Оптимальное управление очередью с динамически изменяющимися приоритетами[1]

А. В. Тарасюк
Ин-т прикл. математ. исслед. КарНЦ РАН, Петрозаводск[2]

Рассмотрена задача управления двухприоритетной очередью с изменяющимся старшим приоритетом. Очередь реализована в виде двух FIFO очередей. Предложена модель, описывающая поведение очередей, и разработан алгоритм их оптимальной реализации в зависимости от входных данных.

1. Введение

Приоритетная очередь дает определенным заявкам, поступающим в нее, безусловный приоритет над другими заявками. Все заявки классифицируются на основании настраиваемых правил и помещаются в очередь. В первую очередь обслуживание получают заявки с высшим приоритетом, т. е. в очереди с N приоритетами (1 — высший, N — низший) заявка i-го приоритета получает обслуживание только в том случае, если в очереди нет заявок приоритетов $1, 2, \ldots, (i-1)$. Метод приоритетной очереди достаточно широко используется, например, как один из аппаратно-независимых методов управления перегрузками — одной из основных технологий QoS. Так, в операционной системе Cisco IOS реализация приоритетной очереди содержит четыре различные подочереди: high, medium, normal и low priority [1].

Существуют различные способы реализации очередей с приоритетами, одним из часто используемых является представление N-приоритетной очереди в виде N FIFO очередей. В данной работе предложена подобная реализация двухприоритетной очереди в конечной области памяти. Будем считать, что в очереди поступают заявки двух типов — первого и второго соответственно, и для каждой из FIFO очередей выделяется свой буфер памяти. В работе [2] рассматривалась задача управления двумя последовательными циклическими FIFO очередями, когда в случае переполнения одной или обеих очередей (т. е. когда очередь исчерпает всю предоставленную ей память) все входящие заявки терялись. Одна-

[1]Работа велась при финансовой поддержке РФФИ (грант №06-01-00303).
[2]©А. В. Тарасюк, 2006

ко, возможна ситуация, когда потеря заявки с высшим приоритетом, является более критичной, чем потеря заявки с низшим. Тогда, в случае переполнения первой очереди, все входящие заявки можно помещать во вторую очередь, т. е. понижать приоритет заявок первого типа. Следовательно, потеря заявок первого типа возможна лишь в том случае, когда заполнена вся память предоставленная для работы. Такой способ реализации приоритетной очереди мы назвали очередью с динамически изменяющимися приоритетами. Естественно, что необходимо разделить память между первой и второй очередями таким образом, чтобы минимизировать потери заявок обоих типов.

2. Постановка задачи

Для удобства, обозначим нашу приоритетную очередь Q, а очереди, посредством которых она реализована Q_1 и Q_2. Пусть $m + 2 < \infty$ размер памяти предоставленный нам для работы, тогда максимальная длина очереди Q равна m, это значение на две единицы меньше, так как в реализации циклической FIFO очереди один элемент памяти всегда остается свободным для того, чтобы отличить пустую очередь от переполненной.[3] Пусть размер буфера памяти для заявок первого типа равен s, тогда для заявок второго типа он $m - s$.

Будем считать, что время дискретно и в каждый момент времени происходит либо поступление в очередь заявки первого или второго типа, с вероятностью p_1 и p_2 соответственно, либо исключение заявки с вероятностью q, ясно, что $p_1 + p_2 + q = 1$. Предполагается, что в очередях хранятся данные фиксированного размера.

Обозначим через x и y текущие длины очередей Q_1 и Q_2, включение и исключение заявок это некий случайный процесс, который естественно рассматривать как двумерное случайное блуждание по целочисленной решетке, которое начинается в точке $(0, 0)$. Определим область блуждания. Очевидно, что блуждание описывающее работу с очередями до переполнения, происходит в прямоугольной области $0 \le x \le s$, $0 \le y \le m - s$, а для учета потерь при переполнении добавим точки $(x, m - s + 1)$, где $0 \le x \le s + 1$ (см. Рис. 1).

