

Model of the peer-to-peer distributed system for securable information storage and processing without traffic prioritization (TheOoL project)

Alexey V. Nenashev¹, Alexandr Yu. Tolstenko¹ and Rostislav S. Oleshko¹

¹ Samara State Technical University, Address, Samara, Index, Russia

Abstract

The mathematical model “Peer-to-peer distributed system for securable information storage and processing in enterprise networks” is described hereinafter. It is a versatile distributed operating system designed for the protection of distributed computing and insulation of private networks without restricting the possibilities of effective interactions, cryptographic security, protection from unauthorized access with the application of biometry and an innovative protocol of data exchange for topology control based on distributed ledger technology. The modeling was performed with the purpose of evaluation of performance of the system depending on productivity of the hardware of its nodes and the network’s telecommunications equipment.

Keywords

Peer-to-peer, Distributed computing networks, Enterprise networks, Cybersecurity, Queueing network, Simulation of queueing networks, compute node, multimedia data, queues, distributed data storage, Securable Information Storage)

1. Introduction

Mainstream systems for data storage and data protection are built, for the most part, using centralized architecture or they have operation centers, the control over which may be intercepted via hardware- or software vulnerabilities or by way of planting a mole in the technical staff. Also normally for data calculation and data storage they do not use advanced laptops and personal computers installed on user workplaces (workstations) featuring significant computing resources (terabytes of ROM [2], dozens of gigabytes of RAM [3], multicore high performance processors). Corporate information systems are created using multitier architecture that relays computing load to data center resources, while local resources of workstations remain largely untapped. To solve the problem of efficient utilization of computing resources systems for distributed data storage and distributed computing are being designed and developed. These systems do not resolve the issues of cybersecurity, leaving these issues at the mercy of specialist software vendors. As a result, the indicators of speed and reliability decline, while security vulnerabilities remain, which is caused by possibly incomplete documentation of information systems (IS) protected or incidental and/or intentional errors in the implementation of security systems. In our view, a possible solution would be a system that integrates the system of distributed data storage and data processing with subsystems for unauthorized access protection (UA), cryptographic protection (CP), automatic maintenance and investigation of cybersecurity incidents, and which hides topology of the network, inter alia, from internal corporate personnel. Such a system not only would provide for the ultimate protection of user data, but would reduce corporate expenditure on information technology infrastructure, which has been demonstrably proven in the following work [1]. That said a significant

III International Workshop on Modeling, Information Processing and Computing (MIP: Computing-2021), May 28, 2021, Krasnoyarsk, Russia

EMAIL: alexvlnenashev@gmail.com (Alexey Nenashev); tolstenko.ay@samgtu.ru (Alexandr Tolstenko); zonaTostATY@gmail.com (Rostislav Oleshko)

ORCID: 0000-0003-1348-1766 (Alexey Nenashev); 0000-0001-6620-8793 (Alexandr Tolstenko); 0000-0002-9068-2580 (Rostislav Oleshko)



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

challenge in the design and implementation of such a system is ensuring its high-speed performance from the perspective of its end user.

2. Peer-to-peer Distributed System for Securable Information Storage and Processing

The peer-to-peer distributed system for securable information storage and processing (“the system”) is first and foremost designed for removal of cybersecurity threats to servers, central computation and subscriber nodes within a network. It is a distributed operating system, in which each node gives away its computing resources (storage subsystem, central processor and graphic processor, and random-access memory) to the system and has no independent meaning. However, nodes vary in terms of their functional purposes: subscriber node (it can simultaneously serve as a storage node or metadata node), storage node, and metadata node. This functional purpose is assigned at the stage of implementation of the system and can be changed automatically later. The assignment of a role to a node occurs under the control of an automatic maintenance subsystem without participation of the owner of a specific node. Data storage in the system is performed in such a manner that any user data block (a file) is divided into N identical packets, then it is encrypted and submitted for storage to $K \cdot N$ network nodes (K – redundancy coefficient) in encrypted form. The list of block storage nodes is placed in the metadata block, which is, in turn, encrypted and placed in metadata nodes. No one, including the owner, knows in which nodes, at a particular time, parts of the file are stored, except for the maintenance subsystem that has access to metadata.

When designing the system it is necessary to bear in mind that at any discrete instant of time significant volumes of information must not only be passed between nodes of the system with high speed, but they must also undergo encryption and decryption procedures. It is also necessary to consider additional dataflow of service blocks of the distributed ledger (metadata) which contain data about writing/reading nodes for data and information of the data access control system. If the data processing speed turns out to be insufficiently high, the system would not be able to deliver comfortable user experience, which would, in turn, put into question the possibility of application of this system in a real-life enterprise network. To assess the possibility of implementing the system with sufficient data access speed let’s build its mathematical model. For clarity, sufficient data access speed means the speed of reading/writing operation comparable with the average user data access speed in existing data storage network systems.

