DOI: 10.24412/2413-2527-2022-331-37-45

# Cluster Load Balancing Algorithms Based on Shortest Queue Models

PhD V. A. Goncharenko, Grand PhD V. A. Lokhvitsky
Mozhaisky Military Space Academy
Saint Petersburg, Russia
vlango@yandex.ru, lokhv_va@mail.ru

***Abstract.*** **Various algorithms for redistributing tasks in cluster computing systems are described. The results of calculating the probabilistic-time characteristics of the system with connection to the shortest queue and transitions between queues are presented. A number of models with different performance and node failures, with delays in the transition between nodes are described. The results of analytical and simulation modeling of the considered systems are compared.**

***Keywords:*** **cluster, load balancing, shortest queue models, queue theory, dispatching, transition between queues, join-the-shortest-queue.**

## INTRODUCTION

Cluster technologies are currently widely used to solve the problems of ensuring the stability of the functioning and survivability of computing systems (CS) [1]. At the same time, there are cluster systems for various purposes – to increase fault tolerance by duplicating calculations (HA-clusters, High-availability), to ensure a uniform load of cluster nodes by redistributing it (LB-clusters, Load Balancing) or to ensure high performance by parallelizing calculations between cluster nodes (HPC clusters, High performance computing). It is also possible to organize the work of a computing cluster in a mixed mode – with switching functions.

Let's consider in more detail the problem of optimal load redistribution in cluster computing systems. It is relevant in solving problems of both optimizing bandwidth and increasing fault tolerance of distributed computing systems [1, 2]. Examples of such systems can be database query processing systems, Web factories, firewalls, mail and Web traffic content analysis systems, where sufficiently high response times are required.

One of the load balancing mechanisms is dispatching incoming service requests. This mechanism redistributes the workload between several servers of the cluster system, which in general may have different performance. If they fail, the load is redistributed to other nodes of the cluster. At the same time, in a distributed system, there may be delays in transferring the load from one processing node to another.

The objective of the article is to consider algorithms and analytical and simulation models of load balancing with heterogeneous cluster architecture and various methods of dispatching organization.

## ALGORITHMS FOR DISPATCHING TASKS IN CLUSTERS

Consider the models of a cluster computing system (Fig. 1), where the distribution of tasks between nodes is carried out by a hardware or software dispatcher (switching processor, specialized load balancing server, special software). Each node has the necessary means to organize a queue of tasks. The dispatcher has either a centralized or distributed implementation, when the dispatcher functions are performed in each of the nodes under consideration. The homogeneity of the CS nodes is not mandatory, i. e. nodes of different performance are allowed.

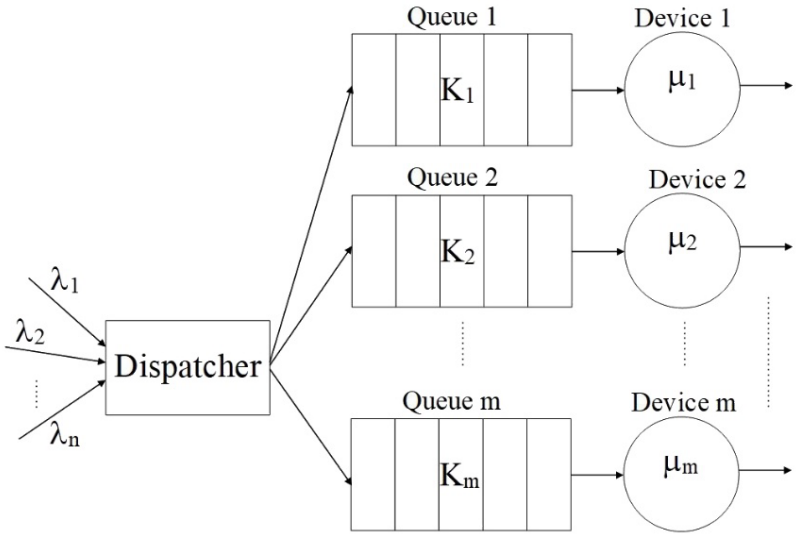


Fig. 1. Model with dispatching

There are deterministic, stochastic and adaptive dispatch algorithms.

1. *Deterministic algorithms.*

The dispatcher directs the received task to a specific server:

a) fixed dispatching (each task flow is sent to its «own» predefined server);

b) cyclic dispatching (each newly received task is sent to the next server by number, for example, in Round Robin, WRR, DRR cyclic algorithms).

2. *Stochastic algorithms*.

The dispatcher directs tasks to one of the cluster nodes with equal probability (as a generalization– with a given probability, depending on performance and other factors). The algorithm does not take into account the current degree of node load.

3. *Adaptive algorithms*.

The dispatcher directs the next incoming task based on the ratio of queue lengths to individual servers (as a generalization — based on the ratio of productivity or serviceability of servers [3].

Obviously, adaptive algorithms do a better job with load balancing [4], but require additional information. A feature of the algorithms is the possibility of making a decision on load redistribution based on operational dynamically changing information, for example, information about queue lengths to servers [5, 6].

A large number of publications are devoted to the study of the problem of the shortest queue [7–10]. For the first time such a model was considered in [11]. At the same time, there are no exact analytical calculations in the literature for models with more than two servers – approximation methods are used [12]. Thus, approximations of the average response time for the case of K queues are presented in [13], assuming that different queue lengths can differ by no more than one. The boundaries for the average residence time of requirements in a two-channel system were obtained in [7] using linear programming methods.

In [14], an approximation was developed to generalize the shortest queue model, namely, the model with the shortest expected delay in routing clients to servers with different operating speeds.

Below we will consider various strategies for organizing the work of adaptive dispatch algorithms [3]:

1. The dispatcher receives or does not receive additional information about the performance of nodes.

2. The dispatcher directs the task to the node with the shortest queue length or (if additional information is available) to the node with the lowest delay (the ratio of queue length to node performance).