Теперь определим само блуждание: если в в произвольный момент времени процесс находится в состоянии (x, y), то в следующий

[3] Далее, говоря об размере памяти, будем всегда иметь в виду m.

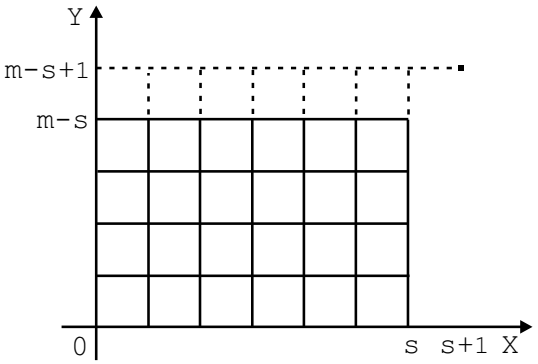

Рис. 1: Область блуждания.

момент времени он переходит в состояние (x', y'), которое определяются следующими правилами:

$$(x,y) \xrightarrow{p_1} (x',y') = \begin{cases} (x+1,y), & 0 \le x < s, 0 \le y \le m-s, \\ (x+1,y-1), & 0 \le x < s, y = m-s+1, \\ (x,y+1), & x = s, 0 \le y < m-s, \\ (x+1,y+1), & x = s, y = m-s, \\ (x+1,y), & x = s, y = m-s+1, \\ (x,y), & x = s+1, y = m-s+1, \end{cases}$$

$$(x,y) \xrightarrow{p_2} (x',y') = \begin{cases} (x,y+1), & 0 \le x \le s, 0 \le y \le m-s, \\ (x,y), & 0 \le x \le s, y = m-s+1, \\ (x-1,y), & x = s+1, y = m-s+1, \end{cases}$$

$$(x,y) \xrightarrow{q} (x',y') = \begin{cases} (x-1,y), & 0 < x \le s, 0 \le y \le m-s, \\ (x-1,y-1), & 0 < x \le s, y = m-s+1, \\ (x,y), & x = 0, y = 0, \\ (x,y-1), & x = 0, 0 < y \le m-s, \\ (x,y-2), & x = 0, y = m-s+1, \\ (x-2,y-1), & x = s+1, y = m-s+1. \end{cases}$$

Пусть $T_1 = T_1(m, s, p_1, p_2)$ — доля времени, которое процесс проводит в состоянии, соответствующем потерям заявок первого типа,

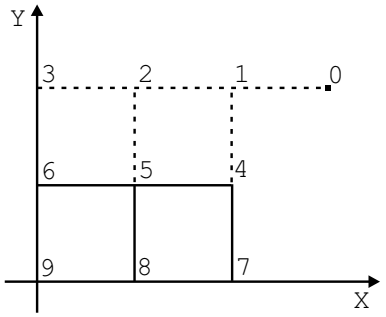

Рис. 2: Нумерация состояний для $m = 3$ и $s = 2$.

T_2 — в состояниях, соответствующих потерям заявок второго типа и c_1, c_2 — стоимости потерь заявок первого и второго типов. Тогда наша задача может быть записана следующим образом:

$$T = c_1 \cdot T_1 + c_2 \cdot T_2 \to min.$$

Пронумеруем точки области блуждания справа налево и сверху вниз, нумерацию начнем с 0 (пример нумерации состояний для $m = 3$ и $s = 2$ показан на Рис. 2). Эти точки будут соответствовать состояниям конечной однородной регулярной цепи Маркова. Всего таких состояний будет $n = (s + 1) \cdot (m - s + 2) + 1$.

3. Матрица переходных вероятностей

Для решения поставленной задачи нам необходимо определить вид матрицы переходных вероятностей марковской цепи, обозначим ее P. Сформулируем теорему о виде матрицы P и, затем, подробнее рассмотрим ее вид и приведем короткое доказательство теоремы.

