УДК 681.142.2+519.2

Е.А. АКСЕНОВА, А.В. СОКОЛОВ

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ДИНАМИЧЕСКИМИ СТРУКТУРАМИ ДАННЫХ

Abstract: *This paper concerns issues related to building mathematical models and optimal algorithms of LIFO-stacks, FIFO and priority queues control. These models were constructed as 1, 2 and 3-dimensional random walks.*
Key words: *dynamic data structures, stack, FIFO-queue, priority queue, Markov chain, random walks.*

Анотація: *У роботі запропоновані математичні моделі та оптимальні алгоритми керування такими динамічними структурами даних, як LIFO-стеки, FIFO-черги, пріоритетні черги. Як моделі розглянуті одновимірні, двовимірні та тривимірні випадкові блукання.*
Ключові слова: *динамічні структури даних, стек, FIFO-черга, пріоритетна черга, ланцюг Маркова, випадкові блукання.*

Аннотация: *В работе предложены математические модели и оптимальные алгоритмы управления такими динамическими структурами данных, как LIFO-стеки, FIFO-очереди, приоритетные очереди. В качестве моделей рассмотрены одномерные, двумерные и трехмерные случайные блуждания.*
Ключевые слова: *динамические структуры данных, стек, FIFO-очередь, приоритетная очередь, цепь Маркова, случайные блуждания.*

1. Вступление

Такие базовые структуры данных, как LIFO-стеки и FIFO-очереди, используются при разработке прикладного и системного обеспечения широкого класса задач. В приложениях, наиболее критичных к использованию ресурсов, алгоритмы работы с ними реализуются аппаратно. Например, при разработке стековых компьютеров [1] (или RISC-процессоров) для управления стеками в двухуровневой памяти (регистры – оперативная память) использовались специальные алгоритмы, а не универсальные алгоритмы кэш-памяти. Еще один пример – алгоритмы работы с очередями, необходимые при разработке встроенных операционных систем, управляющих потоками пакетов в Internet, таких, например, как Cisco IOS [2], где требования на время обработки пакетов маршрутизатором очень жесткие. Механизм страничной виртуальной памяти здесь не используется, и вся работа происходит в нескольких пулах оперативной памяти.

FIFO-очереди используются в компьютерных сетях, операционных системах, графических системах, устройствах промышленной автоматики. Ряд фирм выпускает микросхемы, реализующие работу с несколькими параллельными FIFO-очередями на одном кристалле. Число очередей и длина каждой очереди устанавливаются программным путем на этапе инициализации устройства.

В некоторых приложениях, например, в системах моделирования, в сетевых маршрутизаторах, в операционных системах для организации работы с несколькими процессами применяются очереди с приоритетами, в которых основными операциями являются операция включения элемента с заданным приоритетом (ключом) и операция получения и удаления элемента с наивысшим приоритетом. Для представления приоритетных очередей можно применять упорядоченные и неупорядоченные списки и массивы или специальные структуры данных, в которых организован поиск по ключу (приоритету).

В данной работе рассмотрим представление очереди по приоритетам в виде нескольких FIFO-очередей. Преимуществом такого способа представления является то, что при этом не требуется хранить в памяти ключи (приоритеты), так как приоритет определяется просто номером FIFO-очереди. Но возникает проблема оптимального разбиения памяти между FIFO-очередями.

Здесь предложены математические модели и оптимальные алгоритмы управления такими динамическими структурами данных, как LIFO-стеки, FIFO-очереди, приоритетные очереди. Рассматривались последовательный, связный и страничный способы представления двух, трех и четырех стеков и двух и трех очередей в памяти одного уровня, а также одного, двух и трех стеков в двухуровневой памяти.

2. Оптимальное управление одним стеком в двухуровневой памяти

Пусть мы работаем со стеком, значительно превосходящим выделенный для него участок быстрой памяти размера m. В этом случае в быстрой памяти всегда хранится вершина стека, а остальная часть находится в памяти второго уровня. Ставится задача определить, как производить перераспределение стека между уровнями памяти, чтобы минимизировать средние временные затраты. Под перераспределением будем понимать перепись x_0+1 верхних элементов стека с памяти второго уровня в быструю память, если быстрая память стала пустой (теперь, после исключения элемента, в быстрой памяти будет x_0 элементов), или выталкивание $m-x_0+1$ элементов на второй уровень и, возможно, перепись x_0-1 верхних элементов стека в начало быстрой памяти, если произошло ее переполнение (теперь, после включения нового элемента в стек, в быстрой памяти станет x_0 элементов).

