The optimal implementation of n FIFO-queues in single-level memory

Prof. A.V. Sokolov[†], A.V. Drac[‡]

[†,‡]Institute of Apply Mathematical Researh,
Karelian Scientific Center of Russian Academic of Science [‡] Petrozavodsk State University, Petrozavodsk

E-mail: avs@krc.karelia.ru, adeon88@mail.ru

Abstract

In this article we research methods of n FIFO-queues allocation in the memory of size m units. The problem of optimal memory partition between queues in the case of consecutive circular implementation and the problem of the analysis of linked list implementation are investigated. As mathematical models we proposed random walks into different areas of n-measured space.

1 Introduction

In many applications there is a problem of allocation of multiple queues in single-level memory. There are two fundamentally different ways of organizing work with dynamic data structures – consecutive and linked list allocation. This paper is the extension of [1]. For queues with sequential presentation all the memory is splitted into several parts and each queue is allocated in separate section of memory. In this case we will have losses of memory when any queue overflows and other queues don't overflow.

The linked list implementation is the second method. In this view the data structure is stored as a list. In this case any number of elements of the lists can coexist in the memory area until the free memory is exhausted. But on the other hand, this approach requires an additional link field for each element.

In this article we considered the system where memory overflow is not an emergency situation. If free memory is exausted then all attempts to include an element into queue will lead to it loss until the appearance of free memory i.e. until the deletion of an element from the tail of queue. Such scheme is used for example in the routers and such behaviour of queues is name ”reset tail”. As the optimality criterion we considered the minimal part of time which the system is situated in the state of reset tail. It is reasonably to minimize this value in order to minimize the part of lost elements. In this paper we considered linked and sequential presentation of queues and calculated in symbolic form the average part of time which the system is situated in the state of ”reset tail”.

2 The problem

Consider n queues in single level memory size m. Assume that the time is discrete and only one of the following operations can happen during each time step:

- insertion of element into j-th data stucture with the probability p_j $(1 \le j \le n)$,
- deletion of element from j-th data stucture with the probability q_j $(1 \le j \le n)$,
- access the element with the probability r (Data structures don't change their lengths).

$p_1 + \cdots + p_n + q_1 + \cdots + q_n + r = 1.$
Values p_j, q_j, r are constants. They don't depend on the current lengths of queues and on the operations on the previous steps. All elements have the same lengths. The length of queue is the number of elements that it contains. Denote i_j is the lengths of data structure with number j.

Consider the memory overflow and transition into the states of ”reset tail” in different cases:

1. Linked list implementation.
   Denote l is the ratio of the size of element to the size of a pointer (for the linked presentation). Denote $m(1 - 1/l) = M$. An overflow will occur (and, hence, the system will move to the state of ”reset tail”) when the queues will occupy all memory (i.e. $i_1 + \cdots + i_n = M$) and an element to any of them will be attempted to include.

2. Consecutive presentation.
   Each queue is allocated in its own part of memory. k_j is the size of allocated memory for j-th queue. An overflow will occur when j-th queue occupy all allocated memory (its length will be equal to k_j)

and an element to this queue will be attempted to include. The free memory allocated to other queues will not be redistributed.

We suppose, that the process starts from the state, when data structures are empty, and in the case of deletion from an empty data structure there is no shutdown. The problem is to find the average part of time when the system is situated in the state of reset tail in both cases of representation and compare them. To solve this problem we used apparatus of regular markov's chains.

3 Consecutive allocation of the queues

3.1 The problem

Consider $k_1, \ldots, k_n$ is the fixed partition of the memory.

As the mathematical model we consider the random walk on an integer lattice space inside n-dimensional parallelepiped with vertex at the origin, edges parallel to the axes and the lengths of edges $k_1, \ldots, k_n$. Number of states is equal to $\prod_{i=1}^{n}(k_i + 1)$. $(i_1, \ldots, i_n)$ is the state of the system. $0 \leq i_1 \leq k_1 + 1, \ldots, 0 \leq i_n \leq k_n + 1$. $i_j = k_j + 1$ are the states of "reset tail". $\alpha_{i_1 \ldots i_n}$ is the limit probability which system is situated in the state $(i_1, \ldots, i_n)$.

Conversion of the system from state $(i_1, \ldots, i_n)$ to state $(i'_1, \ldots, i'_n)$ occurs in according to the following rules (fig. 1):

$$(\ldots, i_s, \ldots, i_t, \ldots) \xrightarrow{p_s} \begin{cases} (\ldots, i_s+1, \ldots, i_t, \ldots) & 0 \leq i_s \leq k_s, i_t \leq k_t \\ (\ldots, i_s+1, \ldots, i_t-1, \ldots) & 0 \leq i_s \leq k_s, i_t = k_t+1 \\ (\ldots, i_s, \ldots, i_t, \ldots) & i_s = k_s+1, i_t \leq k_t \\ (\ldots, i_s, \ldots, i_t-1, \ldots) & i_s = k_s+1, i_t = k_t+1 \end{cases}$$

$$(\ldots, i_s, \ldots, i_t, \ldots) \xrightarrow{q_s} \begin{cases} (\ldots, i_s, \ldots, i_t, \ldots) & i_s = 0, x_j \leq k_t \\ (\ldots, i_s, \ldots, i_t-1, \ldots) & i_s = 0, x_j = k_t+1 \\ (\ldots, i_s-1, \ldots, i_t, \ldots) & 1 \leq i_s \leq k_s, i_t \leq k_t \\ (\ldots, i_s-1, \ldots, i_t-1, \ldots) & 1 \leq i_s \leq k_s, i_t = k_t+1 \\ (\ldots, i_s-2, \ldots, i_t, \ldots) & i_s = k_s+1, i_t \leq k_t \\ (\ldots, i_s-2, \ldots, i_t-1, \ldots) & i_s = k_s+1, i_t = k_t+1 \end{cases}$$

$$(\ldots, i_s, \ldots, i_t, \ldots) \xrightarrow{r} \begin{cases} (\ldots, i_s, \ldots, i_t, \ldots) & 0 \leq i_s \leq k_s, i_t \leq k_t \\ (\ldots, i_s, \ldots, i_t-1, \ldots) & 0 \leq i_s \leq k_s, i_t = k_t+1 \\ (\ldots, i_s-1, \ldots, i_t, \ldots) & i_s = k_s+1, i_t \leq k_t \\ (\ldots, i_s-1, \ldots, i_t-1, \ldots) & i_s = k_s+1, i_t = k_t+1 \end{cases}$$

