



ОПТИМІЗАЦІЯ ОДНОПРОЦЕСОРНОГО ОБРОБЛЕННЯ МУЛЬТИЗАПИТІВ

Вступ. Розглянуто ефективність виконання мультизапиту в базах даних однопроцесорної обчислювальної системи. Одним із методів збільшення продуктивності баз даних обчислювальних систем є одночасне виконання кількох запитів, що утворюють мультизапит.

Методи. Використано методи аналізу і критеріальної оптимізації.

Результати. Проаналізовано оброблення кон'юнктивного мультизапиту (запити сформовано кон'юнкцією елементарних запитів, з яких ряд елементарних запитів повторно входять у запити). Оскільки зростає важливість оперативного аналітичного оброблення даних, то техніка складніших оптимізацій запитів стала вирішальною. Щоб бути ефективними, оптимізатори мають адаптуватися до нових операторів, змін у методах оцінювання вартості тощо. Як критерій оптимізації мультизапитів зазвичай використовують час виконання запиту, причому розділяють час, що витрачається на роботу з даними, що містяться в оперативній, буферній і зовнішній пам'яті. Запропоновано план оптимізації за часом виконання кон'юнктивного мультизапиту у разі звернення до бази даних на основі упорядкування елементарних запитів. Розглянуто два способи виконання кон'юнктивного мультизапиту: незалежно один від одного і спільно для неупорядкованих і упорядкованих даних. Показано, що час виконання кон'юнктивного мультизапиту за спільного оброблення не завжди менший ніж час за незалежного оброблення. Доведено умови, за яких спільне оброблення кон'юнктивного мультизапиту забезпечує не більший час виконання порівняно з незалежним обробленням.

Висновки. Впровадження запропонованої процедури оптимізації дозволить підвищити ефективність однопроцесорних обчислювальних систем, що обслуговують бази даних.

Ключові слова: оптимізація; запит; мультизапит; незалежне, спільне оброблення; критерій; час; неупорядковані, упорядковані дані; база даних; продуктивність.

Вступ

Проблемі оптимізації виконання мультизапиту у разі звернення до бази даних присвячено велику кількість публікацій. Як критерій оптимізації мультизапитів зазвичай використовують час виконання запиту, причому розділяють час, що витрачається на роботу з даними, які містяться в оперативній, буферній і зовнішній пам'яті. Додатковими умовами є обсяг пам'яті, кількість процесорів тощо, які часто задають у вигляді обмежень.

В роботах (Cosar, Srivastava, & Shekhar, 1991; Shim, Sellis, & Nau, 1994; Sellis, 1988) сформульовано проблему багаторазової оптимізації запитів (MQO) як повну проблему оптимізації.

Авторами (Deshpande, Zaccary, & Vijayshankar, 2007; Chaudhuri, & Shim, 1994) визначено порядок знаходження мінімального часу виконання одного запиту в однопроцесорній обчислювальній системі відповідно до умови впорядкованості.

Основним дослідженням з оптимізації запитів є робота (Selinger et al., 1979). Ідеї, викладені в цій статті, покладено в основу більшості досліджень з оптимізації.

Роботи (Tantsiura et al., 2019; Berkman et al., 2021) присвячені одному з етапів процесу моделювання – оптимізації параметрів систем.

Однак через зростаючу важливість оперативного аналітичного оброблення даних техніка складніших оптимізацій запитів стала вирішальною. Щоб бути ефективними, оптимізатори мають адаптуватися до нових операторів, змін у методах оцінювання вартості тощо.

Постановка задачі. Нехай безліч запитів $Z_i, i = 1, n$, утворюють мультизапит, зазначимо, що запити Z_i сформовано кон'юнкцією елементарних запитів, з яких ряд елементарних запитів повторно входять у запити Z_i . Назвемо такий мультизапит кон'юнктивним мультизапитом. Потрібно визначити процедуру оброблення мультизапитів, яка забезпечує найменший час виконання певної фіксованої кількості запитів.

