

Оптимальное управление двумя очередями, работающими по принципу настраиваемых очередей¹

А. М. Сазонов²,

*Институт прикладных математических исследований Карельского
научного центра Российской академии наук, Петрозаводск, РФ,
sazonov@cs.karelia.ru*

В работе предлагается математическая модель и решается задача оптимального управления настраиваемой очередью, представленной в виде двух очередей в случае их последовательного циклического представления, а также имитационная модель для представления друг за другом.

Ключевые слова: Структуры данных, настраиваемые очереди, случайные блуждания, регулярные марковские цепи.

1. Введение

Во многих приложениях, например, при разработке различных сетевых устройств и встроенных операционных систем требуется работа с несколькими очередями, расположенными в общем пространстве памяти. Механизм страничной виртуальной памяти здесь не используется, и вся работа происходит в нескольких пулах оперативной памяти.

Так, в сетевых устройствах очереди являются частью механизма управления заторами. В условиях высокого трафика маршрутизатор не всегда справляется с обработкой приходящих пакетов данных. Старые пакеты накапливаются, занимая собой память, в результате чего новые пакеты отбрасываются и теряются. С помощью механизма управления заторами можно смягчить эту ситуацию и настроить маршрутизатор, выставив пакетам приоритеты и обрабатывая их согласно некой стратегии обслуживания.

Одна из распространенных стратегий обслуживания это настраиваемые очереди (Custom Queues) [1], или взвешенные очереди в

¹Работа выполнена при поддержке РФФИ, проект №15-01-03404-а.

²©А. М. Сазонов, 2017

терминологии [2]. Механизм настраиваемых очередей разработан для того, чтобы можно было предоставить всем классам трафика какой-то минимум пропускной способности. Под весом данного класса понимается процент предоставляемой классу трафика пропускной способности от полной пропускной способности выходного интерфейса.

SQ обрабатывает трафик, ориентируясь на количество пакетов или байтов, которые надо отправить. Очереди обслуживаются по алгоритму round-robin – часть пакетов обрабатывается и пересылается для каждой очереди. Если очередь пуста, то маршрутизатор будет посылать пакеты из следующей очереди.

В [4] - [8] предлагались математические модели и алгоритмы оптимального управления FIFO-очередями. В [9] была предложена математическая модель и решалась задача оптимального управления для настраиваемой очереди, представленной в виде двух очередей в случае их последовательного циклического представления. В качестве критерия оптимальности рассматривается минимальная доля потерянных элементов при бесконечном времени работы очередей. В данной работе предложена имитационная модель этого процесса, проведено ее сравнение с математической моделью, а также проанализирована разработанная имитационная модель для настраиваемой очереди, представленной в виде двух очередей в случае их представления друг за другом. Такой метод был предложен в [10] для представления FIFO-очередей. В [11] этот метод представления FIFO-очередей анализировался для случая, когда включения происходят на нечетных шагах времени, а исключения – на четных. Задача оптимального управления двумя настраиваемыми очередями для случая, когда включения происходят на нечетных шагах, а исключения – на четных, была решена в [12].

2. Постановка задачи для последовательного циклического представления

Пусть в памяти размером в n единиц мы работаем с двумя последовательными циклическими очередями, работающими по принципу настраиваемых очередей, с элементами фиксированного размера, и каждой очереди выделен размер памяти $k = \frac{n}{2}$. На каждом

шаге дискретного времени происходят операции включения или исключения элементов в одну из очередей с заданными вероятностями. Исключение элементов происходит одновременно из обеих очередей во время так называемого цикла просмотра очередей. Количество исключаемых элементов из первой очереди равно c_1 , из второй – c_2 . Заметим, что в этой задаче мы не можем выбирать параметры c_1 и c_2 произвольно, т.к. они зависят от длительности цикла просмотра очередей в рассматриваемом сетевом устройстве t и скорости выходного трафика v ($c_1 + c_2 = vt$). Например, если $t = 2$ сек., а $v = 100$ Мбит/сек., то $c_1 + c_2 = 200$ Мбит. Это значит, что в зависимости от вероятностных характеристик очередей мы можем менять веса очередей, но сумма весов определяется характеристиками сети и устройства.

Будем искать такие c_1, c_2 , чтобы вероятность потери в стационарном режиме в обеих очередях P_{loss} была минимальной. Шаги цикла просмотра очередей объединяем в один, чтобы обеспечить марковость процесса.

