

В.В. Душеба, м. Київ
О.О. Сігарьов, м. Київ
І.М. Лях, Ужгород

АНАЛІЗ ЧАСОВИХ ВТРАТ НА СИНХРОНІЗАЦІЮ І ПЕРЕДАЧУ СЛУЖБОВОЇ ІНФОРМАЦІЇ В БАГАТОПРОЦЕСОРНИХ СИСТЕМАХ

Abstract. The article provides an analysis of time losses to ensure the synchronization of processes and the transfer of service information in multiprocessor computing systems.

Вступ

Необхідність організації взаємодії обчислювальних вузлів в багатопроесорних системах тягне за собою виконання деяких дій, таких як синхронізація паралельних обчислень і передача службової інформації, що призводить до додаткових часових втрат [1 – 4]. Латентності, підтримання когерентності кешів, синхронізація, перекомутація каналів зв'язку в сучасних комп'ютерах поглинають більше половини корисної продуктивності.

Оцінка часових втрат на передачу службової інформації і синхронізацію обміну

Два чинники – втрати на передачу службової інформації і синхронізацію обміну – мають істотну і принципову відмінність від тексту повідомлення. З ростом числа вузлів в системі величина тіла повідомлень буде зменшуватися, в той час як втрати на передачу службової інформації і синхронізацію обміну будуть зростати прямо пропорційно n і починаючи з певного моменту домінувати в системі. З урахуванням впливу згаданих чинників коефіцієнт прискорення набуває вигляду:

$$k = \frac{T_{\text{посл}}}{\frac{T_{\text{посл}}}{n} + 2\left(\frac{\alpha W}{n} + \tau\right)(n-1)}, \quad (1)$$

τ – час передачі службової інформації (заголовка повідомлення) і витрат на синхронізацію обміну, латентності.

Знайдемо граничне значення для виразу (1):

$$k_{\text{пр}} = \lim_{n \rightarrow \infty} k = 0.$$

Зменшення прискорення з ростом числа обчислювальних вузлів (ОВ) в системі обумовлено впливом τ .

Дослідження функції (1) показує, що вона має максимум при числі

обчислювальних вузлів:

$$n_{opt} = \sqrt{\frac{T_{носл} - 2\alpha W}{2\tau}}, \quad (2)$$

$$k(n_{opt}) = \max k < k_{crit}.$$

$$n_1 = \frac{T_{лосл} - 2\alpha W}{2\tau}.$$

Після проходження максимуму функція (1) монотонно і дуже повільно збуває. При $n_1 = n_{opt}^2$, вона знову прийме значення рівне 1. Іншими словами, якщо $n_{opt} = 10^2$, то, за умови $n_1 = 10^4$, алгоритм буде мати прискорення рівне 1, тобто додавання 9900 процесорів зробить продуктивність системи, яка дорівнює продуктивності одного процесора.

Слабке місце систем з десятками тисяч процесорів полягає в тому, що максимальне розпаралелювання програми може дати нульове прискорення.

З виразу (2) слідує, що $\left(\frac{f_{лц}}{2f_{гд}} < 1\right) \leftrightarrow (k_{crit} < 1)$. Це означає, що

збільшення частоти процесорів вище деякого граничного значення при незмінній частоті $f_{гд}$ неминуче призводить до прискорення, яке менше одиниці на всьому робочому інтервалі алгоритму.

Розглянута модель має деякі особливості. З одного боку вона ніяк не враховує часові витрати на формування повідомлень, неминучі затримки, пов'язані з перекомутацією каналів передачі і очікуваннями звільнення зайнятих каналів і т.п. З іншого боку немає обмежень на інтенсивність потоків міжпроцесорних передач даних і ступінь розпаралелювання алгоритму. Вона задає верхню межу прискорення для конкретного алгоритму при використанні механізму передачі повідомлень. Звернемося до конкретних числових прикладів, на яких розглянемо деякі інші властивості механізму передачі повідомлень.

