AMSD-2019 IOP Publishing
Journal of Physics: Conference Series **1441** (2020) 012175 doi:10.1088/1742-6596/1441/1/012175

Simulation of data distribution between server stations using fuzzy technologies

E O Vikulov[1], **O V Denisov**[1], **V A Meshcheryakov**[2] **and L A Denisova**[1]

[1]Omsk State Technical University, 11, Pr. Mira, Omsk, 644050, Russia
[2] Siberian State Automobile and Highway University (SibADI), 5, Pr. Mira, Omsk, 644080, Russia

e-mail: vikuloveo@gmail.com, denisova@asoiu.com, meshcheryakov_va@sibadi.org

Abstract. This article is about development and modeling of data distribution system through the server stations. An algorithm for selecting a server for downloading data based on fuzzy logic is proposed. A simulation model of a server complex created by MATLAB / Simulink / SimEvents software is presented. This model allows us to study the server complex as a system with discrete states based on the theory of queuing systems. Numerical experiments on the distribution of data among servers using different algorithms for selecting a server for user loading data were carried out. The advantage of the proposed approach to server selection based on fuzzy inference is shown. This advantage is expressed in reducing the length of queues in the system and increasing the number of serviced requests for data processing by servers.

1. Introduction

One of the most important problems of the modern client-server applications is balancing the computational load and increasing the performance of the server complex. Load balancing can be performed using specially developed software. There are various load balancing algorithms for server stations, for example, based on a circular algorithm Round Robin, weighted circular distribution, based on the current server load [1, 2]. The easiest and most popular load balancing algorithm is Round Robin [1, 2]. This algorithm allows the system to distribute computing tasks between server stations in turn (“in a circle”). This helps to efficiently allocate resources and reduce the processing time for each request. A weighted circular distribution algorithm is more efficient [1, 2]. It lies in the fact that each server is assigned a weighting factor, and servers with higher ratings (weights) are getting more requests sent to them. But these algorithms (as well as the distribution algorithm based on the current load [3]) do not use (fully or partially) the parameters and capabilities of the server stations. We decided to develop our own load balancing system, to solve the problem of load distribution. This system will consider the state of the server stations.

2. Problem statement

The purpose of the study is to increase the efficiency of high-loaded information processing systems by rational distribution of the computational load between servers and ensuring fast information delivery to the user. In order to implement load balancing between server stations, it seems to us appropriate to develop load balancing algorithms based on fuzzy inference. This approach to load balancing will consider the state of the server stations. If a fuzzy inference is used in decision making process, the balancer will be able to choose a server for downloading files in the conditions of incomplete information [4]. Such a balancing system will allow users to speed up work with servers, as it will reduce the time spent on downloading and reading files. In order to develop and explore the load balancing between the server station, we need to create a simulation model of the server complex.

This model should provide the ability to perform the study of the server complex as a system with discrete states based on the theory of queuing systems. Using the simulation model, we will be able to conduct experiments on the distribution of data with different server selection algorithms.

3. Theory. Application of the queuing systems for server complex research

To describe objects that function under conditions of random factors, such as a server complex, a class of mathematical models called queuing systems (QS) is used [4, 5]. The functioning of these objects is in the nature of service applications received in the system. To perform a set of actions included in the concept of “maintenance”, the system has sets of equipment, called service channels or lines. In our case, such channels are channels for submitting requests to servers, switching between which is performed using the balancer server. We assume that service requests form a stream, that is, a sequence of requests with alternating moments of their appearance in time. The notion of flow based on the assumption that an event will occur at previously unknown random times. Simulation modeling of queuing systems consists of modeling the flow of requests and the operation of service channels [4, 5]. To describe the QS, we must specify:

- the incoming flow of applications that come to the service;
- the order of application to the queue and selection from it;
- the rule by which the service is performed (the algorithm of the balancer server).