В качестве математической модели процесса рассмотрим случайное блуждание на состояниях $x_0-1,0,1,...,x,...,m-1,m,m+1$, где x – число верхних элементов стека в быстрой памяти. Будем считать, что вероятность перехода из состояния x в $x+1$ (увеличение длины стека на единицу в результате включения нового элемента) равна p, а из x в $x-1$ (уменьшение длины стека на единицу в результате исключения одного элемента) равна $q=1-p$. Блуждание начинается в состоянии x_0, а перераспределение происходит в состояниях -1 и $m+1$. Попадание в состояние -1 означает, что вершина стека стала пустой и нужно обратиться к верхнему элементу стека в памяти второго уровня, а попадание в состояние $m+1$ означает, что после заполнения быстрой памяти мы должны включить еще один элемент в стек. Нашей целью является определение такого состояния x_0 $(1\le x_0\le m)$, в которое происходит переход после перераспределения памяти, чтобы оно доставляло оптимальное значение соответствующей функции критерию. Если не учитывать стоимость обменов, то нашей целью является минимизация среднего числа обращений к памяти второго уровня. Следовательно, необходимо найти такое x_0, которое максимизирует математическое ожидание времени работы до перераспределения памяти при условии, что процесс начинается в этом состоянии.

Заметим, что наша модель сводится к задаче о разорении игрока [3], если общий капитал игроков равен $m+2$, а капитал первого игрока равен x_0+1. Тогда поглощение в состоянии $m+1$ эквивалентно выигрышу первого игрока, а в -1 – его разорению.

Показано, что оптимальное значение вершины стека выражается по следующей формуле:

$$x_0 = \frac{m}{2},\ \textit{если}\ p = q = 1/2,$$

$$x_0 = \log_{\frac{q}{p}} \frac{((q/p)^{m+2} - 1)p}{(m+2)q\ln(q/p)},\ \textit{если}\ p \neq q.$$

Также рассмотрена задача определения оптимального значения x_0 с учетом затрат на перераспределение стека [4]. Некоторые специалисты [5] считают, что модель классического случайного блуждания недостаточно точно описывает поведение стеков, а более адекватной была бы модель, в которой вероятность следующей операции со стеком зависит от того, какая операция была предыдущей.

Рассмотрим состояние процесса после очередного переполнения или опустошения вершины стека (при этом предполагаем, что в памяти второго уровня остается невершинная часть стека, то есть это опустошение не последнее), когда в быстрой памяти остается x_0 верхних элементов стека так, что процесс начинается из состояния x_0.

Пусть в начальный момент времени вершина стека с некоторыми вероятностями может перейти вправо или влево. Затем вероятности переходов изменяются следующим образом: если мы перешли в состояние $x_0 - 1$, то возвращаемся в состояние x_0 с вероятностью p_1 и переходим в состояние $x_0 - 2$ с вероятностью $1 - p_1$, а если перешли в состояние $x_0 + 1$, то возвращаемся в состояние x_0 с вероятностью p_2 и переходим в состояние $x_0 + 2$ с вероятностью $1 - p_2$. То есть параметры p_1 и p_2 определяют вероятность возвращения в то состояние, из которого мы перешли в заданное, в зависимости от того, произошло исключение или включение элемента.

Заметим, что в случае $p_1 = p_2 = p$ вероятность вернуться не зависит от операции, которая произошла со стеком, а в случае $p_1 = p_2 = 1/2$ имеем классическое симметричное случайное блуждание. Задача оптимального управления вершиной стека в рамках этой модели решалась в [6].