Теорема 1 *При размере памяти m и заданной величине s матрица переходных вероятностей P имеет вид:*

$$P_{n\times n} = \left(\begin{array}{c|ccccc} p_1 & p_2 & 0 & \dots & 0 & q \\ \hline p_1 & & & & & \\ 0 & & & & & \\ \vdots & & & P'_{(n-1)\times(n-1)} & & \\ 0 & & & & & \\ p_1 & & & & & \end{array}\right),$$

где $n = (s+1)\cdot(m-s+2)+1$, $k = s+1$, элементы со значением p_1 - это элементы $P_{0,0}$, $P_{1,0}$ и $P_{s+2,0}$, со значением p_2 - $P_{0,1}$ и со значением q - $P_{0,s+3}$, и

$$P'_{(n-1)\times(n-1)} = \begin{pmatrix} A_{k\times k} & B_{k\times k} & C_{k\times k} & & & O \\ A_{k\times k} & B_{k\times k} & C_{k\times k} & & & \\ & A'_{k\times k} & B_{k\times k} & C_{k\times k} & & \\ & & A'_{k\times k} & \dots & \dots & \\ & & & \dots & B_{k\times k} & C_{k\times k} \\ O & & & & A'_{k\times k} & B'_{k\times k} \end{pmatrix}.$$

Рассмотрим подматрицы A, A', B, B', C входящие в матрицу P'. Матрицы A описывают процесс включений элементов второго типа в Q_2, когда она переполнена, матрицы A' - процесс включений элементов второго и первого типов в Q_2, когда она не переполнена. Через B обозначены матрицы процесса обращений (включений и исключений элементов) к Q_1, матрица B' отличается от B только последним элементом главной диагонали - он равен не 0, а q, т.к. он соответствует попытке исключения элемента из пустых очередей - процесс остается на месте. И, наконец, C - матрицы, описывающие процесс исключений элементов из Q_2.

Пусть, $\{m_{ij}\} = diag^1_k(a)$ — скалярная матрица размерности k с элементом a и $\{m_{ij}\} = diag^2_k(a,b)$ — квадратная матрица размерности k, т. ч. $m_{i,\,i+1} = a$, $m_{i+1,\,i} = b, \forall\, i = \overline{1, k-1}$ и остальные ее элементы нулевые. Тогда, при заданных ранее нумерации состояний и переходных вероятностях, $A = diag^1_k(p_2)$ (матрицы A' отличаются от A тем, что к первому элементу их главной диагонали прибавлена величина p_1), $B = diag^2_k(q, p_1)$ (в B' последний элемент главной диагонали q), в матрицах C последний элемент их главной диагонали равен q, а остальные элементы матриц C нулевые.

Коротко приведем доказательство теоремы.

Доказательство. Значения элементов $P_{0,0}$, $P_{1,0}$ и $P_{s+2,0}$, $P_{0,1}$ и $P_{0,s+3}$ очевидны в силу заданной ранее нумерации состояний. Для доказательства вида матрицы P' воспользуемся методом математической индукции.

1) База индукции. Пусть $m = 2$ и $s = 1$. Тогда размерность матрицы P' будет 6×6 и она будет иметь следующий вид:

$$P'_{6\times 6} = \left(\begin{array}{cc|c|cc|cc} p_2 & 0 & 0 & q & 0 & 0 \\ 0 & p_2 & p_1 & 0 & 0 & q \\ \hline p_2 & 0 & 0 & q & 0 & 0 \\ 0 & p_2 & p_1 & 0 & 0 & q \\ \hline 0 & 0 & p_1 + p_2 & 0 & 0 & q \\ 0 & 0 & 0 & p_2 & p_1 & q \end{array}\right).$$

2) Индуктивное предположение. Предположим, что для размера памяти m выполняется утверждение теоремы. Размерность матрицы P' будет $(s+1)(m-s+2) \times (s+1)(m-s+2)$ и она будет иметь необходимый вид.