To describe the incoming stream of applications, it is necessary to describe the time of their arrival in the server complex. The distribution law can be deterministic or probabilistic. In our case, the incoming flow of applications is described by the probability distribution of time intervals between neighboring applications. The rules for servicing requests by servers are characterized by the duration of service (distribution of service time) and discipline of service. To characterize the properties of each line, the duration of service of the request by the server or the time the server is busy is set as a random variable with a given distribution law. In our case, we have assumed that the servicing system consists of three lines (in accordance with the number of servers) capable of servicing the bids simultaneously and independently of each other. At any time, the server is in one of two states: idle or busy. When there is a queue of applications in the system, we will use the following rule for accepting applications. Applications are accepted for service in turn. The vacated line starts servicing the application that previously entered the system. This discipline of servicing applications is called “I’ve done it before - I’ve served it before. The output parameters of the server complex as a queuing system are values characterizing the quality of operation. Such parameters as:

- maximum and average length of queues in the system;
- the time the applications are located in queues and service channels;
- number of serviced requests by servers.

The simulation model of the server complex as a QS allows us to investigate the behavior of the system and its state changes over time for given flows of applications arriving at the system inputs. In our simulation model, it should be possible to investigate the distribution of data across servers using different server selection algorithms for the user to load data.

4. Development of data distribution imitation model

In order to develop and explore the load balancing system, we have created a simulation model for the distribution of data between the server station, which was implemented using MATLAB / Simulink / SimEvents package development tools [3-6]. Used MATLAB / Simulink environment provides visual modeling tools that have flexibility in constructing the models and allow users to assemble models from ready-made blocks [7, 8]. The application of the MATLAB / SimEvents package makes it possible to simulate systems with discrete states based on queuing theory and queuing systems (QS) [3, 6 - 8]. The SimEvents library allows us to create simulation models for passing an object through networks and queues, provides simulation of systems that depend not only on time, but also on discrete states. The object in our case is understood as the user's request to upload files and download files from the server to the user's computer. In addition, this model allows us to consider the variable values of the delay in the processing and transmission channels. The total processing time of requests by server changes, depending on server load and file processing time.

We propose to perform a load balancing using the methods of fuzzy logic [3]. The use of fuzzy logic allows us to make a decision (in a server selection process) in conditions where the data on the object parameters are not sufficiently defined. We will consider such server station parameters as remoteness from the user and server available resources. Let us introduce the following notation: parameter D_s – distance from the user uploading the file to the server station S_1, relates unit; U_{hd} – parameter means hard disk load, %; U_{ram} – means the amount of RAM used, %. These are the input signals for decision making process. We will enter a designation for the output parameter: this is R_1 – the decision to choose an S_1 server to download a file that may have values: P_1 – positive; Z_1 – Neutral; N_1 – negative.

Let's consider the scheme of the mathematical model of the server complex with load distribution using the balancer server, which is shown in figure 1. The model is assembled from SimEvents and Simulink library blocks. For each of the blocks, we can bring up dialog boxes for setting properties and parameters [7, 8]. The source of applications (entities) in the model is the Time-Based Entity Generator block. The distribution law and the average time between the occurrence of requests are specified in the parameters of the random number generation block Event-Based Random Number0.

