УДК 519.7, 519.8

ОПТИМІЗАЦІЯ РОЗМІЩЕННЯ РЕПЛІКАЦІЙНИХ БАЗ ДАНИХ У РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Г. Г. Цегелик, Р. П. Краснюк

Львівський національний університет імені Івана Франка

E-mail: kafmmsep@lnu.edu.ua, krasniuk@i.ua

Сформульовано та досліджено задачу оптимального розміщення реплікаційних баз даних у вузлах розподіленої інформаційної системи за такими критеріями: мінімізація витрат на обслуговування; обмеження на ресурси пам'яті; мінімізація часу синхронізації; мінімізація середнього часу, необхідного для пошуку інформації. Побудовано точні розв'язки відповідних оптимізаційних задач із використанням методів динамічного програмування, зокрема, наведено рекурентні рівняння Беллмана та запропоновано і досліджено алгоритми отримання наближеного розв'язку із використанням стратегій жадібного вибору.

Ключові слова: *оптимізація, реплікація баз даних, розподілені інформаційні системи, динамічне програмування, рівняння Беллмана, жадібний вибір.*

THE OPTIMIZATION OF DATABASES REPLICATION IN DISTRIBUTED INFORMATION SYSTEMS

G. G. Tsegelyk, R. P. Krasniuk

Ivan Franko National University of Lviv

New mathematical models of optimal distribution of databases replication in nodes of distributed information systems are formulated by the criteria: minimization of maintenance costs; restricted memory resources; minimizing synchronization time; minimizing the average time needed to search information. Precise solutions of the problems with the use of dynamic programming methods are constructed, Bellman recursive equations are obtained. The general scheme of the computational algorithm using the "greedy" choice procedure is presented and an algorithm for improving the obtained result is proposed. The strategies of greedy choice were investigated, the choice of criteria in the strategy of greedy choice is substantiated. The proposals have been formed regarding the formation of a balance between the accuracy and computational complexity of the algorithm through the introduction of a restricted search strategy. The computational complexity of the algorithm is estimated and its correctness is substantiated.

Keywords: *optimization, databases replication, distributed information systems, dynamic programming, Bellman equation, Greedy algorithm.*

Сучасні інформаційні системи (ІС) висувають доволі високі вимоги щодо швидкості обробки інформації за умови одночасного звертання до серверів великої кількості клієнтів. Крім того, розвиток інформаційних систем має забезпечувати вимогу масштабності без зниження швидкісних характеристик системи.

Одним зі способів задоволення цієї потреби є створення розподіленої бази даних (БД), яка підтримує механізм асинхронної реплікації. В цьому випадку замість однієї централізованої бази даних, з якою повинні працювати усі клієнти інформаційної системи, створюється декілька серверів БД у вузлах розподіленої мережі.

Для використання розподілених баз даних не для всіх задач необхідно забезпечити ідентичність БД на різних вузлах у довільний час. Достатньою є підтримка ідентичності даних лише у певні критичні моменти часу. Тому можна накопичувати зміни даних у вигляді трансакцій в одному з вузлів та періодично копіювати ці зміни в інші вузли мережі. Цей процес має назву тиражування даних (*Data Replication – DR*) – асинхронне перенесення змін об'єктів вихідної бази даних (*source database*) у БД, які належать різним вузлам розподіленої системи. Функції *DR* виконує спеціальний модуль СУБД – сервер тиражування даних – реплікатор (*replicator*). Його завдання – підтримання ідентичності даних у приймаючих базах даних (*target databases*) до даних у вихідній БД.

Принциповою характеристикою тиражування даних є відмова від фізичного їх розподілу. Значення *DR* полягає в тому, що довільна БД (як для СУБД, так і для її користувачів) завжди є локальною (розміщується на комп'ютері користувача для монопольного доступу чи у локальній мережі компанії для забезпечення доступу до БД її співробітників); дані розміщуються локально у тому вузлі мережі, де вони обробляються; усі трансакції у системі завершуються локально.

Технологія реплікації необхідна для ІС, які використовують у галузях із високими вимогами щодо гарантій отримання, своєчасності та цілісності даних, які передаються. Це є системи міжнародних безготівкових банківських платежів, обробки даних розподілених технологічних процесів реального часу, білінгові системи, а також спеціалізовані корпоративні системи.