3. System Model

The system is a queueing network (QN) which may be shown as an entire graph [3], the nodes of which (workstations and servers) are the centers for processing and/or generation of remote jobs, while its edges are duplex communications which only have the parameter $v_{i,k}$ - speed of data frame transmission between i^{th} and k^{th} nodes of the system.

The matrix V describes the speed of data transfer between nodes of the system:

$$V = \begin{bmatrix} v_{1,1}, \dots, v_{1,k}, \dots, v_{1,\xi} \\ \dots \\ v_{i,1}, \dots, v_{i,k}, \dots, v_{i,\xi} \\ \dots \\ v_{\xi,1}, \dots, v_{\xi,k}, \dots, v_{\xi,\xi} \end{bmatrix}, \quad (1)$$

where $i = \overline{1, \xi}$, $k = \overline{1, \xi}$, $v_{i,k} = \min(v_i^{\max}, v_k^{\max})$, $v_i^{\max} \in \overline{V}$, $v_k^{\max} \in \overline{V}$ – local maximum connection speeds of i^{th} and k^{th} nodes of the system, correspondingly, \overline{V} – countable set of possible values of local maximum speeds of connection of system’s nodes.

Nodes are independent queueing systems (QS) with confined queue [2, 4]. Let’s introduce the classification of nodes of the system: Type 1 node – metadata and routing control node; type 2 node – data storage and data processing node; type 3 node – subscriber node which includes a subsystem for job stream generation (JSG) as part of a virtual environment for execution of user software (VM) and

the subsystem for data encoding and data mixing (SDEM), as well as a full-fledged type 2 node. Thus, the QN can be subdivided into 2 sub-QN's: a data processing network comprising ξ_2 second type nodes and ξ_3 third type nodes, and metadata processing network comprising ξ_1 first type nodes. The total number of nodes within QN: $\xi = \xi_1 + \xi_2 + \xi_3$.

Although the hardware of type 3 nodes is oftentimes less powerful than the hardware of type 2 nodes, where specialized server equipment is normally used, in reality the following formula is executed: $\xi_1 + \xi_2 \ll \xi_3$, while the cumulative computing resource of type 3 nodes is substantially greater than the relevant indicator for types 1 and 2 nodes. Therefore the application of type 3 nodes as data centers for QN network in distributed computing systems is more than substantiated.

On top of QN on ξ_3 nodes $G_j \in \bar{G}$ of type 3 there is the functional network \bar{G} of JSG, which even though consumes the resources of nodes within QN network it functions in an absolutely independent and isolated manner, and it acts as an external source of jobs in relation to QN network. The input source of jobs in the system is VM of the nodes, each of them generating an ordinary random flow of initial jobs Φ_j :

$$\Phi_j(t) = \{T_j, \vec{\chi}_j, \vec{b}_j\}; \quad (2)$$

where $T_j = t_{j,1}, t_{j,2}, \dots, t_{j,q}, \dots$; $\vec{\chi}_j = \chi_{j,1}, \chi_{j,2}, \dots, \chi_{j,q}, \dots$; $\vec{b}_j = b_{j,1}, b_{j,2}, \dots, b_{j,q}, \dots$; $j = \overline{1, \xi_3}$; $q = \overline{1, \infty}$; $b_{j,q} \in \{0, 1\}$.

In which at random times $t_{j,q}$ random size jobs $\chi_{j,q}$ are generated of one of the two types: 1. job for reading data from the system ($b_{j,q} = 0$); 2. job for recording data into the system ($b_{j,q} = 1$). The jobs are generated in sequence, not more than one per any specific time $t_{i,q}$. The values $\chi_{i,q}$, $b_{i,q}$ and $t_{i,q}$ are mutually independent.

The stream $\Phi_j(t)$ is generated with the intensity $\lambda_j^\chi(t)$ and the statistical expectation M_{χ_j} of the value $\chi_{j,q}$ [1,3]:

$$\lambda_j^\chi(t) = \lambda_j^{(\chi^0)}(t) + \lambda_j^{(\chi^1)}(t); \quad M_{\chi_j} = \lim_{n \rightarrow \infty} \frac{\sum_{q=1}^n \chi_{j,q}}{n}; \quad G_j = \{\lambda_j^\chi(t), M_{\chi_j}, j\} \in \bar{G}; \quad j = \overline{1, \xi_3} \quad (3)$$

where $\lambda_j^{(\chi^0)}(t)$ – intensity of the job stream for reading, $\lambda_j^{(\chi^1)}(t)$ – intensity of the job stream for recording.

