

# Оптимальное разбиение общей памяти для двух параллельных FIFO-очередей<sup>1</sup>

А. В. Соколов д. ф.-м. н., Е. А. Барковский<sup>2</sup>  
ПетрГУ, ИПМИ КарНЦ РАН, Петрозаводск  
*avs@krc.karelia.ru, barkevgen@gmail.com*

---

В работе предлагается математическая модель и решается задача оптимального разбиения общей памяти для двух FIFO-очередей в случае их последовательного циклического представления, в предположении, что на нечетном шаге допускаются операции включения элементов в одну из очередей с заданными вероятностями, а на четном шаге — операции исключения элементов из очередей. Наряду с последовательным, возможно и параллельное выполнение операций с очередями с заданными вероятностями.

*Ключевые слова:* FIFO-очереди, динамические структуры данных, математическое моделирование.

## 1. Введение

В многих приложениях, например, при разработке различных сетевых устройств и встроенных операционных систем требуется работа с несколькими FIFO-очередями, расположенными в общем пространстве памяти. Механизм страничной виртуальной памяти здесь не используется, и вся работа происходит в нескольких пулах оперативной памяти. Количество очередей в таких устройствах может достигать нескольких сотен и тысяч, а в будущем, по экспертным оценкам, может достигнуть нескольких миллионов [1]. Для представления FIFO-очередей применяют различные программные или аппаратные решения [2–4].

В [5–7] предлагались модели для последовательного, связанного и страничного способов представления нескольких FIFO-очередей в памяти одного уровня. В этих моделях предполагается, что на каждом шаге дискретного времени с заданными вероятностями происходят некоторые операции со структурами данных. В работе [2] приведены результаты имитационных экспериментов и поставлена задача построить математическую модель процесса работы с несколькими FIFO-очередями в общей памяти, когда операции с

---

<sup>1</sup>Работа выполнена при финансовой поддержке РФФИ (грант №12-01-00253) и Программы стратегического развития ПетрГУ в рамках реализации комплекса мероприятий по развитию научно-исследовательской деятельности.

<sup>2</sup>©А. В. Соколов, Е. А. Барковский, 2013

очередями выполняются по несколько другому принципу. В данной схеме работы на нечетном шаге допускаются операции включения элементов в одну из  $n$  очередей с равными вероятностями, а на четном шаге — операции исключения элементов из очередей с равными вероятностями. Исключение из пустой очереди не приводит к завершению работы. В [2] ставилась задача определить вероятность (как функцию от  $n$  и  $j$ ) того, что очередь, выбранная для операции на  $j$  том шаге, будет пустой, а также вычислить математическое ожидание количества элементов в очередях после  $j$  операций. В данной задаче не рассматривался конкретный способ представления очередей в памяти, то есть предполагалось, что очереди могут быть неограниченной длины, что на практике невыполнимо. Эта задача была решена в [8].

В [9] предложена математическая модель этого процесса для числа очередей  $n = 2$ , и решается задача оптимального разбиения общей памяти для двух FIFO-очередей в случае их последовательного циклического представления в предположении, что операции с очередями выполняются по этому принципу, но с неравными вероятностями.

В [10] построены математическая и имитационная модели процесса работы с двумя очередями, когда они двигаются по кругу друг за другом. Этот метод работы с FIFO-очередями предложен в [11].

В данной работе мы предлагаем математическую модель и решаем задачу оптимального разбиения общей памяти для двух FIFO-очередей в случае последовательного циклического представления очередей, в предположении, что операции с очередями выполняются по этому принципу, но возможно, наряду с последовательным, и параллельное выполнение операций с очередями с заданными вероятностями.

В случае многоядерных архитектур возможна параллельная работа нескольких потоков со структурой данных (стек, очередь, список, хеш-таблица и т.д.). Для синхронизации такой работы были предложены различные механизмы [12]. Существует ряд программных систем (Intel TBB, Microsoft TPL и др.), где реализована параллельная работа с некоторыми базовыми структурами данных.