3. If the queue lengths (delays) are equal, the dispatcher directs the task:

a) to the node specified for each task flow;

b) to the next node after the last node that received the task;

c) to any node with equal probability;

d) to the node with the highest performance;

e) to any node with a probability proportional to performance.

4. If the node capacities are equal, the dispatcher directs the task:

a) to the node specified for each task flow;

b) to the next node after the last node that received the task;

c) to any node with equal probability.

5. In addition to dispatching input tasks, it is possible to organize the transition of tasks between queues. After servicing the next task, when the difference between the shortest queue and the longest queues is more than ΔL (sensitivity threshold):

a) redistributes the last task of the nearest of the longest queues preceding the shortest queue to the shortest queue;

b) redistributes to the shortest queue the last task of one of the longest queues, selected equally likely;

c) no longer redistributes tasks from the longest queues.

Figure 2 shows the classification of algorithms for dispatching input tasks depending on the selected model.

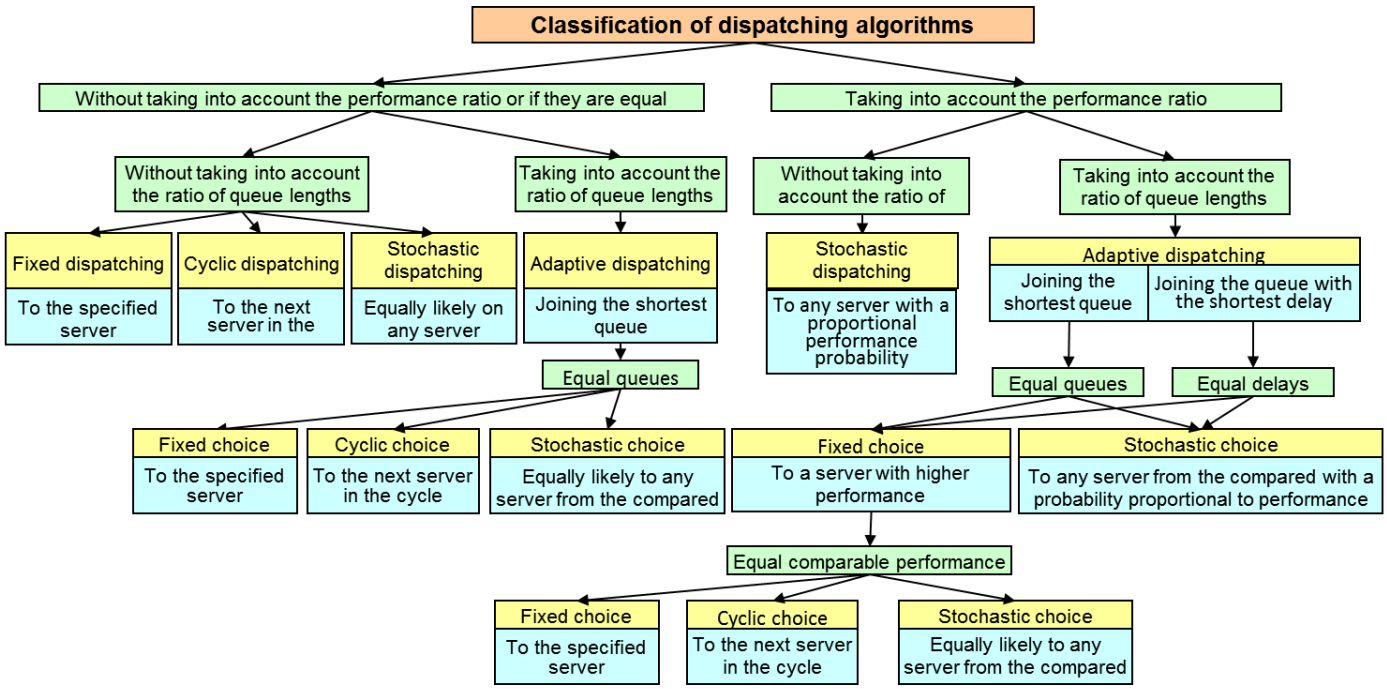


Fig. 2. Classification of dispatching algorithms

### A MODEL WITH JOINING THE SHORTEST QUEUE AND TRANSITIONS BETWEEN QUEUES

Despite the considerable interest in models with the shortest queue, the analytical results are still very modest, even with the simplest assumptions about the input flow and service flows. At the same time, models and algorithms have been developed that, in addition to joining the shortest queue, allow requests to move between queues during the waiting process. For the first time such a two-channel model was considered in [15]. In [5], expressions are obtained for the main characteristics of the model with connection to the shortest queue and transition between queues based on a two-channel system with one input flow, different channel capacities and an infinite queue. The algorithm of functioning and a device for modeling a two-channel system with connection to the shortest queue and transition between queues are described in [6].

Let's call this system a system with *join the shortest queue* and *transition between queues* — JSQ/TBQ. Consider the case of a two-channel system ($m = 2$) with a limited capacity of queue buffers $K_i$, $i = 1, 2$. Figure 3 shows the block diagram of this system [16].

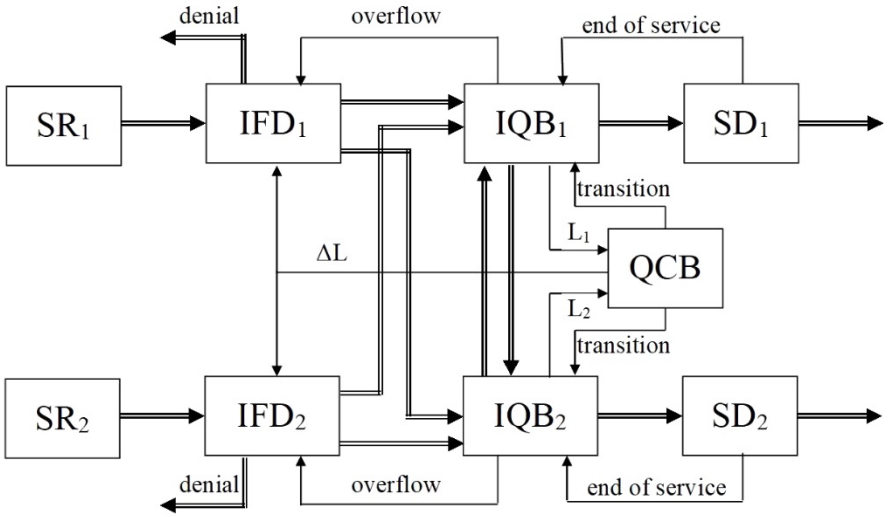


Fig. 3. Block diagram of the JSQ/TBQ system

The figure uses abbreviations:

$SR_i$ — $i$-th source of requests;

$IFD_i$ — $i$-th input flow dispatcher;

$IQB_i$ — $i$-th input queue block;

QCB — queue comparison block;

$SD_i$ — $i$-th service device.

