УДК 519.7, 519.8

ОПТИМІЗАЦІЯ РОЗПОДІЛУ ЗАВДАНЬ МІЖ КОМП'ЮТЕРАМИ *GRID*-СИСТЕМИ

Г. Г. Цегелик, Р. П. Краснюк

Львівський національний університет імені Івана Франка

E-mail: krasniuk@i.ua

Розглянуто задачу оптимального розподілу завдань однакової складності між комп'ютерами *GRID*-системи. Розроблено ефективний обчислювальний алгоритм, який використовує методи динамічного програмування та продемонстровано його використання на тестовому прикладі розподілу восьми завдань між трьома комп'ютерами *GRID*-системи.

Ключові слова: *оптимізація, GRID-система, методи динамічного програмування, обчислювальні алгоритми.*

OPTIMISATION DIVISION OF TASKS BETWEEN *GRID*-SYSTEM COMPUTERS

G. G. Tsehelyk, R. P. Krasniuk

Ivan Franko National University of Lviv

At present the formulation and study of the mathematical models of optimal allocation resources in *GRID*-systems have bee not are not sufficiently developed, therefore the creation of new architectural solutions raises the question of the efficient software development for such management information systems. But creation of the software is impossible without the formation of appropriate mathematical models and effective numerical algorithms. In managing the *GRID*-system, the goal of calculations that ensure the optimum mode solution of two classes of problems: problems which can perform parallel computing process or flow problems for which parallel operation is not possible is formulated. Therefore, the construction and study of the optimal allocation of tasks between computers of *GRID*-system is an urgent task. For the existing optimization model the complexity of the problem of computation for optimization model is equal to the distribution of tasks between computers on the network. The developed efficient computational algorithm which uses dynamic programming methods minimizes the total time of solving flow problems. To demonstrate the effectiveness of the calculation algorithm test case distribution of 8 tasks between 3 computers *GRID*-system is shown. As the circuit test shows the problem solution proposed by the algorithm is simple to be implemented in object-oriented programming language.

Keywords: *optimization, GRID-system, methods of dynamic programming, computational algorithms.*

У розподілених динамічних віртуальних організаціях спільне координоване використання ресурсів, різних за своєю структурою, підтримується технологіями та інфраструктурами *GRID*. Це дозволяє створювати обчислювальні віртуальні системи, які здатні підтримувати спільно достатній рівень обслуговування, в тому числі з географічно розосереджених компонентів, що використовуються в організаціях із різними умовами та правилами роботи. Ресурси спільного доступу – це не тільки обмін файлами, але і прямий доступ до комп'ютерів та програмного забезпечення, доступ до даних, необхідних для спільного вирішення задач у науці, бізнесі та промисловості.

GRID-технології надають на комерційній основі різноманітні ресурси і засоби для виконання найширшого кола запитів від користувачів, які надходять через *Internet*. Обслуговування таких запитів забезпечують центри, які мають більші обрахункові потужності та універсальні пакети прикладних програм [1].

Питанням моделювання різних аспектів функціонування розподілених систем, в тому числі побудованих на *GRID*-технології, та проблемам системної інте-

грації присвячено роботи таких вітчизняних та зарубіжних учених: М. Я. Бартіш, В. В. Воєводіна, М. З. Згуровський, В. М. Коваль, О. Я. Матова, О. А. Павлова, О. В. Палагіна, С. Ю. Яковлєва, Г. Г. Цегелик, I. Foster, L. Fusco, D. Menasce, L. Hluchy, B. Javadi, R. Buyya та ін. [2−5].

Так, у фундаментальній праці [2] проаналізовано ефективні архітектури систем розподілених обчислень, зокрема *GRID*-систем; наведено моделі та технології їх функціонування. Моделюванню та оптимізації доступу до інформації файлів баз даних для однопроцесорних та багатопроцесорних систем присвячено працю [3]. У ній запропоновано низку нових та ефективних математичних моделей оптимізації розподілу завдань у системах розподілених обчислень.

Монографія [4] присвячена моделюванню бізнес-процесів за допомогою *GRID*-технологій. Досліджено галузь математичного моделювання бізнес-процесів та методів прийняття рішення з оптимізацією за заданим критерієм. Оглядова праця [5] містить вичерпну характеристику *GRID*-систем та технологій, методів та підходів до побудови систем розподілених обчислень.