3) Индуктивный переход. Проверим, что при размере памяти $m+1$ выполняется утверждение теоремы. Мы добавили еще одну единицу памяти. Эта единица памяти может быть добавлена либо к первой, либо ко второй очереди.

В первом случае значения s и n увеличиваются на 1. Для области блуждания это сдвиг на единицу вправо по оси Ox и, следовательно, появление $m-s+2$ новых состояний. Очевидно, что в этом случае размерность матрицы P' увеличится на $m-s+2$, размерность всех подматриц увеличится на 1, а их число не изменится. В силу заданной нумерации состояний к подматрицам A, A', B, B', C добавятся первая строка и первый столбец и, например, подматрица A будет выглядеть следующим образом:

$$A_{(k+1)\times(k+1)} = \left(\begin{array}{c|ccc} p_2 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & A_{k\times k} & \\ 0 & & & \end{array}\right).$$

Остальные подматрицы будут выглядеть подобным образом, т. е. их структура не изменится, а значит и структура матрицы P' останется прежней.

Если же свободный элемент памяти достанется второй очереди, то область блуждания вытянется на единицу вверх по оси Oy и, следовательно, мы получим $s+1$ новых состояний. Очевидно, что в этом случае размерность матрицы P' увеличится на $s+1$, а размерность всех подматриц останется прежней. Кроме того, сами подматрицы останутся прежними, их число увеличится на 1 (кроме подматриц A и B'), а структура самой матрицы P' сохранится.

Для решения задачи был разработан и реализован, при помощи пакета Mathematica, алгоритм, который генерирует матрицу P для произвольных входных данных и находит предельное распределение для рассматриваемой Марковской цепи. Далее, используя закон больших чисел для регулярных конечных однородных цепей Маркова [3] мы находим значения величин T_1 и T_2, и в зависимости от c_1 и c_2 минимизируем время T.

4. Модель с учетом "числа" изменений приоритета

В задаче, рассмотренной ранее, мы старались свести к минимуму потери поступающих заявок, не учитывая "числа" заявок первого типа, помещенных во вторую очередь. А ведь это тоже является не желательным, т. к. заявки обладающие более низким приоритетом ожидают обслуживания дольше, чем заявки с более высоким. Следовательно, представляется интересным изменение модели таким образом, чтобы минимизировать также и время, которое проводит наш процесс в состояниях, соответствующих снижению приоритета заявок первого типа.

Для учета числа изменений приоритета добавим к области блуждания, заданной в пункте 2, новые состояния $(s+1, y)$, где $0 < y \leq m - s + 1$ (см. Рис. 3).

Изменится и случайное блуждание: теперь, если процесс переходит из состояния (x, y) в состояние (x', y'), то оно определяется следующими правилами:

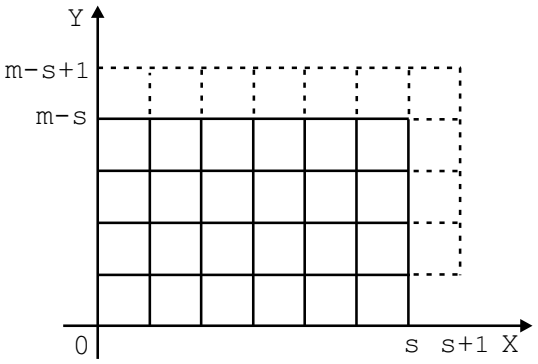

Рис. 3: Область блуждания.