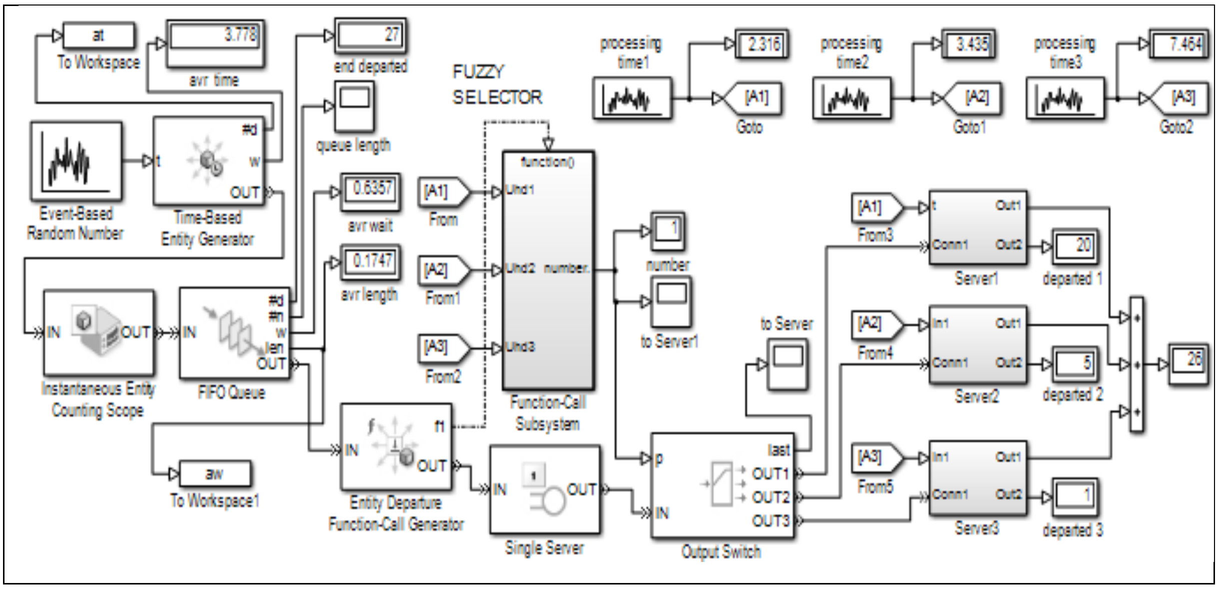


Figure 1. Scheme of the server complex model

Applications are sent to the FIFO Queue block, from where they are transferred to one of the three server stations, which are represented by the Server 1 ... Server 3 subsystems, through the Automatic Switch output block [9, 10]. A diagram of the mathematical model of the FUZZY SELECTOR subsystem, which makes a fuzzy selection of a server station during load distribution using a balancer server, is shown in figure 2. The FUZZY SELECTOR subsystem, in turn, incorporates three Fuzzy Server Selection S_1 – S_3 subsystems (for each of the three servers), making a fuzzy logical conclusion about the suitability of servers for downloading files. The input signals to each of the Fuzzy Server Selection S_1 – S_3 subsystems in these subsystems are converted to fuzzy variables. This procedure is carried out in the fuzzification subsystems Input MF. A diagram of one of the Input1 MF subsystems (for the Ds parameter of the S_1 server) is shown in Figure 3. Fuzzification of the remaining input parameters (U_{hd}, U_{ram}) for each of the servers is similar. The ranges of change of all input variables are represented by the terms of the values: S – small, M – average, L – large.