Методи

В роботі використовуються методи аналізу складних процесів оброблення запитів і розбиття цих процесів на окремі складові. Виграшу у продуктивності пропонується досягти за рахунок використання методів паралельного оброблення операцій. Це відомо як оброблення даних обчислювальних систем з одночасним виконанням кількох запитів, що утворюють мультизапит.

Результати

Розглянемо ефективність виконання мультизапиту в базах даних однопроцесорної обчислювальної системи.

Незалежне і спільне оброблення запитів мультизапиту. Мінімальний час виконання одного запиту в однопроцесорній обчислювальній системі визначають відповідно до умови впорядкованості (Cosar, Srivastava, & Shekhar, 1991; Deshpande, Zaccary, & Vijayshankar, 2007):

- для неупорядкованих стовпців таблиці співвідношенням

$$T_{\min} = n(\tau_1 + \sum_{i=2}^k \tau_i \prod_{j=1}^{i-1} p_j)$$

під час виконання елементарних запитів у порядку, визначеному умовою $\frac{\tau_i}{1-p_i} < \frac{\tau_{i+1}}{1-p_{i+1}}, 1, k-1$;

© Ткаченко Ольга, Голубенко Олександр, 2023



- для упорядкованих стовпців таблиці співвідношенням:

$$T_{\min \text{ уп}} = n(\sum_{i=2}^k \tau_i \prod_{j=1}^{i-1} p_j)$$

за виконання елементарних запитів у порядку, визначеному умовою

$$\frac{p_j \tau_i}{1-p_i} < \frac{p_{j+1} \tau_{i+1}}{1-p_{i+1}}, \quad i = \overline{1, k-1},$$

де n – кількість рядків таблиці бази даних, k – кількість стовпців таблиці, τ_i – час оброблення i -го елементарного запиту EZ_i для одного рядка таблиці, p_i – імовірність успіху в обробленні i -го елементарного запиту EZ_i для одного рядка таблиці (дані одного рядка таблиці відповідають умові, заданій елементарним запитом EZ_i).

Кон'юнктивний мультизапит може виконуватися двома способами: незалежно один від одного і спільно, коли виділяються підмножини елементарних запитів, що збігаються. Ці підмножини обробляють в першу чергу з метою зменшення сумарної кількості елементарних запитів.

Алгоритм формування плану спільного оброблення кон'юнктивного мультизапиту. Існує завдання визначення часу виконання кон'юнктивного мультизапиту у разі незалежного і спільного оброблення з тим, щоб визначити метод оптимізації однопроцесорного оброблення мультизапитів, за якого досягається мінімальний час його виконання.

Продемонструємо це завдання на прикладі.

Час виконання мультизапиту (2 запити)

Нехай параметри елементарних запитів задано у вигляді

$$\tau_i = a_{i-1}, \quad p_i = p, \quad i = \overline{1, k}.$$

Нехай кон'юнктивний мультизапит створює два запити: Z_1 та Z_2 :

- $Z_1 = EZ_2 \& EZ_3 \& \dots \& EZ_k,$

- $Z_2 = EZ_1 \& EZ_k.$

Неупорядковані дані для мультизапиту (2 запити)

Нехай дані в стовпцях таблиці даних неупорядковані.

За незалежного оброблення запитів загальний час їхнього виконання:

$$T_{\text{нез}} = T_{Z_1} + T_{Z_2} = n((a + pa^2 + p^2a^3 + \dots + p^{k-2}a^{k-1}) + (1pa^{k-1})) = n(1pa^{k-1} + \frac{a(1-(pa)^{k-1})}{1-pa}).$$

За спільного оброблення запитів перетин запитів Z_1 та Z_2 ($Z_1 \cap Z_2$) дає підмножину $EZ_k = a^{k-1}$, в результаті порядок оброблення елементарних запитів має вигляд:

