

Анализ некоторых методов представления двухприоритетной очереди¹

Д. В. Зайцева,

Петрозаводский университет,

А. В. Соколов, д. ф.-м. н.²

ИПМИ КарНЦ РАН, Петрозаводск

zaiceva@cs.karelia.ru, avs@krc.karelia.ru

В работе рассматриваются методы представления очереди с двумя приоритетами в памяти одного уровня в виде двух последовательных циклических и в виде двух связанных FIFO-очереди.

Возможны операции “вставить”, “удалить элемент с наибольшим приоритетом” и “найти элемент”. Предложены алгоритмы нумерации состояний, построения соответствующих регулярных цепей Маркова и вычисления средней доли потерянных элементов при бесконечном времени работы для последовательного и связанного способов представления FIFO-очереди, и нахождения оптимального разбиения памяти для последовательного способа с точки зрения минимизации средней доли потерянных элементов.

Ключевые слова : очереди с приоритетами, FIFO-очереди, случайное блуждание, цепи Маркова, динамические структуры данных.

1. Введение

Во многих приложениях используется структура данных, в которой основными операциями являются вставка элемента и удаление элемента с наибольшим приоритетом. Такую структуру данных называют приоритетной очередью. Основными методами реализации такой структуры данных являются упорядоченные и неупорядоченные списки, массивы, бинарные деревья, пирамиды [1–4].

В [5, 6] рассматривалась задача выбора оптимального представления приоритетной очереди с точки зрения максимизации среднего времени работы до переполнения памяти. Здесь мы рассмотрим задачу выбора оптимального представления приоритетной очереди с точки зрения минимизации среднего числа потерянных элементов. Будет рассмотрен случай двух приоритетов и представления

¹Работа выполнена при финансовой поддержке РФФИ (грант №09-01-00330).

²©Д. В. Зайцева, А. В. Соколов, 2010

очереди в виде двух последовательных и двух связанных FIFO-очереди.

2. Постановка задачи

Рассмотрим очередь с двумя приоритетами, расположенную в памяти размера m . Будем считать, что время дискретно, и в каждый момент времени происходит одна из следующих операций:

- Вставить элемент с приоритетом i с вероятностью p_i ($1 \leq i \leq 2$);
- Удалить элемент с наибольшим приоритетом с вероятностью q ;
- Найти элемент с наибольшим приоритетом с вероятностью r .

$$p_1 + p_2 + q + r = 1.$$

Наивысший приоритет равен 2, наименьший — 1. Работа начинается с пустой очереди. Предположим также, что не происходит завершения работы в случае исключения элемента из пустой структуры данных.

Требуется найти среднюю долю потерянных элементов при бесконечном времени работы для представления приоритетной очереди в виде двух последовательных циклических и двух связанных FIFO-очереди, а в случае представления в виде двух последовательных FIFO-очереди найти оптимальное разбиение памяти между очередями.

3. Связное представление

Определим схему переходов между состояниями. Заметим, что в отличие от последовательного представления, в связном представлении часть памяти тратится на связи (будем считать, что половина — наихудший вариант), зато каждая из очередей может занять всю оставшуюся часть памяти. Для построения модели введем границу области переходов l (см. рис. 1). Ее можно определить через уравнение прямой: $y = \frac{m}{2} - x$.

Введем специальную нумерацию состояний: нумерация начинается с нуля и идет снизу-вверх и слева-направо вдоль прямых, параллельных l . Состояния, соответствующие сбросу хвоста (за пределами l) нумеруем в последнюю очередь (таким же образом). Пример нумерации см. на рис. 1 для случая $m = 8$.

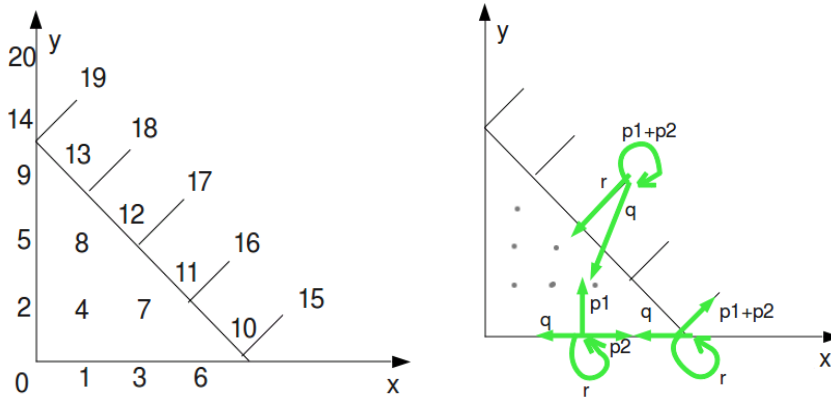


Рис. 1: Граница области переходов l и схема переходов для случая $m = 8$.

