MATHEMATICAL MODELING OF TASKS MANAGERS WITH THE STRATEGY OF SEPARATION IN SPACE WITH A HOMOGENEOUS AND HETEROGENEOUS INPUT FLOW AND FINITE QUEUE

A. I. Martyshkin
Penza State Technological University, Baydukov Proyezd / Gagarin Street, 1a/11, Penza, Penza Region, Russia
E-Mail: alexey314@yandex.ru

ABSTRACT

The article deals with mathematical modeling for analyzing the capacity of a multiprocessor system with Tasks Managers with the strategy of separation in space. Research methods are based on concepts of the analytical modeling theory, systems theory and queuing networks, probability theory and stochastic processes. The article presents analytical equations for the research of tasks managers with the strategy of separation in space for multiprocessor systems based on open queuing networks. The analytical part has been checked by simulation modeling. The dependence under study has been illustrated by diagrams. Finally, conclusions have been made.

Keywords: mathematical modeling, task manager, separation in space, stochastic process, stochastic network, priority, probability.

1. INTRODUCTION

In real computing systems (CSs) queries for servicing come at random times. Times allocation of queries and servicing may follow certain laws. In this article, for simplicity, mathematical modeling of a system is performed by means of mass servicing systems (MSSs) of the M/M/1 type (exponential MSS; the flow of tasks is the simplest one, the duration of servicing is allocated according to the exponential law). While creating and making research of models of a real CS on the basis of MSSs of the M/M/1 type, some finite characteristics of the system under consideration will be obtained. This model provides the worst behavior of a CP, where we get upper boundary probability-time characteristics and guarantee the efficiency of simulated computer system under the given parameters.

In existing CSs, especially in real-time systems (RTSs), input tasks for servicing are a heterogeneous flow containing queries of different types. For example, in RTSs tasks of different priority can come for executing (different measures of significance). High-priority queries will be serviced faster, as it is necessary to respond to them and provide a result as soon as possible. Low-priority tasks are interrupted if a received task has an absolute priority in comparison with a current one. This query having been served, the interrupted task continues its servicing (if, of course, during the process of a high priority task processing no query with a higher priority than the current task has come to the system). The interrupted task is in the standby mode. If a received task has a relative priority compared to a running one, it is in the standby mode. On this basis let’s consider a multiprocessor system (MPS) with allocated tasks managers (TMs) as a system with homogeneous and heterogeneous tasks flow for the servicing.

2. GOAL SETTING

There are two types of tasks managers (TMs) of a multiprocessor system (MPS): with the strategy of separation in time and separation in space [1, 2, 3]. TMs with the strategy of time separation are reviewed in some scientific articles [4, 5, 6, 7]. Here we will focus on TMs with the strategy of separation in space.

Mathematical model of TMs with the strategy of separation in space [1, 6] consists of n single-channels MSSs ($S_1,...,S_n$) (Figure-1). Each MSS simulates servicing in the subsystem "Task Manager-Processing Unit" ("TM-PU"). The source S_0 simulates tasks flows λ_0 and absorbs tasks serviced. For MSSs $S_1, S_3,..., S_{n-3}, S_{n-1}$ queues with a finite number of spaces *k* are formed. If there is a free space in one of the queues, then a task takes it and is stored in the local queue as long as the execution time in PU starts. If the quantization mode is used, an incomplete task at the end of the current quantum is assigned by TM for servicing and placed at the end of the queue, where it was previously placed. Otherwise, a user is provided with the result, and there is a vacancy in the local queue. At the end of the execution TM revises its queue. If there are queries for servicing a task at the top of the list is assigned to be executed. If the queue is empty, the system "TM-PU" is positioned in standby mode. "TM-PU" in this case produces load balancing according to some algorithm. So, in case of the queue overflow i-th (with probability $p_{reallocation_i}$) the tasks are retrieved and transmitted with a certain probability to the least loaded j-th queue.

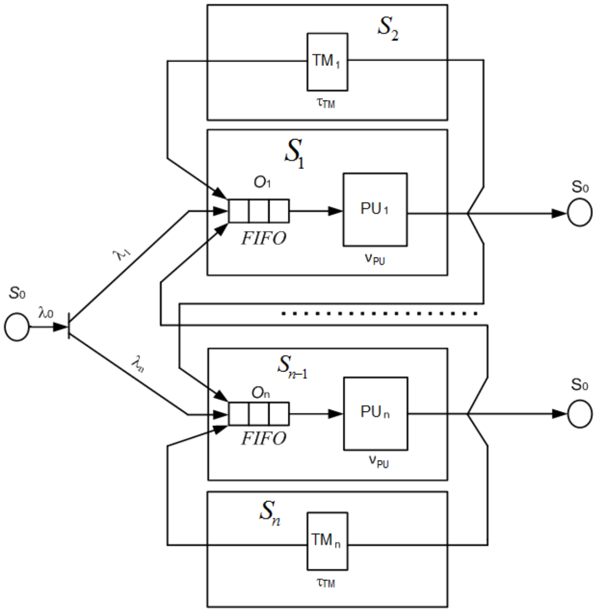

Figure-1. MPS with TMs with tasks space separation.

