

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет) - МАИ

На правах рукописи

Тан Хлаинг Мьинт

«Оптимизация обработки вложенных запросов в многопроцессорной базе данных»

Специальность 05.13.11

«Математическое и программное обеспечение вычислительных машин, комплексов и
компьютерных сетей»

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель:

д.т.н., профессор,
заслуженный деятель науки РФ

Брехов Олег Михайлович

Москва, 2014 г.

Содержание

Введение

| | |
|--|-------------------------------------|
| Содержание | 2 |
| Введение | 4 |
| 1. Вложенные запросы (подзапросы) в MySQL Server, T-SQL или SQL Oracle | 8 |
| 1.1. Теоретические основы | 8 |
| 1.1.1. Подзапрос во фразе WHERE | 10 |
| 1.1.2. Подзапрос в условии сравнения групп | 12 |
| 1.1.3. Подзапрос в условии проверки вхождения элемента во множество | 14 |
| 1.1.4. Подзапрос в условии EXISTS | 15 |
| 1.1.5. Связанные подзапросы | 15 |
| 1.1.6. Связанные подзапросы во фразе WHERE | 16 |
| 1.1.7. Простые и связанные подзапросы во фразе HAVING | 16 |
| 1.1.8. Простые подзапросы во фразе FROM | 17 |
| 1.1.9. Подзапросы во фразе SELECT | 17 |
| 1.2. Обработка оптимизации вложенных запросов | 19 |
| 1.2.1. Вложенные подзапросы | 19 |
| 1.2.2. Простые вложенные подзапросы | 19 |
| 1.2.3. Использование одной и той же таблицы во внешнем и вложенном подзапросе | 21 |
| 1.2.4. Вложенный подзапрос с оператором сравнения, отличным от IN | 22 |
| 1.2.5. Коррелированные вложенные подзапросы | 23 |
| 1.2.6. Запросы, использующие EXISTS | 24 |
| 1.2.7. Функции в подзапросе | Error! Bookmark not defined. |
| 1.3. Выполнение вложенного запроса в однопроцессорной базе данных | 26 |
| 1.3.1. Введение | 26 |
| 1.3.2. Влияние параметров запроса на время выполнения вложенного запроса | 27 |
| 1.3.2.1. Неупорядоченные таблицы с параметрами, соответствующими геометрической прогрессии | 28 |
| 1.3.2.2. Упорядоченные таблицы с параметрами, соответствующими геометрической прогрессии | 32 |
| 1.3.2.3. Неупорядоченные таблицы с параметрами, соответствующими арифметической прогрессии | 35 |

| | |
|---|----|
| 1.3.2.4. Упорядоченные таблицы с параметрами, соответствующими арифметической прогрессии | 39 |
| 1. 4. Постановка задачи | 42 |
| Выводы по главе 1 | 44 |
| Глава 2. Обоснование квазиоптимального порядка распределения элементарных запросов в многопроцессорной базе данных | 45 |
| 2.1. Введение..... | 45 |
| 2.2. Распределение номеров элементарных запросов по процессорам..... | 45 |
| 2.3. Обоснование квазиоптимального порядка распределения..... | 47 |
| 2.3.1. Обработки запроса для неупорядоченных столбцов таблицы естественный порядок распределения. Арифметическая прогрессия | 47 |
| 2.3.2. Квазиоптимальный порядок распределения. Арифметическая прогрессия | 49 |
| 2.3.2.1. Время выполнения | 49 |
| 2.3.2.2. Соотношения времени выполнения на i -м и j -м процессорах | 50 |
| 2.3.2.3. Эффективность квазиоптимального распределения | 51 |
| 2.3.3. Обработки запроса для упорядоченных столбцов таблицы естественный порядок распределения. Арифметическая прогрессия | 53 |
| 2.3.4. Квазиоптимальный порядок распределения. Арифметическая прогрессия | 54 |
| 2.3.4.1. Время выполнения | 54 |
| 2.3.4.2. Соотношения времени выполнения на i -м и j -м процессорах | 54 |
| 2.3.4.3. Эффективность квазиоптимального распределения | 55 |
| 2.3.5. Обработки запроса для неупорядоченных столбцов таблицы естественный порядок распределения. Геометрическая прогрессия | 57 |
| 2.3.6. Квазиоптимальный порядок распределения. Геометрическая прогрессия..... | 58 |
| 2.3.6.1. Время выполнения | 58 |
| 2.3.6.2. Соотношения времени выполнения на i -м и j -м процессорах | 59 |
| 2.3.6.3. Эффективность квазиоптимального распределения | 62 |
| 2.3.7. Обработки запроса для упорядоченных столбцов таблицы естественный порядок распределения. Геометрическая прогрессия..... | 64 |
| 2.3.8. Квазиоптимальный порядок распределения. Геометрическая прогрессия..... | 65 |
| 2.3.8.1. Время выполнения | 65 |

| | | |
|-------------------------|--|-----|
| 2.3.8.2. | Соотношения времени выполнения на i -м и j -м процессорах | 65 |
| 2.3.8.3. | Эффективность квазиоптимального распределения | 67 |
| Выводы по главе 2 | | 69 |
| 3. | Оптимизация числа процессоров при выполнении вложенных запросов | 71 |
| 3.1. | Введение | 71 |
| 3.2. | Квазиоптимальное распределение номеров элементарных запросов по процессорам | 71 |
| 3.3. | Минимизация времени обработки вложенного запроса в многопроцессорной базе данных | 72 |
| 3.4. | Численные результаты | 73 |
| 3.4.1. | Вложенные запросы для таблиц с параметрами, соответствующими геометрической прогрессии | 74 |
| 3.4.1.1. | Несовместное выполнение вложенных запросов | 74 |
| 3.4.1.2. | Совместное выполнение вложенного запроса | 80 |
| 3.4.2. | Вложенные запросы для таблиц с параметрами, соответствующими арифметической прогрессии | 88 |
| 3.4.2.1. | Несовместное выполнение вложенных запросов | 88 |
| 3.4.2.2. | Совместное выполнение вложенного запроса | 94 |
| Выводы по главе 3 | | 103 |
| Литература | | 105 |

Введение

Типичная БД – это организованный набор связанной информации, которая хранится в виде «записей», которые разделены на «поля» (области информации). Например, БД работников предприятия может иметь запись о каждом работнике, и каждая запись имеет поля, где указаны специфические данные в отношении работника – фамилия, домашний адрес, размер зарплаты и т.д.

Между действительной физической БД (т.е. данными хранящимися на носителе хранимой информации) и пользователями системой находится СУБД (система управления базами данных) как слой или средства программного обеспечения (ПО). СУБД позволяет пользователю не интересоваться и не заботиться о деталях аппаратного обеспечения, о

схемах, обеспечивающих использование БД. Обычно все запросы пользователей в отношении доступа к БД обрабатываются СУБД. Например, данные могут добавляться или удаляться из файлов данных, информация в файлах данных может быть восстановлена или обновлена и т.д., а пользователь при этом может и не знать о соответствующем, обеспечивающем эти операции функционировании системы.

Одновременно с этим пользователи ожидают от применения СУБД высокой производительности и надежности.

Задача оптимизации вложенных запросов в базах данных заключается в создании такой надстройки над какой-либо СУБД (например, MySQL Server, T-SQL или Oracle), которая способна последовательно выполнять действия, необходимые для выбора наиболее эффективного плана выполнения запроса, также осуществляя минимизацию накладных расходов.

Входными данными в данной задаче является некоторая формулировка запроса к БД. Но, поскольку язык SQL декларативен, эта формулировка содержит информацию о том, что мы хотим получить, но не о том, как мы хотим этого добиться. Способ, которым будет выполняться данная задача, определяет именно оптимизатор запросов. В наиболее популярных СУБД присутствуют собственные оптимизаторы запросов, однако достаточно часты случаи, когда они выдают не оптимальный план выполнения вложенного запроса. Вложенный запрос может быть улучшен вручную, но эта процедура также поддается и автоматизации, чему и посвящена данная работа.

Актуальность темы

Проблеме оптимизации выполнения вложенных запросов при обращении к базе данных посвящено большое число публикаций. В качестве критерия оптимизации вложенных запросов обычно используют время выполнения запроса, при этом подразделяют время, затрачиваемое на работу с данными, находящимися в оперативной, буферной и внешней памяти. Дополнительными условиями являются объем памяти, число процессоров и др., которые часто задают в виде стоимостных ограничений.

Проблемами создания и оценки качества ООЗ занимались ряд российских и зарубежных исследователей: Григорьев Ю.А., Кузнецов С.Д., AmolDeshpande, ZaccharyIves, VijayshankarRaman, SelingerP.G, AstrahanM.M, ChamberlinD.D и др.

В данной работе развита методика формирования плана оптимизации обработки вложенных запросов многопроцессорными базами данных, что наиболее соответствует бортовым базам данных перспективных авиационных систем, таким как базы данных для систем управления полетом.

Цель диссертационной работы

Оптимизации по времени выполнения вложенных запросов при обращении к многопроцессорной базе данных на основе упорядочивания элементарных запросов.

Методы исследования

Математическое моделирование. Компьютерное моделирование. Современная методология организации баз данных.

Научная новизна

- Разработана методика оптимизации по времени выполнения конъюнктивных вложенных запросов при обращении к многопроцессорной базе данных на основе упорядочивания элементарных запросов.
- Определены соотношения времени выполнения запроса в многопроцессорной базе данных для естественного и квазиоптимального порядка их распределения.
- Доказана эффективность квазиоптимального распределения на основе абсолютного и относительного уменьшения границ времени выполнения запросов при использовании квазиоптимального распределения вместо естественного распределения.
- Определено минимальное время выполнения вложенного запроса для упорядоченных или неупорядоченных данных таблиц при совместной обработке процессорами данных объединенного множества элементарных запросов всех таблиц, образующих вложенные запросы.
- Определено минимальное число процессоров, при котором достигается минимальное время выполнения вложенного запроса, что является важным решением для оптимизации многопроцессорных баз данных авиационно-космических систем.

Достоверность полученных в диссертационной работе результатов подтверждается:

Применяемым математическим и имитационным аппаратом. Подобием полученных результатов аналитического и имитационного моделирования. Соответствием полученных и известных результатов.

Практическая значимость

Разработан модуль формирования плана выполнения вложенных запросов и оценки времени его выполнения.

Реализация результатов работы

Результаты диссертационной работы используются в учебном процессе кафедры «Вычислительные машины и системы» Московского авиационного института (государственного технического университета) в форме информационного обеспечения блока дисциплин, а так же в лекционном курсе «Моделирование».

На защиту выносятся следующие положения:

- Методика оптимизации по времени выполнения конъюнктивных вложенных запросов при обращении к многопроцессорной базе данных на основе упорядочивания элементарных запросов.
- Соотношения времени выполнения запроса в многопроцессорной базе данных для естественного и квазиоптимального порядка их распределения.
- Утверждение эффективности квазиоптимального распределения на основе абсолютного и относительного уменьшения границ времени выполнения запросов при использовании квазиоптимального распределения вместо естественного распределения.
- Соотношения для минимального времени выполнения вложенного запроса для упорядоченных или неупорядоченных данных таблиц при совместной обработке процессорами объединенного множества элементарных запросов всех таблиц, образующих вложенные запросы.

Апробация работы

Основные положения и результаты диссертационного исследования докладывались автором и обсуждались: на 11-я Международная конференция МАИ «Авиация и космонавтика-2012», 13-15 ноября 2012г года МАИ, Москва, 13-я Международная конференция МАИ «Авиация и космонавтика-2013», 12–15 ноября 2013 года, Московская молодежная конференция «Инновации в авиации и космонавтике», Москва. 22–24 апреля 2014года.

Публикации

Основные материалы диссертационной работы опубликованы в 3 печатных работах из перечня ВАК.

Структура и объем диссертации

Диссертация состоит из введения, четырёх глав, заключения, библиографического списка из 154 наименований. Работа изложена на 114 страницах, содержит 37 таблиц и 32 рисунка.

1. Вложенные запросы (подзапросы) в MySQL Server, T-SQL или SQL Oracle

1.1. Теоретические основы

Запрос – это операция, которая позволяет отыскивать данные из одной или несколько таблиц. При наличии вложенных запросов запрос верхнего уровня называется предложением SELECT, а запрос, вложенный в предложение SELECT называется подзапросом. Таким образом, *подзапрос* (вложенный запрос) – это запрос, результат которого передается в качестве аргумента в другой запрос. Подзапросы позволяют связывать в единое целое несколько запросов.

Подзапросы используются для:

Определения множества строк, который должны быть вставлены в целевую таблицу в предложениях INSERT или CREATE TABLE;

Определения одного или более значений, присваиваемых существующим строка в предложении UPDATE;

Получения значений для фраз WHERE, HAVING или START WITH в предложениях SELECT, UPDATE, и DELETE;

Определения значений указанного столбца в списке INSERT ... VALUES;

Определения таблицы, которая используется соответствующим запросом.

Это производится путем размещения подзапроса во фразе FROM соответствующего запроса как если бы это было именем таблицы. Вы можете также использовать таким образом подзапросы вместо таблиц в предложениях INSERT, UPDATE и DELETE.

Используемые таким образом подзапросы могут использовать переменные связывания (correlation variables), однако только такие, которые определены только в самом подзапросе, ссылки на внешние переменные не допустимы. Внешние ссылки (подзапросы с левой корреляцией - left-correlated subqueries) допустимы только во фразе FROM предложения SELECT .

Подзапрос дает ответ на содержательные запросы, имеющие сложную структуру. Например, для определения, кто работает на кафедре Иванова, вы сначала используете подзапрос для определения кафедры, на которой работает Иванов, а затем отвечаете на основной запрос путем формулировки предложения SELECT.

Подзапрос может содержать другие подзапросы. Oracle не ограничивает глубину вложенности подзапросов.

Если таблица в подзапросе имеет такое же имя, что и таблица внешнего запроса, то для ссылки на столбцы внешнего запроса их необходимо уточнять именем таблицы или алиасом таблицы. Чтобы ваши запросы было легче воспринимать, всегда квалифицируйте столбцы в подзапросе именем или алиасом таблицы.

MySQL Server, T-SQL, SQL Oracle или выполняет **корреляционные (связанный) подзапрос**, когда подзапрос ссылается на столбец таблицы внешнего запроса. Связанный подзапрос вычисляется для каждой строки, обрабатываемой внешним предложением. Внешним предложением может быть SELECT, UPDATE или DELETE.

Связанный подзапрос дает ответы на такие содержательные запросы, ответы которых требуют вычисления подзапросов для каждой строки внешнего запроса. Например, связанный подзапрос используется для определения преподавателей, которые

зарабатывают больше, чем средняя зарплата по кафедре. В этом случае связанный подзапрос для каждого преподавателя вычисляет среднюю зарплату на его кафедре.

Обсудим использование подзапросов в предложении WHERE.

1.1.1. Подзапрос во фразе WHERE

Подзапрос в простом условии сравнения

Синтаксис:

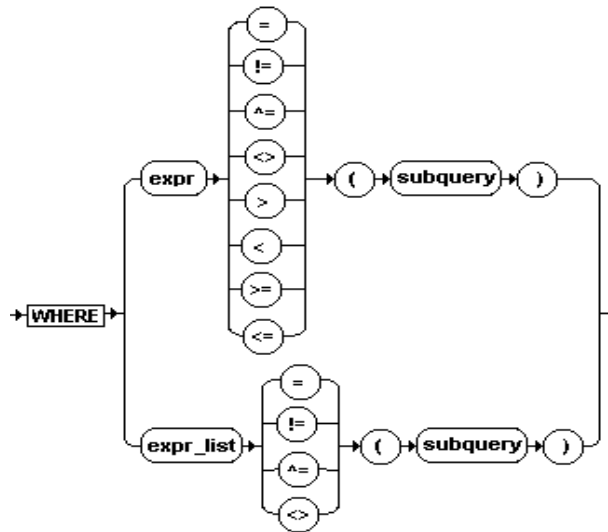


Рис.1.1. Подзапрос во фразе WHERE.

Описание:

При использовании простых условий сравнения с подзапросом во фразе WHERE применяются следующие правила:

Подзапрос должен возвращать единственную строку.

Если левая часть равна *expr*, то подзапрос должен возвращать единственную строку с единственным значением с типом, совместимым с типом *expr*.

Если левая часть является списком выражений (*expr_list*), то подзапрос должен возвращать единственную строку со списком значений, который соответствует по количеству и типу значениям из *expr_list*. В этом случае оператор сравнения дает TRUE, если каждое значение в *expr_list* равно (в случае =) или не равно (в случае !=, ^=, <>) каждому значению, возвращаемому подзапросом.

Примеры:

1. Выбрать кафедры, которые располагаются в том же корпусе, что факультет информатики:

```
SELECT Name
FROM DEPARTMENT
WHERE Building = (SELECT Building
                  FROM FACULTY
                  WHERE UPPER(Name) = 'INFORMATICS');
```

2. Выбрать факультеты, чьи фонды меньше фонда кафедры CAD:

```
SELECT Name
FROM FACULTY
WHERE Fund < (SELECT Fund
              FROM DEPARTMENT
              WHERE UPPER(Name) = 'CAD');
```

3. Выбрать преподавателей, у которых salary + commission превышает более чем на 100 половину salary + commission преподавателя Bill:

```
SELECT Name
FROM TEACHER
WHERE Salary + Commission + 100 > (SELECT (Salary + Commission) / 2
                                   FROM TEACHER
                                   WHERE UPPER(Name) = 'BILL');
```

4. Выбрать преподавателей, которые работают на той же кафедре, что и Bill и занимают ту же должность, что и Bill:

```
SELECT Name
FROM TEACHER
WHERE (DepNo, Post) = (SELECT DepNo, Post
                      FROM TEACHER
                      WHERE UPPER(Name) = 'BILL');
```

1.1.2. Подзапрос в условии сравнения групп

Синтаксис:

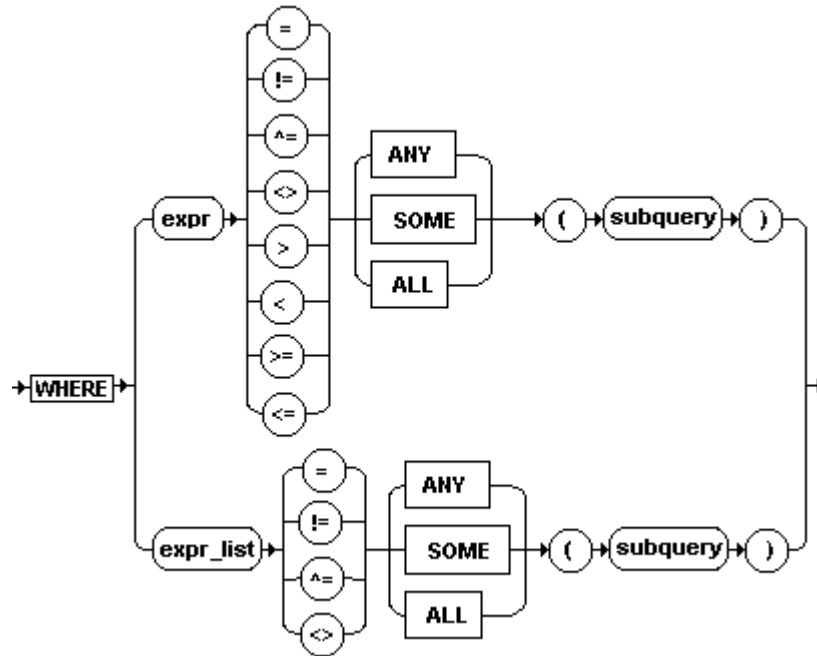


Рис. 1.2. Подзапрос в условии сравнения групп

Описание:

При использовании условий сравнения групп с подзапросом во фразе WHERE применяются следующие правила:

Подзапрос может возвращать ноль или более строк.

Если левая часть равна *expr*, то подзапрос должен возвращать строки с единственным значением, которые совместимы по типу с *expr*.

Если левая часть равна *expr_list*, то подзапрос должен возвращать строки со списком значений, который соответствует по количеству и типу с *expr_list*.

ANY и SOME эквивалентны и сравнивают значение слева с каждым значением списка справа, возвращаемого подзапросом. Подзапрос может вернуть ноль или более строк. Условие равно TRUE, если по крайней мере одна строка подзапроса удовлетворяет условию (соответствует оператору сравнения) по отношению к значению (списку

значений) определенному левым операндом, в противном получаем FALSE. Если подзапрос не возвращает строк, то получаем FALSE.

ALL сравнивают значение слева с каждым значением списка справа, возвращаемого подзапросом. Дает TRUE, если ВСЕ строки, возвращаемые подзапросом, удовлетворяют условию (соответствуют оператору сравнения) по отношению к значению (списку значений) определенному левым операндом, в противном получаем FALSE. Если подзапрос не возвращает строк, то получаем TRUE

Примеры:

1. Выдать кафедры, фонд которых больше фонда по крайней мере одного из факультетов:

```
SELECT Name
FROM DEPARTMENT
WHERE Fund > ANY (SELECT Fund FROM FACULTY);
```

ANY, ALL и агрегатные функции. Обратите внимание, что <ANY эквивалентно утверждению: “левое значение меньше, чем максимальное значение из множества, задаваемого правым операндом”, а оператор >ANY эквивалентен следующему утверждению “левое значение больше, чем минимальное значение из множества, задаваемого правым операндом”. Поэтому операторы ANY могут быть выражены через функции MAX и MIN в подзапросе. В свою очередь, <ALL (>ALL) эквивалентно утверждению: «левое значение меньше (больше), чем минимальное (максимальное) значение из множества, задаваемого правым операндом ».

2. Выдать кафедры, фонд которых больше фонда по крайней мере одного из факультетов:

```
SELECT Name
FROM DEPARTMENT
WHERE Fund > ANY (SELECT Fund FROM FACULTY);
```

```
SELECT Name
FROM DEPARTMENT
WHERE Fund > (SELECT MIN(Fund) FROM FACULTY);
```

3. Выдать группы, которые имеют рейтинг больше, чем рейтинг всех групп пятого курса кафедры “DBMS”:

```
SELECT Num
```

```

FROM SGROUP
WHERE Rating > ALL (SELECT Rating
                    FROM SGROUP, DEPARTMENT
                    WHERE SGROUP.DepNo = DEPARTMENT.DepNo AND
                        UPPER(DEPARTMENT.Name) = 'DBMS' AND
                        SGROUP.Course = 5);

```

1.1.3. Подзапрос в условии проверки вхождения элемента во множество

Синтаксис:

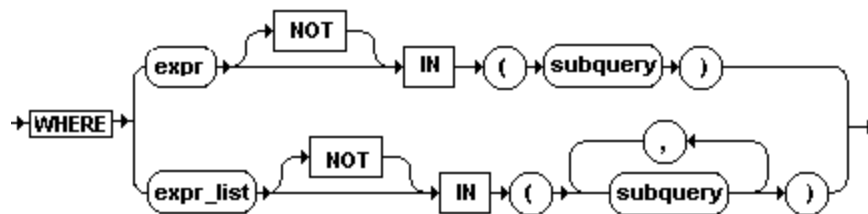


Рис.1.3. Подзапрос в условии проверки вхождения элемента во множество

Описание:

Это условие в таком синтаксисе проверяет вхождение элемента (списка элементов) во множество (множество списков), создаваемое подзапросом.

Пример:

1. Выбрать преподавателей, которые имеют лекции по крайней мере одному такому предмету, по которым читает лекции преподаватель Bill:

```

SELECT Name
FROM TEACHER T, LECTURE L
WHERE T.TchNo = L.TchNo AND
      SbjNo IN (SELECT SbjNo
              FROM TEACHER TCH, LECTURE LEC

```

```
WHERE TCH.TchNo = LEC.TchNo AND UPPER(TCH.Name) =  
'BILL');
```

1.1.4. Подзапрос в условии EXISTS

Синтаксис:

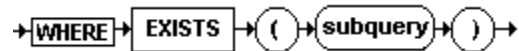


Рис.1.4. подзапрос в условии EXISTS

Описание:

Дает TRUE, если подзапрос возвращает по крайней мере одну строку.

Так как EXISTS обычно используется в связанных подзапросах, мы его обсудим подробнее позже.

1.1.5. Связанные подзапросы

Для того, чтобы связать подзапрос с внешним запросом (предложением), необходимо в подзапросе была ссылка на столбец внешнего запроса. Подзапрос вычисляется для каждой строки, обрабатываемой внешним запросом (предложением). В качестве внешнего предложения могут выступать SELECT, UPDATE или DELETE.

Следующие примеры дают общий синтаксис использования связанных подзапросов:

```
SELECT select_list
```

```
FROM table1 t_alias1
```

```
WHERE expr operator
```

```
    (SELECT column_list
```

```
    FROM table2 t_alias2
```

```
    WHERE t_alias1.column operator t_alias2.column);
```

```
UPDATE table1 t_alias1
```

```
SET column =
```

```
    (SELECT expr
```

```
    FROM table2 t_alias2
```

```
    WHERE t_alias1.column = t_alias2.column);
```

```
DELETE FROM table1 t_alias1
```

```
WHERE column operator
```

```
(SELECT expr
```

```
FROM table2 t_alias2
```

```
WHERE t_alias1.column = t_alias2.column);
```

Далее мы обсудим использование связанных подзапросов во фразе WHERE предложения SELECT.

1.1.6. Связанные подзапросы во фразе WHERE

Примеры:

1. Выдать преподавателей, которые имеют по крайней мере одну лекцию:

```
SELECT Name
```

```
FROM TEACHER
```

```
WHERE EXISTS (SELECT *
```

```
FROM LECTURE
```

```
WHERE LECTURE.TchNo = TEACHER.TchNo);
```

Здесь в условии LECTURE.TchNo = TEACHER.TchNo подзапроса мы ссылаемся на внешний запрос. Поэтому подзапрос является связанным.

2. Выдать преподавателей, которые не имеют ни одной лекции:

```
SELECT Name
```

```
FROM TEACHER
```

```
WHERE NOT EXISTS (SELECT *
```

```
FROM LECTURE
```

```
WHERE LECTURE.TchNo = TEACHER.TchNo);
```

1.1.7. Простые и связанные подзапросы во фразе HAVING

Использование простые и связанные подзапросы во фразе HAVING.

Если используется связанный подзапрос в фразе HAVING, то в подзапросе можно ссылаться на те столбцы внешнего запроса, которые могут использоваться в фразе HAVING (обычно это столбцы, по которым производится группирование).

Примеры:

1. Перечислить факультеты, у которых сумма фондов финансирования всех их кафедр превышает более чем на 20000 фонд финансирования той кафедры факультета, которая имеет максимальный фонд.

```
SELECT F1.Name
FROM FACULTY F1, DEPARTMENT D1
WHERE F1.FacNo = D1.FacNo
GROUP BY F1.Name
HAVING SUM(D1.Fund) > (SELECT 200000 + MAX(D2.Fund)
FROM FACULTY F2, DEPARTMENT D2
WHERE F2.FacNo = D2.FacNo AND F1.Name = F2.Name);
```

1.1.8. Простые подзапросы во фразе FROM

Фраза FROM может содержать не только список имен таблиц, но и подзапросы. Для ссылки на такие таблицы-подзапросы следует приписать подзапросу алиас.

Имеется класс запросов, которые не могут быть выражены без подзапроса во фразе FROM. К ним, в частности, относятся такие запросы, которые требуют независимого вычисления двух или более запросов, и затем совместного использования результатов такого запроса.

Пример:

Выдать средний фонд финансирования факультетов и среднюю зарплату преподавателей:

```
SELECT Fac.AvgFund, Tch.AvgSalary
FROM (SELECT AVG(Fund) AS AvgFund FROM FACULTY) Fac,
     (SELECT AVG(Salary) AS AvgSalary FROM TEACHER) Tch;
```

1.1.9. Подзапросы во фразе SELECT

Во фразе SELECT можно использовать простые (независимые, несвязанные) и связанные (коррелированные) запросы. **В обоих случаях подзапрос должен возвращать одно значение.**

Подзапрос является простым, если в нем не используются атрибуты таблиц, определенных в основном (внешнем запросе). При использовании простого подзапроса он вычисляется

однократно и возвращенное им значение вставляется в соответствующее место во все строки, формируемые внешним запросом.

Пример. Для каждого факультета вывести его название, фонд финансирования, а также максимальный и минимальный фонды финансирования среди всех кафедр ВУЗа

```
SELECT Name AS "Факультет",  
       Fund AS "Фонд факультета",  
       (SELECT MAX(Fund) FROM DEPARTMENT) AS "МАКС фонд кафедр",  
       (SELECT MIN(Fund) FROM DEPARTMENT) AS "МИН фонд кафедр"  
FROM FACULTY
```

Подзапрос является связанным (коррелированным), если в нем используются атрибуты таблиц, определенных во внешней запросе. В этом случае подзапрос вычисляется для каждой строки, формируемой для фразы SELECT.

Пример. По каждому факультету, расположенному в корпусе 6, вывести:

- название факультета
- количество групп этого факультета с рейтингом, более 20
- количество преподавателей-профессоров

```
SELECT Name AS "Факультет",  
       (SELECT COUNT (DISTINCT GrpPK)  
        FROM DEPARTMENT d, SGROUP g  
        WHERE f.FacPK=d.FacFK AND d.DepPK=g.DepFK AND g.Rating > 20) AS "К-во  
групп",  
       (SELECT COUNT (DISTINCT TchPK)  
        FROM DEPARTMENT d, TEACHER t  
        WHERE f.FacPK=d.FacFK AND d.DepPK=t.DepFK AND  
              UPPER(t.Post)= 'профессор') AS "К-во профессоров"  
FROM FACULTY f  
WHERE Building= '6';
```

1.2. Обработка оптимизации вложенных запросов

1.2.1. Вложенные подзапросы

Вложенный подзапрос - это подзапрос, заключенный в круглые скобки и вложенный в WHERE (HAVING) фразу предложения SELECT или других предложений, использующих WHERE фразу. Вложенный подзапрос может содержать в своей WHERE (HAVING) фразе другой вложенный подзапрос и т.д. Вложенный подзапрос создается для того, чтобы при отборе строк таблицы, сформированным основным запросом, можно было использовать данные из других таблиц.

Существуют простые и коррелированные вложенные подзапросы. Они включаются в WHERE (HAVING) фразу с помощью условий IN, EXISTS или одного из условий сравнения (= | <> | < | <= | > | >=). Простые вложенные подзапросы обрабатываются системой "снизу вверх". Первым обрабатывается вложенный подзапрос самого нижнего уровня. Множество значений, полученное в результате его выполнения, используется при реализации подзапроса более высокого уровня и т.д.

Запросы с коррелированными вложенными подзапросами обрабатываются системой в обратном порядке. Сначала выбирается первая строка рабочей таблицы, сформированной основным запросом, и из нее выбираются значения тех столбцов, которые используются во вложенном подзапросе (вложенных подзапросах). Если эти значения удовлетворяют условиям вложенного подзапроса, то выбранная строка включается в результат. Затем выбирается вторая строка и т.д., пока в результат не будут включены все строки, удовлетворяющие вложенному подзапросу (последовательности вложенных подзапросов).

1.2.2. Простые вложенные подзапросы

Простые вложенные подзапросы используются для представления множества значений, исследование которых должно осуществляться в каком-либо предикате IN, что иллюстрируется в следующем примере: выдать название и статус поставщиков продукта с номером 11, т.е. Микропроцессор.

