулюємо задачу в цій статті: необхідно отримати число, до якого можна знайти найдовшу послідовність додатних цілих чисел таких, що сума послідовності є менша або дорівнює сумі чисел, і така, що усі суми послідовностей (зокрема послідовності довжини один) унікальні.

Ця задача еквівалентна задачі знаходження лінійки Голомба за умови, що сума послідовності є менша або дорівнює L_n, і еквівалентна задачі знаходження ідеальної кільцевої в’язанки за умови, що сума послідовності точно дорівнює S_n, тому відповідно можемо користуватися результатами лінійок Голомба та ідеальних кільцевих в’язанок.

У математиці лінійкою Голомба називається набір невід'ємних цілих чисел, розташованих у вигляді поділок на уявній лінійці так, що відстань між будь-якими двома поділками є унікальною. Іншими словами, на всій лінійці не можна знайти два числа, різниця між якими повторювалася б двічі [1, 4, 7].

Число поділок на лінійці Голомба називають її порядком N, а найбільшу відстань між двома її поділками – довжиною. Іноді лінійки Голомба описуються відстанями n між сусідніми поділкA-ми, а не абсолютними координатами поділків, де $N = n+1$. Прикладом такої лінійки є $1,3,5,2$ [3, 5].

Загальна кількість пар поділок для лінійки з N поділками дорівнює числу сполучень з N по 2. Максимальне число пар, які можна скласти з поділок лінійки порядку N, визначають за формулою:

$$\binom{2}{N} = \left(\frac{N(N-1)}{2}\right). \tag{1}$$

Максимальне число пар, які можна скласти з відстаней n між сусідніми поділками лінійки порядку N, визначають за формулою:

$$L_n = \binom{2}{n} = \left(\frac{n(n+1)}{2}\right). \qquad (2)$$

Лінійки Голомба одержують за цією формулою шляхом зсуву кожної поділки на фіксоване число, наприклад, 1, 2, 5, 10, 12 або перерахуванням поділок лінійки у зворотному порядку. Тому в канонічному поданні для лінійок Голомба найменша поділка відповідає нульовій координаті, а наступна за ним поділка розташовується на найменшій з двох можливих відстаней.

Лінійку Голомба називають оптимальною, якщо не існує коротших лінійок того самого порядку. Лінійка є оптимальною, якщо в канонічному поданні значення її останньої поділки мінімально можливе. Відстані між поділками, кратні деякому мінімальному значенню, відповідають невід'ємним цілим числам, причому відстань між будь-якою парою поділів повинна бути унікальною.

Простою ідеальною кільцевою в'язанкою (ІКВ) називається послідовність $K_n = (k_1, k_2, ..., k_n)$ чисел, на якій всі можливі кільцеві суми вичерпують значення чисел натурального ряду $1, 2, ..., S_n$, де:

$$S_n = n(n-1)+1. \qquad (3)$$

Зазначимо, що в цьому випадку поняття кільця має зовсім інший зміст, ніж в алгебричній теорії груп.

Наведемо приклад простої ІКВ четвертого $(n=4)$ порядку. За формулою (3) маємо $S_4 = 4(4-1)+1 = 13$.

У таблиці кільцевих сум, які відповідають цій ІКВ, містяться всі числа натурального ряду від 1 до 13.

Кільцеві суми для ІКВ шостого порядку $(1, 3, 2, 7)$

p_j	q_j			
	1	2	3	4
1	1	4	6	13
2	13	3	5	12
3	10	13	2	9
4	8	11	13	7

Як видно з формул (2) та (3), значення суми ІКВ S_n приблизно в рази більше ніж сума лінійки Голомба L_n, тому доцільніше для побудови списків відновлень застосовувати ІКВ. Список відновлень отримується додаванням значень послідовностей − це послідовність з часткових сум. Перша частина списків відновлень складається з лінійки Голомба або ІКВ, тобто перші x входи для найбільшого x такі, що сума оптимальної послідовності довжини x менша, ніж L_n+1 для Голомба або S_n+1 для ІКВ. Частина списку відновлень, що залишилася, наповнена залишком номерів (комп'ютерів) аж до суми лінійки Голомба L_n або суми ІКВ S_n.