3. THE CALCULATION OF CHARACTERISTICS OF QUEUING NETWORK

The system under consideration is presented in the form of Queuing network (QN) being a set of single-channels MSSs. The input of each MSS ($S_1, S_3, ..., S_{n-3}, S_{n-1}$) receives a flow of queries in general with intensities $\lambda_1, \lambda_3, ..., \lambda_{n-3}, \lambda_{n-1}$. The intensity of the "TM-PU" subsystem servicing of a queries flow equals to $1\big/\left(\tau_{TM}+\nu_{PU}\right)$, where τ_{TM} is the time of TM operation, ν_{PU} – the time of PU operation. The task received at the time when the subsystem is busy is placed in the queue and in the standby mode for servicing. For example, regardless of the number of tasks being in the queue, it cannot accommodate more than k tasks, one of which is being serviced, and k–1 are in standby mode. Queries that are not in the standby queue O$_i$, are served in other place, directed to another queue. Consider the intensity of tasks flows in a MPS. The system of equations of the intensities of the tasks within the system looks like this

$$\begin{cases} \lambda_0 = P_{10}\lambda_1 + P_{30}\lambda_3 + ... + P_{(n-1)0}\lambda_{n-1}; \\ \lambda_1 = P_{01}\lambda_0 + P_{21}\lambda_2 + P_{31}\lambda_3 + ... + P_{(n-1)1}\lambda_{n-1}; \\ \lambda_2 = P_{12}\lambda_1; \\ \lambda_3 = P_{03}\lambda_0 + P_{13}\lambda_1 + P_{43}\lambda_4 + ... + P_{(n-1)3}\lambda_{n-1}; \\ \lambda_4 = P_{34}\lambda_3; \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \lambda_{n-1} = P_{0(n-1)}\lambda_0 + P_{1(n-1)}\lambda_1 + P_{3(n-1)}\lambda_3 + ... + P_{n(n-1)}\lambda_n; \\ \lambda_n = P_{(n-1)n}\lambda_{n-1}. \end{cases}$$

Intensities $\lambda_1, \lambda_3, ..., \lambda_{n-3}, \lambda_{n-1}$ depend on the flow tasks input intensity of the source S_0 (λ_0) and the transitions probabilities from MSSs S_i to MSSs S_j. As MSSs $S_1, S_3, ..., S_{n-3}, S_{n-1}$ can cause failures in servicing and allocating a task to another subsystem "TM-PU" (with probability $p_{fail_i} = \sum_{\forall n[n \neq j]} p_{i,j}$), the intensities of the flow $\lambda_1, \lambda_3, ..., \lambda_{n-3}, \lambda_{n-1}$ will drop. On the other hand, this drop will be offset by the reassignment of tasks from the completed queue. The transmission coefficient showing how much time the task will pass through the i-th MSS is determined by the equation $\alpha_i = \dfrac{\lambda_i + \sum\limits_{\forall n(n \neq i)} p_{ij} \cdot \lambda_i}{\lambda_0}$, $i,\ j = \overline{1,n}$. The task is allocated to another "TM-PU" subsystem in case where both the current subsystem "TM-PU" and all the k spaces are busy in the queue, and the probability of reallocation is $p_{reallocation_i} = \dfrac{\rho_i^{k+1}(1-\rho_i)}{1-\rho_i^{k+2}}$, where $\rho_i = \lambda_i \cdot (\tau_{TM} + \nu_{PU})$.

According to the equations and calculations in [8, 9, 10], we get the formula for finding the average queue length in the system "TM-PU"

$$l_i = \frac{\rho_i^2 \cdot (1-\rho_i) \cdot (1-\rho_i^k \cdot (k+1-k \cdot \rho_i))}{(1-\rho_i^{k+2}) \cdot (1-\rho_i)^2} =$$

$$= \frac{\rho_i^2 \cdot (1-\rho_i^k \cdot (k+1-k \cdot \rho_i))}{(1-\rho^{k+2}) \cdot (1-\rho_i)}$$

Now we determine the average standby period w_i of tasks in the queue "TM-PU". The query comes to the system at a specific period of time. With the probability p_0 the system "TM-PU" is free and the standby period is zero. With the probability p_1 a task will pass to MSS, there will be no queue, and it will be in the standby mode for servicing for the time period $\tau_{TM} + \nu_{PU}$ (the average servicing time of one task).

