Анализ некоторых методов реализации приоритетной очереди[1]

Е. А. Аксенова, канд. ф.-м. н., А. В. Соколов, д. ф.-м. н.[2]
ИПМИ КарНЦ РАН, Петрозаводск
(aksenova, avs)@krc.karelia.ru

В работе анализируются некоторые методы реализации приоритетной очереди. Допускаются операции “вставить”, “удалить наибольший”, “найти наибольший”. Предполагаются известными вероятности выполнения этих операций. Рассмотрены способы представления очереди в виде массива и в виде нескольких последовательных FIFO-очередей. Предложены математические модели этих методов и на их базе методы сравниваются с точки зрения среднего времени до переполнения памяти.

Ключевые слова: приоритетная очередь, случайные блуждания, цепи Маркова.

1. Введение

Во многих приложениях требуется структура данных, в которой основными операциями являются операции вставки нового элемента и удаления элемента с наибольшим значением ключа (приоритета). Такую структуру данных называют приоритетной очередью [1] (очередью с приоритетами [2,4], очередью по приоритетам [3]). Основными методами реализации таких структур данных являются упорядоченные и неупорядоченные списки и массивы, пирамиды [1,4](двоичные кучи [2], сортирующие деревья [3]), различные варианты бинарных деревьев поиска и другие методы [5]. Все эти методы имеют разную стоимость основных операций и требуют разных объемов памяти. В этой работе рассмотрены способы представления приоритетной очереди в виде массива и в виде нескольких последовательных FIFO-очередей.

[1]Работа выполнена при финансовой поддержке РФФИ (грант №06-01-00303).

[2]©Е. А. Аксенова, А. В . Соколов, 2008

Пусть N – число элементов в очереди, K – число приоритетов и допускаются операции:

- “вставить” – в наихудшем случае ее стоимость для упорядоченного массива и списка $\mathcal{O}(N)$, а для неупорядоченных и при представлении в виде нескольких FIFO-очередей – $\mathcal{O}(1)$;
- “удалить наибольший” – в наихудшем случае ее стоимость для упорядоченного массива и списка $\mathcal{O}(1)$, а для неупорядоченных – $\mathcal{O}(N)$, при представлении в виде нескольких FIFO-очередей – $\mathcal{O}(K)$;
- “найти наибольший” – в наихудшем случае ее стоимость для упорядоченного массива и списка $\mathcal{O}(1)$, а для неупорядоченных – $\mathcal{O}(N)$, при представлении в виде нескольких FIFO-очередей – $\mathcal{O}(K)$.

Эффективность метода представления приоритетной очереди зависит не только от набора операций и их стоимостей, но и от частот выполнения операций.

В данной работе предполагаем, что на каждом шаге дискретного времени операция “вставить” может произойти с вероятностью p, операция “удалить наибольший” – с вероятностью q, и операция “найти наибольший” – с вероятностью r, где $p + q + r = 1$. В начале работы очередь пустая. Также предполагаем, что объем памяти равен n условных единиц, все элементы, включаемые в очередь, имеют размер a, ключи (приоритеты) имеют размер b, указатели на начало и конец очереди имеют размер c.

Целью исследования является сравнение изучаемых способов представления с точки зрения среднего времени работы до переполнения памяти, т. е. изучение динамики изменения длины очереди. Допускаем возможность переполнения очереди. Заметим, что в большинстве работ такие ситуации не исследуются [1-4], хотя, например, в маршрутизаторах потеря элементов очереди (пакетов) при переполнении является обычной ситуацией [6].

2. Представление в виде массива

В качестве математической модели для данного представления рассмотрим одномерное случайное блуждание в интервале длины

$m = \frac{n}{a+b}$ (считаем, что здесь n кратно $a+b$, т. к. надо хранить и ключи, и элементы). На каждом шаге с вероятностью p в массив добавляется новый элемент, с вероятностью q исключается элемент, и с вероятностью r длина массива на данном шаге не изменяется.