$1 \leq s \leq n, \quad 1 \leq t \leq n, \quad s \neq t$

Construct the balance equation $\alpha_i = \sum\limits_i \alpha_j P_{ji}$. For our system it will be the following (for internal states):

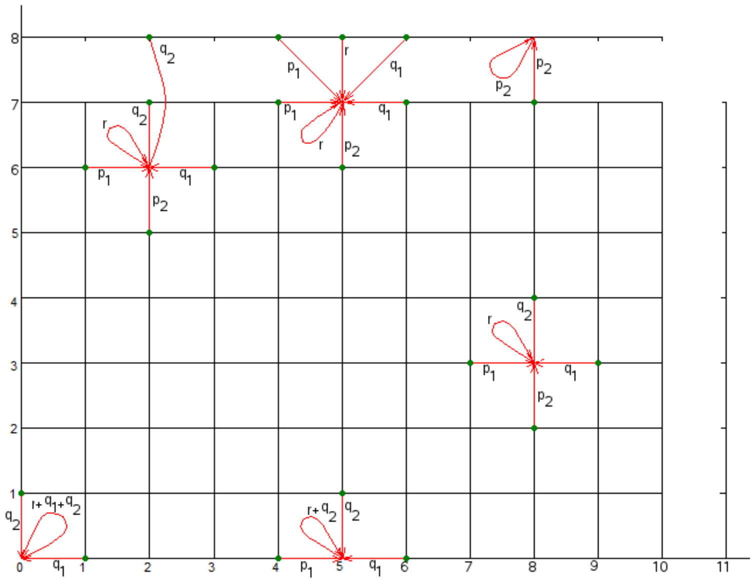

Figure 1: Transition between states in the case of consecutive presentation

1. $\alpha_{i_1 i_2 \dots i_n} = p_1 \alpha_{i_1-1, i_2 \dots i_n} + q_1 \alpha_{i_1+1, i_2 \dots i_n} + \dots + p_n \alpha_{i_1 i_2 \dots i_n - 1} +$
   $q_n \alpha_{i_1 i_2 \dots i_n + 1} + r \alpha_{i_1 i_2 \dots i_n}$
   $(1 \le i_j \le k_j - 2,\ 1 \le j \le n)$

2. $\alpha_{0 i_2 \dots i_n} = q_1 \alpha_{1, i_2 \dots i_n} + \dots + p_n \alpha_{0 i_2 \dots i_n - 1} + q_n \alpha_{0 i_2 \dots i_n + 1} +$
   $(r + q_1) \alpha_{0 i_2 \dots i_n}$
   $(i_1 = 0, 1 \le i_j \le k_j - 2,\ 2 \le j \le n)$

3. $\alpha_{k_1 - 1 i_2 \dots i_n} = p_1 \alpha_{k_1-2, i_2 \dots i_n} + q_1 \alpha_{k_1, i_2 \dots i_n} + q_1 \alpha_{k_1+1, i_2 \dots i_n} + \dots +$
   $p_n \alpha_{i_1 i_2 \dots i_n - 1} + q_n \alpha_{i_1 i_2 \dots i_n + 1} + r \alpha_{k_1 - 1, i_2 \dots i_n}$
   $(i_1 = k_1 - 1, 1 \le i_j \le k_j - 2,\ 2 \le j \le n)$

4. $\alpha_{k_1 i_2 \dots i_n} = p_1 \alpha_{k_1-1, i_2 \dots i_n} + r \alpha_{k_1+1, i_2 \dots i_n} + \dots + p_n \alpha_{i_1 i_2 \dots i_n - 1} +$
   $q_n \alpha_{i_1 i_2 \dots i_n + 1} + r \alpha_{k_1 i_2 \dots i_n}$
   $(i_1 = k_1, 1 \le i_j \le k_j - 2,\ 2 \le j \le n)$

5. $\alpha_{k_1 + 1 i_2 \dots i_n} = p_1 \alpha_{k_1, i_2 \dots i_n} + p_1 \alpha_{k_1+1, i_2 \dots i_n}$
   $(i_1 = k_1, 1 \le i_j \le k_j - 2,\ 2 \le j \le n)$

Split the set of indexes $I = \{1, \dots, n\}$ into the subsets:
$I_1 = \{j : i_j = 0\}$

$I_2 = \{j : 1 \le i_j \le k_j - 2\}$
$I_3 = \{j : i_j = k_j - 1\}$
$I_4 = \{j : i_j = k_j\}$
$I_5 = \{j : i_j = k_j + 1\}$

Since only one of queues can be into the state of "reset tail" then $|I_5| = 0$ or $|I_5| = 1$.