With the probability p_2 there will be one more task in the queue and the standby period in average will be $2\cdot\left(\tau_{TM}+\nu_{PU}\right)$, and so on. When $q=k+1$, i.e. a newly input task finds "TM-PU" to be busy and k tasks to be in the queue, the standby period is also zero in this case, because the task is not accepted in this queue. Following [8, 11, 12], the average latency of a task in i-th queue is equal to $w_i=\dfrac{l_i}{\lambda_i}$.

The average standby period of tasks in queues of the network is calculated according to the equation [8, 13, 14] $W=\sum_{i=1}^{n-1}\alpha_i\cdot w_i$, where α_i– the coefficient of transmission; w_i - the standby period in the queue for the i-th MSS. The response time in the system with the allocated TM is calculated by the formula [8] $U=\sum_{i=1}^{n-1}\alpha_i\cdot u_i$, where α_i– the coefficient of transmission; u_i – the response time i-th of the "TM-PU" subsystem.

Firstly, let's consider the basic method to carry out the analysis of the developed in [3, 6] method. The task parameters and architectural parameters of basic models under study (PU performance, quantum time, intensity of tasks input flow, PU number, average PU loading, etc.) are the same. Queues for processor nodes are different (in the basic version they are of infinite length; in case of TMs with the strategy of separation in space they are finite; also a distinctive feature is the presence of a reallocation of tasks in queues to provide system load balancing).

For computational experiments, let's take a system with 4 PUs and with 4 TMs, correspondingly. Imagine a graph of this system (Figure 2). Here S_0 is the source of problems; S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8 are the subsystems "TM-PU" (S_1, S_3, S_5, S_7 are CPU nodes of the system, and S_2, S_4, S_6, S_8 are TMs). The reallocation of tasks from the source over the queues is equally probable. The probability that a task will be reallocated to other queues after managing is 0,05 if a task is long. If a task is long the probability of providing users with results of the task served is 0,05; if a task is medium - 0,3; if a task is short - 0,7. The probability of long tasks servicing is 0,9; the medium one - 0,65; the short one - 0,25. Based on the data presented, examine a MPS with 4 PUs.

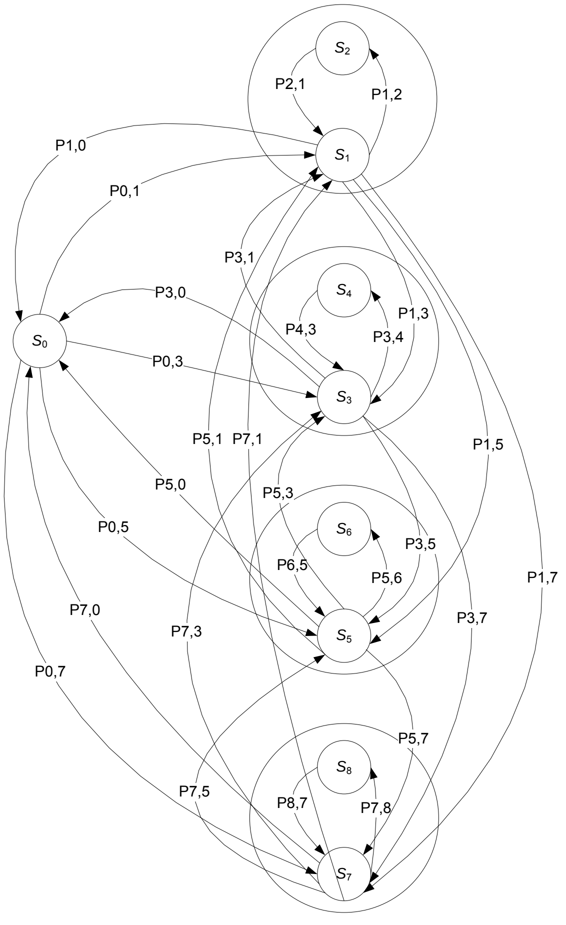

Figure-2. Graph of the system with 4 "TM-PU" subsystems.

