УДК 002.53:681.3.06

С. П. ТИМОФЕЕВ, О. Я. ЛАЗАРЕВА

АССОЦИАТИВНЫЙ ДОСТУП К ТЕКСТОВЫМ БАЗАМ ДАННЫХ

Важнейшей задачей реализации интеллектуальных систем является организация ассоциативного доступа к данным во внешней памяти.

Разработчики интеллектуальных систем могут использовать для этой цели либо готовые пакеты прикладных программ (ППП), либо имеющиеся в ОС ассоциативные (индексные) методы доступа.

Благодаря ППП значительно уменьшается трудоемкость разработки, однако при достижении объемов базы данных (БД) 200 Мб и выше возникает проблема эффективности хранения данных во внешней памяти.

Опыт эксплуатации больших БД с текстовым наполнением во Всесоюзном научно-исследовательском институте технической информации, классификации и кодирования (ВНИИКИ) Госстандарта СССР показал недостаточную эффективность доступных ППП [1].

Предлагаемая нами структура позволяет эффективно использовать внешнюю память за счет особого способа сжатия ключей в индексе; уменьшить время обработки наиболее часто встречающихся запросов за счет помещения в элементы инвертированных списков поисковых элементов данных.

Опишем концепции программного комплекса «Специальный индексный метод доступа» (СИМД), реализующего инвертированные индексы для текстовых БД [2—4].

Концепции и возможности СИМД. С помощью СИМД реализуются инвертированные индексы — ссылочный аппарат инвертированных файлов. Данный метод ориентирован на использование в информационных системах с текстовым наполнением.

Инвертированным индексом в логическом плане служит множество инвертированных списков, которые состоят из ключа переменной длины и переменного числа элементов данных фиксированной длины, возможно, различной в отдельных списках; стандартной является реализация элементов как 3-байтовых адресов хранения записей БД, содержащих данное значение ключа.

Отличительная особенность организации хранения индексов — отказ от обязательной физической смежности блоков индекса. Это значительно упрощает управление внешней памятью и является перспективным, поскольку уже появились образцы накопителей на магнитный дисках (НМД) с неподвижными многодорожечными головками.

Реализация СИМД базируется на специальном базисном методе доступа (СБМД), разработанном в рамках сотрудничества ХГУ и ВНИИКИ. Язык реализации — ПЛ/1, оптимизирующий и Ассемблер ОС ЕС.

СБМД позволяет получить прямой доступ к записям переменной длины, поддерживая концепцию статического адреса хранения [1], и использует Базисный прямой метод доступа ОС ЕС.

СИМД функционирует в операционной системе ОС ЕС версии 6.1 и выше.

СИМД предоставляет следующие возможности: ключ переменной длины 1...255 байт; прямой (по ключу) и последовательный доступы, начиная с указанного ключа; обработку по неполному ключу; инвертированные списки с переменным числом (0...2**31—1) элементов заданной длины; размещение в инвертированных списках поисковых элементов данных, что ускоряет обработку наиболее часто встречающихся запросов. Данная воз-

можность отсутствует в других СУБД—наиболее широко используемой системе обработки инвертированных файлов; автономное создание и манипулирование линейными списками в БД, необязательно интерпретируемыми как инвертированные; одновременный доступ к индексу со стороны одного или нескольких процессов; параллельная обработка произвольного числа индексов (реентерабельные модули); динамическое восстановление внешней памяти, в связи с чем отпадает необходимость в регулярной реорганизации файлов.

СИМД состоит из двух компонентов — оперативной обработки и утилита базы данных.

СИМД имеет два уровня команд: 1 — позволяет раздельно манипулировать индексными деревьями и инвертированными списками. Пользователь может организовать нестандартную структуру хранения, однако ему должно быть известно детальное представление хранимых структур; 2 — дает возможность манипулировать инвертированными индексами как единым целым. Реализован один из возможных способов стыковки индексных деревьев и инвертированных списков. От пользователя скрыта структура хранения, однако теряется гибкость в организации хранения. Настоящая версия команд 2-го уровня поддерживает только 3-байтовые элементы списков.

Структура индекса

Концептуальная структура индекса описывает представление, реализуемое системой команд СИМД уровня 2. Ее знанием может ограничиться пользователь этого уровня. Пользователю уровня 1 приходится учитывать структуру хранения (физическую структуру).

Индекс является упорядоченным по возрастанию ключа множеством подындексов; дубликаты ключей не допускаются. Подындекс состоит из одного экземпляра ключа и соответствующего ему инвертированного списка (ИС). Ключ является строкой переменной длины 1...255. ИС может отсутствовать. Элементы ИС (ЭИС) имеют фиксированную длину в пределах списка, различную в разных списках. Внутри ЭИС выделяется «поле сортировки», длина и позиция которого фиксируются для каждого списка. ИС упорядочен по возрастанию поля сортировки, дубликаты не допускаются. Структурные параметры ИС (длина ЭИС, длина и позиция поля сортировки) не хранятся в индексе с целью экономии ВП. Они являются параметрами команд СИМД. Предполагается наличие внешних средств, позволяющих получить параметры ИС по значению ключа.