Для этой же размерности на рис. 1 приведена схема переходов.

После этого построим регулярную матрицу переходов P с учетом схемы переходов (для произвольных m) размерностью

$$\sum_{i=0}^{\frac{m}{2}} \frac{m}{2} + 1 - i.$$

При приведенной выше нумерации матрица переходов имеет определенную структуру и ее удобней представить в следующем виде:

$$P = \begin{pmatrix} Q & Q' \\ & Q'' \end{pmatrix}. \quad (1)$$

Подматрица Q описывает блуждание в области, ограниченной прямыми $x = 0, y = 0, y = \frac{m}{2} - x$ (далее l), подматрица Q' описывает вероятности перехода на прямую $y = \frac{m}{2} - x + 1$ (далее $l1$), т. е. первое включение элементов в переполненные очереди, и матрица Q'' описывает поведение процесса в случаях, когда уже было переполнение.

Утверждение: Матрица Q при указанной нумерации и раз-

матрица Q имеет следующий вид:

$$Q_{h \times h} = \begin{pmatrix} A_{k \times k} & B_{k \times (k+1)} & O & \dots & O \\ C_{k \times (k-1)} & A_{k \times k} & B_{k \times (k+1)} & O & \dots \\ O & C_{k \times (k-1)} & \ddots & \ddots & \dots \\ \vdots & \vdots & \ddots & \ddots & \dots \\ O & O & \dots & C_{k \times (k-1)} & A_{k \times k} \end{pmatrix},$$

где $h = \sum_{i=0}^{\frac{m}{2}} \frac{m}{2} + 1 - i$, k — переменная величина, равная количеству элементов на каждом уровне нумерации, а уровни нумерации определяются прямыми параллельными l ($y = \frac{m}{2} - x - i$, где $i \leq \frac{m}{2}$), и указанные подматрицы имеют следующую структуру:

$$A_{k \times k} = \begin{pmatrix} r & 0 & \dots & 0 \\ 0 & r & 0 & \dots \\ 0 & 0 & \ddots & \dots \\ \vdots & \vdots & \dots & \dots \\ 0 & 0 & \dots & r \end{pmatrix},$$

(но у первой матрицы первый элемент равен $r + q$)

$$B_{k \times (k+1)} = \begin{pmatrix} p2 & p1 & 0 & \dots & 0 \\ 0 & p2 & p1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \dots \\ 0 & 0 & \dots & p2 & p1 \end{pmatrix},$$

$$C_{k \times (k-1)} = \begin{pmatrix} q & 0 & \dots & 0 \\ q & 0 & \dots & 0 \\ 0 & q & \dots & 0 \\ \vdots & \vdots & \ddots & \dots \\ 0 & 0 & \dots & q \end{pmatrix}.$$

Доказательство.

1). База индукции. Пусть $m = 2$. Тогда размерность $Q = 3 \times 3$.

$$Q_{3 \times 3} = \begin{pmatrix} r + q & p2 & p1 \\ q & r & 0 \\ q & 0 & r \end{pmatrix}$$

2). Индуктивное предположение. Предположим, что для размера памяти m выполняется утверждение.

3). Проверим, что при размере памяти $m + 2$ утверждение выполняется (в силу предположения о том, что половина памяти расходуется на связи, рассматриваем только четные размерности). Мы добавили еще одну единицу памяти. Таким образом мы добавили еще один уровень нумерации (в нем на один элемент больше, чем в предыдущем). Следовательно добавилось $k + 1$ новых состояний (k — количество элементов в последнем уровне). Таким образом количество подматриц в Q возросло на 1 — добавилась “кайма” вокруг уже существующей матрицы

$$\left(\begin{array}{c|c} Q & O \\ \hline O & C_{(k+1) \times k} \end{array} \middle| \begin{array}{c} O \\ B_{(k+1) \times (k+2)} \\ A_{(k+1) \times (k+1)} \end{array} \right),$$

но к уже существующим матрицам ничего не добавилось. Рассмотрим, как выглядят добавленные матрицы A, B, C : Для размера памяти $m + 2$ матрица A будет иметь следующий вид:

$$A_{(k+1) \times (k+1)} = \left(\begin{array}{c|c} A_{k \times k} & 0 \\ \hline 0 & r \end{array} \right),$$

т. е. добавилась последняя строка и последний столбец, а общая структура не изменилась.