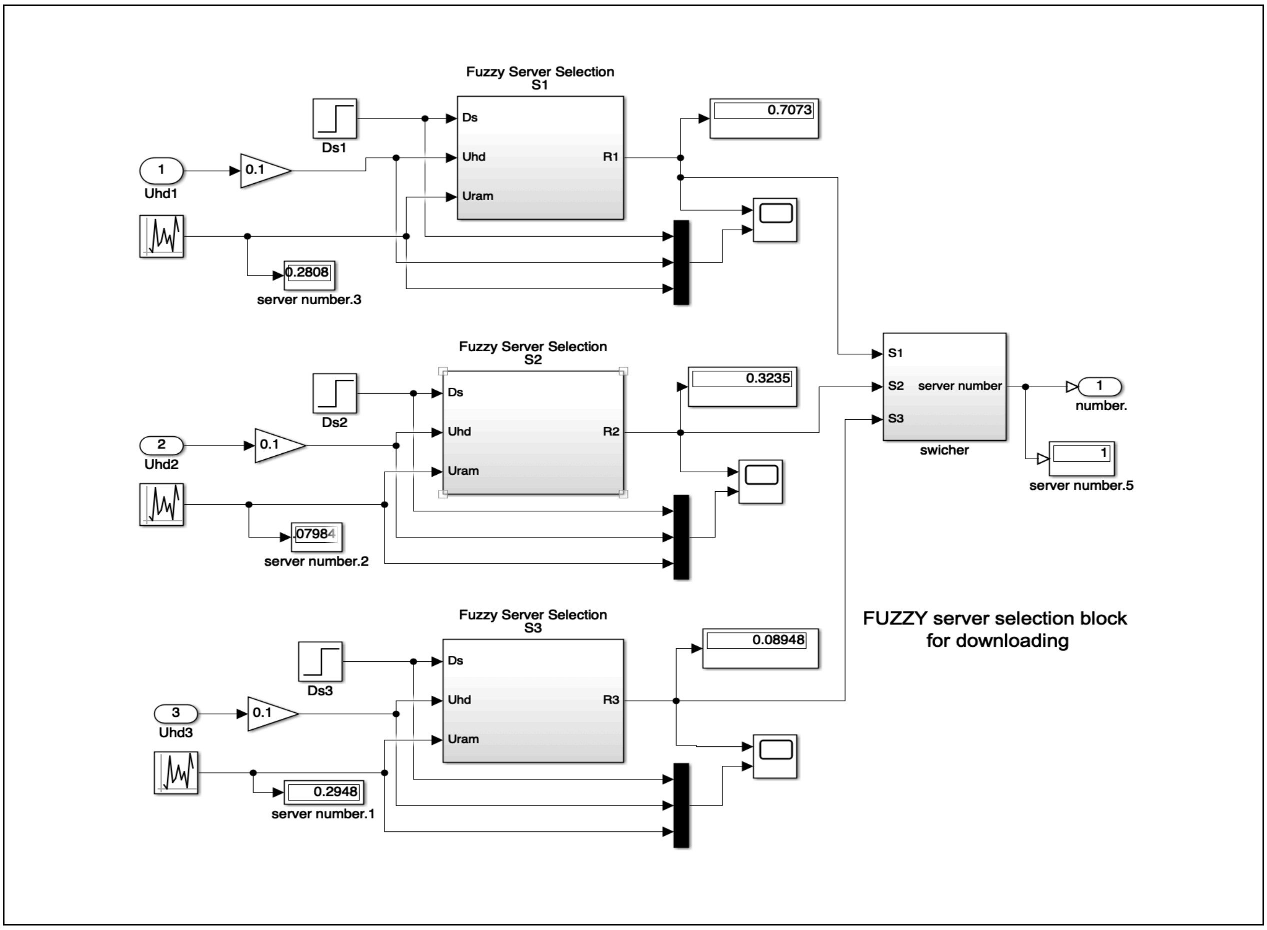


Figure 2. Fuzzy server selection subsystem FUZZY SELECTOR

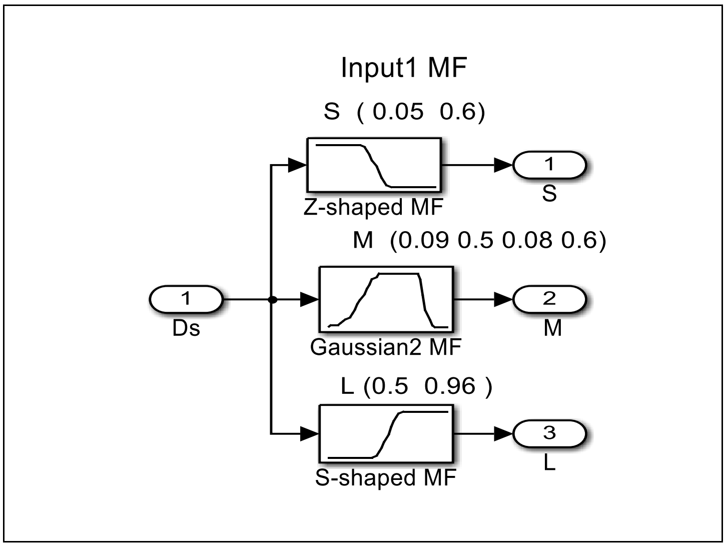
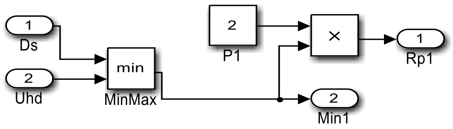
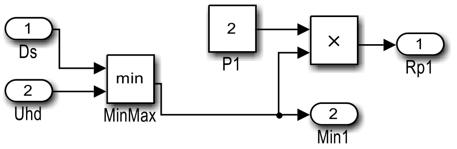
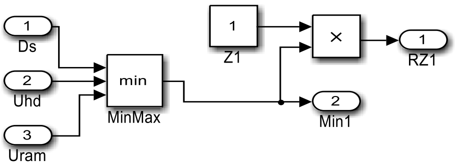
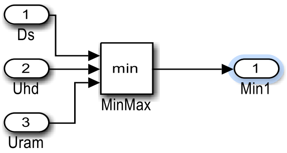
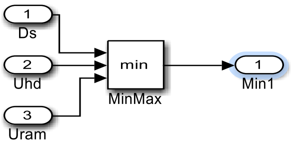
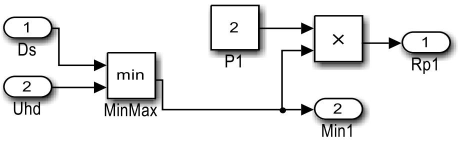
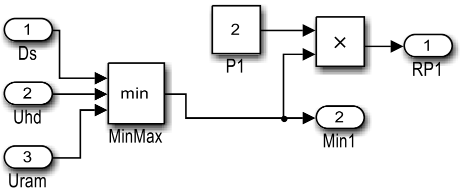
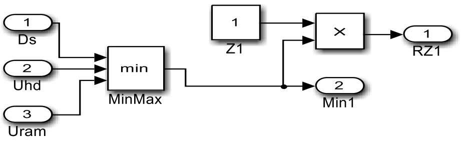
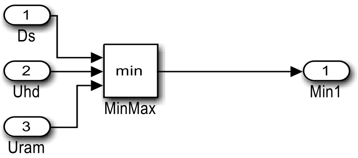
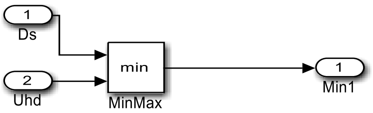