3. Оптимальное управление двумя стеками

Пусть в памяти объема m расположены два стека, растущие навстречу друг другу. В этом случае место их встречи можно рассматривать как случайную величину. Пусть известно, что на каждом шаге с вероятностью p может произойти исключение элемента из одного из стеков и с вероятностью $1 - p$ может произойти включение информации в один из стеков. Обозначим через $M(m, p)$ математическое ожидание случайной величины $\max(k_1, k_2)$, где k_1 *и* k_2 – длины стеков при встрече. Д. Кнут [7] поставил задачу разработать математическую модель этого процесса и установить вид функции $M(m, p)$.

В работах [8–11] была построена математическая модель процесса в виде двумерного случайного блуждания в треугольной области с двумя отражающими экранами и одним поглощающим экраном. Рассмотрим обобщение этой задачи.

Пусть в памяти объема m расположены два стека, растущие навстречу друг другу и превосходящие объем быстрой памяти. В этом случае в быстрой памяти хранятся только вершины обоих стеков, а остальные части – в памяти второго уровня. Если же вершина одного из стеков стала пустой или стеки заполнили всю быструю память, происходит перераспределение стеков и обмен с памятью второго уровня. Наша задача – минимизировать среднее число перераспределений памяти, то есть максимизировать среднее время работы до перераспределения памяти. Будем считать, что при перераспределении мы всегда приходим к некоторому заданному состоянию памяти, после чего начинается следующий этап работы.

Обозначим через x_1 текущую длину первого стека, а через x_2 – второго. Тогда в качестве математической модели процесса имеем случайное блуждание по целочисленной решетке в области $x_1 \geq 0,\ x_2 \geq 0,\ x_1 + x_2 \leq m$ с вероятностями из точки с координатами (x_1, x_2): q_1 - сдвинуться влево, p_1 – вправо, q_2 – вниз, p_2 – вверх. Цель – определить такое состояние $(x_1^{\,0}, x_2^{\,0})$, при котором среднее время блуждания до поглощения на прямой $x_1 + x_2 = m+1$ и на прямых $x_1 = -1,\ x_2 = -1$ было максимальным при условии, что процесс начинается в состоянии $(x_1^{\,0}, x_2^{\,0})$. Также рассматривалась задача оптимального управления двумя параллельными стеками в двухуровневой памяти [12]. Предполагалось, что вершины двух стеков растут навстречу друг другу в быстрой памяти, к которой разрешен доступ нескольких параллельных процессоров.

4. Оптимальное управление тремя стеками

Пусть в памяти объема m единиц расположены три стека. Двум стекам, растущим навстречу друг другу, отведено s единиц памяти, а третьему стеку $m-s$ единиц. Обозначим через x_1, x_2, x_3 текущие длины стеков. В этом случае в качестве модели имеем трехмерное случайное блуждание в треугольной призме с тремя отражающими экранами $x_1 = -1, x_2 = -1, x_3 = -1$ и двумя поглощающими экранами $x_1 + x_2 = s+1, x_3 = m-s+1.$ Задача определения оптимального начального распределения памяти заключается в определении значения s и установлении нумерации стеков (т.е. определении стека, размещаемого отдельно от пары других) так, чтобы максимизировать среднее время блуждания внутри призмы до поглощения на ее границе при условии, что процесс блуждания начинается в начале координат.

Построены математические модели метода работы со стеками в случае, если два стека растут навстречу друг другу с концов памяти, а третий стек растет навстречу им обоим одновременно, начиная с некоторой средней точки в случае связного и страничного представления трех стеков, а также математическая модель и алгоритм оптимального управления тремя стеками в двухуровневой памяти в случае, если два стека растут навстречу друг другу, а третий расположен отдельно.

5. Оптимальное управление FIFO-очередями

Предложены математические модели и алгоритмы оптимального управления двумя и тремя последовательными циклическими FIFO-очередями в памяти одного уровня, когда в качестве критерия оптимальности выбрано среднее время до переполнения памяти, и если целью оптимизации является минимизация доли потерянных элементов (пакетов) на бесконечном времени, а также математические модели, предназначенные для анализа связного и страничного методов управления двумя и тремя FIFO-очередями в памяти одного уровня [13].