1. For usual states (equations 1–4 in common view, i.e. when $I_5 = \emptyset$):
$\alpha_{i_1 \dots i_n} = (r + \sum_{j \in I_1} q_j)\alpha_{i_1 \dots i_n} + \sum_{j \in I_1} q_j \alpha_{i_1 \dots i_j + 1 \dots i_n} +$
$\sum_{j \in I_2 + I_3} (p_j \alpha_{i_1 \dots i_j - 1 \dots i_n} + q_j \alpha_{i_1 \dots i_j + 1 \dots i_n}) + \sum_{j \in I_3} q_j \alpha_{i_1 \dots i_j + 2 \dots i_n} +$
$\sum_{j \in I_4} \Big(r \alpha_{i_1 \dots i_j + 1 \dots i_n} + p_j \alpha_{i_1 \dots i_j - 1 \dots i_n} +$
$\sum_{l \in I_2 + I_3} (p_l \alpha_{i_1 \dots i_l - 1 \dots i_j + 1 \dots i_n} + q_l \alpha_{i_1 \dots i_l + 1 \dots i_j + 1 \dots i_n}) +$
$\sum_{l \in I_1} q_l (\alpha_{i_1 \dots i_l \dots i_j + 1 \dots i_n} + \alpha_{i_1 \dots i_l + 1 \dots i_j + 1 \dots i_n}) \Big)$

2. For the states of "reset tail" (equation 5):
$\alpha_{i_1 \dots k_j + 1 \dots i_n} = p_j (\alpha_{i_1 \dots k_j \dots i_n} + \alpha_{i_1 \dots k_j + 1 \dots i_n})$

The system has the following solution:

$$\alpha_{i_1 \dots i_n} = C \left(\frac{p_1}{q_1}\right)^{i_1} \dots \left(\frac{p_n}{q_n}\right)^{i_n} \left(1 - \sum_{j \in I_4} p_j\right) \qquad I_5 = \emptyset$$

$$\alpha_{i_1 \dots i_j \dots i_n} = C \left(\frac{p_1}{q_1}\right)^{i_1} \dots \left(\frac{p_j}{q_j}\right)^{i_j - 1} \dots \left(\frac{p_n}{q_n}\right)^{i_n} p_j \qquad I_5 = \{j\}$$

Denote $p_i / q_i = x_i$, $i = 1, \dots, n$. Let for queues with numbers $1, \dots, s$ $x_i \ne 1$, i.e. $p_i \ne q_i$, and for queues with number $s+1, \dots, n$ $x_i = 1$, i.e. $p_i = q_i$. Find the constant C from the normalizatin equation:

$$\begin{aligned} \tfrac{1}{C} &= \sum_{i_1=0}^{k_1} \cdots \sum_{i_n=0}^{k_n} x_1^{i_1} \dots x_n^{k_n} = \sum_{i_1=0}^{k_1} x_1^{i_1} \cdots \sum_{i_n=0}^{k_n} x_n^{k_n} = \\ &= \prod_{i=1}^{s} \frac{x_i^{k_i+1} - 1}{x_i - 1} \prod_{i=s+1}^{n} (k_i + 1) \end{aligned}$$

Summarise all $\alpha_{i_1 \dots i_n}$ which the states of "reset tail". For queue with number 1 it will be the states with the condition $i_1 = k_1$, $0 \le i_j \le i_n$,

$2 \le j \le n$.

$$p_1^* = p_1 x_1^{k_1} \frac{1}{C} \sum_{i_2=0}^{k_2} \cdots \sum_{i_n=0}^{k_n} x_2^{i_2} \dots x_n^{k_n} = p_1 x_1^{k_1} \frac{1}{C} \sum_{i_2=0}^{k_2} x_2^{i_2} \cdots \sum_{i_n=0}^{k_n} x_n^{k_n} =$$
$$p_1 x_1^{k_1} \frac{1}{C} \prod_{i=1}^{s} \frac{x_i^{k_i+1}-1}{x_i-1} \prod_{i=s+1}^{n} (k_i+1) = p_1 x_1^{k_1} \frac{x_1-1}{x_1^{k_1+1}-1} =$$
$$\frac{p_1 \left(\frac{p_1}{q_1}\right)^{k_1} \left(\frac{p_1}{q_1}-1\right)}{\left(\frac{p_1}{q_1}\right)^{k_1+1}-1} = \frac{p_1(p_1^{k_1+1}-p_1^{k_1}q)}{p_1^{k_1+1}-q_1^{k_1+1}} = \frac{p_1-q_1}{1-\left(\frac{q_1}{p_1}\right)^{k_1+1}} = \frac{q_1-p_1}{\left(\frac{q_1}{p_1}\right)^{k_1+1}-1}$$

Similarly for the queues with numbers $2, \dots, s$.

For the queue with number $s+1$:

$$p_{s+1}^* = \frac{p_{s+1}}{k_{s+1}+1}$$

Similarly for the queues with numbers $s+2, \dots, n$.

Then the summary part of time which the system is situated in the state of "reset tail" is equal to:

$$P_c^* = \sum_{i=1}^{n} p_i^* = \sum_{i=1}^{s} \frac{q_i - p_i}{\left(\frac{q_i}{p_i}\right)^{k_i+1} - 1} + \sum_{i=s+1}^{n} \frac{p_i}{k_i+1}$$

The problem of optimal division of memory was solved in [1].

4 Linked list presentation of queues

As the mathematical model we consider the random walk on an integer lattice space inside n-dimensional pyramid with edges $0 \le x_1 \le M$, $0 \le x_2 \le M$, ..., $0 \le x_n \le M$ and base $x_1 + x_2 + \cdots + x_n = M$.

For each state in the face $x_1 + x_2 + \cdots + x_n = M$, i.e. when all the memory is exausted, define the corresponding state of "reset tail". Denote it as $(\bar{x_1}, \dots, \bar{x_n})$. It can be reached in case of inserting of an element into any of queues. Conversion of the system from state $(x_1, \dots, x_n)$ to state $(x_1', \dots, x_n')$ occurs in according to the following rules (fig. 2):

$$(\dots, i_s, \dots, i_t, \dots) \xrightarrow{p_s} \begin{cases} (\dots, i_s+1, \dots, i_t, \dots) & 0 \le i_1 + \cdots + i_n < M \\ (\dots, \bar{x_s}, \dots, \bar{x_j}, \dots) & i_1 + \cdots + i_n = M \end{cases}$$

$$(\dots, i_s, \dots, i_t, \dots) \xrightarrow{q_s} \begin{cases} (\dots, i_s-1, \dots, i_t, \dots) & i_s > 0 \\ (\dots, i_s, \dots, i_t, \dots) & i_s = 0 \end{cases}$$

$$(\dots, i_s, \dots, i_t, \dots) \xrightarrow{r} (\dots, i_s, \dots, i_t, \dots)$$