В данной работе мы рассматриваем возможность параллельной работы сразу с несколькими структурами данных в общей памяти. При реализации на практике работы с несколькими структура-

ми данных будут использованы какие-то общепринятые механизмы синхронизации, которые не влияют на динамику изменения длин структур данных и, следовательно, на наши модели.

## 2. Математическая модель

Пусть в памяти размером в  $m$  единиц мы работаем с двумя последовательными циклическими FIFO-очередями с элементами фиксированного размера в одну условную единицу.

Для последовательного представления каждой очереди выделим некоторое количество единиц памяти из общего объема равного  $m$  единиц. Пусть  $s$  — количество единиц памяти, выделенных первой очереди, тогда  $(m - s)$  — количество единиц памяти, выделенных второй очереди.

Операции, производимые с очередями, выполняются по следующей схеме: на нечетном шаге происходит операция включения элемента в одну из очередей, на четном шаге — операция исключения элемента из какой-либо очереди, причем известны некоторые вероятностные характеристики операций, производимых с очередями. Пусть  $p_1$  и  $p_2$  — вероятности включения элемента в первую и вторую очередь, соответственно,  $p_{12}$  — вероятность одновременного включения в обе очереди.  $q_1$  и  $q_2$  — вероятности исключения элемента из первой и второй очередей, соответственно,  $q_{12}$  — вероятность одновременного исключения из обеих очередей.

Поскольку построенная на основе такой постановки задачи марковская цепь не будет регулярной и однородной, два последовательных шага объединяют в один, а также вводим в рассмотрение вероятности выполнения операций, не изменяющих длины очередей, например, операции чтения:  $r_1$  — вероятность выполнения операции на нечетном шаге и  $r_2$  — на четном, при этом  $r_1 \neq 0$ ,  $r_2 \neq 0$ . Соответственно,  $p_1 + p_2 + p_{12} + r_1 = 1$ ,  $q_1 + q_2 + q_{12} + r_2 = 1$ .

Тогда состояния на каждом шаге определяются наступлением одной из следующих комбинаций событий:

- 1) включение в первую, исключение из второй очереди с вероятностью  $p_1 q_2$ ;
- 2) включение во вторую, исключение из первой очереди с вероятностью  $p_2 q_1$ ;
- 3) включение в первую очередь с вероятностью  $p_1 r_2 + p_{12} q_2$ ;

- 4) включение во вторую очередь с вероятностью  $p_2r_2 + p_{12}q_1$ ;
- 5) включение параллельно в обе очереди с вероятностью  $p_{12}r_2$ ;
- 6) исключение из первой очереди с вероятностью  $q_1r_1 + p_2q_{12}$ ;
- 7) исключение из второй очереди с вероятностью  $q_2r_1 + p_1q_{12}$ ;
- 8) исключение параллельно из обеих очередей с вероятностью  $q_{12}r_1$ ;
- 9) выполнение над очередями сохраняющих их состояние противоположных операций с вероятностью  $r_1r_2 + p_1q_1 + p_2q_2 + p_{12}q_{12}$ , где  $p_1q_2 + p_2q_1 + p_1r_2 + p_{12}q_2 + p_2r_2 + p_{12}q_1 + p_{12}r_2 + q_1r_1 + p_2q_{12} + q_2r_1 + p_1q_{12} + q_{12}r_1 + r_1r_2 + p_1q_1 + p_2q_2 + p_{12}q_{12} = 1$ .

Целью исследования является определение оптимального распределения памяти между очередями, когда в качестве критерия оптимальности рассматривается минимальная средняя доля потерянных при переполнении элементов очередей. По закону больших чисел для регулярных цепей Маркова [13] это эквивалентно нахождению решения, доставляющего минимум значению вероятности переполнения памяти на бесконечном промежутке времени.