Then the stream is transferred to SDEM, where a distributed ledger transaction is opened, the job $\chi_{i,q}$ is converted into a set of standard data blocks (jobs) of the system: 1. Metadata blocks with the size of α_m bits containing the status of the distributed ledger transaction of the system, and are processed by type 1 nodes; 2. Data blocks with the size of α_d bits are processed by type 2 and 3 nodes. Thus, 3 classes of standard jobs are generated in the system: D1 class – job for recoding the data block α_d ; D0 class – job for reading the data block α_d ; M class – job for processing the metadata block α_m .

Let's define the average time for transfer of jobs between the node G_j of the system and the node H_i as arithmetic mean of the matrix V_i (1) for the packets α_m and α_d , correspondingly:

$$t_{vm} = \frac{\sum_{i=1}^{\xi} 1/v_{i,j}}{\xi} \cdot \alpha_m; \quad t_{vd} = \frac{\sum_{i=1}^{\xi} 1/v_{i,j}}{\xi} \cdot \alpha_d; \quad (4)$$

The system provides for simultaneous processing of class D1 jobs generated by SDEM of one $\chi_{j,q}$ in the amount of $p_{j,q} = K \cdot \chi_{j,q}/\alpha_d$, and it strictly prohibits to record two packets from one $\chi_{j,q}$ into one node of processing classes 1 or 2. I.e. the average number of class D1 jobs as generated by SDEM j^{th} source must satisfy the inequality $\bar{p}_j = K \cdot M_{\chi_j}/\alpha_d \leq \frac{\xi_2 + \xi_3}{p_j}$, where p_j – coefficient that characterizes mean-square deviation of the value $\chi_{j,q}$ from its statistical expectation M_{χ_j} , which defines the requirement to assigning the size α_d of the system:

$$\alpha_d \geq \max_j \frac{p_j \cdot M_{\chi_j}}{\xi_2 + \xi_3}; \quad j = \overline{1, \xi_3}; \quad (5)$$

It is admissible to determine the size of the block α_m in an arbitrary way.

The average time spent by SDEM to process one job from the stream Φ_j :

$$\bar{t}_j^{cr} = f(X_j^{cr}, M_{\chi_j}); \quad (6)$$

will be the function of resources X_j^{cr} represented by the node for working with SDEM, intensity $\lambda_i^\chi(t)$ and statistical expectation $M_{\chi i}$, which characterizes the average value of a job in bits.

The total intensities of generation of the jobs of classes D0, D1 and M in the simulated QN are, correspondingly:

$$\lambda_{D0}(t) = \sum_{j=1}^{\xi_3} \lambda_{D0j}(t); \quad \lambda_{D0j}(t) = \lambda_j^{(\chi^0)}(t) \cdot \frac{M_{\chi j}}{\alpha_d}; \quad (7)$$

$$\lambda_{D1}(t) = \sum_{j=1}^{\xi_3} \lambda_{D1j}(t); \quad \lambda_{D1j}(t) = K \cdot \lambda_j^{(\chi^1)}(t) \cdot \frac{M_{\chi j}}{\alpha_d}; \quad (8)$$

$$\lambda_M(t) = \sum_{j=1}^{\xi_3} \lambda_{Mj}(t); \quad \lambda_{Mj}(t) = (\xi_1 + 2) \cdot \lambda_j^\chi(t); \quad (9)$$

The streams Φ_j are independent, and none of them can be compared in terms of capacity with the cumulative stream, therefore in accordance with Khinchin theorem [7, 8, 9] it will be fair to consider the streams D0, D1 and M to be asymptotically Poisson ones, the simplest cores with possible nonstationarity. If the number of nodes $\xi_3 \rightarrow \infty$ in the network \bar{G} the cumulative stream

$$\Phi(t) = \sum_{j=1}^{\xi_3} \Phi_j(t) \quad (10)$$

will work for the simplest one [9].

Let's determine common determinate parameters of the node H_i of QS (job processing centers) which depend on hardware parameters of the node, namely: ml_i – the queue size, sl_i – the storage size, A_i – the bandwidth of the node's QS, $t_{\alpha,i}$ – the job processing time. In accordance with the classification established the processing node receives jobs with a fixed length of $\alpha \in \{\alpha_m, \alpha_d\}$ bits. The vector of hardware and identification parameters of the node which are significant for the modeling:

$$H_i = \{h_{ty}^{(i)}, h_{Ram}^{(i)}(t), X_i^{max}, i\} \in \bar{H}; \quad (11)$$

$$h_{Ram}^{(i)}(t) = \begin{cases} 1, & ml_i - L_i(t) > 0 \\ 0, & ml_i - L_i(t) \leq 0 \end{cases}; \quad h_{ty}^{(i)} \in \{1, 2, 3\}; \quad (12)$$