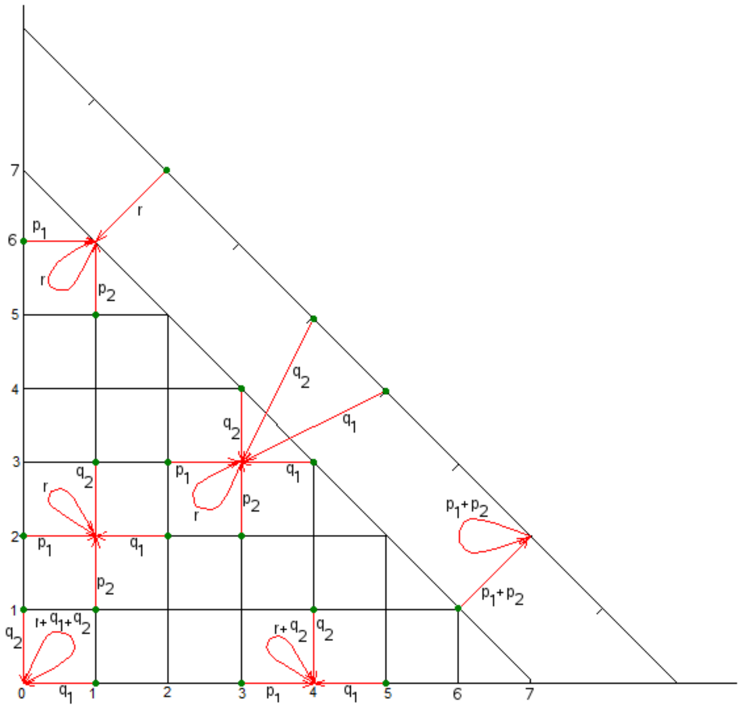

Figure 2: Transition between states in the case of linked list presentation

$$(\ldots,\bar{i_s},\ldots,\bar{i_t},\ldots)\xrightarrow{\quad p_s \quad}(\ldots,\bar{i_s},\ldots,\bar{i_t},\ldots)$$

$$(\ldots,\bar{i_s},\ldots,\bar{i_t},\ldots)\xrightarrow{\quad q_t \quad}\begin{cases}(\ldots,i_s-1,\ldots,i_t,\ldots) & i_s>0\\ (\ldots,i_s,\ldots,i_t,\ldots) & i_s=0\end{cases}$$

$$(\ldots,\bar{i_s},\ldots,\bar{i_t},\ldots)\xrightarrow{\quad r \quad}(\ldots,i_s,\ldots,i_t,\ldots)$$

Split the set of indexes $I=\{1,\ldots,n\}$ into the subsets:
$I_1=\{j:i_j=0\}$
$I_2=\{j:i_j>0\}$
Construct the balance equation $\alpha_i=\sum\limits_i \alpha_j P_{ji}$. For our system it will be the following:

1. $\alpha_{i_1\ldots i_n} = \sum\limits_{j\in I_1}(q_j\alpha_{i_1\ldots i_j\ldots i_n}+q_j\alpha_{i_1\ldots i_j+1\ldots i_n})+\sum\limits_{j\in I_2}(p_j\alpha_{i_1\ldots i_j-1+i_n}+q_j\alpha_{i_1,\ldots i_j+1\ldots i_n})+r\alpha_{i_1\ldots i_n}$
$(0\leq i_1+\cdots+i_n\leq M-2)$

2. $\alpha_{i_1\dots i_n} = \sum\limits_{j\in I_1} (q_j\alpha_{i_1,\dots i_j+i_n} + q_j(\alpha_{i_1,\dots i_j+1\dots i_n} + \overline{\alpha_{i_1\dots i_j\dots i_n}})) +$
$\sum\limits_{j\in I_2} (p_j\alpha_{i_1\dots i_j-1+i_n} + q_j(\alpha_{i_1,\dots i_j+1\dots i_n} + \overline{\alpha_{i_1\dots i_j\dots i_n}})) + r\alpha_{i_1\dots i_n}$
$(i_1 + \dots + i_n = M-1)$

3. $\alpha_{i_1\dots i_n} = p_1\alpha_{i_1-1,i_2\dots i_n} + \dots + p_n\alpha_{i_1 i_2\dots i_n-1} + r(\alpha_{i_1 i_2\dots i_n} + \overline{\alpha_{i_1,i_2\dots i_n}})$
$(i_1 + \dots + i_n = M)$

4. $\overline{\alpha_{i_1\dots i_n}} = (p_1 + \dots + p_n)(\alpha_{i_1 i_2\dots i_n} + \overline{\alpha_{i_1 i_2\dots i_n}})$
$(i_1 + \dots + i_n = M)$

The system has the following solution:

$$\alpha_{i_1 i_2\dots i_n} = C\left(\frac{p_1}{q_1}\right)^{i_1}\dots\left(\frac{p_n}{q_n}\right)^{i_n} \qquad 0 \le i_1 + \dots + i_n \le M-1$$

$$\alpha_{i_1 i_2\dots i_n} = C(1-p_1-\dots-p_n)\left(\frac{p_1}{q_1}\right)^{i_1}\dots\left(\frac{p_n}{q_n}\right)^{i_n} \qquad i_1 + \dots + i_n = M$$

$$\overline{\alpha_{i_1 i_2\dots i_n}} = C(p_1+\dots+p_n)\left(\frac{p_1}{q_1}\right)^{i_1}\dots\left(\frac{p_n}{q_n}\right)^{i_n} \qquad i_1 + \dots + i_n = M$$

Find the constant C from the normalizatin equation:

$$\frac{1}{C} = \sum_{i_n=0}^{M}\sum_{i_{n-1}=0}^{M-i_n}\cdots\sum_{i_1=0}^{M-i_2-\dots-i_n} x_1^{i_1}\dots x_n^{i_n}$$

Find the summary part of time which system situated in the state of "reset tail".