Нами был предложен новый метод реализации работы с несколькими последовательными циклическими FIFO-очередями, когда очереди двигаются по кругу друг за другом. В этом методе память заранее не делится между очередями, а очереди двигаются по кругу друг за другом, начиная с некоторого начального места в памяти. В этом случае, если одна из очередей стала пустой, то вторая может занимать всю доступную область памяти, пока не догонит сама себя. Если же оживет вторая очередь, то ее можно пустить с середины свободного участка или с некоторой оптимальной точки, которая зависит от вероятностных характеристик очередей.

Заметим, что два стека, направленные навстречу друг другу, это заведомо лучший метод в сравнении с раздельным хранением стеков. В случае же очередей, например, Д. Кнут считал [7], что эффективно работать в общей памяти с двумя последовательными очередями нельзя. Но, видимо, он не рассматривал возможности того, что две очереди могут двигаться по кругу одна за другой. Анализ этого метода представления для двух очередей производился с помощью разработанной имитационной модели.

6. Выводы

В статье предложена математическая модель метода представления приоритетной очереди в виде нескольких FIFO-очередей. Решается задача оптимального управления приоритетными очередями для случая двух очередей.

Математические модели представлены в виде случайных блужданий по целочисленным решеткам в различных областях. Разработаны алгоритмы нумерации состояний и алгоритмы построения соответствующих поглощающих или регулярных цепей Маркова и вычисления значения критерия оптимальности для связного и страничного представлений и нахождения оптимального разбиения памяти для последовательных представлений. Доказаны теоремы, устанавливающие вид необходимых матриц. Обсуждаются результаты численных экспериментов с разработанными программами. Работа поддержана грантом РФФИ № 06-01-00303.

СПИСОК ЛИТЕРАТУРЫ

1. Koopman P. Stack Computers. – Ellis Horwood, 1989. – 232 p.
URL:http://www.ece.cmu.edu/~koopman/stack_computers/index.html.
2. Боллапрагада В. и др. Структура операционной системы Cisco IOS / В. Боллапрагада, К. Мэрфи, Р. Уайт. – Вильямс, 2002. – 197 с.
3. Феллер В. Введение в теорию вероятностей и ее приложения. – М.: Мир, 1964. – 498 р.
4. Соколов А.В. Математические модели и алгоритмы оптимального управления динамическими структурами данных. – Петрозаводск: Изд-во ПГУ, 2002. – 215 с.
5. Ertl Anton M. Stack caching for interpreters // SIGPLAN'95 Conf. Program. Lang. Des. and Implem. (PLDI). – 1995. – June. – P. 315 – 327.

6. Аксенова Е.А. и др. Исследование немарковской модели управления стеком в двухуровневой памяти / Е.А. Аксенова, А.А. Лазутина, А.В. Соколов // Программирование. – 2004. – № 1. – С. 1 – 10.
7. Кнут Д. Искусство программирования для ЭВМ. – М.: Мир, 1976. – Т. 1. – 712 с.
8. Соколов А.В. О распределении памяти для двух стеков // Автоматизация эксперимента и обработки данных. – Петрозаводск: Изд-во Карельского филиала АН СССР, 1980. – С. 65 – 71.
9. Yao A.C. An analysis of a memory allocation scheme for implementating stacks // SIAM J. Computing. – 1981. – N 10. – P. 398 – 403.
10. Flajolet P. The evolution of two stacks in bounded space and random walks in a triangle // Lec. Notes Computer Sci. – 1986. – Vol. 223. – P. 325 – 340.
11. Random walks, heat equation and distributed Algorithms / G. Louchard, R. Schott, M. Tolley et al. // J. Comput. Appl. Math. – 1994. – N 53. – P. 243 – 274.
12. Аксенова Е.А., Соколов А.В. Оптимальное управление двумя параллельными стеками // Дискретная математика. – 2007. – №1. – С. 67 – 75.
13. Аксенова Е.А., Соколов А.В. Некоторые задачи оптимального управления FIFO-очередями // Труды Второй Всероссийской научной конференции "Методы и средства обработки информации". – М.: Изд. отдел ВМК МГУ им. М.В. Ломоносова, 2005. – С. 318 – 322.

Стаття надійшла до редакції 23.09.2008