The total input flow of requests from the sources of requests ($SR_i$) will be distributed in such a way as to load both nodes most optimally, since any task that enters the system will join the shortest queue. To do this, the input flow dispatchers ($IFD_i$) use information about the difference in the lengths of the queues of the input queue blocks ($IQB_i$) $\Delta L = L1 - L2$ from the queue comparison block (QCB). In order to reduce the difference in queue lengths that occurs during the waiting for service due to the random nature of the request service process, a mechanism for transferring requests between queues is used. We will assume that the transfer of requests from queue to queue is carried out at $|\Delta L| \geq 2$.

We describe the algorithm of the system functioning [6, 7].

*Step 1.* Requests received from $SR_i$ to $IFD_i$, depending on the state of the system:

a) are sent to the queue of the first node if $\Delta L<0$;

b) are sent to the queue of the second node if $\Delta L>0$;

c) are removed from the system if the queues are full.

*Step 2*. In cases a) and b) of step 1, the task enters the corresponding service channel and becomes in the service queue in the input queue block (IQB). In case c), the request simply does not enter the system and is deleted.

*Step 3.* The ratio of queue lengths is reported to the dispatcher by the QCB, which receives information about the lengths of queues $L1$ and $L2$ from both IQB. In case of inequality of queues depending on the signal $\Delta L$, the dispatcher sends the request to the shortest queue. If the queues are equal, then the request is sent to the channel to which it was received.

*Step 4.* In case of queue overflow, IQB signals to the dispatcher, who closes access to this channel for requests and transfers it to the neighboring channel. When both queues overflow, in addition to the overflow signal, the $IFD_i$ receives a queue equality signal from the QCB. The receipt of request s in the system is stopped until the seats in the queues are vacated.

*Step 5.* From the IQB, the request is sent to the service device ($SD_i$) for maintenance, the end of which it signals to the IQB in order to accept the next service request and replenish the queue if there was a limit number of requests in it.

*Step 6.* If there is a difference in the queue lengths of more than one request, the BSO generates a signal for the transition of the last request from a longer queue to the end of a shorter one. In the QCB, after the transfer of the request is completed, the $\Delta L$ is changed.

Thus, the alignment of queue lengths occurs not only due to the redistribution of the incoming input flow, but also due to the transfer of requests between queues. A special case of the system is with one incoming flow and one fiberboard.

The request distribution strategy can be of two types. The first type is when the ratio of the service rates of $SD_1$ and $SD_2$ is known, the second is when there is no a priori information about their ratio. In the first case, if the queues are equal, the request is sent to the queue to the SD with greater rate, in the second — with equal probability.

CALCULATION OF THE CHARACTERISTICS
OF THE SHORTEST QUEUE TWO-SERVER MODEL

Consider a two-channel system JSQ/TBQ with two input flows, queue end drives and different node performance (Fig. 4).

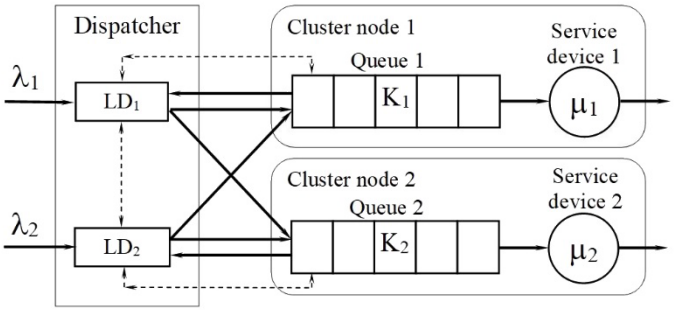


Fig. 4. Model of two-channel Queuing system JSQ/TBQ with finite storage devices

The general dispatcher is distributed, consists of local dispatchers $LD_1$ and $LD_2$, exchanging information about the status of queues. Requests come from two different input flows and are sent to the node with the smallest queue. If the queue lengths are equal, the incoming request is sent to a node with a higher service rate, if the same or unknown ratio of service rates is equal — to a node with the same number. During the waiting process, the last task from the longest queue goes to the shortest queue with a queue difference equal to the sensitivity threshold. In the simplest case, the sensitivity threshold is two. The transition time to the next queue, both when a request is received and during the waiting process, is generally not equal to zero. If both queues overflow, the incoming request is rejected.

The transition graph of the system is shown in Figure 5.