Приклад 1. Побудуємо діаграму прискорень за формулою (1) для алгоритму і системи з конкретними числовими параметрами (рис. 1).

Параметри алгоритму: $T_{носл} = 10^4, W = 10^2$.

Параметри системи: $\alpha = 5, \tau = 45, N_p = 100(\text{число ВУ})$.

Визначимо: $k_{crit} = 10, \beta = 10^2, n_1 = 100, n_{opt} = 10, \max k = 3.7$.

Діаграма прискорень займає кінцевий інтервал $[1, W]$ цілочисельних значень n , який назвемо робочим інтервалом алгоритму на даній системі. У загальному випадку робочий інтервал ділиться на дві області: зростання і зменшення прискорення реалізації алгоритму. Область збування займає 90% загального обсягу діаграми. Вона не повинна бути використана даними

алгоритмом. При рівномірній шкалі діаграма буде асиметричною з дуже пологою областю убунання. З цієї причини далі буде використана нерівномірною шкала числа ОВ області убунання.

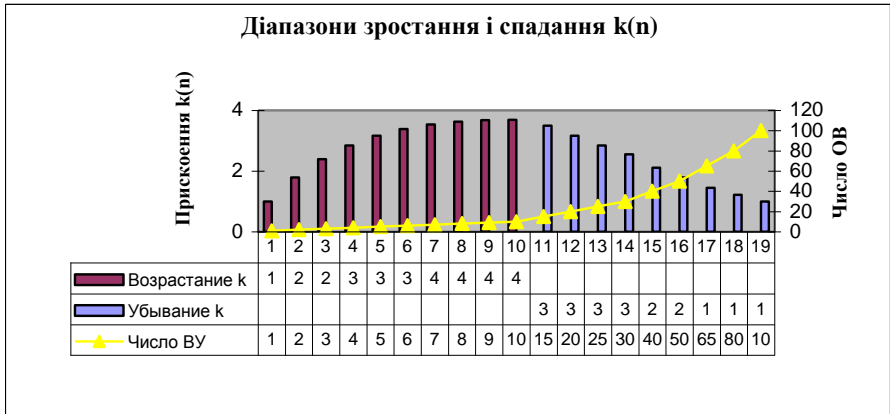


Рис. 1. Діаграма коефіцієнтів прискорень реалізації паралельного алгоритму

Розподіл обчислень між усіма ОВ системи призводить до прискорення $k = 1$.

Максимальне прискорення становить 3.7% від пікової продуктивності. На даній системі більшого прискорення отримати в принципі неможливо. В даному випадку механізм передачі повідомлень не дозволяє реалізувати потенційні можливості алгоритму, який допускає більш глибоке розпаралелювання. З діаграми видно що $k(100) = 1$. Такий варіант діаграми прискорень обумовлений значенням параметра τ , яке є в деякому роді прикордонним. Його збільшення призведе до отримання прискорень менших 1.

Приклад 2. Вибираємо систему і алгоритми А1, А2, А3, А4 з різними обсягами обчислень, але з однаковими обсягами передачі даних. Початкові дані зведені в табл. 1.

Таблиця 1

Алгоритми	А1	А2	А3	А4
$T_{\text{посл}}$	10^6	10^7	10^8	10^9

Проведемо розрахунки визначених параметрів і побудуємо графіки коефіцієнтів прискорень реалізації для кожного з алгоритмів (рис. 2).

Вихідні параметри загальні для всіх алгоритмів.

Параметри системи: $\alpha = 5, \tau = 50, N_p = 10^4$.