Аналогично можно доказать, что не изменится и общая структура остальных подматриц.

Теперь рассмотрим подматрицу $Q'_{(l-k) \times (k)}$ (k см. выше). В ней ненулевыми элементами будут только состояния, соответствующие переходам из $x \in l$ в $x \in l1$ (заполняются $p2 + p1$ диагональные элементы последнего блока размером $k \times k$) (см. пример выше). И, наконец, рассмотрим матрицу Q'' : ее k строк дублируют объединенные последние k строк матриц Q, Q' . Эти элементы соответствуют переходам с прямой $l1$ (конец док-ва).

Построив матрицу переходов, найдем вектор-строку $\alpha = (a_1, a_2, \dots, a_n)$, которая удовлетворяет уравнению

$$\alpha = \alpha \cdot P.$$

Найденная вектор-строка α и будет состоять из долей времени a_j нахождения в каждом состоянии j (согласно закону больших чисел из [7]).

4. Последовательное представление

Введем специальную нумерацию состояний: нумерация начинается с нуля и идет снизу-вверх и слева-направо для всех состояний, где $0 \leq x < s + 1$ и $0 \leq y < m - s + 1$. Состояния, соответствующие сбросу хвоста $x = s + 1$ или $y = m - s + 1$ нумеруем в последнюю очередь (также сначала снизу-вверх, потом слева-направо) Пример нумерации для случая $m = 4, s = 2$ и схема переходов приведены на рис. 2.

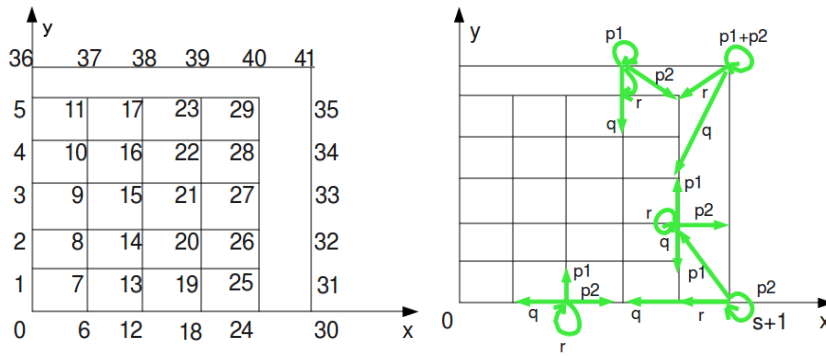


Рис. 2: Нумерации и схема переходов для случая $m = 4, s = 2$.

Теперь построим регулярную матрицу переходов P с учетом схемы переходов (для произвольных m и s) размерностью $((m - s + 1) * (s + 1) + m + 2) * ((m - s + 1)(s + 1) + m + 2)$. При приведенной выше нумерации эта матрица переходов имеет структуру вида (1), где подматрица Q описывает блуждание в области, ограниченной прямыми $x = 0, y = 0, x = s, y = m - s$, подматрица Q' описывает вероятности перехода на прямые $x = s + 1, y = m - s + 1$, т. е. первое включение элементов в переполненные очереди, и матрица Q'' описывает поведение процесса в случаях, когда переполнены одна или обе очереди.

Утверждение: Матрица Q при указанной нумерации, размере

памяти m и заданной величине s имеет следующий вид:

$$Q_{l \times l} = \begin{pmatrix} A_{k \times k} & B_{k \times k} & O & \dots & O \\ C_{k \times k} & A_{k \times k} & B_{k \times k} & O & \dots \\ O & C_{k \times k} & \ddots & \ddots & \dots \\ \vdots & \vdots & \ddots & \ddots & \dots \\ O & O & \dots & C_{k \times k} & A_{k \times k} \end{pmatrix},$$

где $l = (s+1) \times (m-s+1)$, $k = (m-s+1)$, а указанные подматрицы имеют следующую структуру:

$$A_{k \times k} = \begin{pmatrix} r & p1 & 0 & \dots & 0 \\ q & r & p1 & 0 & \dots \\ 0 & q & \ddots & \ddots & \dots \\ \vdots & \vdots & \ddots & \ddots & \dots \\ 0 & 0 & \dots & q & r \end{pmatrix},$$

(но у первой матрицы первый элемент равен $r+q$)

$$B_{k \times k} = \begin{pmatrix} p2 & 0 & \dots & 0 \\ 0 & p2 & 0 & \dots \\ \vdots & \vdots & \ddots & \dots \\ 0 & 0 & \dots & p2 \end{pmatrix},$$

$$C_{k \times k} = \begin{pmatrix} q & 0 & \dots \\ 0 & \dots & 0 \\ \vdots & \ddots & \dots \\ 0 & \dots & 0 \end{pmatrix}.$$

Доказательство.