Результат:

```
SELECT Название, Статус
FROM Поставщики
WHERE ПС IN
    ( SELECT ПС
      FROM Поставки
      WHERE ПР = 11 );
```

| Название | Статус |
|----------|---------------|
| КОМП | Дилер |
| ЭВМ | Производитель |
| СТЕЛА | ОАО |
| ВЫМПЕЛ | ЗАО |

При обработке полного запроса система выполняет прежде всего вложенный подзапрос. Этот подзапрос выдает множество номеров поставщиков, которые поставляют продукт с кодом ПР = 11, а именно множество (1, 5, 6, 8). Поэтому первоначальный запрос эквивалентен такому простому запросу:

```
SELECT Название, Статус
FROM Поставщики
WHERE ПС IN (1, 5, 6, 8);
```

Подзапрос с несколькими уровнями вложенности можно проиллюстрировать на том же примере. Пусть требуется узнать не поставщиков продукта 11, а поставщиков микропроцессоров, являющихся продуктом с номером 11. Для этого можно дать запрос

```
SELECT Название, Статус
FROM Поставщики
WHERE ПС IN
    ( SELECT ПС
      FROM Поставки
      WHERE ПР IN
          ( SELECT ПР
            FROM Продукты
            WHERE Продукт = Микропроцессор));
```

В данном случае результатом самого внутреннего подзапроса является только одно значение (11). Как уже было показано выше, подзапрос следующего уровня в свою

очередь дает в результате множество (1, 5, 6, 8). Последний, самый внешний SELECT, вычисляет приведенный выше окончательный результат. Вообще допускается любая глубина вложенности подзапросов.

Тот же результат можно получить с помощью соединения

```
SELECT Название, Статус
FROM Поставщики, Поставки, Продукты
WHERE Поставщики.ПС = Поставки.ПС
AND Поставки.ПП = Продукты.ПП
AND Продукт = Микропроцессор;
```

При выполнении этого компактного запроса система должна одновременно обрабатывать данные из трех таблиц, тогда как в предыдущем примере эти таблицы обрабатываются поочередно. Естественно, что для их реализации требуются различные ресурсы памяти и времени.

1.2.3. Использование одной и той же таблицы во внешнем и вложенном подзапросе

Выдать номера поставщиков, которые поставляют хотя бы один продукт, поставляемый поставщиком 6.

| | Результат: |
|--------------------|-------------------|
| SELECT DISTINCT ПС | ПС |
| FROM Поставки | 1 |
| WHERE ПП IN | 3 |
| (SELECT ПП | 5 |
| FROM Поставки | 6 |
| WHERE ПС = 6); | 8 |

Отметим, что ссылка на Поставки во вложенном подзапросе означает не то же самое, что ссылка на Поставки во внешнем запросе. В действительности, два имени

Поставки обозначают различные значения. Чтобы этот факт стал явным, полезно использовать псевдонимы, например, X и Y:

```
SELECT DISTINCT X.ПС
FROM   Поставки X
WHERE  X.ПР IN
      ( SELECT Y.ПР
        FROM   Поставки Y
        WHERE  Y.ПС = 6 );
```

Здесь X и Y \sqrt произвольные псевдонимы таблицы Поставки, определяемые во фразе FROM и используемые как явные уточнители во фразах SELECT и WHERE. Напомним, что псевдонимы определены лишь в пределах одного запроса.

1.2.4. Вложенный подзапрос с оператором сравнения, отличным от IN

Выдать номера поставщиков, находящихся в том же городе, что и поставщик с номером 6.

| | Результат: |
|-------------------|------------|
| SELECT ПС | ПС |
| FROM Поставщики | 1 |
| WHERE Город = | 4 |
| (SELECT Город | 6 |
| FROM Поставщики | |
| WHERE ПС = 6); | |

В подобных запросах можно использовать и другие операторы сравнения (\lt , \leq , \lt , \geq или \gt), однако, если вложенный подзапрос возвращает более одного значения и не используется оператор IN, будет возникать ошибка.

1.2.5. Коррелированные вложенные подзапросы

Выдать название и статус поставщиков продукта с номером 11.

```
SELECT Название, Статус
FROM Поставщики
WHERE 11 IN
    ( SELECT ПР
      FROM Поставки
      WHERE ПС = Поставщики.ПС );
```

Такой подзапрос отличается от рассмотренного в п.1.2.2. тем, что вложенный подзапрос не может быть обработан прежде, чем будет обрабатываться внешний подзапрос. Это связано с тем, что вложенный подзапрос зависит от значения Поставщики.ПС а оно изменяется по мере того, как система проверяет различные строки таблицы Поставщики. Следовательно, с концептуальной точки зрения обработка осуществляется следующим образом:

1. Система проверяет первую строку таблицы Поставщики. Предположим, что это строка поставщика с номером 1. Тогда значение Поставщики.ПС будет в данный момент имеет значение, равное 1, и система обрабатывает внутренний запрос

```
( SELECT ПР
  FROM Поставки
  WHERE ПС = 1 );
```

получая в результате множество (9, 11, 12, 15). Теперь система может завершить обработку для поставщика с номером 1. Выборка значений Название и Статус для ПС=1 (КОМП и Дилер) будет проведена тогда и только тогда, когда ПР=11 будет принадлежать этому множеству, что, очевидно, справедливо.

2. Далее система будет повторять обработку такого рода для следующего поставщика и т.д. до тех пор, пока не будут рассмотрены все строки таблицы Поставщики.

Подобные подзапросы называются коррелированными, так как их результат зависит от значений, определенных во внешнем подзапросе. Обработка коррелированного подзапроса, следовательно, должна повторяться для каждого значения извлекаемого из внешнего подзапроса, а не выполняться раз и навсегда.

Рассмотрим пример использования одной и той же таблицы во внешнем подзапросе и коррелированном вложенном подзапросе.

Выдать номера всех продуктов, поставляемых только одним поставщиком.

Результат:

```
ELECT DISTINCT X.ПР
```

X.ПР

```
FROM   Поставки X
```

17

```
WHERE  X.ПР NOT IN
```

```
(    SELECT Y.ПР
```

```
      FROM   Поставки Y
```

```
      WHERE  Y.ПС <> X.ПС );
```

Действие этого запроса можно пояснить следующим образом: "Поочередно для каждой строки таблицы Поставки, скажем X, выделить значение номера продукта (ПР), если и только если это значение не входит в некоторую строку, скажем, Y, той же таблицы, а значение столбца номер поставщика (ПС) в строке Y не равно его значению в строке X".

Отметим, что в этой формулировке должен быть использован по крайней мере один псевдоним - либо X, либо Y.

1.2.6. Запросы, использующие EXISTS

Квантор EXISTS (существует) - понятие, заимствованное из формальной логики. В языке SQL предикат с квантором существования представляется выражением EXISTS (SELECT * FROM ...).

Такое выражение считается истинным только тогда, когда результат вычисления "SELECT * FROM ..." является непустым множеством, т.е. когда существует какая-либо запись в таблице, указанной во фразе FROM подзапроса, которая удовлетворяет условию

WHERE подзапроса. (Практически этот подзапрос всегда будет коррелированным множеством.)

Рассмотрим примеры. Выдать названия поставщиков, поставляющих продукт с номером 11.

```
SELECT Название
FROM Поставщики
WHERE EXISTS
  ( SELECT *
    FROM Поставки
    WHERE ПС = Поставщики.ПС
      AND ПР = 11 );
```

Результат:

Название
КОМП
ЭВМ
СТЕЛА
ВЫМПЕЛ

Система последовательно выбирает строки таблицы Поставщики, выделяет из них значения столбцов Название и ПС, а затем проверяет, является ли истинным условие существования, т.е. существует ли в таблице Поставки хотя бы одна строка со значением ПР=11 и значением ПС, равным значению ПС, выбранному из таблицы Поставщики. Если условие выполняется, то полученное значение столбца Название включается в результат.

Предположим, что первые значения полей Название и ПС равны, соответственно, КОМП и 1. Так как в таблице Поставки есть строка с ПР=11 и ПС=1, то значение КОМП должно быть включено в результат.

Хотя этот первый пример только показывает иной способ формулировки запроса для задачи, решаемой и другими путями (с помощью оператора IN или соединения), EXISTS представляет собой одну из наиболее важных возможностей SQL. Фактически любой запрос, который выражается через IN, может быть альтернативным образом сформулирован также с помощью EXISTS. Однако обратное высказывание несправедливо.

Выдать название и статус поставщиков, не поставляющих продукт с номером 11.

Результат:

```
SELECT    Название, Статус
FROM      Поставщики
WHERE NOT EXISTS
(        SELECT *
FROM      Поставки
WHERE    ПС = Поставщики.ПС
AND      ПР = 11 );
```

| Название | Статус |
|----------|---------------|
| АВИОН | Дилер |
| ТРЕНИР | Производитель |
| АРГОН | ОАО |
| КВАНТ | ЗАО |

1. 3. Выполнение вложенного запроса в однопроцессорной базе данных

1.3.1. Введение

Проблеме оптимизации выполнения вложенных запросов при обращении к базе данных посвящено большое число публикаций, включая [8]. В качестве критерия оптимизации вложенных запросов обычно используют время выполнения запроса. В данной работе развивается аналитический подход предложенный в работе [8] для оптимизации плана выполнения вложенных запросов.

Пусть запрос представляет собой конъюнкцию элементарных запросов. В простейшем случае вложенный запрос к базе данных можно рассматривать как обращение к таблицам, содержащим множество записей (строк), характеризующихся одинаковым множеством свойств (столбцов), с целью получения множества записей, удовлетворяющих заданному условию вложенного запроса. Будем называть часть запроса, относимую к i -му свойству записи (i -му столбцу), i -ым элементарным запросом \mathbb{E}_i .

При выполнении вложенного запроса для каждой строки необходимо выполнить проверку условия, связанного с \mathbb{E}_i , требуемое для этого время обозначим через τ_i вероятность успеха выполнения условия через p_i . Вариация значений времени τ_i зависит от формулировки условия \mathbb{E}_i (точечное, интервальное или более сложное условие), а так же от технических и/или программных особенностей реализации (кэширование, диски и др.). Вариация значений вероятности выполнения условия p_i определяется содержимым базы данных.

Основным критерием при определении плана реализации вложенного запроса является время выполнения запроса, которое, вообще говоря, зависит от порядка выполнения элементарных запросов, его составляющих, и указанных параметров элементарных запросов $\mathcal{E}Z_i$ - времени проверки τ_i и вероятности успеха p_i в j -ой строке i -го столбца. $i = \overline{1, k}, j = \overline{1, n}$

Время выполнения элементарного запроса зависит от метода доступа к столбцу таблицы, при этом выделяются два базовых метода, когда данные в столбцах неупорядочены и когда данные в столбцах упорядочены построчно. В первом случае время выполнения элементарного запроса $\mathcal{E}Z_i$ определяется временем проверки всех n строк i -го столбца таблицы и равно $n \tau_i$. Во втором случае, например, при использовании индексной организации обращений к данным, определяется временем проверки $n p_i$ строк и равно $n p_i \tau_i$.

Будем рассматривать задачу минимизации времени выполнения запроса при наличии вложенных запросов.

1.3.2. Влияние параметров запроса на время выполнения вложенного запроса

Определим задачу минимизации времени выполнения запроса при наличии вложенных запросов, рассмотрев следующий пример 1.

Пример 1.

Пусть запрос Z использует множество элементарных запросов $\mathcal{E}Z_i, i = \overline{1, k}$, выполнение которых осуществляется в соответствии условию $\frac{\tau_i}{1-p_i} < \frac{\tau_{i+1}}{1-p_{i+1}}, i = \overline{1, k-1}$.

При этом запрос Z образуют два запроса прямой $Z_{\text{п}}$ и вложенный $Z_{\text{вл}}$, а именно:

$$Z_{\text{пр}} = \mathcal{E}Z_1 \& \mathcal{E}Z_3 \& \dots \& \mathcal{E}Z_{k-1} \& \mathcal{E}Z_{\text{вл}}$$

$$Z_{\text{вл}} = \mathcal{E}Z_2 \& \mathcal{E}Z_4 \& \dots \& \mathcal{E}Z_k,$$

т.е. в прямом запросе используются элементарные запросы с нечетными номерами, а во вложенном запросе - с четными номерами.

Выполнение запросов может осуществляться разными вариантами. Мы рассмотрим три варианта:

1. Первыми выполняются элементарные запросы прямого запроса, затем элементарные запросы вложенного запроса;

2. Первыми выполняются элементарные запросы вложенного запроса, затем элементарные запросы прямого запроса;
3. Выполнение элементарных запросов и прямого и вложенного запросов осуществляются интегрированно

Для оценки влияния изменения значений параметров времени τ_i и вероятности p_i на изменения времени выполнения запроса рассмотрим два закона задания функций изменения параметров времени τ_i (геометрической прогрессии и арифметической прогрессии) при двух базовых методах доступа к столбцу таблицы, когда данные в столбцах неупорядочены и упорядочены.

Пусть запрос образуют конъюнкции k элементарных запросов:

1. С изменением параметра времени по закону геометрической прогрессии $\tau_i = a^{i-1}$, и с постоянным значением параметра вероятности $p_i = p, i \in 1, k$.
2. С измерением параметра времени по закону арифметической прогрессии, $\tau_i = 1 + (i - 1)\Delta$, с постоянным значением параметра вероятности $p_i = p, i \in 1, k$.

1.3.2.1. Неупорядоченные таблицы с параметрами, соответствующими геометрической прогрессии

Вариант 1

$$T_1 = \left((1 + pa^2 + \dots + p^{(k-2)/2} a^{k-2}) + p^{k/2} (a + pa^3 + \dots + p^{(k-2)/2} a^{k-1}) \right) n$$

$$= \left(1 + ap^{\frac{k}{2}} \right) \left(\frac{1 - (pa^2)^{k/2}}{1 - pa^2} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min_{ny}} / n$ при вариации значений параметров отражены в таблице 1.1 и рис.1.5.

Таблица 1.5

| p | $k=12$ | |
|-----|---------|---------|
| | a | |
| | 1.1 | 1.2 |
| 0.1 | 1.1377 | 1.1682 |
| 0.2 | 1.3191 | 1.4038 |
| 0.3 | 1.5675 | 1.7507 |
| 0.4 | 1.9217 | 2.2835 |
| 0.5 | 2.4489 | 3.1315 |
| 0.6 | 3.2751 | 4.5346 |
| 0.7 | 4.6562 | 6.9855 |
| 0.8 | 7.1375 | 11.5657 |
| 0.9 | 11.8913 | 20.684 |

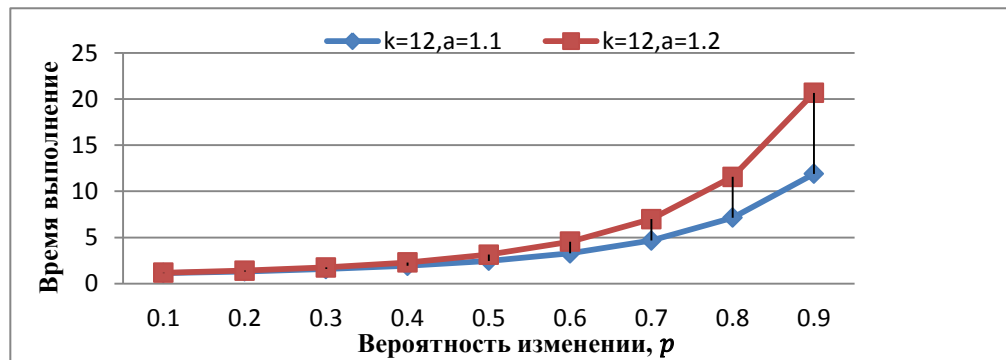


Рис.1.5. Время выполнения запроса для неупорядоченной таблицы при изменении параметра времени τ_i по закону геометрической прогрессии.

Вариант 2

$$T_2 = \left(\left(a + pa^3 + \dots + p^{\frac{k-2}{2}} a^{k-1} \right) + p^{k/2} \left(1 + pa^2 + \dots + p^{(k-2)/2} a^{k-2} \right) \right) n$$

$$= \left(a + p^{\frac{k}{2}} \right) \left(\frac{1 - (pa^2)^{k/2}}{1 - pa^2} \right) n .$$

Соответствующие значения времени выполнения запроса $T_{\min_{\text{ну}}} / n$ при вариации значений параметров отражены в таблице 1.2 и рис.1.6.

Таблица 1.2

| p | $k=12$ | |
|-----|---------|---------|
| | a | |
| | 1.1 | 1.2 |
| 0.1 | 1.2514 | 1.4019 |
| 0.2 | 1.451 | 1.6845 |
| 0.3 | 1.724 | 2.1002 |
| 0.4 | 2.1122 | 2.7361 |
| 0.5 | 2.6859 | 3.7367 |
| 0.6 | 3.5721 | 5.3534 |
| 0.7 | 5.02 | 8.0657 |
| 0.8 | 7.5463 | 12.864 |
| 0.9 | 12.2429 | 21.8675 |

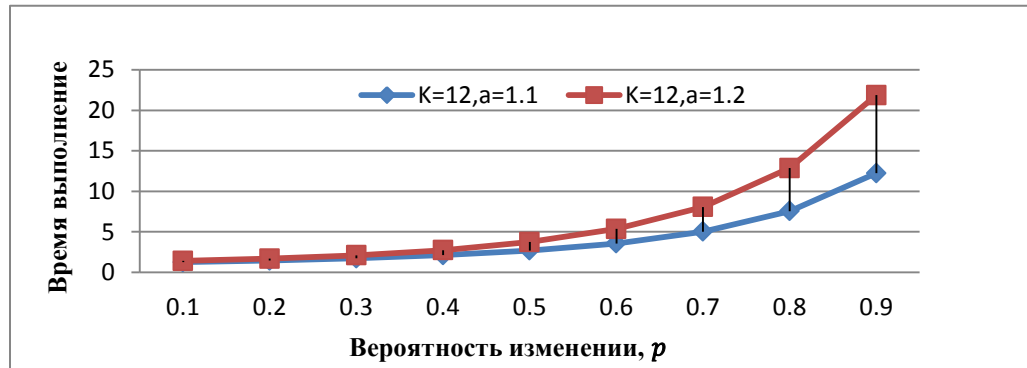


Рис.1.6. Минимальное время выполнения запроса для неупорядоченной таблицы при изменении параметра времени τ_i по закону геометрической прогрессии.

Вариант 3

$$T_3 = (1 + pa + \dots + (pa)^{k-1})n = \left(\frac{1 - (pa)^k}{1 - pa} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min \text{ ну}} / n$ при вариации значений параметров отражены в таблице 1.3 и рис.1.7.

Таблица 1.3

| p | $k=12$ | |
|-----|---------|---------|
| | a | |
| | 1.1 | 1.2 |
| 0.1 | 1.1236 | 1.1364 |
| 0.2 | 1.2821 | 1.3158 |
| 0.3 | 1.4925 | 1.5625 |
| 0.4 | 1.7856 | 1.9228 |
| 0.5 | 2.2205 | 2.4946 |
| 0.6 | 2.9211 | 3.5021 |
| 0.7 | 4.159 | 5.4787 |
| 0.8 | 6.5361 | 9.6823 |
| 0.9 | 11.3615 | 18.9771 |

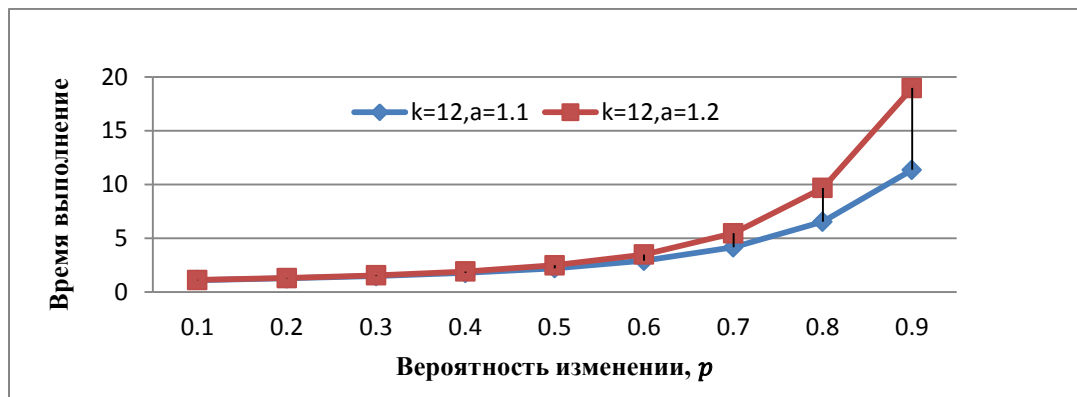


Рис.1.7. Минимальное время выполнения запроса для неупорядоченной таблицы при изменении параметра времени τ_i по закону геометрической прогрессии.

1.3.2.2. Упорядоченные таблицы с параметрами, соответствующими геометрической прогрессии

Вариант 1

$$T_1 = p \left(1 + pa^2 + \dots + p^{\frac{k-2}{2}} a^{k-2} \right) + p^{\frac{k}{2}} \left(a + pa^3 + \dots + p^{\frac{k-2}{2}} a^{k-1} \right) n$$

$$= p \left(1 + ap^{\frac{k}{2}} \right) \left(\frac{1 - (pa^2)^{k/2}}{1 - pa^2} \right) n;$$

Соответствующие значения времени выполнения запроса $T_{\min y} / n$ при вариации значений параметров отражены в таблице 1.4 и рис.1.8.

Таблица 1.4

| <i>p</i> | <i>k</i> =12 | |
|----------|--------------|---------|
| | <i>a</i> | |
| | 1.1 | 1.2 |
| 0.1 | 0.11377 | 0.11682 |
| 0.2 | 0.26382 | 0.28076 |
| 0.3 | 0.47026 | 0.5252 |
| 0.4 | 0.76868 | 0.91341 |
| 0.5 | 1.2244 | 1.5658 |
| 0.6 | 1.9651 | 2.7208 |
| 0.7 | 3.2593 | 4.8898 |
| 0.8 | 5.71 | 9.2525 |
| 0.9 | 10.7022 | 18.6156 |

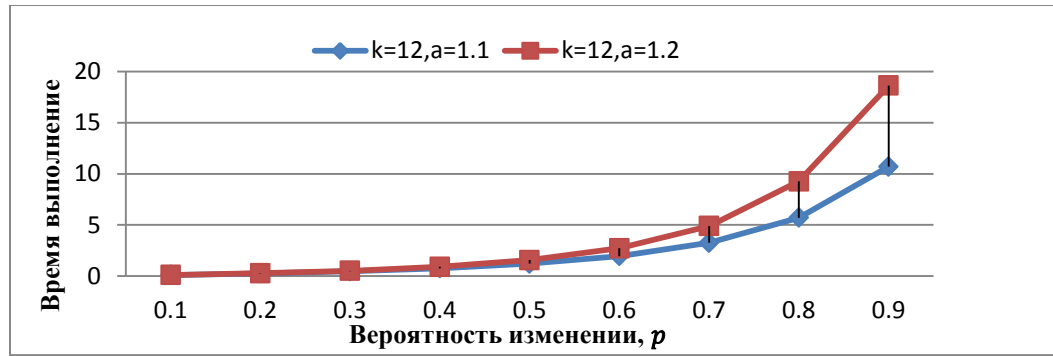


Рис.1.8. Минимальное время выполнения запроса для упорядоченной таблицы при изменении параметра времени τ_i по закону геометрической прогрессии.

Вариант 2

$$T_2 = p \left(a + pa^3 + \dots + p^{\frac{k-2}{2}} a^{k-1} \right) + p^{k/2} \left(1 + pa^2 + \dots + p^{(k-2)/2} a^{k-2} \right) n$$

$$= p \left(a + p^{\frac{k}{2}} \right) \left(\frac{1 - (pa^2)^{k/2}}{1 - pa^2} \right) n;$$

Соответствующие значения времени выполнения запроса $T_{\min y} / n$ при вариации значений параметров отражены в таблице 1.5 и рис.1.9.

Таблица 1.5

| p | k = 12 | |
|-----|---------|---------|
| | a | |
| | 1.1 | 1.2 |
| 0.1 | 0.12514 | 0.14019 |
| 0.2 | 0.2902 | 0.3369 |
| 0.3 | 0.51721 | 0.63007 |
| 0.4 | 0.84489 | 1.0945 |
| 0.5 | 1.3429 | 1.8683 |
| 0.6 | 2.1433 | 3.212 |
| 0.7 | 3.514 | 5.646 |
| 0.8 | 6.037 | 10.2912 |
| 0.9 | 11.0187 | 19.6808 |

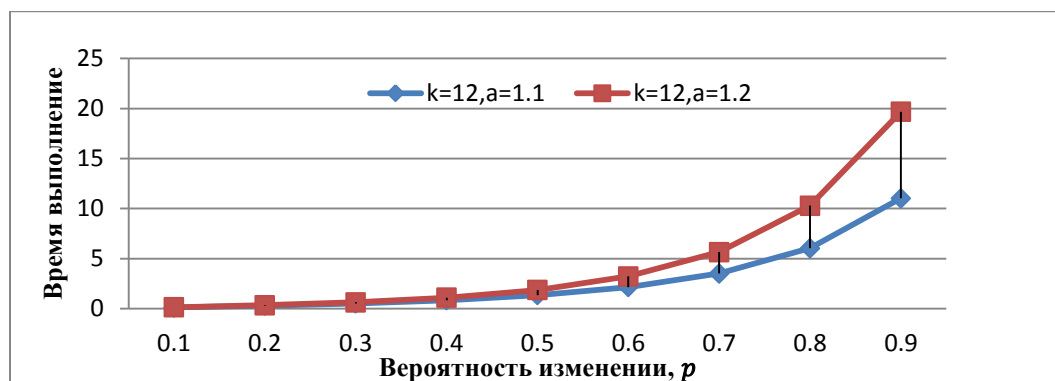


Рис.1.9. Минимальное время выполнения запроса для упорядоченной таблицы при изменении параметра времени τ_i по закону геометрической прогрессии.

Вариант 3

$$T_3 = p(1 + pa + \dots + (pa)^{k-1})n = p \left(\frac{1-(pa)^k}{1-pa} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min y} / n$ при вариации значений параметров отражены в таблице 1.6 и рис.1.10.

Таблица 1.6

| p | $k = 12$ | |
|-----|----------|---------|
| | a | |
| | 1.1 | 1.2 |
| 0.1 | 0.11236 | 0.11364 |
| 0.2 | 0.25641 | 0.26316 |
| 0.3 | 0.44776 | 0.46875 |
| 0.4 | 0.71425 | 0.76912 |
| 0.5 | 1.1103 | 1.2473 |
| 0.6 | 1.7526 | 2.1013 |
| 0.7 | 2.9113 | 3.8351 |
| 0.8 | 5.2289 | 7.7458 |
| 0.9 | 10.2254 | 17.0794 |

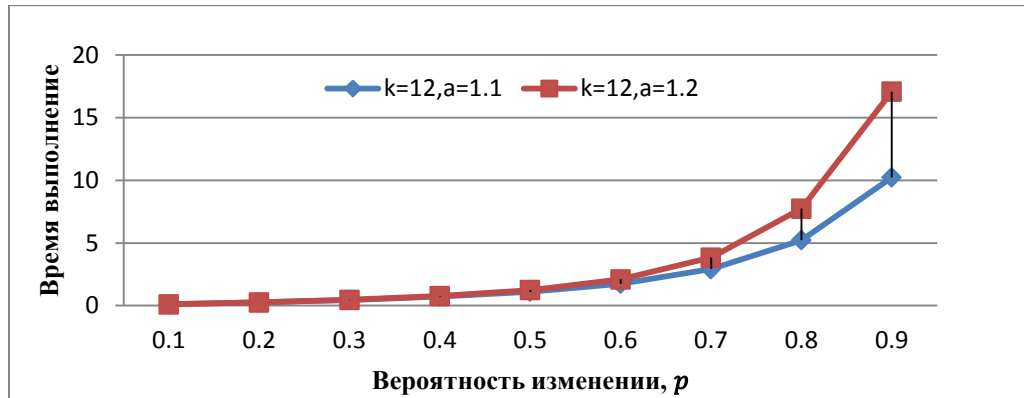


Рис.1.10. Минимальное время выполнения запроса для упорядоченной таблицы при изменении параметра времени τ_i по закону геометрической прогрессии.

1.3.2.3. Неупорядоченные таблицы с параметрами, соответствующими арифметической прогрессии

Вариант 1

$$T_1 = \left(\left(1 + (1 + \Delta)p^{\frac{k}{2}} \right) \frac{1 - p^{\frac{k}{2}}}{1 - p} + 2p\Delta \left(1 + p^{\frac{k}{2}} \right) \frac{1 - \frac{k}{2}p^{\frac{k}{2}-1} + \left(\frac{k}{2} - 1\right)p^{\frac{k}{2}}}{(1 - p)^2} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min_{ny}} / n$ при вариации значений параметров отражены в таблице 1.7 и рис.1.11.

Таблица 1.7

| p | $k=12$ | |
|-----|----------|-------|
| | Δ | |
| | 0.2 | 0.3 |
| 0.1 | 1.160 | 1.185 |
| 0.2 | 1.374 | 1.437 |
| 0.3 | 1.671 | 1.792 |

| | | |
|-----|--------|--------|
| 0.4 | 2.096 | 2.310 |
| 0.5 | 2.729 | 3.094 |
| 0.6 | 3.720 | 4.333 |
| 0.7 | 5.372 | 6.415 |
| 0.8 | 8.329 | 10.166 |
| 0.9 | 13.973 | 17.372 |

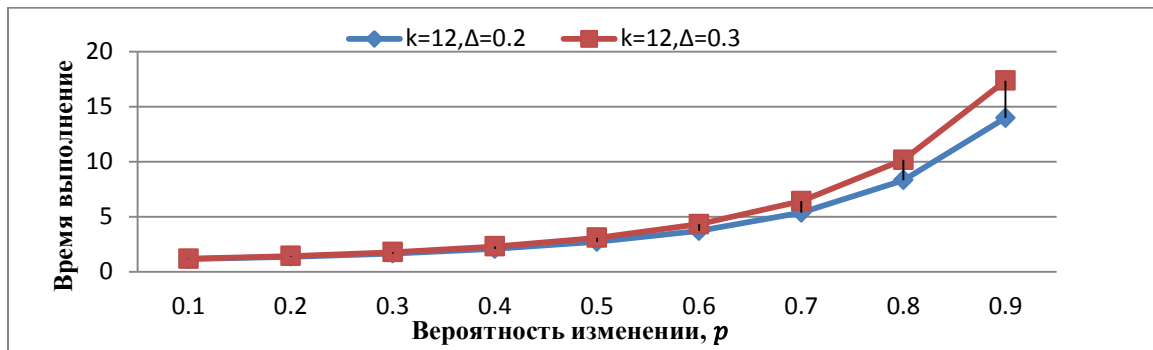


Рис.1.11. Минимальное время выполнения запроса для неупорядоченной таблицы при изменении параметра времени τ_i по закону арифметической прогрессии.

Вариант 2

$$T_2 = \left(\left(1 + \Delta + p^{\frac{k}{2}} \right) \frac{1 - p^{\frac{k}{2}}}{1 - p} + 2p\Delta \left(1 + p^{\frac{k}{2}} \right) \frac{1 - \frac{k}{2} p^{\frac{k}{2}-1} + \left(\frac{k}{2} - 1 \right) p^{\frac{k}{2}}}{(1 - p)^2} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min ny} / n$ при вариации значений параметров отражены в таблице 1.8 и рис.1.12.