Figure 3. Fuzzification subsystem Input1 MF

The base of fuzzy rules for the selection of the S_1 server (as well as S_2, S_3), which is implemented in the Fuzzy Server Selection S_1 subsystem, is presented in table 1. It also shows the implementation of each fuzzy production rule. A number R_I (the value of the server's fitness for downloading files), specifying the conclusion of each rule, takes the values $N_I = 0$; $Z_I = 1$; $P_I = 2$.

Table 1. Compliance with fuzzy inference rules and schemes for their implementation

Fuzzy rule for S_1 server	*Schem of fuzzy rule implementation*
RULE1: ***IF*** $D_s = S$ ***AND*** $U_{hd} = S$, ***THEN*** $R_1 = P_1$;	
RULE2: ***IF*** $D_s =S$ ***AND*** $U_{hd} = M$, ***THEN*** $R_1 = P_1$;	
RULE3: ***IF*** $D_s = S$ ***AND*** $U_{hd} = L$ ***AND*** $U_{ram} = S$, ***THEN*** $R_1 = Z_1$;	 1 Ds 2 Uhd 3 Uram min MinMax 1 Z1 × 1 RZ1 2 Min1
RULE4: ***IF*** $D_s = S$ ***AND*** $U_{hd} = L$ ***AND*** $U_{ram} = M$, ***THEN*** $R_1 = N_1$;	
RULE5: ***IF*** $D_s = S$ ***AND*** $U_{hd} = L$ ***AND*** $U_{ram} = L$, ***THEN*** $R_1 = N_1$;	
RULE6: ***IF*** $D_s = M$ ***AND*** $U_{hd} = S$, ***THEN*** $R_1 =P_1$;	
RULE7: ***IF*** $D_s = M$ ***AND*** $U_{hd} = M$ ***AND*** $U_{ram} = S$, ***THEN*** $R_1 = P_1$;	 1 Ds 2 Uhd 3 Uram min MinMax 2 P1 × 1 RP1 2 Min1
RULE8: ***IF*** $D_s = M$ ***AND*** $U_{hd} = M$ ***AND*** $U_{ram} =M$, ***THEN*** $R_1 = Z_1$;	 1 Ds 2 Uhd 3 Uram min MinMax 1 Z1 × 1 RZ1 2 Min1
RULE9: ***IF*** $D_s = M$ ***AND*** $U_{hd} = M$ ***AND*** $U_{ram} = L$, ***THEN*** $R_1 = N_1$;	
RULE10: ***IF*** $D_s =M$ ***AND*** $U_{hd} =L$, ***THEN*** $R_1= N_1$;	

Further, fuzzy variables are used in fuzzy logical inference: they are processed according to the production rules. Based on the processing of input parameters for each server S_1, S_2 and S_3, the results were obtained in the form of the degree of belonging to positive solutions set. After this, the Switch subsystem made the final decision on the appropriate server number. Switch subsystem choose the server with maximum value of R_1 (value of server availability for uploading files).

5. Results of experimental studies

After execution of field experiments [2], we conducted experimental studies of data distribution methods for server stations using simulation modeling. Experiment was conducted on the distribution of file download requests, based on the data received from real server stations. We considered load balancing using four distribution methods: Equiprobable, RoundRobin (circular distribution), First Port that is not blocked, Fuzzy Logic. We present the source state data of server stations in table 2. This data received from field experiments (real server stations). Values of U_{hd}, U_{ram} were divided to minimum and maximum values, and distance is a normalized permanent parameter for all set of experiments. These data were used for our simulation model.