Однак використання технології реплікації даних ставить низку задач оптимізації, а саме: оптимального розподілу реплікаційних баз у розподіленій інформаційній системі за умов мінімізації витрат на обслуговування, обмежень на обчислювальні ресурси вузлів, мінімізації часу синхронізації даних або мінімізації середнього часу, необхідного для пошуку інформації за відсутності синхронізації реплікаційних БД. Практика застосування ІС розташовує масив даних у декількох БД за певними логічними чи технологічними ознаками. Наприклад, останнім часом популярним є поєднання реляційної БД (наприклад, *Microsoft SQL Server*), яка виступає сховищем бізнес-даних, та *NoSQL* бази даних (наприклад, *MongoDB*) як сховища службової інформації (наприклад, лог файлів) [1]. Тому можливе розміщення в одному вузлі розподіленої інформаційної системи декількох БД. Як наслідок, дослідження оптимального розподілу реплікаційних баз даних у інформаційних системах є актуальним і тому формулювання відповідних математичних моделей та розв'язок задач оптимізації є предметом досліджень цієї праці.

Зважаючи на важливість проблематики оптимального розподілу БД в ІС, систематизували [2] математичні підходи та моделювання систем розподілених баз даних. Математичні моделі багатокритеріального синтезу фізичних структур розподілених БД розглядали у праці [3]. Сформульовано та розроблено математичні моделі за критеріями мінімуму витрат, часу доступу до даних та мережевого трафіку. Концепції побудови та вибору розподілених БД інформаційно-пошукових систем присвячено працю [4]. Запропонована концепція заснована на аналізі якісних і кількісних ознак для виявлення пріоритетної технології та набору її домінантних ознак, що будуть залучені в автоматизованих інструментальних засобах проектування. Алгоритмічне забезпечення розподілених БД розглядали у праці [5]. Розглянуто схему проектування розподіленої бази даних та описано її етапи. Сформульована задача проектування розподіленої БД та оцінена її складність. Обґрунтовано вибір алгоритмів для вирішення поставленої задачі й запропоновано підхід, що дозволяє врахувати взаємозалежність етапів проектування.

Відмінність цієї роботи від результатів інших авторів полягає у розробленні нових математичних моделей оптимізації розподілу реплікаційних БД, побудові як точних, так і наближених розв'язків із формуванням ефективних числових алгоритмів. Точні розв'язки задач побудовано з використанням методів динамічного програмування – отримано рекурентні рівняння Беллмана, що мають як самостійне значення, так і можуть бути застосовані для аналізу точності розрахунків відповідних задач оптимізації з використанням числових методів. Крім того, на прикладі задачі оптимізації розміщення БД у розподіленій ІС наведено загальну схему обчислювального алгоритму, що використовує процедуру жадібного вибору, та запропоновано алгоритм поліпшення отриманого результату, який знижує середню похибку наближеного розв’язку до рівня, прийнятного для практичного застосування.

Математичне формулювання задачі оптимального розподілу реплікаційних баз даних у вузлах мережі за умови мінімізації витрат на обслуговування. Нехай m баз даних необхідно розподілити у n вузлах мережі ($m < n$). Витрати на обслуговування i-ї бази у j-му вузлі становлять c_{ij} одиниць. Умовою оптимальності у цьому випадку є мінімізація загальних витрат F на обслуговування баз даних інформаційної системи. Тоді математична модель задачі набуває вигляду

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min , \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = 1 , \quad x_{ij} = \{0, 1\} , \quad i = 1, 2, ..., m; \quad j = 1, 2, ..., n . \tag{2}$$

За математичного формулювання цієї оптимізаційної задачі накладається обмеження щодо розміщення тільки однієї БД у вузлі (2). Якщо ж позбутися цієї умови, то природно виникає обмеження щодо обсягів ресурсів пам’яті у вузлі розподіленої ІС.