Приймаємо, $W = N_p = 10^4$. $f_{ГД} = \frac{1}{\alpha} = 0.2$

Алгоритми відрізняються тільки часом (див. табл. 1) послідовної реалізації $T_{посл}$. При однакових обсягах переданих даних найбільша частота генерації даних і найменше прискорення буде у алгоритму А1 (рис. 2). Зростання частоти генерації даних веде до падіння прискорення алгоритму. При розпаралелюванні алгоритмів необхідно враховувати значення $n_{онт}$, ігнорування якого призводить до великих втрат продуктивності, супроводжуване великим і невиправданим енергоспоживанням.

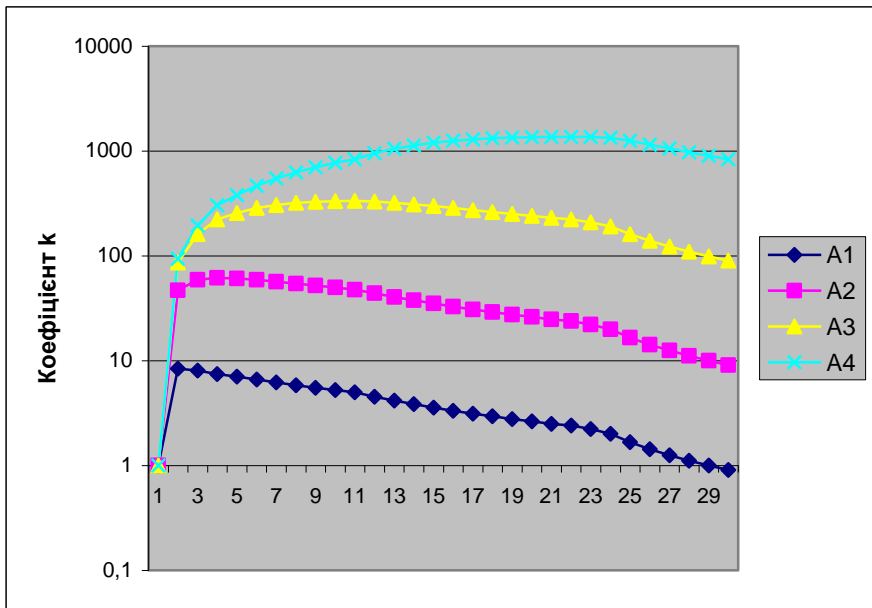


Рис. 2. Вплив частоти генерації даних на прискорення

Вплив частоти процесорів і частоти передачі даних на продуктивність паралельних алгоритмів. В якості вихідних параметрів візьмо паралельний алгоритм і багатопроцесорну систему та розглянемо, як впливає на продуктивність пари алгоритм – система підвищення частоти процесорів при незмінній частоті механізму передачі даних.

Приклад 3. Задаємо параметри вихідної пари алгоритм – система:
параметри системи: $\alpha = 5, \tau = 50, N_p = 10^4, f_{ГД} = 0,2$. $W = N_p, f = 1$;

загальні параметри алгоритму і системи:

$$T_{посл} = 10^7, \beta = 10^3, k_{ср} = 10^2, f_{ГД} = 10^{-3}.$$

f – тактова частота процесорів.

Підвищення частоти процесорів в вихідній моделі призводить до деякого зростання прискорення реалізації алгоритму (рис. 3). При збільшенні частоти на порядок ($f = 10$) приріст прискорення в точці максимуму складе 21%. На частоті 100, як випливає з графіка, алгоритм може бути реалізований тільки в однопроцесорному варіанті.

