

УДК 004.01:006.72 (470.22)

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ПРИОРИТЕТНОЙ ОЧЕРЕДЬЮ В ПАМЯТИ ОДНОГО УРОВНЯ

А. В. Соколов, А. В. Драц

*Институт прикладных математических исследований
Карельского научного центра РАН*

В работе предложены новые математические модели для представления приоритетной очереди в виде n последовательных FIFO-очереди. На основе этих моделей предложены алгоритмы и комплекс программ, которые позволяют для заданных вероятностей выполнения основных операций находить оптимальный способ представления приоритетной очереди в смысле максимизации среднего времени до переполнения памяти и в смысле минимизации доли потерянных при переполнении элементов.

Ключевые слова: приоритетная очередь, FIFO-очереди, случайное блуждание, цепи Маркова, динамические структуры данных.

A. V. Sokolov, A. V. Drats. OPTIMAL CONTROL OF PRIORITY QUEUE IN SINGLE LEVEL MEMORY

The paper contains new mathematical models concerning representation of a priority queue in single level memory as n serial FIFO queues. Proceeding from these models we propose the algorithms and programs which allow to find the optimal way of representation of a priority queue in the sense of maximizing the average time until memory overflow and in the sense of minimizing the proportion of lost elements when the probabilities are known.

Key words: Priority queue, FIFO-queues, random walks, Markov chains, dynamic data structures.

ВВЕДЕНИЕ

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

([Кормени др., 2000; Кнут, 2001; Боллапаргада и др., 2002]).

В этих и других работах различные методы представления приоритетной очереди сравниваются с точки зрения стоимости (времени) выполнения основных операций. Целью данной работы является изучение динамики изменения длины очереди в разных методах представления. При этом допускается возможность ее переполнения, что, например, в маршрутизаторах является обычной ситуацией. В

работе будут рассмотрены представления в виде массива и в виде n последовательных FIFO-очереди. Предлагаются новые математические модели в виде случайных блужданий по целочисленной n -мерной решетке, описывающие работу приоритетной очереди, представленной в виде n последовательных FIFO-очереди. Для представления в виде массива использована предложенная ранее модель в виде одномерного случайного блуждания. На основе этих моделей разработаны программы, которые позволяют для заданных значений параметров проектируемой или настраиваемой программной или аппаратной системы выбирать оптимальный метод представления приоритетной очереди.

В случае представления в виде массива вместе с каждым элементом необходимо хранить его приоритет. В этом случае переполнение наступит только тогда, когда вся свободная память закончится, но часть памяти тратится на хранение приоритетов.

В случае представления приоритетной очереди в виде n последовательных FIFO-очереди все элементы с одинаковыми приоритетами помещаются в одну FIFO-очередь. (Число FIFO-очереди равно количеству приоритетов в приоритетной очереди). Каждой FIFO-очереди выделен свой участок памяти. В этом случае память не тратится на хранение приоритетов, но могут возникнуть потери памяти тогда, когда одна из FIFO-очереди полностью исчерпает свою часть памяти, а остальные заполняют ее не полностью. Здесь возникает дополнительная задача оптимального разбиения памяти между FIFO-очередями.

Если использовать связанное представление очереди, т. е. хранить каждую FIFO-очередь не в виде массива, а в виде связанного списка, то этот метод представления приоритетной очереди сводится к представлению в виде массива. Вместе с каждым элементом хранить нужно будет не его приоритет, а указатель на следующий элемент FIFO-очереди. Поэтому этот случай представления приоритетной очереди мы отдельно рассматривать не будем.

В качестве первого критерия оптимальности рассмотрим среднее время работы до переполнения T . Этот критерий применяется в том случае, если переполнение является аварийной ситуацией.

Переполнение памяти не всегда является аварийной ситуацией. Если допускаются потери элементов, то в случае попытки включения элемента, он просто отбрасывается, если нет

свободной памяти для него. Так продолжается до тех пор, пока не появится свободная память, т. е. до исключения элемента. Такое поведение очереди называется «сбросом хвоста» [Боллапаргада и др., 2002]. Такая схема применяется, например, в работе сетевых маршрутизаторов. В этом случае разумно будет минимизировать долю времени, которую приоритетная очередь проводит в состоянии «сброса хвоста» P^* . Тем самым минимизируется доля потерянных элементов. Эта величина рассмотрена в качестве второго критерия. Будем считать, что такая схема работы применяется в течение длительного промежутка времени, поэтому время можно условно считать бесконечным.