4.1 Statement

Let $a_1,\dots,a_k$ are the distinct numbers, $0 \le s \le k-1$, then:

$$\frac{a_1^s}{(a_1-a_2)(a_1-a_3)\dots(a_1-a_k)} + \frac{a_2^s}{(a_2-a_1)(a_2-a_3)\dots(a_2-a_k)} + \dots +$$
$$+ \frac{a_k^s}{(a_k-a_1)(a_k-a_2)\dots(a_k-a_{k-1})} = 0$$

Proof:

Lead the left part to the common denominator. p-th summand will have $p-1$ changes of sign. All brackets without a_p will be into the p-th summand a_p

$$\frac{1}{\prod\limits_{1\le i<j\le k}(a_i-a_j)}\left(a_1^s\prod_{2\le i<j\le k}(a_i-a_j)+\dots+(-1)^{p+1}a_p^s\prod_{\substack{1\le i<j\le k\\ i\ne p\ne j}}(a_i-a_j)+\right.$$
$$\dots +$$
$$\left.+\,a_k^s\prod_{1\le i<j\le k-1}(a_i-a_j)\right) =$$

Represent the sum in brackets in the form of a determinant. In every summand the procuct will be the value of Vandermond's determinant $(p-1)$-th order.

$$= \begin{vmatrix} a_1^s & a_2^s & a_3^s & \dots & a_k^s \\ 1 & 1 & 1 & \dots & 1 \\ a_1 & a_2 & a_3 & \dots & a_k \\ a_1^2 & a_2^2 & a_3^2 & \dots & a_k^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{k-1} & a_2^{k-1} & a_3^{k-1} & \dots & a_k^{k-1} \end{vmatrix} = 0$$

Determinant is equal to 0, because it has 2 equal rows

4.2 Case of different values of probabilities

Consider the case when $x_i \neq 1 \quad \forall i$ and $x_i \neq x_j$ when $i \neq j$:

$$\frac{1}{C} = \frac{x_1^{M+n}}{(x_1-1)(x_1-x_2)\dots(x_1-x_n)} + \dots + \frac{x_n^{M+n}}{(x_n-1)(x_n-x_1)\dots(x_n-x_{n-1})} + \frac{1}{(1-x_1)\dots(1-x_n)} = f_n(X,M)$$

Method of mathematical induction:

1) Base $n=2$:

$$\sum_{i_2=0}^{M}\sum_{i_1=0}^{M-i_2} x_1^{i_1}x_2^{i_2} = \sum_{i_2=0}^{M} x_2^{i_2}\sum_{i_1=0}^{M-i_2} x_2^{i_2} = \sum_{i_2=0}^{M} x_2^{i_2}\frac{x_1^{M-i_2+1}-1}{x_1-1} =$$

$$= \sum_{i_2=0}^{M}\frac{x_1^{M-i_2+1}x_2^{i_2}}{x_1-1} - \sum_{i_2=0}^{M}\frac{x_2^{i_2}}{x_1-1} = \frac{x_1}{x_1-1}\frac{x_2^{M+1}-x_1^{M+1}}{x_2-x_1} - \frac{x_2^{M+1}-1}{(x_1-1)(x_2-1)} =$$

$$= \frac{x_1^{M+2}}{(x_1-1)(x_1-x_2)} + \frac{1}{(1-x_1)(1-x_2)} - x_2\left(\frac{x_1}{(x_1-1)(x_1-x_2)} + \frac{1}{(1-x_1)(1-x_2)}\right) = \frac{x_1^{M+2}}{(x_1-1)(x_1-x_2)} + \frac{x_2^{M+2}}{(x_2-1)(x_2-x_1)} + \frac{1}{(1-x_1)(1-x_2)}$$

2) Suppose it is true for $(n-1)$-th queue.

3) Proof for n:

$$f_n(X,M) = \sum_{i_n=0}^{M} x_n^{i_n}\sum_{i_{n-1}=0}^{M-i_n}\dots\sum_{i_1=0}^{M-i_2-\dots-i_n} x_1^{i_1}\dots x_{n-1}^{i_{n-1}} = \sum_{j=0}^{n} x_n^j(f_{n-1}(M))^{n-j}$$

Simplify summands:

$$\sum_{j=0}^{M} x_n^j = \frac{x_n^{M+1}-1}{x_n-1}$$

$$\sum_{j=0}^{M} x_n^j x_i^{M-j+n-1} = \frac{x_i^{n-1}(x_n^{M+1})-x_i^{M+1}}{x_n-x_i} = \frac{x_i^{M+n}}{x_i-x_n} - \frac{x_n^{M+1}x_i^{n-1}}{x_i-x_n}$$

$(1 \le i \le n-1)$

Then:

$$f_n(X,M) = \frac{x_1^{M+n}}{(x_1-1)(x_1-x_2)\dots(x_1-x_n)} + \dots + \frac{x_{n-1}^{M+n}}{(x_{n-1}-1)(x_{n-1}-x_1)\dots(x_{n-1}-x_n)} + \frac{1}{(1-x_1)\dots(1-x_n)} - x_n^{M+1}\left(\frac{x_1^{n-1}}{(x_1-1)(x_1-x_2)\dots(x_1-x_n)} + \dots + \frac{x_{n-1}^{n-1}}{(x_{n-1}-1)(x_{n-1}-x_1)\dots(x_{n-1}-x_n)} + \frac{1}{(1-x_1)\dots(1-x_n)}\right) =$$
$$= \frac{x_1^{M+n}}{(x_1-1)(x_1-x_2)\dots(x_1-x_n)} + \dots + \frac{x_n^{M+n}}{(x_n-1)(x_n-x_1)\dots(x_n-x_{n-1})} + \frac{1}{(1-x_1)\dots(1-x_n)}$$

