

УДК 681.3(07)

А.А. Брюховецкий, доцент, канд. техн. наук,**А.В. Скатков, профессор, д-р техн. наук***Севастопольский национальный технический университет**ул. Университетская, 33, г. Севастополь, Украина, 99053**E-mail: a.alexir@mail.ru***ОПТИМИЗАЦИЯ СТРУКТУРНЫХ РЕШЕНИЙ УПРАВЛЕНИЯ ПАМЯТЬЮ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИСКУССТВЕННЫХ ИММУННЫХ СИСТЕМ**

Рассматривается задача оптимального размещения отношений в буфере при выполнении реляционных запросов с использованием моделей искусственных иммунных систем (ИИС) как задача многомерной оптимизации. Представлена аналитическая модель и алгоритм решения задачи, базирующийся на операторах клонирования, мутации и негативной селекции.

Ключевые слова: *аффинность, антитело, процедура клонирования, мутация, селекция.*

В настоящее время возрос интерес к разработке ИИС для решения прикладных вычислительных задач, таких как распознавание образов, классификация, идентификация объектов и других. Решение многих из перечисленных задач сводится к решению задач многомерной оптимизации. Поэтому разработка методов решения подобных задач, базирующихся на моделях ИИС, является актуальной.

ИИС оперирует данными, которые описывают свойства отдельного антитела. Каждое антитело представляет собой одно решение поставленной задачи. В классическом варианте задачи оптимизации сводятся к поиску решений в многомерном пространстве с наперед заданной целевой функцией. В случае искусственной иммунной системы антитела представляют собой вектор переменных, при подстановке которых в целевую функцию достигается оптимальное значение. В качестве скалярной величины оценки, показывающей близость результата к оптимальному значению, в ИИС используется понятие аффинности антитела.

Целью настоящей работы является разработка алгоритма оптимизации выполнения запросов на основе ИИС. Метод нахождения оптимального решения базируется на использовании операторов клонального отбора, мутации и негативной селекции [1]. Каждое из антител представляет собой одно из возможных решений и оценивается значением среднего времени выполнения запросов. Наиболее подходящие антитела, имеющие минимальное время, получают возможность "воспроизводить" потомство с помощью операции клонирования. Это приводит к появлению новых антител, которые в процессе мутации улучшают своё соответствие целевой функции. Антитела с худшей аффинностью, имеющие большее время выполнения, замещаются случайно сгенерированными антителами, равномерно распределёнными по всей области определения целевой функции. Таким образом, происходит итеративный процесс воспроизведения новых популяций из лучших представителей предыдущего поколения. Каждое новое поколение содержит более высокое соотношение характеристик, которыми обладают лучшие члены предыдущих поколений.

В данной работе рассматривается метод последовательного сканирования (вложенных циклов), который базируется на сканировании таблиц, одна из которых носит название "внутренней", а вторая внешней. Минимальная выделенная память отводится под два блока: первый — для внутреннего блока, а другой — для внешнего. Внешнее отношение читается только один раз, а внутреннее будет читаться столько раз, сколько кортежей или блоков (страниц) во внешнем отношении. Чтобы обеспечить меньшее число обращений к диску, внешним выбирается отношение с меньшим числом блоков. Если внутреннее отношение может быть размещено в памяти, то блоки отношений читаются один раз. В связи с этим рассматриваются две стратегии управления памятью. Пусть m , n — число страниц двух отношений и $m < n$. Тогда в первой стратегии оба отношения загружаются полностью в буфер и количество обращений к диску будет $m + n$. Во второй — количество операций чтения будет $mn/q + m$, где q — часть отношения n , размещаемая в буфере. Приведем пример. Пусть задан запрос, в котором участвует три отношения R_1 , R_2 , R_3 , каждое из которых состоит из 10, 20 и 40 страниц с числом кортежей (C_i , $i = 1, 2, 3$) 50, 100 и 500 соответственно. Используется последовательный доступ к отношениям и коэффициент селективности равен 0,01. При связывании отношений в приведенном порядке, в случае когда ни одно из отношений не хранится в буфере, число страничных связываний будет: $3000 = 20 \cdot C_1 + 40 \cdot 0,01 \cdot C_1 \cdot C_2$. При другом порядке соединений R_3 , R_2 , R_1 — будет получено значение 15000 ($= 20 \cdot C_3 + 10 \cdot 0,01 \cdot C_2 \cdot C_3$). Число операций обращения к диску может быть уменьшено, если отношения R_2 , R_3 можно загрузить в буфер. При этом потребуется 61 страница буферной памяти ($= 20 + 40 + 1$). Наконец, если в буфере размещаются все страницы отношений R_1 , R_2 , то общее число страниц буфера составит 31. Поэтому на этапе оптимизации будут рассматриваться различные планы размещения страниц отношений в буфере, размер

котрого будет варьироваться с помощью параметров λ и β на этапах поиска локального и глобального минимума.

