УДК 004.02 DOI:10.31673/2412-4338.2021.026168

Ткаченко О.М., Лемешко А.В., Замрій І.В., Кращенко Д.В., Підмогильний О.О.
Державний університет телекомунікацій, Київ

ДИНАМІЧНЕ РІШЕННЯ ПРИ БАГАТОРАЗОВІЙ ОПТИМІЗАЦІЇ ЗАПИТІВ

Анотація: *Розглянуто проблему багаторазової оптимізації запитів як повну проблему оптимізації, визначено вхідні параметри для оптимізації: запити, задачі, плани. Запропонована і обґрунтована архітектура системи, в якій застосовується СУБД з оптимізатором запитів - комп'ютерна система, в якої один або більше число комп'ютерів (система обробки інформації, один або більше число комп'ютерів клієнта, керуючий комп'ютер, один або більше число серверів БД) з'єднані між собою з боку клієнта мережею і з боку сервера мережею. Система обробки даних отримує первинний запит, зроблений одним з комп'ютерів клієнта, створює один або більше число вторинних запитів і передає їх серверам БД, в тому випадку, якщо це необхідно, виконує посилання або оновлення даних, так як це визначено в первинному запиті, а потім повертає отримані в результаті дані на комп'ютер клієнта, з якого був зроблений первинний запит. Модуль обробки введення/виводу, аналізатор запитів, оптимізатор запитів, модуль обробки запитів, контролер оптимізації, пристрій зовнішньої пам'яті є елементами, складовими модуля обробки інформації. Модуль обробки введення/виводу приймає заявку про запит від комп'ютерів клієнта та заявку про управління від управляючого комп'ютера і відповідає на ці заявки. Оптимізатор запитів оптимізує запит, використовуючи дерево запиту, генероване аналізатором запитів, і розробляє процедуру для серії операцій (план виконання запиту), щоб отримати результати обробки запиту. Модуль обробки запитів виконує план виконання запиту, розроблений оптимізатором запиту. Розроблений порядок виконання елементарних запитів, які забезпечують мінімальний час для одного, двох та трьох процесорів. Представлено порівняння часу виконання мультизапиту при спільній і несумісній обробці в залежності від числа процесорів. Мінімальний час виконання запиту може бути досягнуто при виконанні елементарних запитів у відповідному порядку, визначеному умовою впорядкованості.*

Ключові слова: *запит, оптимізація, доступ, оптимізатор, сервер, система обробки даних, модуль обробки введення/виводу, аналізатор, час виконання мультизапиту, процесор*

Tkachenko O.M., Lemeshko A.V., Zamrii I.V., Kraschenko D.V., Pidmohylnyy O.O.
State University of Telecommunications, Kyiv

DYNAMIC SOLUTION FOR MULTIPLE QUESTION OPTIMIZATION

Abstract: *The problem of multiple query optimization is considered as a complete optimization problem, the input parameters for optimization are determined: queries, tasks, plans. Proposed and substantiated architecture of the system in which the DBMS with the query optimizer is used - a computer system in which one or more computers (information processing system, one or more client computers, control computer, one or more the number of database servers) are interconnected by the client network and the server network. The data processing system receives the primary request made by one of the client computers, creates one or more secondary requests and transmits them to the database servers, in the case, if necessary, performs links or data updates, as defined in the primary request, and then returns the resulting data to the computer of the client from which the initial request was made. The I / O processing module, query analyzer, query optimizer, query processing module, optimization controller, external memory device are the elements that make up the information processing module. The I / O processing module receives the request from the client computers and the control request from the control computer and responds to these requests. The query optimizer optimizes the query using the query tree generated by the query analyzer, and develops a procedure for a series of operations (query execution plan) to obtain the results of query processing. The query processing module executes the query execution plan developed by the query optimizer. The order of*

execution of elementary requests which provide the minimum time for one, two and three processors is developed. A comparison of multi-request execution time with joint and incompatible processing depending on the number of processors is presented. The minimum execution time of the query can be achieved by executing elementary queries in the appropriate order, determined by the condition of order.