1). База индукции. Пусть $m = 2$, $s = 1$. Тогда размерность Q — 4×4 и

$$Q_{4 \times 4} = \begin{pmatrix} r+q & p1 & p2 & 0 \\ q & r & p2 & 0 \\ q & 0 & r & p1 \\ 0 & 0 & q & r \end{pmatrix}.$$

2). Индуктивное предположение. Предположим, что для размера памяти m выполняется утверждение.

3). Проверим, что при размере памяти $m+1$ утверждение выполняется. Мы добавили еще одну единицу памяти. Пусть она попала в первую очередь, тогда область случайных блужданий сдвинулась на единицу по оси ОУ. Следовательно добавилось $s + 1$ новых состояний. Таким образом количество подматриц в Q не изменилось (т. к. не менялось s), но к каждой матрице добавилось по одной строке и столбцу (размерность Q возросла на $s + 1$). Рассмотрим как изменятся матрицы A, B, C : Для размера памяти матрица A будет иметь следующий вид:

$$A_{(k+1) \times (k+1)} = \left(\begin{array}{c|c} A_{k \times k} & O \\ \hline O & \begin{array}{c} p1 \\ r \end{array} \end{array} \right),$$

т. е. добавилась последняя строка и последний столбец, а общая структура не изменилась.

Аналогично можно доказать, что не изменится и общая структура остальных подматриц. Если мы добавим единицу памяти во вторую очередь, тогда область блужданий увеличится на единицу по оси ОХ. Следовательно, в матрицу добавятся $m - s + 1$ новых состояний. Таким образом размер матрицы Q возрастет на $m - s + 1$ единиц, т. е. добавится по каждому экземпляру подматриц A, B, C , при этом их структура останется как при размере памяти равному m (из-за того, что меняется только s , а $m - s + 1$ не изменится).

Теперь рассмотрим подматрицу $Q'_{l \times (m+2)}$. В ней ненулевыми элементами будут только состояния, соответствующие переходам из $x = s$ в $x = s + 1$ (заполняется $p2$) и из $y = m - s$ в $y = m - s + 1$ (заполняется $p1$) (см. пример выше). Первые описанные переходы заполняют Q' вероятностями $p2$ элементы по диагонали, начиная с $s * (m - s + 1)$ строки и нулевого столбца (всего $m - s + 1$ элементов). Вторые переходы отображаются в матрице переходов как элементы, расположенные на $i \cdot (m - s + 1)$ строках ($i = 0 \dots s + 1$), в столбцах, начиная с $m - s + 1$ (всего $s + 1$ элементов).

И, наконец, рассмотрим матрицу Q'' : первые $m - s + 1$ строк дублируют объединенные последние $m - s + 1$ строк матриц Q, Q' . Оставшиеся $s + 2$ элемента соответствуют переходам на прямую $y = m - s + 1$. Первые $s + 1$ блока заполняются следующим образом: на местах $(i + (m - s + 1), (m - s + 1) \cdot i - 2)$, где $i = 0 \dots s$ и еще $((m + 2), (m - s + 1) \cdot 2 - 2)$ заполняются q , аналогично заполняются

элементы с r , только сдвигаются относительно q на 1 состояние вправо. Элементы $(i + (m - s + 1) - 2, (m - s + 1) \cdot i - 1$, где $i = 2 \dots s + 1$, заполняются $p2$, последний же блок размером $(s + 2) \times (s + 2)$ заполняется $p1$ по диагонали, а к элементам $((m + 2), (m + 2))$ и $((m + 1), (m + 2))$ добавляется $p2$ (конец док-ва).

Построив матрицу P , найдем вектор-строку $\alpha = (a_1, a_2, \dots, a_n)$, которая удовлетворяет уравнению

$$\alpha = \alpha \cdot P.$$

Найденная вектор-строка α и будет состоять из долей времени a_j нахождения в каждом состоянии j (согласно закону больших чисел из [7]). Остается минимизировать сумму a_j , соответствующих состояниям сброса хвоста:

$$\sum_{j=m \cdot s - s^2}^{(m-s+1) \cdot (s+1)} a_j \rightarrow \min$$

(всего $m + 1$ слагаемое), т. е. необходимо найти такое s при заданном m и вероятностях $p1, p2, q, r$, чтобы представленная сумма была минимальна.

