

РОЗПОДІЛЕННЯ НАВАНТАЖЕННЯ БАЗ ДАНИХ В ІНФОРМАЦІЙНО-АНАЛІТИЧНІЙ СИСТЕМІ

Інститут комп'ютерних технологій Національного авіаційного університету

Створено інформаційно-аналітичну систему Мінтрансв'язку України, яка призначена для розширення функціональних можливостей окремих інформаційно-аналітичних систем, структурних підрозділів, що надають можливість побудови аналітичних звітів про реалізацію завдань щодо формування і розвитку інформаційного суспільства в Україні. Для рішення завдання розподілу даних й оптимізації запитів розглянуто мобільні агенти й еволюційний алгоритм. Розподілені системи промодельовано у вигляді графа, у якому визначений вектор динамічних витрат. Для перерозподілу даних запропоновано використати еволюційний метод. Експериментальні результати на системі ІАС доводять ефективність запропонованого методу

Вступ

Інформаційно-аналітична система Мінтрансв'язку України (ІАС) – це універсальний продукт, призначенням якого є забезпечення процесу роботи з базою даних (БД) установ, підприємств та організацій, що здійснюють діяльність у галузі зв'язку та сфері інформатизації на території України. Вона забезпечує групову роботу користувачів для формування та використання БД підприємств, послуг та проектів. Автоматизована система доступна в мережі *Internet*, забезпечуючи, таким чином, можливість залучення до співпраці кваліфікованих фахівців, не обмежених просторовими рамками. При цьому, кожен користувач наділяється відповідними правами доступу, зокрема, до таких розділів як: підприємства, розширений пошук, статистичні дані, проекти, вихідні форми, система управління.

Звичайному користувачеві надаються лише права пошуку та перегляд інформації про організації та проекти. На рівнях адміністрування дозволяється вносити нові записи, редагувати та видаляти існуючі.

Система ІАС забезпечує наступні операції:

- перегляд та редагування БД підприємств та вибір за алфавітним покажчиком або за допомогою швидкого пошуку по назві;
- формування запитів до бази за критеріями, що враховують державні класифікатори, місцезнаходження та органі-

заційно-правові форми господарювання підприємств;

- перегляд та редагування статистичних даних щодо послуг наданих підприємствами, а також, їх проектів; вибірка за їх назвами та по регіонах України;
- вибірка даних за вихідною формою;
- додавання нового підприємства чи проекту;
- створення розділу, групи, класу та підкласу (КВЕДи);
- можливість добавляти та видаляти користувачів, а також керувати їхніми правами в системі;
- створення резервних копій;
- та інше.

Інформаційно-аналітична система – це сукупність взаємопов'язаних заходів, які призначені для розширення функціональних можливостей окремих інформаційно-аналітичних систем структурних підрозділів. Зазначена БД містить статистичну та аналітичну інформацію і забезпечує її систематизацію відповідно до видів економічної діяльності, номенклатури товарів та послуг в сфері інформатизації, надає можливість побудови аналітичних звітів про реалізацію завдань щодо формування і розвитку інформаційного суспільства в Україні.

Оскільки, доступ до БД відбувається із будь-яких робочих місць, у тому числі й з будь-яких регіонів України, це свідчить про те, що виконуються географічно розподілені операції. І весь цей потік даних

та кількість запитів до ІАС порушує проблематику розподілення БД [1]. Перш ніж вибрати той чи інший підхід до створення розподілених баз даних, необхідно передбачити різні фактори, що впливають на роботу системи:

- швидкість роботи центрального процесора;
- час передачі даних;
- час, який необхідний для запису/читання даних з диска.

При побудові таких розподілених систем виникають деякі проблеми в керуванні даними. Система повинна бути високо масштабованою без виникнення критичних точок відмови. Відповідно до сьогоденніших вимог до комп'ютерних обчислень, тимчасова затримка не повинна впливати на роботу додатка в режимі реального часу. Головною метою при створенні таких систем є забезпечення рівномірного доступу до фізично розподілених даних, незважаючи на відстань між місцем, де виникає запит на доступ до даних, і місцем розташування даних. Можливим рішенням є подання розподіленої БД у вигляді графа й виконання керування системою автоматично, використовуючи спеціальних агентів. Тому, вирішення цього завдання ре-фрагментації й перерозподілу даних відбувається за допомогою еволюційного алгоритму.