Математичне формулювання задачі оптимального розподілу реплікаційних баз даних у вузлах мережі за умови мінімізації витрат на обслуговування та обмеження ресурсів пам’яті. Нехай, як і раніше, необхідно розподілити m баз даних, які можуть займати відповідно a_i одиниць пам’яті у n вузлах мережі, де загальні об’єми ресурсів пам’яті становлять b_j одиниць відповідно. Витрати на обслуговування i-ї бази у j-му вузлі – c_{ij} одиниць. Умовою оптимальності, як і у попередньому випадку, є мінімізація загальних витрат F на обслуговування баз даних інформаційної системи. Тоді математичну модель задачі можна сформулювати так:

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min , \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} \le m , \quad \sum_{i=1}^{m} a_i x_{ij} \le b_j , \quad x_{ij} = \{0, 1\} , \quad i = 1, 2, ..., m; \quad j = 1, 2, ..., n . \tag{4}$$

Математичне формулювання задачі оптимального розміщення реплікаційних баз у розподілений інформаційній системі за умов мінімізації часу синхронізації. Аналогічно до попередніх двох випадків, маємо m баз даних розподілити у n вузлах мережі ($m < n$). Пропускна здатність каналу зв’язку за одиницю часу від вузла i до вузла j становить ρ_{ij}. Середній обсяг даних, який необхідно передати від вузла i до вузла j для синхронізації реплікаційних баз – α_{ij}. Якщо ввести у розгляд коефіцієнти $x_{ij} = \{0,1\}$ використання каналу зв’язку у задачі оптимізації, то математична модель набуває вигляду

$$F = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\alpha_{ij}}{\rho_{ij}} x_{ij} \to \min , \tag{5}$$

$$x_{ij} = \{0,1\}, \quad \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} \le 2m, \quad \sum_{i=1}^{n} x_{ij} \le m-1. \tag{6}$$

Умовою оптимальності в цьому випадку є мінімізація загального часу F синхронізації даних між реплікаційними БД. Очевидно, якщо для певного вузла i виконується умова

$$\sum_{j=1}^{n} x_{ij} = 0,$$

то в цьому вузлі база даних відсутня. За математичного формулювання цієї оптимізаційної задачі накладається обмеження (6) щодо розміщення тільки однієї БД у вузлі мережі.

Математичне формулювання задачі оптимального розподілу реплікаційних баз даних у вузлах мережі за умови мінімізації часу, необхідного для пошуку інформації. Знову ж таки, m баз даних необхідно розподілити у n вузлах мережі ($m < n$), λ_{ij} − інтенсивність запитів із вузла i до бази j, β_{ij} − кількість даних, які мають бути переміщені відповідно до запиту між вузлом i обчислювальної мережі та базою j, ρ_{ij} − пропускна здатність каналу зв'язку за одиницю часу від вузла i до бази даних j. Якщо ввести коефіцієнти $x_{ij} = \{0,1\}$, які визначають, чи база даних j розміщена у вузлі i, то математична модель задачі набуває вигляду

$$F = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{\lambda_{ij}\beta_{ij}}{\rho_{ij}}\left(1 - x_{ij}\right)}{\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_{ij}} \to \min, \tag{7}$$

$$x_{ij} = \{0,1\}, \quad \sum_{i=1}^{n} x_{ij} \ge 1; \quad j = 1,\ldots,m, \quad \sum_{j=1}^{m}\left(1 - x_{ij}\right) = 1; \quad j = 1,\ldots,n. \tag{8}$$

Ефективним підходом до побудови аналітичних розв'язків сформульованих оптимізаційних задач (1)–(2), (3)–(4), (5)–(6) та (7)–(8) є використання методів динамічного програмування [6]. Не наводячи проміжні викладки, подаємо остаточні результати – рекурентні рівняння Беллмана. Так, для задачі (1)–(2):

$$F_1\left(\{x_{ij}\}\right) = g_1\left(\{x_{1j}\}\right), \quad F_k\left(\{x_{ij}\}\right) = \min\left\{g_k\left(x_{kj}\right) + F_{k-1}\left(\{x_{ij}\} \setminus \{x_{kj}\}\right)\right\},$$

$$g_k\left(x_{kj}\right) = \min_{j}\left\{c_{kj}\right\}, \quad j = 1, 2, \ldots, n; \quad k = 2, 3, \ldots, m;$$

для задачі (3)–(4):

$$F_1\left(\{x_{ij}\}\right) = g_1\left(\{x_{1j}\}\right),$$

$$F_k\left(\{x_{ij}\}\right) = \min\left\{g_k\left(x_{kj}\right) + F_{k-1}\left(\{x_{ij}\}\right)_{\{b_1, b_2, \ldots, b_j - a_k, \ldots, b_n\}}\right\},$$

$$g_k\left(x_{kj}\right) = \min_{j,\, a_k \le b_j}\left\{c_{kj}\right\}, \quad j = 1, 2, \ldots, n; \quad k = 2, 3, \ldots, m,$$