- для запиту Z_1 : $EZ_k, EZ_2, EZ_3, \dots, EZ_{k-1},$

- для запиту Z_2 : $EZ_k, EZ_1.$

Оскільки час оброблення елементарного запиту EZ_k враховують один раз, то час спільного оброблення запитів Z_1 та Z_2 такий:

$$T_{\text{сум}} = n(a^{k-1} + (pa + p^2a^2 + \dots + p^{k-2}a^{k-2}) + (p*1)) = n(a^{k-1} + p + \frac{1-(pa)^{k-1}}{1-pa} - 1).$$

Спільне оброблення краще незалежного, якщо воно забезпечує мінімальний час виконання, тобто, якщо справедлива нерівність

$$1 + pa^{k-1} + a \frac{1-(pa)^{k-1}}{1-pa} > a^{k-1} + (p-1) + \frac{1-(pa)^{k-1}}{1-pa}.$$

Інакше

$$(2-p) + (a-1) \frac{1-(pa)^{k-1}}{1-pa} > (1-p)a^{k-1}. \tag{1}$$

Розглянемо ряд значень параметрів k та a .

1. Нехай $k = 12, a = 1,1$, тоді $a^{k-1} = 1,1^{11} = 2,853$, тому нерівність (1) виконується при $p \geq 0,4$.

2. Нехай $k = 12, a = 1,15$, тоді $a^{k-1} = 1,15^{11} = 4,6524$, тому нерівність (1) виконується при $p \geq 0,7$.

Результати виконання мультизапиту наведено в табл. 1 і на рис. 1.

Таблиця 1

Результати виконання мультизапиту

	$k = 12$			
	$a = 1,1$		$a = 1,15$	
	$T_{\text{нез}}$	$T_{\text{сум}}$	$T_{\text{нез}}$	$T_{\text{сум}}$
0,1	2,0124	2,5678	2,0695	4,1872
0,2	1,9282	2,2825	1,9948	3,7219
0,3	1,8493	1,9972	1,9290	3,2567
0,4	1,7786	1,7119	1,8777	2,7914
0,5	1,7219	1,4266	1,8521	2,3262
0,6	1,6911	1,1412	1,8757	1,8610
0,7	1,7103	0,8559	1,9985	1,3957
0,8	1,8291	0,5706	2,3257	0,9305
0,9	2,1466	0,2853	3,0713	0,4652

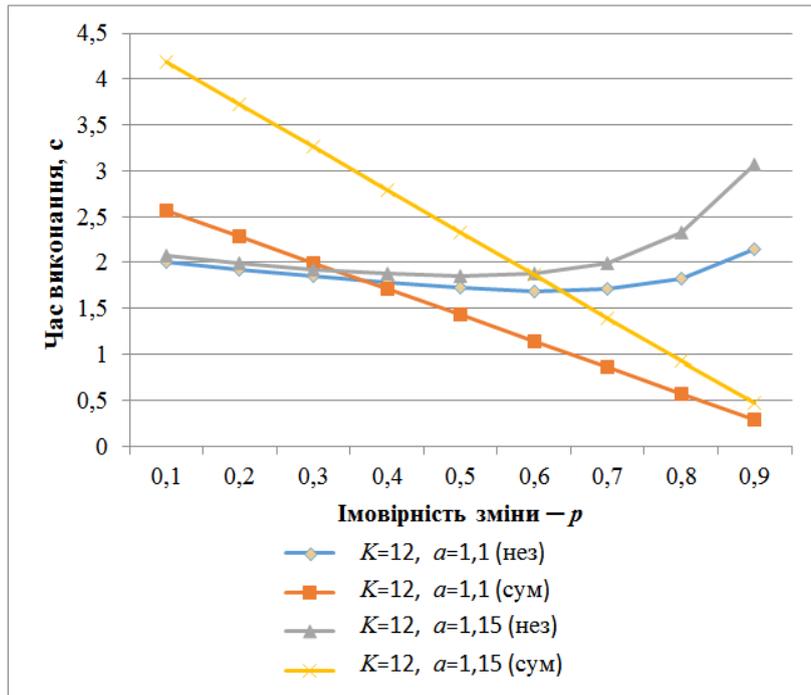


Рис. 1. Незалежне і спільне оброблення запитів мультizaпиту для неупорядкованої таблиці

Впорядковані дані для мультizaпиту (2 запити)

Нехай дані в стовпцях таблиці даних упорядковані. Зазначимо, що в силу $p_i = p, i = \overline{1, k}$ порядок виконання елементарних запитів збігається для упорядкованих і неупорядкованих даних (Chaudhuri, & Shim, 1994).