The purpose of queries over queues of the network under study is chosen equally probable to evaluate approximately the situation with mathematical modeling of a real process, to avoid the system overloading when all the tasks try to receive servicing in one or more PUs, and some PUs stand idle $\lambda_i={}^{\lambda_0}\!/_{n}$ under the conditions of equally probable queries allocations. The response time in the system with allocated TMs

$$U=\sum_{i=1}^{n}\frac{\lambda_i+\sum\limits_{\forall n(n\neq i)}p_{ji}\cdot\lambda_i}{\lambda_0}\cdot\left(w_i+k\cdot\left(t_k+\delta+\tau+\varsigma\right)\right).$$

Now consider the system with n PUs, with a heterogeneous tasks flow and relative priorities of a task selection from the queue. The source S_0 simulates a heterogeneous tasks flow servicing (λ_0) allocated over n-1 of MSSs. The queries flow for servicing consists of H classes (priorities) of tasks $\lambda_i=\lambda_{j1}+\lambda_{j2}+\ldots+\lambda_{jH}$. For a PU a tasks queue is formed that receives heterogeneous queries. An accepted for servicing a task is in a queue until it comes for the execution to the subsystem "TM-PU". A Queuing network consists of n single-channels MSSs. Tasks with relative priorities are executed by the subsystem "TM-PU" in the input order. In case we get a query with a higher priority in the queue, it will be serviced out of turn after finishing work with the current task.

The servicing average time of a queries flow by the system "TM-PU" equals $\tau_{TM} + \nu_{PU}$. If a task comes when the system is busy it is located in the queue limited by the number of spaces *k* and is in the standby mode. Queries that are out of the standby Oi with the probability of a failure are sent to other queues. The values of flow intensities allocating over n-queues (λ_1, λ_3,..., λ_{n-3}, λ_{n-1}) depend on the input flow intensity of the source task S_0 (λ_0), the number of tasks types and the transition probabilities from MSSs S_i to MSSs S_j . According to [15], we can replace the system M/G/1 with the finite queue to a similar system with the infinite queue. QNW consisting of MSSs with infinite queues of tasks is easier to analyze and identify basic characteristics than that consisting of MSSs with the finite queue only in case if this queue has a large number of spaces *k* (not less than 16 for RTS tasks). In this case a computational error will be less than 2, 5% [15]. The average standby time of tasks with relative priorities in the queue for the i-th subsystem "TM-PU" will be determined in accordance with the equation [8]

$$w_k^{RP} = \frac{\sum_{i=1}^{H} \lambda_i \cdot t_i^2 \cdot (1+\nu_{t_i}^2)}{2 \cdot (1-\sum_{i=1}^{k-1} \rho_i) \cdot (1-\sum_{i=1}^{k} \rho_i)} \quad (k=1,...,H),$$

where ρ_i – the load coefficient of the system ($\rho_i = \frac{\lambda_i}{\mu_i}$, λ_i – the intensity of input tasks of the i-th priority; μ_i – the intensity of task servicing of the i-th priority), t_i - the average servicing time of tasks of the i-th priority ($t_i = \frac{1}{\mu_i}$), ν_{t_i} – the coefficient of duration variation of the task servicing of the i-th priority. Let's calculate the average standby time of all priority tasks in all queues of the system under study: $W = \sum_{i=1}^{n-1} \alpha_i \cdot \sum_{k=1}^{H} w_k^{RP}$, i=1, 3, ..., n-3, n-1.

The response time of the system with TMs and the strategy of separation in space with a heterogeneous input flow of tasks with relative priorities is:

$$U = \sum_{i=1}^{n} \alpha_i \cdot u_i , \quad \text{where} \quad u_i = W_i^{RP} + \tau_{obslugi} ,$$

$$W_i^{RP} = \sum_{i=1}^{k} w_i^{RP} , \ \tau_{obslugi} = t_{TM} + \nu_{PUi} .$$

Let's consider an n-processor system with a heterogeneous tasks flow and with an absolute priority. The flow of queries for servicing as in case of relative priorities consists of H-classes of tasks $\lambda_i = \lambda_{j1} + \lambda_{j2} + ... + \lambda_{jH}$. For the i-th subsystem "TM-PU" a tasks queue with diverse queries is formed. A task taken for servicing is in the local queue until it comes to be executed in PU. The Queuing network consists of n single-channels MSSs.

In case of servicing with absolute priorities tasks are executed in the input order; however, in case of inputting a query of a higher priority for the execution it will be served out of turn interrupting the execution of the current task [8]. The flow of H-types of tasks for executing with intensities $\lambda_i = \lambda_{j1} + \lambda_{j2} + ... + \lambda_{jH}$ comes to the queue entry of the subsystem "TM-PU", where H is the priority of these queries. Suppose the query of the j-th priority is being served. The task of the i-th priority comes to the system entry. If i< j, the task is put on the queue. If i> j the query servicing of the j-th priority is interrupted and the servicing of the task of the i-th priority starts. After executing of the query of the i-th priority the task with the j-th priority proceeds servicing, if no query with a much higher priority comes.