$$(x,y) \xrightarrow{p_1} (x',y') = \begin{cases} (x+1,y), & 0 \le x < s, 0 \le y \le m-s, \\ (x+1,y-1), & 0 \le x < s, y = m-s+1, \\ (x+1,y+1), & x = s, 0 \le y \le m-s, \\ (x+1,y), & x = s, y = m-s+1, \\ (x+1,y), & x = s+1, 0 < y \le m-s, \\ (x,y), & x = s+1, y = m-s+1, \end{cases}$$

$$(x,y) \xrightarrow{p_2} (x',y') = \begin{cases} (x,y+1), & 0 \le x \le s, 0 \le y \le m-s, \\ (x,y), & 0 \le x \le s, y = m-s+1, \\ (x-1,y+1), & x = s+1, 0 < y \le m-s, \\ (x-1,y), & x = s+1, y = m-s+1, \end{cases}$$

$$(x,y) \xrightarrow{q} (x',y') = \begin{cases} (x-1,y), & 0 < x \le s, 0 \le y \le m-s, \\ (x-1,y-1), & 0 < x \le s, y = m-s+1, \\ (x,y), & x = 0, y = 0, \\ (x,y-1), & x = 0, 0 < y \le m-s, \\ (x,y-2), & x = 0, y = m-s+1, \\ (x-2,y), & x = s+1, 0 < y \le m-s, \\ (x-2,y-1), & x = s+1, y = m-s+1. \end{cases}$$

Немного изменим нумерацию точек области блуждания: сначала пронумеруем новые, добавленные в этой модели точки, а затем,

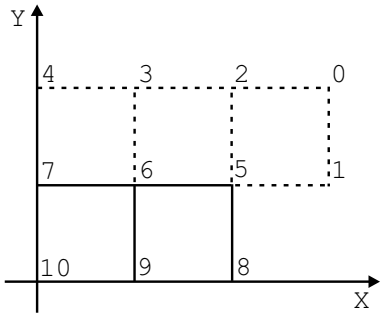

Рис. 4: Нумерация состояний для $m = 3$ и $s = 2$.

все остальные справа налево и сверху вниз (см. Рис. 4), число состояний также стало больше — $n = (s + 2) \cdot (m - s + 2) - 1$.

Пусть T_3 — доля времени, которое проводит исследуемый процесс в состояниях, соответствующих понижениям приоритета и c_3 стоимость понижения. Тогда наша задача переписывается естественным образом:

$$\mathcal{T} = c_1 \cdot T_1 + c_2 \cdot T_2 + c_3 \cdot T_3 \to min.$$

Очевидно, что матрица переходов соответствующей регулярной цепи Маркова также изменится, обозначим ее $\mathcal{P}$:

Т е о р е м а 2 *При размере памяти m и заданной величине s матрица переходных вероятностей $\mathcal{P}$ имеет вид:*

$$\mathcal{P}_{n\times n} = \left(\begin{array}{c|c} {\mathcal{P}'''}_{(n-l)\times n} & \\ \hline {\mathcal{P}''}_{l\times(n-l)} & {\mathcal{P}'}_{l\times l} \end{array} \right),$$

где

$$ {\mathcal{P}'}_{l\times l} = \begin{pmatrix} A_{k\times k} & B_{k\times k} & C_{k\times k} & & & O \\ A_{k\times k} & B_{k\times k} & C_{k\times k} & & & \\ & A_{k\times k} & B_{k\times k} & C_{k\times k} & & \\ & & A_{k\times k} & \dots & \dots & \\ & & & \dots & B_{k\times k} & C_{k\times k} \\ O & & & & A_{k\times k} & B'_{k\times k} \end{pmatrix}. $$

$n = (s+2)(m-s+2)-1$, $l = (s+1)\cdot(m-s+2)$. *Элементы матрицы* $\mathcal{P}''$ *— нули, за исключением:*

$$\mathcal{P}''_{0,0}, \mathcal{P}''_{0,s+1}, \mathcal{P}''_{1,2(s+1)}, \ldots, \mathcal{P}''_{i,(i+1)(s+1)}, \ldots, \mathcal{P}''_{(n-l-1),(n-l)(s+1)},$$

которые равны p_1*, а матрица* $\mathcal{P}'''$ *дублирует строки с номерами* $i \cdot (s+1)$*,* $i = 0, \ldots, m-s$ *объединенных матриц* $\mathcal{P}''$ *и* $\mathcal{P}'$*.*

Матрица $\mathcal{P}'$ аналогична матрице P' из п. 3 и, следовательно, доказательство того, что она имеет данную структуру полностью повторяет соответствующее доказательство для матрицы P'. Вид матриц $\mathcal{P}''$ и $\mathcal{P}'''$ следует из заданной нами нумерации состояний области блужданий.