Максимальное количество элементов в очереди равно m. Предположим, что к началу работы в очереди находится z элементов. Необходимо найти среднее время работы с очередью до переполнения с условием, что при попытке исключения элемента из пустого массива ничего не происходит, т. е. до следующего включения он остается пустой с вероятностью $r+q$. Эту задачу можно рассматривать как случайное блуждание на прямой. Блуждание начинается в точке z. С вероятностью p частица может сдвинуться на единицу вправо, с вероятностью q – влево, с вероятностью r – остаться на месте. Точки -1 и $m+1$ – это отражающий и поглощающий экраны соответственно. Продолжительность такого блуждания имеет конечное математическое ожидание и удовлетворяет следующему разностному уравнению [7]:

$$T(z) = p \cdot T(z+1) + q \cdot T(z-1) + r \cdot T(z) + 1, \quad -1 < z < m+1,$$

с граничными условиями $T(-1) = T(0), \quad T(m+1) = 0$.

Можно показать [7, 8], что решение этого уравнения имеет вид:

$$T(z) = \frac{z-m-1}{q-p} + \frac{q}{(q-p)^2}\left(\left(\frac{q}{p}\right)^{m+1} - \left(\frac{q}{p}\right)^{z}\right), \ p \neq q;$$

$$T(z) = -\frac{1}{2q}\left(z^2 + z - (m+1)(m+2)\right), \ p = q.$$

Заметим, что при $z = 0$ (в нашем случае это так, т. к. работа начинается с пустой очереди) формулы для среднего времени принимают вид:

$$T(0) = \frac{-m-1}{q-p} + \frac{q}{(q-p)^2}\left(\left(\frac{q}{p}\right)^{m+1} - 1\right), \ p \neq q;$$

$$T(0) = \frac{1}{2q}(m+1)(m+2), \ p = q.$$

Так как в нашем случае $m = \frac{n}{a+b}$, то формулы примут вид:

$$T(0) = \frac{-\frac{n}{a+b} - 1}{q-p} + \frac{q}{(q-p)^2}\left(\left(\frac{q}{p}\right)^{\frac{n}{a+b}+1} - 1\right),\ p \neq q;$$

$$T(0) = \frac{1}{2q}\left(\frac{n}{a+b} + 1\right)\left(\frac{n}{a+b} + 2\right),\ p = q.$$

3. Представление в виде k последовательных FIFO-очередей

При таком представлении необходимо хранить $k-1$ указателей на границы очередей и $2k$ указателей на начало и конец каждой очереди, но не надо хранить приоритеты. Приоритет определятся по номеру очереди. Поэтому для хранения элементов остается $n - (k-1)c - 2kc = n - (3k-1)c$ единиц памяти. Учитывая размер элемента, максимальное число элементов в очереди будет $m = \frac{n-(3k-1)c}{a}$.

Сравним полученное значение с максимальным количеством элементов в очереди, представленной в виде массива. Для этого надо решить неравенство $\frac{n-(3k-1)c}{a} > \frac{n}{a+b}$. В случае $a = b = c = 1$, это неравенство верно для $n > 6k - 2$.

Число приоритетов на практике конечно много меньше, чем размер памяти, выделенный для очереди. Например, в операционной системе Cisco IOS [6] число приоритетов $k = 4$. В данной работе предложена модель в случае двух приоритетов.

Рассмотрим очередь с двумя приоритетами, расположенную в области памяти размера m единиц. Такую приоритетную очередь представим в виде двух FIFO-очередей. Первой FIFO-очереди присвоим приоритет 1, второй FIFO-очереди – приоритет 2. Наивысший приоритет 2. Будем считать, что время дискретно. В каждый момент времени могут произойти следующие операции:

- включение элемента с первым приоритетом с вероятностью p_1 (включение элемента в первую FIFO-очередь);
- включение элемента со вторым приоритетом с вероятностью p_2 (включение элемента во вторую FIFO-очередь);

- исключение элемента из очереди с вероятностью q;
- очередь не изменяет своей длины с вероятностью r (только чтение или отсутствие операции),

где $p_1+p_2+q+r=1$, $p_1+p_2=p$. Исключение элемента из очереди происходит по наивысшему приоритету. Это означает, что пока вторая FIFO-очередь не пуста, с вероятностью q исключение элементов происходит из этой очереди. Как только вторая FIFO-очередь станет пустой, с вероятностью q исключение элементов будет происходить из первой FIFO-очереди. При исключении элемента из пустой очереди не происходит завершение работы.