The average servicing time of the subsystem "TM-PU" of a task flow equals $\tau_{TM} + \nu_{PU}$. The query coming at the time when the system is busy is put on the queue with the finite number of spaces *k* and is in the standby mode. Tasks out of the standby queue Oi with the probability of a failure are sent to other queues.

The average standby time for tasks with absolute priorities in the queue for the i-th system "TM-PU" are determined in accordance with the equation [8] taking into account the assumptions taken from [2]

$$w_k^{AP} = \frac{\sum_{i=1}^{H} \lambda_i \cdot t_i \cdot (1+\nu_{t_i}^2)}{2 \cdot (1-\sum_{i=1}^{k-1} \rho_i) \cdot (1-\sum_{i=1}^{k} \rho_i)} + \frac{\sum_{i=1}^{k-1} \rho_i \cdot t_i}{1-\sum_{i=1}^{k-1} \rho_i} \quad (k=1,...,H)$$

, where ρ_i is the load coefficient of the system ($\rho_i = \frac{\lambda_i}{\mu_i}$, λ_i – the intensity of a tasks input of the i-th priority; μ_i – the intensity of the task servicing of the i-th priority), t_i - the average servicing time of tasks of the i-th priority ($t_i = \frac{1}{\mu_i}$), ν_{t_i} – the coefficient of duration variation of the task servicing of the i-th priority.

Let's calculate the average standby time of all priority tasks in all queues of the system $W = \sum_{i=1}^{n-1} \alpha_i \cdot \sum_{k=1}^{H} w_k^{AP}$ i=1 , 3, ..., n-3, n-1.

The response time of the system with TMs and the strategy of separation in space with a heterogeneous inputting flow of tasks with absolute priorities is: $U = \sum_{i=1}^{n} \alpha_i \cdot u_i$, where $u_i = W_i^{AP} + \tau_{obslugi}$, $W_i^{AP} = \sum_{i=1}^{k} w_i^{AP}$, $\tau_{obslugi} = t_{TM} + \nu_{PUi}$.

4. AN EXAMPLE AND SIMULATION RESULTS

During the experiments (the modelling was carried out using the program [16, 17]) the tasks processing time changed (low - for tasks requiring a quick response, medium and high - for tasks requiring a slow response). The load of the PU was at the level of 65%, which corresponds to the average load of the system. The number of PUs ranged from 2 to 20. The tasks processing time is as follows: for tasks requiring a quick response it is 0,1 msec, for tasks with the medium processing time - 0,5 msec, finally, for the most time-consuming tasks - 1,0 msec. The quantum time for the carried out experiments is taken constant and equals 0, 1 msec. The TMs working time in tasks context changing is 5 microseconds (obtained by measuring via a system- prototype using the measurement program [16, 17]); the time of the cache memory reloading is assumed to be 5 microseconds (obtained using the test package RightMark Memory Analyzer [18]).

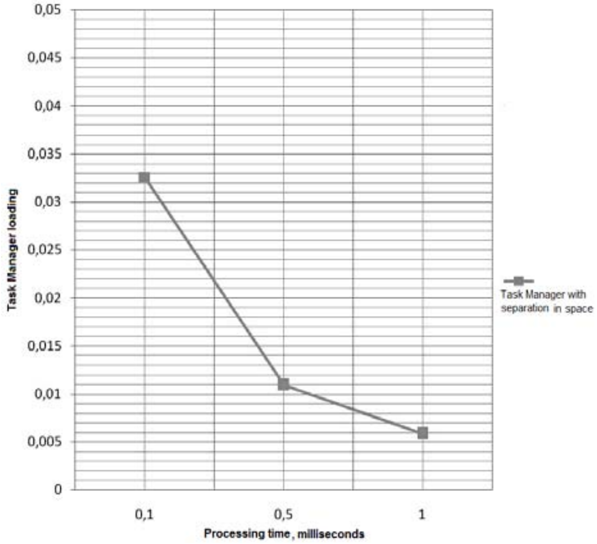

Figure-3. Dependence of the TMS load on the tasks processing time.

From the graph (Figure-3) we can see that the TMs load with the strategy of separation in space increases with a decrease of the tasks processing time (increase of the MPS response) but significantly to a less extent in comparison with TMs with strategy of separation in time with the same tasks and architectural parameters. This shows the potential high performance of TMs with the strategy of separation in space.