Структура поля ключа интерпретируется пользователем СИМД и может быть самой различной. Например, первые два байта отводятся под тип ключа, а остальные — под значение. Аналогично, интерпретация структуры ЭИС остается на усмотрение пользователя. Обычно поле сортировки содержит адрес хранения (АХР) записи.

Инвертированный индекс состоит из двух типов структур: индексного (поискового) дерева; линейных списков частного вида, которые реализуют инвертированные списки.

Типы структур, описанные выше, реализуются соответствующими модулями (процессорами абстрактных типов данных): модулем индексных деревьев и модулем списков.

Индексная запись может содержать произвольную информацию; управлять ею можно посредством команд первого уровня. Аналогично можно управлять структурой и содержимым элементов списка. Инвертированный индекс в целом как тип данных реализуется «интерфейсным модулем СИМД». Он организует следующий способ подсоединения списков к индексному дереву.

Поле данных индексной записи имеет вариантную структуру:

двоичные единицы во всех разрядах, если список пуст;

двоичные единицы в разрядах 1—16 и 3-байтовый элемент одноэлементного списка;

АХР многоэлементного списка: номер страницы, 4 байта, и номер сегмента, 1 байт.

Индексное дерево и инвертированные списки могут храниться в различных областях, однако команды уровня 2 ограничивают хранение в одной области.

Физическая структура индексных деревьев. Индексное дерево (ИД) целиком хранится в одной области, которая не может содержать других ИД. Оно сбалансировано с произвольным числом уровней, нумеруемых от листьев с нуля, и является вариантом B*-дерева.

Каждая индексная таблица занимает отдельную страницу области и в настоящей реализации является единственным «максимальным сегментом» на странице.

Номер корневой страницы указывается в префиксе области, в первой версии — в неиспользованном ранее элементе «приращение числа страниц».

Таблицы верхних уровней содержат переменное число упорядоченных по ключу пар: максимальный ключ в таблице лежащего ниже уровня; номер страницы, содержащей эту таблицу.

Терминальные таблицы отличаются тем, что вместо номеров страниц содержат поля данных индексных записей.

Все таблицы имеют префикс: тег, определяющий тип таблицы, 2 байта, и поле номера уровня, 2 байта.

Значения тега: ХК — корневая таблица, ХМ — промежуточная таблица, ХО — терминальная таблица.

Детальная структура таблиц скрыта в соответствующем модуле и не влияет на пользователя.

Физическая структура списков. Список хранится в виде однонаправленной цепочки сегментов переменной длины. Число сегментов произвольное. Все списки для отдельного индексного дерева хранятся в одной области. Область может содержать списки для различных деревьев, а также другую информацию. Сегмент списка состоит из 12-байтового префикса (вторичного пре-

фикса, первичный скрыт системой прямого метода доступа) и блока, представляющего собой последовательность элементов списка. Структура блока списка не является самоопределенной: длина элемента, длина и позиция поля сортировки являются параметрами команд обработки.

Команды манипулирования индексом. Команды обработки инвертированных индексов: открыть индекс; закрыть индекс; построить блок доступа; читать по ключу; читать следующий; отменить чтение; модифицировать индекс.

Команды обработки индексных деревьев: построить блок доступа к индексному дереву; уничтожить блок доступа к индексному дереву; получить индексную запись; добавить индексную запись; удалить индексную запись; модифицировать поле данных индексной записи.

Команды обработки списков: построить блок доступа к списку; уничтожить блок доступа к списку; создать список; найти элемент списка; получить следующий сегмент списка; добавить элемент; удалить элемент; уничтожить список.

Настоящая работа преследовала, прежде всего, исследовательские цели в области оптимизации структуры хранения.

Почти все модули написаны на ПЛ/1; ожидается значительное ускорение работы после перехода на Ассемблер, в первую очередь, после перепрограммирования модулей таблиц. Результаты работы могут быть полезны разработчикам ППП, осуществляющих поддержку больших баз данных с текстовым наполнением.

Список литературы: 1. *Ц51.804.006—001 Д110 ЕС ЭВМ.* Операционная система. Виртуальный метод доступа. Руководство программиста. 85 с. 2. *Дейт К.* Введение в системы баз данных. М., 1980. 314 с. 3. *Олле Т. В.* Предложения КОДАСИЛ по управлению базами данных. М., 1981. 286 с. 4. *Системы* управления базами данных для ЕС ЭВМ. Справочник. М., 1985. 250 с.

Поступила в редколлегию 04.04.88