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  $0 \leq x \leq s + 2$ ,  $0 \leq y \leq m - s + 2$  (рис. 1), где прямые  $x = s + 1$ ,  $y = m - s + 1$  образуют первый отражающий экран — попадая на эти прямые, мы будем находиться на них до тех пор, пока не произойдет исключение элемента из очереди; прямые  $x = s + 2$ ,  $y = m - s + 2$  образуют второй отражающий экран, который определен для случаев включения элемента в заполненную очередь и немедленного исключения элемента из этой же очереди. Введением данного экрана учитывается произошедшая потеря элемента, очередь формально переходит на экран, а фактически в область  $0 \leq x < s$ ,  $0 \leq y < m - s$ , конкретно на прямые  $x = s - 1$  или  $y = m - s - 1$ .

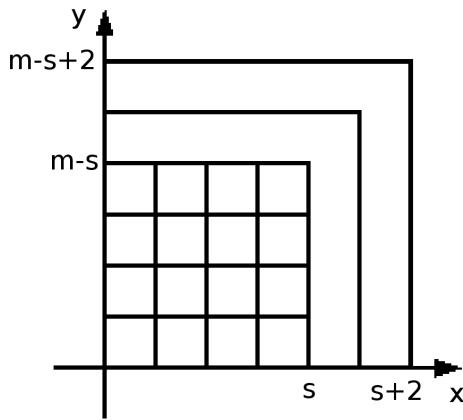


Рис. 1: Область блуждания

Определим схему переходов между состояниями. Пусть  $(x, y)$  — текущее состояние процесса, тогда блуждание в области  $0 \leq x \leq s + 1, 0 \leq y \leq m - s + 1$  (включает в себя и первый отражающий экран) можно описать следующим образом:

$$(x, y) \xrightarrow{p_1 q_2} (x', y') = \begin{cases} (x, y), & x = s + 1, y = 0 \\ (x, y - 1), & x = s + 1, 0 < y \leq m - s \\ (x, y - 2), & x = s + 1, y = m - s + 1 \\ (x + 1, y), & 0 \leq x \leq s, y = 0 \\ (x + 1, y - 1), & 0 \leq x \leq s, 0 < y \leq m - s \\ (x + 1, y - 2), & 0 \leq x \leq s, y = m - s + 1 \end{cases}$$

$$(x, y) \xrightarrow{p_2 q_1} (x', y') = \begin{cases} (x, y), & x = 0, y = m - s + 1 \\ (x, y + 1), & x = 0, 0 \leq y \leq m - s \\ (x - 1, y), & 0 \leq x \leq s, y = m - s + 1 \\ (x - 1, y + 1), & 0 \leq x \leq s, 0 \leq y \leq m - s \\ (x - 2, y), & x = s + 1, y = m - s + 1 \\ (x - 2, y + 1), & x = s + 1, 0 \leq y \leq m - s \end{cases}$$

$$(x, y) \xrightarrow{p_1 r_2} (x', y') = \begin{cases} (x, y), & x = s + 1, 0 \leq y \leq m - s \\ (x, y - 1), & x = s + 1, y = m - s + 1 \\ (x + 1, y), & 0 \leq x \leq s, 0 \leq y \leq m - s \\ (x + 1, y - 1), & 0 \leq x \leq s, y = m - s + 1 \end{cases}$$

$$(x, y) \xrightarrow{p_{12} q_2} (x', y') = \begin{cases} (x, y), & x = s + 1, 0 \leq y < m - s \\ (x, y + 1), & x = s + 1, y = m - s + 1 \\ (x, y + 2), & x = s + 1, y = m - s \\ (x + 1, y), & 0 \leq x \leq s, 0 \leq y < m - s \\ (x + 1, y + 1), & 0 \leq x \leq s, y = m - s + 1 \\ (x + 1, y + 2), & 0 \leq x \leq s, y = m - s \end{cases}$$