Постановка задачи

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

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

$$p_1 + \dots + p_n + q + r = 1.$$

Вероятности не зависят от времени и количества элементов в очереди. Наивысший приоритет равен n , наименьший — 1. Работа начинается с пустой очереди. Предположим также, что не происходит завершения работы в случае исключения элемента из пустой структуры данных.

Требуется построить математические модели рассматриваемых методов представления приоритетной очереди, на их основе найти среднее время работы T и долю времени P^* и разработать программы выбора лучшего метода представления приоритетной очереди для данных критериев оптимальности.

КРИТЕРИЙ МАКСИМИЗАЦИИ ВРЕМЕНИ РАБОТЫ ДО ПЕРЕПОЛНЕНИЯ. ПРЕДСТАВЛЕНИЕ В ВИДЕ МАССИВА

Введем обозначения: $p = p_1 + \dots + p_n$ — суммарная вероятность включения элемента в приоритетную очередь, $x = x_1 + \dots + x_n$ — общее количество элементов, l — отношение размера памяти, которую занимает ключ к размеру информационной части.

Будем считать, что информационная часть занимает одну единицу памяти. $M = \left\lfloor \frac{m}{1+l} \right\rfloor$ – размер памяти, который тратится на размещение информационных частей, он же равен общему количеству элементов, которые можно разместить в памяти. Например, если элемент массива объявлен как `struct node{DataT info; DataKey key;}; sizeof(info)= 4*sizeof(key); sizeof(info) = 4 байт` и размер памяти равен

40 байт, то $m = 10$ единиц памяти, $l = 1/4$ и $M = \frac{m}{1+l} = 8$ единиц памяти (единица памяти равна 4 байта). В работах [Феллер, 1964], [Тарасюк, 2008], [Аксенова, Соколов, 2008] в качестве математической модели было предложено случайное блуждание в интервале длины M . $x = -1$ – отражающий экран. $x = M + 1$ – поглощающий экран. Переходы между состояниями:

$$\begin{aligned} (x) &\xrightarrow{p} (x') = \begin{cases} (x+1) & 0 \leq x \leq M-1 \\ \text{Поглощающее состояние} & x = M \end{cases} \\ (x) &\xrightarrow{q} (x') = \begin{cases} (x) & x = 0 \\ (x-1) & 1 \leq x \leq M \end{cases} \\ (x) &\xrightarrow{r} (x') = (x) \quad 0 \leq x \leq M \end{aligned} \quad (1)$$

Было показано, что среднее время работы до переполнения равно

$$T = \begin{cases} \frac{-M-1}{q-p} + \frac{q}{(q-p)^2} \left(\left(\frac{q}{p} \right)^{M+1} - 1 \right), & p \neq q \\ \frac{1}{2q} (M+1)(M+2), & p = q \end{cases} \quad (2)$$

КРИТЕРИЙ МАКСИМИЗАЦИИ ВРЕМЕНИ РАБОТЫ ДО ПЕРЕПОЛНЕНИЯ. ПРЕДСТАВЛЕНИЕ В ВИДЕ n ПОСЛЕДОВАТЕЛЬНЫХ FIFO-ОЧЕРЕДЕЙ

Пусть k_1, \dots, k_n – фиксированное разбиение памяти между очередями. В качестве математической модели рассмотрим блуждание

по целочисленному n -мерному параллелепипеду, ограниченному гиперплоскостями $x_i = 0$, $x_i = k_i$, ($1 \leq i \leq n$). $x_i = -1$ ($1 \leq i \leq n$) – отражающие барьеры. $x_i = k_i + 1$ – поглощающие барьеры. Количество состояний равно $\prod_{i=1}^n (k_i + 1)$.