Необходимо определить, как распределить память между FIFO-очередями. В качестве критерия оптимальности рассмотрим максимальное среднее время работы с приоритетной очередью до переполнения выделенного объема памяти.

В последовательном способе каждой FIFO-очереди выделим некоторое количество единиц памяти из данных m единиц. Пусть s – количество единиц памяти, выделенных первой FIFO-очереди, тогда $(m-s)$ – количество единиц памяти, выделенных второй FIFO-очереди (рис. 1).

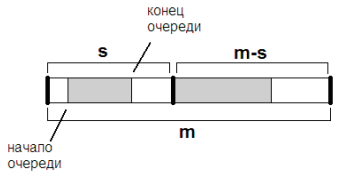

Рис. 1: Последовательное представление.

Обозначим текущие длины FIFO-очередей x_1 и x_2. В качестве математической модели процесса работы с приоритетной очередью рассмотрим случайное блуждание по целочисленной решетке в двухмерном пространстве $0 \le x_1 < s+1$, $0 \le x_2 < m-s+1$ (рис. 2), где $x_1 = s+1$, $x_2 = m-s+1$ – поглощающие экраны, попадание на которые характеризуется как переполнение одной из FIFO-очередей, $(-1,0)$ и $(0,-1)$ – отражающие точки, т. к. только в состоянии $x_1 = x_2 = 0$ возможно исключение из пустой очереди.

Исключение элементов из первой FIFO-очереди может происходить только при $x_2 = 0$.

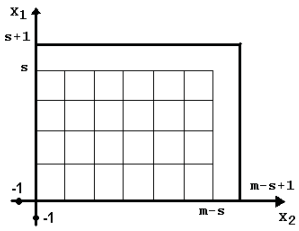

Рис. 2: Область блуждания при последовательном представлении.

Блуждание начинается в точке $x_1 = x_2 = 0$. Необходимо определить такое значение $0 \le s \le m$, чтобы время блуждания до попадания на поглощающие экраны было максимально.

Если в произвольный момент времени процесс блуждания находится в состоянии (x_1, x_2), то в следующий момент времени он переходит в состояние (x_1', x_2'), которое определяется правилами:

$$(x_1, x_2) \xrightarrow{p_1} (x_1', x_2') = \{(x_1 + 1, x_2), \quad 0 \le x_1 \le s, \ 0 \le x_2 \le m - s;$$

$$(x_1, x_2) \xrightarrow{p_2} (x_1', x_2') = \{(x_1, x_2 + 1), \quad 0 \le x_1 \le s, \ 0 \le x_2 \le m - s;$$

$$(x_1, x_2) \xrightarrow{q} (x_1', x_2') = \begin{cases} (x_1 - 1, x_2), & 0 < x_1 \le s, \ x_2 = 0; \\ (x_1, x_2 - 1), & 0 \le x_1 \le s, \ 0 < x_2 \le m - s; \\ (x_1, x_2), & x_1 = 0, \ x_2 = 0; \end{cases}$$

$$(x_1, x_2) \xrightarrow{r} (x_1', x_2') = \{(x_1, x_2), \quad 0 \le x_1 \le s, \ 0 \le x_2 \le m - s.$$

Случайное блуждание будем рассматривать как конечную однородную поглощающую цепь Маркова [9]. Пронумеруем невозвратные состояния цепи так, как показано графически (рис. 3). Количество невозвратных состояний в такой цепи равно $(s+1)(m-s+1)$. Например, для $m = 6, s = 4$ количество невозвратных состояний равно 15 (рис. 3).

Рассмотрим матрицу Q вероятностей переходов из невозвратных состояний в невозвратные. Для размера памяти m и заданного s размерность матрицы Q будет $(s+1)(m-s+1) \times (s+1)(m-s+1)$.

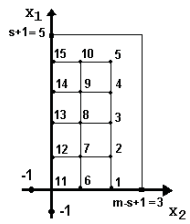

Рис. 3: Область блуждания и нумерация невозвратных состояний при $m = 6, s = 4$.