$$(x, y) \xrightarrow{p_2 r_2} (x', y') = \begin{cases} (x, y), & 0 \leq x \leq s, y = m - s + 1 \\ (x, y + 1), & 0 \leq x \leq s, 0 \leq y \leq m - s \\ (x - 1, y), & x = s + 1, y = m - s + 1 \\ (x - 1, y + 1), & x = s + 1, 0 \leq y \leq m - s \end{cases}$$

$$(x, y) \xrightarrow{p_{12} q_1} (x', y') = \begin{cases} (x, y), & 0 \leq x < s, y = m - s + 1 \\ (x, y + 1), & 0 \leq x < s, 0 \leq y \leq m - s \\ (x + 1, y), & x = s + 1, y = m - s + 1 \\ (x + 1, y + 1), & x = s + 1, 0 \leq y \leq m - s \\ (x + 2, y), & x = s, y = m - s + 1 \\ (x + 2, y + 1), & x = s, 0 \leq y \leq m - s \end{cases}$$

$$(x, y) \xrightarrow{p_{12} r_2} (x', y') = \begin{cases} (x, y), & x = s + 1, y = m - s + 1 \\ (x, y + 1), & x = s + 1, 0 \leq y \leq m - s \\ (x + 1, y), & 0 \leq x \leq s, y = m - s + 1 \\ (x + 1, y + 1), & 0 \leq x \leq s, 0 \leq y \leq m - s \end{cases}$$

$$(x, y) \xrightarrow{q_1 r_1} (x', y') = \begin{cases} (x, y), & x = 0, 0 \leq y \leq m - s \\ (x, y - 1), & x = 0, y = m - s + 1 \\ (x - 1, y), & 0 < x \leq s, 0 \leq y \leq m - s \\ (x - 1, y - 1), & 0 < x \leq s, y = m - s + 1 \\ (x - 2, y), & x = s + 1, 0 \leq y \leq m - s \\ (x - 2, y - 1), & x = s + 1, y = m - s + 1 \end{cases}$$

$$(x, y) \xrightarrow{p_2 q_{12}} (x', y') = \begin{cases} (x, y), & x = 0, 0 \leq y < m - s \\ (x, y + 1), & x = 0, y = m - s + 1 \\ (x, y + 2), & x = 0, y = m - s \\ (x - 1, y), & 0 < x \leq s, 0 \leq y < m - s \\ (x - 1, y + 1), & 0 < x \leq s, y = m - s + 1 \\ (x - 1, y + 2), & 0 < x \leq s, y = m - s \\ (x - 2, y), & x = s + 1, 0 \leq y < m - s \\ (x - 2, y + 1), & x = s + 1, y = m - s + 1 \\ (x - 2, y + 2), & x = s + 1, y = m - s \end{cases}$$

$$(x, y) \xrightarrow{q_2 p_1} (x', y') = \begin{cases} (x, y), & 0 \leq x \leq s, y = 0 \\ (x, y - 1), & 0 \leq x \leq s, 0 < y \leq m - s \\ (x, y - 2), & 0 \leq x \leq s, y = m - s + 1 \\ (x - 1, y), & x = s + 1, y = 0 \\ (x - 1, y - 1), & x = s + 1, 0 < y \leq m - s \\ (x - 1, y - 2), & x = s + 1, y = m - s + 1 \end{cases}$$

$$(x, y) \xrightarrow{p_1 q_{12}} (x', y') = \begin{cases} (x, y), & 0 \leq x < s, y = 0 \\ (x, y - 1), & 0 \leq x < s, 0 < y \leq m - s \\ (x, y - 2), & 0 \leq x < s, y = m - s + 1 \\ (x + 1, y), & x = s + 1, y = 0 \\ (x + 1, y - 1), & x = s + 1, 0 < y \leq m - s \\ (x + 1, y - 2), & x = s + 1, y = m - s + 1 \\ (x + 2, y), & x = s, y = 0 \\ (x + 2, y - 1), & x = s, 0 < y \leq m - s \\ (x + 2, y - 2), & x = s, y = m - s + 1 \end{cases}$$