Перенумеруем состояния в лексикографическом порядке следующим образом:

$$\begin{aligned} &(0, 0, 0, \dots, 0, 0), (0, 0, 0, \dots, 0, 1), \dots, (0, 0, 0, \dots, 0, k_n - 1), (0, 0, 0, \dots, 0, k_n) \\ &(0, 0, 0, \dots, 1, 0), (0, 0, 0, \dots, 1, 1), \dots, (0, 0, 0, \dots, 1, k_n - 1), (0, 0, 0, \dots, 1, k_n) \\ &\dots \\ &(0, 0, 0, \dots, k_{n-1} - 1, 0), (0, 0, 0, \dots, k_{n-1} - 1, 1), \dots, (0, 0, 0, \dots, k_{n-1} - 1, k_n - 1), \\ &(0, 0, 0, \dots, k_{n-1} - 1, k_n) \\ &(0, 0, 0, \dots, k_{n-1}, 0), (0, 0, 0, \dots, k_{n-1}, 1), \dots, (0, 0, 0, \dots, k_{n-1}, k_n - 1), \\ &(0, 0, 0, \dots, k_{n-1}, k_n) \\ &\dots \\ &(1, 0, 0, \dots, 0, 0), (1, 0, 0, \dots, 0, 1), \dots, (1, 0, 0, \dots, 0, k_n - 1), (1, 0, 0, \dots, 0, k_n) \\ &\dots \\ &(k_1, 0, 0, \dots, 0, 0), (k_1, 0, 0, \dots, 0, 1), \dots, (k_1, 0, 0, \dots, 0, k_n - 1), (k_1, 0, 0, \dots, 0, k_n) \\ &\dots \\ &(k_1, k_2, k_3, \dots, k_{n-1}, 0), (k_1, k_2, k_3, \dots, k_{n-1}, 1), \dots, (k_1, k_2, k_3, \dots, k_{n-1}, k_n - 1), \\ &(k_1, k_2, k_3, \dots, k_{n-1}, k_n) \end{aligned} \quad (3)$$

Пример нумерации состояний показан на рис. 1. Переход системы из состояния (x_1, \dots, x_n)

в состояние (x'_1, \dots, x'_n) происходит в соответствии с правилами (4).