а для задач (5)–(6) та (7)–(8) отримали:

$$F_1\left(\left\{x_{ij}\right\}\right)=q_1\left(\left\{x_{1j}\right\}\right),\quad F_k\left(\left\{x_{ij}\right\}\right)=\min\left\{q_k\left(x_{kj}\right)+F_{k-1}\left(\left\{x_{ij}\right\}\setminus\left\{x_{kj}\right\}\right)\right\}, \tag{9}$$

$$q_k\left(x_{kj}\right)=\min_j\left\{\Lambda_{ij}\right\},\quad j = 1, 2,\ldots, n;\quad k = 2, 3,\ldots, m,$$

де Λ_{ij} у формулі (9) для задачі (5)–(6) рівне α_{ij}/ρ_{ij}, а для задачі (7)–(8) – $\left(\lambda_{ij}\beta_{ij}\right)/\rho_{ij}$.

Ще одним підходом до побудови розв'язків розглянутих задач є використання жадібних алгоритмів. Далі наведено загальну схему обчислювального алгоритму на прикладі задачі оптимізації розміщення БД у розподіленій ІС, що має також і самостійне значення.

Нехай n – кількість комп'ютерів різної потужності, які складають розподілену ІС; m – кількість БД, які необхідно розмістити у системі; x_i – кількість БД, які планують розмістити на i-му комп'ютері системи (шукані величини); $c_i\left(x_i\right)$ – витрати на зберігання та супроводження x_i баз даних на i-му комп'ютері системи. Тоді математична модель задачі матиме вигляд: необхідно мінімізувати загальні витрати на супроводження системи

$$F=\sum_{i=1}^{n}c_i\left(x_i\right)\to\min, \tag{10}$$

за умов

$$\sum_{i=1}^{n}x_i\le m,\quad x_i\ge 0,\quad i=\overline{1,n}. \tag{11}$$

За обмеження обчислювальних можливостей комп'ютерів системи $\left(0\le x_i\le l_i, i=\overline{1,n}\right)$ оптимізаційна задача (10)–(11) набуває вигляду: необхідно знайти мінімум цільової функції (10) за обмеження

$$\sum_{i=1}^{n}x_i\le m,\quad 0\le x_i\le l_i,\quad i=\overline{1,n}. \tag{12}$$

За алгоритмом, який використовує жадібний вибір, позначимо через $X\left(x_i^*\right)$ множину допустимих планів вихідної задачі, для яких $x_i=x_i^*$, $I=\left\{1,2,...,n\right\}$ – множина індексів змінних, I_0 – множина індексів змінних, яким процедурою жадібного вибору надано нові значення.

Алгоритм.

Крок 0. Покладемо $X_0=X$, $I_0=\varnothing$ і виберемо початковий допустимий план $x^0=\left(x_1^0,x_2^0,...,x_n^0\right)\in X_0$. Для кожної із n змінних знаходимо межі $x_i^L\le x_i\le x_i^R$ так, що $x^0=\left(x_1^0,x_2^0,...,x_i,...,x_n^0\right)\in X_0$ для всіх цілих значень x_i, $x_i^L\le x_i\le x_i^R$. Межі нерівностей можна задавати наближено.

Нехай сформовано множини I_k та $X_k\in X$.

Крок 1. Якщо $I_k=I$, то всі змінні отримали нові значення. Тобто побудовано допустимий вектор x, який приймаємо за наближений розв'язок. В іншому випадку переходимо до *кроку 2*.

Крок 2. Для $j\in I\setminus I_k$ знаходимо:

$$j_0 = \max_{j \in I \setminus I_k} \left\{ \min_{x_i^L \le x_i \le x_i^R} \sum_{l=1}^{n} c_l\left(x_l\right) \right\}, \quad \text{де} \quad x_k^{L,R} = \begin{cases} x_k^0, & k \in I \setminus I_k, k \ne j, \\ x_k^*, & k \in I_k. \end{cases}$$

Покладемо $$x_{j_0}^* = \min_{x_{i_0}^L \le x_{i_0} \le x_{i_0}^R} \left(c_1\left(x_1\right) + c_2\left(x_2\right) + \ldots + c_{i_0}\left(x_{i_0}\right) + \ldots + c_n\left(x_n\right) \right),$$

$X_{k+1} = X_k\left(x_{j_0}^*\right)$, $I_{k+1} = I_k \cup \left\{ j_0 \right\}$ та переходимо до *кроку 1*.