Нехай $V_{0,i}$ є списком відновлення лінійки Голомба для процесу нуль у кластері з j комп'ютерами. Тоді він містить лінійку Голомба з сумою, меншою ніж *j* або такою самою і залишок номерів (комп'ютерів) аж до *j*−1. Наприклад, для процесу нуль у кластері з 12 комп'ютерів, маємо: $V_{0,12}(0) = \{1,4,9,11,2,3,5,6,7,8,10\}$. Усі інші списки відновлень виходять з $V_{0,j}(0)$, використовуючи $V_{i,j}(x) = (V_{0,j}(0)+i)\bmod(j+1)$ для усіх $i \le j$.

Нехай $V_{0,j}$ є списком відновлення ІКВ для процесу нуль у кластері з j комп'ютерами. Тоді він містить ІКВ із сумою, що дорівнює j і залишок номерів (комп'ютерів) аж до *j*−1. Наприклад, для процесу нуль у кластері з 14 комп'ютерів маємо: $V_{0,14}(0)=\{1,4,6,13,2,3,5,7,8,9,10,11,12\}$. Усі інші списки відновлень для процесів x отримуємо з $V_{0,j}(0)$, використовуючи $V_{i,j}(x)=(V_{0,j}(0)+i)\mathrm{mod}(j+1)$ для всіх $i \le j$.

Користуючись схемою лінійок Голомба, можна гарантувати оптимальну поведінку тільки до 26 пошкоджених комп'ютерів, де $L_{26}=492$, оскільки для більших n ще не відомо, чи відповідні лінійки Голомба оптимальні або ні. Користуючись схемою ІКВ, можна гарантувати близьку до оптимальної поведінку для будь-якої кількості пошкоджених комп'ютерів, оскільки Зінгерові ІКВ існують для n за умови:

$$n = p^a + 1, \tag{4}$$

де p – просте число; a – натуральне число.

Висновок

У багатьох кластерних і розподілених системах розробник повинен забезпечити схему відновлення. Такі схеми визначають, як робоче навантаження треба перерозподілити, коли один або більше комп'ютерів виходять з ладу. Мета − розподілити навантаження порівну, навіть за найбільш несприятливих комбінацій комп'ютерів, що вийшли з ладу, тобто оптимізувати поведінку за найгіршого випадку.

Розглянуто n ідентичних комп'ютерів, які за нормальних умов виконують один процес. Усі процеси виконують ту ж кількість роботи. Схемою відновлення, що гарантує оптимальне поширення навантаження за найгіршого випадку, коли x комп'ютерів вийшли з ладу, є так звана оптимальна схема відновлення для значення n і x на основі лінійок Голомба та ІКВ.

1. Bloom, G. S., Golomb, S. W., Applications of Numbered, Undirected Graphs, Proceedings of the IEEE, Vol. 65, No. 4, April, 1977, pp. 562−571. 2. Pfister, G. F., In Search of Clusters, Prentice−Hall, 1998. 3. Dimitromanolakis, A., Analysis of the Golomb Ruler and the Sidon Set Problems, and Determination of large, near−optimal Golomb Rulers, Department of Electronic and Computer Engineering Technical University of Crete, June, 2002. 4. Soliday, S. W., Homaifar, A., Lebby, G. L., Genetic Algorithm Approach to the Search for Golomb Rulers, International Conference on Genetic Algorithms, Pittsburg, PA, USA, 1995, pp. 528−535. 5. Dollas, A., Rankin, W. T., McCracken, D., A new Algorithm for Golomb Ruler Derivation and Proof of the 19 Marker Ruler, IEEE Transactions on Information Theory, Vol. 44, No. 1, January 1998, pp. 379−382. 6. Hayes, B., Computing Science: Collective Wisdom, American Scientist, Vol. 98, No. 2, March−April, 1998, pp. 118−122. 7. http://en.wikipedia.org/wiki/Golomb_ruler.