Литература

1. Грищук И.П. Компиляторы. –М., Высшая школа, 1992.
2. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. –М., Издательство «МИР», 1975.

ОРГАНИЗАЦИЯ ГРУППОВЫХ ЗАПРОСОВ К МНОГОТАБЛИЧНЫМ ДАННЫМ

Галерштейн В.Е., Ревотюк М.П.
Белорусский государственный университет информатики и радиоэлектроники
Минск, ул. П.Бровки, 6, кафедра ИТАС

Предлагается метод оптимизации групповых запросов к многотабличным данным, обсуждаются условия его применимости и оценка выигрыша в скорости доступа по сравнению с традиционными подходами. Рассматриваются способы его реализации при разработке новых СУБД.

Ключевые слова: базы данных, методы доступа к данным, оптимизация запросов.

В реальных системах обработки информации часто возникает задача выборки данных из нескольких таблиц данных, связанных по некоторому ключу. Например, в системах семейства xBase подобная связь устанавливается оператором SET RELATION, а в SQL - системах - предикатом выражения WHERE оператора SELECT [8,10]. Организация выборки записей из группы таблиц наиболее эффективно строится, как правило, на основе индексных файлов. Каждой таблице ставится в соответствие один индексный файл.

Известно, что наиболее распространенная организация индексных файлов - деревья Байера (B+ деревья) [2,3,6]. Среднее время доступа T к искомой записи в B+ дереве [3,4]:

$$T \sim = \log_a n, \quad (1)$$

где n - число записей таблицы, а - число индексов на одной странице B+ дерева.

Если применять традиционную схему последовательной обработки всех M связанных таблиц, то среднее время обработки T(M) составит:

$$T(M) = \Sigma \log_a n_i, \; i = 1 \ldots M. \quad (2)$$

Логарифмический закон времени доступа к индексированной таблице (1) побуждает предложить следующую схему организации групповых запросов. Построим индексный файл, где выражения ключа включают ссылку на идентификатор таблицы, а также флаг уникальности ключа для данной таблицы:

$$Kc = K + TableID(i) + Unicue(i),\ i = 1..M \qquad (3)$$

где Kc - выражение ключа групповой связи таблиц, TableID(i) - идентификатор таблицы, Unicue(i) - флаг уникальности ключа.

Так как количество отображаемых записей в групповом индексном файле

$$N = \Sigma n_i,\ i = 1 \ldots M \qquad 4)$$

а схема его организации - B+ дерево, то среднее время доступа:

$$T'(M) = \log_a \Sigma n_i,\ i = 1 \ldots M \qquad (5)$$

Преимущество введения группового индексного файла объясняется следующим из (2) и (5) выражением:

$$\Pi n_i > \Sigma n_i,\ i = 1 \ldots M \qquad (6)$$

Обсудим вопрос реализации групповых запросов по предложенной схеме. Например, в системе FOXPRO, являющейся типичным представителем xBase систем, реализация (3) может выглядеть так:

```
* СОЗДАНИЕ БАЗ ДАННЫХ С КЛЮЧЕМ
 CREATE TABLE T1 (K C(10), D1 N(16,2))
 CREATE TABLE T2 (K C(10), D2 C(80))
 CREATE TABLE TM (K C(10), DM D(8))
* ИНДЕКСИРОВАНИЕ БАЗ ДАННЫХ:
 FOR I=0 TO M
 PRIV TABLEID
 TABLEID='T'+STR(I)
 SELECT (TABLEID)
 INDEX ON K+TABLEID TAG KC ADDITIVE
 END FOR
```

Операция создания и индексирования баз данных может выполняться и операторами языка SQL [9].

Индексные файлы T1.CDX ... TM.CDX далее требуется слить в единый файл, что в системе FOXPRO не предусмотрено, но подобная задача разрешима при программировании средствами более низкого уровня, причем ее вычислительная сложность не превосходит $N \log_2 N$ [1,3,6].