5. Сравнение различных способов представлений

Разработана программа для построения матрицы P и решения уравнение $\alpha = \alpha \cdot P$ приближенным методом с использованием системы Pardiso из библиотеки mkl [8, 9]. В табл. 1 приведены результаты вычислений. В строке q находится доля вероятности исключения, в строках $pl_i, i = 10, 60, 100$ — оптимальное разбиение памяти для последовательного представления, в строках $p_i, s_i, i = 10, 60, 100$ — доли потерянных пакетов при размерах памяти соответственно $m = 10, 60, 100$.

Из полученных результатов можно сделать следующие выводы:

1. Для последовательного представления двухприоритетной очереди при малых q лучшим разделением памяти является то, при котором размер второй памяти s — намного больше размера первой памяти (с большим приоритетом). Это можно объяснить тем,

Таблица 1: Результаты вычислений.

q	0.1	0.2	0.4	0.6	0.8	0.9
pl_{10}	8	8	6	5	5	5
pl_{60}	52	39	52	38	24	20
pl_{100}	92	46	91	47	17	15
p_{10}	0.5	0.33335	$4.458_{10}-2$	$4.976_{10}-4$	$1.13_{10}-6$	$10_{10}-8$
p_{60}	0.5	0.33(3)	$7.391_{10}-3$	0.0	0.0	0.0
p_{100}	0.5	0.33(3)	$4.225_{10}-3$	0.0	0.0	0.0
s_{10}	0.5	0.3342	0.0666	$2.589_{10}-3$	$1.429_{10}-5$	$1.4_{10}-7$
s_{60}	0.5	0.33(3)	0.0129	0.0	0.0	0.0
s_{100}	0.5	0.33(3)	$1.593_{10}-3$	0.0	0.0	0.0

что при малых q редко происходят исключения из очередей, и если первая очередь будет большой, то из второй почти ничего не будет исключаться, что приведет к большим потерям (например, $q = 0.35$: наилучшее $s = 94$ при $m = 100$). С последующим ростом q растет и размер первой очереди, а следовательно уменьшается размер второй.

2. В обоих случаях при достаточно больших размерах памяти ($m \geq 100$) потери стремятся к нулю уже при $q \geq 0.45$.

3. При сравнении потерь данных для связного и последовательного представления очередей можно заметить, что они очень близки (при $m = 100$ отличаются только для $q \geq 0.4$), но все-таки доля потерянных пакетов для оптимально распределенного последовательного представления несколько лучше.

В связи с этим можно сделать вывод, что при оптимальном распределении памяти в последовательном представлении двухприоритетной очереди потери данных будут меньше при описанных выше вероятностных характеристиках, кроме того не будет расходоваться память на хранение связей между элементами.

4. Первые два описанных пункта совпадают с ранее полученными результатами работы программы, основанной на построении имитационной модели для последовательного представления очереди.

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. / Роберт Седжвик. — К.: Издательство “ДиаСофт”. 2001.— 688с.
- [4] *Боллапаргада В., Мэрфи К., Расс У.* Структура операционной системы Cisco IOS. — М.: Вильямс. 2002. С. 163–189.
- [5] *Аксенова Е.А., Соколов А.В.* Анализ некоторых методов реализации приоритетной очереди // Стохастическая оптимизация в информатике. Вып. 4. 2008. С. 61–71.
- [6] *Драц А.В., Соколов А.В.* Оптимальное размещение в памяти одного уровня п стеков и/или очередей // Стохастическая оптимизация в информатике. Вып.4. 2008. С. 115–121.
- [7] *Кемени Дж., Снелл Дж.* Конечные цепи Маркова. — М.: Наука. 1970.
- [8] *Соловьев С.А., Пудов С.Г.* Out-of-core PARDISO — параллельный прямой решатель Intel MKL больших разреженных систем линейных уравнений. Электронный ресурс. Режим доступа: http://www.sbras.ru/ws/show_abstract.dhtml?ru+199+15550
- [9] Intel Math Kernel Library Quick Reference/Intel Corporation. Электронный ресурс. Режим доступа: www.intel.com/software/products/mkl/docs/mklqref/index.htm