Then $C = \dfrac{1}{f_n(X,M)}$. Summarise $\alpha_{i_1\dots i_n}$, which corresponds to the states of "reset tail" :

$(p_1 + \dots + p_n) \sum\limits_{i_n=0}^{M} \sum\limits_{i_{n-1}=0}^{M-i_n} \dots \sum\limits_{i_2=0}^{M-i_3-\dots-i_n} x_1^{M-i_2-\dots-i_n} x_2^{i_2}\dots x_n^{i_n} = g_n(X,M)$

Proof that

$g_n(X,M) = (p_1 + \dots + p_n)\dfrac{x_1^{M+n-1}}{(x_1-x_2)\dots(x_1-x_n)} + \dots + \dfrac{x_n^{M+n-1}}{(x_n-x_1)\dots(x_n-x_{n-1})}$

Method of mathematical induction:

1) Base $n = 2$:

$(p_1 + p_2)\sum\limits_{i=0}^{M} x_1^i x_2^{M-i} = (p_1 + p_2)\dfrac{x_1^{M+1} - x_2^{M+1}}{x_1 - x_2} = (p_1 + p_2)\left(\dfrac{x_1^{M+1}}{x_1-x_2} + \dfrac{x_2^{M+1}}{x_2-x_1}\right)$

Steps 2) and 3) are the same as in calculating constant C

$P^* = (p_1 + \dots + p_n)\dfrac{g_n(X,M)}{f_n(X,M)}$

4.3 Common case

Suppose that there are k_0 queues that have the probabilities: $\dfrac{p_i}{q_i} = x_0 = 1$,

k_1 queues, that: $\dfrac{p_{i_1}}{q_{i_1}} = \dots = \dfrac{p_{i_{k_1}}}{q_{i_{k_1}}} = x_1$

...

k_s queues, that: $\dfrac{p_{j_1}}{q_{j_1}} = \dots = \dfrac{p_{j_{k_s}}}{q_{j_{k_s}}} = x_s$

$k_0 + k_1 + \dots + k_s = n$

Find the constant C:

$$\begin{aligned}\frac{1}{C} &= \sum_{i_s=0}^{M}\sum_{i_{s-1}=0}^{M-i_s}\cdots\sum_{i_1=0}^{M-i_2-\cdots-i_s}\sum_{i_0=0}^{M-i_1-\cdots-i_s} \\ &\binom{k_s+i_s-1}{k_s}x_s^{i_s}\binom{k_{s-1}+i_{s-1}-1}{k_{s-1}}x_{s-1}^{i_{s-1}}\cdots\binom{k_1+i_1-1}{k_1}x_1^{i_1}\binom{k_0+i_0}{k_0} = \\ &= f_s^*(X,M)\end{aligned}$$

Proof that

$$\begin{aligned}f^*(X,M) &= \frac{\partial^{n-s}}{\partial^{k_0}x_0\partial^{k_1-1}x_1\ldots\partial^{k_s-1}x_s}\left\{\frac{1}{k_0!(k_1-1)!\ldots(k_s-1)!}\right. \\ &\left.\left(\frac{x_0^{M+n}}{(x_0-x_1)\ldots(x_0-x_s)}+\cdots+\frac{x_s^{M+n}}{(x_s-x_0)\ldots(x_s-x_{s-1})}\right)\right\}\end{aligned}$$

Method of mathematical induction:

1) Base $s=1$:

$\sum_{j=0}^{M}\binom{k_1+j-1}{j}x_1^j\binom{M+k_0-j}{M-j} = \sum_{j=0}^{M}\frac{(k_1+j-1)!}{j!(k_1-1)!}x_1^j\frac{(M+k_0-j)!}{(M-j)!k_0!}x_0^{M-j} =$

$= \frac{1}{k_0!(k_1-1)!}\sum_{j=0}^{M}(M-j+1)\ldots(M-j+k_0)x_0^{M-j}(j+1)\ldots(j+k_1-1)x_1^j =$

$= \frac{\partial^{k_0+k_1-1}}{\partial^{k_0}x_0\partial^{k_1-1}x_1}\left\{\frac{1}{k_0!(k_1-1)!}\sum_{j=0}^{M}x_0^{M-j+k_0}x_1^{j+k_1-1}\right\} =$

$= \frac{\partial^{k_0+k_1-1}}{\partial^{k_0}x_0\partial^{k_1-1}x_1}\left\{\frac{1}{k_0!(k_1-1)!}\sum_{j=-k_1+1}^{M+k_0}x_0^{M-j+k_0}x_1^{j+k_1-1}\right\} =$

$= \frac{\partial^{k_0+k_1-1}}{\partial^{k_0}x_0\partial^{k_1-1}x_1}\left\{\frac{1}{k_0!(k_1-1)!}\frac{x_0^{M+k_0+k_1}-x_1^{M+k_0+k_1}}{x_0-x_1}\right\} =$

$= \frac{\partial^{n-s}}{\partial^{k_0}x_0\partial^{k_1-1}x_1}\left\{\frac{1}{k_0!(k_1-1)!}\left(\frac{x_0^{M+n}}{x_0-x_1}+\frac{x_1^{M+n}}{x_1-x_0}\right)\right\}$

2) It is true for $0,\ldots,s-1$

3) Proof for s:

$$\begin{aligned}f_s^*(X,M) &= \sum_{i_s=0}^{M}\binom{k_s+i_s-1}{k_s}x_s^{i_s}\sum_{i_{s-1}=0}^{M-i_s}\cdots\sum_{i_1=0}^{M-i_2-\cdots-i_s}\sum_{i_0=0}^{M-i_1-\cdots-i_s} \\ &\binom{k_{s-1}+i_{s-1}-1}{k_{s-1}}x_{s-1}^{i_{s-1}}\cdots\binom{k_1+i_1-1}{k_1}x_1^{i_1}\binom{k_0+i_0}{k_0} = \\ &= \sum_{j=0}^{M}\binom{k_s+i_s-1}{k_s}x_s^{i_s}f_{s-1}^*(X,M-j)\end{aligned}$$