Keywords: *request, optimization, access, optimizer, server, data processing system, I / O processing module, analyzer, multi-request execution time, processor*

Ткаченко О.Н., Лемешко А.В., Замрий И.В., Кращенко Д.В., Пидмогильный А.А.
Государственный университет телекоммуникаций, Киев

ДИНАМИЧЕСКОЕ РЕШЕНИЕ ПРИ МНОГОКРАТНОЙ ОПТИМИЗАЦИИ ЗАПРОСОВ

Аннотация: *Рассмотрена проблема многократной оптимизации запросов как полная проблема оптимизации, определены входные параметры для оптимизации: запросы, задачи, планы. Предложена и обоснована архитектура системы, в которой применяется СУБД с оптимизатором запросов - компьютерная система, в которой один или большее число компьютеров (система обработки информации, один или большее число компьютеров клиента, управляющий компьютер, один или более число серверов БД) соединены между собой со стороны клиента сетью и со стороны сервера сетью. Система обработки данных получает первичный запрос, сделанный одним из компьютеров клиента, создает один или большее число вторичных запросов и передает их серверам БД, в том случае, если это необходимо, выполняет ссылки или обновления данных, так как это определено в первичном запросе, а затем возвращает полученные в результате данные на компьютер клиента, с которого был сделан первоначальный запрос. Модуль обработки ввода / вывода, анализатор запросов, оптимизатор запросов, модуль обработки запросов, контроллер оптимизации, устройство внешней памяти являются элементами, составляющими модуля обработки информации. Модуль обработки ввода / вывода принимает заявку о запросе от компьютеров клиента и заявку об управлении от управляющего компьютера и отвечает на эти заявки. Оптимизатор запросов оптимизирует запрос, используя дерево запроса, сгенерированное анализатором запросов, и разрабатывает процедуру для серии операций (план выполнения запроса), чтобы получить результаты обработки запроса. Модуль обработки запросов выполняет план выполнения запроса, разработанный оптимизатором запроса. Разработан порядок выполнения элементарных запросов, которые обеспечивают минимальное время для одного, двух и трех процессоров. Представлено сравнение времени выполнения мультизапроса при совместной и независимой обработке в зависимости от числа процессоров. Минимальное время выполнения запроса может быть достигнуто при выполнении элементарных запросов в соответствующем порядке, определенном условием упорядоченности.*

Ключевые слова: *запрос, оптимизация, доступ, оптимизатор, сервер, система обработки данных, модуль обработки ввода / вывода, анализатор, время выполнения мультизапроса, процессор*

1. Вступ

Існує ряд можливих альтернатив плану обробки запитів.

У задач є деяка вартість, пов'язана з ними, яка відображає вартість центрального процесора, вартість введення / виведення, необхідних при їх обробці. Вартість плану доступу - вартість обробки її складових завдань.

Якщо є ряд запитів, то глобальний план доступу відповідає плану, який забезпечує спосіб обчислити результати всіх запитів. Глобальний план доступу може бути створений, вибираючи один план щодо кожного запиту і потім об'єднуючи їх разом. Об'єднання в місцевому масштабі оптимальних планів не обов'язково дає глобально оптимальний план.

2. Аналіз літературних даних і постановка проблеми

Проблемі оптимізації виконання мультизапиту при зверненні до бази даних присвячена велика кількість публікацій. В якості критерію оптимізації мультизапитів зазвичай використовують час виконання запиту, при цьому підрозділяють час, що витрачається на роботу з даними, що знаходяться в оперативній, буферній та зовнішній пам'яті. Додатковими умовами є обсяг пам'яті, число процесорів і ін., які часто задають у вигляді вартісних обмежень.

В роботах [1-3] сформульована проблема багаторазової оптимізація запитів (MQO) як повна проблема оптимізації.

Основним критерієм при визначенні плану реалізації запиту є час виконання мультизапиту, який залежить від порядку виконання елементарних запитів, його складових, і від часу перевірки в рядку і ймовірності успіху в рядку [4].

Основною роботою по оптмізації запитів є робота [5]. Ідеї, викладені в цій статті, лягли в основу більшості досліджень оптимізації.

Однак зі зростаючою важливістю оперативної аналітичної обробки даних техніка більш складних оптимізацій запитів стала вирішальною. Для того щоб бути ефективними, оптимізатори повинні адаптуватися до нових операторів, зміни в методах оцінки вартості і т.д.

3. Мета і задачі дослідження

Метою дослідження є вдосконалення і розробка способу багаторазової оптимізації запитів.