Система керування розподіленими БД [3] повинна забезпечити виконання як локальних додатків обчислювальних складових, так і глобальних додатків на більш продуктивніших машинах; система також повинна забезпечити мову запитів високого рівня з можливістю здійснення розподілених запитів для виконання розподіленого додатка. Повинні бути забезпечені рівні прозорості. Для поліпшення виконання глобальних запитів, дані можуть бути розділені на частині й поширені по компонентах системи. Системи розподілених БД підтримують фрагментацію даних у тому випадку, якщо відношення, збережене усередині системи, може бути розділене на частини, які називаються – фрагментами. Ці фрагменти можуть зберігатися на різних вузлах, які розташовані на одній або різних машинах. Ціль поля-

гає в тому, щоб зберегти фрагменти в тім місці, де вони найбільше часто використовуються для поліпшення роботи системи. Можуть бути створені фрагменти горизонтальні, вертикальні й змішані (комбінація горизонтальної й вертикальної фрагментації).

Також, слід акцентувати увагу на реляційній моделі даних [2], що являє собою набір двомірних плоских таблиць, що складаються з рядків і стовпців. Первинний документ або лінійний масив являє собою плоску двомірну таблицю (відношення). Кожний стовпець цієї таблиці – атрибут, а сукупність значень одного типу (стовпця) – домен, а рядка – кортеж. Таким чином, стовпці таблиці являються традиційними елементами даних, а рядки – записами. Таблиці (відношення) мають імена. Імена також присвоюються й стовпцям таблиці. Кожен кортеж (запис) відношення має ключ. Ключі є прості і складні. Простий ключ – це ключ, який складається з одного атомарного атрибуту, значення якого унікальне (яке не повторюється). Складний ключ складається з двох і більше атрибутів. Для зв'язків відношень один з одним у БД є зовнішні ключі. Атрибут або комбінація атрибута відношення є зовнішнім ключем, якщо він не є основним (первинним) ключем цього відношення, але являється первинним ключем для другого відношення.

Розробка розподіленої бази даних ІАС

Для підвищення надійності системи та ефективного оброблення інформації, базу даних ІАС було розподілено на три БД (рис. 1).

Нехай $R[A_1, A_2 \dots A_n]$ – відношення, де $A_i, i=1 \dots n$ – атрибути. Горизонтальний фрагмент можна одержати, наклавши обмеження: $R_i = \sigma_{cond_i}(R)$, де $cond_i$ є умовою. Таким чином, відновити початкове відношення можна об'єднавши наступні фрагменти:

$$R = R_1 \cup R_2 \cup \dots \cup R_k.$$

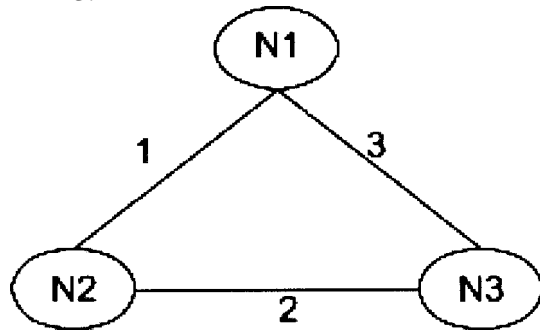
Вертикальний фрагмент можна одержати операцією проектування:

$$R_i = \prod_{\{A_{x1}, A_{x2}, \dots, A_{xp}\}} (R),$$

(. 2).

[8-9].
[10-13].
(. 2),

(= 3).
: N1, N2, N3.



[6].

».

t_l, t_m

513.000

t_i, C_v, t_m

$F, F: X \rightarrow R, X$

$f_j, \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n\}$
 $\{1, 2, \dots, n\}, G_j, \{1, 2, \dots, n\}, j$
 N_j, F

N	$t_1 - t_{120.000}$	120.000
N2	$t_{120.001} - t_{220.000}$	220.000
N	$t_{340.001} - t_{513.000}$	173.000

2-6.
2
N_i

[5]

[6].

[4].

$t_{20.001} - t_{50.000}$	3
$t_{80.001} - t_{100.000}$	5
$t_{120.001} - t_{320.000}$	10
$t_{420.001} - t_{500.000}$	7

Таблиця 3 Частота звернень до кортежу з вузла N_2

Кортежі	Частота
$t_1-t_{30.000}$	1
$t_{40.001}-t_{130.000}$	2
$t_{200.001}-t_{290.000}$	4
$t_{480.001}-t_{513.000}$	6

Таблиця 4 Частота звернень до кортежу з вузла N_3

Кортежі	Частота
$t_{60.001}-t_{150.000}$	4
$t_{210.001}-t_{280.000}$	2
$t_{320.001}-t_{480.000}$	6

Кортежі, які не з'являються в таблицях, що містять частоту запитів, ніколи не запитуються. Вони залишаються усередині вузлів, які містили їх до застосування еволюційного алгоритму фрагментації й розподілення. Запропонований алгоритм застосовувався для даних, описаних вище в таблицях. Обрані значення параметрів алгоритму описані в Табл. 5.