Таблица 1.8

| p | $k=12$ | |
|-----|----------|--------|
| | Δ | |
| | 0.2 | 0.3 |
| 0.1 | 1.382 | 1.518 |
| 0.2 | 1.624 | 1.812 |
| 0.3 | 1.956 | 2.220 |
| 0.4 | 2.426 | 2.806 |
| 0.5 | 3.116 | 3.675 |
| 0.6 | 4.175 | 5.015 |
| 0.7 | 5.891 | 7.193 |
| 0.8 | 8.874 | 10.983 |
| 0.9 | 14.415 | 18.030 |

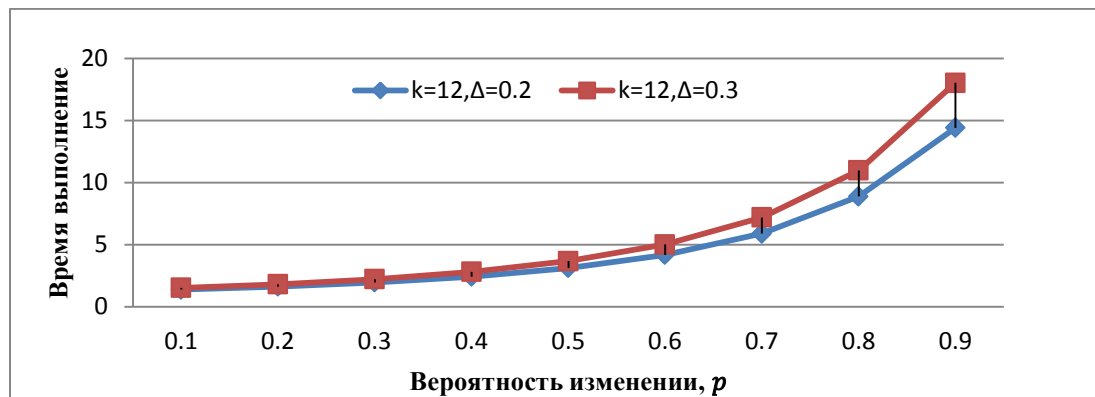


Рис.1.12. Минимальное время выполнения запроса для неупорядоченной таблицы при изменении параметра времени τ_i по закону арифметической прогрессии.

Вариант 3

$$T_3 = \left(\frac{1 - p^k}{1 - p} + p\Delta \frac{1 - kp^{k-1} + (k-1)p^k}{(1-p)^2} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min_{ny}} / n$ при вариации значений параметров отражены в таблице 1.9 и рис.1.13.

Таблице 1.9

| p | $k=12$ | |
|-----|----------|--------|
| | Δ | |
| | 0.2 | 0.3 |
| 0.1 | 1.135 | 1.148 |
| 0.2 | 1.312 | 1.343 |
| 0.3 | 1.551 | 1.612 |
| 0.4 | 1.888 | 1.999 |
| 0.5 | 2.398 | 2.597 |
| 0.6 | 3.229 | 3.597 |
| 0.7 | 4.710 | 5.422 |
| 0.8 | 7.556 | 9.007 |
| 0.9 | 13.313 | 16.382 |

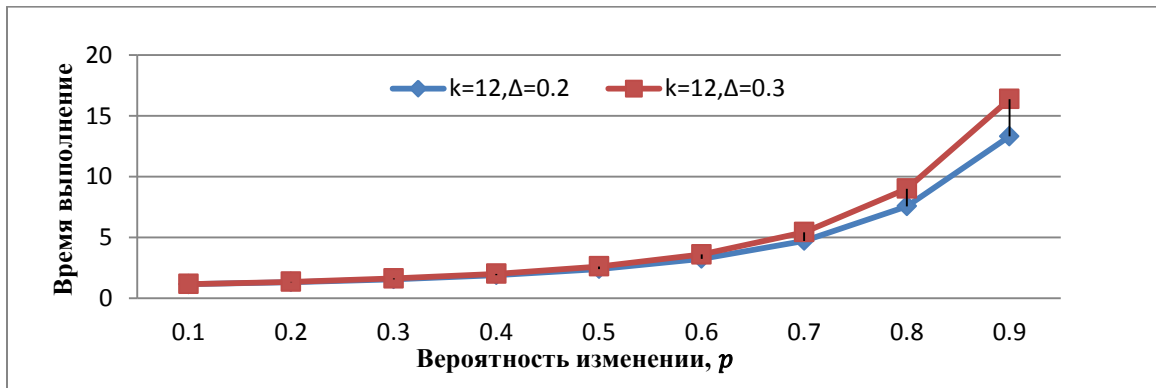


Рис.1.13. Минимальное время выполнения запроса для неупорядоченной таблицы при изменении параметра времени τ_i по закону арифметической прогрессии.

1.3.2.4. Упорядоченные таблицы с параметрами, соответствующими арифметической прогрессии

Вариант 1

$$T_1 = p \left(\left(1 + (1 + \Delta)p^{\frac{k}{2}} \right) \frac{1 - p^{\frac{k}{2}}}{1 - p} + 2p\Delta \left(1 + p^{\frac{k}{2}} \right) \frac{1 - \frac{k}{2}p^{\frac{k}{2}-1} + \left(\frac{k}{2} - 1\right)p^{\frac{k}{2}}}{(1 - p)^2} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min y} / n$ при вариации значений параметров отражены в таблице 1.10 и рис.1.14.

Таблица 1.10

| p | $k = 12$ | |
|-----|----------|--------|
| | Δ | |
| | 0.2 | 0.3 |
| 0.1 | 0.116 | 0.118 |
| 0.2 | 0.274 | 0.287 |
| 0.3 | 0.501 | 0.537 |
| 0.4 | 0.838 | 0.924 |
| 0.5 | 1.364 | 1.547 |
| 0.6 | 2.23 | 2.600 |
| 0.7 | 3.760 | 4.490 |
| 0.8 | 6.663 | 8.133 |
| 0.9 | 12.576 | 15.634 |

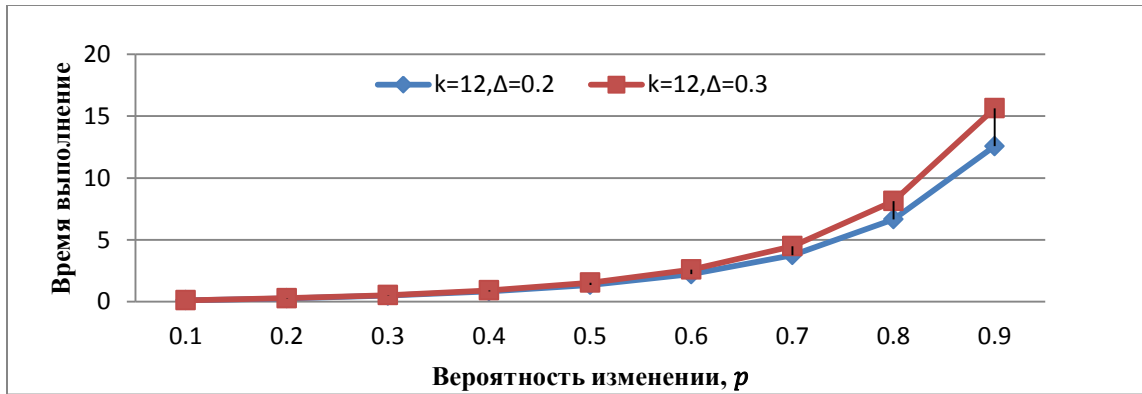


Рис.1.14. Минимальное время выполнения запроса для упорядоченной таблицы при изменении параметра времени τ_i по закону арифметической прогрессии.

Вариант 2

$$T_2 = p \left(\left(1 + \Delta + p^{\frac{k}{2}} \right) \frac{1 - p^{\frac{k}{2}}}{1 - p} + 2p\Delta \left(1 + p^{\frac{k}{2}} \right) \frac{1 - \frac{k}{2} p^{\frac{k}{2}-1} + \left(\frac{k}{2} - 1 \right) p^{\frac{k}{2}}}{(1 - p)^2} \right) n.$$

Соответствующие значения времени выполнения запроса T_{\min} / n при вариации значений параметров отражены в таблице 1.11 и рис.1.14.

Таблица 1.11

| p | $k=12$ | |
|-----|----------|-------|
| | Δ | |
| | 0.2 | 0.3 |
| 0.1 | 0.138 | 0.151 |
| 0.2 | 0.324 | 0.362 |
| 0.3 | 0.586 | 0.666 |
| 0.4 | 0.970 | 1.122 |
| 0.5 | 1.558 | 1.837 |
| 0.6 | 2.505 | 3.009 |
| 0.7 | 4.124 | 5.035 |

| | | |
|-----|--------|--------|
| 0.8 | 7.099 | 8.786 |
| 0.9 | 12.971 | 16.227 |

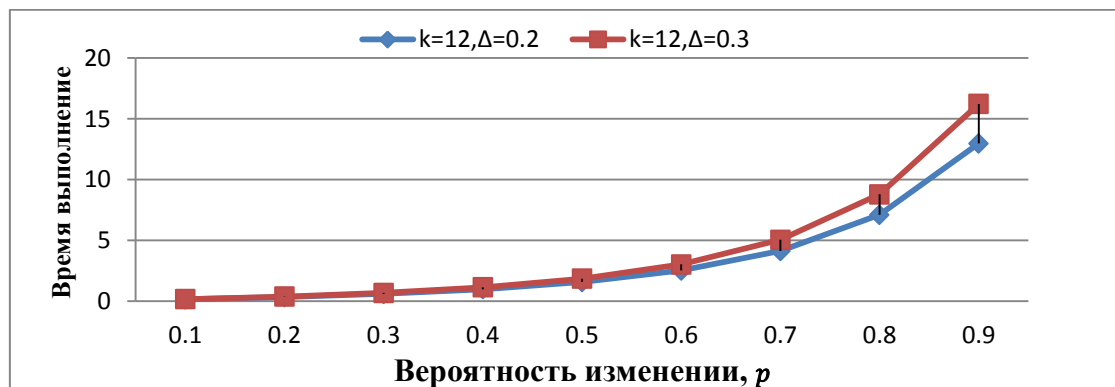


Рис.1.14. Минимальное время выполнения запроса для упорядоченной таблицы при изменении параметра времени τ_i по закону арифметической прогрессии.

Вариант 3

$$T_3 = p \left(\frac{1 - p^k}{1 - p} + p\Delta \frac{1 - kp^{k-1} + (k-1)p^k}{(1-p)^2} \right) n.$$

Соответствующие значения времени выполнения запроса $T_{\min y} / n$ при вариации значений параметров отражены в таблице 1.12 и рис.1.16.

Таблице 1.12

| p | k=12 | |
|-----|-------|-------|
| | Δ | |
| | 0.2 | 0.3 |
| 0.1 | 0.113 | 0.114 |
| 0.2 | 0.262 | 0.268 |
| 0.3 | 0.465 | 0.483 |

| | | |
|-----|--------|--------|
| 0.4 | 0.755 | 0.799 |
| 0.5 | 1.199 | 1.298 |
| 0.6 | 1.937 | 2.158 |
| 0.7 | 3.297 | 3.795 |
| 0.8 | 6.045 | 7.205 |
| 0.9 | 11.982 | 14.744 |

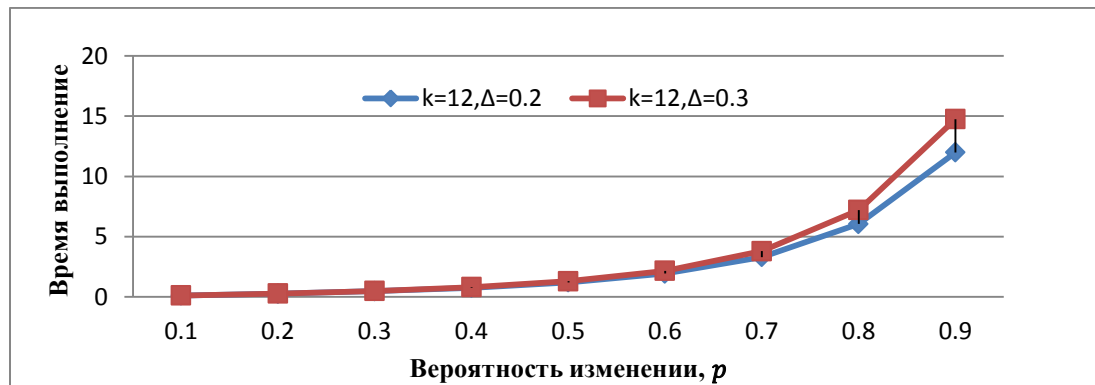


Рис.1.16. Минимальное время выполнения запроса для упорядоченной таблицы при изменении параметра времени τ_i по закону арифметической прогрессии.

Приведенные результаты примера позволяют сделать общий вывод по отношению к обработке запросов, включающих вложенные запросы, - порядок обработки совокупных элементарных запросов оказывает существенную роль на время выполнения общего запроса при значениях вероятностей $p_i = 0.6, 0.7$. Минимальное время обработки запроса достигается при совместной обработке объединенных элементарных запросов всех таблиц, образующих общий вложенный запрос.

1. 4. Постановка задачи

Выполнение вложенных запросов можно рассматривать как отдельную обработку таблиц в некотором порядке, или интегрированно, как это мы видим в выше приведенном примере.

Время обработки в зависимости от вариантов обработки достаточно сильно варьируется. Обработка этих таблиц может быть выполнена параллельно несколькими процессорами.

При этом возможно распараллеливание за счет параллельной обработки строк и или столбцов таблиц. В данной работе мы анализируем эффективность параллельной обработки столбцов таблиц, при этом учитывается порядок обработки столбцов таблиц на каждом процессоре.

Выводы по главе 1

1. Рассмотрены методы формирования вложенных запросов.
2. Выполнение вложенных запросов можно рассматривать как отдельную обработку таблиц в некотором порядке, или интегрированно.
3. Сформулирована основная задача исследования оптимизация плана реализации вложенного запроса в многопроцессорных базах данных с учетом порядка реализации элементарных запросов.

Глава 2. Обоснование квазиоптимального порядка распределения элементарных запросов в многопроцессорной базе данных

2.1. Введение

В работе [9] предложены два плана оптимизации по времени выполнения запросов при обращении к многопроцессорной базе данных на основе естественного и квазиоптимального методов упорядочивания элементарных запросов по процессорам.

В этой работе дается обоснование эффективности квазиоптимального метода распределения элементарных запросов в многопроцессорной базе данных, что является одной из актуальных задач разработки методики оптимизации обработки запросов в многопроцессорных базах данных авиационно-космических систем.

2.2. Распределение номеров элементарных запросов по процессорам

Критерий распределения элементарных запросов по процессорам естественно определить как получение минимального времени обработки запросов, когда все процессоры завершили бы свою работу одновременно, при этом, конечно, необходимо сохранить порядок обработки элементарных запросов на каждом из процессоров в соответствии с теоремой .

Можно воспользоваться естественным распределением элементарных запросов по процессорам в соответствии с правилом, показанным в таблице 2.1, где в i -ой строке таблицы указаны номера элементарных запросов выполняемых i -м процессором в порядке слева направо, при этом мы используем (для простоты изложения) в качестве значения числа элементарных запросов k целое четное число, и число процессоров r отвечает условию $[k/r]r = k$.

В этом случае i -ый ($i=1, \dots, r$) процессор получает элементарные запросы с номерами $i, r + i, 2r + i, 3r + i, 4r + i, 5r + i, 6r + i, \dots$

Таблица 2.1. Распределение номеров элементарных запросов по процессорам

| Номера процессоров | Номера элементарных запросов | | | | | |
|--------------------|------------------------------|-------|--------|--------|--|---------|
| 1 | 1 | $r+1$ | $2r+1$ | $3r+1$ | | $k-r+1$ |
| 2 | 2 | $r+2$ | $2r+2$ | $3r+2$ | | $k+1$ |
| 3 | 3 | $r+3$ | $2r+3$ | $3r+3$ | | |
| | | | | | | |
| i | i | $r+i$ | $2r+i$ | $3r+i$ | | $k+i$ |
| | | | | | | |
| r | r | $2r$ | $3r$ | $4r$ | | k |

В работе [9] предложен квазиоптимальный метод оптимизации, когда распределение элементарных запросов по процессорам осуществляется в соответствии с правилом, показанным в таблице 2.2, где в i -ой строке таблицы указаны номера элементарных запросов выполняемых i -м процессором в порядке слева направо, при этом мы используем (для простоты изложения) в качестве значения числа элементарных запросов k целое четное число, и число процессоров r отвечает условию $[k/r]r = k$.

В этом случае i -ый ($i = 1, \dots, r$) процессор получает элементарные запросы с номерами $i, 2r+1-i, 2r+i, 4r+1-i, 4r+i, 6r+1-i, 6r+i, \dots$

Таблица 2.2. Квазиоптимальное распределение элементарных запросов по процессорам

| Номера процессоров | Номера элементарных запросов | | | | | |
|--------------------|------------------------------|----------|--------|----------|--|---------|
| 1 | 1 | $2r$ | $2r+1$ | $4r$ | | k |
| 2 | 2 | $2r-1$ | $2r+2$ | $4r-1$ | | $k-1$ |
| 3 | 3 | $2r-2$ | $2r+3$ | $4r-2$ | | $k-2$ |
| | | | | | | |
| i | i | $2r-i+1$ | $2r+i$ | $4r-i+1$ | | $k-i+1$ |
| | | | | | | |
| $r-1$ | $r-1$ | $r+2$ | $3r-1$ | $3r+2$ | | $k-r+2$ |

| | | | | | | |
|-----|-----|-------|------|--------|--|---------|
| r | r | $r+1$ | $3r$ | $3r+1$ | | $k-r+1$ |
|-----|-----|-------|------|--------|--|---------|

2.3. Обоснование квазиоптимального порядка распределения

Для оценки влияния изменения числа процессоров на изменение времени выполнения запроса рассмотрим два закона задания функций изменения параметров времени τ_i (геометрической прогрессии и арифметической прогрессии) при двух базовых методах доступа к столбцам таблицы, когда данные в столбцах неупорядочены и упорядочены.

Пусть запрос образуют конъюнкции k элементарных запросов:

1. С изменением параметра времени по закону геометрической прогрессии $\tau_i = a^{i-1}$, и с постоянным значением параметра вероятности $p_i = p, i \in 1, k$.
2. С изменением параметра времени по закону арифметической прогрессии $\tau_i = 1+(i-1)\Delta$, и с постоянным значением параметра вероятности $p_i = p, i \in 1, k$.

Для простоты, но, не теряя общности, формирования подмножеств элементарных запросов для r процессоров положим, что число процессоров r отвечает условию $[k/r]r = k$.

2.3.1. Обработки запроса для неупорядоченных столбцов таблицы естественный порядок распределения. Арифметическая прогрессия

Пусть число процессоров вычислительной системы равно r , время выполнения элементарного запроса подчиняется закону арифметической прогрессии.

$$\tau_i = 1 + \Delta(i - 1), p_i = p, i = \overline{1, k}.$$

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при естественном порядке их распределения по процессорам, когда данные в столбцах неупорядочены равно:

$$T_{r,i,e} = \left((1 + (i - 1)\Delta) \frac{1-p^{\frac{k}{r}}}{1-p} + rp\Delta \frac{1-\frac{k}{r}p^{\frac{k}{r}-1} + (\frac{k}{r}-1)p^{\frac{k}{r}}}{(1-p)^2} \right) n, i = \overline{1, r}.$$

В самом деле, в соответствии с теоремой 1 имеем:

$$T_{r,i,a,e} = ((1 + (i - 1)\Delta) + p(1 + (r + i - 1)\Delta) + p^2(1 + (2r + i - 1)\Delta) + \dots + p^{\frac{k}{r}-1}(1 + (\frac{k}{r} - 1)r + i - 1)\Delta))n$$

$$= \left((1 + (i - 1)\Delta) \left(1 + p + p^2 + \dots + p^{\frac{k}{r}-1} \right) + rp\Delta \left(1 + 2p + 3p^2 + 4p^3 + \dots + \left(\frac{k}{2} - 1 \right) p^{\frac{k}{r}-2} \right) \right) n$$

$$= \left((1 + (i - 1)\Delta) \frac{1-p^{\frac{k}{r}}}{1-p} + rp\Delta \frac{1-\frac{k}{r}p^{\frac{k}{r}-1} + (\frac{k}{r}-1)p^{\frac{k}{r}}}{(1-p)^2} \right) n, i = \overline{1, r}.$$

Другой вид представления выражения для $T_{r,i,a,e}$, используемый для обоснования эффективности квазиоптимального порядка распределения:

$$T_{r,i,e} = ((1+(i-1)\Delta) + p^2(1 + (2r + (i - 1)\Delta) + p^4(1 + (4r + (i - 1)\Delta) + \dots + (p^{\frac{k}{r}-2} (1 + ((\frac{k}{r} - 2)r + i - 1)\Delta) + p(1 + (r + i - 1)\Delta) + p^3(1 + (3r + i - 1)\Delta) + \dots + p^{\frac{k}{r}-1} (1 + ((\frac{k}{r} - 1)r + i - 1)\Delta)) n, i = \overline{1, r}.$$

или

$$T_{r,i,a,e} = ((1+(i-1)\Delta) + (1 + (r + i - 1)\Delta)p) * (1 + p^2 + p^4 + \dots + p^{\frac{k}{r}-2} + 2rp^2\Delta(1 + p)(1 + 2p^2 + 3p^4 + 4p^6 + \dots + (\frac{k}{2r} - 1)p^{\frac{k}{r}-4}) n, i = \overline{1, r}.$$

или

$$T_{r,i,a,e} = \left((1 + (i - 1)\Delta + (1 + (r + i - 1)\Delta)p) \frac{1-p^{\frac{k}{r}}}{1-p^2} + 2r\Delta p^2(1 + p) \left(\frac{1-\frac{k}{2r}p^{\frac{k}{r}-2} + (\frac{k}{2r}-1)p^{\frac{k}{r}}}{(1-p^2)^2} \right) \right) n,$$

$$i = \overline{1, r}.$$

2.3.2. Квазиоптимальный порядок распределения. Арифметическая прогрессия

2.3.2.1. Время выполнения

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам, когда данные в столбцах неупорядочены равно (для простоты записей будем считать, что k/r есть целое четное число):

$$T_{r,i,a,o} = \left((1 + (i - 1)\Delta) + (1 + (2r - i)\Delta)p \frac{1-p^{\frac{k}{r}}}{1-p^2} + 2r\Delta p^2(1 + p) \frac{1-\frac{k}{2r}p^{\frac{k}{r}-2} + (\frac{k}{2r}-1)p^{\frac{k}{r}}}{(1-p^2)^2} \right) n,$$

$$i = \overline{1, r}.$$

В самом деле, в соответствии с теоремой 1 имеем:

$$T_{r,i,a,o} = ((1 + (i - 1)\Delta) + (1 + (2r - i)\Delta)p^1) + p^2(1 + (2r + i - 1)\Delta) + p^3(1 + (4r - i)\Delta) + p^4(1 + (4r + (i - 1))\Delta) + p^5(1 + (6r - i)\Delta) + \dots + p^{\frac{k}{2}-1} \left(1 + \left(\frac{k}{r} r - i \right) \right) n, i = \overline{1, r}.$$

или

$$T_{r,i,a,o} = \left((1 + (i - 1)\Delta) \left(1 + p^2 + \dots + p^{\frac{k}{r}-2} \right) + 2r\Delta p^2 \left(1 + 2p^2 + 3p^4 + 4p^6 + \dots + p^{\frac{k}{r}-4} \right) + p(1 + (2r - i)\Delta) \left(1 + p^2 + \dots + p^{\frac{k}{r}-2} \right) + 2r\Delta p^3 \left(1 + 2p^2 + 3p^4 + 4p^6 + \dots + \left(\frac{k}{2r} - 1 \right) p^{\frac{k}{r}-4} \right) \right) n$$

или

$$T_{r,i,a,o} = \left((1 + (i - 1)\Delta) + (1 + (2r - i)\Delta)p \frac{1-p^{\frac{k}{r}}}{1-p^2} + 2r\Delta p^2(1 + p) \frac{1-\frac{k}{2r}p^{\frac{k}{r}-2} + (\frac{k}{2r}-1)p^{\frac{k}{r}}}{(1-p^2)^2} \right) n, i = \overline{1, r}.$$

2.3.2.2. Соотношения времени выполнения на i -м и j -м процессорах

Для **естественного** порядка распределения элементарных запросов непосредственно из первого выражения для $T_{r,i,a,e}$ имеем неравенства:

$$T_{r,1,a,e} < T_{r,2,a,e} < \dots < T_{r,r,a,e}.$$

Для **квазиоптимального** порядка распределения элементарных запросов так же выполняются аналогичные неравенства:

$$T_{r,1,a,0} < T_{r,2,a,0} < \dots < T_{r,i,a,0} < \dots < T_{r,j,a,0} < T_{r,r,a,0}, 1 \leq i < j.$$

В самом деле, непосредственно из выражения для $T_{r,i,a,0}$ и $T_{r,j,a,0}$ имеем $1 + (i - 1)\Delta + (1 + (2r - i))p < 1 + (j - 1)\Delta + (1 + (2r - j)\Delta)p$,
или

$$p(j - i)\Delta < (j - i)\Delta,$$

или $p < 1$,

что справедливо всегда при любых $i < j$.

Максимальный разброс времени выполнения обработки запросов на r процессорах составляет:

- при **естественном** порядке

$$T_{r,r,a,e} - T_{r,1,a,e} = (r - 1)(1 + p)\Delta \left(\frac{1 - p^{\frac{k}{r}}}{1 - p^2} \right) n,$$

- при **квазиоптимальном** порядке

$$\begin{aligned} T_{r,r,a,0} - T_{r,1,a,0} &= (1 + (r - 1)\Delta + (1 + r\Delta)p) - (1 + (1 + (2r - 1)\Delta)p) \frac{1 - p^{\frac{k}{r}}}{1 - p^2} n \\ &= \left((r - 1)\Delta(1 - p) \frac{1 - p^{\frac{k}{r}}}{1 - p^2} \right) n. \end{aligned}$$

Имея в виду, что

$$T_{r,1,a,e} < T_{r,1,a,0},$$

так как выполняется неравенство

$$1 + p(1 + r\Delta) < 1 + p(1 + 2r - 1),$$

и так же

$$T_{r,r,a,0} < T_{r,1,a,e},$$

так как выполняется неравенство

$$1 + (r - 1)\Delta + (1 + r\Delta)p < 1 + (r - 1) + (1 + (2r - 1)\Delta)p,$$

находим, что минимальная и максимальная границы времени выполнения при квазиоптимальном распределении лежат внутри минимальной и максимальной границ при естественном распределении.

Абсолютное уменьшение границ времени выполнения запросов при использовании оптимального распределения вместо естественного распределения равно:

$$\delta_{abc} = (r - 1)\Delta \frac{1 - p^{\frac{k}{r}}}{1 - p^2} (1 + p - (1 - p))n = \left((r - 1)\Delta \frac{1 - p^{\frac{k}{r}}}{1 - p^2} \right) 2pn.$$

Относительное уменьшение границ времени выполнения запросов при использовании квазиоптимального распределения вместо естественного распределения равно:

$$\delta_{отн} = \frac{\delta_{abc}}{\left((r-1)\Delta(1+p) \left(\frac{1-p^{\frac{k}{r}}}{1-p^2} \right) n \right)} 100\% = \left(\frac{2p}{1+p} \right) 100\%,$$

что при практически часто используемом $p \geq 0,25$ составляет больше или равно 40 % и при $p \geq 0,5$ составляет $\geq 67\%$.

2.3.2.3. Эффективность квазиоптимального распределения

Полученные результаты характеризуют эффективность квазиоптимального распределения элементарных запросов по процессорам, т.к. это распределение обеспечивает минимальное время выполнения запроса на r процессорах.

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам равно (здесь, как и ранее, k/r есть целое четное число)

$$T_{r,i,g,0} = (a^{i-1} + pa^{2r-i}) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, i = \overline{1, r}.$$

Максимальное время выполнения k/r элементарных запросов на одном из r процессоров в частных случаях равно:

- для $r=1$

$$T_{r=1,1,a,0} = T_{r=1,1,a,e} = \left(\frac{1-p^k}{1-p} + p\Delta \frac{1-kp^{k-1}+(k-1)p^k}{(1-p)^2} \right) n,$$

- для $r = \frac{k}{4}$

$$T_{r=\frac{k}{4},\frac{k}{4},a,0} = \left(1 + \left(\frac{k}{4} - 1 \right) \Delta + p \left(1 + \frac{k}{4} \Delta \right) \right) (1 + p^2) + \frac{k}{2} \Delta p^2 (1 + p) n$$

- для $r = \frac{k}{2}$

$$T_{r=\frac{k}{2},\frac{k}{2},a,0} = \left(1 + \left(\frac{k}{2} - 1 \right) \Delta + p \left(1 + \frac{k}{2} \Delta \right) \right) n,$$

- для $r = k$

$$T_{r=k,k,a,0} = T_{r=k,k,a,e} = (1 + (k - 1)\Delta) n .$$

Приведем ряд численных результатов для этих формул, см. таблицу 2.3 и рис.2.1.

Таблица 2.3. Численные результаты

| N- Процессора | k=8 | | | | | |
|------------------|----------|----------|----------|----------|----------|----------|
| | Δ= 0.4 | | | Δ= 0.5 | | |
| | P=0.5 n | P=0.6 | P=0.7 | P=0.5 | P=0.65 | P=0.7 |
| r=1 | 2.7641 n | 3.7984 n | 5.4580 n | 2.9570 n | 4.9705 n | 6.0372 n |
| r=2 | 3.4750 n | 4.2944 n | 5.2962 n | 3.8750 n | 5.3773 n | 5.9870 n |
| r=4 | 3.5000 n | 3.7600 n | 4.0200 n | 4 n | 4.4500 n | 4.6000 n |
| r=8 | 3.8000 n | | | 4.5000 n | | |

Отметим, что минимальное время выполнения запроса на r процессорах для вероятности успеха $p < 0.65$ обеспечивается при промежуточном (не максимальном) числе процессоров $r=4$.

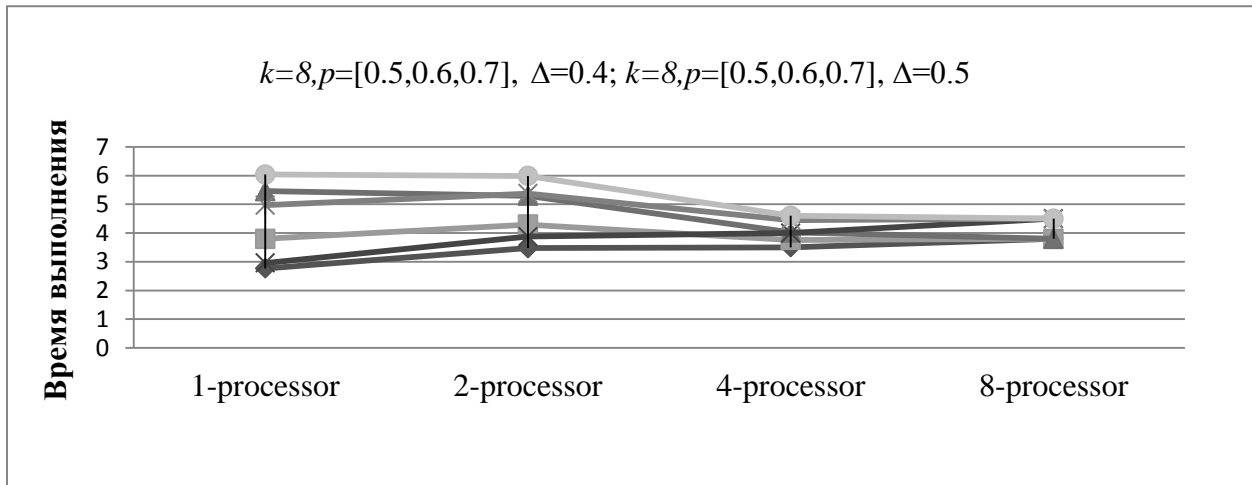


Рис.2.1. Время выполнения на i -м процессоре k элементарных запросов при квазиоптимальном порядке их распределения по процессорам для неупорядоченных столбцов таблицы

2.3.3. Обработки запроса для упорядоченных столбцов таблицы естественный порядок распределения. Арифметическая прогрессия

Пусть число процессоров вычислительной системы равно r , время выполнения элементарного запроса подчиняется закону арифметической прогрессии

$$\tau_i = 1 + \Delta(i - 1), p_i = p, i = \overline{1, k}.$$

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при естественном порядке их распределения по процессорам, когда данные в столбцах упорядочены равно:

$$T_{r,i,e,y} = p \left((1 + (i - 1)\Delta) \frac{1 - p^{\frac{k}{r}}}{1 - p} + rp\Delta \frac{1 - \frac{k}{r} p^{\frac{k}{r} - 1} + (\frac{k}{r} - 1) p^{\frac{k}{r}}}{(1 - p)^2} \right) n, i = \overline{1, r}.$$

или

$$T_{r,i,a,e,y} = p \left((1 + (i - 1) \Delta + (1 + (r + i - 1) \Delta) p \frac{1-p^{\frac{k}{r}}}{1-p^2} + 2r \Delta p^2 (1 + p) \frac{1 - \frac{k}{2r} p^{\frac{k}{r} - 2} + (\frac{k}{2r} - 1) p^{\frac{k}{r}}}{(1-p^2)^2} \right) n,$$

$i = \overline{1, r}.$

2.3.4. Квазиоптимальный порядок распределения. Арифметическая прогрессия

2.3.4.1. Время выполнения

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам, когда данные в столбцах упорядочены, равно (для простоты записей будем считать, что k/r есть целое четное число):

$$T_{r,i,a,o,y} = p \left((1 + (i - 1) \Delta) + (1 + (2r - i) \Delta) p \frac{1-p^{\frac{k}{r}}}{1-p^2} + 2r \Delta p^2 (1 + p) \frac{1 - \frac{k}{2r} p^{\frac{k}{r} - 2} + (\frac{k}{2r} - 1) p^{\frac{k}{r}}}{(1-p^r)^2} \right) n,$$

$i = \overline{1, r}.$

2.3.4.2. Соотношения времени выполнения на i -м и j -м процессорах

Для естественного порядка распределения элементарных запросов непосредственно из первого выражения для $T_{r,i,e}$ имеем неравенства:

$$T_{r,1,a,e,y} < T_{r,2,a,e,y} < \dots < T_{r,r,a,e,y}$$

Для квазиоптимального порядка распределения элементарных запросов так же выполняются аналогичные неравенства:

$$T_{r,1,a,0,y} < T_{r,2,a,0,y} < \dots < T_{r,i,a,0,y} < \dots < T_{r,j,a,0,y} < T_{r,r,a,0,y} < 1, 1 \leq i < j.$$

В самом деле, непосредственно из выражения для $T_{r,i,a,0,y}$ и $T_{r,j,a,0,y}$ имеем

$$1 + (i - 1) \Delta + (1 + (2r - i)) p < 1 + (j - 1) \Delta + (1 + (2r - j)) \Delta p,$$

или

$$p(j - i) \Delta < (j - i) \Delta,$$

или $p < 1,$

что справедливо всегда при любых $i < j.$

Максимальный разброс времени выполнения обработки запросов на r процессорах составляет:

- при **естественном** порядке

$$T_{r,r,a,e,y} - T_{r,1,a,e,y} = p(r-1)(1+p)\Delta \frac{1-p^{\frac{k}{r}}}{1-p^2} n,$$

- при **квазиоптимальном** порядке

$$T_{r,r,a,0,y} - T_{r,1,a,0,y} = p \left((r-1)\Delta(1-p) \frac{1-p^{\frac{k}{r}}}{1-p^2} \right) n.$$

Поэтому минимальная и максимальная границы времени выполнения при квазиоптимальном распределении для упорядоченных столбцов таблицы лежат внутри минимальной и максимальной границ при естественном распределении.