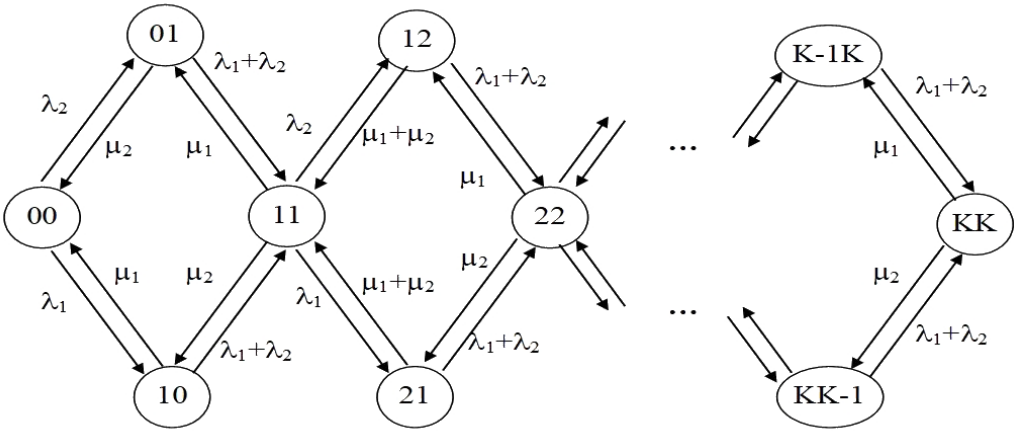


Fig. 5. The transition graph of the system

The states characterize the number of requests in each node. Each arrow is set in accordance with the rate of transitions. At the same time, the number of tasks in each node does not differ from each other by more than 1, which corresponds to the dispatching algorithm. We denote by $P_{i,i}$, $P_{i,i+1}$, $P_{i+1,i}$ the stationary probabilities of the state of the system. Based on the transition graph, in accordance with the conservation laws of queue theory [17], we will compile a system of equations and transform it to the following form:

$$\left.\begin{aligned} p_{01} &= p_{00}\frac{\rho(\lambda_1+\lambda_2)+\lambda_2}{(2\rho+1)\mu_2} \\ p_{10} &= p_{00}\frac{\rho(\lambda_1+\lambda_2)+\lambda_1}{(2\rho+1)\mu_1} \\ &\vdots \\ p_{ii} &= \rho^{2i-1}(p_{10}+p_{01}), i = 1 \div K \\ p_{i,i+1} &= \rho^{2i-1}(p_{10}+p_{01})\frac{\rho^2\mu_1+\lambda_2}{\lambda_1+\lambda_2+\mu_1+\mu_2} \\ p_{i+1,i} &= \rho^{2i-1}(p_{10}+p_{01})\frac{\rho^2\mu_2+\lambda_1}{\lambda_1+\lambda_2+\mu_1+\mu_2} \\ &\vdots \\ p_{KK} &= \rho^{2K-1}(p_{10}+p_{01}). \end{aligned}\right\} \quad (1)$$

Herein $\rho=\lambda/\mu$ — system load factor; $\lambda=\lambda_1+\lambda_2$ — the total arrival rate; $\mu=\mu_1+\mu_2$ — total service rate.

We take $\lambda_1 = r\lambda$, $\mu_1 = s\mu$, where $r$ and $s$ are the coefficients of the asymmetry of the input flow and the service flow. Based on this and the conditions $p_{ij}+p_{ji}=p_{i+j}$, $p_{ii}=p_{2i}$, we bring the system (1) to the following form:

$$\left.\begin{aligned} p_1 &= p_0\frac{\rho^2+\rho(r+s-2rs)}{(2\rho+1)(s-s^2)} \\ p_2 &= \rho\times p_1 \\ &\vdots \\ p_i &= \rho^{i-1}p_1 \\ &\vdots \\ p_{2K} &= \rho^{2K-1}p_1. \end{aligned}\right\} \quad (2)$$

From (2) and the normalization condition (the sum of all probabilities of states is equal to one), we find the probability of a free state of the system:

$$p_0 = \frac{1}{\left[1+\frac{1-\rho^{2K}}{1-\rho}\times\frac{\rho(\rho+r+s-2rs)}{(1+2\rho)(s-s^2)}\right]}. \quad (3)$$

If the node capacities are equal, the formula for $p_0$ completely coincides with the similar formula for a two-channel system *M/M/2/K*:

$$p_0 = \frac{1-\rho}{1+\rho-2\rho^{2K+1}}. \quad (4)$$

With an infinite queue accumulator, formula (3) takes the form:

$$p_0 = 1/\left[1+\frac{\rho(\rho+r+s-2rs)}{(1-\rho)(1+2\rho)(s-s^2)}\right].$$

And with equal productivity $\mu_1 = \mu_2$:

$$p_0 = (1-\rho)/(1+\rho).$$

Average response time *T* in the system under consideration:

$$T = p_0\frac{(\rho+r+s-2rs)}{\mu(1+2\rho)(s-s^2)}\times\left[\frac{1-\rho^{2K}}{(1-\rho)^2}-\frac{2K\rho^{2K}}{1-\rho}\right]. \quad (5)$$

From (5), you can get the average number of requests in the system: $N = \lambda T$.

It is also of interest what proportion of the total number of requests is served in the first and which in the second node, how it varies depending on the coefficients *s* and *r*.

The probabilities of servicing requests in the corresponding node $P_{serv1}$ and $P_{serv2}$ depend on three events – on the probability of joining the request from the common input flow to the corresponding queue, on the probability of transferring the request from the neighboring queue and the probability of transferring the request to the neighboring queue:

$$P_{serv1} = P_{join1} + P_{trans}^{2\to1} - P_{trans}^{1\to2}; \quad (6)$$

$$P_{serv2} = P_{join2} + P_{trans}^{1\to2} - P_{trans}^{2\to1}, \quad (7)$$

where $P_{join1}$ — probability of joining the input request to the first queue;

$P_{join2}$ — probability of joining the input request to the second queue;

$P_{trans}^{2\to1}$ — the probability of the transition of the request from the second stage to the first;

$P_{trans}^{1\to2}$ — the probability of the transition of the request from the first stage to the second.

$$P_{join1} = p_{01} + p_{12}+\ldots+p_{k-1k} + r\left(p_{00} + p_{11}+\ldots+p_{k-1,k-1}\right);$$

$$P_{join2} = p_{10} + p_{21}+\ldots+p_{kk-1} + (1-r)\left(p_{00} + p_{11}+\ldots+p_{k-1,k-1}\right).$$

