

UDC 519.872

# The Queue Distribution in Multiprocessor Systems with the Waiting Time Restriction

Artur P. Vardanyan and Vladimir G. Sahakyan

Institute for Informatics and Automation Problems of NAS RA  
e-mail: artvardanyan@asnet.am, vladimir.sahakyan@sci.am

## Abstract

A queueing system model is considered, consisting of  $m$  ( $m \geq 1$ ) servicing devices and a maximum number of tasks with  $n$  ( $n \geq 1$ ) in the waiting queue. Each task is characterized by three random parameters  $(\nu, \beta, \omega)$ , where  $\nu$  is the number of servicing devices required to perform the task,  $\beta$  is the maximum time required to complete the task and  $\omega$  is the possible time that the task can wait before assigning to run, after which it leaves the system without service. Tasks are accepted for service in the order of their entry into the system, i.e., FIFO (First-In-First-Out) discipline is used. In paper the equations are obtained for the state probabilities of the system in the stationary mode, which can serve as an assessment for real multiprocessor systems using MPI and OpenMP technologies.

**Keywords:** Multiprocessor cluster-type system, Cluster computing, Queueing theory, Waiting time restriction.

## 1. Introduction

The optimal use of processor time in multiprocessor cluster-type systems depends on many factors: the method of receiving and queuing the task, determining the order of execution, the possibility of dynamically distributing computing resources, the ability to move the task during different phases of execution to the minimum necessary environment or stop the execution with the possibility of continuing, etc.. The reception of a task in the system for execution plays an important role in the organization of this process. The ability to interact distributed processes in certain periods of time requires synchronization and simultaneous execution both in one and different computer systems. Therefore, accepting a task from the queue for execution imposes the responsibility on the scheduler for ensuring its timely execution. At the same time, tasks arriving for execution may be "impatient", that is, they leave the queue after a certain waiting time. In this paper, the probabilities of the queue state are obtained for the exponential distributions of the task of receipt, execute, and failure of service. Such models play an important role on multiprocessor systems using MPI and OpenMP technologies [1].

Suppose that a task stream enters a computing system consisting of  $m$  processors ( $m \geq 1$ ). Each task is characterized by three random parameters  $(\nu, \beta, \omega)$ , where  $\nu$  is the number of

computational resources(processors, cores, cluster nodes, etc.,) required to perform the task,  $\beta$  is the maximum time required to complete the task and  $\omega$  is the possible time that the task can wait before assigning to run, after which it leaves the system without service [2].

By using David Kendall's notation(which is widely used to describe elementary queueing systems)[3], the system under consideration can be represented as  $M|M|m|n$ . So, the system parameters are described:

$m$  - the maximum number of computational resources;

$n$  - the maximum permissible number of tasks in the queue;

$\alpha$  - a random value of the time interval between neighboring entrances, which has the probability distribution:

$$P(\alpha < t) = 1 - e^{-at},$$

where  $a$  is the intensity of the incoming stream;

$\beta$  - a random value of the task execution time, which has the probability distribution:

$$P(\beta < t) = 1 - e^{-bt},$$

where  $b$  is the intensity of service;

$\omega$  - a random value of the permissible waiting time for a task in the queue, which has the probability distribution:

$$P(\omega < t) = 1 - e^{-wt},$$

where  $w$  is the intensity of the failure of service for a task from the queue;

$\nu$  - a random value of the number of required computational resources for performing a task, which has the probability distribution:

$$P(\nu \leq k) = \frac{k}{m}, k = 1, 2, \dots, m.$$

Tasks will be accepted for service in the order of their entry into the system, i.e., FIFO discipline is used (First-In-First-Out). Those tasks that arrive at the time of full occupation of the queue (there are already  $n$  tasks in the queue) receive a denial of service.