Абсолютное уменьшение границ времени выполнения запросов при использовании оптимального распределения вместо естественного распределения равно:

$$\delta_{abc} = p \left((r-1)\Delta \frac{1-p^{\frac{k}{r}}}{1-p^2} \right) 2pn.$$

Относительное уменьшение границ времени выполнения запросов при использовании квазиоптимального распределения вместо естественного распределения равно:

$$\delta_{отн} = \frac{\delta_{abc}}{\left((r-1)\Delta(1+p) \frac{1-p^{\frac{k}{r}}}{1-p^2} n \right)} 100\% = \left(\frac{2p}{1+p} \right) 100\%,$$

что при $p \geq 0,25$ составляет больше или равно 40 % и при $p \geq 0,5$ составляет $\geq 67\%$.

2.3.4.3. Эффективность квазиоптимального распределения

Указанные результаты характеризуют эффективность квазиоптимального распределения элементарных запросов по процессорам, т.к. это распределение обеспечивает минимальное время выполнения запроса на r процессорах.

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам равно (здесь, как и ранее, k/r есть целое четное число)

$$T_{r,i,g,o,y} = p(a^{i-1} + pa^{2r-i}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

Максимальное время завершения обработки k/r элементарных запросов на всех процессорах в частных случаях равно:

- для $r=1$

$$T_{r=1,1a,0,y} = T_{r=1,1a,e,y} = p \left(\frac{1-p^k}{1-p} + p\Delta \frac{1-kp^{k-1}+(k-1)p^k}{(1-p)^2} \right) n,$$

- для $r = \frac{k}{4}$

$$T_{r=\frac{k}{4},\frac{k}{4},a,0,y} = p \left(1 + \left(\frac{k}{4} - 1 \right) \Delta + p \left(1 + \frac{k}{4} \Delta \right) \right) (1 + p^2) + \frac{k}{2} \Delta p^2 (1 + p) n$$

- для $r = \frac{k}{2}$

$$T_{r=\frac{k}{2},\frac{k}{2},a,0,y} = p \left(1 + \left(\frac{k}{2} - 1 \right) \Delta + p \left(1 + \frac{k}{2} \Delta \right) \right) n,$$

- для $r = k$

$$T_{r=k,k,a,0,y} = T_{r=k,k,a,e,y} = p(1 + (k-1)\Delta)n.$$

Приведем ряд численных результатов для этих формул, см. таблицу 2.4:

Таблица 2.4 Численные результаты

| N- Процессора | k=8 | | | | | |
|------------------|----------|----------|----------|----------|----------|----------|
| | Δ= 0.4 | | | Δ= 0.5 | | |
| | P=0.5 | P=0.6 | P=0.7 | P=0.5 | P=0.65 | P=0.7 |
| r=1 | 1.3821n | 2.2790 n | 3.8206 n | 1.4785 n | 2.6868 n | 4.2260 n |
| r=2 | 1.7375 n | 2.5766 n | 3.7073 n | 1.9375 n | 3.1356 n | 4.1909 n |
| r=4 | 1.7500 n | 2.2560 n | 2.8140 n | 2 n | 2.7950 n | 3.2200 n |
| r=8 | 2.2800 n | | | 2.9250 n | | |

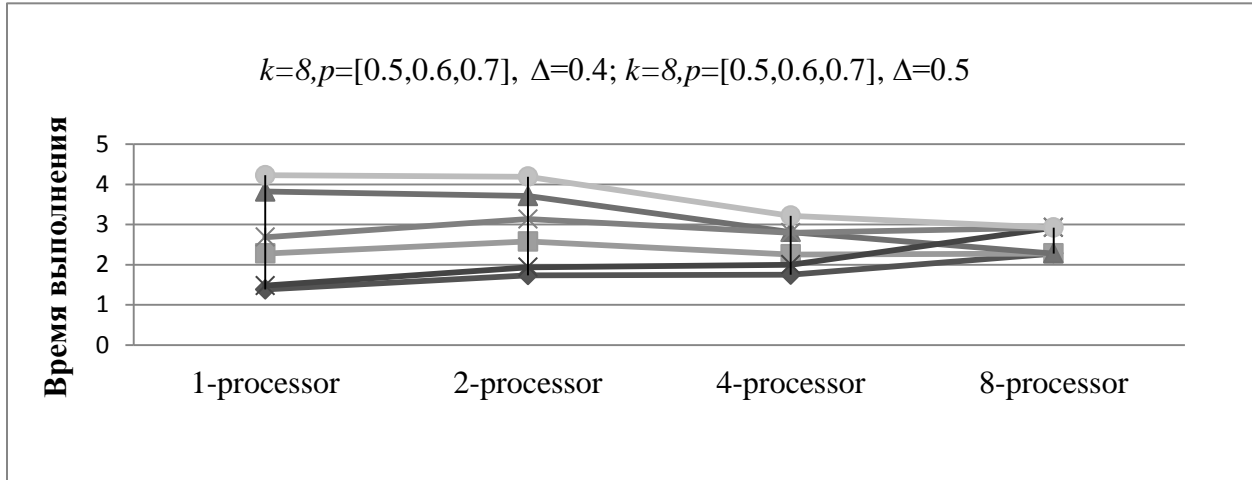


Рис.2.2. Время выполнения на i -м процессоре k элементарных запросов при квазиоптимальном порядке их распределения по процессорам для упорядоченных столбцов таблицы

2.3.5. Обработки запроса для неупорядоченных столбцов таблицы естественный порядок распределения. Геометрическая прогрессия

Пусть время выполнения элементарного запроса подчиняется закону геометрической прогрессии

$$\tau_i = a^{i-1}, a > 1, p_i = p, i = \overline{1, k};$$

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при естественном порядке их распределения по процессорам, когда данные в столбцах неупорядочены равно:

$$T_{r,i,g,e} = (a^{i-1}) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - pa^r} \right) n, i = \overline{1, r}.$$

В самом деле, в соответствии с теоремой 1[1] имеем:

$$T_{r,i,g,e} = (a^{i-1} + pa^{r+i-1} + p^2 a^{2r+i-1} + p^3 a^{3r+i-1} + p^4 a^{4r+i-1} + p^5 a^{5r+i-1} + \dots + p^{\frac{k}{r}-2} a^{k-2r+i-1} + p^{\frac{k}{r}-1} a^{k-r+i-1}) n = a^{i-1} \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - pa^r} \right) n, i = \overline{1, r}.$$

Можно представить $T_{r,i,g,e}$ в другом виде

$$T_{r,i,g,e} = (a^{i-1} + pa^{r+i-1}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

так как

$$T_{r,i,g,e} = (a^{i-1}(1+p^2a^{2r} + p^4a^{4r} + \dots + a^{k-2r}) + pa^{r+i-1}(1 + p^2a^{2r} + p^4a^{4r} + \dots + a^{k-2r})) n, i = \overline{1, r}.$$

или

$$T_{r,i,g,e} = (a^{i-1} + pa^{r+i-1}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

Очевидно, что эти два выражения идентичны, так как

$$a^{i-1} + pa^{r+i-1} = (a^{i-1})(1 + pa^r).$$

Однако, последнее выражение для $T_{r,i,g,e}$ дает возможность показать, что оптимальный алгоритм обеспечивает меньшее время вычисления запроса r процессорами.

2.3.6. Квазиоптимальный порядок распределения. Геометрическая прогрессия

2.3.6.1. Время выполнения

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам, когда данные в столбцах неупорядочены равно: (для простоты записей будем считать, что k/r есть целое четное число):

$$T_{r,i,g,o} = (a^{i-1} + pa^{2r-i}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

В самом деле, в соответствии с теоремой 1 имеем:

$$T_{r,i,g,o} = \left((a^{i-1} + pa^{2r-i} + p^2a^{2r+i-1} + p^3a^{4r-i} + p^4a^{4r+i-1} + p^5a^{6r-i} + \dots + p^{\frac{k}{r}-2}a^{k-2r+i-1} + p^{\frac{k}{r}-1}a^{k-i}) n, i = \overline{1, r} \right)$$

или

$$T_{r,i,g,o} = \left(a^{i-1} \left(1 + p^2 a^{2r} + p^4 a^{4r} + p^6 a^{6r} + \dots + p^{\frac{k}{r}-2} a^{k-2r} \right) + pa^{2r-i} \left(1 + p^2 a^{2r} + p^4 a^{4r} + \dots + p^{\frac{k}{r}-2} a^{k-2r} \right) \right) n, i = \overline{1, r}.$$

или

$$T_{r,i,g,o} = (a^{i-1} + pa^{2r-i}) \left(1 + (p^2 a^{2r})^1 + (p^2 a^{2r})^2 + (p^2 a^{2r})^3 + \dots + (p^2 a^{2r})^{\frac{k}{2r}-1} \right) n, i = \overline{1, r}.$$

или

$$T_{r,i,g,o} = (a^{i-1} + pa^{2r-i}) \left(\frac{1 - p^{\frac{k}{r}} a^k}{1 - (pa^r)^2} \right) n = (a^{i-1} + pa^{2r-i}) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, i = \overline{1, r}.$$

2.3.6.2. Соотношения времени выполнения на i -м и j -м процессорах

Для **естественного** порядка распределения элементарных запросов непосредственно из выражения для $T_{r,i,g,e}$, $i = \overline{1, r}$, $r > 1$, имеем неравенства:

$$T_{r,1,g,e} < T_{r,2,g,e} < \dots < T_{r,r,g,e}.$$

Для **квазиоптимального** порядка распределения элементарных запросов на основании выражений для $T_{r,i,g,o}$, $i = \overline{1, r}$, $r > 1$,

при $pa^2 > 1$ имеем неравенства:

$$T_{r,1,g,o} > T_{r,2,g,o} > \dots > T_{r,r-1,g,o} > T_{r,r,g,o}$$

В самом деле, неравенство $T_{r,r-1,g,o} > T_{r,r,g,o}$ выполняется тогда и только тогда, когда

$$a^{r-2} + pa^{r+1} > a^{r-1} + pa^r$$

или когда

$$pa^r(a-1) > a^{r-2}(a-1)$$

или когда

$$pa^2 > 1.$$

Неравенство $T_{r,r-2,g,o} > T_{r,r-1,g,o}$ выполняется, когда

$$pa^{r+1}(a-1) > a^{r-3}(a-1)$$

или когда

$$pa^4 > 1,$$

что выполняется при $pa^2 > 1$

и так далее.

Наконец, неравенство $T_{r,1,g,o} > T_{r,2,g,o}$ выполняется, когда

$$pa^{2r-2}(a-1) > a-1$$

или когда

$$pa^{2(r-1)} > 1,$$

что естественно выполняется при

$$pa^2 > 1.$$

Обобщая, находим, что все исходные неравенства выполняются при $pa^2 > 1$, что и т. д.

Максимальный разброс времени выполнения запросов здесь составляет:

$$T_{r,1,g,o} - T_{r,r,g,o} = (a^{r-1} - 1)(pa^r - 1) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n.$$

При $pa^2 < 1$, но при $pa^4 > 1$, выполняется следующая цепочка неравенств:

$$T_{r,1,g,o} > T_{r,2,g,o} > \dots > T_{r,r-1,g,o} < T_{r,r,g,o},$$

при этом неравенство $T_{r,1,g,o} > T_{r,r,g,o}$ выполняется, если

$$1 + pa^{2r-1} > a^{r-1} + pa^r$$

или если $pa^r > 1$ ($r \geq 4$).

Напротив, неравенство

$T_{r,1,g,o} < T_{r,r,g,o}$ выполняется, если $pa^r < 1$, ($r \leq 3$).

Аналогично, при $pa^4 < 1$, $pa^6 > 1$ имеем неравенства:

$$T_{r,1,g,o} > T_{r,2,g,o} > \dots > T_{r,r-2,g,o} < T_{r,r-1,g,o} < T_{r,r,g,o},$$

и вообще, при $pa^{2i} < 1$, $pa^{2(i+1)} > 1$, $\overline{i = 1, r-2}$, имеем:

$$T_{r,1,g,o} > T_{r,2,g,o} < \dots < T_{r,r-i,g,o} < T_{r,r-i+1,g,o} < \dots < T_{r,r,g,o}.$$

Наконец, при $pa^{2(r-1)} < 1$ имеем:

$$T_{r,1,g,o} < T_{r,2,g,o} < \dots < T_{r,r-2,g,o} < T_{r,r-1,g,o} < \dots < T_{r,r,g,o}.$$

Максимальный разброс времени выполнения обработки запросов

При $pa^{2(r-1)} < 1$

составляет: $T_{r,r,g,o} - T_{r,1,g,o} = (a^{r-1} - 1)(1 - pa^r) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n$.

Минимальная и максимальная границы времени выполнения

Имея в виду, что выполняется $T_{r,1,g,e} < T_{r,1,g,o}$,

так как $(1 + pa^r) < (1 + pa^{2r-1})$ или $1 < a^{r-1}$, при $r > 1$ и $a > 1$,

и также выполняется $T_{r,r,g,o} < T_{r,r,g,e}$,

так как $a^{r-1} + pa^r < a^r(1 + pa^r)$ или $pa(a^r - 1) + (a - 1) > 0$,

находим, что минимальная и максимальная границы времени выполнения при оптимальном распределении лежат внутри минимальной и максимальной границы при естественном распределении.

Максимальный разброс времени выполнения обработки запросов на r процессорах составляет:

- при естественном порядке

$$T_{r,r,g,e} - T_{r,1,g,e} = \left((a^{r-1} - 1) - (1 + pa^r) \frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n$$

- при квазиоптимальном порядке

$$T_{r,1,g,o} - T_{r,r,g,o} = (a^{r-1} - 1)(pa^r - 1) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \text{ при } pa^2 > 1,$$

$$T_{r,r,g,o} - T_{r,1,g,o} = (a^{r-1} - 1)(1 - pa^r) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \text{ при } pa^r < 1.$$

Абсолютное уменьшение границ времени выполнения запросов при использовании оптимального распределения вместо естественного распределения равно:

$$\delta_{\text{абсо}} = \left(((a^{r-1} - 1)(1 + pa^r) - (a^{r-1} - 1)(1 - pa^r)) \frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right)$$

$$= (a^{r-1} - 1) 2pa^r \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right).$$

Относительное уменьшение границ времени выполнения запросов при оптимальном распределении

$$\delta_{\text{отно}} = \frac{\delta_{\text{абсо}}}{\left((a^{r-1}-1)(1+pa^r) \frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right)} = \frac{2pa^r}{(1+pa^r)}.$$

2.3.6.3. Эффективность квазиоптимального распределения

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам равно (здесь, как и ранее, k/r есть целое четное число)

$$T_{r,i,g,o} = (a^{i-1} + pa^{2r-i}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

Максимальное время завершения обработки k/r элементарных запросов на всех процессорах в частных случаях равно:

- для $r=1$

$$T_{1,1,g,o} = \left(\frac{1-(pa)^k}{1-pa} \right) n,$$

- для $r = \frac{k}{4}$

при $pa^2 > 1$

$$T_{r=\frac{k}{4},1,g,o} = \left((1 + pa^{k/2-1})(1 + (pa^{k/4})^2) \right) n,$$

при $pa^{2r-2} < 1$

$$T_{r=\frac{k}{4},\frac{k}{4},g,o} = \left(\left(a^{\frac{k}{4}-1} + pa^{\frac{k}{4}} \right) \left(1 + \left(pa^{\frac{k}{4}} \right)^2 \right) \right) n,$$

- для $r = \frac{k}{2}$

при $pa^2 > 1$

$$T_{r=\frac{k}{2},1,g,o} = (1 + pa^{k-1})n,$$

при $pa^{2r-2} < 1$

$$T_{r=\frac{k}{2},\frac{k}{2},g,o} = \left(a^{\frac{k}{2}-1}(1 + pa) \right) n,$$

- для $r = k$

$$T_{r=k,k,g,e} = (a^{k-1})n .$$

Приведем ряд численных результатов для этих формул при $k=8$; $a=1.15, 1.2, 1.25, 1.3$; $p=0.6, 0.7$, см. таблицу 2.5:

Таблица 2.5 Численные результаты

| N-Процессора | k=8 | | | |
|--------------|----------|----------|----------|----------|
| | a = 1.15 | | a = 1.2 | |
| | P=0.6 | P=0.7 | P=0.6 | P=0.7 |
| r=1 | 3.0601n | 4.2239 n | 3.3135 n | 4.7008 n |
| r =2 | 3.1167 n | 3.8340 n | 3.5573 n | 4.4547 n |
| r =4 | 2.5960 n | 2.8620 n | 3.1499 n | 3.5082 n |
| r =k | 2.6600 n | | 3.5832 n | |
| N-Процессора | k=8 | | | |
| | a = 1.25 | | a = 1.3 | |
| | P=0.6 | P=0.7 | P=0.6 | P=0.7 |
| r=1 | 3.5995 n | 5.2511 n | 3.9227 n | 5.8861 n |
| r =2 | 4.0807 n | 5.1990 n | 4.7018 n | 6.0897 n |
| r =4 | 3.8610 n | 4.3379 n | 4.7649 n | 5.3924 n |
| r =k | 4.7684 n | | 6.2749 n | |

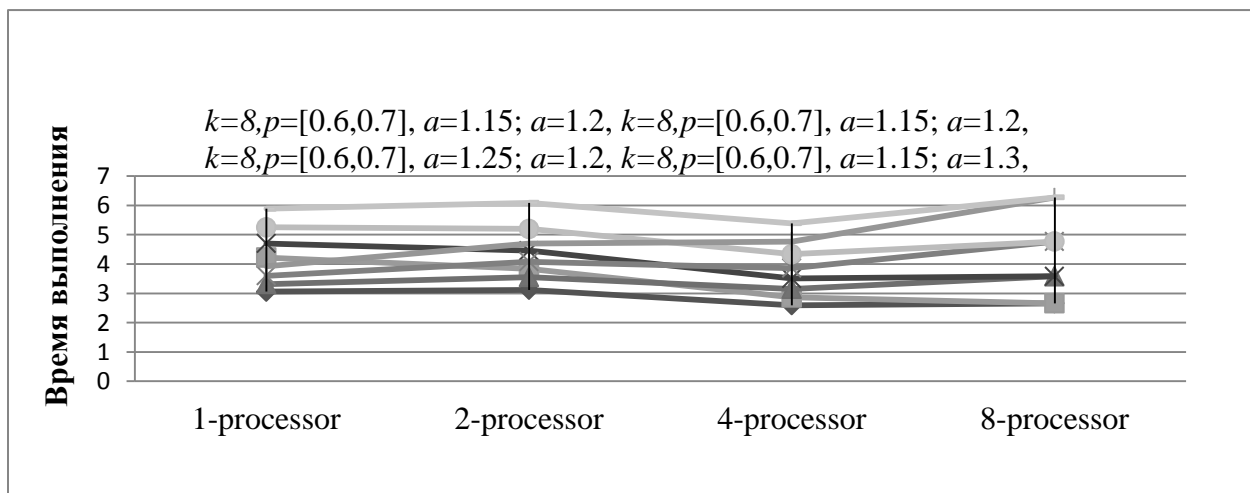


Рис.2.3. Время выполнения на i -м процессоре k элементарных запросов при квазиоптимальном порядке их распределения по процессорам для неупорядоченных столбцов таблицы

2.3.7. Обработки запроса для упорядоченных столбцов таблицы естественный порядок распределения. Геометрическая прогрессия

Пусть время выполнения элементарного запроса подчиняется закону геометрической прогрессии

$$\tau_i = a^{i-1}, a > 1, p_i = p, i = \overline{1, k};$$

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при естественном порядке их распределения по процессорам, когда в столбцах упорядочены равно:

$$T_{r,i,g,e,y} = p(a^{i-1}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-pa^r} \right) n, i = \overline{1, r}.$$

Можно представить $T_{r,i,g,e,y}$ в другом виде

$$T_{r,i,g,e,y} = p(a^{i-1} + pa^{r+i-1}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

Очевидно, что эти два выражения идентичны, так как

$$a^{i-1} + pa^{r+i-1} = (a^{i-1})(1 + pa^r).$$

Однако, последнее выражение для $T_{r,i,g,e,y}$ дает возможность показать, что оптимальный алгоритм обеспечивает меньшее время вычисления запроса r процессорами.

2.3.8. Квазиоптимальный порядок распределения. Геометрическая прогрессия

2.3.8.1. Время выполнения

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам, когда данные в столбцах упорядочены равно (для простоты записей будем считать, что k/r есть целое четное число):

$$T_{r,i,g,o,y} = p(a^{i-1} + pa^{2r-i}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

2.3.8.2. Соотношения времени выполнения на i -м и j -м процессорах

Для **естественного** порядка распределения элементарных запросов непосредственно из выражения для $T_{r,i,g,e,y}$, $i = \overline{1, r}$, $r > 1$, имеем неравенства:

$$T_{r,1,g,e,y} < T_{r,2,g,e,y} < \dots < T_{r,r,g,e,y}$$

Для **квазиоптимального** порядка распределения элементарных запросов на основании выражений для $T_{r,i,g,o,y}$, $i = \overline{1, r}$, $r > 1$,

при $pa^2 > 1$ имеем неравенства:

$$T_{r,1,g,o,y} > T_{r,2,g,o,y} > \dots > T_{r,r-1,g,o,y} > T_{r,r,g,o,y}$$

В самом деле, неравенство $T_{r,r-1,g,o,y} > T_{r,r,g,o,y}$ выполняется тогда и только тогда, когда

$$a^{r-2} + pa^{r+1} > a^{r-1} + pa^r$$

или когда

$$pa^r(a-1) > a^{r-2}(a-1)$$

или когда

$$pa^2 > 1.$$

Неравенство $T_{r,r-2,g,o,y} > T_{r,r-1,g,o,y}$ выполняется, когда

$$pa^{r+1}(a-1) > a^{r-3}(a-1)$$

или когда

$$pa^4 > 1,$$

что выполняется при $pa^2 > 1$

и так далее.

Наконец, неравенство $T_{r,1,g,o,y} > T_{r,2,g,o,y}$ выполняется, когда

$$pa^{2r-2}(a-1) > a-1$$

или когда

$$pa^{2(r-1)} > 1,$$

что естественно выполняется при

$$pa^2 > 1.$$

Обобщая, находим, что все исходные неравенства выполняются при $pa^2 > 1$, что и т. д.

Максимальный разброс времени выполнения запросов здесь составляет:

$$T_{r,1,g,o,y} - T_{r,r,g,o,y} = p(a^{r-1} - 1)(pa^r - 1) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \overline{1, r}.$$

При $pa^2 < 1$, но при $pa^4 > 1$, выполняется следующая цепочка неравенств:

$$T_{r,1,g,o,y} > T_{r,2,g,o,y} > \dots > T_{r,r-1,g,o,y} < T_{r,r,g,o,y}$$

при этом неравенство $T_{r,1,g,o,y} > T_{r,r,g,o,y}$ выполняется, если

$$1 + pa^{2r-1} > a^{r-1} + pa^r$$

или если $pa^r > 1$ ($r \geq 4$).

Напротив, неравенство

$$T_{r,1,g,o,y} < T_{r,r,g,o,y} \text{ выполняется, если } pa^r < 1, (r \leq 3).$$

Аналогично, при $pa^4 < 1, pa^6 > 1$ имеем неравенства:

$$T_{r,1,g,o,y} > T_{r,2,g,o,y} > \dots > T_{r,r-2,g,o,y} < T_{r,r-1,g,o,y} < T_{r,r,g,o,y}$$

и вообще, при $pa^{2i} < 1, pa^{2(i+1)} > 1, \overline{i, r-2}$, имеем:

$$T_{r,1,g,o,y} > T_{r,2,g,o,y} < \dots < T_{r,r-i,g,o,y} < T_{r,r-i+1,g,o,y} < \dots < T_{r,r,g,o,y}.$$

Наконец, при $pa^{2(r-1)} < 1$ имеем:

$$T_{r,1,g,o,y} < T_{r,2,g,o,y} < \dots < T_{r,r-2,g,o,y} < T_{r,r-1,g,o,y} < \dots < T_{r,r,g,o,y}.$$

Максимальный разброс времени выполнения обработки запросов при $pa^{2(r-1)} < 1$

составляет:

$$T_{r,r,g,o,y} - T_{r,1,g,o,y} = p(a^{r-1} - 1)(1 - pa^r) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \overline{1, r}.$$

Минимальная и максимальная границы времени выполнения

Поэтому минимальная и максимальная границы времени выполнения при оптимальном распределении лежат внутри минимальной и максимальной границы при естественном распределении.

Максимальный разброс времени выполнения обработки запросов на r процессорах составляет:

- при естественном порядке

$$T_{r,r,g,e,y} - T_{r,1,g,e,y} = p \left((a^{r-1} - 1) - (1 + pa^r) \frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \overline{1, r}.$$

- при квазиоптимальном порядке

$$T_{r,1,g,o,y} - T_{r,r,g,o,y} = p(a^{r-1} - 1)(pa^r - 1) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \text{ при } pa^2 > 1,$$

$$T_{r,r,g,o,y} - T_{r,1,g,o,y} = p(a^{r-1} - 1)(1 - pa^r) \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n, \text{ при } pa^r < 1.$$

Абсолютное уменьшение границ времени выполнения запросов при использовании оптимального распределения вместо естественного распределения равно:

$$\delta_{\text{абсо}} = p(a^{r-1} - 1)2pa^r \left(\frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right) n.$$

Относительное уменьшение границ времени выполнения запросов при оптимальном распределении

$$\delta_{\text{отно}} = \frac{\delta_{\text{абсо}}}{\left((a^{r-1} - 1)(1 + pa^r) \frac{1 - (pa^r)^{\frac{k}{r}}}{1 - (pa^r)^2} \right)} = \frac{2pa^r}{(1 + pa^r)}.$$

2.3.8.3. Эффективность квазиоптимального распределения

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по

процессорам, когда данных в столбцах упорядочены равно (здесь, как и ранее, k/r есть целое четное число)

$$T_{r,i,g,o,y} = p(a^{i-1} + pa^{2r-i}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r}.$$

Максимальное время завершения обработки k/r элементарных запросов на всех процессорах в частных случаях равно:

- для $r=1$

$$T_{1,1,g,o,y} = p \left(\frac{1-(pa)^k}{1-pa} \right) n,$$

- для $r = \frac{k}{4}$

при $pa^2 > 1$

$$T_{r=\frac{k}{4},1,g,o,y} = p \left((1 + pa^{k/2-1})(1 + (pa^{k/4})^2) \right) n,$$

при $pa^{2r-2} < 1$

$$T_{r=\frac{k}{4},g,o,y} = p \left(\left(a^{\frac{k}{4}-1} + pa^{\frac{k}{4}} \right) \left(1 + \left(pa^{\frac{k}{4}} \right)^2 \right) \right) n,$$

- для $r = \frac{k}{2}$

при $pa^2 > 1$

$$T_{r=\frac{k}{2},1,g,o,y} = p(1 + pa^{k-1})n,$$

при $pa^{2r-2} < 1$

$$T_{r=\frac{k}{2},g,o,y} = p \left(a^{\frac{k}{2}-1}(1 + pa) \right) n,$$

- для $r = k$

$$T_{r=k,k,g,e,y} = p(a^{k-1})n.$$

Приведем ряд численных результатов для этих формул при $k=8$; $a=1.15, 1.2, 1.25, 1.3$; $p=0.6, 0.7$, см. таблицу 2.6:

Таблица 2.6 Численные результаты

| N-Процессора | k=8 | | | |
|--------------|----------|---------|---------|-------|
| | a = 1.15 | | a = 1.2 | |
| | P=0.6 | P=0.7 n | P=0.6 | P=0.7 |
| | | | | |

| | | | | |
|--------------|-----------|-----------|-----------|-----------|
| $r=1$ | $1.8361n$ | $2.9567n$ | $1.9881n$ | $3.2906n$ |
| $r=2$ | $1.8700n$ | $2.6838n$ | $2.1344n$ | $3.1183n$ |
| $r=4$ | $1.5576n$ | $2.0034n$ | $1.8899n$ | $2.4557n$ |
| $r=k$ | $1.8620n$ | | $2.1499n$ | |
| N-Процессора | $k=8$ | | | |
| | $a=1.25$ | | $a=1.3$ | |
| | P=0.6 | P=0.7 | P=0.6 | P=0.7 |
| $r=1$ | $2.1597n$ | $3.6758n$ | $2.3536n$ | $4.1203n$ |
| $r=2$ | $2.4484n$ | $3.6393n$ | $2.8211n$ | $4.2628n$ |
| $r=4$ | $2.3166n$ | $3.0365n$ | $2.8589n$ | $3.7747n$ |
| $r=k$ | $3.3379n$ | | $3.7649n$ | |

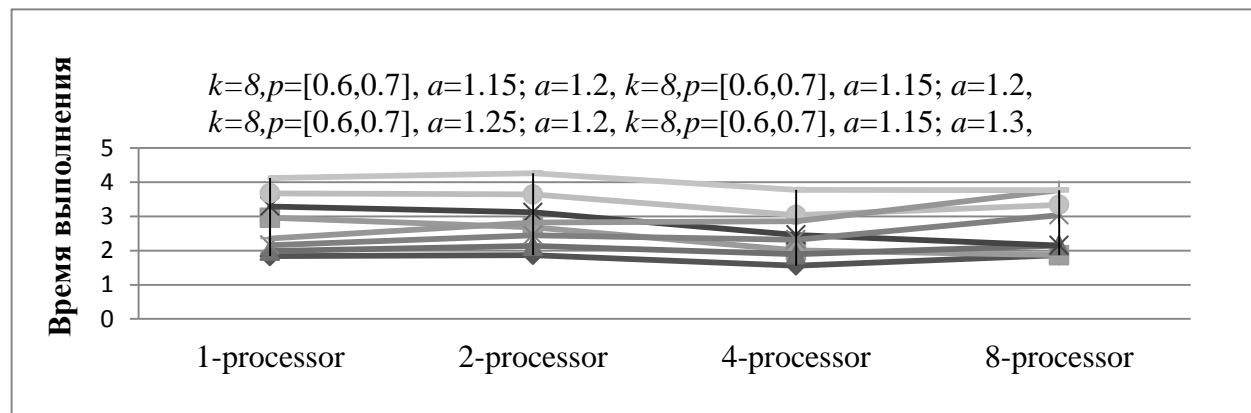


Рис.2.4. Время выполнения на i -м процессоре k элементарных запросов при квазиоптимальном порядке их распределения по процессорам для неупорядоченных столбцов таблицы

Выводы по главе 2

1. Определены соотношения времени выполнения вложенного запроса в многопроцессорной базе данных для естественного и квазиоптимального порядка их распределения для неупорядоченных и для упорядоченных столбцов таблицы.