Для решения этой задачи был использован аналогичный подход, только теперь вычисляются значения величин T_1, T_2 и T_3. Некоторые результаты вычислений для обеих моделей представлены в следующем пункте.

5. Результаты вычислений

Таблица 1. Выбор оптимального параметра s для $m = 30$.

p_1	p_2	c_1	c_2	c_3	T	s_T	$\mathcal{T}$	$s_{\mathcal{T}}$
0,4	0,2	4c	2c	c	0,4332604	25	0,433260422	25
0,4	0,2	4c	c	2c	0,2455174	28	0,2455174	28
0,4	0,2	9c	7c	5c	1,4374665	23	1,43746749	23
0,2	0,4	4c	2c	c	0,400165183	12	0,400165304	12
0,2	0,4	4c	c	2c	0,200123674	13	0,200123753	13
0,2	0,4	9c	7c	5c	1,400366232	11	1,400368081	11
0,3	0,3	4c	2c	c	0,401624008	18	0,401624129	18
0,3	0,3	4c	c	2c	0,201536324	19	0,201536421	19
0,3	0,3	9c	7c	5c	1,402722768	16	1,402725946	16

В Табл. 1 для некоторых значений вероятностей p_1, p_2 и различных параметров c_1, c_2 и c_3 приведены оптимальные значения параметра s для начальной и измененной моделей (s_T и $s_{\mathcal{T}}$ соответственно). Оказалось, что при учете случаев изменения приоритета оптимальное значение s не меняется, т. к. T_3 очень сильно отличается в меньшую сторону от T_1 и T_2, причем, чем больше m, тем больше эта разница. Это происходит в результате того, что в нашей модели время блуждания бесконечно. Представляется интересным

рассмотреть данную задачу, когда число обращений к очереди Q конечно и посмотреть, изменится ли результат. Но мы пока оставляем этот вопрос открытым.

Таблица 2. Изменение значений функции $\mathcal{T}$, при изменении значений параметра s ($m = 10$, $p_1 = 0,4$, $p_2 = 0,2$, $c_1 = 3c$, $c_2 = 2c$, $c_3 = c$).

s	T_1	T_2	T_3	$\mathcal{T}$
1	$0,11252$	$0,0900163$	$0,0874796$	$0,6050722$
2	$0,0951822$	$0,107706$	$0,0381511$	$0,5391097$
3	$0,0816208$	$0,121812$	$0,0183792$	$0,5068656$
4	$0,0708994$	$0,133331$	$0,00910062$	$0,48846082$
5	$0,0622877$	$0,143092$	$0,00437897$	$0,47742607$
6	$0,0552193$	$0,151824$	$0,00192359$	$0,47122949$
7	$0,0492901$	$0,160216$	$0,000709853$	$0,469012153$
8	$0,0442513$	$0,169082$	$0,000193141$	$0,471111041$
9	$0,0399707$	$0,180029$	$0,0000292969$	$0,4799993969$

Во второй таблице при фиксированных параметрах задачи приводятся значения долей времени T_1, T_2, T_3 и функции $\mathcal{T}$ для всех возможных значений s, что дает некоторую возможность оценить эффективность оптимального выбора параметра s (для данного примера это $s = 7$).

Список литературы

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

[2] *Соколов А. В., Тарасюк А. В.* Об оптимальном управлении циклическими FIFO очередями. Системы управления и информационные технологии // Москва-Воронеж: Научная книга. 2005. № 3(20). С. 29–32.

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

[4] *Феллер В.* Введение в теорию вероятностей и ее приложения. Т. 1. М.: Мир. 1984.