## 2. Basic Notations and Lemmas

To obtain a system of equations, we need the values of some probabilistic characteristics. By  $P(i, k)$  is denoted the probability that  $k$  processors will be occupied by  $i$  tasks:

$$P(i, k) = P\left(\sum_{j=1}^i \nu_j = k\right).$$

**Lemma 2.1:** *The probability that  $k$  processors will be occupied by  $i$  tasks, can be obtained in the following way:*

$$P(i, k) = \frac{1}{m^i} \binom{k-1}{i-1}, 1 \leq i \leq k \leq m.$$

**Proof.** To prove the lemma we use the mathematical induction technique. The method of induction requires two cases to be proved. The first case, called the base case, proves that the property holds for  $i = 1$ :

$$P(1, k) = \frac{1}{m} \binom{k-1}{0} = \frac{1}{m}.$$

The statement is true because if  $i = 1$ , then

$$P(1, k) = P(\nu = k) = \frac{1}{m}.$$

The second case, called the induction step, proves that if the property holds for number  $i - 1$ , then it holds for the next natural number  $i$ :

$$P(i, k) = \sum_{j=i-1}^{k-1} P(i-1, j)P(1, k-j) = \frac{1}{m} \sum_{j=i-1}^{k-1} \frac{1}{m^{i-1}} \binom{j-1}{i-2} = \frac{1}{m^i} \sum_{j=i-1}^{k-1} \binom{j-1}{i-2} \quad (1)$$

from combinatorics we know this equality [4]:

$$\binom{i}{i} + \binom{i+1}{i} + \dots + \binom{i+k-1}{i} = \binom{i+k}{i+1} \quad (2)$$

considering (2) equality to count (1), we get the formula, which was mentioned in Lemma 2.1.:

$$P(i, k) = \frac{1}{m^i} \binom{k-1}{i-1}.$$

■

**Lemma 2.2:** *The probability that  $i$  tasks will occupy no more than  $k$  processors, can be obtained in the following way:*

$$P\left(\sum_{j=1}^i \nu_j \leq k\right) = \frac{1}{m^i} \binom{k}{i}, 1 \leq i \leq k \leq m.$$

**Proof.** To prove the lemma we use the formula, which we got in Lemma 2.1.

$$P\left(\sum_{j=1}^i \nu_j \leq k\right) = \sum_{j=i}^k P(i, j) = \sum_{j=i}^k \frac{1}{m^i} \binom{j-1}{i-1} = \frac{1}{m^i} \sum_{j=i}^k \binom{j-1}{i-1}$$

to calculate the last sum, we again use the (2) equality and as a result we get that

$$P\left(\sum_{j=1}^i \nu_j \leq k\right) = \frac{1}{m^i} \binom{k}{i}.$$

■

**Lemma 2.3:**

$$P\left(\sum_{i=1}^k \nu_i \leq s < \sum_{i=1}^{k+1} \nu_i\right) = \frac{1}{m^{k+1}} \left(m - \frac{s-k}{k+1}\right) \binom{s}{k}, 1 \leq k \leq s \leq m.$$

**Proof.** It's obvious that:

$$P\left(\sum_{i=1}^k \nu_i \leq s < \sum_{i=1}^{k+1} \nu_i\right) = \sum_{j=k}^s P\left(\sum_{i=1}^k \nu_i = j\right) P(\nu_{k+1} > s - j).$$

Primarily, we use the obvious fact that

$$P(\nu_{k+1} > s - j) = \frac{m - s + j}{m},$$

and then we use the formula, which we got in Lemma 2.1. for the first probability in sum, as a result we get:

$$\begin{aligned} P\left(\sum_{i=1}^k \nu_i \leq s < \sum_{i=1}^{k+1} \nu_i\right) &= \frac{1}{m^{k+1}} \left( \sum_{j=k}^s (m - s) \binom{j-1}{k-1} + \sum_{j=k}^s j \binom{j-1}{k-1} \right) = \\ &= \frac{1}{m^{k+1}} \left( (m - s) \binom{s}{k} + k \binom{s+1}{k+1} \right) = \frac{1}{m^{k+1}} \left( m - \frac{s-k}{k+1} \right) \binom{s}{k}. \end{aligned}$$

■

### 3. The Equations for System State

To analyze our system we need to identify the following basic notation:

$P_{i,j}(t)$  - the probability that at the moment of time  $t$  there are  $i$  tasks in service, and in the queue  $j$  tasks wait for service;

Due to the finite number of possible states of the system ( $m*n+1$ ) with long-term operation, the system goes into a steady mode of operation, i.e., in a stationary state [5]. In this case, the limiting values  $P_{i,j}(t)$  are considered as  $t$  tends to infinity, which will be denoted by  $P_{i,j}$ .