$$(x, y) \xrightarrow{q_{12} p_1} (x', y') = \begin{cases} (x, y), & x = 0, y = 0 \\ (x, y - 1), & x = 0, 0 \leq y \leq m - s \\ (x, y - 2), & x = 0, y = m - s + 1 \\ (x - 1, y), & 0 \leq x \leq s, y = 0 \\ (x - 1, y - 1), & 0 \leq x \leq s, 0 \leq y \leq m - s \\ (x - 1, y - 2), & 0 \leq x \leq s, y = m - s + 1 \\ (x - 2, y), & x = s + 1, y = 0 \\ (x - 2, y - 1), & x = s + 1, 0 \leq y \leq m - s \\ (x - 2, y - 2), & x = s + 1, y = m - s + 1 \end{cases}$$

$$(x, y) \xrightarrow{r_1 r_2} (x', y') = \begin{cases} (x, y), & 0 \leq x \leq s, 0 \leq y \leq m-s \\ (x, y-1), & 0 \leq x \leq s, y = m-s+1 \\ (x-1, y), & x = s+1, 0 \leq y \leq m-s \\ (x-1, y-1), & x = s+1, y = m-s+1 \end{cases}$$

$$(x, y) \xrightarrow{p_1 q_1} (x', y') = \begin{cases} (x, y), & 0 \leq x < s, 0 \leq y \leq m-s \\ (x, y-1), & 0 \leq x < s, y = m-s+1 \\ (x+1, y), & x = s+1, 0 \leq y \leq m-s \\ (x+1, y-1), & x = s+1, y = m-s+1 \\ (x+2, y), & x = s, 0 \leq y \leq m-s \\ (x+2, y-1), & x = s, y = m-s+1 \end{cases}$$

$$(x, y) \xrightarrow{p_2 q_2} (x', y') = \begin{cases} (x, y), & 0 \leq x \leq s, 0 \leq y < m-s \\ (x, y+1), & 0 \leq x \leq s, y = m-s+1 \\ (x-1, y), & x = s+1, 0 \leq y < m-s \\ (x-1, y+1), & x = s+1, y = m-s+1 \\ (x, y+2), & 0 \leq x \leq s, y = m-s \\ (x-1, y+2), & x = s+1, y = m-s \end{cases}$$

$$(x, y) \xrightarrow{p_{12} q_{12}} (x', y') = \begin{cases} (x, y), & 0 \leq x < s, 0 \leq y < m-s \\ (x, y+1), & 0 \leq x < s, y = m-s+1 \\ (x, y+2), & 0 \leq x < s, y = m-s \\ (x+1, y), & x = s+1, 0 \leq y < m-s \\ (x+1, y+1), & x = s+1, y = m-s+1 \\ (x+1, y+2), & x = s+1, y = m-s \\ (x+2, y), & x = s, 0 \leq y < m-s \\ (x+2, y+1), & x = s, y = m-s+1 \\ (x+2, y+2), & x = s, y = m-s \end{cases}$$

Переходы со второго отражающего экрана (рис. 2):

- 1) при  $y = m - s + 2$  переходы из состояния  $(x, y)$  соответствуют переходам из состояния  $(x', y')$ , где  $x' = x$ ,  $y' = y - 3$ ;
- 2) при  $x = s + 2$  переходы из состояния  $(x, y)$  соответствуют переходам из состояния  $(x', y')$ , где  $x' = x - 3$ ,  $y' = y$ ;
- 3) при  $x = s + 2$  и  $y = m - s + 2$  переходы из состояния  $(x, y)$  соответствуют переходам из состояния  $(x', y')$ , где  $x' = x - 3$ ,  $y' = y - 3$ .