На сьогодні у розробці системи моделювання GRID-інфраструктури виділилися такі напрямки:

- побудова та дослідження ефективних архітектурних рішень;
- формування та дослідження математичних моделей;
- розробка API для програмної реалізації GRID-інфраструктури.

На думку авторів, напрям формування та досліджень математичних моделей оптимального розподілу ресурсів у *GRID*-системах ще не достатньо розвинутий, оскільки з формуванням нових архітектурних рішень постає питання розробки ефективного програмного забезпечення управління таким класом інформаційних систем. Створення ж програмного забезпечення неможливе без формування відповідних математичних моделей та ефективних числових алгоритмів. Певний клас математичних моделей оптимізації обчислень у розподілених системах наведено в працях [2, 4], однак застосування методів динамічного програмування під час побудови оптимізаційних задач розподілу завдань між комп'ютерами *GRID*-системи залишилося поза увагою дослідників.

З обчислювальної точки зору *GRID*-системи за відповідної організації обчислювального процесу дають змогу за прийнятний час загалом розв'язати два класи обчислювальних завдань: задачі, в яких можливе паралельне виконання обчислювального процесу та потоку задач, для яких розпаралелювання неможливе. Можливий варіант одночасного обслуговування *GRID*-системою двох визначених вище типів завдань. Тому під час управління *GRID*-системою постає завдання організації обчислень, що забезпечує їх оптимальний режим. Як наслідок, побудова та дослідження оптимізаційних моделей, пов'язаних з функціонуванням комп'ютерних систем, є актуальною та важливою проблемою, яку розглянемо нижче.

Відмінність роботи від результатів інших авторів, що досліджували цю проблематику, полягає у формуванні ефективного обчислюваного алгоритму для оптимізації розподілу завдань між комп'ютерами *GRID*-системи. Запропонований алгоритм ґрунтується на використанні методів динамічного програмування [6]. Вибір методу побудови розв'язку поставленої задачі визначається характером математичної моделі.

Математична постановка проблеми. Нехай необхідно розподілити m завдань однакової складності серед n комп'ютерів $K_1, K_2, \ldots, K_n$ *GRID*-системи (комп'ютери можуть мати не однакову потужність). Відомо час розв'язування завдань на кожному комп'ютері. Нехай $t_i(x_j)$, $i = 1, 2, \ldots, n$ − час розв'язування $x_j = j$, $j = 1, 2, \ldots, m$ завдань на i-му комп'ютері K_i. Задача полягає в такому розподілі завдань серед комп'ютерів, щоб сумарний час розв'язування завдань був мінімальним.

Якщо позначити через x_i, $i = 1, 2, \ldots, n$ кількість завдань, що планується розв'язувати на i-му комп'ютері, то математична модель матиме вигляд [4]:

$$T = \sum_{i=1}^{n} t_i(x_i) \to \min \tag{1}$$

за умов

$$\sum_{i=1}^{n} x_i = m, \quad x_i \in \{1, 2, \ldots, m\}, \, i = 1, 2, \ldots, n. \tag{2}$$

Для розв'язування поставленої задачі (1)–(2) використаємо метод динамічного програмування [6].

Формування обчислювального алгоритму. Позначимо через $T_i(x_j)$, $i = 1, 2, \ldots, n$ – сумарний час розв'язування $x_j = j$, $j = 0, 1, \ldots, m$ завдань на перших i комп'ютерах $K_1, K_2, \ldots, K_i$, а через $T_i^*(x_j)$, $i = 1, 2, \ldots, n$ – сумарний час розв'язування завдань у разі оптимального розподілу $x_j = j$, $j = 0, 1, \ldots, m$ завдань серед перших i-их комп'ютерів $K_1, K_2, \ldots, K_i$. Процес розв'язування задачі розіб'ємо на n кроків.

На першому кроці визначаємо мінімальний час у разі розв'язування $x_j = j$, $j = 0, 1, 2, \ldots, m$ завдань на першому комп'ютері K_1. На другому кроці – мінімальний час у випадку розв'язування $x_j = j$, $j = 0, 1, 2, \ldots, m$ завдань на перших двох комп'ютерах K_1, K_2 і т. д. Нарешті, на n-му кроці визначаємо мінімальний час у разі розв'язування m завдань на n комп'ютерах $K_1, K_2, \ldots, K_n$.