By  $L_{i,j}$  we denote the state of the system when  $i$  tasks are serviced and  $j$  tasks are waiting in the queue. Cases when the system can pass  $L_{i,j}$  state from the other state are presented in the following scheme:

$$\begin{array}{c} L_{i-k+1,j+k} \xrightarrow{q^{(1)}(i,j)} L_{i,j} \xleftarrow{q^{(2)}(i,j)} L_{i,j-1} \\ \uparrow q^{(3)}(i,j,k) \\ L_{i,j+1} \text{ or } L_{i-k,j+k+1} \end{array}$$

where  $q^{(1)}(i,j)$ ,  $q^{(2)}(i,j)$ ,  $q^{(3)}(i,j)$  are probabilities for appropriate passing and when the passing is from  $L_{i-k+1,j+k}$ , then  $k = 1, 2, \dots, \min(i, n - j)$ , but when the passing is from  $L_{i-k,j+k+1}$ , then  $k = 0, \dots, i - 1$ . Note that if  $j = 0$ , then there won't be the passing from  $L_{i,j-1}$  and if  $j = n$ , then there won't be the passing from  $L_{i,j+1}$  or  $L_{i-k,j+k+1}$ . We also assume that at the passing from  $L_{i-k,j+k+1}$  the first task from the queue leaves the queue and at the passing from  $L_{i,j+1}$  not the first task, but another task from the queue leaves the queue. Obviously,

$$q^{(1)}(i,j) = \sum_{k=0}^{\min(i,n-j)} \left( \frac{(i-k+1)b}{a + (i-k+1)b + (j+k)w} P_{i-k+1,j+k} \bar{P}(i,k) \right),$$

where  $\bar{P}(i, k) = 0$  if  $i = 0$ , but if  $0 < i \leq m$ , then  $\bar{P}(i, k)$  is the following conditional probability:

$$\bar{P}(i, k) = P \left( \sum_{s=1}^{i-k} \nu_s + \sum_{s=i-k+2}^{i+1} \nu_s \leq m, \sum_{s=1}^{i-k} \nu_s + \sum_{s=i-k+2}^{i+2} \nu_s > m \middle/ \sum_{s=1}^{i-k+1} \nu_s \leq m, \sum_{s=1}^{i-k+2} \nu_s > m \right),$$

where we consider that  $\nu_{i-k+1}$  is the number of required computational resources required to service the task that was being left the system(it was serviced) and due to which the system has changed its state,

$$q^{(2)}(i, j) = \frac{a}{a + ib + (j - 1)w} P_{i, j-1},$$

$$q^{(3)}(i, j) = \frac{jw}{a + ib + (j + 1)w} P_{i, j+1} + \sum_{k=0}^{i-1} \left( \frac{w}{a + (i - k)b + (j + k + 1)w} P_{i-k, j+k+1} \bar{\bar{P}}(i, k) \right),$$

where  $\bar{\bar{P}}(i, k) = 0$  if  $i = 0$ , but if  $0 < i \leq m$ , then  $\bar{\bar{P}}(i, k)$  is the following conditional probability:

$$\bar{\bar{P}}(i, k) = P \left( \sum_{s=1}^{i-k} \nu_s + \sum_{s=i-k+2}^{i+1} \nu_s \leq m, \sum_{s=1}^{i-k} \nu_s + \sum_{s=i-k+2}^{i+2} \nu_s > m \middle/ \sum_{s=1}^{i-k} \nu_s \leq m, \sum_{s=1}^{i-k+1} \nu_s > m \right),$$

where we consider that  $\nu_{i-k+1}$  is the number of required computational resources required to service the task that was being left the system(it left the queue) and due to which the system has changed its state.