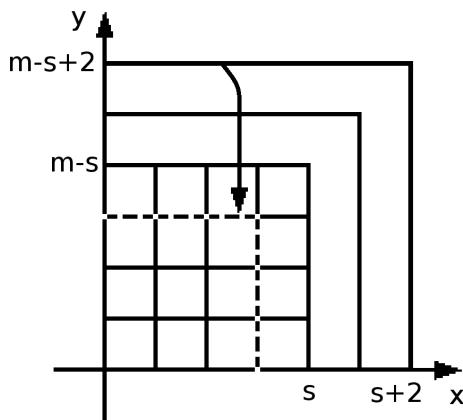


Рис. 2: Состояния в области блуждания, соответствующие второму экрану

Необходимо минимизировать число потерянных элементов при переполнении очередей. Иначе говоря, нужно найти такое  $s$ , чтобы математическое ожидание доли времени, которое процесс проводит на отражающих экранах, было минимальным. Для решения данной задачи используются результаты теории регулярных цепей Маркова.

### 3. Некоторые примеры численного анализа

В таблице 1 приведены некоторые результаты вычислений (указанные результаты были подтверждены имитационными экспериментами). Взятые здесь вероятности являются теоретическими — для большей наглядности результатов. На практике же, эти вероятности должны быть получены в результате предварительных статистических исследований.

Таблица 1

Входные данные	Величина потерь при переполнении ( $m = 12$ )	
	Оптимальное разбиение	Разбиение пополам ( $s = 6$ )
$p_1 = 0.25, p_2 = 0.25, p_{12} = 0.25,$ $r_1 = 0.25$ $q_1 = 0.25, q_2 = 0.25, q_{12} = 0.25$ $r_2 = 0.25$	0.075 ( $s = 6$ )	0.075
$p_1 = 0.35, p_2 = 0.20, p_{12} = 0.10,$ $r_1 = 0.35$ $q_1 = 0.25, q_2 = 0.25, q_{12} = 0.25$ $r_2 = 0.25$	0.015 ( $s = 8$ )	0.019
$p_1 = 0.40, p_2 = 0.10, p_{12} = 0.10,$ $r_1 = 0.40$ $q_1 = 0.25, q_2 = 0.25, q_{12} = 0.25$ $r_2 = 0.25$	0.028 ( $s = 9$ )	0.038
$p_1 = 0.425, p_2 = 0.05, p_{12} = 0.10,$ $r_1 = 0.425$ $q_1 = 0.25, q_2 = 0.25, q_{12} = 0.25$ $r_2 = 0.25$	0.041 ( $s = 9$ )	0.052

Анализируя результаты, можно сказать, что с увеличением вероятности включения в одну очередь (в данном случае — в первую) и уменьшением вероятности включения в другую — при разбиении памяти пополам потери при переполнении увеличиваются. При оптимальном разбиении мы выделяем одной очереди  $s$  единиц памяти, а второй —  $m - s$  единиц памяти. Так, при вероятностях включения: 0.35 в одну очередь и 0.2 в другую, разница потерь, между оптимальным разбиением и разбиением пополам, составляет 0.004. То есть из тысячи элементов (пакетов в случае сетевых приложений FIFO-очередей) мы теряем на 4 элемента меньше, разбивая память оптимально. А при вероятностях включения 0.4 и 0.1 соответственно — уже на 10 элементов меньше.

Рассмотрим влияние параллельности на переполнение очередей. На рисунке 3 приведен график зависимости потерь от вероятностей исключения из одной ( $q(1)$ ) и из обеих очередей ( $q(2)$ ). В первом случае:  $q = 1 - r_2$ , где  $q_1 = q_2 = \frac{q}{2}, q_{12} = 0$ ; во втором:  $q = q_{12} = 1 - r_2$ ,  $q_1 = q_2 = 0$ . То есть, в первом случае вероятность исключения ( $q$ ) делится поровну между очередями, а во втором с этой вероятностью происходит одновременное исключение из очередей. В обеих ситуациях  $p_1 = p_2 = p_{12} = r_1 = 0.25$ .