Table 2. The source state data of server stations

Server	Server performance parameters				
	$U_{hd\ min}$	$U_{ram\ min}$	$U_{hd\ max}$	$U_{ram\ max}$	D
S_1	0,2 - 0,3	0,3 - 0,4	0,8 - 0,9	0,6 - 0,7	0,7
S_2	0,1 - 0,2	0,6 - 0,7	0,4 - 0,5	0,9 - 1	0,75
S_3	0,1 - 0,2	0,2 - 0,3	0,99 - 1	0,9 - 1	0,8

Figure 4 shows the results of data distribution modeling. Note that the average length of the request queue for the fuzzy logic method is the smallest. This suggests that this method can be used to distribute data. In order to confirm the results of the numerical experiment, we will conduct similar calculations for the results of the operation of server stations during different periods of observation. We have data from field experiments, when we received information about the operation of the server complex during one to three hundred days. The results of the system functioning simulation are presented in Figure 5. The average queue length for the method of fuzzy logic is the smallest. This suggests that the balancing method based on fuzzy logic inference allows us to distribute data among servers better than other methods: the performance of the server complex has increased.

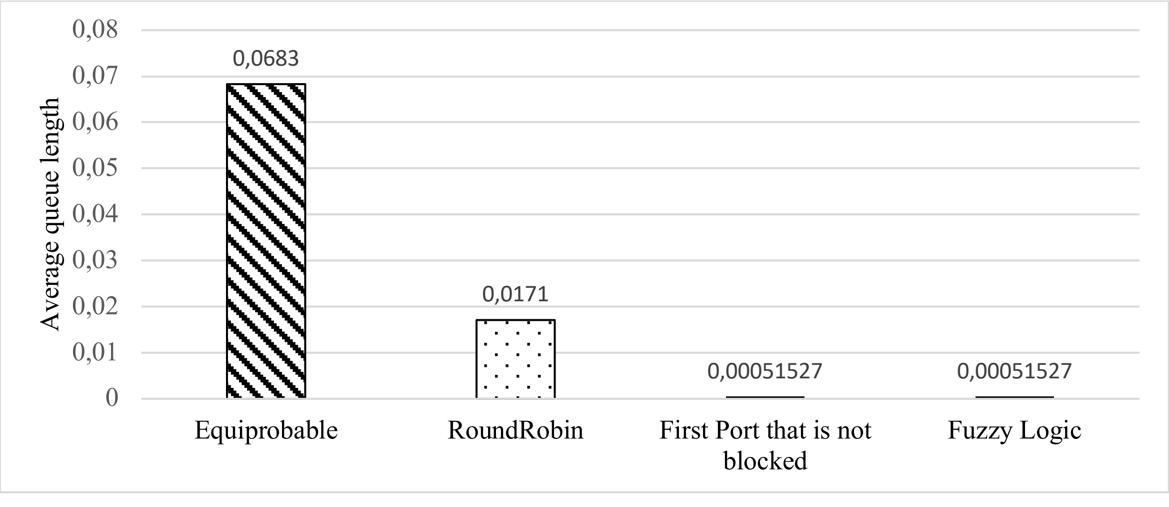


Figure 4. Dependence of the average queue length on the distribution method.

Figure 5 shows the results of data distribution for server station indicators collected in the interval from 1 to 300 days.