In this case, the equations for system state are given in the following way:

$$P_{i, j} = \eta_j q^{(1)}(i, j) + \theta_j q^{(2)}(i, j) + \eta_j q^{(3)}(i, j), \quad (3)$$

where  $0 \leq i \leq m$ ,  $0 \leq j \leq n$  and

$$\eta_j = \begin{cases} 0, & \text{for } j = n \\ 1, & \text{for } 0 \leq j < n \end{cases},$$

$$\theta_j = \begin{cases} 0, & \text{for } j = 0 \\ 1, & \text{for } 0 < j \leq n \end{cases}.$$

Note that if  $i = 0$ , then  $P_{0, j} = 0$  for  $0 \leq j \leq n$ .

To calculate  $\bar{P}(i, k)$  probability, we first perform a simple transformation, then use the conditional probability formula:

$$\begin{aligned} \bar{P}(i, k) &= P \left( \sum_{s=1}^{i+1} \nu_s \leq m + \nu_{i-k+1} < \sum_{s=1}^{i+2} \nu_s \middle/ \sum_{s=1}^{i-k+1} \nu_s \leq m < \sum_{s=1}^{i-k+2} \nu_s \right) = \\ &= \frac{P \left( \sum_{s=1}^{i+1} \nu_s \leq m + \nu_{i-k+1} < \sum_{s=1}^{i+2} \nu_s, \sum_{s=1}^{i-k+1} \nu_s \leq m < \sum_{s=1}^{i-k+2} \nu_s \right)}{P \left( \sum_{s=1}^{i-k+1} \nu_s \leq m < \sum_{s=1}^{i-k+2} \nu_s \right)}. \end{aligned}$$

By using Lemma 2.3. we can calculate the probability, which is in the denominator of the last fraction:

$$P \left( \sum_{s=1}^{i-k+1} \nu_s \leq m < \sum_{s=1}^{i-k+2} \nu_s \right) = \frac{i-k+1}{m^{i-k+2}} \binom{m+1}{i-k+2}.$$

Before the calculation of the probability, which is in the numerator of the fraction, it is denoted by  $\delta_k$ , then it is calculated in the following way:

$$\delta_k = \sum_{u=i-k}^{m-k+1} P \left( \sum_{s=1}^{i-k} \nu_s = u \right) \tilde{P}_u,$$

where  $k = 1, 2, \dots, \min(i, n-j)$  and

$$\tilde{P}_u = P \left( \sum_{s=i-k+2}^{i+1} \nu_s \leq m-u < \sum_{s=i-k+2}^{i+2} \nu_s, \nu_{i-k+1} \leq m-u < \nu_{i-k+1} + \nu_{i-k+2} \right).$$

Obviously, in the last probability we deal with independent probabilities and with the help of Lemma 2.3. for  $\tilde{P}_u$  we get the following formula:

$$\tilde{P}_u = \frac{(m-u)(m+u+1)((m+1)k+u)}{2(k+1)m^{k+3}} \binom{m-u}{k}. \quad (4)$$

By using Lemma 2.1. as a result we get the following formula for  $\delta_k$ :

$$\delta_k = \frac{1}{m^{i-k}} \sum_{u=i-k}^{m-k+1} \binom{u-1}{i-k-1} \tilde{P}_u, \quad (5)$$

where  $\tilde{P}_u$  is calculated by formula (4). So, we get formula for  $\bar{P}(i, k)$  probability:

$$\bar{P}(i, k) = \frac{m^{i-k+2}}{(i-k+1) \binom{m+1}{i-k+2}} \delta_i. \quad (6)$$

Note that we can calculate the probability  $\bar{\bar{P}}(i, k)$  in the same way as  $\bar{P}(i, k)$ .

Thus, according to equation (3), probabilistic equation of the state of the system is given. As we know, if  $i = 0$  for all  $0 \leq j \leq n$

$$P_{0,j} = 0,$$

and please note that

$$\sum_{i=0}^m \sum_{j=0}^n P_{i,j} = 1.$$

The probability that the system will refuse a new arrival task is denoted by  $R$ .

**Corollary 3.1.**

$$R = \sum_{i=1}^m \frac{a}{a + ib + nw} P_{i,n}.$$

**4. Conclusion**

In classical queueing systems, one task does not require more than one servicing device, but in this paper we suggest a queueing system model that differs from other queueing systems. In the suggested new model it may take more than one servicing devices to perform one task. Such a queueing system model can play an important role on multiprocessor systems using MPI and OpenMP technologies. In paper for the exponential distributions of the task of receipt, execute, and failure of service, the probabilistic equation of state of the system is obtained in the stationary mode.