На першому кроці покладемо

$$T_i(x_j) = t_i(x_j), \quad T_i^*(x_j) = t_i(x_j), \quad j = 0, 1, \ldots, m.$$

На другому кроці:

$$T_2(x_j) = \begin{cases} t_2(0) + T_1^*(x_j - 0), \\ t_2(1) + T_1^*(x_j - 1), \\ \ldots \\ t_2(x_j) + T_1^*(0) \end{cases}$$

і $T_2^*(x_j) = \min\limits_{0 \le k \le j} \{t_2(k) + T_1^*(x_j - k)\}$ для $j = 0, 1, \ldots, m$.

Взагалі, на s-му кроці ($s = 3, 4, \ldots, n - 1$)

$$T_s(x_j) = \begin{cases} t_s(0) + T_{s-1}^*(x_j - 0), \\ t_s(1) + T_{s-1}^*(x_j - 1), \\ \ldots \\ t_s(x_j) + T_{s-1}^*(0) \end{cases}$$

і $T_s^*(x_j) = \min\limits_{0 \le k \le j} \{t_s(k) + T_{s-1}^*(x_j - k)\}$ для $j = 0, 1, \ldots, m$.

На останньому n-му кроці досить обчислити $T_n(m)$ і $T_n^*(m)$, де

$$T_n(m) = \begin{cases} t_n(0) + T^*_{n-1}(m), \\ t_n(1) + T^*_{n-1}(m-1), \\ \dots \\ t_n(m) + T^*_{n-1}(0) \end{cases}$$

і $T^*_n(m) = \min\limits_{0 \le k \le m} \{t_n(k) + T^*_{n-1}(m-k)\}$.

Оптимальний розподіл m задач серед n комп'ютерів *GRID*-системи визначаємо так.

Нехай $T^*_n(m)$ досягає мінімуму для $k = l_1$. Тоді l_1 завдань треба виконувати на останньому n-му комп'ютері K_n. Далі необхідно розподілити $m - l_1$ завдань серед перших $n - 1$ комп'ютерів $K_1, K_2, \dots, K_{n-1}$. Припустимо, що $T^*_{n-1}(m - l_1)$ досягає мінімуму для $k = l_2$. Це означає, що l_2 завдань треба виконувати на $(n - 1)$-му комп'ютері K_{n-1}. Якщо $T^*_{n-2}(m - (l_1 + l_2))$ досягає мінімуму для $k = l_3$, то l_3 завдань необхідно розв'язувати на $(n - 2)$-му комп'ютері K_{n-2}. І т. д. Нехай $T^*_2(m - (l_1 + l_2 + \dots + l_{n-2}))$ досягає мінімуму для $k = l_{n-1}$. Тоді l_{n-1} завдань треба виконувати на другому комп'ютері K_2. Нарешті, $l_n = m - (l_1 + l_2 + \dots + l_{n-1})$ завдань треба виконувати на першому комп'ютері K_1.

Мінімальний сумарний час виконання всіх завдань становитиме $T^*_n(m)$ одиниць.

Приклад побудови розв'язку задачі. Для демонстрації алгоритму розглянемо тестовий приклад. Необхідно розподілити вісім завдань однакової складності серед трьох комп'ютерів *GRID*-системи. Час виконання завдань кожним комп'ютером залежно від їх кількості подано у табл. 1.

Таблиця 1. Час виконання завдань комп'ютерами *GRID*-системи

$t_i(x_j)$	x_j								
	0	1	2	3	4	5	6	7	8
$t_1(x_j)$	0	5	7	9	11	14	17	21	25
$t_2(x_j)$	0	3	6	9	14	17	21	25	29
$t_3(x_j)$	0	4	5	6	8	11	15	19	24