It is clear from formulas (8)–(9) that

$$P_{serv1} + P_{serv2} = P_{join1} + P_{join2}.$$

If both queues overflow, the request will be denied service:

$$P_{serv1} + P_{serv2} + p_{den} = 1.$$

The probability of denial of service is defined as the probability that all places in the queue are occupied, i. e.

$$p_{den} = p_{KK} = p_0\times\rho^{2K-1}\times\frac{\rho(\rho+r+s-2rs)}{(1+2\rho)(s-s^2)}.$$

For the probabilities of request transitions between queues after joining the shortest queue, we have:

$$P_{trans}^{2\to1} = \left(p_{12} + p_{23} + \cdots + p_{K-1,K}\right)\times\frac{\mu_1}{\mu} = $$

$$= G\times(\rho s + 1 - r)\times s;$$

$$P_{trans}^{1\to2} = \left(p_{21} + p_{32} + \cdots + p_{K,K-1}\right)\times\frac{\mu_2}{\mu} = $$

$$= G\times(\rho(1-s)+r)\times(1-s),$$

where

$$G = p_0\rho^3\frac{(\rho+r+s-2rs)\times(1-\rho^{2K-2})}{(s-s^2)(1+2\rho)(1+\rho)(1-\rho^2)}.$$

We will determine what the parameters of the system under study should be in order to meet the requirements of the optimality of the service process. It follows from (3) that $p_0$ will be the maximum at the minimum of the function

$$z = \frac{\rho(\rho+r+s-2rs)}{(1+2\rho)(s-s^2)}.$$

Let *r* and ρ be constant. Then we have

$$\frac{dz}{ds} = \frac{\rho(1-2r)s^2 - \rho(1-2s)(r+\rho)}{(1+2\rho)(s-s^2)^2}.$$

Equate the numerator to 0:

$$\rho(1-2r)s^2 - \rho(1-2s)(r+\rho) = 0;$$

$$(1-2r)s^2 + 2(r+\rho)s - (r+\rho) = 0.$$

The first root of the quadratic equation:

$$s = \frac{\sqrt{\rho^2 - r^2 + \rho + r} - \rho - r}{1-2r}. \quad (8)$$

The second root is negative.

Thus, the maximum value of $p_0$ will be for the corresponding load with a certain ratio of node service rates determined by expression (8). Let's make a table (Table 1) the maxima $p_0$ corresponding to the optimal values of the coefficient *s* at different load factors ρ.

Table 1

The maximum values of $p_0$ at different *r* and ρ and optimal values of *s*, *K* = 25

| | | **ρ = 0.1** | **ρ = 0.5** | **ρ = 0.9** |
|---|---|---|---|---|
| ***r* = 0.1** | $s_{opt}$ | 0.309 | 0.396 | 0.427 |
| | $p_{0max}$ | 0.838 | 0.343 | 0.054 |
| ***r* = 0.3** | $s_{opt}$ | 0.414 | 0.449 | 0.464 |
| | $p_{0max}$ | 0.822 | 0.336 | 0.053 |
| ***r* = 0.5** | $s_{opt}$ | 0.500 | 0.500 | 0.500 |
| | $p_{0max}$ | 0.818 | 0.333 | 0.053 |
| ***r* = 0.7** | $s_{opt}$ | 0.586 | 0.551 | 0.536 |
| | $p_{0max}$ | 0.822 | 0.336 | 0.053 |
| ***r* = 0.9** | $s_{opt}$ | 0.691 | 0.604 | 0.573 |
| | $p_{0max}$ | 0.838 | 0.343 | 0.054 |

For r = 0.5, expression (8) does not make sense, since at this point *s* cannot be optimal, and $p_0$ cannot be greater than that of the M/M/2/K system. The boundary value will be $s = 0.5$, at which the $p_0$ of the system under study coincides with the $p_0$ of the system M/M/2/K.

We define the boundaries within which the values of the coefficients r and s should lie, so that the system under study is not inferior to the system M/M/2/K in probabilistic characteristics. To do this, we compare formulas (3) and (4) and set the condition:

$$1-\rho+(1-\rho^{2K})\times\frac{\rho(\rho+r+s-2rs)}{(1+2\rho)(s-s^2)}\le 1+\rho-2\rho^{2K+1}.$$

As a result of the transformations, we get

$$(4\rho+2)s^2-(1+2r+4\rho)s+(\rho+r)\le 0.$$

From the square inequality we obtain two roots that define the boundaries (sectors) of the optimal values of the coefficients *s* and *r*:

$$s_1=\frac{2r+4\rho+1+(2r-1)}{4(2\rho+1)}=\frac{\rho+r}{2\rho+1}; \qquad (9)$$

$$s_2=\frac{2r+4\rho+1-(2r-1)}{4(2\rho+1)}=0.5. \qquad (10)$$

Let's make a table of the boundary values of *s* and *r* for different ρ (Table 2). For example, using the values (8)–(10), we will plot a graph for ρ = 0.1 (Fig. 6).

Table 2

Boundary values of the coefficient *s*

| | **ρ = 0.1** | **ρ = 0.3** | **ρ = 0.5** | **ρ = 0.7** | **ρ = 0.9** |
|---|---|---|---|---|---|
| ***r* = 0.0** | 0.080 | 0.188 | 0.250 | 0.292 | 0.321 |
| ***r* = 0.5** | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 |
| ***r* = 1.0** | 0.917 | 0.813 | 0.750 | 0.708 | 0.679 |

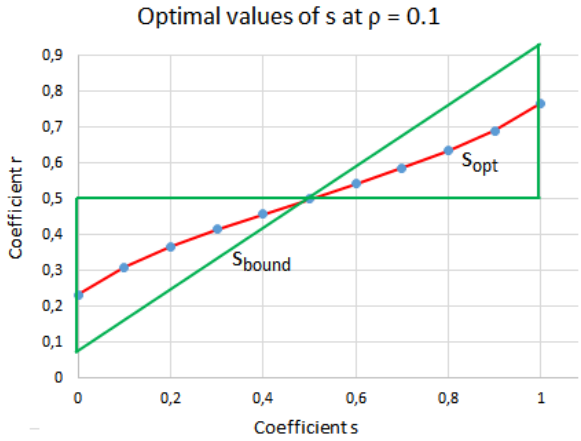


Fig. 6. Optimal values of the coefficient *s*

The graph shows that at $r = 0.5$ there is only one optimal point $s = 0.5$, at which the characteristics of the system under study and the system *M/M/2/K* coincide. As the load increases, the angle of the sector, and hence the range of optimal values of *r* and *s*, decreases.