In Figure-4 we can see a combined graph showing the time dependence on the system response including TMs with the strategy of separation in time and space.

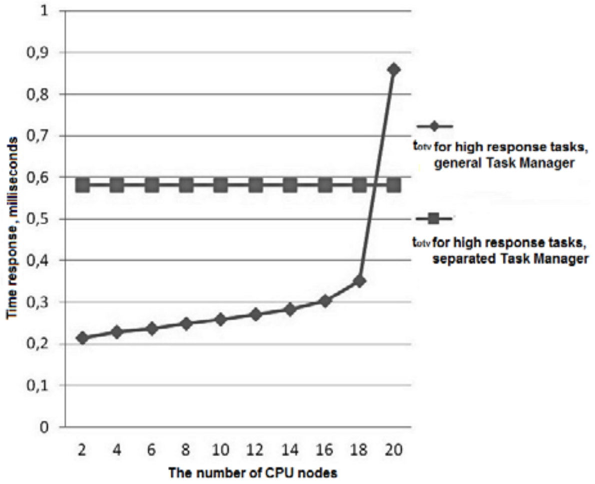

Figure-4. Dependence of MPS response time on the number of PUs with the task flow requiring system slow response.

Figure-4 illustrates that with the increase of the PUs number the response time of the system including TMs with the strategy of separation in time increases. Processing low time-consuming tasks (with quick response) it is minimal for any number of PUs and increases with their processing time. The response time of the system including TMs with the strategy of separation in space remains constant at any number of PUs in the system. The simulation results prove that TMs with the strategy of separation in time with the given architectural and tasks parameters show the best performance results in the range from 2 to 19 PUs in comparison with TMs with the strategy of separation in space.

A system with relative priorities. Consider a MPS with 3 priority tasks flows for servicing. First priority tasks are super urgent; second priority tasks are urgent; third priority tasks are common not requiring quick response and results. Queries come to one common queue for MSSs "TM-PU". According to the obtained equation we get a formula for finding the average standby time of 3 mentioned above priorities. We denote w_1^{RP} – the standby time in the queue of the first priority, w_2^{RP} – the standby time in the queue of the second priority, w_3^{RP} – the standby time in the queue of the third priority. The variation coefficient is assumed to be 1. We get the following equation:

$$w_1^{RP} = \frac{\rho_1 \cdot t_1 + \rho_2 \cdot t_2 + \rho_3 \cdot t_3}{(1-\rho_1)}; \; w_2^{RP} = \frac{\rho_1 \cdot t_1 + \rho_2 \cdot t_2 + \rho_3 \cdot t_3}{(1-\rho_1)\cdot(1-\rho_1-\rho_2)};$$

$$w_3^{RP} = \frac{\rho_1 \cdot t_1 + \rho_2 \cdot t_2 + \rho_3 \cdot t_3}{(1-\rho_1-\rho_2)\cdot(1-\rho_1-\rho_2-\rho_3)},$$

where ρ_1, ρ_2, ρ_3 – the load of the subsystem "TM-PU" by the tasks of these three priority classes ($\rho_1 = \frac{\lambda_1}{\mu_1}$, $\rho_2 = \frac{\lambda_2}{\mu_2}$, $\rho_3 = \frac{\lambda_3}{\mu_3}$), t_1, t_2, t_3 is the average servicing time of tasks of the 1st, 2nd and 3d priority class ($t_1 = \frac{1}{\mu_1}$, $t_2 = \frac{1}{\mu_2}$, $t_3 = \frac{1}{\mu_3}$). Let's calculate the average standby time of priority tasks in queues of the network

$$W = \sum_{i=1}^{n-1} \alpha_i \cdot \sum_{k=1}^{H} w_k^{RP}, \; n = 1,3,...,n-1$$

$$W = \alpha_1 \cdot (w_1^{RP} + w_2^{RP} + w_3^{RP}) + ... + \alpha_{n-1} \cdot (w_1^{RP} + w_2^{RP} + w_3^{RP}) =$$
$$= (w_1^{RP} + w_2^{RP} + w_3^{RP}) \cdot (\alpha_1 + \alpha_2 + ... + \alpha_{n-1})$$

The response time of TMs with the strategy of separation in space with a heterogeneous input flow of tasks with relative priorities system is: $U = \sum_{i=1}^{n} \alpha_i \cdot u_i$,

where $u_i = W_i^{RP} + \tau_{obslugi}$, $W_i^{RP} = \sum_{i=1}^{k} w_i^{RP}$, $\tau_{obslugi} = t_{TM} + \nu_{PUi}$.