На першому кроці алгоритму покладемо

$$T_1(j) = T^*_1(j) = t_1(j),\ j = 0,1,\dots, 8.$$

На другому кроці визначаємо:

$$T_2(0) = T^*_2(0) = t_2(0) + T^*_1(0);$$

$$T_2(1) = \begin{cases} t_2(1) + T^*_1(0), \\ t_2(0) + T^*_1(1), \end{cases} \qquad T^*_2(1) = \min_{0 \le k \le 1} \{t_2(k) + T^*_1(1-k)\};$$

$$T_2(2) = \begin{cases} t_2(2) + T^*_1(0), \\ t_2(1) + T^*_1(1), \\ t_2(0) + T^*_1(2), \end{cases} \qquad T^*_2(2) = \min_{0 \le k \le 2} \{t_2(k) + T^*_1(2-k)\};$$

$$T_2(3)=\begin{cases} t_2(3)+T_1^*(0), \\ t_2(2)+T_1^*(1), \\ t_2(1)+T_1^*(2), \\ t_2(0)+T_1^*(3), \end{cases} \qquad T_2^*(3)=\min_{0\le k\le 3}\left\{t_2(k)+T_1^*(3-k)\right\};$$

$$T_2(4)=\begin{cases} t_2(4)+T_1^*(0), \\ t_2(3)+T_1^*(1), \\ t_2(2)+T_1^*(2), \\ t_2(1)+T_1^*(3), \\ t_2(0)+T_1^*(4), \end{cases} \qquad T_2^*(4)=\min_{0\le k\le 4}\left\{t_2(k)+T_1^*(4-k)\right\};$$

$$T_2(5)=\begin{cases} t_2(5)+T_1^*(0), \\ t_2(4)+T_1^*(1), \\ t_2(3)+T_1^*(2), \\ t_2(2)+T_1^*(3), \\ t_2(1)+T_1^*(4), \\ t_2(0)+T_1^*(5), \end{cases} \qquad T_2^*(5)=\min_{0\le k\le 5}\left\{t_2(k)+T_1^*(5-k)\right\};$$

$$T_2(6)=\begin{cases} t_2(6)+T_1^*(0), \\ t_2(5)+T_1^*(1), \\ t_2(4)+T_1^*(2), \\ t_2(3)+T_1^*(3), \\ t_2(2)+T_1^*(4), \\ t_2(1)+T_1^*(5), \\ t_2(0)+T_1^*(6), \end{cases} \qquad T_2^*(6)=\min_{0\le k\le 6}\left\{t_2(k)+T_1^*(6-k)\right\};$$

$$T_2(7)=\begin{cases} t_2(7)+T_1^*(0), \\ t_2(6)+T_1^*(1), \\ t_2(5)+T_1^*(2), \\ t_2(4)+T_1^*(3), \\ t_2(3)+T_1^*(4), \\ t_2(2)+T_1^*(5), \\ t_2(1)+T_1^*(6), \\ t_2(0)+T_1^*(7), \end{cases} \qquad T_2^*(7)=\min_{0\le k\le 7}\left\{t_2(k)+T_1^*(7-k)\right\};$$

$$T_2(8)=\begin{cases} t_2(8)+T_1^*(0), \\ t_2(7)+T_1^*(1), \\ t_2(6)+T_1^*(2), \\ t_2(5)+T_1^*(3), \\ t_2(4)+T_1^*(4), \\ t_2(3)+T_1^*(5), \\ t_2(2)+T_1^*(6), \\ t_2(1)+T_1^*(7), \\ t_2(0)+T_1^*(8), \end{cases} \qquad T_2^*(8)=\min_{0\le k\le 8}\left\{t_2(k)+T_1^*(8-k)\right\}.$$

Результати обчислень наведено у табл. 2.

Таблиця 2. Результати обчислень за другим кроком алгоритму

x_j	k	$t_2(k)$	$T_1^*(x_j-k)$	$T_2(x_j)$
1	2	3	4	5
0	0	0	0	0^*
1	1	3	0	3^*
	0	0	5	5
2	2	6	0	6^*
	1	3	5	8
	0	0	7	7
3	3	9	0	9^*
	2	6	5	11
	1	3	7	10
	0	0	9	9^*
4	4	14	0	14
	3	9	5	14
	2	6	7	13
	1	3	9	12
	0	0	11	11^*
5	5	17	0	17
	4	14	5	19
	3	9	7	16
	2	6	9	15
	1	3	11	14^*
	0	0	14	14^*
6	6	21	0	21
	5	17	5	22
	4	14	7	21
	3	9	9	18
	2	6	11	17^*
	1	3	14	17^*
	0	0	17	17^*