Теорема 1. *Матрица Q при заданной нумерации, размере памяти m и величине $s > 0$ имеет следующую структуру:*

$$Q = \begin{pmatrix} A & B & & \\ C & \dots & \dots & \\ & \dots & A & B \\ & & C & A' \end{pmatrix},$$

где A, A', B, C – матрицы размера $(s+1) \times (s+1)$ имеют вид:

$$A = \begin{pmatrix} r & p_1 & & \\ & \dots & \dots & \\ & & r & p_1 \\ & & & r \end{pmatrix}, \quad A' = \begin{pmatrix} r+q & p_1 & & \\ q & r & \dots & \\ & \dots & \dots & p_1 \\ & & q & r \end{pmatrix},$$

$$B = \begin{pmatrix} q & & & \\ & q & & \\ & & \dots & \\ & & & q \end{pmatrix}, \qquad C = \begin{pmatrix} p_2 & & & \\ & p_2 & & \\ & & \dots & \\ & & & p_2 \end{pmatrix}.$$

Доказательство. Индукция проводится по размеру памяти m.

1) Пусть размер памяти $m = 2, s = 1$ (Рис. 4). Размерность матрицы Q будет 4×4, и матрица имеет вид

$$Q = \left(\begin{array}{cc|cc} r & p_1 & q & 0 \\ 0 & r & 0 & q \\ \hline p_2 & 0 & r+q & p_1 \\ 0 & p_2 & q & r \end{array}\right).$$

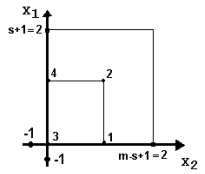

Рис. 4: Область блуждания и нумерация невозвратных состояний при $m = 2, s = 1$

2) Предположим, что для размера памяти $m = d - 1$ и заданного значения $s > 0$ выполняется утверждение теоремы. Количество невозвратных состояний в цепи равно $(s+1)(d-1-s+1) = (s+1)(d-s)$. Матрица Q имеет структуру:

$$Q = \begin{pmatrix} A & B & & \\ C & \dots & \dots & \\ & \dots & A & B \\ & & C & A' \end{pmatrix}.$$

3) Проверим, что при размере памяти $m = d$ выполняется утверждение теоремы. Если добавить еще одну единицу памяти, то при фиксированном s поглощающая граница области блуждания $x_2 = d - s$ при $m = d - 1$ сдвинется на единицу вправо и будет $x_2 = d + 1$. Тогда получим $s + 1$ новых невозвратных состояний. Согласно введенной нумерации эти состояния будут иметь младшие номера. Количество невозвратных состояний будет $(s+1)(d-s) + (s+1) = (s+1)(d-s+1)$. Тогда в матрицу Q при $m = d - 1$ добавятся подматрицы: A, B, C. Получили матрицу Q размерности $(s+1)(d-s+1) \times (s+1)(d-s+1)$:

$$Q = \left(\begin{array}{c|cccc} A & B & & & \\ \hline C & A & B & & \\ & C & \dots & \dots & \\ & & \dots & A & B \\ & & & C & A' \end{array}\right).$$

Теорема доказана.

Подматрицы матрицы Q характеризуют определенные операции над очередями. Подматрица A характеризует операции, при которых не изменяется длина очереди (например, чтение), и операции включения элементов с приоритетом 1, B – операции исключе-

ния элементов с приоритетом 2, C – операции включения элементов с приоритетом 2, A' – операции, при которых не изменяется длина очереди и операции включения и исключения элементов с приоритетом 1.

4. Решение задачи и результаты численных экспериментов

Для решения задачи в случае представления в виде FIFO-очередей вычислим фундаментальную матрицу $N = (I-Q)^{-1}$ [9] для всевозможных значений s. Для вычисления среднего времени блуждания просуммируем элементы матрицы N в строке, которая соответствует начальному состоянию $x_1 = x_2 = 0$. Согласно введенной нумерации это будет строка с номером $(s+1)(m-s+1)-s$. Затем сравним полученное время для разных значений s и выберем максимальное. Соответствующее максимальному времени значение s будет оптимальным разбиением памяти для данного представления.