Використовуючи сформульований алгоритм, наведемо розв’язок задачі (10), (12). Він починається з початкового допустимого плану $x = \left(0, 0, \ldots, 0\right)$.

Крок 1. Обчислюємо *j*-ту координату вектора x: $x_j = l_j$, де $l_j = \max_i l_i$ та $c_j\left(l_j\right) \to \min$ за умови, що $\sum_{i=1}^{n} x_i \le m$.

Крок 2. Перетворюємо задачу так: із цільової функції (10) і лівої частини обмеження (12) виключаємо член, що містить зміну x_j, тобто отримуємо задачу:

$$c_1\left(x_1\right) + c_2\left(x_2\right) + \ldots + c_{j-1}\left(x_{j-1}\right) + c_{j+1}\left(x_{j+1}\right) + \ldots + c_n\left(x_n\right) \to \min, \tag{13}$$

$$x_1 + x_2 + \ldots + x_{j-1} + x_{j+1} + \ldots + x_n \le m.$$

У результаті перетворень кількість змінних у початковій задачі зменшується на одиницю. Повертаємося до *кроку 1*.

Кроки 1 і *2* повторюються доти, доки всі змінні вектора x не отримають нові значення або буде порушено умову (12).

Для зниження похибки наближеного розв’язку, який отримуємо за сформованим алгоритмом, пропонуємо його поліпшити. Зокрема, надалі припускатимемо, що змінні вектора *x* пронумеровані за порядком отримання значень у жадібному алгоритмі.

Алгоритм поліпшення розв’язку.

Як початковий план, вибираємо розв’язок, який отримано за жадібним алгоритмом. Покладаємо $K = \left\{1, 2, \ldots, n\right\}$, $k = 1$.

Крок 1. Вибираємо ще неопрацьовану змінну $x_k > 0$ з мінімальним індексом $k \in K$ та підбираємо значення $\delta_k > 0$, на яке необхідно зменшити величину x_k.

Крок 2. Знаходимо змінну x_q, $q > k$ та величину $\delta_q > 0$, на яку можна зменшити значення x_q, не порушуючи допустимості плану та зменшивши при цьому цільову функцію. Для цього δ_k та δ_q повинні задовольняти умову $c_q\left(\delta_q\right) < c_k\left(\delta_k\right)$.

На закінчення *кроку 2* незалежно від того знайдена чи ні змінна x_q, покладаємо $K = K \setminus \left\{k\right\}$, збільшуємо k на одиницю і повертаємося до *кроку 1*. Припиняємо тоді, коли усі змінні в початковому плані будуть переглянуті (K = Ø).

Як показали результати числових експериментів, середня похибка наближеного розв’язку знижується від 15% до 4% за застосування алгоритму поліпшення розв’язку.

Очевидно, що якість виконання алгоритму і, як наслідок, можливість отримання результату, близького до оптимального, суттєво залежить від використовуваних критеріїв у стратегії жадібного вибору. Далі розглянемо стратегію вибору критеріїв, яка ґрунтується на функції вартості p_l вибору елемента *l* (наприклад, для задачі (1)–(2) вартістю вибору елемента x_{ij} є $p_{ij} = \max\limits_{i,j}\{1/c_{ij}\}$). Оскільки в загальному випадку елемент описуємо сукупністю значень необхідних ресурсів, що впливають на його вибір, то пропонуємо обчислювати вартість вибору двома способами:

- обрання найдефіцитнішого ресурсу на момент вибору та розрахунок вартості за вибраною характеристикою;
- обчислення зваженої суми характеристик ресурсів згідно із дефіцитністю цих ресурсів.

Нехай елемент l описано вектором необхідних характеристик ресурсів $\{d_{lk}\}, k = \overline{1,r}$. Для першого з описаних способів вибору вартість p_l обраховуємо за формулою $p_l = \max\limits_k \dfrac{d_{lk}}{D_k}$, де D_k – загальна місткість ресурсів розподіленої системи за характеристикою k.

