

ВЫБОР ОПТИМАЛЬНОЙ СТРАТЕГИИ ВЫТЕСНЕНИЯ СТРАНИЦ В ПАРАЛЛЕЛЬНОЙ СУБД ОМЕГА ДЛЯ МУЛЬТИПРОЦЕССОРНОЙ СИСТЕМЫ МВС-100*

М.Л. Цымблер, Л.Б. Соколинский

Челябинск, Челябинский государственный университет

mzym@cgu.chel.su, sokolinsky@acm.org

Ключевые слова: параллельная система управления базами данных, управление буферным пулом, оптимальная стратегия вытеснения страниц.

1. Введение

Для работы параллельной системы управления базами данных (СУБД) необходим ряд общесистемных сервисов, обеспечивающих межпроцессорное взаимодействие, управление процессами, систему управления файлами и управление буферным пулом.

Существует два пути обеспечения данных сервисов. Первый – использовать соответствующие функции из контекста операционной системы, в которой работает СУБД. Однако, как правило, они недостаточно эффективны и не совсем пригодны для нужд СУБД [1]. Второй – реализовать эти функции как подсистемы СУБД. Современные СУБД обычно используют этот способ.

Современные СУБД обеспечивают буферный пул, и все операции ввода-вывода с диском используют этот пул. Операция чтения с диска возвращает данные непосредственно из буфера, если страница уже находится там. В противном случае необходимо выбрать некоторую страницу буфера, которая будет вытеснена на диск, чтобы освободить место для запрошенной страницы. Цель каждой стратегии вытеснения – минимизировать число неудач при поиске в буфере требуемой страницы.

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 00-07-90077).

В данной работе описывается подход, позволяющий осуществить выбор оптимальной стратегии вытеснения страниц. Этот подход применен при разработке системы управления файлами параллельной СУБД Омега [2] для мультипроцессорной вычислительной системы МВС-100 [3].

2. Выбор оптимальной стратегии вытеснения страниц

В настоящее время известно несколько стратегий вытеснения, например, FIFO, LIFO, LRU, CLOCK и другие [4]. Стратегии вытеснения обычно используют "возраст" страниц, находящихся в буфере или частоту их использования для предсказания будущего поведения операций с диском.

Хотя некоторые из них показывают в среднем удовлетворительную производительность, они могут быть наихудшими стратегиями в отдельных случаях [1].

Например, в соответствии со стратегией LRU (Least Recently Used) из буферного пула вытесняется страница, к которой дольше всего не было обращений. LRU является наихудшей стратегией при последовательном доступе к набору страниц, повторяющемуся в цикле.

Попытка улучшить данные алгоритмы приводит к новым, концептуально сложным алгоритмам, которые зависят от большого числа параметров. Нами предлагается другой подход, когда из фиксированного множества стратегий вытеснения *динамически* выбирается оптимальная для заданного набора страниц стратегия вытеснения.

Ключевую роль в данном подходе играет *избыточный индекс буферного пула*. Размер этого индекса определяется как $k * M$, где M – размер буферного пула в страницах, а целое $k \gg 1$. Избыточный индекс хранит историю загрузки страниц в буфер. Элемент избыточного индекса имеет статистические атрибуты для реализации общих стратегий вытеснения: счетчик обращений к странице, время последнего доступа к странице, очередность помещения в буфер и т.п. Эти атрибуты изменяются в ходе работы системы.

Менеджер буферного пула динамически анализирует эти атрибуты, чтобы найти циклы подкачки страниц. Если такой цикл имеет место, то Менеджер буферного пула выбирает стратегию вытеснения, оптимальную для данного набора страниц и вида доступа к нему.

Менеджер буферного пула вытесняет на диск некоторую страницу согласно выбранной стратегии вытеснения. Однако часто с точки зрения других компонент СУБД, находящихся на более высоких уровнях системной иерархии, эта страница должна оставаться в буфере либо вытесняться на диск не в том порядке, какой предписывает стратегия вытеснения.

Классическим примером, описанным в [1], является порядок вытеснения страниц для обеспечения восстановления базы данных после сбоя. Страницы, содержащие журнал транзакций, должны вытесняться на диск строго после страниц, которые претерпели изменения в ходе транзакций. Другой пример – выполнение запроса к отношению, которое имеет индекс в виде В-дерева и используется стратегия вытеснения FIFO. Тогда страница, содержащая корень этого дерева, будет циклически вытесняться на диск, что приведет к большим накладным расходам.

Для предотвращения подобных ситуаций в СУБД Омега используется техника вытеснения, основанная на концепции статических и динамических рейтингов страниц. Каждый элемент избыточного индекса страниц имеет *статический рейтинг*. Это целое число из диапазона [0;20]. Статический рейтинг назначается при помещении страницы в буфер. Статический рейтинг страницы остается постоянным все время нахождения страницы в буфере и теряется после вытеснения страницы из буфера.

Динамический рейтинг – это функция от вышеупомянутых статистических атрибутов страницы, возвращающая вещественное число из интервала [0;1]. Динамический рейтинг изменяется в процессе работы системы.

Суммарный рейтинг подсчитывается как сумма статического и динамического рейтингов. При необходимости освободить место в буфере вытесняется страница с минимальным суммарным рейтингом.

Механизм статических рейтингов обеспечивает системные приоритеты вытеснения страниц. Страницы, имеющие больший статическим рейтинг, будут оставаться в буфере независимо от выбранной стратегии вытеснения.

Механизм динамических рейтингов позволяет имитировать практически любую стратегию вытеснения страниц.

3. Заключение

В работе описан подход к выбору оптимальной стратегии вытеснения страниц в параллельной СУБД Омега.

Главная идея подхода – использование избыточного индекса буферного пула. Размер этого индекса определяется как $k*M$, где M – размер буферного пула в страницах, а целое $k \gg 1$. Избыточный индекс хранит историю загрузки страниц в буфер.

Анализ избыточного индекса позволяет динамически выбирать оптимальную стратегию вытеснения для заданного множества страниц.

Также предложена техника вытеснения страниц, основанная на концепции статических и динамических рейтингов страниц. Данная техника обеспечивает системные приоритеты вытеснения страниц и позволяет имитировать практически любую стратегию вытеснения.

Описанный подход реализован для мультипроцессорной системы МВС-100 на языке программирования Си. Данный подход ориентирован в основном на использование в рамках проекта Омега. Однако он может быть применен в аналогичных приложениях для МВС-100, требующих интенсивных дисковых обменов.

Литература

1. Stonebraker M. Operating System Support for Database Management // Communications of the ACM (CACM). July 1981. Vol. 24. No.7. P. 412-418.
2. Соколинский Л.Б., Цымблер М.Л. Принципы реализации системы управления файлами для МВС-100 // Вестник ЧелГУ. Серия "Математика и механика". 1999. No.2(5). С. 176-199.
3. Гольдштейн М.Л. Мультипроцессорная вычислительная система на базе транспьютерной идеологии // Алгоритмы и программные средства параллельных вычислений: [Сб. науч. тр.]. Екатеринбург: УрО РАН. 1995. С. 61-68.
4. Effelsberg W., Harder T. Principles of Database Buffer Management // ACM Trans. on Database Systems. Dec. 1984. Vol. 9. No.4. P. 560-595.