2. Определены минимальная и максимальная границы времени выполнения групп элементарных запросов в отдельных процессорах для естественного и квазиоптимального порядка распределения элементарных запросов для неупорядоченных и для упорядоченных столбцов таблицы.
3. Доказана эффективность квазиоптимального распределения на основе абсолютного и относительного уменьшения границ времени выполнения запросов при использовании квазиоптимального распределения вместо естественного распределения, что является важным решением для оптимизации обработки запросов в многопроцессорных базах данных авиационно-космических систем.

3. Оптимизация числа процессоров при выполнении вложенных запросов

3.1. Введение

В работе [10] предложен план оптимизации по времени выполнения конъюнктивных вложенных запросов при обращении к однопроцессорной базе данных на основе упорядочивания элементарных запросов.

Минимальное время вложенного запроса для неупорядоченных данных таблиц достигается при совместной обработке объединенного множества элементарных запросов в порядке, определяемым условиями теоремы 1, и для упорядоченных данных таблиц достигается при совместной обработке объединенного множества элементарных запросов в порядке, определяемым условиями теоремы 2.

В работе методика формирования плана оптимизации развита на случай обработки вложенных запросов многопроцессорными базами данных, что является одной из актуальных задач разработки методики оптимизации обработки запросов в многопроцессорных базах данных авиационно-космических систем.

3.2. Квазиоптимальное распределение номеров элементарных запросов по процессорам

В работе [9] предложен, а в работе [10] обоснован квазиоптимальный метод оптимизации, когда распределение элементарных запросов по процессорам осуществляется в соответствии с правилом, показанным в таблице 3.1, где в i -ой строке таблицы указаны номера элементарных запросов выполняемых i -м процессором в порядке слева направо, при этом мы используем (для простоты изложения) в качестве значения числа элементарных запросов k целое четное число, и число процессоров r отвечает условию $[k/r]r=k$. В этом случае i -ый ($i = 1, \dots, r$) процессор получает элементарные запросы с номерами $i, 2r + 1 - i, 2r + i, 4r + 1 - i, 4r + i, 6r + 1 - i, 6r + i, \dots$

Таблица 3.1. Квазиоптимальное распределение элементарных запросов по процессорам

| Номера процессоров | Номера элементарных запросов | | | | | |
|--------------------|------------------------------|--------|------|--------|---|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1 | 2r | 2r+1 | 4r | | k |
| 2 | 2 | 2r-1 | 2r+2 | 4r-1 | | k-1 |
| 3 | 3 | 2r-2 | 2r+3 | 4r-2 | | k-2 |
| | | | | | | |
| i | i | 2r-i+1 | 2r+i | 4r-i+1 | | k-i+1 |
| | | | | | | |
| r-1 | r-1 | r+2 | 3r-1 | 3r+2 | | k-r+2 |
| r | r | r+1 | 3r | 3r+1 | | k-r+1 |

Время выполнения на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам равно (для простоты записей предполагается, что k/r есть целое четное число) [5]:

$$T_{r,i,a,o} = \left((1 + (i-1)\Delta) + (1 + (2r-i)\Delta)p \frac{1-p^{\frac{k}{r}}}{1-p^2} + 2r\Delta p^2 (1+p) \frac{1-\frac{k}{2r}p^{\frac{k}{r}-2} + (\frac{k}{2r}-1)p^{\frac{k}{r}}}{(1-p^r)^2} \right) n,$$

$$i = \overline{1, r},$$

$$\text{если } \tau_i = 1 + \Delta(i-1), p_i = p, i = \overline{1, r};$$

или

$$T_{r,i,g,o} = (a^{i-1} + pa^{2r-i}) \left(\frac{1-(pa^r)^{\frac{k}{r}}}{1-(pa^r)^2} \right) n, i = \overline{1, r},$$

$$\text{если } \tau_i = a^{i-1}, a > 1, p_i = p, i = \overline{1, k};$$

3.3. Минимизация времени обработки вложенного запроса в многопроцессорной базе данных

В многопроцессорной базе данных минимальное время выполнения конъюнктивных вложенных запросов определяется теоремами 3 и 4.

Теорема 3. В многопроцессорной базе данных минимальное время выполнения вложенного запроса, для неупорядоченных данных таблиц достигается при совместной обработке i -ым ($i = 1, \dots, r$) процессором объединенного множества элементарных запросов всех таблиц. При этом элементарные запросы в соответствии с квазиоптимальным методом оптимизации распределены по процессорам с номерами $i, 2r + 1 - i, 2r + i, 4r + 1 - i, 4r + i, 6r + 1 - i, 6r + i, \dots$ в порядке, определяемым условиями теоремы 1.

В самом деле. Пусть $t_{i,j}$ – время обработки i -ым ($i = 1, \dots, r$) процессором j -го элементарного запроса, $j = (i, 2r + 1 - i, 2r + i, 4r + 1 - i, 4r + i, 6r + 1 - i, \dots, k - i + 1)$, $p_{i,j}$ – вероятность успеха при обработке j -го элементарного запроса.

В соответствии с Теоремой 1 время обработки элементарных запросов в порядке, заданным условием $\frac{t_{i,j}}{1-p_{i,j}} < \frac{t_{i,l}}{1-p_{i,l}}$, где j и l – номера соседних элементарных запросов, является минимальным и определяется выражением:

$$\begin{aligned} & (t_{i,i} + p_{i,i}t_{i,2r-i+1} + p_{i,i}p_{i,2r-i-1}t_{i,2r+i} + p_{i,i}p_{i,2r-i+1}p_{i,2r+i}t_{i,4r-i+1} + \dots \\ & + p_{i,i}p_{i,2r-i+1}p_{i,2r+i} \dots p_{i,k-2r+i} \dots t_{i,k-i+1})n \end{aligned}$$

Теорема 4. В многопроцессорной базе данных минимальное время выполнения вложенного запроса, для упорядоченных данных таблиц достигается при совместной обработке i -ым ($i = 1, \dots, r$) процессором объединенного множества элементарных запросов всех таблиц. При этом элементарные запросы в соответствии с квазиоптимальным методом оптимизации распределены по процессорам с номерами $i, 2r + 1 - i, 2r + i, 4r + 1 - i, 4r + i, 6r + 1 - i, 6r + i, \dots$ в порядке, определяемым условиями теоремы 2.

В соответствии с теоремой 2 время обработки элементарных запросов в порядке, заданным условием $\frac{p_{i,j}t_{i,j}}{1-p_{i,j}} < \frac{p_{i,l}t_{i,l}}{1-p_{i,l}}$, где j и l – номера соседних элементарных запросов, является минимальным и определяется выражением:

$$p_{i,i}(t_{i,i} + p_{i,2r-i+1}t_{i,2r-i+1} + p_{i,2r-i+1}p_{i,2r+i}t_{i,2r+i} + \dots + p_{i,2r-i+1}p_{i,2r+i} \dots p_{i,k-i+1}t_{i,k-i+1})n$$

3.4. Численные результаты

Рассмотрим ряд числовых примеров для неупорядоченных и упорядоченных таблиц, для которых выполняются вложенные запросы, с параметрами, соответствующими геометрической и арифметической прогрессий.

3.4.1. Вложенные запросы для таблиц с параметрами, соответствующими геометрической прогрессии

3.4.1.1. Несовместное выполнение вложенных запросов

Пусть заданы четыре таблицы (T1, T2, T3, T4), для которых выполняются вложенные запросы, с параметрами, соответствующими геометрической прогрессии, где $\tau_{i,j} = a_j^{i-1}$ - время обработки i -го ЭЗ, $p_{i,j} = p_j, j \in \overline{1, k_j}$, для таблицы Tj.

Выполнение вложенных запросов может осуществляться 24 последовательными вариантами этих таблиц: T1, T2, T3, T4; T1, T2, T4, T3; T1, T3, T2, T4 и т.д. с сохранением внутреннего порядка элементарных запросов указанных таблиц в соответствии с условиями теоремы 1.

Время выполнения вложенных запросов, например, для варианта неупорядоченных таблиц T1, T2, T3, T4 для r процессорной базы данных определяется выражением:

$$T_{1,2,3,4} = (a_1^{i-1} + p_1 a_1^{2r-i}) \left(\frac{1 - (p_1 a_1^r)^{\frac{k_1}{r}}}{1 - (p_1 a_1^r)^2} \right) + p_1^{k_1} (a_2^{i-1} + p_2 a_2^{2r-i}) \left(\frac{1 - (p_2 a_2^r)^{\frac{k_2}{r}}}{1 - (p_2 a_2^r)^2} \right) + p_1^{k_1} p_2^{k_2} (a_3^{i-1} + p_3 a_3^{2r-i}) \left(\frac{1 - (p_3 a_3^r)^{\frac{k_3}{r}}}{1 - (p_3 a_3^r)^2} \right) + p_1^{k_1} p_2^{k_2} p_3^{k_3} (a_4^{i-1} + p_4 a_4^{2r-i}) \left(\frac{1 - (p_4 a_4^r)^{\frac{k_4}{r}}}{1 - (p_4 a_4^r)^2} \right). n, i = \overline{1, r},$$

Для числовых значений таблиц T1, T2, T3, T4, заданных в таблице 3.2:

Таблица 3.2. Параметры таблиц T1, T2, T3, T4

| | | k=8 | | | | | | | | |
|----|-------|-------|---|------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Tj | p_j | a_j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| T1 | 0.82 | 1.15 | 1 | 1.15 | $(1.15)^2=1.32$ | $(1.15)^3=1.52$ | $(1.15)^4=1.74$ | $(1.15)^5=2.01$ | $(1.15)^6=2.31$ | $(1.15)^7=2.66$ |
| T2 | 0.82 | 1.2 | 1 | 1.2 | $(1.2)^2=1.44$ | $(1.2)^3=1.7$ | $(1.2)^4=2.07$ | $(1.2)^5=2.48$ | $(1.2)^6=2.98$ | $(1.2)^7=3.58$ |
| T3 | 0.83 | 1.25 | 1 | 1.25 | $(1.25)^2=1.56$ | $(1.25)^3=1.95$ | $(1.25)^4=2.44$ | $(1.25)^5=3.05$ | $(1.25)^6=3.81$ | $(1.25)^7=4.76$ |
| T4 | 0.83 | 1.3 | 1 | 1.3 | $(1.3)^2=1.69$ | $(1.3)^3=2.19$ | $(1.3)^4=2.85$ | $(1.3)^5=3.71$ | $(1.3)^6=4.82$ | $(1.3)^7=6.27$ |

имеем следующие результаты для неупорядоченных таблиц при $r=1$:

$$T_{1,2,3,4} = \left(\left(\frac{1 - ((1.15) * (0.82))^8}{1 - (1.15) * (0.82)} \right) + (0.82)^8 * \left(\frac{1 - ((1.2) * (0.82))^8}{1 - (1.2) * (0.82)} \right) + (0.82)^8 * (0.82)^8 * \left(\frac{1 - ((1.25) * (0.838))^8}{1 - (1.25) * (0.838)} \right) + (0.82)^8 * (0.82)^8 * (0.838)^8 * \left(\frac{1 - ((1.3) * (0.838))^8}{1 - (1.3) * (0.838)} \right) \right) = 8.6274n$$

Аналогично,

$$T_{2,1,3,4} = \left(\left(\frac{1 - ((1.2) * (0.82))^8}{1 - (1.2) * (0.82)} \right) + (0.82)^8 * \left(\frac{1 - ((1.15) * (0.82))^8}{1 - (1.15) * (0.82)} \right) + (0.82)^8 * (0.82)^8 * \left(\frac{1 - ((1.25) * (0.838))^8}{1 - (1.25) * (0.838)} \right) + (0.82)^8 * (0.82)^8 * (0.838)^8 * \left(\frac{1 - ((1.3) * (0.838))^8}{1 - (1.3) * (0.838)} \right) \right) = 9.4171n$$

Ряд этих и других значений времени выполнения вложенных запросов приведены в таблице 3.3.

Таблица 3.3. Время выполнения вложенных запросов при $r = 1$

| Варианты | Время вариантов |
|----------------|------------------|
| T1, T2, T3, T4 | 8.6274 <i>n</i> |
| T2, T1, T3, T4 | 9.4171 <i>n</i> |
| T1, T3, T2, T4 | 8.9361 <i>n</i> |
| T3, T2, T1, T4 | 11.2358 <i>n</i> |
| T1, T4, T2, T3 | 9.2354 <i>n</i> |
| T4, T1, T2, T3 | 12.7606 <i>n</i> |

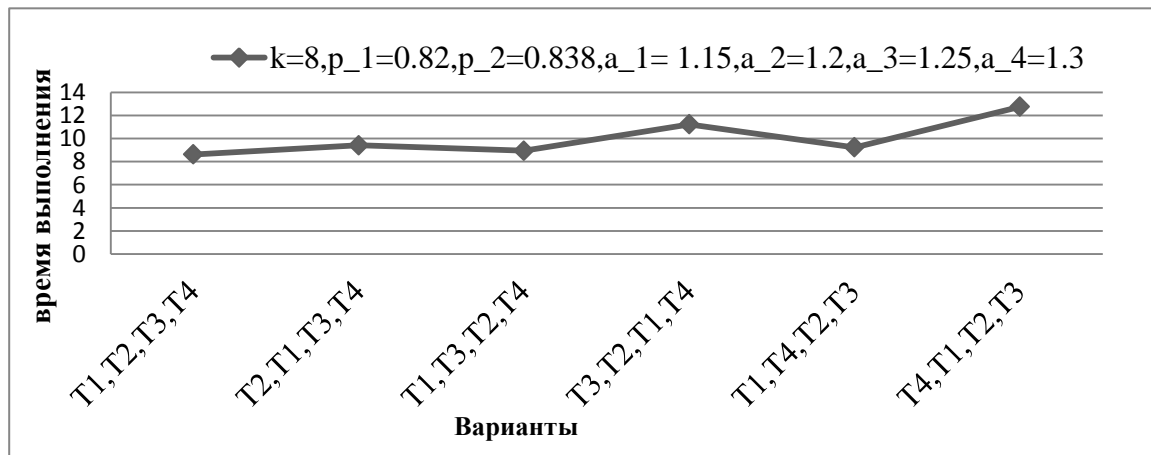


Рис.3.1. Время выполнения вложенных запросов при $r = 1$ последовательными вариантами

Ясно, что, так как в примере таблицы 2 $a_1 < a_1 < a_1 < a_1$, то для неупорядоченных таблиц в соответствии с теоремой 1 минимальное время выполнения вложенных запросов достигается при обработке таблиц T_j в порядке T_1, T_2, T_3, T_4 (время $T_{1,2,3,4} = 8.6274 n$).

Время выполнения вложенных запросов для последовательного варианта выполнения неупорядоченных таблиц в порядке T_1, T_2, T_3, T_4 для различных r на i -м процессоре k/r элементарных запросов из общего числа k элементарных запросов при квазиоптимальном порядке их распределения по процессорам равно (для k/r , равным целому четному числу) приведены в таблице 3.4, где в паре $r=i,j$ обозначены i – число процессоров, j – номер одного из процессоров.

Таблица 3.4. Последовательное выполнение неупорядоченных таблиц T_1, T_2, T_3, T_4

| Процессы $r=i,j$ | $k=32$ | Время выполнения i -ом процессор/ n | Максимальное время выполнения/ n |
|---------------------|---|--|---------------------------------------|
| | $p_1=0.82, p_2 = 0.838, a_1=1.15, a_2 = 1.2, a_3 = 1.25, a_4=1.3$ | | |
| | Порядок выполнения ЭЗ | | |
| $r=1,1$ | 1, 1.15, (1.15) ² , (1.15) ³ , (1.15) ⁴ , (1.15) ⁵ , (1.15) ⁶ , (1.15) ⁷ , 1, 1.2, (1.2) ² , (1.2) ³ , (1.2) ⁴ , (1.2) ⁵ , (1.2) ⁶ , (1.2) ⁷ , 1, 1.25, (1.25) ² , (1.25) ³ , (1.25) ⁴ , (1.25) ⁵ , (1.25) ⁶ , (1.25) ⁷ , 1, 1.3, (1.3) ² , (1.3) ³ , (1.3) ⁴ , (1.3) ⁵ , (1.3) ⁶ , (1.3) ⁷ | 10.2142 | 10.2142 |
| $r=2,1$ | 1, (1.15) ³ , (1.15) ⁴ , (1.15) ⁷ , 1, (1.2) ³ , (1.2) ⁴ , (1.2) ⁷ , 1, (1.25) ³ , (1.25) ⁴ , (1.25) ⁷ , 1, (1.3) ³ , (1.3) ⁴ , (1.3) ⁷ | 9.8301 | 9.8301 |
| $r=2,2$ | 1.15, (1.15) ² , (1.15) ⁵ , (1.15) ⁶ , 1.2, (1.2) ² , (1.2) ⁵ , (1.2) ⁶ , 1.25, (1.25) ² , (1.25) ⁵ , (1.25) ⁶ , 1.3, (1.3) ² , (1.3) ⁵ | 9.6826 | |
| $r=4,1$ | 1, (1.15) ⁷ , 1, (1.2) ⁷ , 1, (1.25) ⁷ , 1, (1.3) ⁷ | 10.0750 | 10.0750 |
| $r=4,2$ | 1.15, (1.15) ⁶ , 1.2, (1.2) ⁶ , 1, (1.25) ⁶ , 1.3, (1.3) ⁶ | 9.2074 | |
| $r=4,3$ | (1.15) ² , (1.15) ⁵ , (1.2) ² , (1.2) ⁵ , (1.25) ² , (1.25) ⁵ , (1.3) ² , (1.3) ⁵ | 8.6992 | |
| $r=4,4$ | (1.15) ³ , (1.15) ⁴ , (1.2) ³ , (1.2) ⁴ , (1.25) ³ , (1.25) ⁴ , (1.3) ³ , (1.3) ⁴ | 8.5258 | |

| | | | |
|-----------|---|--------|--------|
| $r=8,1$ | 1, $(1.2)^7$, 1, $(1.3)^7$, | 8.1463 | 8.1463 |
| $r=8,2$ | 1.15, $(1.2)^6$, 1.25, $(1.3)^6$, | 7.1588 | |
| $r=8,3$ | $(1.15)^2$, $(1.2)^5$, $(1.25)^2$, $(1.3)^5$, | 6.5057 | |
| $r=8,4$ | $(1.15)^3$, $(1.2)^4$, $(1.25)^3$, $(1.3)^4$, | 6.1438 | |
| $r=8,5$ | $(1.15)^4$, $(1.2)^3$, $(1.25)^4$, $(1.3)^3$, | 6.0455 | |
| $r=8,6$ | $(1.15)^5$, $(1.2)^2$, $(1.25)^5$, $(1.3)^2$, | 6.1964 | |
| $r=8,7$ | $(1.15)^6$, 1.2, $(1.25)^6$, 1.3 | 6.5946 | |
| $r=8,8$ | $(1.15)^7$, 1, $(1.25)^7$, 1 | 7.2497 | |
| $r=16,1$ | 1, $(1.3)^7$, | 6.1454 | 6.1454 |
| $r=16,2$ | 1.15, $(1.3)^6$ | 5.1080 | |
| $r=16,3$ | $(1.15)^2$, $(1.3)^5$, | 4.3671 | |
| $r=16,4$ | $(1.15)^3$, $(1.3)^4$, | 3.8629 | |
| $r=16,5$ | $(1.15)^4$, $(1.3)^3$, | 3.5505 | |
| $r=16,6$ | $(1.15)^5$, $(1.3)^2$, | 3.3972 | |
| $r=16,7$ | $(1.15)^6$, (1.3) | 3.3791 | |
| $r=16,8$ | $(1.15)^7$, 1 | 3.4800 | |
| $r=16,9$ | 1, $(1.25)^7$ | 4.9101 | |
| $r=16,10$ | (1.2) , $(1.25)^6$, | 4.3281 | |
| $r=16,11$ | $(1.2)^2$, $(1.25)^5$, | 3.9424 | |
| $r=16,12$ | $(1.2)^3$, $(1.25)^4$, | 3.7300 | |
| $r=16,13$ | $(1.2)^4$, $(1.25)^3$, | 3.6752 | |
| $r=16,14$ | $(1.2)^5$, $(1.25)^2$, | 3.7696 | |
| $r=16,15$ | $(1.2)^6$, (1.25) | 4.0110 | |
| $r=16,16$ | $(1.2)^7$, 1 | 4.4032 | |
| $r=32,32$ | 1 до 32 | 6.27 | 6.27 |

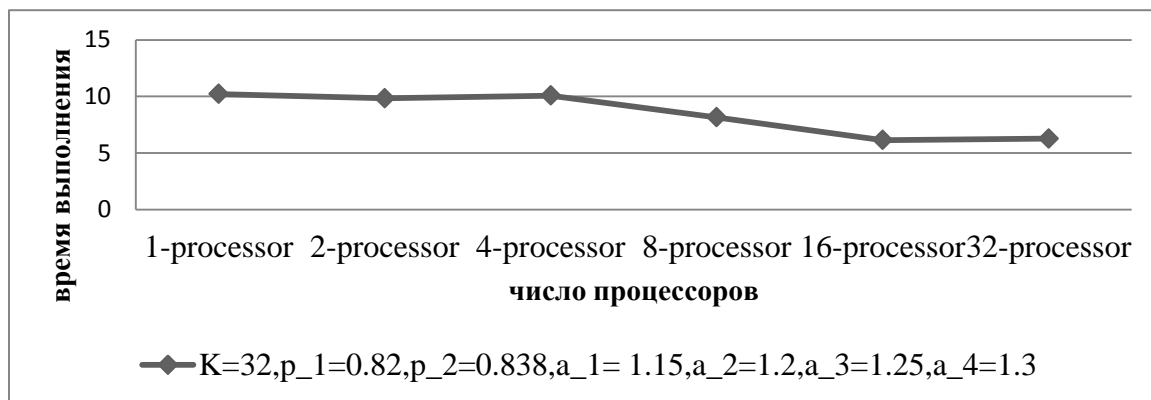


Рис.3.2. Последовательное выполнение неупорядоченных таблиц T1, T2, T3, T4 при несовместной обработке соответствующими геометрической прогрессии.

Время выполнения вложенных запросов для упорядоченных таблиц T1, T2, T3, T4 при последовательном их выполнении для r процессорной базы данных определяется выражением:

$$T_{1,2,3,4} = p_1(a_1^{i-1} + p_1 a_1^{2r-i}) \left(\frac{1-(p_1 a_1^r)^{\frac{k_1}{r}}}{1-(p_1 a_1^r)^2} \right) + p_1^{k_1} p_2 (a_2^{i-1} + p_2 a_2^{2r-i}) \left(\frac{1-(p_2 a_2^r)^{\frac{k_2}{r}}}{1-(p_2 a_2^r)^2} \right) +$$

$$p_1^{k_1} p_2^{k_2} p_3 (a_3^{i-1} + p_3 a_3^{2r-i}) \left(\frac{1-(p_3 a_3^r)^{\frac{k_3}{r}}}{1-(p_3 a_3^r)^2} \right) +$$

$$p_1^{k_1} p_2^{k_2} p_3^{k_3} p_4 (a_4^{i-1} + p_4 a_4^{2r-i}) \left(\frac{1-(p_4 a_4^r)^{\frac{k_4}{r}}}{1-(p_4 a_4^r)^2} \right) n,$$

на основании которого числовые значения для различных r приведены в таблице 3.5.

Таблица 3.5. Последовательное выполнение упорядоченных таблиц T1, T2, T3, T4

| Процессы $r=i, j$ | $k=32$ | Время выполнения i -ом процессор/ n | Максимальное время выполнения/ n |
|----------------------|--|--|---------------------------------------|
| | $p_1 = 0.82, p_2 = 0.838, a=[1.15, 1.2, 1.25, 1.3]$ | | |
| | Порядок выполнения ЭЗ | | |
| $r=1, 1$ | 1, 1.15, (1.15) ² , (1.15) ³ , (1.15) ⁴ , (1.15) ⁵ , (1.15) ⁶ , (1.15) ⁷ , 1, 1.2, (1.2) ² , (1.2) ³ , (1.2) ⁴ , (1.2) ⁵ , (1.2) ⁶ , (1.2) ⁷ , 1, 1.25, (1.25) ² , (1.25) ³ , (1.25) ⁴ , (1.25) ⁵ , (1.25) ⁶ , (1.25) ⁷ , 1, 1.3, (1.3) ² , | 7.0836 | 7.0836 |

| | | | |
|-----------|---|--------|--------|
| | $(1.3)^3, (1.3)^4, (1.3)^5, (1.3)^6, (1.3)^7$ | | |
| $r=2,1$ | $1, (1.15)^3, (1.15)^4, (1.15)^7, 1, (1.2)^3, (1.2)^4, (1.2)^7, 1, (1.25)^3, (1.25)^4, (1.25)^7, 1, (1.3)^3, (1.3)^4, (1.3)^7$ | 8.1025 | 8.1025 |
| $r=2,2$ | $1.15, (1.15)^2, (1.15)^5, (1.15)^6, 1.2, (1.2)^2, (1.2)^5, (1.2)^6, 1.25, (1.25)^2, (1.25)^5, (1.25)^6, 1.3, (1.3)^2, (1.3)^5$ | 7.9801 | |
| $r=4,1$ | $1, (1.15)^7, 1, (1.2)^7, 1, (1.25)^7, 1, (1.3)^7$ | 8.3380 | 8.3380 |
| $r=4,2$ | $1.15, (1.15)^6, 1.2, (1.2)^6, 1, (1.25)^6, 1.3, (1.3)^6$ | 7.6168 | |
| $r=4,3$ | $(1.15)^2, (1.15)^5, (1.2)^2, (1.2)^5, (1.25)^2, (1.25)^5, (1.3)^2, (1.3)^5$ | 7.1943 | |
| $r=4,4$ | $(1.15)^3, (1.15)^4, (1.2)^3, (1.2)^4, (1.25)^3, (1.25)^4, (1.3)^3, (1.3)^4$ | 7.0499 | |
| $r=8,1$ | $1, (1.2)^7, 1, (1.3)^7$ | 6.7557 | 6.7557 |
| $r=8,2$ | $1.15, (1.2)^6, 1.25, (1.3)^6$ | 5.9343 | |
| $r=8,3$ | $(1.15)^2, (1.2)^5, (1.25)^2, (1.3)^5$ | 5.3912 | |
| $r=8,4$ | $(1.15)^3, (1.2)^4, (1.25)^3, (1.3)^4$ | 5.0906 | |
| $r=8,5$ | $(1.15)^4, (1.2)^3, (1.25)^4, (1.3)^3$ | 5.0092 | |
| $r=8,6$ | $(1.15)^5, (1.2)^2, (1.25)^5, (1.3)^2$ | 5.1351 | |
| $r=8,7$ | $(1.15)^6, 1.2, (1.25)^6, 1.3$ | 5.4669 | |
| $r=8,8$ | $(1.15)^7, 1, (1.25)^7, 1$ | 6.0126 | |
| $r=16,1$ | $1, (1.3)^7$ | 5.1318 | 5.1318 |
| $r=16,2$ | $1.15, (1.3)^6$ | 4.2598 | |
| $r=16,3$ | $(1.15)^2, (1.3)^5$ | 3.6358 | |
| $r=16,4$ | $(1.15)^3, (1.3)^4$ | 3.2097 | |
| $r=16,5$ | $(1.15)^4, (1.3)^3$ | 2.9439 | |
| $r=16,6$ | $(1.15)^5, (1.3)^2$ | 2.8106 | |
| $r=16,7$ | $(1.15)^6, (1.3)$ | 2.7900 | |
| $r=16,8$ | $(1.15)^7, 1$ | 2.8684 | |
| $r=16,9$ | $1, (1.25)^7$ | 4.0966 | |
| $r=16,10$ | $(1.2), (1.25)^6$ | 3.6053 | |

| | | | |
|-----------|----------------------|--------|--------|
| $r=16,11$ | $(1.2)^2, (1.25)^5,$ | 3.2778 | |
| $r=16,12$ | $(1.2)^3, (1.25)^4,$ | 3.0946 | |
| $r=16,13$ | $(1.2)^4, (1.25)^3,$ | 3.0425 | |
| $r=16,14$ | $(1.2)^5, (1.25)^2,$ | 3.1141 | |
| $r=16,15$ | $(1.2)^6, (1.25)$ | 3.3075 | |
| $r=16,16$ | $(1.2)^7, 1$ | 3.6254 | |
| $r=32,32$ | 1 до 32 | 5.2542 | 5.2542 |

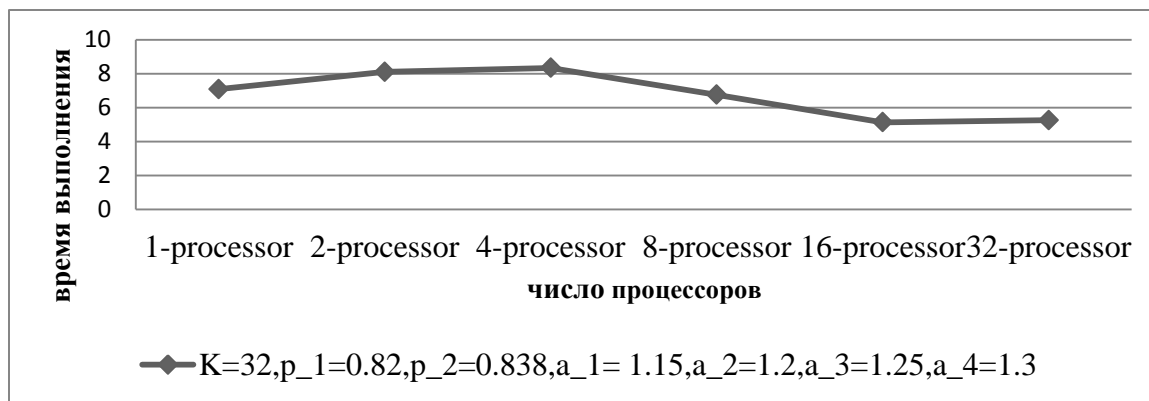


Рис.3.3 Последовательное выполнение упорядоченных таблиц T1, T2, T3, T4 при несовместной обработке соответствующими геометрической прогрессии.