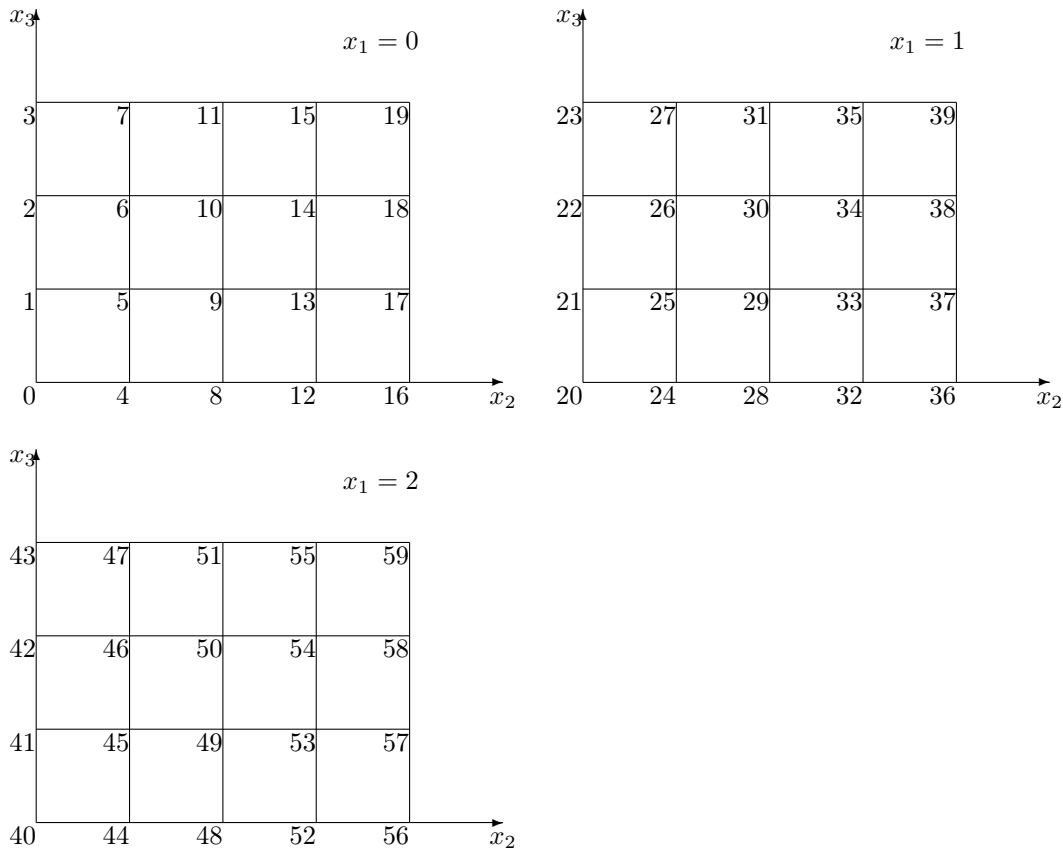


Рис. 1. Нумерация состояний при $m = 9$, $n = 3$, $k_1 = 2$, $k_2 = 4$, $k_3 = 3$

$$\begin{aligned}
 (\dots, x_i, \dots, x_j, \dots) &\xrightarrow{p_i} \begin{cases} (\dots, x_i + 1, \dots, x_j, \dots) & 0 \leq x_i < k_i \\ \text{Поглощающее состояние} & x_i = k_i \end{cases} \\
 (\dots, x_i, \dots, x_j, \dots) &\xrightarrow{q} \begin{cases} (\dots, x_i, \dots, x_j, \dots) & x_1 = \dots = x_n = 0 \\ (\dots, x_i - 1, \dots, x_j, \dots) & 1 \leq x_i \leq k_i, x_{i+1} = \dots = x_n = 0 \end{cases} \quad (4) \\
 (\dots, x_i, \dots, x_j, \dots) &\xrightarrow{r} (\dots, x_i, \dots, x_j, \dots) \\
 1 \leq i \leq n, \quad 1 \leq j \leq n, \quad i \neq j.
 \end{aligned}$$

Для описания процесса построим соответствующую поглощающую цепь Маркова. Необходимо построить матрицу вероятностей переходов Q . Для этого нужно знать номер каждого состояния. Для вычисления номера состояния построим функцию $F(x_1, x_2, \dots, x_n) = I$, где I – номер состояния.

Если есть только одна очередь, то функция имеет вид: $F(x_1) = x_1$. Если появляется вторая очередь, то состояния (x_i) становятся эквивалентными $(x_i, 0)$. Между состояниями $(x_i, 0)$ и $(x_{i+1}, 0)$ нужно вставить $k_2 - 1$ состояний: $(x_i, 1), (x_i, 2), \dots, (x_i, k_{n-1} - 1)$, поэтому функция будет иметь вид $F(x_1, x_2) = k_2 x_1 + x_2$.

При появлении третьей очереди получаем функцию

$$F(x_1, x_2, x_3) = k_3(k_2 x_1 + x_2) + x_3.$$

Рассуждая таким образом, получим для n очередей функцию

$$F(x_1, x_2, \dots, x_n) = k_n(k_{n-1}(\dots(k_3(k_2 x_1 + x_2) + x_3) + \dots + x_{n-1})) + x_n.$$

Для построения матрицы переходных вероятностей нужно в цикле перебрать все состояния (от $(0, 0, \dots, 0, 0)$ до $(0, 0, \dots, 0, M)$) и вычислить вероятности перехода из I -того состояния в остальные ($I = F(x_1, x_2, \dots, x_n)$):

$$\begin{aligned}
Q[F(x_1, x_2, \dots, x_n)][F(x_1 + 1, x_2, \dots, x_n)] &= p_1 \\
\vdots \\
Q[F(x_1, x_2, \dots, x_n)][F(x_1, x_2, \dots, x_n + 1)] &= p_n \\
Q[F(x_1, \dots, x_i, \dots, x_n)][F(x_1, \dots, x_i - 1, \dots, x_n)] &= q, \text{ где } i = \max_{1 \leq j \leq n} \{j : x_j > 0\} \\
Q[F(x_1, x_2, \dots, x_n)][F(x_1, x_2, \dots, x_n)] &= r
\end{aligned} \tag{5}$$

Для вычисления среднего времени блуждания нужно вычислить фундаментальную матрицу $N = (E - Q)^{-1}$ [Кемени, Снелл, 1970]. N_{ij} – среднее время, которое процесс проведет в состоянии j , если он начался в состоянии i . Поскольку работа начинается с пустых структур данных, то $i = 0$ и $T = \sum_j N_{0j}$.