A MODEL WITH A DELAY IN TRANSMISSION BETWEEN QUEUES

Let's now briefly consider the case when the delay in transferring tasks from the local dispatcher or from the queue to another queue is not zero. This is possible in global cluster systems, in which the transmission time is comparable to the service time in the nodes, and not taking into account this delay time will introduce a significant error in the calculations. The transfer of an request to a neighboring queue occurs when a serviced request drops out of a node with less than 1 number of request s in the queue and the difference in queues reaches 2 (Figure 7). Also, the transition occurs when the request arrives at its «own» node, in which there is 1 more in the queue than in the neighboring queue. In principle, the trigger threshold may be higher to prevent frequent transitions between queues.

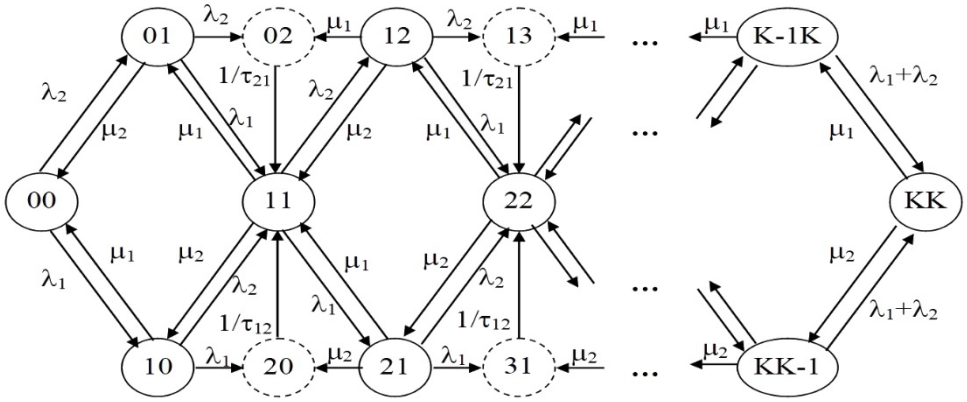


Fig. 7.The transition graph of the system

The temporary states *i* - 1, *i* + 1 and *i* + 1, *i* - 1 are marked with a dotted line, because the system, after a random delay time in transmitting a request from channel to channel ($\tau_{12}$ or $\tau_{21}$), enters the equilibrium state *i, i*. Then the rate of the transition from states *i, i* - 1 and *i* - 1, *i* to state *i, i* through intermediate states *i* - 1, *i* + 1 and *i* + 1, *i* - 1 can be expressed from the equations:

$$\frac{1}{\lambda_1^*}=\frac{1}{\lambda_1}+\tau_{12};\ \frac{1}{\lambda_2^*}=\frac{1}{\lambda_2}+\tau_{21};$$

$$\frac{1}{\mu_1^*}=\frac{1}{\mu\lambda_1}+\tau_{21};\ \frac{1}{\mu_2^*}=\frac{1}{\mu_2}+\tau_{12}.$$

We get:

$$\lambda_1^* = \frac{\lambda_1}{1+\lambda_1\tau_{12}}\ ;\ \lambda_2^* = \frac{\lambda_2}{1+\lambda_2\tau_{21}}\ ;$$

$$\mu_1^* = \frac{\mu_1}{1+\mu_1\tau_{21}}\ ;\ \mu_2^* = \frac{\mu_2}{1+\mu_2\tau_{12}}.$$

Composing a transition graph and a system of equations based on it, we find expressions for the stationary probabilities of the states of the system. So, for $\lambda_1 = \lambda_2 = \lambda$, $\mu_1 = \mu_2 = \mu$ and delay $\tau_{12} = \tau_{21} = \tau_d$ the probability of downtime $p_0$ has the form:

$$p_0 = \frac{1}{\left[1 + \frac{1-R^K}{1-R} \times (2\rho + \rho^2 \frac{2+\lambda\tau_d}{1+\lambda\tau_d}\right]}, \qquad (11)$$

where

$$R = \rho^2 \frac{2+\lambda\tau_d}{1+\lambda\tau_d} \times \frac{1+\mu\tau_d}{2+\mu\tau_d}.$$

For $\tau_d = 0$, formula (11) reduces to expression (4).

## Simulation models of adaptive dispatching simulation results

Unfortunately, not all models can be studied analytically. Therefore, during the research, the following variants of two- and three-channel simulation models of cluster systems with a finite queue storage in the GPSS World language were also developed and investigated:

1) models with the shortest queue and transition between queues;

2) models with node failures;

3) models with delayed transmission of requests between nodes;

4) models with the shortest delay in the system (the ratio of queue length to node performance).

During the simulation, various load variants were tested: subcritical (ρ = 0.5), critical (ρ = 0.95) and supercritical (ρ = 1.5; 2.0).

A comparative analysis of 4 two-channel simulation models was carried out: M1 (the system with the lowest delay), M2 (the system with the shortest queue), M3 (the M/M/2/K system), M4 (two single M/M/1/K systems). The M1–M2 models also have a mechanism for setting a non-zero delay in the transfer of requests between nodes. Three analytical models M2–M4 are also considered.

In the model with the lowest delay M1, the application is attached to the node with the lowest ratio of queue length to service intensity. In general, the model shows better results compared to the model with the shortest queue, but the implementation of the dispatcher will be more difficult due to the calculation of the node with the least delay, which may affect the decision time on the distribution of the next request.