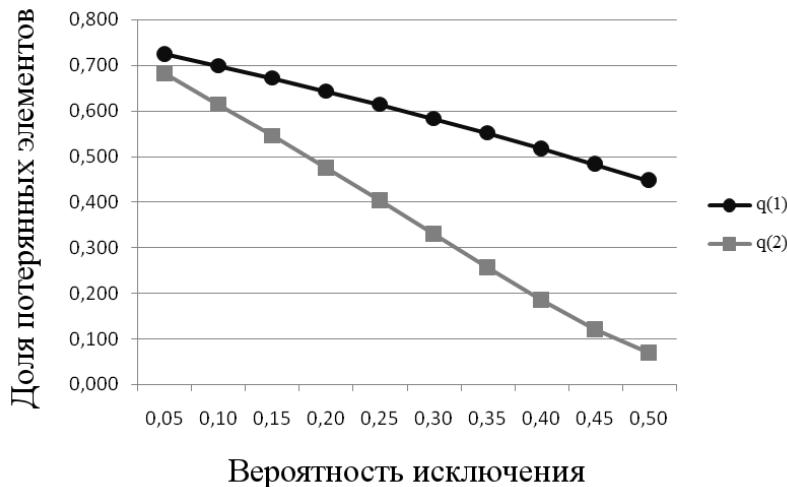


Рис. 3: График зависимости доли потерянных элементов от вероятности исключения из одной и из обеих очередей

На основе построенного графика можно сделать вывод, что исключение параллельно из обеих очередей уменьшает долю потерянных элементов. Для примера: при вероятности исключения 0.05 доля потерянных элементов для обеих очередей на 0.042 меньше, чем для одной, а для вероятности 0.50 — на 0.378.

Противоположная ситуация показана на графике зависимости потерь от вероятностей включения в одну ( $p(1)$ ) и обе очереди ( $p(2)$ ) (рис. 4). Здесь,  $p = 1 - r_1$ , где  $p_1 = p_2 = \frac{p}{2}$ ,  $p_{12} = 0$ ;  $p = p_{12} = 1 - r_1$ , где  $r_1 = r_2 = 0$ . И в том и в другом случае,  $q_1 = q_2 = q_{12} = r_2 = 0.25$ .

Анализируя график можно сказать, что включение параллельно в обе очереди заметно увеличивает рост доли потерянных элементов. Так, при вероятности включения 0.15 разница в доле потерянных элементов для одной и обеих очередей составляет 0.000342. При вероятности в 0.30 — 0.036979, а при 0.45 потеря элементов при включении в обе очереди уже на 0.212665 больше, чем при включении в одну очередь.

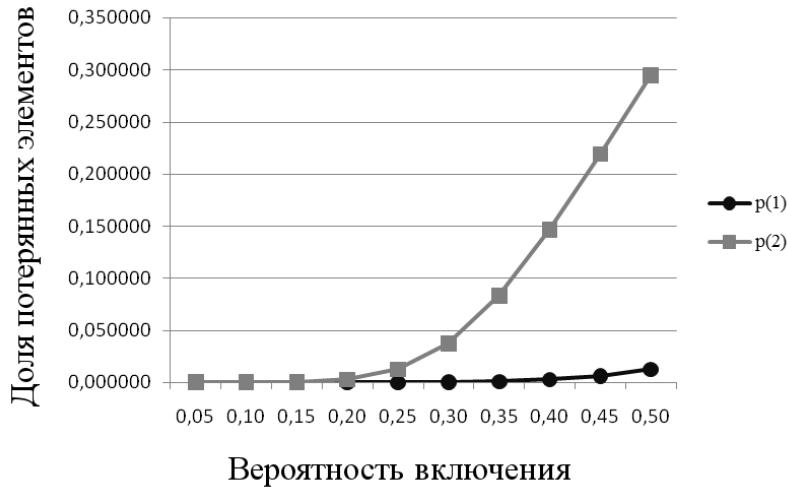


Рис. 4: График зависимости доли потерянных элементов от вероятности включения в одну и обе очереди