3.4.1.2. Совместное выполнение вложенного запроса

При совместном выполнении вложенного запроса для неупорядоченных таблиц 3.1-3.4 в соответствии с теоремой 3 на основе информации таблиц 3.1-3.4 необходимо провести переупорядочение порядка выполнения элементарных запросов:

$$\begin{aligned}
\frac{\tau_1}{1-p_1} &\leq \frac{\tau_2}{1-p_2} \leq \frac{\tau_3}{1-p_3} \leq \frac{\tau_4}{1-p_4} \leq \frac{\tau_5}{1-p_5} \leq \frac{\tau_6}{1-p_6} \leq \frac{\tau_7}{1-p_7} \leq \frac{\tau_8}{1-p_8} \\
&\leq \frac{\tau_9}{1-p_9} \leq \frac{\tau_{10}}{1-p_{10}} \leq \frac{\tau_{11}}{1-p_{11}} \leq \frac{\tau_{12}}{1-p_{12}} \leq \frac{\tau_{13}}{1-p_{13}} \leq \frac{\tau_{14}}{1-p_{14}} \\
&\leq \frac{\tau_{15}}{1-p_{15}} \leq \frac{\tau_{16}}{1-p_{16}} \leq \frac{\tau_{17}}{1-p_{17}} \leq \frac{\tau_{18}}{1-p_{18}} \leq \frac{\tau_{19}}{1-p_{19}} \leq \frac{\tau_{20}}{1-p_{20}} \\
&\leq \frac{\tau_{21}}{1-p_{21}} \leq \frac{\tau_{22}}{1-p_{22}} \leq \frac{\tau_{23}}{1-p_{23}} \leq \frac{\tau_{24}}{1-p_{24}} \leq \frac{\tau_{25}}{1-p_{25}} \leq \frac{\tau_{26}}{1-p_{26}} \\
&\leq \frac{\tau_{27}}{1-p_{27}} \leq \frac{\tau_{28}}{1-p_{28}} \leq \frac{\tau_{29}}{1-p_{29}} \leq \frac{\tau_{30}}{1-p_{30}} \leq \frac{\tau_{31}}{1-p_{31}} \leq \frac{\tau_{32}}{1-p_{32}}
\end{aligned}$$

в соответствии с таблицей 3.6

Таблица 3.6. Значения отношений $\frac{\tau_i}{1-p_i}$

| Порядок ЭЗ | | | | | | | | |
|------------------------|----------|----------|------------|------------|------------|------------|------------|------------|
| τ_i | 1 | 1.15 | $(1.15)^2$ | $(1.15)^3$ | $(1.15)^4$ | $(1.15)^5$ | $(1.15)^6$ | $(1.15)^7$ |
| $1-p_i$ | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) |
| $\frac{\tau_i}{1-p_i}$ | 5.55 | 6.38 | 7.34 | 8.44 | 9.71 | 11.17 | 12.85 | 14.77 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1 | 1.2 | $(1.2)^2$ | $(1.2)^3$ | $(1.2)^4$ | $(1.2)^5$ | $(1.2)^6$ | $(1.2)^7$ |
| p_i | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) |
| $\frac{\tau_i}{1-p_i}$ | 5.55 | 6.66 | 8.00 | 9.60 | 11.52 | 13.82 | 16.58 | 19.90 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1 | 1.25 | $(1.25)^2$ | $(1.25)^3$ | $(1.25)^4$ | $(1.25)^5$ | $(1.25)^6$ | $(1.25)^7$ |
| p_i | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 |
| $\frac{\tau_i}{1-p_i}$ | 6.17 | 7.71 | 9.64 | 12.05 | 15.07 | 18.83 | 23.54 | 29.43 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1 | 1.3 | $(1.3)^2$ | $(1.3)^3$ | $(1.3)^4$ | $(1.3)^5$ | $(1.3)^6$ | $(1.3)^7$ |
| p_i | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 |
| $\frac{\tau_i}{1-p_i}$ | 6.17 | 8.02 | 10.43 | 13.56 | 17.63 | 22.91 | 29.79 | 38.73 |

В результате получаем выполнение следующих неравенств:

$$\begin{aligned} \frac{1}{1-0.82} &= \frac{1}{1-0.82} \leq \frac{1}{1-0.838} = \frac{1}{1-0.838} \leq \frac{1.15}{1-0.82} \leq \frac{1.2}{1-0.82} \leq \frac{(1.15)^2}{1-0.82} \leq \frac{1.25}{1-0.838} \leq \\ \frac{(1.2)^2}{1-0.82} &\leq \frac{1.3}{1-0.838} \leq \frac{(1.15)^3}{1-0.82} \leq \frac{(1.2)^3}{1-0.82} \leq \frac{(1.25)^2}{1-0.838} \leq \frac{(1.15)^4}{1-0.82} \leq \frac{(1.3)^2}{1-0.838} \leq \frac{(1.15)^5}{1-0.82} \leq \\ \frac{(1.2)^4}{1-0.82} &\leq \frac{(1.25)^3}{1-0.838} \leq \frac{(1.15)^6}{1-0.82} \leq \frac{(1.3)^3}{1-0.838} \leq \frac{(1.2)^5}{1-0.82} \leq \frac{(1.15)^7}{1-0.82} \leq \frac{(1.25)^4}{1-0.838} \leq \frac{(1.2)^6}{1-0.82} \leq \\ \frac{(1.3)^4}{1-0.838} &\leq \frac{(1.25)^5}{1-0.838} \leq \frac{(1.2)^7}{1-0.82} \leq \frac{(1.3)^5}{1-0.838} \leq \frac{(1.25)^6}{1-0.838} \leq \frac{(1.25)^7}{1-0.838} \leq \frac{(1.3)^6}{1-0.838} \leq \frac{(1.3)^7}{1-0.838}. \end{aligned}$$

На основе этих неравенств определяем соответствующий порядок выполнения элементарных запросов для упорядоченных таблиц 3.1-3.4 и время выполнения вложенных запросов при их совместной обработке приведены в таблице 3.7.

Таблица 3.7. Время выполнения вложенного запроса при совместной обработке неупорядоченных таблиц 3.1-3.4

| Процессы $r=i,j$ | $k=32$ | Время выполнения i -ом процессом p/n | Максимальное время выполнения/ n |
|---------------------|--|--|---|
| | $p_1=0.82, p_2 = 0.838, a_1=1.15, a_2 = 1.2, a_3 = 1.25, a_4=1.3$ | | |
| | Порядок обработки ЭЗ | | |
| $r = 1,1$ | 1,1,1,1,1.15,1.2, 1.15 ² ,1.25, 1.2 ² ,1.3, 1.15 ³ ,1.2 ³ , 1.25 ² , 1.15 ⁴ , 1.3 ² , 1.15 ⁵ , 1.2 ⁴ , 1.25 ³ , 1.15 ⁶ , 1.3 ³ , 1.2 ⁵ , 1.15 ⁷ , 1.25 ⁴ , 1.2 ⁶ , 1.3 ⁴ , 1.25 ⁵ , 1.2 ⁷ , 1.3 ⁵ , 1.25 ⁶ , 1.25 ⁷ , 1.3 ⁶ , 1.3 ⁷ , | 9.2644 | 9.2644 |
| $r = 2,1$ | 1,1,1.15, 1.2 ² , 1.2 ³ , 1.25 ² , 1.15 ⁵ , 1.2 ⁴ , 1.3 ³ , 1.2 ⁵ , 1.2 ⁶ , 1.3 ⁴ , 1.3 ⁵ , 1.25 ⁶ , 1.3 ⁷ , | 8.3110 | 8.3110 |
| $r = 2,2$ | 1,1,1.2, 1.15 ² ,1.3, 1.15 ³ , 1.15 ⁴ , 1.3 ² , 1.25 ³ , 1.15 ⁶ , 1.15 ⁷ , 1.25 ⁴ , 1.25 ⁵ , 1.2 ⁷ , 1.25 ⁷ , 1.3 ⁶ , | 8.1659 | |
| $r = 4,1$ | 1,1.25, 1.2 ² , 1.15 ⁵ , 1.2 ⁴ , 1.2 ⁴ , 1.3 ⁴ , 1.7, | 8.7474 | 8.7474 |

| | | | |
|-----------|---|--------|--------|
| $r=4,2$ | 1, $1.15^2, 1.3, 1.3^2, 1.25^3, 1.25^4, 1.25^5, 1.3^6,$ | 7.9516 | |
| $r=4,3$ | 1,1.2, $1.15^3, 1.15^4, 1.15^6, 1.15^7, 1.2^7, 1.25^7,$ | 8.4309 | |
| $r=4,4$ | 1,1.15, $1.2^3, 1.25^2, 1.3^3, 1.2^5, 1.3^5, 1.25^6,$ | 8.0963 | |
| $r=8,1$ | 1, $1.15^5, 1.2^4, 1.3^7,$ | 7.5034 | 7.5034 |
| $r=8,2$ | 1, $1.3^2, 1.25^3, 1.3^6,$ | 6.5074 | |
| $r=8,3$ | 1, $1.15^4, 1.15^6, 1.25^7,$ | 6.7420 | |
| $r=8,4$ | 1, $1.25^2, 1.3^3, 1.25^6,$ | 6.0971 | |
| $r=8,5$ | 1.15, $1.2^3, 1.2^5, 1.3^5,$ | 6.6662 | |
| $r=8,6$ | 1.2, $1.15^3, 1.15^7, 1.2^7,$ | 6.2114 | |
| $r=8,7$ | $1.15^2, 1.3, 1.25^4, 1.25^5,$ | 5.8235 | |
| $r=8,8$ | 1.25, $1.2^2, 1.2^6, 1.3^4,$ | 6.1179 | |
| $r=16,1$ | 1, $(1.3)^7$ | 6.1454 | |
| $r=16,2$ | 1, $(1.3)^6$ | 4.9580 | |
| $r=16,3$ | 1, $(1.25)^7$ | 4.9959 | |
| $r=16,4$ | 1, $(1.25)^6$ | 4.1967 | |
| $r=16,5$ | 1.15, $(1.3)^5$ | 4.1946 | |
| $r=16,6$ | 1.2, $(1.2)^7,$ | 4.1382 | |
| $r=16,7$ | $1.15^2, (1.25)^5,$ | 3.8249 | |
| $r=16,8$ | 1.25, $(1.3)^4$ | 3.6434 | |
| $r=16,9$ | $(1.2)^2, (1.2)^6$ | 3.8885 | |
| $r=16,10$ | 1.3, $(1.25)^4$ | 3.3459 | |
| $r=16,11$ | $(1.15)^3, (1.15)^7$ | 3.7021 | |
| $r=16,12$ | $(1.2)^3, (1.2)^5$ | 3.4804 | |
| $r=16,13$ | $(1.25)^2, (1.3)^3$ | 3.4036 | |
| $r=16,14$ | $(1.15)^4, (1.15)^6$ | 3.6457 | |
| $r=16,15$ | $(1.3)^2, (1.25)^3$ | 3.3267 | |
| $r=16,16$ | $(1.15)^5, (1.2)^4$ | 3.7117 | |
| $r=32,32$ | 1 до 32 | 6.27 | 6.27 |

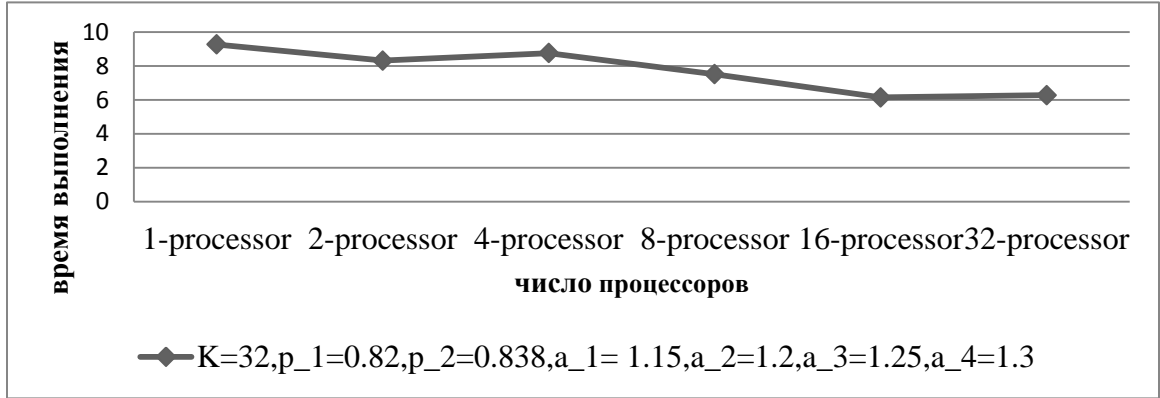


Рис.3.4. Время выполнения вложенного запроса неупорядоченных таблиц при совместной обработке соответствующими геометрической прогрессии.

При совместном выполнении вложенного запроса для упорядоченных таблиц 3.1-3.4 в соответствии с теоремой 3 на основе информации таблиц 3.1-3.4 необходимо провести переупорядочение порядка выполнения элементарных запросов:

$$\frac{p_1\tau_1}{1-p_1} \leq \frac{p_1\tau_2}{1-p_2} \leq \frac{p_1\tau_3}{1-p_3} \leq \frac{p_1\tau_4}{1-p_4} \leq \frac{p_1\tau_5}{1-p_5} \leq \frac{p_1\tau_6}{1-p_6} \leq \frac{p_1\tau_7}{1-p_7} \leq \frac{p_1\tau_8}{1-p_8} \leq$$

$$\frac{p_1\tau_9}{1-p_9} \leq \frac{p_1\tau_{10}}{1-p_{10}} \leq \frac{p_1\tau_{11}}{1-p_{11}} \leq \frac{p_1\tau_{12}}{1-p_{12}} \leq \frac{p_1\tau_{13}}{1-p_{13}} \leq \frac{p_1\tau_{14}}{1-p_{14}} \leq \frac{p_1\tau_{15}}{1-p_{15}} \leq$$

$$\frac{p_1\tau_{16}}{1-p_{16}} \leq \frac{p_2\tau_{17}}{1-p_{17}} \leq \frac{p_2\tau_{18}}{1-p_{18}} \leq \frac{p_2\tau_{19}}{1-p_{19}} \leq \frac{p_2\tau_{20}}{1-p_{20}} \leq \frac{p_2\tau_{21}}{1-p_{21}} \leq \frac{p_2\tau_{22}}{1-p_{22}} \leq$$

$$\frac{p_2\tau_{23}}{1-p_{23}} \leq \frac{p_2\tau_{24}}{1-p_{24}} \leq \frac{p_2\tau_{25}}{1-p_{25}} \leq \frac{p_2\tau_{26}}{1-p_{26}} \leq \frac{p_2\tau_{27}}{1-p_{27}} \leq \frac{p_2\tau_{28}}{1-p_{28}} \leq \frac{p_2\tau_{29}}{1-p_{29}} \leq$$

$$\frac{p_2\tau_{30}}{1-p_{30}} \leq \frac{p_2\tau_{31}}{1-p_{31}} \leq \frac{p_2\tau_{32}}{1-p_{32}}.$$

Таблица 3.8. Значения отношений $\frac{p_i \tau_i}{1-p_i}$

| Порядок ЭЗ | | | | | | | | |
|------------------------|-------------|----------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| τ_i | $1*(0.82)$ | $1.15*(0.82)$ | $(1.15)^2*(0.82)$ | $(1.15)^3*(0.82)$ | $(1.15)^4*(0.82)$ | $(1.15)^5*(0.82)$ | $(1.15)^6*(0.82)$ | $(1.15)^7*(0.82)$ |
| $1 - p_i$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ |
| $\frac{\tau_i}{1-p_i}$ | 4.55 | 5.23 | 6.02 | 6.92 | 7.96 | 9.16 | 10.53 | 12.11 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | $1*(0.82)$ | $1.2*(0.82)$ | $(1.2)^2*(0.82)$ | $(1.2)^3*(0.82)$ | $(1.2)^4*(0.82)$ | $(1.2)^5*(0.82)$ | $(1.2)^6*(0.82)$ | $(1.2)^7*(0.82)$ |
| p_i | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ | $1-(0.82)$ |
| $\frac{\tau_i}{1-p_i}$ | 4.55 | 5.46 | 6.56 | 7.87 | 9.44 | 11.33 | 13.60 | 16.32 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | $1*(0.838)$ | $1.25*(0.838)$ | $(1.25)^2*(0.838)$ | $(1.25)^3*(0.838)$ | $(1.25)^4*(0.838)$ | $(1.25)^5*(0.838)$ | $(1.25)^6*(0.838)$ | $(1.25)^7*(0.838)$ |
| p_i | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ |
| $\frac{\tau_i}{1-p_i}$ | 5.17 | 6.46 | 8.08 | 10.10 | 12.62 | 15.78 | 19.73 | 24.66 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | $1*(0.838)$ | $1.3*(0.838)$ | $(1.3)^2*(0.838)$ | $(1.3)^3*(0.838)$ | $(1.3)^4*(0.838)$ | $(1.3)^5*(0.838)$ | $(1.3)^6*(0.838)$ | $(1.3)^7*(0.838)$ |
| p_i | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ | $1-0.838$ |
| $\frac{\tau_i}{1-p_i}$ | 5.17 | 6.72 | 8.74 | 11.36 | 14.77 | 19.20 | 24.96 | 32.45 |

В результате получаем следующие неравенства:

$$\begin{aligned}
 \frac{1 * 0.82}{1 - 0.82} &= \frac{1 * 0.82}{1 - 0.82} \leq \frac{1 * 0.838}{1 - 0.838} = \frac{1 * 0.838}{1 - 0.838} \leq \frac{1.15 * 0.82}{1 - 0.82} \leq \frac{1.2 * 0.82}{1 - 0.82} \leq \frac{(1.15)^2 * 0.82}{1 - 0.82} \\
 &\leq \frac{1.25 * 0.838}{1 - 0.838} \leq \frac{(1.2)^2 * 0.82}{1 - 0.82} \leq \frac{1.3 * 0.838}{1 - 0.838} \leq \frac{(1.15)^3 * 0.82}{1 - 0.82} \leq \frac{(1.2)^3 * 0.82}{1 - 0.82} \\
 &\leq \frac{(1.15)^4 * 0.82}{1 - 0.82} \leq \frac{(1.25)^2 * 0.838}{1 - 0.838} \leq \frac{(1.3)^2 * 0.838}{1 - 0.838} \leq \frac{(1.15)^5 * 0.82}{1 - 0.82} \\
 &\leq \frac{(1.2)^4 * 0.82}{1 - 0.82} \leq \frac{(1.25)^3 * 0.838}{1 - 0.838} \leq \frac{(1.15)^6 * 0.82}{1 - 0.82} \leq \frac{(1.2)^5 * 0.82}{1 - 0.82} \\
 &\leq \frac{(1.3)^3 * 0.838}{1 - 0.838} \leq \frac{(1.15)^7 * 0.82}{1 - 0.82} \leq \frac{(1.25)^4 * 0.838}{1 - 0.838} \leq \frac{(1.2)^6 * 0.82}{1 - 0.82} \\
 &\leq \frac{(1.3)^4 * 0.838}{1 - 0.838} \leq \frac{(1.25)^5 * 0.838}{1 - 0.838} \leq \frac{(1.2)^7 * 0.82}{1 - 0.82} \leq \frac{(1.3)^5 * 0.838}{1 - 0.838} \\
 &\leq \frac{(1.25)^6 * 0.838}{1 - 0.838} \leq \frac{(1.25)^7 * 0.838}{1 - 0.838} \leq \frac{(1.3)^6 * 0.838}{1 - 0.838} \leq \frac{(1.3)^7 * 0.838}{1 - 0.838}
 \end{aligned}$$

На основе этих неравенств определяем соответствующий порядок выполнения элементарных запросов для упорядоченных таблиц 3.1-3.4 и время выполнения вложенных запросов при их совместной обработке приведены в таблице 3.9.

Таблица 3.9. Время выполнения вложенного запроса при совместной обработке упорядоченных таблиц 3.1-3.4

| Процессоры $r=i,j$ | $k=32$ | Время выполнения i -ом процессор/п | Максимальное время выполнения/п |
|-----------------------|--|--------------------------------------|---------------------------------|
| | $p_1=0.82, p_2 = 0.838, a_1=1.15, a_2 = 1.2, a_3 = 1.25, a_4=1.3$ | | |
| | Порядок выполнения ЭЗ | | |
| $r = 1,1$ | 1,1,1,1,1.15,1.2, (1.15) ² ,1.25, (1.2) ² ,1.3, (1.15) ³ , (1.2) ³ (1.15) ⁴ , (1.25) ² , (1.3) ² , (1.15) ⁵ , (1.2) ⁴ , (1.25) ³ , (1.15) ⁶ (1.2) ⁵ ,(1.3) ³ ,(1.15) ⁷ ,(1.25) ⁴ ,(1.2) ⁶ ,(1.3) ⁴ (1.25) ⁵ ,(1.2) ⁷ ,(1.3) ⁵ , (1.25) ⁶ , (1.25) ⁷ , (1.3) ⁶ , (1.3) ⁷ | 5.7856 | 5.7856 |
| $r = 2,1$ | 1,1,(1.15),(1.25), (1.2) ² , (1.2) ³ , (1.15) ⁴ , (1.15) ⁵ , | 7.0642 | 7.0642 |

| | | | |
|--------------|---|--------|--------|
| | $(1.2)^4, (1.2)^5, (1.3)^3, (1.2)^6, (1.3)^4, (1.3)^5, (1.25)^6, (1.3)^7$ | | |
| $r = 2, 2$ | $1, 1, 1.2, (1.15)^2, 1.3, (1.15)^3, (1.25)^2, (1.3)^2,$ $(1.25)^3, (1.15)^6, (1.15)^7, (1.25)^4, (1.25)^5, (1.2)^7, (1.25)^7, (1.15)^7$ | 6.8459 | |
| $r = 4, 1$ | $1, 1.25, (1.2)^2, 1, (1.15)^5, (1.2)^4, (1.2)^6, (1.3)^4, (1.3)^7$ | 6.5021 | 6.9707 |
| $r = 4, 2$ | $1, (1.15)^2, 1.3, (1.3)^2, (1.25)^3, (1.25)^4, (1.25)^5, (1.3)^6$ | 6.6849 | |
| $r = 4, 3$ | $1, 1.2, (1.15)^3, (1.25)^2, (1.15)^6, (1.15)^7, (1.2)^7, (1.25)^7$ | 6.9707 | |
| $r = 4, 4$ | $1, 1.15, (1.2)^3, (1.15)^4, (1.2)^5, (1.3)^3, (1.3)^5, (1.25)^6$ | 6.8888 | |
| $r = 8, 1$ | $1, (1.15)^5, (1.2)^4, (1.3)^7,$ | 6.2150 | |
| $r = 8, 2$ | $1.15, (1.3)^2, (1.25)^3, (1.3)^6,$ | 5.4352 | |
| $r = 8, 3$ | $1, (1.25)^2, (1.15)^6, (1.25)^7,$ | 6.1360 | |
| $r = 8, 4$ | $1, (1.15)^4, (1.2)^5, (1.25)^6,$ | 5.2432 | |
| $r = 8, 5$ | $1.15, (1.2)^3, (1.3)^3, (1.3)^5,$ | 5.0961 | |
| $r = 8, 6$ | $1.2, (1.15)^3, (1.15)^7, (1.2)^7,$ | 5.8171 | |
| $r = 8, 7$ | $(1.15)^2, 1.3, (1.25)^4, (1.25)^5$ | 4.8563 | |
| $r = 8, 8$ | $1.25, (1.2)^2, (1.2)^6, (1.3)^4$ | 5.0681 | |
| $r = 16, 1$ | $1, (1.3)^7,$ | 5.1318 | 5.1318 |
| $r = 16, 2$ | $1, (1.3)^6$ | 4.1368 | |
| $r = 16, 3$ | $1, (1.25)^7,$ | 4.1866 | |
| $r = 16, 4$ | $1, (1.25)^6,$ | 3.5168 | |
| $r = 16, 5$ | $1.15, (1.3)^5,$ | 3.0811 | |
| $r = 16, 6$ | $1.2, (1.2)^7,$ | 3.3933 | |
| $r = 16, 7$ | $(1.15)^2, (1.25)^5$ | 3.1815 | |
| $r = 16, 8$ | $1.25, (1.3)^4$ | 3.0532 | |
| $r = 16, 9$ | $(1.2)^2, (1.2)^6$ | 3.1886 | |
| $r = 16, 10$ | $(1.3), (1.25)^4,$ | 2.8039 | |
| $r = 16, 11$ | $(1.15)^3, (1.15)^7,$ | 3.0357 | |
| $r = 16, 12$ | $(1.2)^3, (1.3)^3,$ | 2.9267 | |
| $r = 16, 13$ | $(1.15)^4, (1.2)^5,$ | 3.1073 | |
| $r = 16, 14$ | $(1.25)^2, (1.15)^6,$ | 2.8988 | |

| | | | |
|-------------|---------------------|--------|--------|
| $r = 16,15$ | $(1.3)^2, (1.15)^3$ | 2.7878 | |
| $r = 16,16$ | $(1.15)^5, (1.2)^4$ | 3.0436 | |
| $r = 32,32$ | 1 до 32 | 5.2542 | 5.2542 |

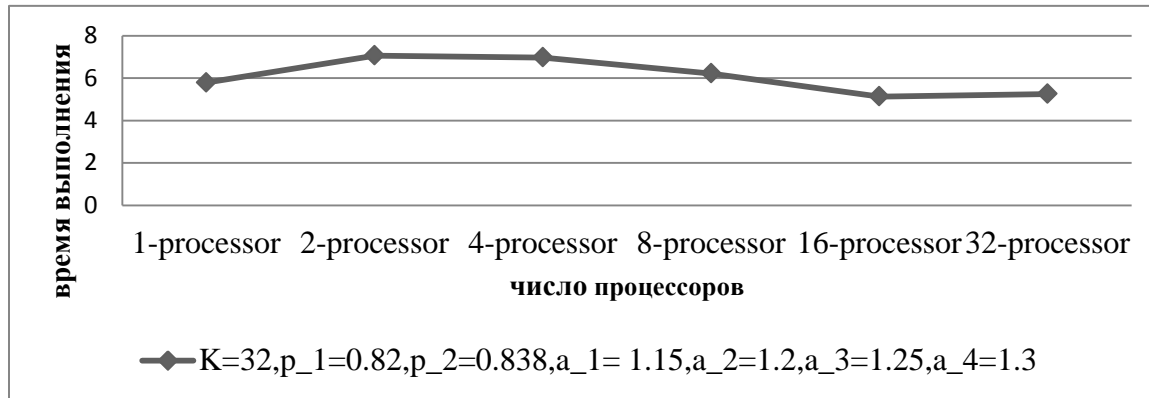


Рис.3.5. Время выполнения вложенного запроса упорядоченных таблиц при совместной обработке соответствующими геометрическая прогрессия.

3.4.2. Вложенные запросы для таблиц с параметрами, соответствующими арифметической прогрессии

3.4.2.1. Несовместное выполнение вложенных запросов

Пусть заданы четыре таблицы (T5, T6, T7, T8), образующие вложенные запросы, с параметрами, соответствующими арифметической прогрессии,

где $\tau_{i,j} = 1 + (i - 1)\Delta$ - время обработки i -го ЭЗ, $p_{i,j} = p_j, j \in \overline{1, k_j}$, для таблицы Tj.

Время выполнения вложенных запросов для неупорядоченных таблиц T5, T6, T7, T8 для r процессорной базы данных определяется выражением:

$$\begin{aligned}
T_{1,2,3,4} = & \left(1 + (i - 1)\Delta_1 + (1 + (2r - i)\Delta_1)p_1 \left(\frac{1 - p_1 \frac{k_1}{r}}{1 - p_1^2} \right) + \right. \\
& 2r\Delta_1 p_1^2 (1 + p_1) \left(\frac{1 - \frac{k_1}{2r} p_1 \frac{k_1}{r} - 2 + \left(\frac{k_1}{2r} - 1 \right) p_1 \frac{k_1}{r}}{(1 - p_1 r)^2} \right) \left. + \right. \\
& p_1^{k_1} \left(1 + (i - 1)\Delta_2 + (1 + (2r - i)\Delta_2)p_2 \left(\frac{1 - p_2 \frac{k_2}{r}}{1 - p_2^2} \right) + \right. \\
& 2r\Delta_2 p_2^2 (1 + p_2) \left(\frac{1 - \frac{k_2}{2r} p_2 \frac{k_2}{r} - 2 + \left(\frac{k_2}{2r} - 1 \right) p_2 \frac{k_2}{r}}{(1 - p_2 r)^2} \right) \left. + p_1^{k_1} p_2^{k_2} \left(1 + (i - 1)\Delta_3 + (1 + (2r - \right. \right. \\
& i)\Delta_3) p_3 \left(\frac{1 - p_3 \frac{k_3}{r}}{1 - p_3^2} \right) + 2r\Delta_3 p_3^2 (1 + p_3) \left(\frac{1 - \frac{k_3}{2r} p_3 \frac{k_3}{r} - 2 + \left(\frac{k_3}{2r} - 1 \right) p_3 \frac{k_3}{r}}{(1 - p_3 r)^2} \right) \left. + p_1^{k_1} p_2^{k_2} p_3^{k_3} \left(1 + \right. \right. \\
& (i - 1)\Delta_4 + (1 + (2r - i)\Delta_4) p_4 \left(\frac{1 - p_4 \frac{k_4}{r}}{1 - p_4^2} \right) + 2r\Delta_4 p_4^2 (1 + p_4) \left(\frac{1 - \frac{k_4}{2r} p_4 \frac{k_4}{r} - 2 + \left(\frac{k_4}{2r} - 1 \right) p_4 \frac{k_4}{r}}{(1 - p_4 r)^2} \right) \left. \right). n, i = \\
& \overline{1, r},
\end{aligned}$$

Пусть числовые значения таблиц Т5, Т6, Т7, Т8 заданы в таблице 3.10:

Таблица 3.10. Параметры таблиц Т5, Т6, Т7, Т8

| p_j | $k_j=8, \Delta_1=0.4, \Delta_2=0.5, \Delta_3=0.6, \Delta_4=0.7$ | | | | | | | |
|-------|---|----------|----------|----------|----------|----------|----------|----------|
| | τ_1 | τ_2 | τ_3 | τ_4 | τ_5 | τ_6 | τ_7 | τ_8 |
| 0.82 | 1 | 1.4 | 1.8 | 2.2 | 2.6 | 3 | 3.4 | 3.8 |
| 0.82 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 |
| 0.838 | 1 | 1.6 | 2.2 | 2.8 | 3.4 | 4 | 4.6 | 5.2 |
| 0.838 | 1 | 1.7 | 2.4 | 3.1 | 3.8 | 4.5 | 5.2 | 5.9 |

Тогда время выполнения вложенных запросов для неупорядоченных таблиц Т5, Т6, Т7, Т8 для r процессорной базы данных приведены в таблице 3.11.