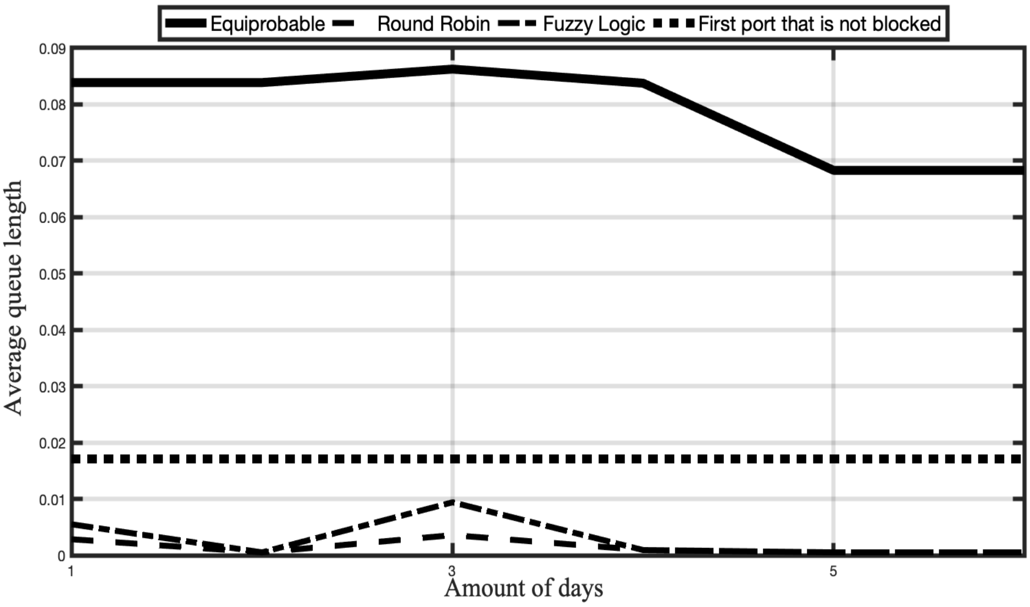


Figure 5. Results of experiments.

The obtained results show that the developed method is the fastest, since the average queue length is as small as for the method distributing applications for the first free port. Based on the obtained results, we concluded that the selected software tools MATLAB / Simulink / SimEvents / Stateflow are suitable for modeling the operation of the server complex. It should be noted that the developed simulation model makes it possible to investigate the capabilities of the balancer server in distributing the load between the server stations, taking into account the random nature of the moments of receipt of applications for processing, as well as the variable duration of delays in receipt of tasks. That is, such a model represents an opportunity to test the server balancer software for server stations, in conditions close to reality. In addition, we consider it promising to use the proposed approach to the development of simulation models for testing and analyzing functional correctness, as well as the reliability of the distribution of the load of server stations in various operating modes.

6. Conclusion

We developed and researched load balancing system using simulation modeling. We proposed to use fuzzy logic to develop an algorithm for selecting a server for downloading data by the user. For the study, we created a simulation model of the server complex using the software tools MATLAB / Simulink / SimEvents. The study of the server complex was carried out on the basis of the theory of queuing systems. We conducted numerical experiments on the distribution of data across servers using different algorithms for selecting a server. We have shown the advantage of approach to choosing a server based on fuzzy inference, which is expressed in reducing the length of queues (arising in the server complex) and increasing the number of requests served by servers.

References

[1] Vikulov E, Denisov O, Meshcheryakov V 2018 Event-Driven Simulation of Server Stations Load Balancing *International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)* Moscow, Russia.

[2] Vikulov E O, Denisov O V and Denisova L A 2018 Data distribution system: preparation of server stations data IOP Conference Series: Mechanical Science and Technology Update (MSTU-2018) *J. Phys.: Conf. Ser.* **1050** 012097

[3] Vikulov E O, Denisova L A 2019 Data distribution system: clustering based on neural network technologies *J. Phys.: Conf. Ser.* **537** 052030

[4] Suherman, Zarlis M, Stirous S and Al-Akaidi M 2016 Applying a rateless code in content delivery networks *IOP Conference Series: Materials Science and Engineering* vol 237 conf 1
[5] Membrey P, Plugge E and Hows D 2013 *Practical Load Balancing: Ride the Performance Tiger (Expert's Voice in Networking)*
[6] Wilson S, Prakash P and Deepalakshmi P 2017 Server-based Dynamic Load Balancing *IEEE Networks & Advances in Computational Technologies (NetACT) International Conference* (Thiruvanthapuram)
[7] Shtovba S 2007 *Designing fuzzy systems using MATLAB* (Moscow: Hot line–Telecom) 187-223
[8] Zadeh L A 1965 Fuzzy sets *Information and Control* **8** pp 338-53
[9] Donald Gross, John F. Shortie, James M. Thompson, Carl M 2013 *Harris Fundamentals of Queueing Theory* John Wiley & Sons **1002** 9781118625651
[10] Zadorozhnyi V N 2016 Optimization of uniform non-Markov queueing networks using resources and transition probabilities redistribution *Communications in Computer and Information Science* **638** P. 366-381.