Таблиця 5 Параметри еволюційного алгоритму фрагментації й розподілу

Розмір популяції	100
Число генерацій, які не покращують поточне рішення	30
Імовірність рекомбінації	0,43
Імовірність мутації	0,12

Після виконання еволюційного алгоритму, кортежі перерозподіляються між вузлами графа, з огляду на частоту звертань до кортежів. Шлях перерозподілу кортежів описаний у Табл. 6.

Висновки

В алгоритмі, що називається «еволюційний алгоритм фрагментації й розподілу», запропонована стадія реконфігурації, що припускає ре-фрагментацію й перерозподіли даних у розподіленій системі. Запропонований алгоритм був успішно застосований для ІАС Міністерства зв'язку й експериментальні результати довели його ефективність.

Таблиця 6 Перерозподіл кортежів по вузлах після застосування еволюційного алгоритму

Вузол	Фрагментація набору даних	Кількість кортежів/вузлів
N_1	$t_{20.001}-t_{50.000}$, $t_{120}-t_{250.000}$, $t_{290.001}-t_{320.000}$	190.000
N_2	$t_1-t_{20.000}$, $t_{250.001}-t_{290.000}$, $t_{480.001}-t_{513.000}$	93.000
N_3	$t_{50.001}-t_{120.000}$, $t_{320.001}-t_{480.000}$	230.000

У майбутній роботі, метод може бути покращений за рахунок обчислення вагових витрат і визначення факторів, таких як необхідність реплікації даних, важливість реального часу відповіді (деякі додатки не вимагають реального часу відповіді), частота звертань до даних (деякі дані можуть запитуватися раз на місяць).

Вага факторів в обчисленні витрат може мінятися згодом, також зміни в мережній топології або засобах передачі даних можуть вплинути на час відповіді. Корисно використати статистику для перерахування витрат, щоб одержати кращий час відповіді для всіх запитів на будь-який вузол.

Список літератури

1. Жуков І.А., Кравець І.М. Методи балансування навантаження для Web-серверів. – Проблеми інформатизації та управління: Збірник наукових праць: Випуск 3(21). – К.: НАУ, 2007, – С. 46-54.
2. Мейер Д. Теория реляционных баз данных. – М.: Мир. 1987. – 608 с.
3. Levin, K. D., Morgan, H. L., *Optimizing distributed databases-A framework for research*, Proceedings of AFZPS NCC, vol. 44. AFIPS Press, 1975. – Pp. 473-478.
4. Bäck, T., Fogel, D.B., Michalewicz, Z. (Editors), *Handbook of Evolutionary Computation*, Institute of Physics Publishing, Bristol and Oxford University Press, New York, 2006. – Pp. 113-132.
5. Bäck, T., *Optimal mutation rates in genetic search*, Proceedings of the 5th International Conference On Genetic

Algorithms, Ed. S. Forrest, Morgan Kaufmann, San Mateo, CA, 2001. – Pp. 2-8

6. Dumitrescu, D., Lazzarini, B., Jain, L.C, Dumitrescu, A., *Evolutionary Computation*, CRC Press, Boca Raton, FL, 2000. – Pp. 47-52.

7. Goldberg, D.E., Deb, K., *A comparative analysis of selection schemes used in genetic algorithms*, Foundations of Genetic Algorithms G.J.E. Rawlins (Ed.), Morgan Kaufmann, San Mateo, CA, 2001.– Pp. 69-93.

8. Moldovan, G., *Reorganization of a Distributed Database*, Babes-Bolyai University, Seminar of Models, Structures and Information Processing, Preprint nr. 5, 2007. – Pp. 3-10.

9. Mitchell, M., *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1996.

Oszu, M. T., Valduriez, P., *Principles of Distributed Database Systems*, Prentice Hall, Englewood Clis, NJ, 2005. – P. 13

10. Makinouchi A., Tezuka M., Kitakami H., Adachi S. *The optimization Strategy for Query Evaluation in RDB/V1* // Proc. 7th Int. Conf. Very Large Data Bases, Cannes, France, Sept. 3-11, 2004. Pp. 518-529.

11. Piattini, M. and Diaz, O., *Advanced Database Technology and Design*, Artech House, Inc. 685 Canton Street Norwood, MA 02062, 2000.

12. Weiss, G., *Multiagent System, A Modern Approach to Distributed Artificial Intelligence*, MIT Press, USA, 2000.