Таблица 3.11. Последовательное выполнение неупорядоченных таблиц T5, T6, T7, T8

| Процессоры $r=i,j$ | $k=32$ | Время выполнения я i -ом процессор/ n | Максим время выполн ения/ n |
|-----------------------|--|---|--|
| | $p_1 = 0.82, \Delta; p_2 = 0.838; \Delta_1 = 0.4, \Delta_2 = 0.5, \Delta_3 = 0.6, \Delta_4 = 0.7$ | | |
| | Порядок Отношение Эз | | |
| $r=1,1$ | 1,1.4,1.8,2.2,2.6,3,3.4,3.8,1,1.5,2,2.5,3,3.5,4,4.5 1,1.6,2,2,2.8,3.4,4,4.6,5.2,1,1.7,2.4,3.1,3.8,4.5,5.2,5.9 | 9.6262 | 9.6262 |
| $r=2,1$ | 1,2.2,2.6,3.8,1,2.5,3,4.5,1,2.8,3.4,5.2,1,3.1,3.8,5.9 | 10.7548 | 11.0238 |
| $r=2,2$ | 1.4,1.8,3,3.4,1.5,2,3.5,4,1.6,2.2,4,4.6,1.7,2.4,4.5,5.2 | 11.0238 | |
| $r=4,1$ | 1,3.8,1,4.5,1,5.2,1,5.9 | 9.7443 | 10.4727 |
| $r=4,2$ | 1.4,3.4,1.5,4,1.6,4.6,1.7,5.2 | 9.9871 | |
| $r=4,3$ | 1.8,3,2,3.5,2.2,4,2.4,4.5 | 10.2299 | |
| $r=4,4$ | 2.2,2.6,2.5,3,2.8,3.4,3.1,3.8 | 10.4727 | |
| $r=8,1$ | 1,4.5,1,5.9 | 8.6155 | 7.8943 |
| $r=8,2$ | 1.4,4,1.6,5.2 | 7.5202 | |
| $r=8,3$ | 1.8,3.5,2.2,4.5 | 7.5767 | |
| $r=8,4$ | 2.2,3,2.8,3.8 | 7.8284 | |
| $r=8,5$ | 2.6,2.5,3.4,3.1 | 7.5572 | |
| $r=8,6$ | 3,2,4,2.4 | 7.6354 | |
| $r=8,7$ | 3.4,1.5,4.6,1.7 | 7.8778 | |
| $r=8,8$ | 3.8,1,5.2,1 | 7.8943 | |
| $r=16,1$ | 1,5.9 | 4.9672 | 5.2022 |
| $r=16,2$ | 1.4,5.2 | 4.8965 | |
| $r=16,3$ | 1.8,4.5 | 4.9601 | |
| $r=16,4$ | 2.2,3.8 | 4.8685 | |
| $r=16,5$ | 2.6,3.1 | 4.6844 | |
| $r=16,6$ | 3,2.4 | 4.6138 | |

| | | | |
|-----------|---------|--------|-----|
| $r=16,7$ | 3.4,1.7 | 4.5431 | |
| $r=16,8$ | 3.8,1 | 4.5022 | |
| $r=16,9$ | 1,5.2 | 4.4965 | |
| $r=16,10$ | 1.5,4.6 | 4.7303 | |
| $r=16,11$ | 2,4 | 4.6896 | |
| $r=16,12$ | 2.5,3.4 | 4.7862 | |
| $r=16,13$ | 3,2.8 | 4.9663 | |
| $r=16,14$ | 3.5,2.2 | 4.9793 | |
| $r=16,15$ | 4,1.6 | 5.1236 | |
| $r=16,16$ | 4.5,1 | 5.2022 | |
| $r=32,32$ | 1 до 32 | 5.9 | 5.9 |

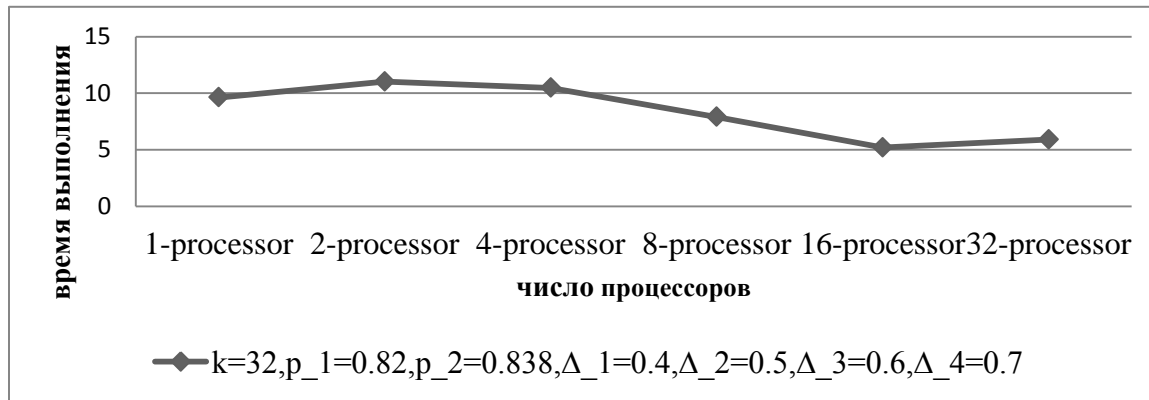


Рис.3.6. Последовательное выполнение неупорядоченных таблиц T5, T6, T7, T8 при несовместной обработке соответствующими арифметической прогрессии.

Время выполнения вложенных запросов для упорядоченных таблиц T5, T6, T7, T8 для r процессорной базы данных определяется выражением:

$$T_{1,2,3,4} = p_1 \left(1 + (i-1)\Delta_1 + (1 + (2r-i)\Delta_1)p_1 \left(\frac{1-p_1^{\frac{k_1}{r}}}{1-p_1^2} \right) + \right. \\ \left. 2r\Delta_1 p_1^2 (1+p_1) \left(\frac{1-\frac{k_1}{2r}p_1^{\frac{k_1}{r}-2} + (\frac{k_1}{2r}-1)p_1^{\frac{k_1}{r}}}{(1-p_1^r)^2} \right) \right) + p_1^{k_1} p_2 \left(1 + (i-1)\Delta_2 + (1 + (2r -$$

$$\begin{aligned}
& i)\Delta_2)p_2 \left(\frac{1-p_2 \frac{k_2}{r}}{1-p_2^2} \right) + 2r\Delta_2 p_2^2 (1+p_2) \left(\frac{1-\frac{k_2}{2r} p_2 \frac{k_2}{r^2-2} + \left(\frac{k_2}{2r}-1\right) p_2 \frac{k_2}{r}}{(1-p_2^r)^2} \right) + p_1^{k_1} p_2^{k_2} p_3 \left(1 + \right. \\
& (i-1)\Delta_3 + (1+(2r-i)\Delta_3) p_3 \left(\frac{1-p_3 \frac{k_3}{r}}{1-p_3^2} \right) + 2r\Delta_3 p_3^2 (1+p_3) \left(\frac{1-\frac{k_3}{2r} p_3 \frac{k_3}{r^2-2} + \left(\frac{k_3}{2r}-1\right) p_3 \frac{k_3}{r}}{(1-p_3^r)^2} \right) \Big) + \\
& p_1^{k_1} p_2^{k_2} p_3^{k_3} p_4 \left(1 + (i-1)\Delta_4 + (1+(2r-i)\Delta_4) p_4 \left(\frac{1-p_4 \frac{k_4}{r}}{1-p_4^2} \right) + \right. \\
& \left. 2r\Delta_4 p_4^2 (1+p_4) \left(\frac{1-\frac{k_4}{2r} p_4 \frac{k_4}{r^2-2} + \left(\frac{k_4}{2r}-1\right) p_4 \frac{k_4}{r}}{(1-p_4^r)^2} \right) \right). n, i = \overline{1, r},
\end{aligned}$$

и приведены в таблице 3.12.

Таблица 3.12. Последовательное выполнение упорядоченных таблиц T5, T6, T7, T8

| Процессоры $r=i,j$ | $k=32$ | Время выполнени я i -ом процессор/ n | Максим время выполн ения/ n |
|-----------------------|--|--|--|
| | $p_1 = 0.82, \Delta; p_2 = 0.838; \Delta_1 = 0.4, \Delta_2 = 0.5, \Delta_3 = 0.6, \Delta_4 = 0.7$ | | |
| | Порядок Отношение Эз | | |
| $r=1,1$ | 1,1.4,1.8,2.2,2.6,3,3.4,3.8,1,1.5,2,2.5,3,3.5,4,4.5 1,1.6,2,2.2,2.8,3,4,4,4.6,5,2,1,1.7,2,4,3,1,3.8,4,5,5,2,5,9 | 9.4470 | 9.4470 |
| $r=2,1$ | 1,2,2,2.6,3.8,1,2.5,3,4,5,1,2.8,3,4,5,2,1,3,1,3.8,5,9 | 10.5785 | 10.7776 |
| $r=2,2$ | 1.4,1.8,3,3.4,1.5,2,3,5,4,1.6,2,2,4,4.6,1.7,2,4,4,5,5,2 | 10.7776 | |
| $r=4,1$ | 1,3.8,1,4.5,1,5,2,1,5,9 | 9.5725 | 10.0994 |
| $r=4,2$ | 1.4,3,4,1.5,4,1.6,4,6,1.7,5,2 | 9.7481 | |
| $r=4,3$ | 1.8,3,2,3,5,2,2,4,2,4,4,5 | 9.9238 | |
| $r=4,4$ | 2,2,2.6,2,5,3,2,8,3,4,3,1,3,8 | 10.0994 | |
| $r=8,1$ | 1,4.5,1,5,9 | 7.1952 | 7.1952 |
| $r=8,2$ | 1,4,4,1,6,5,2 | 7.1945 | |
| $r=8,3$ | 1.8,3,5,2,2,4,5 | 7.1939 | |

| | | | |
|-----------|-----------------|--------|--------|
| $r=8,4$ | 2.2,3,2.8,3.8 | 7.1932 | |
| $r=8,5$ | 2.6,2.5,3.4,3.1 | 7.1926 | |
| $r=8,6$ | 3,2,4,2.4 | 7.1919 | |
| $r=8,7$ | 3.4,1.5,4.6,1.7 | 7.1913 | |
| $r=8,8$ | 3.8,1,5.2,1 | 7.1906 | |
| $r=16,1$ | 1,5.9 | 4.8742 | 4.8742 |
| $r=16,2$ | 1.4,5.2 | 4.7212 | |
| $r=16,3$ | 1.8,4.5 | 4.5682 | |
| $r=16,4$ | 2.2,3.8 | 4.4152 | |
| $r=16,5$ | 2.6,3.1 | 4.2622 | |
| $r=16,6$ | 3,2.4 | 4.1092 | |
| $r=16,7$ | 3.4,1.7 | 3.9562 | |
| $r=16,8$ | 3.8,1 | 3.8032 | |
| $r=16,9$ | 1,5.2 | 4.3932 | |
| $r=16,10$ | 1.5,4.6 | 4.3909 | |
| $r=16,11$ | 2,4 | 4.3886 | |
| $r=16,12$ | 2.5,3.4 | 4.3863 | |
| $r=16,13$ | 3,2.8 | 4.3840 | |
| $r=16,14$ | 3.5,2.2 | 4.3818 | |
| $r=16,15$ | 4,1.6 | 4.3795 | |
| $r=16,16$ | 4.5,1 | 4.3772 | |
| $r=32,32$ | 1 до 32 | 4.9442 | 4.9442 |

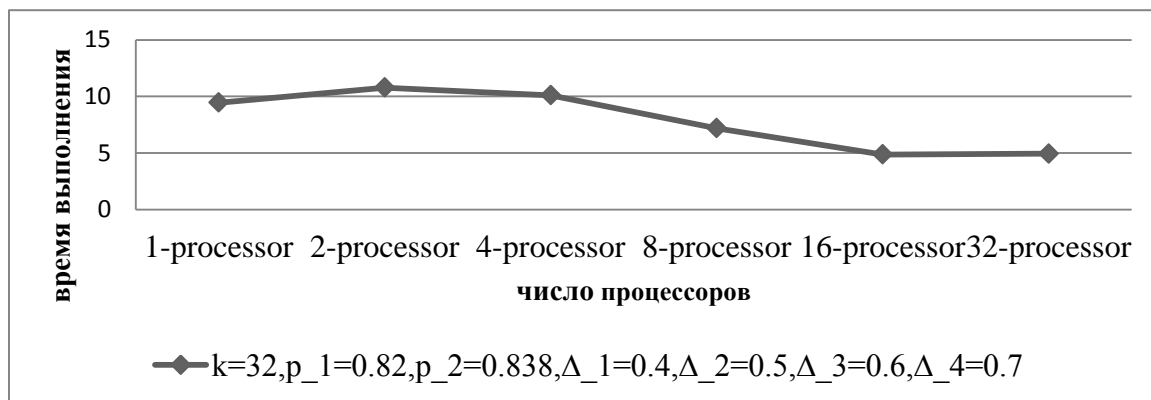


Рис.3.7. Последовательное выполнение упорядоченных таблиц T5, T6, T7, T8 при несовместной обработке соответствующими арифметической прогрессии.

3.4.2.2. Совместное выполнение вложенного запроса

При совместном выполнении вложенного запроса для неупорядоченных таблиц 3.5-3.8 необходимо провести переупорядочение порядка выполнения элементарных запросов.

В соответствии с таблицей 3.13

Таблица 3.13. Значения отношений $\frac{\tau_i}{1-p_i}$

| Порядок ЭЗ | | | | | | | | |
|--------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| τ_i | 1 | 1.4 | 1.8 | 2.2 | 2.6 | 3 | 3.4 | 3.8 |
| $1 - p_i$ | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) |
| $\frac{\tau_i}{1 - p_i}$ | 5.55 | 7.77 | 10 | 12.22 | 14.44 | 16.66 | 18.88 | 21.11 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 |
| p_i | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) |
| $\frac{\tau_i}{1 - p_i}$ | 5.55 | 8.33 | 11.11 | 13.88 | 16.66 | 19.44 | 22.22 | 25 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1 | 1.6 | 2.2 | 2.8 | 3.4 | 4 | 4.6 | 5.2 |

| | | | | | | | | |
|------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| p_i | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 |
| $\frac{\tau_i}{1-p_i}$ | 6.17 | 9.87 | 13.58 | 17.28 | 20.98 | 24.69 | 28.39 | 32.09 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1 | 1.7 | 2.4 | 3.1 | 3.8 | 4.5 | 5.2 | 5.9 |
| p_i | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 |
| $\frac{\tau_i}{1-p_i}$ | 6.17 | 10.49 | 14.81 | 19.13 | 23.45 | 27.77 | 32.09 | 36.41 |

получаем выполнение следующих неравенств:

$$\begin{aligned}
\frac{1}{1-0.82} &= \frac{1}{1-0.82} \leq \frac{1}{1-0.838} = \frac{1}{1-0.838} \leq \frac{1.4}{1-0.82} \leq \frac{1.5}{1-0.82} \leq \frac{1.6}{1-0.838} \\
&\leq \frac{1.8}{1-0.82} \leq \frac{1.7}{1-0.838} \leq \frac{2}{1-0.82} \leq \frac{2.2}{1-0.82} \leq \frac{2.2}{1-0.838} \leq \frac{2.5}{1-0.82} \\
&\leq \frac{2.6}{1-0.82} \leq \frac{2.4}{1-0.838} \leq \frac{3}{1-0.82} \leq \frac{3}{1-0.82} \leq \frac{2.8}{1-0.838} \leq \frac{3.4}{1-0.82} \\
&\leq \frac{3.1}{1-0.838} \leq \frac{3.5}{1-0.82} \leq \frac{3.4}{1-0.82} \leq \frac{3.8}{1-0.82} \leq \frac{4}{1-0.82} \leq \frac{3.8}{1-0.82} \\
&\leq \frac{4}{1-0.838} \leq \frac{4.5}{1-0.82} \leq \frac{4.5}{1-0.838} \leq \frac{4.6}{1-0.838} \leq \frac{5.2}{1-0.838} \leq \frac{5.2}{1-0.838} \\
&\leq \frac{5.9}{1-0.838}
\end{aligned}$$

На основе этих неравенств определяем соответствующий порядок выполнения элементарных запросов и время выполнения вложенных запросов при совместной обработке при различном числе процессоров, приведенные в таблице 3.14.

Таблица 3.14. Время выполнения вложенного запроса при совместной обработке неупорядоченных таблиц 3.5-3.7

| | | | |
|-----------------------|---|---------------------------------|---------------------------|
| Процессоры $r=i,j$ | $k=32$ | Время выполнени я i -ом | Максим время выполн |
| | $p_1 = 0.82, \Delta; p_2 = 0.838; \Delta_1 = 0.4, \Delta_2 = 0.5, \Delta_3 = 0.6, \Delta_4 = 0.7$ | | |

| | Порядок Отношение Эз | процессор/ <i>n</i> | ения/ <i>n</i> |
|-------------|--|------------------------|-------------------|
| $r = 1,1$ | 1,1,1,1,1.4,1.5,1.6,1.8 1.7,2.2,2.2,2.2,2.5,2.6,2.4,3 3,2.8,3.4,3.1,3.5,3.4,3.8,4 3.8,4,4.5,4.5,4.6,5.2,5.2,5.9 | 7.9772 | 7.9772 |
| $r = 2,1$ | 1,1,1.4,1.8,1.7,2.2,2.5,3,3,3.1,3.5,4,3.8,4.5,4.6,5.9 | 10.4126 | 10.4126 |
| $r = 2,2$ | 1,1,1.5,1.6,2,2.2,2.6,2.4,2.8,3.4,3.4,3.8,4,4.5,5.2,5.2 | 10.3793 | |
| $r = 4,1$ | 1,1.8,1.7,3,3,4,3.8,5.9 | 10.9213 | 10.9213 |
| $r = 4,2$ | 1,1.6,2,2.4,2.8,3.8,4,5.2 | 10.3393 | |
| $r = 4,3$ | 1,1.5,2.2,2.6,3.4,3.4,4.5,5.2 | 10.7805 | |
| $r = 4,4$ | 1,1.4,2.2,2.5,3.1,3.5,4.5,4.6 | 10.3860 | |
| $r = 8,1$ | 1,3,3,5.9 | 8.7303 | 8.7303 |
| $r = 8,2$ | 1,2.4,2.8,5.2 | 7.7592 | |
| $r = 8,3$ | 1,2.6,3.4,5.2 | 8.4452 | |
| $r = 8,4$ | 1,2.5,3.1,4.6 | 7.9790 | |
| $r = 8,5$ | 1.4,2.2,3.5,4.5 | 8.0386 | |
| $r = 8,6$ | 1.5,2.2,3.4,4.5 | 8.0713 | |
| $r = 8,7$ | 1.6,2,3.8,4 | 8.1546 | |
| $r = 8,8$ | 1.8,1.7,4,3.8 | 8.1144 | |
| $r = 16,1$ | 1,5.9 | 5.8380 | 5.8380 |
| $r = 16,2$ | 1,5.2 | 5.2640 | |
| $r = 16,3$ | 1,5.2 | 5.3576 | |
| $r = 16,4$ | 1,4.6 | 4.8548 | |
| $r = 16,5$ | 1.4,4.5 | 5.0900 | |
| $r = 16,6$ | 1.5,4.5 | 5.1900 | |
| $r = 16,7$ | 1.6,4 | 4.8800 | |
| $r = 16,8$ | 1.8,3.8 | 4.9844 | |
| $r = 16,9$ | 1.7,4 | 4.9800 | |
| $r = 16,10$ | 2,3 | 4.5140 | |

| | | | | | | | | |
|------------------------|--------------|----------------|---------------|----------------|---------------|----------------|---------------|----------------|
| $\frac{\tau_i}{1-p_i}$ | 4.55 | 6.37 | 8.20 | 10.02 | 11.84 | 13.66 | 15.48 | 17.31 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1*(0.82) | 1.5*(0.8 2) | 2*(0.82) | 2.5*(0.8 2) | 3*(0.82) | 3.5*(0.8 2) | 4*(0.82) | 4.5*(0.8 2) |
| p_i | 1- (0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) | 1-(0.82) |
| $\frac{\tau_i}{1-p_i}$ | 4.55 | 6.83 | 9.11 | 11.38 | 13.66 | 15.94 | 18.22 | 20.50 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1*0.83 8 | 1.6*0.83 8 | 2.2*0.83 8 | 2.8*0.83 8 | 3.4*0.83 8 | 4*0.838 | 4.6*0.83 8 | 5.2*0.83 8 |
| p_i | 1- 0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 |
| $\frac{\tau_i}{1-p_i}$ | 4.65 | 7.44 | 10.24 | 13.03 | 15.82 | 18.62 | 21.41 | 24.20 |
| Порядок ЭЗ | | | | | | | | |
| τ_i | 1*0.83 8 | 1.7*0.83 8 | 2.4*0.83 8 | 3.1*0.83 8 | 3.8*0.83 8 | 4.5*0.83 8 | 5.2*0.83 8 | 5.9*0.83 8 |
| p_i | 1- 0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 | 1-0.838 |
| $\frac{\tau_i}{1-p_i}$ | 4.65 | 7.91 | 11.17 | 14.43 | 17.69 | 20.95 | 24.20 | 27.46 |

получаем выполнение следующих неравенств:

$$\begin{aligned}
\frac{1 * 0.82}{1 - 0.82} &= \frac{1 * 0.82}{1 - 0.82} \leq \frac{1 * 0.838}{1 - 0.838} = \frac{1 * 0.838}{1 - 0.838} \leq \frac{1.4 * 0.82}{1 - 0.82} \leq \frac{1.5 * 0.82}{1 - 0.82} \leq \frac{1.6 * 0.838}{1 - 0.838} \\
&\leq \frac{1.7 * 0.838}{1 - 0.838} \leq \frac{1.8 * 0.82}{1 - 0.82} \leq \frac{2 * 0.82}{1 - 0.82} \leq \frac{2.2 * 0.82}{1 - 0.82} \leq \frac{2.2 * 0.838}{1 - 0.838} \\
&\leq \frac{2.4 * 0.838}{1 - 0.838} \leq \frac{2.5 * 0.82}{1 - 0.82} \leq \frac{2.6 * 0.82}{1 - 0.82} \leq \frac{2.8 * 0.838}{1 - 0.838} \leq \frac{3 * 0.82}{1 - 0.82} \leq \frac{3 * 0.82}{1 - 0.82} \\
&\leq \frac{3.1 * 0.838}{1 - 0.838} \leq \frac{3.4 * 0.82}{1 - 0.82} \leq \frac{3.4 * 0.838}{1 - 0.838} \leq \frac{3.5 * 0.82}{1 - 0.82} \leq \frac{3.8 * 0.82}{1 - 0.82} \\
&\leq \frac{3.8 * 0.838}{1 - 0.838} \leq \frac{4 * 0.82}{1 - 0.82} \leq \frac{4 * 0.838}{1 - 0.838} \leq \frac{4.5 * 0.82}{1 - 0.82} \leq \frac{4.5 * 0.838}{1 - 0.838} \\
&\leq \frac{4.6 * 0.838}{1 - 0.838} \leq \frac{5.2 * 0.838}{1 - 0.838} \leq \frac{5.2 * 0.838}{1 - 0.838} \leq \frac{5.9 * 0.838}{1 - 0.838}
\end{aligned}$$

На основе этих неравенств определяем соответствующий порядок выполнения элементарных запросов и время выполнения вложенных запросов при совместной обработке при различном числе процессоров, приведенные в таблице 3.16.

Таблица 3.16. Время выполнения вложенного запроса при совместной обработке упорядоченных таблиц 3.5-3.7

| Процессоры $r=i,j$ | $k=32$ | Время выполнени я i -ом процессор/ n | Максим время выполн ения/ n |
|-----------------------|---|--|--|
| | $p_1 = 0.82, \Delta; p_2 = 0.838; \Delta_1 = 0.4, \Delta_2 = 0.5, \Delta_3 = 0.6, \Delta_4 = 0.7$ | | |
| | Порядок Отношение Эз | | |
| $r=1,1$ | 1,1,1,1,1.4,1.5,1.6,1.7 1.8,2,2.2,2.2,2.4,2.5,2.6,2.8 3, 3,3.1,3.4,3.4,3.5,3.8,3.8 4,4,4.5,4.5,4.6,5.2,5.2,5.9 | 6.9152 | 6.9152 |
| $r=2,1$ | 1,1,1.4,,1.7,1.8,2.2,2.4,2.8,3,3.4,3.4,3.8,4,4.5,4.6,5.9 | 9.1547 | 9.1547 |
| $r=2,2$ | 1,1,1.5,1.6,2,2.2,2.5,2.6,3,3.1,3.5,3.8, 4,4.5,5.2,5.2 | 8.7411 | |
| $r=4,1$ | 1,1.7,1.8,2.8,3,3.8,4,5.9 | 9.3553 | 9.3553 |
| $r=4,2$ | 1,1.6,2,2.6,3,3.8,4,5.2 | 8.7408 | |

| | | | | |
|-----------|-------------------------------|--------|--------|--------|
| $r=4,3$ | 1,1.5,2.2,2.5,3.1,3.5,4.5,5.2 | 8.8811 | | |
| $r=4,4$ | 1,1.4,2.2,2.4,3.4,3.4,4.5,4.6 | 9.0194 | | |
| $r=8,1$ | 1, 2.2,3,5.9 | 6.7756 | 7.1184 | |
| $r=8,2$ | 1,2.6,3,5.2 | 6.6250 | | |
| $r=8,3$ | 1,2.5,3.1,5.2 | 6.8503 | | |
| $r=8,4$ | 1,2.4,3.4,4.6 | 6.7010 | | |
| $r=8,5$ | 1.4,2.2,3.4,4.5 | 6.7891 | | |
| $r=8,6$ | 1.5, 2.8 ,3.5,4.5 | 7.1184 | | |
| $r=8,7$ | 1.6,2,3.8,4 | 6.7451 | | |
| $r=8,8$ | 1.7,1.8,3.8,4 | 6.7384 | | |
| $r=16,1$ | 1,5.9 | 4.8742 | | 4.8742 |
| $r=16,2$ | 1,5.2 | 4.3932 | | |
| $r=16,3$ | 1,5.2 | 4.4897 | | |
| $r=16,4$ | 1,4.6 | 4.0683 | | |
| $r=16,5$ | 1.4,4.5 | 4.2402 | | |
| $r=16,6$ | 1.5,4.5 | 4.2558 | | |
| $r=16,7$ | 1.6,4 | 4.1498 | | |
| $r=16,8$ | 1.7,4 | 4.1732 | | |
| $r=16,9$ | 1.8,3.8 | 4.0872 | | |
| $r=16,10$ | 2,3.8 | 4.1951 | | |
| $r=16,11$ | 2.2,3.5 | 4.1574 | | |
| $r=16,12$ | 2.2,3.4 | 4.2312 | | |
| $r=16,13$ | 2.4,3.4 | 4.3475 | | |
| $r=16,14$ | 2.5,3.1 | 4.1802 | | |
| $r=16,15$ | 2.6,3 | 4.1492 | | |
| $r=16,16$ | 2.8,3 | 4.4079 | | |
| $r=32,32$ | 1 до 32 | 4.9442 | 4.9442 | |

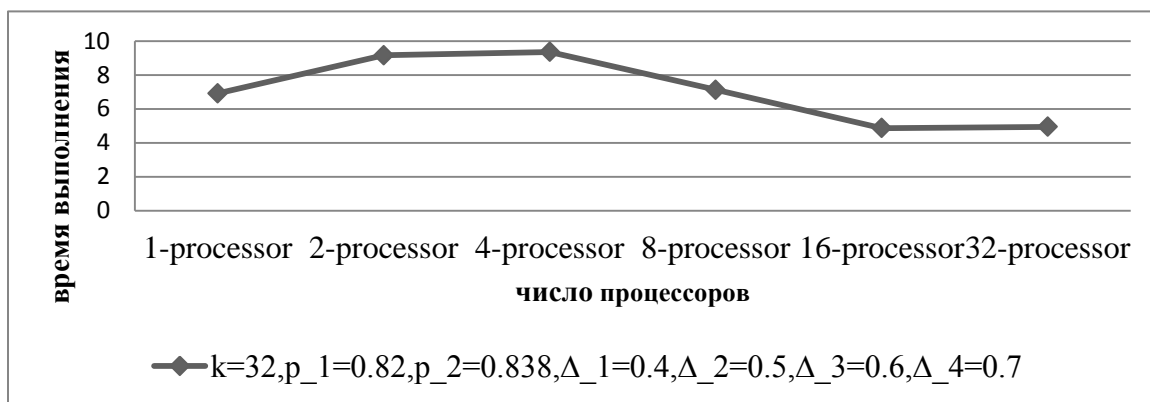


Рис.3.9. Время выполнения вложенного запроса при совместной обработке упорядоченных таблиц соответствующими арифметической прогрессии.

В таблице 17 приведены сравнительные времена выполнения вложенных запросов при совместном и несовместном порядке обработке упорядоченных и неупорядоченных данных таблиц с параметрами соответствующими арифметической и геометрической прогрессий для разного числа процессоров.

Таблица 3.17. Сравнительное время выполнения

| Число процессоров <i>r</i> | <i>k</i> =32 | | | | | | | |
|-------------------------------|--|---------|---------------|--------|-----------------|--------|---------------|--------|
| | $p_1 = 0.82, p_2 = 0.838; a_1=1.15, a_2 = 1.2, a_3 = 1.25, a_4=1.3;$ | | | | | | | |
| | $\Delta_1 = 0.4, \Delta_2 = 0.5, \Delta_3 = 0.6, \Delta_4 = 0.7$ | | | | | | | |
| | несовместное | | | | совместное | | | |
| | неупорядоченные | | упорядоченные | | неупорядоченные | | упорядоченные | |
| арифм | геомет | арифм | геомет | арифм | геомет | арифм | геомет | |
| 1 | 9.6262 | 10.2142 | 9.4470 | 7.0836 | 7.9772 | 9.2644 | 6.9152 | 5.7856 |
| 2 | 11.0238 | 9.8301 | 10.7776 | 8.1025 | 10.4126 | 8.3110 | 9.1547 | 7.0642 |
| 4 | 10.4727 | 10.0750 | 10.0994 | 8.3380 | 10.9213 | 8.7474 | 9.3553 | 6.9707 |
| 8 | 7.8943 | 8.1463 | 7.1952 | 6.7557 | 8.7303 | 7.5034 | 7.1184 | 6.1360 |
| 16 | 5.2022 | 6.1454 | 4.8742 | 5.1318 | 5.8380 | 6.1454 | 4.8742 | 5.1318 |
| 32 | 5.9 | 6.27 | 4.9442 | 5.2542 | 5.9 | 6.27 | 4.9442 | 5.2542 |

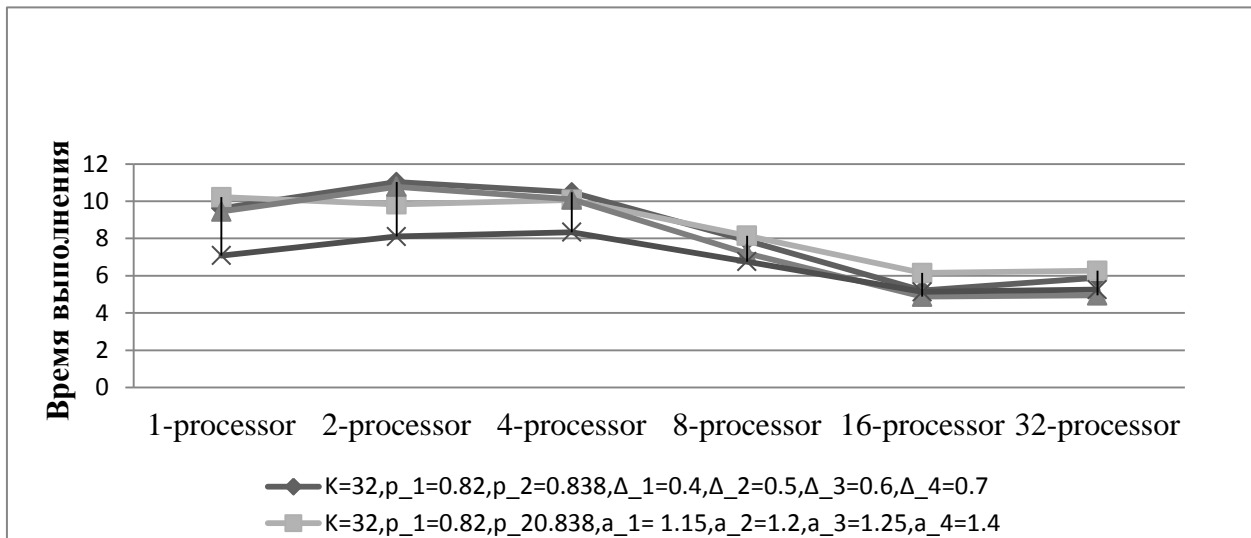


Рис.3.10. Сравнительное несоместное время выполнения для арифметическая прогрессия и геометрическая прогрессия для разного числа процессоров.