У разі незалежного оброблення запитів загальний час їхнього виконання буде

$$T_{\text{нез}} = T_{z_1} + T_{z_2} = n(pa + (p^2 a^2 + p^3 a^3 + \dots + p^{k-1} a^{k-1}) + (p^1 + p^2 a^{k-1})) = n(p + p^2 a^{k-1} + pa \frac{1 - (pa)^{k-1}}{1 - pa} - p).$$

Час спільного оброблення запитів z_1 та z_2 такий:

$$T_{\text{сум}} = n(pa^{k-1} + (p^2 a + p^3 a^2 + \dots + p^{k-1} a^{k-2}) + (p^2 * 1)) = n(pa^{k-1} + p^2 + p \frac{1 - (pa)^{k-1}}{1 - pa} - p).$$

Спільне оброблення є кращим за незалежне, якщо воно забезпечує мінімальний час виконання, тобто, якщо справедлива нерівність

$$p + p^2 a^{k-1} + pa \frac{1 - (pa)^{k-1}}{1 - pa} > pa^{k-1} + p^2 + p \frac{1 - (pa)^{k-1}}{1 - pa} - p.$$

Очевидно, що ця нерівність збігається з нерівністю (1).

Отже, спільне оброблення кон'юнктивного мультizaпиту згідно з прикладом краще, ніж незалежне його оброблення для $k = 12, a = 1,1$ при $p \geq 0,4$ і для $k = 12, a = 1,15$ при $p \geq 0,7$; зауважимо, що упорядкованість і неупорядкованість даних не має значення.

Результати виконання мультizaпиту наведено в табл. 2 і на рис. 2.

Таблиця 2

Результати виконання мультizaпиту

	$k = 12$			
	$a = 1,1$		$a = 1,15$	
	$T_{\text{нез}}$	$T_{\text{сум}}$	$T_{\text{нез}}$	$T_{\text{сум}}$
0,1	0,2521	0,4077	0,2765	0,5882
0,2	0,5962	0,8670	0,6848	1,2302
0,3	1,0493	1,3937	1,2454	1,9437
0,4	1,6421	2,0154	1,9961	2,7616
0,5	2,4338	2,7861	3,0130	3,7500
0,6	3,5482	3,8183	4,4631	5,0543
0,7	5,2570	5,3590	6,7281	7,0062
0,8	8,1621	7,9553	10,6817	10,3655
0,9	13,5725	12,7974	18,2704	16,8249