Алгоритм выбирает оптимальный метод соединения отношений и размещения этих отношений в памяти для всех запросов одновременно. Предлагаемый подход основан на концепции потребления буферной памяти. Потребляемая память (P) представляет собой произведение объема (F) на время (T): $P = F \cdot T$, которое эта память используется. Рассмотрим пример. Предположим, что запрос А выполняется 10 сек при 50-страничном распределении отношений в памяти буфера. Запрос В — 20 сек при 40-страничном распределении. Предположим, что дополнительные 100 страниц, выделенные под отношения, позволят снизить время ответа на 5 сек в каждом запросе. При одинаковом снижении времени выполнения буферные затраты сильно отличаются. Снижение времени обработки запроса В будет достигнуто за счет увеличения потребления памяти на 1300 страниц-сек ($= 15 \text{ сек} \times 140 \text{ страниц} - 20 \text{ сек} \times 40 \text{ страниц}$). В тоже время для запроса А увеличение составит только 250 страниц-сек. Даже если запрос В достигнет 10-секундного сокращения времени обработки на дополнительные 100 страницы распределения, рост потребления памяти будет 600 страниц-сек, который по-прежнему более чем в 2 раза превышает значение для запроса А.

Объем потребляемой памяти учитывается как при выполнении отдельного запроса, так и при совместном выполнении всех запросов. При этом лучший план доступа при одном распределении памяти не обязательно будет лучшим при другом распределении. Более того, можно выделить два крайних случая обработки. С одной стороны, можно выделить максимум памяти для выполнения отдельного запроса и иметь минимальное время его реализации, а с другой стороны, можно использовать минимальную память, но время выполнения тогда становится больше. В первом случае будет образовываться очередь на обработку поступающих запросов, так как ресурсы буфера оказываются заняты отдельными запросами, и, следовательно, среднее время обработки всех запросов будет увеличиваться. Во втором случае наблюдается противоположная ситуация. Поскольку ресурсы буфера при выполнении отдельного запроса используются не полностью, то появляется возможность совмещения обработки разных запросов. Считается, что если часть буфера освобождается при выполнении транзакции А, то она может быть занята следующей транзакцией В, при условии, что план размещения страниц транзакции В в буфере возможен. Таким образом, имеет место совмещение выполнения операций обращения к диску и операций процессора над оперативной памятью. Поэтому в этом случае среднее время обработки всех запросов будет уменьшаться. Очевидно, что при заданной интенсивности запросов, потребление памяти за единицу времени при обработке всех запросов не может быть больше, чем общий размер буфера. Таким образом, задачу распределения буферной памяти можно рассматривать как задачу многомерной оптимизации для достижения минимального времени обработки всех запросов при ограничении на общий объем буфера. Основное отличие предлагаемого метода решения от известного [2] состоит в том, что на этапе глобальной оптимизации используется модель искусственной иммунной системы, базирующейся на операторах клонирования, мутации и негативной селекции.

Решается задача минимизации среднего времени обработки реляционных запросов. Задано: множество поступающих запросов Q_i , $i = \overline{1, N}$, с известной интенсивностью λ_i и отношений R_j , $j = \overline{1, n_i}$, участвующих в каждом i -том запросе. Известно число страниц отношений $|R_j|$, размер буфера — B , время обращения к диску — T_{io} , параметры $0 < \alpha < 1$, $0 < \beta < 1$ — определяющие объем буфера на этапе размещения страниц отношения и на этапе совместной обработки запросов.

Для каждого запроса существует множество различных планов S_{ik} соединения отношений. Каждый план характеризуется специальным размещением отношений в буфере. Например, для двух связанных отношений $R_j \times R_k$ существует две последовательности связывания и для каждой из них четыре варианта размещения в буфере: оба отношения размещены в буфере; одно — в буфере, другое на диске; оба — на диске. Так как размер буфера B ограничен, то все отношения, участвующие в запросе, как правило, не могут быть загружены в буфер. Более того, отдельное отношение тоже не всегда может полностью размещено в буфере, так как его размер превышает B . Поэтому в общем случае в буфер может быть загружена только часть отношения (не все страницы). Обозначим F_{ik} количество страниц, размещенное в буфере при плане S_{ik} . Тогда для каждого Q_i имеет место ограничение $\sum_k F_{ik} \leq B$. Более того, часть буфера должна быть отведена на операции, которые выполняет процессор над отношениями, загруженными в буфер. Поэтому будем использовать параметр $0 < \alpha < 1$, а выделяемый размер буфера под загрузку отношений определится как αB . Основная задача сводится к выбору плана, который потребует минимальное время на обработку запросов. Обозначим D_{ik} число операций чтения отношений с диска при выбранном плане S_{ik} . Тогда функция затрат, учитывающая интенсивность запросов λ_i , может быть представлена так: $W = \sum_i \lambda_i \sum_k D_{ik}$. Принимая во внимание, что время операции обращения к диску T_{io} известно, тогда общее время на выполнение чтения D_{ik} страниц обозначим T_{ik} . Поэтому приведенное выше ограничение может быть представлено как $\sum_i \lambda_i \sum_k T_{ik} F_{ik} \leq B$. Принимая во внимание совмещение