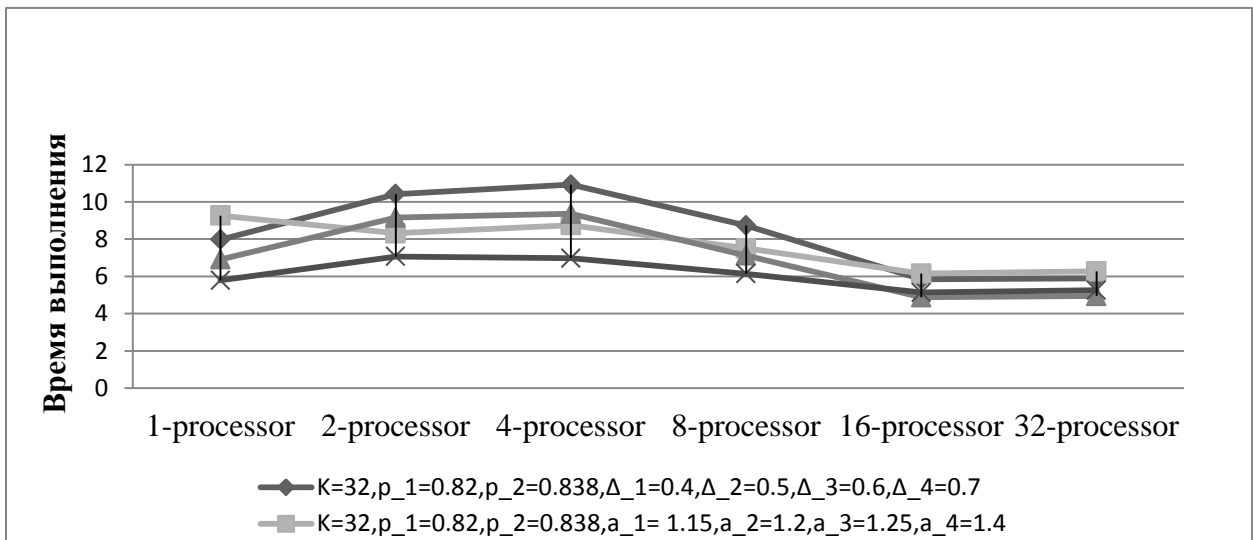


Рис.3.11. Сравнительное совместное время выполнения для арифметическая прогрессия и геометрическая прогрессия для разного числа процессоров.

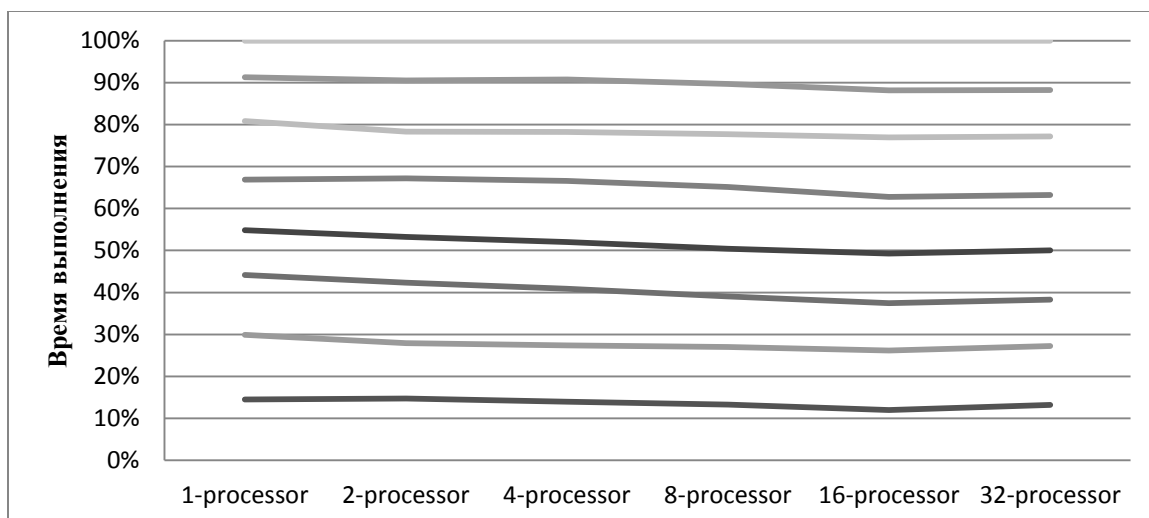


Рис.3.12. Сравнительное несоместное и совместное время выполнения

На основании сравнения времен выполнения вложенных запросов следует, что минимальное время выполнения вложенного запроса достигается при некотором (не обязательно максимальном) числе процессоров.

Выводы по главе 3

1. Разработана методика оптимизации по времени выполнения конъюнктивных вложенных запросов при обращении к многопроцессорной базе данных на основе упорядочивания элементарных запросов.
2. Определено минимальное время выполнения вложенного запроса для упорядоченных или неупорядоченных данных таблиц при совместной обработке процессорами данных объединенного множества элементарных запросов всех таблиц, образующих вложенные запросы.
3. Определено минимальное число процессоров, при котором достигается минимальное время выполнения вложенного запроса, что является важным решением для оптимизации многопроцессорных баз данных авиационно-космических систем.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Разработана методика оптимизации по времени выполнения конъюнктивных вложенных запросов при обращении к многопроцессорной базе данных на основе упорядочивания элементарных запросов.
2. Определены соотношения времени выполнения запроса в многопроцессорной базе данных для естественного и квазиоптимального порядка их распределения.
3. Доказана эффективность квазиоптимального распределения на основе абсолютного и относительного уменьшения границ времени выполнения запросов при использовании квазиоптимального распределения вместо естественного распределения.
4. Определено минимальное время выполнения вложенных запросов для упорядоченных или неупорядоченных данных таблиц при совместной обработке процессорами данных объединенного множества элементарных запросов всех таблиц, образующих вложенные запросы.
5. Определено минимальное число процессоров, при котором достигается минимальное время выполнения вложенного запроса, что является важным решением для оптимизации многопроцессорных баз данных авиационно-космических систем.

Литература

1. Андреев А.Н., Воеводин Вл.В., Жуматий С.А. Кластеры и суперкомпьютеры - близнецы или братья? // Открытые системы. -2000.
2. Байкова И., Кулагин М. Современные дисковые системы RAID //Открытые системы. 1995.
3. Барон Г.Г. Параллельные архитектуры серверов баз данных // СУБД.-1995.
4. Баженова И. Ю. Основы проектирования приложений баз данных, ИНТУИТ.ру, 2006.
5. Безкоровайный М.М., Костогрызов А.И., Львов В.М. Инструментально-моделирующий комплекс для оценки качества функционирования информационных систем "КОК". М.:СИНТЕГ, 2000.
6. Брехов О.М. Аналитическая оценка оптимальной обработки запросов // Успехи современной радиоэлектроники. 2012. Т.12. №7. С. 37-45.
7. Брехов О.М., Мьо Тант. Оптимизация обработки запросов в многопроцессорной базе данных // Вестник Московского авиационного института. 2012. Т.19. № 5. С.138-146.
8. Брехов О.М., Вунна Джо Джо., Тан Хлаинг Мьинт. Оптимизация плана выполнения мульти и вложенных запросов // Журнал «Наукоёмкие технологии» 2014г. №1, с. 101-106.
9. Брехов О.М. , Тан Хлаинг Мьинт. Обоснование квазиоптимального порядка распределения элементарных запросов в многопроцессорной базе данных // Электронный журнал «Труды МАИ», 2014, № 74.
10. Брехов О.М. , Тан Хлаинг Мьинт. Оптимизация числа процессоров при выполнении вложенных запросов // Электронный журнал «Труды МАИ», 2014.
11. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. -М.: Мир, 1985.
12. Воеводин Вл.В., Капитонова А.П. Методы описания и классификации архитектур вычислительных систем. -М: Изд-во МГУ, 1994.
13. Герберт Шилдт. Полный справочник по С. Изд-во Вильямс, 2009.
14. Гольдштейн М.Л. Мультипроцессорная вычислительная система на базе транспьютерной идеологии // Алгоритмы и программные средства параллельных вычислений: Сб. науч. тр. / ИММ УрО РАН.Екатеринбург: УрО РАН, 1995.

15. Григорьев Ю.А. Разработка научных основ проектирования архитектуры распределенных систем обработки данных: Дис. д-ра техн. наук. МГТУ им. Н.Э. Баумана, 17.10.96 г.
16. Н.А. Гребенников, В.М. Постников //Организация баз данных, 2002 г. (Московский государственный технический университет им. Н.Э. Баумана).
17. Когаловский М.Р. Энциклопедия технологий баз данных. -М: Финансы и статистика, 2002.
18. Корнеев В. Архитектуры с распределенной разделяемой памятью // Открытые системы.
19. Корнеев В.В. Параллельные вычислительные системы. -М.: "Нолидж", 1999.
20. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. 2-е издание. -М.: Нолидж, 2001.
21. Кузнецов С.Д. – «Базы данных: языки и модели», Москва, Бином, 2008, 720 с.
22. Кузнецов С.Д. Основы баз данных. - М.: Изд-во "Интернет-университет информационных технологий - ИНТУИТ.ру", 2005.
23. Кузнецов С.Д. Логическая оптимизация запросов в реляционных СУБД // Программирование.
24. Кузнецов С.Д. Методы оптимизации выполнения запросов в реляционных СУБД // Сб. Итоги науки и техники. Вычислительные науки. -Т.1. -М: ВИНТИ, 1989.
25. Кузнецов С.Д. Операционные системы для управления базами данных // СУБД.
26. Кузнецов С.Д. СУБД и файловые системы. -М: Майор, 2001.
27. Кузьминский М, Волков Д. Современные суперкомпьютеры: состояние и перспективы // Открытые системы. -1995.
28. Лисянский К., Слободяников Д. СУБД Teradata для ОС UNIX // СУБД. 1997.
29. Постников В.М., Гребенников Н.А. Технология обработки запросов пользователей в СУБД ORACLE // Вестник МГТУ им. Н.Э. Баумана. Сер. "Приборостроение". 2001. № 2. С.106–126.
30. Соколинский Л.Б. Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической архитектурой // Программирование.

31. Соколинский Л.Б. Параллельные машины баз данных // Природа. Естественно-научный журнал Российской академии наук. -2001.
32. Сиротюк В.О. Модели и методы синтеза оптимальных логических структур и базы метаданных репозитария распределенных баз данных в АСУ // Автоматика и телемеханика (М.). 1999. № 1.
33. Хаманн Ф. Отказоустойчивая операционная система Tandem NonStop Kernel // Открытые системы.
34. Шмидт В. Системы IBM SP2 // Открытые системы.
35. Шнитман В. Отказоустойчивые серверы ServerNet // Открытые системы.
36. An Environment for the Design and Performance Evaluation of Portable Parallel Software: Final EDPEPPS Simulator: Report / Center for Parallel Computing University of Westminster. EDPEPPS/41. London, July 1997.
37. Amza C., et al. ThreadMarks: Shared Memory Computing on Networks of Workstations // IEEE Computer. -1996. -Vol. 29, No. 2.
38. Amol Deshpande, Zacchary Ives, Vijayshankar Raman – Adaptive Query Processing // Foundations and Trends in Databases. -2007. –Vol.1, No.1(2007), p.1-140.
39. Apers P.M.G., van den Berg C.A., Flokstra J., Grefen P.W.P. J., Kersten M.L., Wilschut A.N. Prisma/DB: a Parallel Main-Memory Relational DBMS // IEEE Transactions on Knowledge and Data Engineering. -1992. -Vol. 4, No. 6.
40. Ballinger C., Fryer R. Born To Be Parallel: Why Parallel Origins Give Teradata an Enduring Performance Edge // IEEE Data Engineering Bulletin. -1997. -Vol. 20, No. 2.
41. Baru C. K., et al. DB2 Parallel Edition // IBM System Journal. -1995. -Vol. 34, No. 2.
42. Bell C.G., Gray J. What's next in high-performance computing? // Communications of the ACM. -2002. -Vol. 45, No. 2.
43. Bellman R.E. Dynamic Programming, Princeton University Press, Princeton, N. Jersey, 1975.
44. Bergsten B., Couprie M., Lopez M. DBS3: A Parallel Data Base System for Shared Store (Synopsis) // Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems (PDIS 1993) ,Issues, Architectures, and Algorithms, San Diego, CA, USA, January 20-23, 1993.IEEE Computer Society, 1993.

45. Bergsten B., Couprie M., Valdurez P. Overview of Parallel Architectures for Databases // The Computer Journal. -1993. -Vol. 36, No. 8.
46. Bergsten B., Couprie M., Valdurez P. Prototyping DBS3, a Shared-Memory Parallel Database System // Proceedings of the First International Conference on Parallel and Distributed Information Systems (PDIS 1991), Fontainebleu Hilton Resort, Miami Beach, Florida, December 4-6, 1991. -IEEE Computer Society, 1991.
47. Boral H., Alexander W., Clay L., Copeland G., Sanforth S., Franklin M., Hart B., Smith M., Valdurez P. Prototyping Bubba: a Highly Parallel Database System // IEEE Trans. on Knowledge and Data Engineering. -1990. -Vol. 2, No. 1.
48. Boral H., DeWitt D.J. Applying Data Flow Techniques to Data Base Machines // IEEE Computer. -1982. -Vol. 15, No. 8.
49. Bouganim L., Florescu D., Valdurez P. Dynamic Load Balancing in Hierarchical Parallel Database Systems // VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3-6, 1996, Mumbai (Bombay), India. -Morgan Kaufmann, 1996.
50. Bultzingsloewen G. Optimizing SQL Queries for Parallel Execution // ACM SIGMOD Record. -1989. -Vol. 18, No. 4.
51. Bultzingsloewen G.v., et al. KARDAMOM - A Dataflow Database Machine For Real-Time Applications // SIGMOD Record. -1988. -Vol. 17, No. 1.
52. Carr J.L., Hennessy J.L. WSClock - A Simple and Effective Algorithm for Virtual Memory Management // Proceedings of the Eighth Symposium on Operating System Principles, 14-16 December 1981, Asilomar Conference Grounds, Pacific Grove, California. -New York: ACM, 1981.
53. [Codd, Edgar Frank: "Is Your DBMS Really Relational?", ComputerWorld, 14. October 1985.](#)
54. Chambers L., Cracknell D. Parallel Features of NonStop SQL // Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems (PDIS 1993), Issues, Architectures, and Algorithms, San Diego, CA, USA, January 20-23, 1993. -IEEE Computer Society, 1993.
55. Chen M.-S., Yu P.S., Wu K.-L. Optimization of Parallel Execution for Multi-Join Queries // IEEE Transactions on Knowledge and Data Engineering. -1996. -Vol. 8, No. 3.

56. Chen M.-S., Yu P.S., Wu K.-L. Scheduling and Processor Allocation for Parallel Execution of Multi-Join Queries // Proceedings of the Eighth International Conference on Data Engineering, February 3-7, 1992, Tempe, Arizona. -IEEE Computer Society, 1992.
57. Cheng J.M., et al. IBM Database 2 Performance: Design, Implementation, and Tuning // IBM Systems Journal. -1984. -Vol. 23, No. 2.
58. Christodoulakis S. Estimating record selectivities // Information Systems. -1983. -Vol. 8, No. 2.
59. Clay D. Informix Parallel Data Query (PDQ) // Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems (PDIS 1993), Issues, Architectures, and Algorithms, San Diego, CA, USA, January 20-23, 1993. -IEEE Computer Society, 1993.
60. Copeland G.P., Keller T. A Comparison Of High-Availability Media Recovery Techniques // roceedings of the 1989 ACM SIGMOD International Conference on Management of Data, Portland, Oregon, May 31 - June 2, 1989. -ACM Press, 1989.
61. Dasgupta S. A Hierarchical Taxonomic System for Computer Architectures // IEEE Computer. -1990. -Vol. 23, No. 3.
62. Davison W. Parallel Index Building in Informix OnLine 6.0 // Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data, San Diego, California, June 2-5, 1992. -ACM Press, 1992.
63. DeWitt D.J., Gerber R.H. Multiprocessor Hash-Based Join Algorithms // VLDB'85, Proceedings of 11th International Conference on Very Large Data Bases, August 21-23, 1985, Stockholm, Sweden. -Morgan Kaufmann, 1985.
64. DeWitt D.J., Gray J. Parallel Database Systems: The Future of High-Performance Database Systems // Communications of the ACM. -1992. -Vol. 35, No. 6.
65. DeWitt D.J., Naughton J.F., Schneider D.A. Parallel Sorting on a Shared-Nothing Architecture using Probabilistic Splitting // Proceedings of the First International Conference on Parallel and Distributed Information Systems (PDIS 1991). Fontainebleu Hilton Resort, Miami Beach, Florida, December 4- 6, 1991. -IEEE-CS, 1991.
66. DeWitt D.J., et al. The Gamma database machine project // IEEE Transactins on Knowledge and Data Engineering. -1990. -Vol. 2, No. 1.

67. Effelsberg W., Haerder T. Principles of Database Buffer Management // ACM Trans. on Database Systems. - 1984. -Vol. 9, No. 4.
68. E.F. Codd: "Relational Completeness of Data Base Sublanguages" (presented at Courant Computer Science Symposia Series 6, "Data Base Systems", New York City, N.Y., May 24th-25th, 1971).
69. Englert S., Glasstone R., Hasan W. Parallelism and its Price: A Case Study of NonStop SQL/MP // ACM SIGMOD Record. -1995. -Vol. 24, No. 4.
70. Flynn M.J. Computer Organization and Architecture // Operating Systems, An Advanced Course. -Springer, 1978 (Lecture Notes in Computer Science; Vol. 60).
71. Flynn M.J. Very High Speed Computing Systems // Proc. IEEE. -1966. -Vol. 54. -P. 1901-1909.
72. Flynn M.J., Rudd K.W. Parallel architectures // ACM Computing Surveys. 1996. -Vol. 28, No. 1.
73. Gardarin G., Sha F., Tang Z.-H. Calibrating the query optimizer cost model of IRO-DB an objectoriented federated database system // Proceedings of the 22nd International Conference on VLDB. Bombay, 1996. P.378-389.
74. Gibson G.A., Vitter J.S., Wilkes J. Strategic directions in storage I/O issues in large-scale computing // ACM Computing Surveys. - 1996. -Vol. 28, No. 4. -P. 779-793.
75. Graefe G. Encapsulation of Parallelism in the Volcano Query Processing Systems // Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25, 1990. -ACM Press, 1990.
76. Graefe G. Query evaluation techniques for large databases // ACM Computing Surveys. - 1993. -Vol. 25, No. 2.
77. Graefe G. Volcano - An Extensible and Parallel Query Evaluation System // IEEE Transactions on Knowledge and Data Engineering. -1994. -Vol. 6, No. 1.
78. Graefe G. Query evaluation techniques for large databases // ACM Computing Survey N.Y., 1993. Vol. 25. No. 2. P.73-170.
79. Graefe G. Query Evaluation Techniques for Large Databases // ACM Computing Surveys. – 1993. –Vol. 25, No 2. –P. 73–169.

80. Graefe Goetz, Dewitt David J. The EXODUS Optimizer Generator. In Proceedings 1987 ACM SIG-MOD International Conference on Management of Data, San Francisco, 1987. P.387-394.
81. Graefe Goetz, McKenna William. The Volcano Optimizer Generator: Extensibility and Efficient Search. In Proceeding of the 12th International Conf. on Data Engineering, 1993. P.209-218.
82. Graefe G. The Cascades Framework for Query Optimization, Bulletin of the IEEE Technical Committee on Data Engineering, 18(3), September 1995, P.19-29.
83. Gray J. Notes on Data Base Operating Systems // Operating Systems, An advanced Course.- Springer, 1978 (Lecture Notes in Computer Science; Vol. 60).
84. Gray J., Graefe G. The Five-Minute Rule Ten Years Later, and Other Computer Storage Rules of Thumb // SIGMOD Record. -1997. -Vol. 26, No. 4.
85. Hasan W. Optimization of SQL Queries for Parallel Machines. (Lecture notes in computer science, Vol. 1182). -Berlin, New York: Springer, 1996.
86. Hitz M., Muck T.A. Interconnection Topologies for Shared Nothing DBMS Architectures // Advances in Database Research, Proceedings of the 4th Australian Database Conference ADC'93, Griffith University, Brisbane, Queensland, Australia, 1-2 February 1993. -World Scientific, 1993.
87. Hong W., Stonebraker M. Optimization of Parallel Query Execution Plans in XPRS // Distributed and Parallel Databases. -1993. -Vol. 1, No. 1.
88. Hsiao H. I., DeWitt D. J. A Performance Study of Three High Availability Data Replication Strategies // Distributed and Parallel Databases. -1993. -Vol. 1, No. 1.
89. Hua K.A., Lee C., Peir J.-K. Interconnecting Shared-Everything Systems for Efficient Parallel Query Processing // Proceedings of the First International Conference on Parallel and Distributed Information Systems (PDIS 1991). Fontainebleu Hilton Resort, Miami Beach, Florida, December 4-6, 1991. -IEEE-CS, 1991.
90. Hwang H-Y., Yao-Tin Y. An Analytical Method for Estimating and Interpreting QueryTime // Proceedings of the 13th VLDB Conference. Brighton, 1987.
91. Ibaraki T., Kameda T. On the optimal nesting order for computing N-relational joins. //ACM Transactions on Database Systems. 1984. 9(3). P.482-502.

92. Jarke M., Koch J. Query optimization in database systems // ACM Computing Surveys. - 1984. -Vol. 16, No. 2.
93. Kim W. Highly Available Systems for Database Applications // ACM Computing Surveys. - 1984. -Vol. 16, No. 1.
94. King G. M., Dias D. M., Yu P. S. Cluster Architectures and S/390 Parallel Sysplex Scalability // IBM Systems Journal. -1997. -Vol. 36, No. 2.
95. Kotsis G. Interconnection Topologies for Parallel Processing Systems // PARS Mitteilungen. -1993.
96. Kronenberg N.P., Levy H.M., Strecker W.D. VAXclusters: A Closely-Coupled Distributed System // ACM Transactions on Computer Systems. -1986. -Vol. 4, No. 2.
97. Lakshmi M.S., Yu P.S. Effectiveness of Parallel Joins // IEEE Transactions on Knowledge and Data Engineering. -1990. -Vol. 2, No. 4.
98. Linder B. Oracle Parallel RDBMS on Massively Parallel Systems // Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems (PDIS 1993), Issues, Architectures, and Algorithms, SanDiego, CA, USA, January 20-23, 1993. -IEEE Computer Society, 1993.
99. Lohman G.M., et al Query Processing in R* // Query Processing in Database Systems. - Springer, 1985.
100. Lorie R., et al. Adding Intra-transaction Parallelism to an Existing DBMS: Early Experience // Data Engineering Bulletin. -1989. -Vol. 12, No. 1.
101. Lorie R.A., Young H.C. A Low Communication Sort Algorithm for a Parallel Database Machine // Proceedings of the Fifteenth International Conference on Very Large Data Bases, August 22-25, 1989, Amsterdam, The Netherlands. -Morgan Kaufmann, 1989.
102. Lu H., Shan M.-C., Tan K.-L. Optimization of Multi-Way Join Queries for Parallel Execution // 17th International Conference on Very Large Data Bases, September 3-6, 1991, Barcelona, Catalonia, Spain, Proceedings. -Morgan Kaufmann, 1991.
103. Lynch C.A. Selectivity Estimation and Query Optimization in Large Databases with Highly Skewed Distribution of Column Values // Fourteenth International Conference on Very Large Data Bases, August 29 - September 1, 1988, Los Angeles, California, USA, Proceedings. -Morgan Kaufmann, 1988.

104. Matthias Jarke, Jurgen Koch. Query Optimization in Database Systems. Computing Surveys, Vol. 16, No. 2, June 1984.
105. Menon J. A Study of Sort Algorithms for Multiprocessor Database Machines // VLDB'86 Twelfth International Conference on Very Large Data Bases, August 25-28, 1986, Kyoto, Japan, Proceedings. -Morgan Kaufmann, 1986.
106. Method and system for query processing, United States Patent 6757670.
107. Michie D. 'Memo' Functions and Machine Learning, Nature, April 1968. No. 218, P.1922.
108. Montgomery A.Y., D'Souza D.J., Lee S.B. The Cost of Relational Algebraic Operations on Skewed Data: Estimates and Experiments // Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, France, September 19-23, 1983. -North-Holland: IFIP, 1983.
109. Nick J. M., Moore B. B., Chung J.-Y., Bowen N. S. S/390 Cluster Technology: Parallel Sysplex // IBM Systems Journal. -1997. -Vol. 36, No. 2.
110. Norman M. G., Zurek T., Thanisch P. Much Ado About Shared-Nothing // ACM SIGMOD Record. -1996. -Vol. 25, No. 3.
111. Oldehoeft R. R. Multithreaded Computer Systems // Proceedings Supercomputing'92, November 16-20, 1992, Minneapolis, MN, USA. -IEEE Computer Society, 1992.
112. Ozsu M.T., Valduriez P. Principles of Distributed Database System. Englewood Cliffs, NJ: Prentice-Hall, 1991.
113. Page J. A Study of a Parallel Database Machine and its Performance: the NCR/Teradata DBC/1012 // Advanced Database Systems, 10th British National Conference on Databases, BNCOD 10, Aberdeen, Scotland, July 6-8, 1992, Proceedings. -Springer, 1992 (Lecture Notes in Computer Science, Vol. 618).
114. Patterson D.A., Gibson G.A., Katz R.H. A Case for Redundant Arrays of Inexpensive Disks (RAID) // Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, June 1-3, 1988. ACM Press 1988. 1988.
115. Pfister G. Sizing Up Parallel Architectures // DataBase Programming & Design OnLine (<http://www.dbpd.com>). May 1998. -Vol. 11, No. 5.

116. Pirahesh Hamid, Hellerstein Joseph M., Hasan Waqar. Extensible/Rule Based Query Rewrite Optimization in Starburst. In Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data, pages 39-48, San Diego, California, June 1992.
117. Pirahesh H., Mohan C., Cheng J.M., Liu T.S., Selinger P.G. Parallelism in Relational Data Base Systems: Architectural Issues and Design Approaches // Proceedings of the Second International Symposium on Databases in Parallel and Distributed Systems, July 2-4, 1990, Trinity College, Dublin, Ireland. -IEEE-CS, 1990.
118. Pramanik S., Tout W.R. The NUMA with Clusters of Processors for Parallel Join // IEEE Transactions on Knowledge and Data Engineering. -1997. -Vol. 9, No. 4. -P. 653-666.
119. Query Processing in Parallel Relational Database Systems / [edited by] Lu H., Ooi B.-C., Tan K.-L. -IEEE Computer Society Press, 1994.
120. Rahm E. A Framework for Workload Allocation in Distributed Transaction Processing Systems // Journal of Systems and Software. -1992. -Vol. 18.
121. Rahm E. Parallel Query Processing in Shared Disk Database Systems // ACM SIGMOD Record. -1993. -Vol. 22, No. 4.
122. Richardson J.P., Lu H., Mikkilineni K. Design and Evaluation of Parallel Pipelined Join Algorithms // Proceedings of the ACM SIGMOD 1987 Annual Conference, San Francisco, California, May 27-29, 1987. -ACM Press, 1987.
123. Schneider D.A., DeWitt D.J. Tradeoffs in Processing Complex Join Queries via Hashing in Multiprocessor Database Machines // 16th International Conference on Very Large Data Bases, August 13-16, 1990, Brisbane, Queensland, Australia, Proceedings. -Morgan Kaufmann, 1990.
124. Selinger P.G., Astrahan M.M., Chamberlin D.D., Lorie R.A., Price T.G. Access Path Selection in a Relational Database Management System // Proc. ACM SIGMOD Int. Conf. Manag. Data, Boston, Mass., May 30 - June 1, 1979. New York, 1979.- C. 23-34.
125. Skelton C.J., et al. EDS: A Parallel Computer System for Advanced Information Processing // PARLE '92: Parallel Architectures and Languages Europe, 4th International PARLE Conference, Paris, France, June 15-18, 1992, Proceedings. -Springer, 1992 (Lecture Notes in Computer Science, Vol. 605).
126. Skillicorn D.B. A Taxonomy for Computer Architectures // IEEE Computer. -1988. -Vol. 21, No. 1.

127. Sokolinsky L.B. Choosing Multiprocessor System Architecture for Parallel Database Systems // Proc. of the 2nd Int. Workshop on Computer Science and Information Technologies (CSIT'2000), Ufa, Russia, September 18-23, 2000. Ufa State Aviation Technical University. -2000. -Vol. 1.
128. Sokolinsky L.B. Design and Evaluation of Database Multiprocessor Architecture with High Data Availability // 12th International DEXA Workshop, Munich, Germany, 3-7 September, 2001, Proceedings. -IEEE Computer Society, 2001.
129. SQLBase. Advanced Topic Guide. 20-2119-9901 SQL designs. Gupta Inc. 2006
130. Sterling T., et al. BEOWULF: A Parallel Workstation for Scientific Computation // Proceedings of the 1995 International Conference on Parallel Processing, August 14-18, 1995, Urbana-Champaign, Illinois, USA. Volume I: Architecture. -CRC Press, 1995.
131. Stonebraker M., Katz R.H., Patterson D.A., Ousterhout J.K. The Design of XPRS // Fourteenth International Conference on Very Large Data Bases, August 29 - September 1, 1988, Los Angeles, California, USA, Proceedings. -Morgan Kaufmann, 1988.
132. Stonebraker M. Operating System Support for Database Management // Communications of the ACM. -1981. -Vol. 24, No. 7.
133. Stonebraker M. The case for shared nothing // Database Engineering Bulletin. -1986. -Vol. 9, No. 1.
134. Strickland J.P., Uhrowczik P.P., Watts V.L. IMS/VS: An Evolving System // IBM Systems Journal. -1982. -Vol. 21, No. 3.
135. Su S.Y.W., Mikkilineni K.P., Liuzzi R.A., Chow Y.C. A Distributed Query Processing Strategy Using Decomposition, Pipelining and Intermediate Result Sharing Techniques // Proceedings of the Second International Conference on Data Engineering, February 5-7, 1986, Los Angeles, California, USA. -IEEE Computer Society, 1986.
136. The Benchmark Handbook for Database and Transaction Processing Systems, Second Edition. -Morgan-Kaufmann, 1993.
137. Valduriez P. Parallel Database Systems: Open Problems and New Issues // Distributed and Parallel Databases. -1993. -Vol. 1, No. 2.
138. Valduriez P., Gardarin G. Join and Semijoin Algorithms for a Multiprocessor Database Machine // ACM Transactions on Database Systems. -1984. -Vol. 9, No. 1.

139. Van Vleck T. H. , Clingen C. T. The Multics System Programming Process // Proceedings of the 3rd International Conference on Software Engineering, May 10-12,1978, Atlanta, Georgia, USA. -IEEE Computer Society, 1978.
140. W. Hasan, D. Florescu, P. Valduriez Open Issues in Parallel Query Optimization // ACM SIGMOD Record. -1996. -Vol. 25, No. 3.
141. Williams M.H., Zhou S. Data Placement in Parallel Database Systems // Parallel database techniques. IEEE Computer society, 1998.
142. Wong, Eugene, Youssefi, K., Decomposition - A Strategy for Query Processing. ACM Transactions on Database Systems, 3 (September) 1976.
143. Xu Y., Dandamudi S.P. Performance Evaluation of a Two-Level Hierarchical Parallel Database System // Proceedings of the Int. Conf. Computers and Their Applications, Tempe, Arizona, 1997.
144. Yannis E. Ioannidis: Query Optimization.ACM Comput. Surv. 28(1): 121-123 (1996)
145. Zipf G.K. Human Behavior and the Principle of Least Effort: an Introduction to Human Ecology. -Cambridge, Mass.: Addison-Wesley, 1949.
146. Amol Deshpande, Zacchary Ives, Vijayshankar Raman – Adaptive Query Processing // Foundations and Trends in Databases. -2007. –Vol.1, No.1(2007), p.1-140.
147. <http://zavantag.com/docs/2307/index-331497.html>
148. <http://5fan.ru/wievjob.php?id=22500>
149. http://citforum.ru/database/sql_kg/3-3-1.shtml