Для реализации задачи разработаны программы для ЭВМ, которые вычисляют среднее время работы с приоритетной очередью для рассмотренных методов представления.

В табл. 1 представлены результаты вычислений. В первых четырех столбцах находятся вероятностные характеристики приоритетной очереди, в столбце с названием n содержится размер выделенной памяти. Как упоминалось ранее, $p_1 + p_2 = p$.

Столбец с названием A_1 соответствует представлению в виде FIFO-очередей, для размера памяти n, столбец с названием A_1' – представлению в виде FIFO-очередей для размера памяти $n-2$, т. е. когда учитываются две единицы памяти при реализации двух циклических FIFO-очередей [1]. В столбце с названием s находится оптимальное количество единиц памяти, выделенных первой FIFO-очереди, в столбце с названием T – среднее время работы. В столбце с названием A_2 находится среднее время работы T для представления очереди в виде массива.

Вычисления проведены с учетом размеров данных, указателей и приоритетов, т. е. $m = n-5$ для представления в виде FIFO-очередей, $m = \frac{n}{2}$ для представления в виде массива.

Таблица 1: Среднее время работы с приоритетной очередью.

r	p_1	p_2	q	n	A_1		A_1'		A_2
					s	T	s	T	T
0.25	0.25	0.25	0.25	26	14	66.347	12	59.497	52
0.1	0.4	0.4	0.1	26	11	26.625	10	24.095	19.796
0	0.1	0.6	0.3	26	6	45.418	5	40.961	33.125
0.2	0.6	0.1	0.1	26	18	31.925	16	28.994	23.056
0	0.7	0.3	0	26	14	19.356	13	17.474	14
0	0.3	0.7	0	26	7	19.356	6	17.474	14
0.1	0.3	0.2	0.4	26	16	137.056	14	120.21	101.759
0.1	0.2	0.3	0.4	26	14	128.789	12	114.055	101.759
0.5	0.2	0.2	0.1	26	12	60.001	11	54.288	45.556
0.34	0.21	0.12	0.33	26	16	493.502	14	400.894	318.182
0.0	0.25	0.25	0.50	26	15	301.001	13	248.297	210

Из табл. 1 видим, что с точки зрения среднего времени работы в случае $a = b = c = 1$ и $k = 2$ представление приоритетной очереди в виде FIFO-очередей лучше, чем представление в виде массива. Это может быть связано с тем, что в первом случае нет затрат памяти на хранение приоритетов.

В дальнейшем представляется интересным рассмотреть случай других значений a, b, c, k, других методов представления приоритетной очереди и других критериев оптимальности.

Список литературы

[1] *Кнут Д.* Искусство программирования для ЭВМ. Т. 3. – М.: Вильямс. 2001.

[2] *Кормен Е., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. – М.:МЦНМО. 2000.

[3] *Седжвик Р.* Фундаментальные алгоритмы на C++. Анализ /Структуры данных /Сортировка/ Поиск: Пер. с англ./ Р. Седжвик. – К.: Издательство “ДиаСофт”. 2001.

[4] *Гудрич М., Тамассия Р.* Структуры данных и алгоритмы в Java. – “Новое знание”. 2003.

[5] *Rongren R. , Ayani R.* A comparative study of parallel and sequential priority queue algorithms // ACM Trans. on Modeling and Computer Simulation. Vol. 7. No 2. April 1997. PP. 157–209.

[6] *Боллапаргада В., Мэрфи К., Расс У.* Структура операционной системы Cisco IOS. – М.: Вильямс. 2002. С. 163–189.

[7] *Феллер В.* Введение в теорию вероятностей и ее приложения. – М.: Мир. 1964.

[8] *Тарасюк А. В.* Оптимальная реализация N FIFO-очередей в памяти одного уровня в случае одновременного выполнения операций // Системы управления и информационные технологии. №1(31). 2008. С. 75–78.

[9] *Кемени Дж., Снелл Дж.* Конечные цепи Маркова. – М.: Наука. 1970.