**References**

- [1] N. Dahnoun, *Multicore DSP: From Algorithms to Real-Time Implementation on the TMS320C66x SoC*, WILEY, Bristol, 2018.
- [2] V. Sahakyan, Y. Shoukourian and H. Astsatryan, "About some queueing models for computational grid systems", *Transactions of IIAP NAS RA, Mathematical Problems of Computer Science*, vol. 46, pp.55-58, 2016.
- [3] L. Kleinrock, *Queueing Systems: Volume 1 - Theory*, Wiley Interscience, New York, 1975.
- [4] Н. Я. Виленкин, *Комбинаторика*, Наука, Москва, 1969.
- [5] Б. В. Гнеденко и И. Н. Коваленко, *Введение в теорию массового обслуживания*, Наука, Москва, 1987.

Submitted 04.02.2019, accepted 26.04.2019.

## Հերթի բաշխումը բազմապրոցետրային համակարգում սպասման ժամանակի սահմանափակման դեպքում

Վլադիմիր Գ. Սահակյան և Արթուր Պ. Վարդանյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ  
e-mail: vladimir.sahakyan@sci.am, artvardanyan@asnet.am

### Անփոփում

Գիտարկված է զանգվածային սպասարկման համակարգի մոդել, որում որպես սպասարկող սարքեր վերցվում է  $m$  ( $m \geq 1$ ) պրոցեսորից բաղկացած բազմապրոցետրային

համակարգը, հերթում պահանջների առավելագույն թույլատրելի քանակը  $n$  ( $n \geq 1$ ) է: Համակարգում յուրաքանչյուր պահանջ բնութագրվում է  $(\nu, \beta, \omega)$  պարամետրեր, որտեղ  $\nu$ -ն պահանջի սպասարկման համար անհրաժեշտ ռեսուրսների(պրոցեսորների) քանակն է,  $\beta$  - ն պահանջի սպասարկման առավելագույն ժամանակն է, իսկ  $\omega$  - ն այն առավելագույն ժամանակն է, որ այդ պահանջը կարող է հերթում սպասել մինչև սպասարկման ուղարկվելը: Արդյունքում յուրաքանչյուր պահանջ ունենում է սպասման ժամանակի սահմանափակում: Աշխատանքում ստացված են համակարգի վիճակը բնութագրող հավանականային հավասարումներ, երբ համակարգը կայուն վիճակում է:

**Բանալի բառեր`** բազմապրոցեսորային կլաստերային տիպի համակարգ, կլաստերային հաշվարկ, մասսայական սպասարկման տեսություն, սպասման ժամանակի սահմանափակում:

## Распределение очереди в многопроцессорной системе при ограничении на время ожидания

Владимир Г. Саакян и Артур П. Варданыан

Институт проблем информатики и автоматизации НАН РА  
e-mail: vladimir.sahakyan@sci.am, artvardanyan@asnet.am

### Аннотация

Рассмотрена модель системы массового обслуживания, состоящая из  $m$  ( $m \geq 1$ ) обслуживающих приборов и с максимальным количеством заданий в очереди ожидания -  $n$  ( $n \geq 1$ ). Каждое задание характеризуется тремя случайными параметрами  $(\nu, \beta, \omega)$ , где  $\nu$  - число требуемых обслуживающих приборов, необходимых для выполнения задания,  $\beta$  - время, требуемое для выполнения задания,  $\omega$  - допустимое время пребывания задания в очереди до начала его выполнения, после которого оно покидает систему без обслуживания. Задания принимаются на обслуживание в порядке поступления их в систему, т.е. используется дисциплина FIFO (First-In, First-Out). В работе получены уравнения для вероятностей состояния системы в стационарном режиме, которые могут служить оценкой для реальных многопроцессорных систем, использующих технологии MPI и OpenMp.

**Ключевые слова:** многопроцессорная система кластерного типа, кластерные вычисления, теория массового обслуживания, ограничение времени ожидания.