Пусть p_1 и p_2 – вероятности включения элемента в первую и вторую очереди соответственно, q – вероятность исключения из обеих очередей. Введем вероятность операции, не изменяющей длины очередей (например, чтение): r . Соответственно, $p_1 + p_2 + q + r = 1$.

Тогда состояние на каждом шаге определяется наступлением одного из следующих событий:

- включение в первую очередь с вероятностью p_1 ;
- включение во вторую очередь с вероятностью p_2 ;
- исключение из первой очереди c_1 и из второй очереди c_2 элементов с вероятностью q ;
- отсутствие событий с вероятностью r .

3. Математическая модель для последовательного циклического представления

Обозначим через x и y текущие длины первой и второй очередей соответственно. В качестве математической модели рассматриваем

случайное блуждание в двумерном пространстве по целочисленной решетке в области $0 \leq x \leq k, 0 \leq y \leq k$.

Определим схему переходов между состояниями. Пусть (x, y) – текущее состояние процесса, тогда блуждание в области $0 \leq x \leq k, 0 \leq y \leq k$ можно описать следующим образом:

$$(x, y) \xrightarrow{r} (x, y)$$

$$(x, y) \xrightarrow{p_1} \begin{cases} (x+1, y) & x < k \\ (x, y) & x = k \end{cases}$$

$$(x, y) \xrightarrow{p_2} \begin{cases} (x, y+1) & y < k \\ (x, y) & y = k \end{cases}$$

$$(x, y) \xrightarrow{q} \begin{cases} (x - c_1, y - c_2) & c_1 + 1 \leq x \leq k, c_2 + 1 \leq y \leq k \\ (0, y - c_2 - (c_1 - x)) & x \leq c_1, c_1 + c_2 - x + 1 \leq y \leq k \\ (x - c_1 - (c_2 - y), 0) & c_1 + c_2 - y + 1 \leq x \leq k, y \leq c_2 \\ (0, 0) & \text{иначе} \end{cases}$$

Необходимо минимизировать число потерянных элементов при переполнении очередей. Иначе говоря, нужно найти такое c_1 ($c_2 = vt - c_1$), чтобы стационарная вероятность потери P_{loss} была минимальной. Таким образом, в случае двух очередей данная задача может быть рассмотрена как одномерный случай задачи целочисленного программирования, где функция критерия оптимальности задается алгоритмически.

Представим случайное блуждание в виде конечной регулярной марковской цепи с переходной матрицей P . Общее количество состояний в цепи $N = (k+1) \cdot (k+1)$.

Далее, согласно вышеуказанной схеме переходов составляется матрица переходных вероятностей $P = P(c_1, c_2)$. При составлении матрицы для каждого конкретного состояния определяются те состояния, в которые процесс переходит при выполнении допустимых операций, и вычисляются соответствующие вероятности переходов.

Следующим шагом является решение уравнения $\alpha P = \alpha$, где α – предельный вектор для полученной марковской цепи.

Элемент вектора α_i – это средняя доля времени, которое процесс проводит в состоянии i . Поскольку моменты прихода не зависят от числа элементов в очереди, для вычисления стационарной вероятности потери, нужно просуммировать элементы вектора α , соответствующие состояниям, когда первая очередь максимально заполнена, умножив их на p_1 , и элементы вектора α , соответствующие состояниям, когда вторая очередь максимально заполнена, умножив их на p_2 . При введенной нумерации это будут элементы с номерами $k + (k + 1) \cdot j$, $j = 0, \dots, k$ и последние k элементов вектора. Получаем все возможные вероятности потери в зависимости от c_1, c_2 .

$$P_{loss}(c_1, c_2) = \sum_{j=0}^k \alpha_{k+(k+1) \cdot j}(c_1, c_2) \cdot p_1 + \sum_{j=N-k}^N \alpha_j(c_1, c_2) \cdot p_2 \quad (1)$$

Далее из всех возможных вероятностей потери выбираем минимальную. Значения c_1, c_2 , при которых достигается минимум, являются оптимальными. Данный процесс был автоматизирован с помощью программы на языке C++.

Для проверки результатов была реализована имитационная модель данного процесса.

4. Имитационная модель для представления друг за другом

При представлении очередей друг за другом память заранее не делится между очередями, а очереди двигаются друг за другом, начиная с некоторого начального места в памяти. При включении элемента в пустую очередь ее можно пустить с середины свободного участка памяти.