Simulation also confirmed that it is impossible to achieve an advantage over a two-channel system with the same service intensities. Models with dispatching asymptotically approach the characteristics of the M/M/2/K system. But the gain can be achieved with a heterogeneous system architecture. In addition, models with adaptive dispatching allow us to study systems with global clustering [1].

## Conclusions

With an increase in the number of nodes and, accordingly, queues, the queue selection strategy and the analytical description of the model become much more complicated.

The inclusion in the block diagram of a model with the shortest queue of connections for the transition of requirements between queues significantly improves its characteristics, which is explained by the greater adaptability of the model to load balancing.

Response time in the system JSQ/TBQ can achieve an advantage in comparison with the M/M/2/K system with different channel capacities and a certain optimal ratio.

Taking into account the performance of nodes during load redistribution significantly improves the time characteristics of job maintenance, but complicates the implementation of the dispatcher.

## Acknowledgments

The study was carried out with the financial support of the Russian Foundation for Basic Research, project No. 18-29-22064\18.

## References

1. Zaleshchansky B. D., Chernikhov D. Ya. Klasternaya tekhnologiya i zhivuchest globalnykh avtomatizirovannykh system [Cluster technology and the survivability of global automated systems]. Moscow, Finance and Statistics Publishers, 2005. 384 p. (In Russian)

2. Dodonov A. G., Kuznetsova M. G., Gorbachik E. S. Vvedenie v teoriyu zhivuchesti vychislitelnykh system [Introduction to the theory of survivability of computing systems]. Kyiv, Naukova Dumka Publishers, 1990, 184 p. (In Russian)

3. Goncharenko V. A. Modeli adaptivnogo pereraspredeleniya nagruzki v klasternykh vychislitelnykh sistemakh [Models of Adaptive Load Redistribution in Cluster Computing Systems, *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie [Journal of Instrument Engineering]*, 2008, Vol. 51, No. 3, Pp. 32–37. (In Russian)

4. Doniants V. N., Udalova T. V. Pereraspredelenie vychislitelnoy nagruzki v lokalnykh setyakh EVM [Redistribution of Computing Load in Local Computer Networks. In: *Lazarev V. G., Chernyaev V. G. (eds.) Upravlenie protsessami i resursami v raspredelennykh sistemakh: Sbornik nauchnykh trudov [Process and resource management in distributed systems*: *Collection of scientific papers]*. Moscow, Nauka Publishers, 1989, Pp. 57–64. (In Russian)

5. Gortsev A. M. Dvukhkanalnaya sistema massovogo obsluzhivaniya s perekhodom trebovaniy iz odnoy ocheredi v druguyu [A Two-Channel Queuing System with the Transition of Requirements from One Queue to Another], *Avtomatika i telemekhanika [Automation and Remote Control]*, 1981, No. 6, Pp. 189–192. (In Russian)

6. Goncharenko V. A., Filimonikhin G. V. Ustroystvo dlya modelirovaniya dvukhkanalnoy sistemy massovogo obsluzhivaniya [Device for Modeling a Two-Channel Queuing System]. Certificate of Authorship SU No. 1509928, published at September 23, 1989, 6 p. (In Russian).

7. Halfin S. The Shortest Queue Problem, *Journal of Applied Probability*, 1985, Vol. 22, Is. 4, Pp. 865–878.
DOI: 10.2307/3213954.

8. Dester P. S., Fricker C., Tibi D. Stationary Analysis of the Shortest Queue Problem, *Queueing Systems: Theory and Applications*, 2017, Vol. 87, No. 3–4, Pp. 211–243. DOI: 10.1007/s11134-017-9556-8.

9. Adan I. J. B. F., Wessels J., Zijm W. H. M. Analysis of the Symmetric Shortest Queue Problem, *Communications in Statistics. Stochastic Models*, 1990, Vol. 6, Is. 4. Pp. 691–713. DOI: 10.1080/15326349908807169.

10. Cohen J. W. Analysis of the Asymmetrical Shortest Two-Server Queueing Model, *Journal of Applied Mathematics and Stochastic Analysis*, 1998, Vol. 11, Is. 2, Pp. 115–162. DOI: 10.1155/S1048953398000112.

11. Haight F. A. Two Queues in Parallel, *Biometrika*, 1958, Vol. 45, Is. 3–4, Pp. 401–410. DOI: 10.1093/biomet/45.3-4.401.

12. Nelson R. D., Tanatawi A. N. Approximating Task Response Times in ForkJoin Queues. In: *Gelenbe E. (ed.) High Performance Computer Systems: Proceedings of the International Symposium on High Performance Computer Systems (Paris, France, December 14–16, 1987)*. Amsterdam, North Holland Publishing Company, 1988, Pp. 157–167.

13. Nelson R. D., Philips T. K. An Approximation to the Response Time for Shortest Queue Routing, *ACM SIGMETRICS Performance Evaluation Review*, 1989, Vol. 17, Is. 1, Pp. 181–189. DOI: 10.1145/75372.75392.

14. Lui J. S. C., Muntz R. R. Algorithmic Approach to Bounding the Mean Response Time of a Minimum Expected Delay Routing System, *ACM SIGMETRICS Performance Evaluation Review*, 1992, Vol. 20, Is. 1, Pp. 140–151. DOI: 10.1145/149439.133099.

15. Saaty T. L. Elementy teorii massovogo obsluzhivaniya i ee prilozheniya [Elements of Queueing Theory: With Applications]. Moscow, Soviet Radio Publishing House, 1971, 520 p.