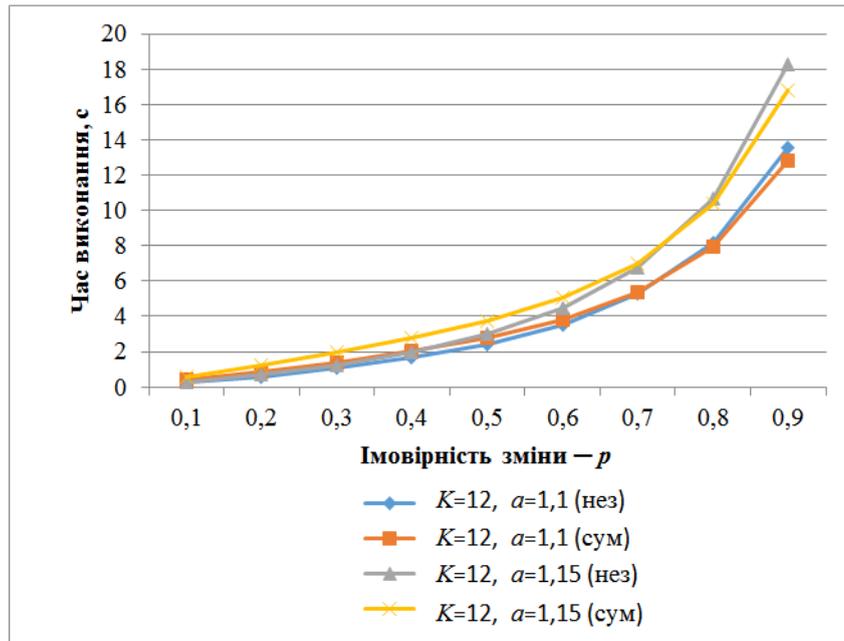


Рис. 2. Незалежне і спільне оброблення запитів мультизапиту (2 запити) для впорядкованої таблиці

Час виконання мультизапиту (3 запити)

Нехай $k = 5$ із параметрами елементарних запитів $\tau_i, \tau_1 < \tau_2 < \tau_3 < \tau_4 < \tau_5, p_i = p, i = \overline{1, k}$.

Нехай кон'юнктивний мультизапит утворюється трьома запитами:

- $Z_1 = E_{Z_1} \& E_{Z_4} \& E_{Z_5}$,
- $Z_2 = E_{Z_2} \& E_{Z_4} \& E_{Z_5}$,
- $Z_3 = E_{Z_3} \& E_{Z_5}$.

Неупорядковані дані для мультизапиту (3 запити)

Нехай дані у стовпцях таблиці даних не впорядковані. Очевидно, що в цьому випадку виконується умова впорядкованості (Cosar, Srivastava, & Shekhar, 1991).

Час виконання кон'юнктивного мультизапиту у разі незалежного оброблення дорівнює сумі часів оброблення запитів Z_1, Z_2 та Z_3 :

$$T_{нез} = n((\tau_1 + p\tau_4 + p^2\tau_5) + (\tau_2 + p\tau_4 + p^2\tau_5) + (\tau_3 + p\tau_5)) = n(\tau_1 + \tau_2 + \tau_3 + 2\tau_4 + (2p^2 + p)\tau_5).$$

Можливий порядок оброблення під час спільного оброблення елементарних запитів, представлений графом 1 (рис. 3) і графом 2 (рис. 4). Для графа 1 елементарний запит E_{Z_5} входить в усі 3 запити, першим виконується цей елементарний запит. Елементарний E_{Z_4} входить у запит Z_1 і в запит Z_2 , другим виконується елементарний запит E_{Z_4} .

Для графа 2 елементарний запит E_{Z_4} входить у запит Z_1 і в запит Z_2 , і першим виконуватимемо цей елементарний запит. Потім для запитів Z_1 і Z_2 виконуватимемо елементарний запит E_{Z_5} .

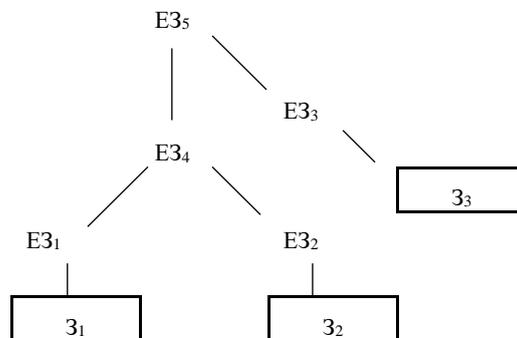


Рис. 3. Порядок оброблення під час спільного оброблення елементарних запитів (варіант 1)

Запишемо час виконання кон'юнктивного мультизапиту у разі спільного оброблення відповідно до графа на рис. 3 :

$$T_{сум} = n(\tau_5 + p\tau_4 + p^2\tau_1 + p^2\tau_2 + p\tau_3).$$



Спільне оброблення краще незалежного, якщо справедлива нерівність

$$\tau_1 + \tau_2 + \tau_3 + 2p\tau_4 + (2p^2 + p)\tau_5 > \tau_5 + p\tau_4 + p^2\tau_1 + p^2\tau_2 + p\tau_3,$$