These procedures are realized in the program [16]. The calculations results are done by the basic method; the developed method and simulation are presented in Figure-5.

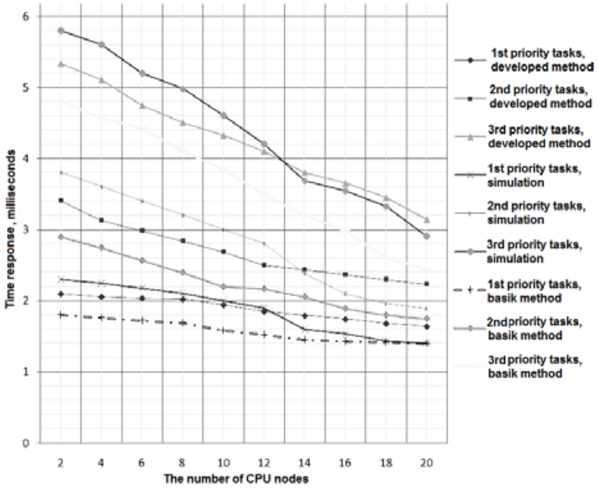

Figure-5. Dependence of response time of MPSs with TMs with separation in space and relative priorities on the number of PUs

A system with absolute priorities. Consider a similar system with 3 priority classes of tasks for servicing. According to the obtained equation, we get a formula for finding the average standby time for tasks of 3 priorities. We denote w_1^{AP} – the standby time in the queries queue of the first priority, w_2^{AP} – the standby time in the queries queue of the second priority w_3^{AP} – the standby time in the queries queue of the third priority. The variation coefficient is assumed to be 1. We get the following equation: $w_1^{AP} = \frac{\rho_1 \cdot t_1}{(1-\rho_1)}$;

$$w_2^{AP} = \frac{\rho_1 \cdot t_1}{(1-\rho_1)} + \frac{\rho_1 \cdot t_1 + \rho_2 \cdot t_2}{(1-\rho_1)\cdot(1-\rho_1-\rho_2)};$$

$$w_3^{AP} = \frac{(\rho_1 + \rho_2)\cdot t_2}{(1-\rho_1-\rho_2)} + \frac{\rho_1 \cdot t_1 + \rho_2 \cdot t_2 + \rho_3 \cdot t_3}{(1-\rho_1-\rho_2)\cdot(1-\rho_1-\rho_2-\rho_3)}$$ where

ρ_1, ρ_2, ρ_3 – the load of the subsystem "TM-PU" by the tasks of these three priority classes ($\rho_1 = \frac{\lambda_1}{\mu_1}$, $\rho_2 = \frac{\lambda_2}{\mu_2}$, $\rho_3 = \frac{\lambda_3}{\mu_3}$), t_1, t_2, t_3 - is the average servicing time of tasks of the 1st, 2nd and 3d priority class correspondingly ($t_1 = \frac{1}{\mu_1}$, $t_2 = \frac{1}{\mu_2}$, $t_3 = \frac{1}{\mu_3}$).

Let's calculate the average standby time of priority tasks in queues of the network

$$W = \alpha_1 \cdot (w_1^{AP} + w_2^{AP} + w_3^{AP}) + ... + \alpha_n \cdot (w_1^{AP} + w_2^{AP} + w_3^{AP}) =$$
$$= (w_1^{AP} + w_2^{AP} + w_3^{AP}) \cdot (\alpha_1 + \alpha_2 + ... + \alpha_n)$$

The response time of TMs with the strategy of separation in space with a heterogeneous input flow of tasks with absolute priorities is: $U^{AP} = \sum_{i=1}^{n} \alpha_i \cdot u_i$, where

$$u_i = W_i^{AP} + \tau_{obslugi}, W_i^{AP} = \sum_{i=1}^{k} w_i^{AP}, \; \tau_{obslugi} = t_{TM} + \nu_{PUi}.$$

These procedures are realized in the program [16]. The calculations results done by the basic method, the developed method and simulation are presented in Figure-6.