#### 4. Заключение

В работе предложена математическая модель, описывающая процесс работы с двумя параллельными циклическими FIFO-очередями в общей памяти, когда на нечетном шаге выполняется операции включения элементов в очередь, а на четном — исключения. Разработан алгоритм и программа нахождения оптимального разбиения памяти между очередями в зависимости от вероятностных характеристик очередей. Для решения поставленной задачи использовался аппарат управляемых случайных блужданий, регулярных цепей Маркова, система Intel®Math Kernel Library PARDISO. Вычисления производились с помощью кластера КарНЦ РАН.

Авторы планируют построить математические модели и алгоритмы оптимального управления для нового способа представления, предложенного ранее [11]. В этом методе память заранее не делится между очередями, а очереди двигаются по кругу друг за другом, начиная с некоторого начального места в памяти.

Перехват работы (work stealing) становится основным способом балансировки нагрузки динамических многопоточных вычислений

на многоядерной аппаратуре [14]. Поэтому в будущем мы планируем рассмотреть математические модели работы с такими структурами данных как деки с ограниченным входом [4], которые используется для реализации этого механизма [15, 16].

## Список литературы

- [1] *Nikologiannis A., Katevenis M.* Multi Queue Management for Advanced QoS in High-Speed Communication Systems Computer Architecture and VLSI Systems Lab, Institute of Computer Science (ICS) Head, <http://archvlsi.ics.forth.gr/muqpro/queueMgt.html>
- [2] Седжвик Р. Фундаментальные алгоритмы на C++. К.: Диа-софт, 2001. 688 с.
- [3] Боллапрагада В., Мэрфи К., Расс У. Структура операционной системы Cisco IOS. М.: Вильямс, 2002. 208 с.
- [4] Кнут Д. Искусство программирования для ЭВМ. Т.1. М.: Ви-льямс, 2001. 736 с.
- [5] Аксенова Е.А., Драц А.В., Соколов А.В. Оптимальное управление  $n$  FIFO-очередями на бесконечном времени // Информационно-управляющие системы. 2009. № 1 С. 46–54.
- [6] Aksanova E.A., Sokolov A.V. The optimal implementation of two FIFO-queues in single-level memory // Applied Mathematics. 2011. Vol. 2, № 10. P. 1297–1302.
- [7] 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. Elsevier Science Publishing Company Inc. 2013. Vol. 13. № 19–21. P. 832–835.
- [8] 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.
- [9] Каблукова Н.В., Соколов А.В. Оптимальное разбиение общей памяти для двух последовательных циклических FIFO-очередей // Прикладная информатика. 2012. № 40. С. 113–125.

- [10] Каблукова Н.В., Соколов А.В. Математический анализ одного способа представления двух FIFO-очередей в общей памяти // Труды Карельского научного центра РАН. 2013. № 1. С. 46–54.
- [11] Соколов А.В. Математические модели и алгоритмы оптимального управления динамическими структурами данных. Петрозаводск: ПетрГУ, 2002. 216 с.
- [12] Herlihy M., Shavit N. The Art of Multiprocessor Programming. Morgan Kaufmann. 2008. 528 p.
- [13] Кемени Дж., Снелл Дж. Конечные цепи Маркова. М.: Наука, 1960. 272 с.
- [14] Lee I.A. Memory Abstractions for Parallel Programming. Massachusetts Institute of Technology, 2012. 163 p.
- [15] Hundler D., Lev Y., Moir M., Shavit N. A Dynamic-Sized Nonblocking Work Stealing Deque // Distributed Computing. Special issue: DISC 04, 2006. V. 18, № 3. P. 189–207.
- [16] Chase D., Lev Y. Dynamic Circular Work-Stealing Deque // SPAA'05. Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures. 2005. P. 21–28.