1	2	3	4	5
7	7	25	0	25
	6	21	5	26
	5	17	7	24
	4	14	9	23
	3	9	11	20^*
	2	6	14	20^*
	1	3	17	20^*
	0	0	21	21
8	8	29	0	29
	7	25	5	30
	6	21	7	28
	5	17	9	26
	4	14	11	25
	3	9	14	23^*
	2	6	17	23^*
	1	3	21	24
	0	0	25	25

На третьому кроці –

$$T_3(8)=\begin{cases} t_3(8)+T_2^*(0), \\ t_3(7)+T_2^*(1), \\ t_3(6)+T_2^*(2), \\ t_3(5)+T_2^*(3), \\ t_3(4)+T_2^*(4), \\ t_3(3)+T_2^*(5), \\ t_3(2)+T_2^*(6), \\ t_3(1)+T_2^*(7), \\ t_3(0)+T_2^*(8), \end{cases} \qquad T_3^*(8)=\min_{0\le k\le 8}\left\{t_3(k)+T_2^*(8-k)\right\}.$$

Результати обчислень наведено у табл. 3.

Таблиця 3. Результати обчислень за третім кроком алгоритму

x_j	k	$t_3(k)$	$T_2^*(x_j-k)$	$T_3(x_j)$
8	8	24	0	24
	7	19	3	22
	6	15	6	21
	5	11	9	20
	4	8	11	19^*
	3	6	14	20
	2	5	17	22
	1	4	20	24
	0	0	23	23

Із табл. 3 бачимо, що $T_3^*(8)=19$ і досягається для k = 4. Це означає, що чотири завдання треба виконувати на третьому комп'ютері. Залишилось розподілити чотири завдання серед перших двох комп'ютерів. Із табл. 2 бачимо, що $T_2^*(4)=11$ і

досягається для $k = 0$. Це означає, що на другому комп'ютері не потрібно виконувати завдань, тобто чотири завдання необхідно виконувати на першому комп'ютері. Загальний час виконання всіх завдань становить 19 одиниць часу.

ВИСНОВКИ

Здійснено математичну постановку задачі оптимального розподілу завдань між комп'ютерами *GRID*-системи з врахуванням їхніх можливостей та запропоновано ефективний обчислювальний алгоритм розв'язання цих завдань, який ґрунтується на методі динамічного програмування. Зазначимо, що вибір методу побудови розв'язку поставленої задачі визначався характером математичної моделі.

Для демонстрації ефективності запропонованого алгоритму розв'язано тестову задачу розподілу восьми завдань серед трьох комп'ютерів *GRID*-системи. Як показує схема розв'язку тестової задачі, алгоритм є простим для реалізації на об'єктно-орієнтованій мові програмування.

Предметом подальших досліджень цього класу задач є інтеграція запропонованого обчислювального алгоритму в програмне забезпечення, яке здійснює керування ресурсами *GRID*-системи.

1. *Менаске Д., Алмейда В.* Производительность Web-служб. Анализ, оценка и планирование. – СПб.: ДиаСофт, 2003. – 430 с.
2. *Куссуль Н. Н., Шелестов А. Ю.* Grid-системы для задач иследования Земли. Архитектура, модели и технологии. – К.: Наук. думка, 2008. – 452 с.
3. *Цегелик Г. Г.* Моделювання та оптимізація доступу до інформації файлів баз даних для однопроцесорних і багатопроцесорних систем. – Львів: ЛНУ ім. Івана Франка, 2010. – 192 с.
4. *Інформаційні* технології та моделювання бізнес-процесів: навч. пос. / О. М. Томашевський, Г. Г. Цегелик, М. Б. Вітер, В. І. Дубук. – К.: Вид-во "Центр учбової літератури", 2012. – 296 с.
5. *Foster I.* What is the Grid? A Three Point Checklis // Department of Computer Science, Univesity of Chicago, Chicago, IL 60637, July 20, 2002. – P. 134–182.
6. *Цегелик Г. Г.* Математичне програмування: навч. пос. – Львів: ЛНУ ім. Івана Франка, 2011. – 168 с.

Одержано 13.03.2015