Для визначення зваженої вартості елемента для кожного ресурсу обчислюємо дефіцит: $s_{lk} = d_{lk} / D_k$. Що більшим він є, то більше необхідно ресурсів цього типу.

Тоді вартість p_l обчислюємо так: $p_l = \sum\limits_{k=1}^{r} s_{lk} d_{lk} / \sum\limits_{k=1}^{r} d_{lk}$ і вона дорівнює зваженій сумі необхідних ресурсів з врахуванням їх дефіциту.

Виділяють два варіанти критеріїв: компактний та збалансований вибір.

Компактний вибір використовують тоді, коли фізичні ресурси розподіленої обчислювальної системи не є критичні, тобто для випадків, коли відношення вартостей необхідних ресурсів системи до наявних ресурсів є менше заданого порогового значення. Тоді жадібні критерії дозволяють вибирати за стратегії *BF – best-fit* [7].

Збалансований вибір використовують тоді, коли певні ресурси системи є критичними, тобто коли відношення вартості необхідних ресурсів системи до вартості наявних ресурсів є більшим за деякий поріг, що забезпечує розподіл елементів серед більшої кількості ресурсних вузлів системи.

Іншою стратегією, на яку звернемо увагу, є обмежене перебирання. Дуже часто за жадібного вибору виникає така ситуація: нехай вузли обчислювальної системи можуть мати місткість пам’яті в одну умовну одиницю. Необхідно розподілити елемент з місткістю пам’яті 0,3 у.о. в одному з трьох вузлів системи:

вузол 1 – 0,3 + 0,5, *вузол 2* – 0,2 + 0,5 + 0,1, *вузол 3* – 0,4 + 0,4.

Очевидно, що сумарної вільної місткості пам’яті у системі є достатньо для розміщення необхідного елемента, але вставити цей елемент без фрагментації в один з вузлів неможливо.

Як наслідок, очевидним є перепризначення елементів у вузлах системи для звільнення необхідних ресурсів. Зручним варіантом є застосування стратегії обмеженого перебирання, яка вказує максимальну кількість вузлів, для яких можна перепризначати. Наприклад, за умови обмеження у два вузли (тобто тоді, коли призначені елементи можуть перепризначатися одночасно не більше ніж у двох вузлах системи) можемо отримати необхідний розподіл:

вузол 1 – 0,3 + 0,5 + 0,1, *вузол 2* – 0,2 + 0,5 + 0,3, *вузол 3* – 0,4 + 0,4.

Далі подаємо загальну стратегію (алгоритм) обмеженого перебирання:

Крок 1. Для заданого обмеження на кількість вузлів k сформувати усі підмножини $S_k \subseteq S$ із не більше, ніж k вузлів і перебирати варіанти відповідно до зменшення сумарного значення, яке обчислюємо за жадібним критерієм.

Крок 2. Якщо для деякої множини S_k^* сумарна залишкова місткість є достатньою для розміщення елемента, то виконати кроки 3–5, інакше перейти до кроку 6.

Крок 3. Вилучити елементи, які призначено на вузли з множини S_k^*.

Крок 4. Сформувати множину, яка складається з поточного елемента та усіх вилучених елементів, та призначити їх у вузли множини S_k^* за жадібним критерієм – за потребою з використанням повного перебирання призначень елементів зі сформованої множини.

Крок 5. За неуспішного перепризначення повернути початкове призначення елементів.

Крок 6. Якщо за перебирання усіх вказаних множин поточний елемент не може бути призначений, то необхідно збільшити обмеження кількості вузлів k на одиницю та повернутися до *кроку 1*.

Крок 7. Якщо значення k дорівнює кількості вузлів системи, то потрібно видати повідомлення про неможливість розміщення поточного елемента у вузлах обчислювальної системи.

Оцінимо складність жадібного алгоритму розміщення обчислювальних ресурсів у вузлах розподіленої інформаційної системи.