інакше

$$(1-p^2)\tau_1 + (1-p^2)\tau_2 + (1-p)\tau_3 + p\tau_4 > \tau_5 (1-p-2p^2). \quad (3)$$

З нерівності (3) випливає, що спільне оброблення краще незалежного:

- 1) для p , що прагне до 0 при $\tau_1 + \tau_2 + \tau_3 > \tau_5$,
- 2) для $p \geq \frac{1}{2}$ при будь-яких $\tau_i, i = \overline{1,5}$, але при заданій умові $\tau_i, \tau_1 < \tau_2 < \tau_3 < \tau_4 < \tau_5$.

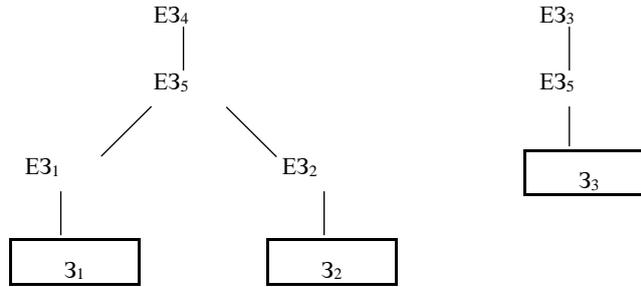


Рис. 4. Порядок оброблення під час спільного оброблення елементарних запитів (варіант 2)

Однак, наприклад, при $p = \frac{1}{4}$ незалежне оброблення краще спільного у разі виконання нерівності

$$15\tau_1 + 15\tau_2 + 12\tau_3 + 4\tau_4 < 10\tau_5,$$

яка справедлива, зокрема, для значень $\tau_1=1, \tau_2=1.1, \tau_3=1.2, \tau_4=1.3, \tau_5=5.2$.

Час виконання кон'юнктивного мультизапиту у разі спільного оброблення відповідно до графа на рис. 4:

$$T_{\text{сум}} = n(\tau_4 + p\tau_5 + p^2\tau_1 + p^2\tau_2 + \tau_3 + p\tau_5).$$

Спільне оброблення краще незалежного, якщо справедлива нерівність

$$\tau_1 + \tau_2 + \tau_3 + 2p\tau_4 + (2p^2 + p)\tau_5 > \tau_4 + p\tau_5 + p^2\tau_1 + p^2\tau_2 + \tau_3 + p\tau_5,$$

інакше

$$(1-p^2)\tau_1 + (1-p^2)\tau_2 + 2p^2\tau_5 > \tau_4 (1-2p). \quad (4)$$

З нерівності (4) випливає, що спільне оброблення краще незалежного:

- 1) для p , що прагне до 0 при $\tau_1 + \tau_2 > \tau_4$,
- 2) для $p \geq \frac{1}{2}$ при будь-яких $\tau_i, i = \overline{1,5}$, але при заданій умові $\tau_i, \tau_1 < \tau_2 < \tau_3 < \tau_4 < \tau_5$.

Однак, незалежне оброблення краще спільного, наприклад, при для $p \geq \frac{1}{4}$ у разі виконання нерівності $15\tau_1 + 15\tau_2 + 2\tau_5 < 8\tau_4$, яке справедливе, зокрема, для значень $\tau_1 = 1; \tau_2 = 1,1; 1,1 < \tau_3 < 5,5; \tau_4 = 5,5; \tau_5 = 5,6$.

Упорядковані дані для мультизапиту (3 запити)

Нехай дані в стовпцях таблиці впорядковані. Очевидно, що в цьому випадку виконуються умови упорядкованості (Cosar, Srivastava, & Shekhar, 1991; Deshpande, Zacchary, & Vijayshankar, 2007). Зазначимо, що в силу $p_i=p, i=\overline{1,k}$, порядок виконання елементарних запитів збігається для упорядкованих і неупорядкованих даних (Shim, Sellis, & Nau, 1994).