Обозначим, x, y – текущие длины очередей, z – расстояние от головы первой очереди до хвоста второй. Потери происходят, если первая очередь догоняет вторую ($z = 0$) или вторая догоняет первую ($x + y + z = n$).

Определим схему переходов между состояниями. Пусть (x, y, z) – текущее состояние процесса. Тогда:

$$(x, y, z) \xrightarrow{r} (x, y, z)$$

$$(x, y, z) \xrightarrow{p_1} \begin{cases} (x+1, y, z-1) & x, y, z > 0 \\ (x, y, z) & z = 0, y \neq 0, x = n \\ (1, y, \frac{n-y}{2}) & x = 0 \\ (x+1, 0, 0) & y = 0, x < n \end{cases}$$

$$(x, y, z) \xrightarrow{p_2} \begin{cases} (x, y+1, z) & x, y, z > 0 \\ (x, y, z) & x+y+z = n \\ (x, 1, \frac{n-x}{2}) & y = 0, x < n \\ (0, y+1, 0) & x = 0, y < n \end{cases}$$

$$(x, y, z) \xrightarrow{q} \begin{cases} (x-c_1, y-c_2, z+c_2) & x > c_1, y > c_2 \\ (x-c_1-(c_2-y), 0, 0) & x > c_1+c_2-y, y \leq c_2 \\ (0, y-c_2-(c_1-x)) & x \leq c_1, y > c_1+c_2-x \\ (0, 0, 0) & \text{иначе} \end{cases}$$

В качестве имитационной модели рассмотрим случайное блуждание по кругу по часовой стрелке, при этом:

1. с вероятностью p_1 увеличивается на 1 размер первой очереди, уменьшается на 1 расстояние между очередями;
2. с вероятностью p_2 увеличивается на 1 размер второй очереди;
3. с вероятностью q уменьшается на c_1 размер первой очереди и на c_2 размер второй очереди, увеличивается на c_2 расстояние между очередями;
4. с вероятностью r процесс остается на месте.

С помощью датчика случайных чисел генерируется последовательность для случайного блуждания и подсчитывается количество находжений в состояниях потери.

Таблица 1: Сравнение вероятностей потерь для математической и имитационной модели

q	p_1	p_2	r	c_1	c_2	P_{loss} (матем.)	P_{loss} (имитац.)
0.1	0.2	0.4	0.3	2	6	0.017181547	0.017416110
0.1	0.2	0.5	0.2	1	7	0.046325628	0.046472030
0.1	0.3	0.2	0.4	6	2	0.004156068	0.004443540
0.1	0.3	0.3	0.3	4	4	0.014469843	0.014889290
0.1	0.3	0.4	0.2	3	5	0.041542893	0.041802270
0.1	0.3	0.5	0.1	3	5	0.089529228	0.089206010
0.1	0.4	0.1	0.4	7	1	0.009562229	0.009967280
0.1	0.4	0.3	0.2	5	3	0.041542893	0.041850260
0.2	0.2	0.4	0.2	1	7	0.000169100	0.000177550
0.2	0.3	0.4	0.1	3	5	0.000337448	0.000322020
0.3	0.1	0.2	0.4	1	7	0.000000002	0.000000003
0.3	0.3	0.2	0.2	7	1	0.000000318	0.000000370
0.3	0.3	0.3	0.1	4	4	0.000000921	0.000000850

5. Некоторые примеры численного анализа

В таблице 1 приведены значения вероятностей потерь для математической и имитационной моделей последовательного циклического представления при оптимальных значениях c_1, c_2 . Так как аналитическое решение не было получено, нам нужно решать задачу для отдельных n, v, t . В данном примере используются значения $n = 40, v = 8, t = 1$. Используемые здесь вероятности являются теоретическими – для большей наглядности результатов. На практике, эти вероятности могут быть получены в результате предварительных статистических исследований.

Исходя из численного анализа можно отметить, что из очереди с большим значением вероятности прихода p_i во время цикла просмотра оптимально исключать больше элементов (т. е. соответствующее значение c_i большее). Кроме того, с увеличением вероятности исключения увеличивается разница между значениями c_i , поскольку вероятность потери в очереди с меньшей вероятностью прихода снизится за счет более частого исключения элементов. Эти выводы совпадают с интуитивными предположениями, результаты подтверждаются имитационными экспериментами.