Чтобы найти оптимальное разбиение памяти, необходимо перебрать все возможные разбиения памяти, для каждого из них найти среднее время работы до переполнения и выбрать наибольшее время. Всего для размера памяти m и n очередей существует $C_{m-1}^{m-1} = \frac{(m-1)!}{(n-1)!(m-n)!}$ способов разбиения [Риордан, 1963].

УПРАВЛЕНИЕ НА БЕСКОНЕЧНОМ ВРЕМЕНИ. ПРЕДСТАВЛЕНИЕ В ВИДЕ МАССИВА

Как и в предыдущем критерии часть памяти тратится на хранение приоритетов. Всего можно хранить $M = \left\lfloor \frac{m}{l+1} \right\rfloor$ элементов в памяти. В качестве математической модели рассмотрим блуждание в одномерном интервале длины M . Переход из состояния (x) в состояние (x') происходит в соответствии со следующими правилами:

$$\begin{aligned}
(x) \xrightarrow{p} (x') &= \begin{cases} (x+1) & 0 \leq x \leq M \\ (x) & x = M+1 \end{cases} \\
(x) \xrightarrow{q} (x') &= \begin{cases} (x) & x = 0 \\ (x-1) & 1 \leq x \leq M \\ (x-2) & x = M+1 \end{cases} \\
(x) \xrightarrow{r} (x') &= \begin{cases} (x) & 0 \leq x \leq M \\ (x-1) & x = M+1 \end{cases}
\end{aligned}$$

Задача сводится к вычислению доли времени, которую проводит одна FIFO-очередь в состоянии «сброса хвоста». Эта задача была решена в работе [Аксенова и др., 2009]:

$$P^* = \begin{cases} \frac{q-p}{\left(\frac{q}{p}\right)^{M+1} - 1} & q \neq p \\ \frac{p}{M+1} & q = p \end{cases} \tag{6}$$

Управление на бесконечном времени. Представление в виде n последовательных FIFO-очереди

Пусть k_1, \dots, k_n – фиксированное разбиение памяти между очередями. В качестве математической модели рассмотрим блуждание по целочисленному n -мерному параллелепипеду, ограниченному гиперплоскостями $x_i = 0$, $x_i = k_i$, ($1 \leq i \leq n$). Дополнительно к каждому состоянию на гиперплоскостях $x_j = k_j$ введем состояние «сброса хвоста». Количество обычных состояний равно $\prod_{i=1}^n (k_i + 1)$.

Количество состояний «сброса хвоста», соответствующих j -й очереди равно

$$\prod_{\substack{i=1 \\ i \neq j}}^n (k_i + 1).$$

Общее количество состояний «сброса хвоста» равно:

$$\sum_{j=1}^n \left(\prod_{\substack{i=1 \\ i \neq j}}^n (k_i + 1) \right).$$

Перенумеруем сначала обычные состояния в лексикографическом порядке, как в предыдущем параграфе, затем перенумеруем состояния «сброса хвоста», соответствующие гиперплоскости $x_1 = k_1$, затем соответствующие гиперплоскости $x_2 = k_2$ и т. д. На каждой гиперплоскости $x_i = k_i$ перенумеруем состояния в лексикографическом порядке, по аналогии с обычными состояниями. Пример нумерации состояний показан на рис. 2. Состояния с номерами больше 35 являются состояниями «сброса хвоста».

Перенумеруем состояния в лексикографическом порядке следующим образом:

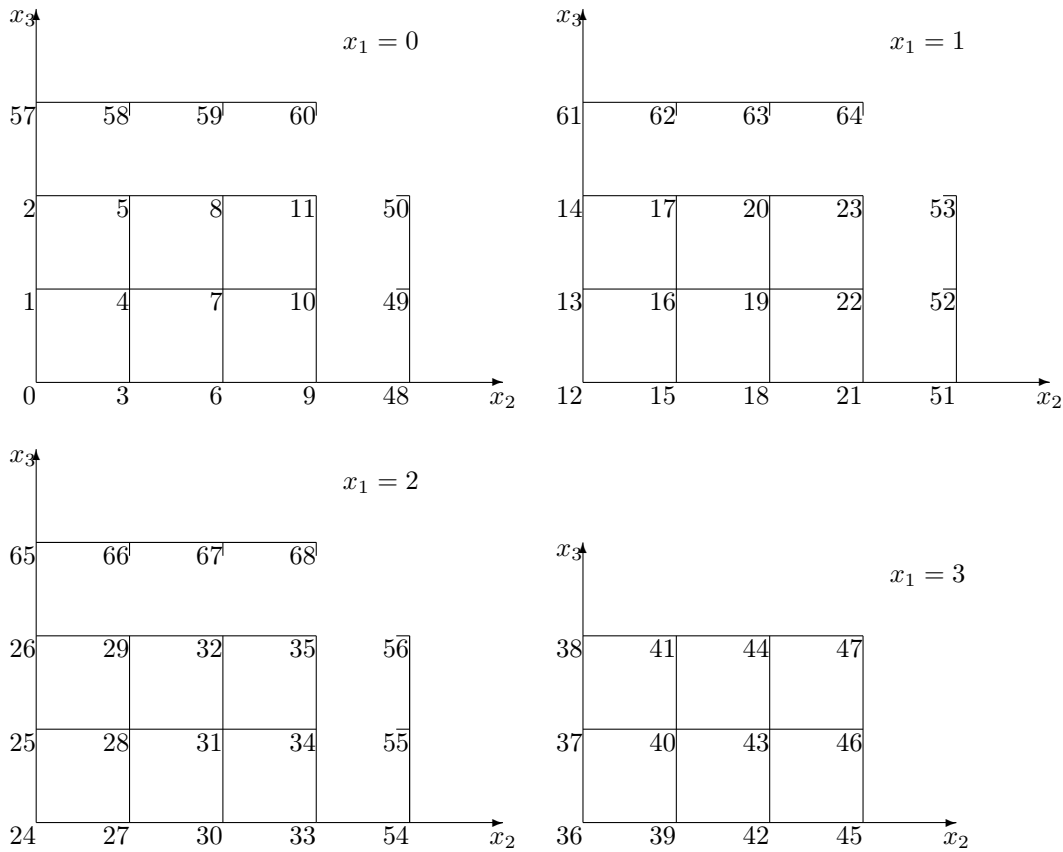


Рис. 2. Нумерация состояний при $m = 7$, $n = 3$, $k_1 = 2$, $k_2 = 3$, $k_3 = 2$

Переход системы из состояния (x_1, \dots, x_n) в состояние (x'_1, \dots, x'_n) происходит в соответствии с правилами (7).

Для описания процесса построим соответствующую регулярную цепь Маркова. Постро-

им матрицу вероятностей переходов Q . Для обычных состояний ее вид будет такой же, как и в предыдущем параграфе. Для состояния «сброса хвоста» на плоскости $x_i = k_i$ матрица вероятностей переходов принимает вид (8).

$$\begin{aligned}
 (\dots, x_i, \dots, x_j, \dots) &\xrightarrow{p_i} \begin{cases} (\dots, x_i + 1, \dots, x_j, \dots) & 0 \leq x_i \leq k_i, x_j \leq k_j \\ (\dots, x_i + 1, \dots, x_j - 1, \dots) & 0 \leq x_i \leq k_i, x_j = k_j + 1 \\ (\dots, x_i, \dots, x_j, \dots) & x_i = k_i + 1, x_j \leq k_j \\ (\dots, x_i, \dots, x_j - 1, \dots) & x_i = k_i + 1, x_j = k_j + 1 \end{cases} \\
 (\dots, x_i, \dots, x_j, \dots) &\xrightarrow{q} \begin{cases} (\dots, x_i, \dots, x_j, \dots) & x_1 = \dots = x_n = 0 \\ (\dots, x_i, \dots, x_j - 1, \dots) & 0 \leq x_i < k_i, x_j > 0, x_{j+1} = \dots = x_n = 0 \\ (\dots, x_i - 1, \dots, x_j - 1, \dots) & x_i = k_i + 1, x_j > 0, x_{j+1} = \dots = x_n = 0 \\ (\dots, x_i, \dots, x_j - 2, \dots) & 0 \leq x_i < k_i, x_j = k_j + 1, x_{j+1} = \dots = x_n = 0 \\ (\dots, x_i - 1, \dots, x_j - 2, \dots) & x_i = k_i + 1, x_j = k_j + 1, x_{j+1} = \dots = x_n = 0 \end{cases} \\
 (\dots, x_i, \dots, x_j, \dots) &\xrightarrow{r} \begin{cases} (\dots, x_i, \dots, x_j, \dots) & 0 \leq x_i \leq k_i, x_j \leq k_j \\ (\dots, x_i, \dots, x_j - 1, \dots) & 0 \leq x_i \leq k_i, x_j = k_j + 1 \end{cases}
 \end{aligned} \tag{7}$$