операций чтения с диска и операций над памятью при совместной обработке запросов, размер буфера определится как βB , где $0 < \beta < 1$. Поэтому обоснованный выбор значений α , β позволит обеспечить минимальное время обработки всех запросов.

Методы аналитического моделирования вычислительных систем, базирующиеся на теории массового обслуживания, позволяют выявить существование задержек вычислительного процесса из-за ограниченности ресурсов [3]. Результаты макроанализа используются для оценки общей производительности системы, времени ответа и получения других характеристик. Если время обработки программы в отсутствие более приоритетных функций обозначить через T , то затраченное время $E = T/(1-p)$. Множитель $1/(1-p)$ называется коэффициентом замедления. Замедление произойдет при следующих условиях:

- ресурс характеризуется максимальной интенсивностью обращений (например, число заданий на выполнение программ в секунду), которая является жестко фиксированной характеристикой аппаратуры. Чем выше интенсивность обращений к устройству, тем более возрастает очередь к нему и, следовательно, увеличивается время ожидания обслуживания;
- ресурс, может использоваться более чем одним процессом одновременно, и существует по крайней мере один процесс, который может завладеть ресурсом раньше всех остальных, также претендующих на этот ресурс. Другой процесс при этом должен ждать освобождения ресурса.

В нашем случае для каждого запроса величина p определяется как $Tio \sum_i Dik \lambda_i$. Тогда среднее время T_i обработки запроса Q_i при выбранном плане Sik определится как $T_i = Dik Tio / (1-p)$, а среднее время T обработки всех запросов Q_i ($i = 1, N$) будет равно $\sum_i (\lambda_i / \lambda) T_i$, где $\lambda = \sum_i \lambda_i$. Уточненная оценка времени выполнения запроса будет включать затраты процессорного времени на выполнение операций в памяти, основными из которых являются сканирование и сравнение кортежей отношений.

Алгоритм решения задачи оптимизации включает два основных этапа. На первом этапе для каждого запроса Q_i формируется множество всевозможных планов соединения Rk . Затем для каждого соединения строятся планы возможных размещений отношений в буфере. Для каждого возможного размещения в буфере вычисляются следующие характеристики: Fik , Dik , Tik . Далее запускается процедура, которая с использованием несложных критериев исключает из рассмотрения планы с худшими значениями Tik и Fik для заданного Q_i . План i хуже плана j , если $P_i > P_j$ и $T_i \geq T_j$. Из двух планов лучше тот, который имеет меньшую загрузженность буфера и меньшее время выполнения запроса.

На втором этапе решается задача глобальной оптимизации на основе моделей искусственных иммунных систем. Цель — минимизация среднего времени выполнения всех запросов. Данная задача сводится к решению задачи многомерной оптимизации. На этом этапе алгоритм исследует возможность использования незанятого пространства буфера, куда могут быть помещены не все страницы отношений, а их часть — q . Таким образом, увеличивая количество загружаемых страниц в буфер, мы уменьшаем время выполнения отдельного запроса. Однако необходимо учитывать вышеприведенное ограничение на возможности буфера при обработке N запросов. Изменяемым параметром в процессе оптимизации является значение Fik , которое в свою очередь будет определяться величинами αB , βB . Для каждого измененного $Fik + \Delta F$ вычисляется новое значение T_i . На каждом шаге формирования популяции антител с помощью операторов мутации, клонирования и негативной селекции выбираются антитела с минимальным значением аффинности — T . Антитела с худшей аффинностью заменяются случайными новыми решениями из числа возможных, полученных на первом этапе алгоритма. Таким образом, алгоритм сочетает исследование локального и глобального экстремума.

Алгоритм включает следующую последовательность действий.

Этап 1. Формирование локальных планов размещения отношений в буфере.