16. Goncharenko V. A. Analiz adaptivnykh algoritmov dispetcherizatsii zadaniy v klasterakh informatsionno-vychislitelnykh setey [Analysis of Adaptive Algorithms for Dispatching Tasks in Clusters of Information and Computing Networks]. In: *Kudryashov I. A. (ed.) Sbornik algoritmov i programm tipovykh zadach. Vypusk 24 [Collection of Algorithms and Programs for Typical Tasks. Issue 24]*. Moscow, Ministry of Defense of the Russian Federation, 2006, Pp. 222–233. (In Russian)

17. Ryzhikov Yu. I. Algoritmicheskiy podkhod k zadacham massovogo obsluzhivaniya [Algorithmic approach to queuing tasks]: Monograph. Saint Petersburg, Mozhaisky Military Space Academy, 2013, 496 p. (In Russian)

DOI: 10.24412/2413-2527-2022-331-37-45

# Алгоритмы балансировки нагрузки кластеров на основе моделей с кратчайшей очередью

к.т.н. В. А. Гончаренко, д.т.н. В. А. Лохвицкий
Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия
vlango@mail.ru, lokhv_va@mail.ru

***Аннотация.*** **Описаны различные алгоритмы перераспределения заданий в кластерных вычислительных системах. Приведены результаты расчета вероятностно-временных характеристик системы с присоединением к кратчайшей очереди и переходами между очередями. Описан ряд моделей с разными производительностями и отказами узлов, с задержками при переходе между узлами. Выполнено сопоставление результатов аналитического и имитационного моделирования рассматриваемых систем.**

***Ключевые слова:*** **кластер, балансировка нагрузки, модели с кратчайшей очередью, теория очередей, диспетчеризация, переход между очередями, присоединение к кратчайшей очереди.**

Литература

1. Залещанский, Б. Д. Кластерная технология и живучесть глобальных автоматизированных систем / Б. Д. Залещанский, Д. Я. Чернихов. — Москва: Финансы и статистика, 2005. — 384 с.

2. Додонов, А. Г. Введение в теорию живучести вычислительных систем. / А. Г. Додонов, М. Г. Кузнецова, Е. С. Горбачик; АН УССР, Ин-т проблем регистрации информации. — Киев: Наукова думка, 1990. — 181 с.

3. Гончаренко, В. А. Модели адаптивного перераспределения нагрузки в кластерных вычислительных системах // Известия высших учебных заведений. Приборостроение. 2008. Т. 51, № 3. С. 32–37.

4. Донианц, В. Н. Перераспределение вычислительной нагрузки в локальных сетях ЭВМ / В. Н. Донианц, Т. В. Удалова // Управление процессами и ресурсами в распределенных системах / АН СССР, Ин-т проблем передачи информации; отв. ред. В. Г. Лазарев, В. Г. Черняев. — Москва: Наука, 1989. — С. 57–64.

5. Горцев, А. М. Двухканальная система массового обслуживания с переходом требований из одной очереди в другую // Автоматика и телемеханика. 1981. № 6. С. 189–192.

6. Авторское свидетельство № 1509928 СССР, G 06 F 15/20. Устройство для моделирования двухканальной системы массового обслуживания: № 4364699/24-24: заявл. 13.01.1988: опубл. 23.09.1989 / Гончаренко В. А., Филимонихин Г. В.; заявитель Военный инженерный краснознаменный институт имени А. Ф. Можайского. — 6 с.

7. Halfin, S. The Shortest Queue Problem // Journal of Applied Probability. 1985. Vol. 22, Is. 4. Pp. 865–878. DOI: 10.2307/3213954.

8. Dester, P. S. Stationary Analysis of the Shortest Queue Problem / P. S. Dester, C. Fricker, D. Tibi // Queueing Systems: Theory and Applications. 2017. Vol. 87, No. 3–4. Pp. 211–243. DOI: 10.1007/s11134-017-9556-8.

9. Adan, I. J. B. F. Analysis of the Symmetric Shortest Queue Problem / I. J. B. F. Adan, J. Wessels, W. H. M. Zijm // Communications in Statistics. Stochastic Models. 1990. Vol. 6, Is. 4. Pp. 691–713. DOI: 10.1080/15326349908807169.

10. Cohen, J. W. Analysis of the Asymmetrical Shortest Two-Server Queueing Model // Journal of Applied Mathematics and Stochastic Analysis. 1998. Vol. 11, Is. 2. Pp. 115–162. DOI: 10.1155/S1048953398000112.

11. Haight, F. A. Two Queues in Parallel // Biometrika. 1958. Vol. 45, Is. 3–4. Pp. 401–410. DOI: 10.1093/biomet/45.3-4.401.

12. Nelson, R. D. Approximating Task Response Times in ForkJoin Queues / R. D. Nelson, A. N. Tantawi // High Performance Computer Systems: Proceedings of the International Symposium on High Performance Computer Systems (Paris, France, 14–16 December 1987) / E. Gelenbe (ed.). — Amsterdam: North Holland Publishing Company, 1988. — Pp. 157–167.

13. Nelson, R. D. An Approximation to the Response Time for Shortest Queue Routing / R. D. Nelson, T. K. Philips // ACM SIGMETRICS Performance Evaluation Review. 1989. Vol. 17, Is. 1. Pp. 181–189. DOI: 10.1145/75372.75392.

14. Lui, J. S. C. Algorithmic Approach to Bounding the Mean Response Time of a Minimum Expected Delay Routing System / J. S. C. Lui, R. R. Muntz // ACM SIGMETRICS Performance Evaluation Review. 1992. Vol. 20, Is. 1. Pp. 140–151. DOI: 10.1145/149439.133099.

15. Саати, Т. Л. Элементы теории массового обслуживания и ее приложения = Elements of Queueing Theory: With Applications / Пер. с англ. Е. Г. Коваленко; под ред. И. Н. Коваленко. — 2-е изд. — Москва: Советское радио, 1971. — 520 с.

16. Гончаренко, В. А. Анализ адаптивных алгоритмов диспетчеризации заданий в кластерах информационно-вычислительных сетей // Сборник алгоритмов и программ типовых задач / Под ред. И. А. Кудряшова. Вып. 24. — Москва: Министерство обороны РФ, 2006. — С. 222–233.

17. Рыжиков, Ю. И. Алгоритмический подход к задачам массового обслуживания: Монография. — Санкт-Петербург: ВКА им. А. Ф. Можайского, 2013. — 496 с.