Для досягнення поставленої мети вирішено такі завдання:

– проаналізовано проблему багаторазової оптимізації запитів, визначено вхідні параметри для оптимізації;

– запропонована і обґрунтована архітектура системи, в якій застосовується СУБД з оптимізатором запитів;

– розроблений порядок виконання елементарних запитів, які забезпечують мінімальний час.

4.1 Багаторазова оптимізація запитів

Багаторазова оптимізація запитів (MQO) намагається зменшити вартість виконання групи запитів, виконуючи загальні завдання тільки один раз, тоді як традиційна оптимізація запитів розглядає єдиний запит за один раз. Проблема MQO була сформульована в [1, 2, 3] як повна проблема оптимізації.

Нехай вхідні параметри для MQO задані як:

- Q запити: $q_1, q_2, \ldots, q_q$.
- T задачі: $t_1, t_2, \ldots, t_T$, $(t_i) = c_i$, *та* $C = \sum c_i)$
- P плани: $P = \sum P_i$, де план P_i для запиту q_i.

Приклад на рис.1 пояснює проблему MQO.

Параметри цього прикладу наступні:

- Кількість запитів, $Q = 3$.
- Кількість завдань, $T = 5$, і витрати завдань: $t_1(c_1 = 3)$, $t_2(c_2 = 3)$, $t_3(c_3 = 1)$, $t_4(c_4 = 1)$, $t_5(c_5 = 2)$.
- Кількість планів, $P = 7$.
- Кількість планів першого запиту $P_1 = 2$, та $P_{11} = P_1 = \{t_1, t_3, t_4\}$, $P_{12} = P_2 = \{t_1, t_2\}$.
- Кількість планів другого запиту, $P_2 = 3$, *та* $P_{21} = P_3 = \{t_2, t_4\}$, $P_{22} = P_4 = \{t_5\}$, $P_{23} = P_5 = \{t_1, t_2, t_3\}$.
- Кількість планів третього запиту, $P_3 = 2$, *та* $P_{31} = P_6 = \{t_1, t_3\}$, $P_{32} = P_7 = \{t_4\}$.
- Загальна вартість запитів, $C = c_1 + c_2 + c_3 + c_4 + c_5 = 10$.
- Витрати планів – $C_1 = 5, C_2 = 6, C_3 = 4, C_4 = 2, C_5 = 7, C_6 = 4, C_7 = 1$.

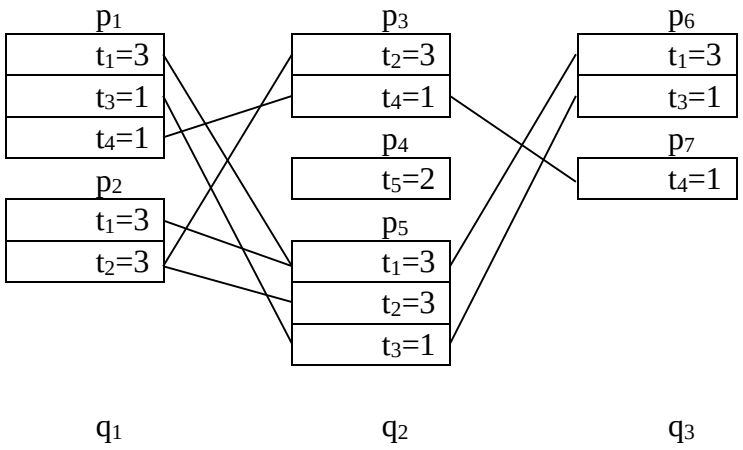

Рис.1. Проблема MQO з 3 запитами, 7 планами і 5 завданнями

4.2. Типова архітектура системи, в якій застосовується СУБД з оптимізатором запитів

Розглянемо типову архітектуру системи, в якій застосовуються СУБД з оптимізатором запитів:

Даний варіант в цілому являє собою комп'ютерну систему, в якої один або більше число комп'ютерів (система обробки інформації 100, один або більше число комп'ютерів клієнта 101, 101a і т.д., керуючий комп'ютер 102, один або більше число серверів БД 105, 105a) з'єднані між собою з боку клієнта мережею 103 і з боку сервера мережею 104.