$$\begin{aligned}
 F(x_1, \dots, x_{i-1}, k_i + 1, x_{i-1}, \dots, x_n) &= \prod_{\substack{i=1 \\ i \neq j}}^n (k_i + 1) + \sum_{j=1}^{i-1} \left(\prod_{\substack{l=1 \\ l \neq j}}^n (k_l + 1) \right) \\
 &+ k_n(k_{n-1} \dots k_{i+1}(k_{i-1}(\dots(k_3(k_2x_1 + x_2) + x_3) + \dots + x_{i-1}) + x_{i+1} + \dots + x_{n-1}) + x_n
 \end{aligned} \tag{8}$$

Таблица 1. Среднее время работы до переполнения T , $n = 4$, $m = 24$

r	0	0	0	0	0,5	0	0	0,1	0
p_1	0,15	0,2	0,05	0,2	0,1	0,4	0,05	0,18	0,1
p_2	0,15	0,15	0,1	0,2	0,1	0,1	0,05	0,1	0,1
p_3	0,15	0,1	0,15	0,2	0,1	0,05	0,1	0,1	0,1
p_4	0,15	0,05	0,2	0,2	0,1	0,05	0,4	0,02	0,1
q	0,4	0,5	0,5	0,2	0,1	0,4	0,4	0,5	0,6
k_1	9	13	6	7	7	17	5	13	9
k_2	6	6	7	7	7	3	4	5	6
k_3	5	3	6	6	6	2	6	4	5
k_4	4	2	5	4	4	2	9	2	4
T	64,3	234,98	200,46	27,32	54,65	78,32	68,37	1165,8	2868,62
l_1	55,05	182	182	21,11	42,22	55,05	55,05	729,5	112102,41
l_2	75,01	306	306	27,78	55,56	75,01	75,01	2000,45	49763,85
l_4	90	420	420	32,78	65,56	90	90	4086,81	14678,92
l_8	100	506	506	36,11	72,22	100	100	6506,26	2839,29

Таблица 2. Доля времени, которую приоритетная очередь проводит в состоянии «сброса хвоста» T , $n = 4$, $m = 16$

r	0	0	0	0	0,5	0	0	0,1	0
p_1	0,15	0,2	0,05	0,2	0,1	0,4	0,05	0,18	0,1
p_2	0,15	0,15	0,1	0,2	0,1	0,1	0,05	0,1	0,1
p_3	0,15	0,1	0,15	0,2	0,1	0,05	0,1	0,1	0,1
p_4	0,15	0,05	0,2	0,2	0,1	0,05	0,4	0,02	0,1
q	0,4	0,5	0,5	0,2	0,1	0,4	0,4	0,5	0,6
k_1	9	10	4	12	12	13	7	9	6
k_2	3	3	5	2	1	1	2	3	4
k_3	3	2	4	1	1	1	3	3	3
k_4	2	1	3	1	2	1	4	1	3
P^*	0,2011	0,0433	0,0456	0,6	0,3	0,2	0,2009	0,01	0,0041
l_1	0,20534	0,0556	0,0556	0,6	0,3	0,1155	0,1155	0,0155	0,00543
l_2	0,20234	0,0455	0,0455	0,6	0,3	0,1094	0,1094	0,0094	0,00234
l_4	0,201	0,0385	0,0385	0,6	0,3	0,1058	0,1058	0,00582	0,00103
l_8	0,2005	0,0333	0,0333	0,6	0,3	0,1036	0,1036	0,00365	0,00046

У построенной цепи Маркова существует предельный вектор α , который удовлетворяет уравнению $\alpha * Q = \alpha$ [Кемени, Снелл, 1970]. По закону больших чисел значение α_i является долей времени, которое процесс проводит в состоянии i . Чтобы найти долю времени, которую система проводит в состоянии «сброса хвоста», необходимо вычислить сумму тех компонент вектора α , которые соответствуют состояниям «сброса хвоста». В примере на рис. 2 необходимо будет просуммировать компоненты с номерами от 36 до 68.