Simplify the summand:

$\sum_{j=0}^{M}\binom{k_s+j-1}{k_s}x_s^jx_i^{M-j+k_0+\cdots+k_{s-1}} = \frac{1}{k_s!}\sum_{j=0}^{M}(k_s+1)\ldots(k_s+i_s-1)x_s^jx_i^{M-j+k_0+\cdots+k_{s-1}} =$

$\frac{\partial^{k_s-1}}{\partial^{k_s-1}x_s}\left\{\frac{1}{(k_s-1)!}\sum_{j=0}^{M}x_s^{j+k_s-1}x_i^{M-j+k_0+\cdots+k_{s-1}}\right\} =$

$\frac{\partial^{k_s-1}}{\partial^{k_s-1}x_s}\left\{\frac{1}{(k_s-1)!}\sum_{j=-k_s+1}^{M}x_s^{j+k_s-1}x_i^{M-j+k_0+\cdots+k_{s-1}}\right\} =$

$$\frac{\partial^{k_s-1}}{\partial^{k_s-1}x_s}\left\{\frac{1}{(k_s-1)!}\left(\sum_{j=-k_s+1}^{M+k_1+\dots+k_{s-1}} x_s^{j+k_s-1}x_i^{M-j+k_0+\dots+k_{s-1}} - \sum_{j=M+1}^{M+k_0+\dots+k_{s-1}} x_s^{j+k_s-1}x_i^{M-j+k_0+\dots+k_{s-1}}\right)\right\} =$$

$$\frac{\partial^{k_s-1}}{\partial^{k_s-1}x_s}\left\{\frac{1}{(k_s-1)!}\left(\frac{x_s^{M+k_0+\dots+k_s}-x_i^{M+k_0+\dots+k_s}}{x_s-x_i} - (x_i^{k_0+\dots+k_{s-1}-1}x_s^{M+k_s}+\dots+x_s^{M+n-1})\right)\right\} =$$

$$\frac{\partial^{k_s-1}}{\partial^{k_s-1}x_s}\left\{\frac{1}{(k_s-1)!}\left(-\frac{x_s^{M+n}}{x_i-x_s}+\frac{x_i^{M+n}}{x_i-x_s} - (x_i^{k_0+\dots+k_{s-1}-1}x_s^{M+k_s}+\dots+x_s^{M+n-1})\right)\right\} \quad (*)$$

$(1 \le i \le s-1)$

Summarise the last expression for i from 1 to $s-1$ and simplify it:

From the Lemma:

$0 \le p \le s-1$:

$$\frac{x_0^p}{(x_0-x_1)\dots(x_0-x_{s-1})}+\frac{x_1^p}{(x_1-x_0)\dots(x_1-x_{s-1})}+\dots+$$
$$+\frac{x_{s-1}^p}{(x_{s-1}-x_0)\dots(x_{s-1}-x_{s-2})}=0$$

Apply the formula from the consecutive representation in reverse order:

$s \le p \le k_0+\dots+k_{s-1}-1$:

$$\frac{x_0^p}{(x_0-x_1)\dots(x_0-x_{s-1})}+\frac{x_1^p}{(x_1-x_0)\dots(x_1-x_{s-1})}+\dots+\frac{x_{s-1}^p}{(x_{s-1}-x_0)\dots(x_{s-1}-x_{s-2})}=$$
$$=\sum_{i_0=0}^{p-s+1}\sum_{i_1=0}^{p-s+1-i_0}\dots\sum_{i_{s-1}=0}^{p-s+1-i_0-\dots-i_{s-2}} x_0^{i_0}x_1^{i_1}\dots x_{s-1}^{i_{s-1}}$$

$$\frac{\partial^{n-s}}{\partial^{k_0}x_0\partial^{k_1-1}x_1\dots\partial^{k_{s-1}-1}x_{s-1}}\left\{\sum_{i_0=0}^{p-s+1}\sum_{i_1=0}^{p-s+1-i_0}\dots\sum_{i_{s-1}=0}^{p-s+1-i_0-\dots-i_{s-2}} x_0^{i_0}x_1^{i_1}\dots x_{s-1}^{i_{s-1}}\right\}=0$$

As all the summands have the total power not more than

$p-s+1 \le k_0+\dots+k_{s-1}-1-s+1 = k_0+\dots+k_{s-1}-s$,

and the order of the partial derivative is

$k_0+(k_1-1)+\dots+(k_{s-1}-1)=k_0+\dots+k_s-s+1$,

i.e. at least one less the total power of each summand.

Hence, while summaring in (*) the derivative of sum the following summands will be equal to 0

$$\frac{\partial^{n-s}}{\partial^{k_0}x_0\partial^{k_1-1}x_1\dots\partial^{k_{s-1}-1}x_{s-1}}\left\{\sum_{i=0}^{s-1}\frac{x_i^{k_0+\dots+k_{s-1}-1}x_s^{M+k_s}+\dots+x_s^{M+n-1}}{(x_i-x_0)\dots(x_i-x_{s-1})}=0\right\} \qquad 1 \le i \le s-1$$

From the Lemma:

$$\frac{1}{(x_0-x_1)\dots(x_0-x_{s-1})(x_0-x_s)}+\dots+\frac{1}{(x_{s-1}-x_0)\dots(x_{s-1}-x_{s-2})(x_{s-1}-x_s)}=\frac{1}{(x_s-x_0)\dots(x_s-x_{s-1})}$$

Finally obtain:

$$\frac{1}{C}=f_s^*(X,M)=\frac{\partial^{n-s}}{\partial^{k_0}x_0\partial^{k_1-1}x_1\dots\partial^{k_s-1}x_s}\left\{\frac{1}{k_0!(k_1-1)!\dots(k_s-1)!}\left(\frac{x_0^{M+n}}{(x_0-x_1)\dots(x_0-x_s)}+\dots+\frac{x_s^{M+n}}{(x_s-x_0)\dots(x_s-x_{s-1})}\right)\right\}$$

5 Comparison between consecutive and linked list presentations

In previous sections we obtain the formulas which express the average part of time which the system is situated in the state of "reset tail". In this section we will compare consecutive and linked list presentations. Our results were obtained when $m\to\infty$, but they are correct in prelimit form when the size of memory is rather small about 10-20 units. To check results we used system of vector algebra maxima.