Мережа клієнта 103 і мережа сервера 104 можуть бути локальними мережами. Ці мережі можуть також бути мережевим з'єднанням між комп'ютерами і/або мережевим з'єднанням між елементами процесора в паралельному комп'ютері. Більше, мережа з боку клієнта 103 і мережа з боку сервера 104 можуть бути однією і тієї ж мережею.

Система обробки даних 100, комп'ютери клієнта 101, 101a, керуючий комп'ютер 102, сервери БД 105, 105a, можуть бути комп'ютерами будь-якого типу, включаючи персональний комп'ютер, робочу станцію, паралельний комп'ютер, комп'ютер мейнфрейм і невеликий переносний ноутбук. На рис. 2 кожен елемент системи обробки даних 100 - комп'ютер клієнта 101, 101a, комп'ютер управління 102, сервери БД 105, 105a показані як незалежні окремі комп'ютери. Однак будь-яка комбінація цих комп'ютерів може також являти собою один комп'ютер. Системи обробки даних 100, комп'ютера клієнта 101, 101a, комп'ютера управління 102, серверів БД 105, 105a, мережі збоку клієнта 103, мережі з боку сервера 104 і їх конфігурація на рис. 2 показані в якості одного з прикладів, і таким чином сфера застосування даного винаходу аж ніяк не обмежується цією конфігурацією.

Додатки 120, 120a, які є програмами для обробки інформації користувачем, функціонують на комп'ютерах клієнта 102, 102a. Додаток 102 робить запит (наприклад, запит, сформульований на SQL), щоб зробити посилання або оновити БД, якщо це необхідно.

Сервери БД 105, 105a та інші мають одну або кілька БД. СУБД 122, яка представляє собою програму, призначену для того, щоб робити посилання або оновлювати БД у відповідь на запит від інших програм, працює на серверах БД 105, 105a. У багатьох випадках СУБД 122 містить одну або більше число БД, як своїх цільових даних для управління в пристроях зовнішньої пам'яті 106, 106a. БД може являти собою реляційну БД, яка складається з однієї або більшого числа таблиць, що мають одну або більше число стовпців, а може також являти собою БД інших типів (БД мережевого типу, ієрархічну БД, об'єктно-орієнтовану БД або сховище об'єктів) або системи файлів.