Нехай n – кількість обчислювальних вузлів; m – кількість обчислювальних ресурсів (наприклад, реплікаційних БД), що мають бути розміщені у вузлах системи; k – коефіцієнт перебирання для стратегії обмеженого перебирання. Тоді:

- сортування обчислювальних ресурсів, згідно з вибраним жадібним критерієм, має складність $O\left(m\log(m)\right)$;
- розміщення обчислювальних ресурсів у вузлі системи (якщо воно можливе) у найгіршому випадку виконується за $O(mn)$ операцій;
- за застосування стратегії обмеженого перебирання кількість поєднань, що перебираються, обчислюємо за формулою C_n^k. На кожному перебиранні відбувається спроба перепризначення, складність якого залежить від вибраного способу обмеженого перебирання:
  - за використання жадібного підходу у стратегії обмеженого перебирання складність перепризначень $O\left(m\left(\log(m)+n\right)\right)$;
  - за використання повного перебирання у стратегії обмеженого перебирання складність у найгіршому випадку досягає $O(m!n)$.

Як наслідок, складність жадібного алгоритму становить $O\left(\left(C_n^k+1\right)m\left(\log(m)+n\right)\right)$ за використання жадібного підходу у стратегії обмеженого перебирання та $O\left(m\left(\log(m)+n\right)\right)+O\left(C_n^k m!n\right)$ за використання підходу повного перебирання у стратегії обмеженого перебирання. Крім того, потрібно зауважити, що за умови $k=0$, складність жадібного алгоритму має квадратичну залежність від n, а зі збільшенням m складність алгоритму зростає як $O\left(m\log(m)\right)$.

Коректність жадібного алгоритму забезпечують виконанням обмежень у відповідних задачах оптимізації:

- на етапі призначення обчислювального ресурсу у відповідний вузол розподіленої системи мають бути виконані відповідні умови обмежень, інакше здійснюється виклик стратегії обмеженого перебирання;

- стратегія обмеженого перебирання елементів за успішного призначення перепризначає множини елементів разом із призначенням поточного елемента так, щоб усі наявні обмеження були виконані.

ВИСНОВКИ

Здійснено математичну постановку задач оптимального розміщення реплікаційних баз даних у розподілених ІС за критеріями: мінімізації витрат на обслуговування; обмежень на ресурси пам'яті; мінімізації часу синхронізації та часу, необхідного для пошуку інформації. Точний розв'язок задач побудовано із використанням методів динамічного програмування – отримано рекурентні рівняння Беллмана, які мають як самостійне значення, так і можуть бути використані для аналізу точності розрахунків відповідних задач оптимізації за допомогою числових методів.

На прикладі задачі оптимізації розміщення баз даних у розподіленій інформаційній системі наведено загальну схему обчислювального алгоритму, який використовує процедуру жадібного вибору, та запропоновано алгоритм поліпшення отриманого результату, що знижує середню похибку наближеного розв'язку до рівня, прийнятного для практичних застосувань.

Досліджено стратегію жадібного вибору, обґрунтовано вибір критеріїв у цій стратегії та сформовано пропозиції щодо формування балансу між точністю та обчислювальною складністю алгоритму шляхом впровадження стратегії обмеженого перебирання елементів, що зменшує кількість вузлів розподіленої ІС, яку використовують при перерозподілі ресурсів системи. Проаналізовано обчислювальну складність алгоритму пошуку наближеного розв'язку за стратегією жадібного вибору та обґрунтовано його коректність.

1. *Sadalage P. J., Fowler M.* NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. – NY: Addison-Wesley, 2012. – 680 p.
2. *Цегелик Г. Г.* Системы распределенных баз данных. – Львов: Свит, 1990. – 168 с.
3. *Бескоровайный В. В., Ульянова О. С.* Математические модели многокритериального синтеза физических структур распределенных баз данных // Восточно-Европейский журнал передовых технологий. – 2010. – № 4. – С. 44–48.
4. *Яковлев Ю. С.* О концепции построения и выбора распределённых баз данных информационно-поисковых систем // Математичні машини і системи. – 2013. – № 2. – С. 35–53.
5. *Чумаченко Е. И., Захаров С. С.* Алгоритмическое обеспечение распределенных баз данных // Штучний інтелект. – 2013. – № 1. – С. 49–54.
6. *Цегелик Г. Г.* Математичне програмування: навч. пос. – Львів: ЛНУ ім. Івана Франка, 2011. – 168 с.
7. *Coffman E. G., Garey M. R., Johnson D. S.* Approximation algorithms for bin packing: A survey // Approximation algorithms for *NP*-hard problems. – Boston: PWS Publishing Co., 1996. – P. 46–93.

Одержано 02.10.2017