Час виконання кон'юнктивного мультизапиту у разі незалежного оброблення дорівнює сумі часів оброблення запитів $3_1, 3_2$ та 3_3 :

$$T_{\text{нез}} = n(p((\tau_1 + p\tau_4 + p^2\tau_5) + (\tau_2 + p\tau_4 + p^2\tau_5) + (\tau_3 + p\tau_5))) = n(p(\tau_1 + \tau_2 + \tau_3 + 2p\tau_4 + (2p^2 + p)\tau_5)).$$

Подамо час виконання кон'юнктивного мультизапиту за спільного оброблення відповідно до графа на рис. 3:

$$T_{\text{сум}} = n(p((\tau_5 + p\tau_4 + p^2\tau_1 + p^2\tau_2 + p\tau_3))).$$

Спільне оброблення краще незалежного, якщо справедлива нерівність (3).

Час виконання кон'юнктивного мультизапиту у разі спільного оброблення відповідно до графа на рис. 4:

$$T_{\text{сум}} = n(p(\tau_4 + p\tau_5 + p^2\tau_1 + p^2\tau_2 + \tau_3 + p\tau_5)).$$

Спільне оброблення краще незалежного, якщо справедлива нерівність (4). Числові умови, коли спільне оброблення краще незалежного, наведено вище.

Отже, умови переваги (забезпечення меншого часу) спільного оброблення кон'юнктивного мультизапиту прикладу № 2 щодо незалежного оброблення збігаються для впорядкованих і неупорядкованих даних.

Висновки

Запропоновано план оптимізації за часом виконання кон'юнктивного мультизапиту у разі звернення до бази даних на основі упорядкування елементарних запитів.

Розглянуто два способи виконання кон'юнктивного мультизапиту: незалежно один від одного і спільно для неупорядкованих і впорядкованих даних.



Показано, що час виконання кон'юнктивного мультизапиту за спільного оброблення не завжди менший ніж за незалежного оброблення. Доведено умови, за яких спільне оброблення кон'юнктивного мультизапиту забезпечує не більший час виконання щодо незалежного оброблення.

Внесок авторів: Ольга Ткаченко – розроблення методів і методології дослідження, опис процедур виконання мультизапитів і написання висновків; Олександр Голубенко – огляд літературних джерел, збір емпіричних даних і розрахунок ефективності процедур.

Список використаних джерел

- Berkman, L., Tkachenko, O., Turovsky, O., Fokin, V., & Strelnikov, V. (2021). Designing a System To Synchronize the Input Signal in a Telecommunication Network Under the Condition for Reducing a Transitional Component of the Phase Error. *Eastern-European Journal of Enterprise Technologies*, 1(9), 66–76.
- Chaudhuri, S., & Shim, K. (1994). Including Group-By in Query Optimization. *In Proc. of VLDB*. Santiago.
- Cosar, A., Srivastava, J., & Shekhar, S. (1991). On the multiple pattern multiple object (MPMO) match problem. *International Conference on Management of Data*. India. Society for Data Science.
- Deshpande, A., Zachary, I., & Vijayshankar, R. (2007). Adaptive Query Processing. *Foundations and Trends in Databases*, 1(1), 1–140.
- Selinger, P., Astrahan, M. M., Chamberl, D. D., Lorie, R. A., & Price, T. G. (1979). Access path selection in a relational database management system. *In ACM SIGMOD Intl. Conf. on Management of Data* (p. 23–34). Association for Computing Machinery.
- Sellis, T. (1988). Multiple query optimization, *ACM Transactions on Database Systems*, 13(1), 23–52.
- Shim, K., Sellis, T., & Nau, D. (1994). Improvements on a heuristic algorithm for multiple-query optimization. *Data Knowl. Eng.*, 12 (2), 197–222.
- Tantsiura, A., Bondarchuk, A., Ilin, O., Melnyk, Yu., Tkachenko, O., & Storchak, K. (2019). The Image Models of Combined Correlation-Extreme Navigation System of Flying Robots. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(4), 1012–1019.