1. Сгенерировать всевозможные соединения отношений R_j для каждого Q_i .
2. Сформировать для каждого Q_i при заданных $|R_j|$, α , B возможные размещения страниц Fij в буфере. Для каждой пары отношений Rk , Rs учитываются следующие варианты:
 - по одной странице от каждого отношения,
 - если возможно, в буфер загружается меньшее отношение,
 - если возможно, в буфер загружаются оба отношения, удовлетворяющие ограничению $\sum_j Fij \leq \alpha B$.
3. Для каждого j -ого размещения отношений определить значения Fij , Dij , Tij и $Pij = Fij * Tij$, где Pij — буферные затраты.
4. Исключить из рассмотрения, те i -тые планы размещения, для которых: $(P_i > P_j)$ и $T_i \geq T_j$, т.е. требующие большие затраты буферной памяти.

Этап 2. Глобальная оптимизация. Задано: $\alpha_n < \alpha \leq \alpha_k$, $\Delta \alpha$, $\beta_n < \beta \leq \beta_k$, $\Delta \beta$, λ_i и параметры, вычисленные в п.3 этапа 1. P_l , $l = 1, W$, W — число популяций; $t = 1, M$, M — размер популяции.

1. Выбираем случайный план j для каждого Q_i , удовлетворяющий условию:

$$\sum_i \lambda_i \sum_j T_{ij} F_{ij} \leq \beta_B.$$

Если условие не выполняется, то $\beta = \beta + \Delta\beta \leq \beta_k$, да $\rightarrow$ п.1, иначе $\rightarrow$ п.2.

2. Если $\beta > \beta_k$, то выбрать новый случайный j -тый план размещения, вернуться на предыдущий шаг п.1 для Q_i с текущим значением β . Иначе, получено t -ое решение со средним временем T_i , значениями α_i, β_i . Если $t < M$, то $t = t + 1 \rightarrow$ п.1, иначе $\rightarrow$ п.3.
3. Сформирована популяция P_l . Упорядочить антитела популяции P_l по значению T_i в возрастающем порядке. Отобрать для клонирования r антител с самой низкой аффинностью.
4. Получить популяцию P_c измененных клонов с помощью механизма мутации для значений α_i, β_i с шагом $\Delta\alpha/2$ и $\Delta\beta/2$ соответственно. Вычислить аффинность каждого клона.
5. Объединить исходную и сформированную популяцию клонов: $P_l = P_l \cup P_c$. Проверить условия окончания алгоритма: $l < W$, где $W = (\alpha_k - \alpha_n) / \Delta\alpha$. Если условие выполнено, то решение найдено. Иначе переход на п.6.
6. Выполнить операцию негативной селекции. Удалить из популяции $n = 2r$ худших антител, у которых значение T_i максимальное. Заменить их r новыми сгенерированными антителами. $t = 1, i = 1, j = 1, \alpha = \alpha + \Delta\alpha, \beta = \beta_n, l = l + 1 \rightarrow$ на п.1.

Таким образом, алгоритм сочетает локальный и глобальный поиск на основе лучших антител, исследуя пространство при различных вариантах значений буферной памяти и разнообразных планах размещения отношений в буфере.

Перспективой дальнейших исследований является применение разработанного алгоритма для различных методов доступа, что и составит предмет будущих исследований.

Бібліографічний список використовуваної літератури

1. Искусственные иммунные системы и их применение / Под ред. Д. Дасгупты. Пер. с англ. — М.: Физматлит, 2006. — 344 с.
2. Cornel D.W. Integration of buffer management and query optimization in RDBS / D.W. Cornel, P.S. Yu // «VLDB: Proc. of the 15 International Conference – 1989», NY, 17-21 april 1989. — NY, 1989. — P. 247–253.
3. Феррари Д. Оценка производительности вычислительных систем / Д. Феррари. — М.: Мир, 1981. — 573 с.

Поступила в редакцию 26.11.2010 г.

Брюховецький О.О., Скатков О.В. Оптимізація структурних рішень управління пам'яттю з використанням методів штучних імунних систем

Розглядається задача оптимального розміщення відносин в буфері при виконанні реляційних запитів з використанням моделей штучних імунних систем як завдання багатовимірної оптимізації. Представлена аналітична модель та алгоритм розв'язання задачі, що базується на операторах клонування, мутації і негативної селекції.

Ключові слова: афінність, антитіло, процедура клонування, мутація, селекція.

Bruhovetskiy A.A., Skatkov A.V. Structural decisions optimization of memory control using the method of artificial immune systems

The problem of optimal placement of relations in the buffer when performing relational queries using models of artificial immune systems as the problem of multivariate optimization is considered. The analytical model and solution algorithm based on the operators of cloning, mutation, and negative selection are presented.

Keywords: affinity, antibody, procedure cloning, mutation, selection..