ЧИСЛЕННЫЕ РЕЗУЛЬТАТЫ

Для решения задачи были написаны программы, которые вычисляют матрицу Q , среднее время работы до переполнения и долю времени, которую система проводит в состоянии «сброса хвоста» и находят оптимальное разбиение памяти между очередями в случае представления приоритетной очереди в виде последовательных FIFO-очереди. Некоторые результаты работы программ представлены в таблицах 1 и 2. В строке T находится среднее время работы до переполнения в случае представления в виде n FIFO-очереди, в строках k_i ($1 \leq i \leq 4$) – оптимальное разбиение памяти. В строке P^* находится доля времени, которую система проводит в состоянии «сброса хвоста» в случае представления в виде n FIFO-очереди. В строке l_1 указано время работы в случае представления в виде массива, когда размер ключа равен размеру информационной части. (На ключи тратится половина памяти.)

В строке l_2 – когда размер ключа равен 1/2 размеру информационной части. (На ключи тратится 1/3 часть памяти.)

В строке l_4 – когда размер ключа равен 1/4 размеру информационной части. (На ключи тратится 1/5 часть памяти.)

В строке l_8 – когда размер ключа равен 1/8

размеру информационной части. (На ключи тратится 1/9 часть памяти.)

Предложенные в статье модели и разработанные на их основе программы могут дать для известных значений вероятностей выполнения операций оптимальные размеры очередей разных приоритетов, которые системный администратор может учитывать при настройке маршрутизатора.

Работа выполнена при финансовой поддержке РФФИ, грант 09-01-00330.

ЛИТЕРАТУРА

Аксенова Е. А., Драц А. В., Соколов А. В. Оптимальное управление n FIFO-очередями на бесконечном времени // Информационно-управляющие системы. 2009. № 6. С. 46–54.

Аксенова Е. А., Соколов А. В. Анализ некоторых методов реализации приоритетной очереди // Межвуз. сб. «Стохастическая оптимизация в информатике» Изд-во С.-Петербургского университета, 2008. Вып. 4. С. 61–71.

Боллапаргада В., Мэрфи К., Расс У. Структура операционной системы Cisco IOS. М.: Вильямс, 2002. 208 с.

Кемени Дж., Снелл Дж. Конечные цепи Маркова. М.: Наука, 1970. 272 с.

Кнут Д. Искусство программирования для ЭВМ. М.: Вильямс, 2001. Т. 3. 434 с.

Кормен Е., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2000. 960 с.

Риордан Дж. Введение в комбинаторный анализ. М.: Издательство иностранной литературы, 1963. 375 с.

Седжвик Р. Фундаментальные алгоритмы на C++. Анализ. Структуры данных. Сортировка. Поиск пер. с англ. Роберт Седжвик. К.: Издательство «ДиаСофт», 2001. 688 с.

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

Феллер В. Введение в теорию вероятностей и ее приложения. М.: Мир, 1964. 499 с.

СВЕДЕНИЯ ОБ АВТОРАХ:

Соколов Андрей Владимирович

д. ф.-м. н.

Институт прикладных математических исследований КарНЦ РАН

ул. Пушкинская, 11, Петрозаводск, Республика Карелия, Россия, 185910

эл. почта: avs@krc.karelia.ru

тел.: (8142) 766313

Драц Андрей Владимирович

Институт прикладных математических исследований КарНЦ РАН

ул. Пушкинская, 11, Петрозаводск, Республика Карелия, Россия, 185910

эл. почта: adeon88@mail.ru

тел.: (8142) 763313

Sokolov, Andrey

Institute of Applied Mathematical Research, Karelian

Research Centre, Russian Academy of Science

11 Pushkinskaya St., 185910 Petrozavodsk, Karelia, Russia

e-mail: avs@krc.karelia.ru

Drats, Andrey

Institute of Applied Mathematical Research, Karelian

Research Centre, Russian Academy of Science

11 Pushkinskaya St., 185910 Petrozavodsk, Karelia, Russia

e-mail: adeon88@mail.ru