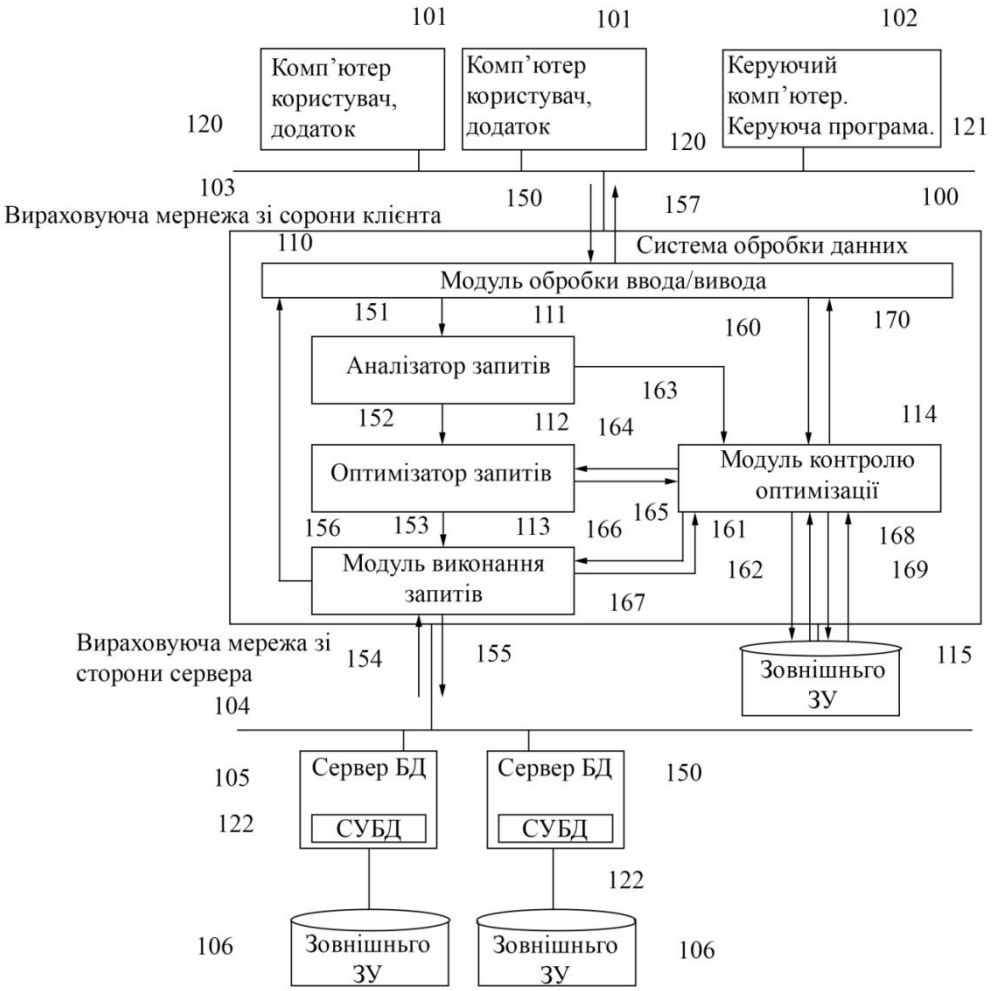

Рис.2. Типова архітектура системи, що використовує оптимізатор запитів

Система обробки даних 100 отримує первинний запит, зроблений одним з комп'ютерів клієнта 101, 101а, створює один або більше число вторинних запитів і передає їх серверам БД 105, 105а, в тому випадку, якщо це необхідно, виконує посилання або оновлення даних, так як це визначено в первинному запиті, а потім повертає отримані в результаті дані на комп'ютер клієнта,з якого був зроблений первинний запит.

Керуючий комп'ютер 102 використовує управляє додаток 121. Керуючий додаток 121 - це програма управління системою обробки інформації 100 або всієї системи.

Модуль обробки введення/виводу 110, аналізатор запитів 111, оптимізатор запитів 112, модуль обробки запитів 113, контролер оптимізації 114, пристрій зовнішньої пам'яті 115 є елементами, складовими модуля обробки інформації 100. Модуль обробки введення/виводу 110 приймає заявку про запит від комп'ютерів клієнта 101, 101а та заявку про управління від управляючого комп'ютера 102 і відповідає на ці заявки.

Аналізатор запитів 111 виробляє лексичний аналіз, синтаксичний аналіз, і семантичний аналіз заявки про запит, отриманої модулем обробки введення/виведення 110, виробляє стандартні перетворення умов запиту, в випадку, якщо це необхідно, а потім створює на основі запиту дерево запиту.

Оптимізатор запитів 112 оптимізує запит, використовуючи дерево запиту, згенероване аналізатором запитів 111, і розробляє процедуру для серії операцій (план виконання запиту), щоб отримати результати обробки запиту. У разі реляційної БД серія операцій в плані виконання запиту включає процес вибірки, проекцію, з'єднання, угруповання і сортування. План виконання запиту - це структура даних, яка описує, як застосовувати ці операції щодо цільової БД (тобто описуються цільова БД, цільової сервер

БД 105 і порядок обробки запиту). Модуль обробки запитів 113 виконує план виконання запиту, розроблений оптимізатором запиту 112.

Модуль виконання запитів 113 може зажадати шляхом посилки запиту на сервери БД 105, 105а, щоб сервери БД 105,105а частково або повністю виконали цю серію операцій. модуль обробки запитів 113 може також сам частково або повністю виконати цю серію операцій шляхом обробки даних отриманих від серверів БД 105, 105а.

Контролер оптимізації 114 інтерпретує заявку про управління, отриману модулем обробки введення/виводу 110, і зберігає визначення класифікації запиту і напрямок операцій з обробки запиту, включені в заявку про управління в пристрої зовнішньої пам'яті 115, в разі необхідності. Більш того, контролер оптимізації 114 отримує в своє розпорядження запит (і додаткову інформацію про запит), який раніше вчинив на аналізатор запитів 111, отримує направлення операцій з обробки запиту, яке відповідає цьому запиту згідно з визначенням класифікації запиту, а потім видає направлення операцій з обробки запитів на оптимізатор запитів 112 і модуль обробки запитів 113.

Розглянемо приклад.