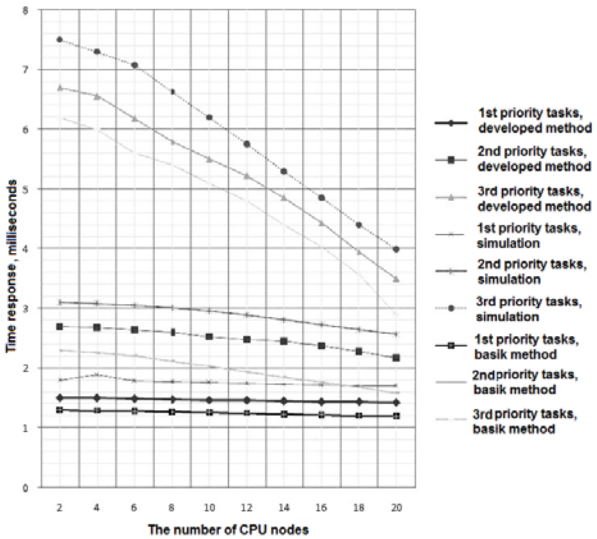

Figure-6. Dependence of response time of MPSs with TMs with separation in space and absolute priorities on the number of Pus.

5. CONCLUSIONS

Thus, for a MPS of the set real-time with a small number of PUs it is necessary to use TMs with the strategy of separation in time, with a greater number of PUs TMs with the strategy of separation in space should be used.

The simulation results show that the developed method of simulation of TMs with the strategy of separation in space shows better characteristics close to real systems than the classical method based on MSSs of the M/M/1 type. Obtained values of correlation coefficients indicate a greater adequacy of the developed method of a simulation model.

ACKNOWLEDGEMENT

The work has been done with the financial support of RFBR (Grant No. 16-07-00012 A).

REFERENCES

[1] Tanenbaum A., Bos H. 2015. Modern operating systems. The 4th Edition. SPb. Piter. p. 1120.

[2] Martyshkin A.I. 2013. Mathematical modelling of Task Managers in multiprocessor computing systems based on stochastic queueing networks: candidate thesis: 05.13.18 / Martyshkin Alexey Ivanovich Penza State Technological University. Penza. p. 160.

[3] Martyshkin A.I., Yasarevskaya O.N. 2015. Mathematical modeling of Task Managers for Multiprocessor systems on the basis of open-loop queuing networks. ARPN Journal of Engineering and Applied Sciences. 10(16): 6744-6749.

[4] Martyshkin A.I. 2012. Investigating Multi-processor Task Managers on Queuing Models. 21st Century: The Resumes of the Past and the Challenges of the Present Plus: Scientific and Methodological Journal. Penza: PGTA. (5): 139-145.

[5] Martyshkin A.I., Biktashev R.A., Vostokov N.G. 2013. The mathematical modeling of Task Managers for parallel processing systems on the basis of open queueing systems. In the World of Scientific Discoveries. 6.1(42) (Math. Mechanics. Computer science). pp. 81-101.

[6] Martyshkin A.I. 2013. Mathematical modelling of Task Managers in multiprocessor computing systems based on stochastic queueing networks: abstract of candidate thesis: 05.13.18 / Martyshkin Alexey Ivanovich Penza State Technological University. Penza. p. 23.

[7] Biktashev R.A., Martyshkin A.I. 2012. Modeling of Task Managers of Multi-processor Systems. Advances of Modern Natural Science: Scientific Theoretical Journal. (6): 83-85.

[8] Aliyev T.I. 2009. The Basics of Discrete System Modeling. SPb. SPbGU ITMO. p. 363.

[9] Matalytskiy M.A., Tichonenko O.M., Koluzayeva Ye.V. 2011. The Queuing Systems and Networks: the Analysis and Applications: The Monograph. Grodno: GrGU. p. 816.

[10] Lozhkovskiy A.G. 2012. The Queuing Theory in Telecommunications: Textbook. Odessa: ONAS named after A.S. Popov. p. 112.

[11] Abramov V. M. 2006. Stochastic Analysis and Applications. 24(6): 1205-1221

[12] Kempa Wojciech M. 2010. Stochastic Models. 26(3): 335-356.

[13] Masuyama Hiroyuki, Takine Tetsuya. 2003. Stochastic Models. 19(3): 349-381.

[14] Nadarajah Saralees. 2008. Stochastic Analysis and Applications. 26(3): 526-536.

[15] Zaharikova E.B. 2013. Computer modeling of processes in systems and queueing networks: candidate thesis: 05.13.18 / Zaharikova Elena Borisovna. Penza. p. 171.

[16] Certificate of state registration, e-number 2013611117. The program package for the calculation of probability-time characteristics of stochastic queuing networks.

[17] Certificate of state registration, e-number 2013611118. Program complex for measuring the performance of the functions of operating systems.

[18] Mikhalev V. 2012. Performance Test Results QNX Neutrino. Modern Automation Technology: Scientific and Technical Journal. (2): 82-88.