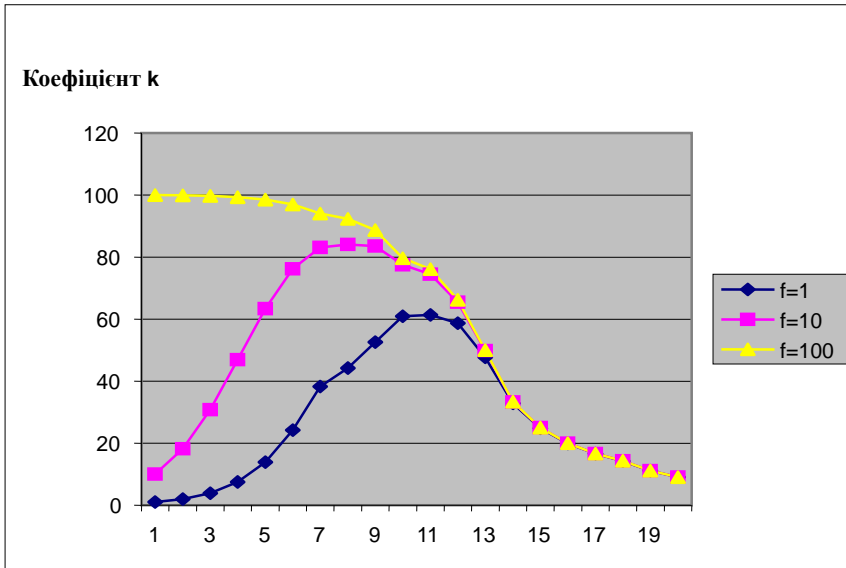


Рис. 3. Графіки прискорень при різних частотах процесорів

Розрахункові дані представлені в табл. 2. З графіків видно, що підвищення частоти процесорів не дає можливість подолати граничне прискорення 100. Для даного алгоритму досить розподілити програму між 300 процесорами.

Таблиця 2

f	1	10	100
k_{\max}	61	84	100
$n_{\text{опт}}$	316	84	1

В даному прикладі використано позамежне зростання частоти процесорів. Таке зростання частот для сучасних процесорів неможливе навіть у віддаленій перспективі, а при використанні технологій передачі повідомлень навіть, якщо він буде отриманий, то не призведе до відчутного зростання прискорень реалізації алгоритмів.

Отже, для паралельних обчислювальних систем підвищення зростання реальної продуктивності не пов'язане з ростом пікової продуктивності цих систем. Граничне значення прискорення 100 досягається при частоті $f = 100$. Воно досягається на одному процесорі. Підвищення частоти процесорів в системі має негативний вплив, тому що розширює безліч паралельних алгоритмів, що реалізуються тільки в послідовному варіанті.

Висновки

Дослідження і аналіз, які були проведені, дозволяють зробити висновок про те, що використання технології передачі повідомлень в загальному випадку суперечить концепції масовості паралелізму, так як не дозволяє: реалізувати закладені в паралельному алгоритмі потенційні можливості щодо прискорення обчислень; максимально використовувати обчислювальну потужність самої системи.

1. *Гергель В.П.* Высокопроизводительные вычисления для многоядерных многопроцессорных систем: Учебное пособие – Нижний Новгород: Издательство ННГУ им. Н.И. Лобачевского, 2010. – 421 с.
2. *Herbert H. J. Hum et al.* Forward state for use in cache coherency in a multiprocessor system, US Patent 6922756, 2005.
3. *Воеводин Вл.В., Жуматий С.А.* Вычислительное дело и кластерные системы. – М.: Изд-во МГУ, 2007. – 161 с.
4. *Волков Д., Фролов А.* Оценка быстродействия нерегулярного доступа к памяти / Открытые системы. СУБД №01, 2008.

<http://doi.org/10.5281/zenodo.3860770>

Поступила 30.09.2019р.

UDC 004.514.6+004.272.34+017.11

T. Neroda, Lviv

R. Ivaskiv, Lviv

TARGET CATALOG GENERATION FOR DIGITIZED HOLDINGS THE SCIENTIFIC AND TECHNICAL ACADEMIC LIBRARY

Abstract. The newest aspects in support of advanced and modern information and communication infrastructure and modernization of library services are analyzed. It is proved that the expansion of the functionality of the academic scientific and technical library can only happen by integrating the enterprise base of educational documentation into the analytical apparatus of the computerized library system. On basis the research of educational space information flows the data structures that