Нехай елементарні запити $ЕЗ_1$, $ЕЗ_2$, $ЕЗ_3$, $ЕЗ_4$, $ЕЗ_5$, $ЕЗ_6$

утворюють 3 мультизапити $З_1, З_2, З_3$:

$З_1 = > ЕЗ_1 , ЕЗ_4 , ЕЗ_5$

$З_2 = > ЕЗ_2 , ЕЗ_4 , ЕЗ_5$

$З_3 = > ЕЗ_5 , ЕЗ_3$

При несумісній обробці запитів

$T_{несум} = (З_1 + З_2 + З_3) = T_{несум1} + T_{несум2} + T_{несум3}$

$t_1 = 1\ ms; t_2 = 5.2\ ms; t_3 = 1.2\ ms; t_4 = 1.3\ ms; t_5 = 1.4\ ms; t_6 = 1.5\ ms$

Маємо такі результати в залежності від порядку виконання елементарних запитів

Для $З_1 = > ЕЗ_1 , ЕЗ_4 , ЕЗ_5$

$T_{несум1,1} =(10t_1 + 2t_4 + 2t_5) = (10*1+2*1.3+2*1.4) = 15.4\ ms$

$T_{несум1,2} =(10t_1 + 2t_5 + 2t_4) = (10*1+2*1.4+2*1.3) = 15.4\ ms$

$T_{несум1,3} =(10t_4 + 6t_5 + 2t_1) = (10*1.3+6*1.4+2*1) = 23.4\ ms$

$T_{несум1,4} =(10t_4 + 6t_1 + 2t_5) = (10*1.3+2*1+6*1.4) = 23.4\ ms$

$T_{несум1,5} =(10t_5 + 3t_1 + 2t_4) = (10*1.4+2*1+6*1.3) = 23.8\ ms$

$T_{несум1,6} =(10t_5 + 3t_4 + 2t_1) = (10*1.4+6*1.3+2*1) = 23.8\ ms$

Для $З_2 = > ЕЗ_2 , ЕЗ_4 , ЕЗ_5$

$T_{несум2,1} =(10t_2 + 6t_4 + 3t_5) = (10*5.2+6*1.3+3*1.4) = 64\ ms$

$T_{несум2,2} =(10t_2 + 6t_5 + 3t_4) = (10*5.2+6*1.4+3*1.3) = 64.3\ ms$

$T_{несум2,3} =(10t_4 + 6t_5 + 3t_2) = (10*1.3+6*1.4+3*5.2) = 37\ ms$

$T_{несум2,4} =(10t_4 + 6t_2 + 3t_5) = (10*1.3+6*5.2+3*1.4) = 48.4\ ms$

$T_{несум2,5} =(10t_5 + 3t_2 + 3t_4) = (10*1.4+3*5.2+3*1.3) = 33.5\ ms$

$T_{несум2,6} =(10t_5 + 3t_4 + 3t_2) = (10*1.4+3*1.3+3*5.2) = 33.5\ ms$

Для $З_3 = > ЕЗ_5 , ЕЗ_3$

$T_{несум3,1} =(10t_5 + 3t_3) = (10*1.4+3*1.2) = 17.6\ ms$

$T_{несум3,2} =(10t_3 + 3t_5) = (10*1.2+3*1.4) = 16.2\ ms$

Використовуючи порядок виконання елементарних запитів, які забезпечують мінімальний час, отримуємо:

для одного процесора

$T_{несум} = (З_1 + З_2 + З_3) = T_{несум1} + T_{несум2} + T_{несум3} = (15.4+33.5+16.2)=65.1\ ms$

для двох процесорів при розподілі елементарних запитів по процесорам в порядку:

N процесора	Порядок обробки *ЕЗ*
1	$(З_1, З_2)$
2	$З_2$

$T_{несум2,1} = (З_1 + З_2)=(T_{несум1} + T_{несум3})=(15,4+16,2)=31,6\ ms$

$T_{несум2,2} = З_2 = 33,5$ *ms*
$T_{несум2} = max(T_{несум2,1}\ T_{несум2,2}) = 33.5$ *ms*
для трьох процесорів при розподілі елементарних запитів по процесорам в порядку:

N процесора	Результат *ЕЗ*
1	65.1 ms
2	33.5ms
3	33.5ms

$T_{несум3,1} = З_1 = 15.4$ *ms*
$T_{несум3,2} = З_2 = 33.5$ *ms*
$T_{несум3,3} = З_3 = 16.2$ *ms*
$T_{несум3} = max(T_{несум3,1}\ T_{несум3,2}\ T_{несум3,3}) = 33.5$ *ms*

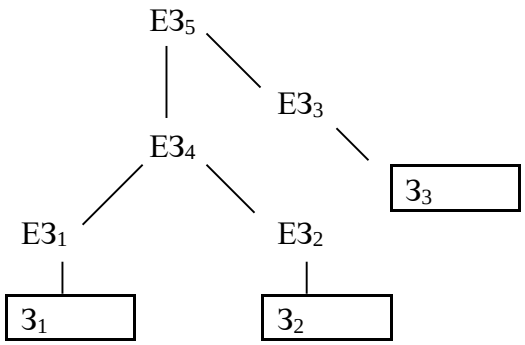

Рис.3. Порядок обробки елементарних запитів

При спільній обробці елементарних запитів в порядку, див. Рис. 3, маємо наступні результати
для одного процесора
$T_{сум1} = (10t_5 + 3t_4 + 2t_1 + 2t_2 + 3t_3) = (10*1.4+3*1.3+2+2*5.2+3*1.2) = 33.9$ *ms*
для двох процесорів при розподілі елементарних запитів по процесорам в порядку:

N процесора	Порядок обробки *ЕЗ*
1	*5, 4, 1*
2	*3, 2*

$T_{сум2.1} = (10t_5 + 3t_4 + 2t_1) = (10*1.4+3*1.3+2) = 19.9$ *ms*
$T_{сум2.2} = (3t_3 + 2t_2) = (3*1.2+2*5.2) = 14$ *ms*
$T_{сум2} = max(T_{сум2.1}\ T_{сум2.2}) = max(19.9, 14) = 19.9$ *ms*
для трьох процесорів при розподілі елементарних запитів по процесорам в порядку:

N процесора	Порядок обробки *ЕЗ*
1	*5, 4, 1*
2	*2, 3*
3	-

$T_{сум3.1} = (10t_5 + 3t_4 + 2t_1) = (10*1.4+3*1.3+2) = 19.9$ *ms*
$T_{сум3.2} = (3t_3 + 2t_2) = (3*1.2+2*5.2) = 14$ *ms*
$T_{сум3.3} = max$
$T_{сум3} = max(T_{сум2.1}\ T_{сум2.2}\ T_{сум3.3}) = max(19.9, 14) = 19.9$ *ms*

В результаті маємо час виконання мультизапиту в залежності від числа процесорів:

Число процесорів	Час виконання
1	33.9 ms
2	19.9 ms
3	19.9 ms

Порівняння часу виконання мультизапиту при спільній і несумісній обробці в залежності від числа процесорів приведено в таблиці і на рис. 4.

Число процесорів	Час виконання(сум)	Час виконання(несум)
1	33.9 ms	65.1 ms
2	19.9 ms	33.5 ms
3	19.9 ms	33.5 ms