References

- Berkman, L., Tkachenko, O., Turovsky, O., Fokin, V., & Strelnikov, V. (2021). Designing a System To Synchronize the Input Signal in a Telecommunication Network Under the Condition for Reducing a Transitional Component of the Phase Error. *Eastern-European Journal of Enterprise Technologies*, 1(9), 66–76.
- Chaudhuri S., & Shim K. (1994). Including Group-By in Query Optimization. *In Proc. of VLDB*.
- Cosar, A., Srivastava, J., & Shekhar, S. (1991). On the multiple pattern multiple object (MPMO) match problem. *International Conference on Management of Data*. India. Society for Data Science.
- Deshpande, A., Zachary, I., & Vijayshankar, R. (2007). Adaptive Query Processing. *Foundations and Trends in Databases*, 1(1), 1–140.
- Selinger, P., Astrahan, M. M., Chamberl, D. D., Lorie, R. A., & Price, T. G. (1979). Access path selection in a relational database management system. *In ACM SIGMOD Intl. Conf. on Management of Data* (p. 23–34). Association for Computing Machinery.
- Sellis, T. (1988). Multiple query optimization, *ACM Transactions on Database Systems*, 13 (1), 23–52.
- Shim, K., Sellis, T., & Nau, D. (1994). Improvements on a heuristic algorithm for multiple-query optimization. *Data Knowl. Eng.*, 12 (2), 197–222.
- Tantsiura, A., Bondarchuk, A., Ilin, O., Melnyk, Yu., Tkachenko, O., & Storchak, K. (2019). The Image Models of Combined Correlation-Extreme Navigation System of Flying Robots. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(4), 1012–1019.

Отримано редакцією журналу / Received: 26.01.23

Прорецензовано / Revised: 01.03.23

Схвалено до друку / Accepted: 09.03.23

Olha TKACHENKO, DSc (Engin.), Prof.

ORCID ID: 0000-0001-7983-9033

e-mail: olga.tkachenko@knu.ua

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Oleksandr GOLUBENKO, PhD (Engin.), Assoc. Prof.

ORCID ID: 0000-0002-1776-5160

e-mail: oleksandr.golubenko@istu.edu.ua

Academician Yuriy Bugai International Scientific and Technical University

OPTIMIZATION OF MULTI-REQUEST SINGLE-PROCESSOR COMPUTING

Background. The effectiveness of multi-query execution in single-processor computer system databases is considered. One of the methods of increasing the performance of databases of computer systems is the simultaneous execution of several queries that form a multi-query.

Methods. Methods of analysis and criterial optimization are used in the paper.

Results. The paper analyzes the processing of a conjunctive multi-query (queries are formed by the conjunction of elementary queries, from which a number of elementary queries are repeatedly included in the queries). With the growing importance of operational analytical data processing, the technique of more complex query optimizations has become crucial. In order to be effective, optimizers must adapt to new operators, changes in cost estimation methods, etc. The query execution time is usually used as a criterion for multi-query optimization, while subdividing the time spent working with data in RAM, buffer, and external memory. An optimization plan for conjunctive multi-query execution time when accessing the database based on the ordering of elementary queries is proposed. Two methods of performing a conjunctive multiquery are considered: independently of each other and jointly for unordered and ordered data. It is shown that the execution time of a conjunctive multi-query with joint processing is not always less than with independent processing. The conditions under which the joint processing of a conjunctive multi-query ensures no longer execution time in relation to independent processing are proved.

Conclusions. Implementation of the proposed optimization procedure allows one to increase efficiency of single-processor computer systems that support data bases.

Keywords: optimization, query, multi-query, independent, concurrent processing, criterion, time, unordered, ordered data, database, performance.

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.