Выборка данных, в общем случае, требует обеспечения механизма обращения к записям отдельных таблиц с форматами различных СУБД. Эта технологическая задача в настоящее время тривиальна и может быть решена с помощью таких стандартизованных средств доступа, как ODBC, DAO, IDAPI, OLE DB [9]. Например, низкоуровневые средства работы с реляционными базами

данных, удовлетворяющие стандарту ISSAM (INFORMIX C ISSAM, Btree Filer Turbo Power, POET Object Database System) [7,10] реализуют следующую технологию работы с ключами и данными:

DBRes GetKey(DBFile file, KeyRec& Key); //Выборка ключа записи по маске

DBRes GetRecord(DBFile, KeyRec Key, DataRec& Rec); // Выборка по ключу

Используя подобные операции, не составляет труда последовательно получить все записи ключей из каждого индексного файла и переслать их в групповой индексный файл, где ключ записи определяется (3).

Очевидно, что современные СУБД не обладают средствами поддержки групповых индексных файлов. Однако при разработке новых систем обработки данных, например, в среде C++, реализация такого механизма вполне реальна.

Рассмотрим случай параллельной работы приложения с групповым индексным файлом и индексным файлом одной из исходных баз данных. Этот случай может иметь место при выполнении операций удаления, добавления или модификации данных в одной из исходных баз данных. Очевидно, что при использовании одиночного индексного файла подобные запросы будут выполняться быстрее, чем при использовании группового индексного файла. В современных СУБД решение о стратегии доступа принимает оптимизатор запросов. Для обеспечения корректности ключей в групповом индексном файле, оптимизатор запросов может породить фоновую задачу модификации группового индексного файла. Это устранит замедление выполнения запроса к одиночной базе данных.

Литература:

1. Атре Ш. Структурный подход к организации баз данных – М.: Финансы и статистика, 1983.
2. Вирт Н. Систематическое программирование – М.: Мир, 1977.
3. Вирт Н. Алгоритмы + структуры данных = программы – М.: Мир, 1985.
4. Дж.Ульман Базы данных на Паскале – М.: Машиностроение, 1990.
Дж. Ульман Основы систем баз данных – М.: Мир, 1980.
5. Кнут Д. Искуство программирования для ЭВМ . Т.3.- М.: Мир, 1978.
6. Smith J.M., and Smith, D.C.P. Database abstractions: Aggregation/Commun. ACM, 20, June, 1997.

7. Smith David, Post-Relational Database:Revitalizing Relational Technology for New Applications. - IDC White Papper, March 1994.
8. ANSI X3.135-1992, American National Standard for Information Systems - Database Language - SQL, November, 1992.
9. ANSI/X3/SPARC DMBS Framework. Report of the study group on data base management systems Inform. Systems, 1978

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ РАБОТЫ С ИЗОБРАЖЕНИЯМИ

Молочко Д. А., Муха В. С.
Белорусский государственный университет информатики и радиоэлектроники
Минск, ул П.Бровки,6, e-mail: mukha@gw.bsuir.unibel.by

В работе описывается разработанный авторами программный модуль для языка BORLAND PASCAL 7.0, предназначенный для работы с графическими файлами формата PCX.

Ключевые слова: анализ и обработка изображений, программные средства.

Задача математической обработки изображений является весьма актуальной, о чем свидетельствуют многочисленные публикации [1-4]. В работе описывается разработанный авторами программный модуль для работы с графическими файлами формата PCX. В сравнении с другими программами по обработке изображений данная разработка обладает следующими особенностями.

1. Предназначена для исследователей, занимающихся разработкой и исследованием *собственных* алгоритмов обработки случайных изображений и полей.
2. Рассчитана на пользователей, *не имеющих глубоких знаний в области форматов графических файлов*, перед которыми стоит задача с помощью общедоступных и простых средств (компилятор языка BORLAND PASCAL 7.0) обрабатывать вводимые изображения и сохранять полученные изображения.
3. Ориентирована на использование *многомерно-матричного подхода* [4] к обработке случайных изображений и полей, когда изображение представляется двумерной матрицей, а его ковариационная матрица – четырехмерной матрицей.

Разработаны и реализованы программно с использованием предложенного модуля новые алгоритмы зашумления изображений и пропускания изображений через систему линз. Модуль и программа могут быть предложены