We will distinguish several cases of dependences between probabilities:

1. $p_1>q_1$ and $\frac{p_1}{q_1}>\frac{p_i}{q_i}$ for $i=2,\dots,n$.

Consecutive implementation:

$$\lim_{m\to\infty}\frac{q_i-p_i}{\left(\frac{q_i}{p_i}\right)^{k_i+1}-1}=\begin{cases}p_i-q_i, & p_i>q_i\\ 0, & p_i<q_i\end{cases}$$

$$\lim_{m\to\infty}\frac{p_i}{k_i+1}=0$$

Hence, $P_N^*\to\sum_{i=1}^{n}\max(p_i-q_i,0)$

Linked list implementation:

$$P_l^*=(p_1+\dots+p_n)\frac{\sum_{i=1}^{n}\frac{x_i^{M+n-1}}{\prod_{\substack{j=1\\ j\neq i}}^{n}(x_i-x_j)}}{\sum_{i=1}^{n}\frac{x_i^{M+n}}{(x_i-1)\prod_{\substack{j=1\\ j\neq i}}^{n}(x_i-x_j)}+\frac{1}{\prod_{j=1}^{n}(1-x_j)}}\to$$

$$(p_1+\dots+p_n)\left(\frac{x_i-1}{x_i}\right)=(p_1+\dots+p_n)\left(1-\frac{q_1}{p_1}\right)$$

$p_i(1-\dfrac{q_1}{p_1})>0 \quad 2\le i\le n$, because $p_1>q_1$

$p_i(1-\dfrac{q_1}{p_1})=p_i-p_i\dfrac{q_1}{p_1}>p_i-q_i$

Hence, $p_i(1-\dfrac{q_1}{p_1})>\max(p_i-q_i,0)$

Summarise the last inequality for i from 2 to n and add p_i-q_i:

$$\lim_{m\to\infty}P_c^*=\sum_{i=1}^{n}\max(p_i-q_i,0)<(p_1+\cdots+p_n)\left(1-\frac{q_1}{p_1}\right)=\lim_{m\to\infty}P_l^*$$

$P_c^*<P_l^*$ even when the size of memory is rather small.

2. $p_i=q_i=\dfrac{1}{2n}$ for $i=1,\ldots,n.$

$$P_c^*=\sum_{i=1}^{n}\frac{p_i}{k_i+1}=\sum_{i=1}^{n}\frac{\frac{1}{2n}}{\frac{m}{n}+1}=\frac{n}{m+n}\quad P_c^*\le\frac{n}{m+n}$$

$$P_l^*=\frac{n}{M+n}$$

$P_c^*<P_l^*$

3. $p_i<q_i$ for $i=1,\ldots,n.$ and $\dfrac{p_1}{q_1}>\dfrac{p_i}{q_i}$

In [1] we found the optimal partition of memory in the case of consecutive presentation using the method of dynamic programming. All the queues will spend roughly the same part of time in the state of "reset tail". I.e.

$$\frac{q_i-p_i}{\left(\frac{q_i}{p_i}\right)^{k_i+1}-1}\approx\frac{q_j-p_j}{\left(\frac{q_j}{p_j}\right)^{k_j+1}-1}\qquad\forall i\ne j$$

Using this equations and condition $k_1+\cdots+k_n$ we can find the roughly values of variables $k_1,\ldots,k_n$:

$$P_c^*\approx\frac{1}{\exp\left(\dfrac{m-n}{\sum_{i=1}^{n}\frac{1}{\log(\frac{q_i}{p_i})}}-\log n\right)}=\mathrm{O}\left(\exp\left(-\frac{m}{\sum_{i=1}^{n}\frac{1}{\log(\frac{q_i}{p_i})}}\right)\right)$$

The behaviour of the function $P_l^*(M)$ will be determined by the value

$$\sum_{i=1}^{n}\frac{x_i^{M+n-1}}{\prod_{\substack{j=1\\ j\ne i}}^{n}(x_i-x_j)}$$

because in the denominator $\displaystyle\sum_{i=1}^{n}\frac{x_i^{M+n}}{(x_i-1)\prod_{\substack{j=1\\ j\ne i}}^{n}(x_i-x_j)}\to 0$

and value $\dfrac{1}{\prod\limits_{j=1}^{n}(1-x_j)}$ is a constant. Thus

$$P_l^* = \mathrm{O}\Bigg(\Big(\frac{q_1}{p_1}\Big)^M\Bigg) = \mathrm{O}\Bigg(\exp\Big(-m(1-\frac{1}{l})\log\frac{q_1}{p_1}\Big)\Bigg)$$

In this case the part of time which the system is situated in the state of "reset tail" exponentially tends to 0 in both cases of presentation. To choose best of them we need to compare the exponents and choose minimal of them.

Bibliography

[1] E. A. Aksenova, A. V. Drac, A. V. Sokolov. *The optimel control of n FIFO-queues for infinit time. Information and Conrot Systems. p. 46-54. 2009. In russian.*

[2] D. E. Knuth. *The art of computer programming. Vol. Addison-Wesley, Reading, MA, 2001.*

[3] M. Tolley, G. Louchard, R. Schott. *Random walks, heat equation and distributed algorithms. J. Comput. Appl. Math. p. 243-274. 1994.*

[4] J. Riordan. *Introduction to Combinatorial Analysis. Dover Publications, 1958.*

[5] J. L. Snell, J. G. Kemeny. *Finite Markov Chains. Van Nostrand, Princeton, New Jersey, 1960.*