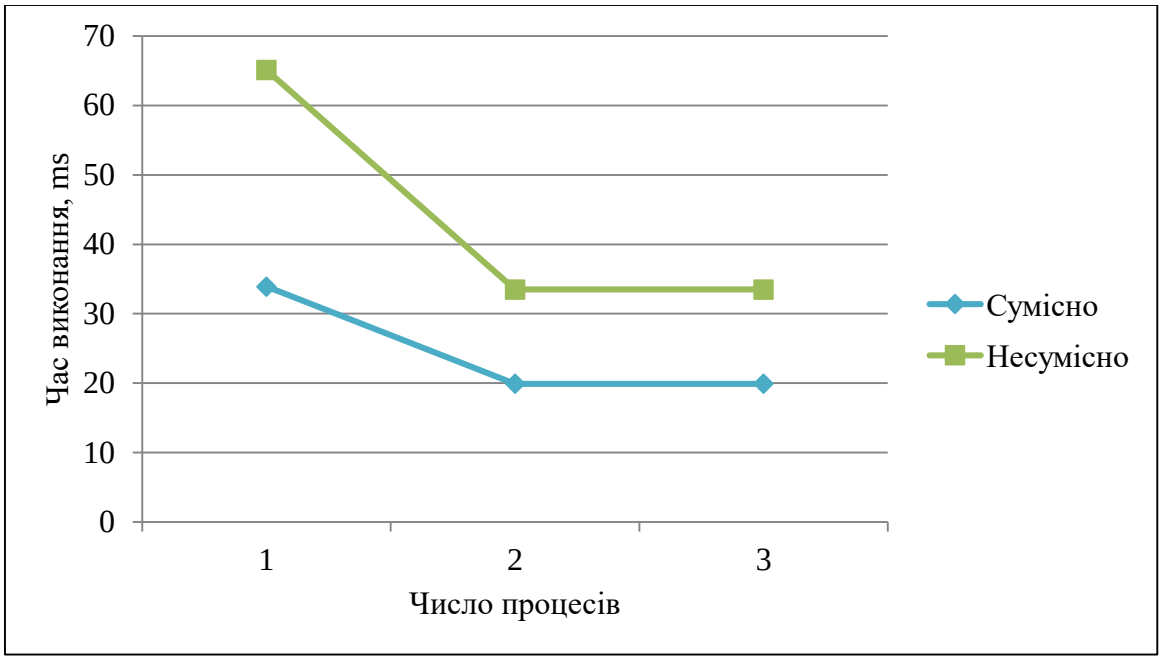

Рис.4. Час виконання оптимізація мультизапитів

Отже, мінімальний час виконання запиту може бути досягнуто при виконанні елементарних запитів у відповідному порядку, визначеному умовою впорядкованості.

Висновки

Розглянуто проблему багаторазової оптимізації запитів як повну проблему оптимізації, визначено вхідні параметри для оптимізації: запити, задачі, плани.

Запропонована і обґрунтована архітектура системи, в якій застосовується СУБД з оптимізатором запитів - комп'ютерна система, в якої один або більше число комп'ютерів (система обробки інформації, один або більше число комп'ютерів клієнта, керуючий комп'ютер, один або більше число серверів БД) з'єднані між собою з боку клієнта мережею і з боку сервера мережею.

Розроблено порядок виконання елементарних запитів, які забезпечують мінімальний час для одного, двох та трьох процесорів. Представлено порівняння часу виконання мультизапиту при спільній і несумісній обробці в залежності від числа процесорів. Мінімальний час виконання запиту може бути досягнуто при виконанні елементарних запитів у відповідному порядку, визначеному умовою впорядкованості.

Список використаної літератури

1. A. Cosar, J. Srivastava, S. Shekhar, On the multiple pattern multiple object (MPMO) match problem, in: International Conference on Management of Data, India, 1991.

2. K. Shim, T. Sellis, D. Nau, Improvements on a heuristic algorithm for multiple-query optimization, Data Knowl. Eng. 12 (2) (1994) 197-222.

3. T. Sellis, Multiple query optimization, ACM Transactions on Database Systems 13 (1) (1988) 23- 52.

4. Брехов О.М. Аналитическая оценка оптимальной обработки запросов // Успехи современной радиоэлектроники. 2012. Т. 12. №7. С. 37-45.

5. Selinger, P., Astrahan, M. M., Chamberl In, D. D., Lorie, R. A., And Price, T. G. Access path selection in a relational database management system. In ACM SIGMOD Intl. Conf. on Management of Data (1979), pp. 23-34.

References

1. A. Cosar, J. Srivastava, S. Shekhar, On the multiple pattern multiple object (MPMO) match problem, in: International Conference on Management of Data, India, 1991.

2. K. Shim, T. Sellis, D. Nau, Improvements on a heuristic algorithm for multiple-query optimization, Data Knowl. Eng. 12 (2) (1994) 197-222.

3. T. Sellis, Multiple query optimization, ACM Transactions on Database Systems 13 (1) (1988) 23- 52.

4. Brekhov O.M. Analytical evaluation of optimal query processing // Success of modern radioelectronics. 2012. T. 12.No. 7. S. 37-45.

5. Selinger, P., Astrahan, M. M., Chamberl In, D. D., Lorie, R. A., And Price, T. G. Access path selection in a relational database management system. In ACM SIGMOD Intl. Conf. on Management of Data (1979), pp. 23-34.