Таблица 2: Сравнение вероятностей потерь для последовательного представления в памяти и представления друг за другом

q	p_1	p_2	r	c_1	c_2	P_{loss} (послед.)	P_{loss} (др. за др.)
0.1	0.2	0.4	0.3	2	6	0.017416110	0.016987100
0.1	0.2	0.5	0.2	1	7	0.046472030	0.045772260
0.1	0.3	0.2	0.4	6	2	0.004443540	0.004149240
0.1	0.3	0.3	0.3	4	4	0.014889290	0.014531450
0.1	0.3	0.4	0.2	3	5	0.041802270	0.041109970
0.1	0.3	0.5	0.1	3	5	0.089206010	0.088837330
0.1	0.4	0.1	0.4	7	1	0.009967280	0.008331470
0.1	0.4	0.3	0.2	5	3	0.041850260	0.041156220
0.2	0.2	0.4	0.2	1	7	0.000177550	0.000147900
0.2	0.3	0.4	0.1	3	5	0.000322020	0.000285870
0.3	0.1	0.2	0.4	1	7	0.000000003	0.000000001
0.3	0.3	0.2	0.2	7	1	0.000000370	0.000000360
0.3	0.3	0.3	0.1	4	4	0.000000850	0.000000760

В таблице 2 приведены значения вероятностей потерь для имитационных моделей последовательного циклического представления и представления друг за другом.

Исходя из численного анализа можно сделать вывод, что при представлении очередей друг за другом вероятность потери может быть несколько ниже, чем при циклическом последовательном, но разница несущественная.

6. Заключение

В работе предложена математическая модель, описывающая процесс работы с настраиваемой очередью, представленной в виде двух очередей в случае их последовательного циклического представления, а также имитационные модели, описывающие процесс работы с настраиваемой очередью, представленной в виде двух очередей в случае их последовательного циклического представления и представления друг за другом. Приводятся результаты численных экспериментов по сравнению двух методов представления настраиваемой очереди.

Список литературы

- [1] *Cisco Systems* Cisco IOS 12.0 Quality of Service. Cisco Press. 1999.
- [2] *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. СПб.: Питер, 2010.
- [3] *Танненбаум Э., Уэзеролл В.* Компьютерные сети (5-е издание). СПб.: Питер, 2012. 960 с.
- [4] *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.
- [5] *Драц А.В., Соколов А.В.* Математический анализ процесса работы с M FIFO-очередями // Стохастическая оптимизация в информатике. 2012. Т. 8. № 2. С. 75–82.
- [6] *Барковский Е.А., Соколов А.В.* Оптимальное управление двумя параллельными FIFO-очередями на бесконечном времени // Информационно-управляющие системы. 2015. № 5. С. 65–71.
- [7] *Aksenova E.A., Sokolov A.V.* The Optimal Implementation of Two FIFO-Queues in Single-Level Memory // Applied Mathematics. 2011. Vol. 2. P. 1297–1302.
- [8] *Sokolov A.V., Drac A.V.* The Linked List Representation of n LIFO-Stacks and/or FIFO-Queues in the Single-Level Memory // Information Processing Letters. 2013. Vol. 13. P. 832–835.
- [9] *Сазонов А.М., Соколов А.В.* Математическая модель оптимального управления настраиваемой очередью из двух последовательных циклических FIFO-очереди в общей памяти // Информационно-управляющие системы. 2017. № 4. С. 44–50. DOI: 10.15217/issn1684-8853.2017.4.44
- [10] *Соколов А.В.* Математические модели и алгоритмы оптимального управления динамическими структурами данных. Петрозаводск: ПетрГУ, 2002. 216 с.
- [11] *Барковский Е.А., Соколов А.В.* Модель управления двумя параллельными FIFO-очередями, двигающимися друг за другом в общей памяти // Информационно-управляющие системы. 2016.

№ 1(80). С. 65–73. DOI: 10.15217/issn1684-8853.2016.1.65 (РИНЦ),
<http://www.i-us.ru/article1034>

- [12] *Барковский Е.А.* Оптимальное управление двумя очередями, работающими по принципу настраиваемых очередей // Стохастическая оптимизация в информатике. 2016. №2 С .1–14
- [13] *Кемени Дж., Снелл Дж.* Конечные цепи Маркова. Наука. 1960.