where $h_{ty}^{(i)}$ – the variable for storage of the new node number; $h_{Ram}^{(i)}$ – the attribute of availability of free RAM space (slots in the node's queue); i – the unique identifier of the node in the system network; $X_i^{max} = \{x_{i1}^{max}, \dots, x_{i7}^{max}, v_i^{max}\}$ – the vector of hardware parameters of the node, in which $x_{i1}^{max}, \dots, x_{i7}^{max}$ is ROM processing speed, maximum available ROM capacity, RAM processing speed, maximum available RAM capacity, number of processors, number of cores per processor, and processing power of the processor's core, correspondingly, $X_i^{max} \in \bar{X}$, $i = \overline{1, \xi}$, $x_{i,l}^{max} > 0$, $v_i^{max} > 0$ $l = \overline{1,7}$; \bar{X} – the confined countable set, members of which are included in the master data by equipment manufacturers; \bar{H} – the node set of the system; $L_i(t)$ – the instantaneous number of unserved jobs sent to the node; ξ – the number of nodes in the system.

Types 1 and 2 nodes function as processing centers. On them, the node's software consumes a certain fixed part, the size of which is determined by the value of the parameter $h_{ty}^{(i)}$:

$$X_i^y(h_{ty}^{(i)}) = \{x_{i1}^y(h_{ty}^{(i)}), \dots, x_{i7}^y(h_{ty}^{(i)}), v_i^y(h_{ty}^{(i)})\}; \quad (13)$$

where $x_{i,l}^y(h_{ty}^{(i)}) \geq 0$; $v_i^y(h_{ty}^{(i)}) \geq 0$; $l = \overline{1,7}$; $i = \overline{1, \xi}$; $X_i^y(1) \neq X_i^y(2) = X_i^y(3)$

Type 3 nodes combine the function of type 1 processing center which consumes the resources as per (13) and the function of JSG as part of VM and SDEM which consume: $X_i^w = \{x_{i1}^w, \dots, x_{i7}^w, 0\}$, и $X_i^{cr} = \{x_{i1}^{cr}, \dots, x_{i7}^{cr}, v_i^{cr}\}$, $i = \overline{1, \xi}$, correspondingly. The vector of resource consumption in JSG:

$$X_{ui}(h_{ty}^{(i)}) = \begin{cases} 0, & h_{ty}^{(i)} \neq 3 \\ X_i^w + X_i^{cr}, & h_{ty}^{(i)} = 3 \end{cases}; \quad x_{i,l}^w \geq 0, x_{i,l}^{cr} \geq 0; \quad v_i^{cr} \geq 0; \quad l = \overline{1,7}; \quad i = \overline{1, \xi}; \quad (14)$$

Let's determine the resources X_i available to the data processing center taking into account (1), (11)-(14):

$$X_i = X_i^{max} - X_i^y(h_{ty}^{(i)}) - X_{ui}(h_{ty}^{(i)}) \quad (15)$$

$$X_i(h_{ty}^{(i)}) = \{x_{i1}(h_{ty}^{(i)}), \dots, x_{i7}(h_{ty}^{(i)}), v_i(h_{ty}^{(i)})\}; x_{i,l}(h_{ty}^{(i)}) \geq 0; v_i(h_{ty}^{(i)}) \geq 0; l = \overline{1,7}; i = \overline{1,\xi}$$

In accordance with (15) let's determine the parameters ml_i, sl_i, A_i and $t_{\alpha,i}$ of i^{th} node:

$$ml_i = \lfloor x_{i,4}(h_{ty}^{(i)}) / \alpha \rfloor; \quad (16)$$

$$sl_i = \lfloor x_{i,2}(h_{ty}^{(i)}) / \alpha \rfloor; \quad (17)$$

$$A_i = \frac{\min\{x_{i,1}(h_{ty}^{(i)}), x_{i,3}(h_{ty}^{(i)}), \prod_{l=5}^7 x_{i,l}(h_{ty}^{(i)}), v_i(h_{ty}^{(i)})\}}{\alpha}; \quad (18)$$

$$t_{\alpha,i} = 1/A_i \quad (19)$$

In accordance with the system's operation logic in QN let's select $\overline{M} \in \overline{H}$ – the subset of ξ_1 nodes H_γ in the independent QN that processes jobs of class M, and let's divide the subset of $\overline{D} \in \overline{H}$ of $\xi_{1,2} = \xi_1 + \xi_2$ the nodes H_δ , which generate the network for processing jobs of classes D0, D1 into the subnetworks (subsets) $\overline{D}_k, k = \overline{1,K}$ in such a way that the subnetwork \overline{D}_1 receives the nodes with the highest bandwidth of the node's QS A_δ , and in \overline{D}_K with the smallest one for this. Let's enter in H_i the additional indices γ, δ : $H_\gamma = \{H_i = \{h_{ty}^{(i)}, h_{ab}^{(i)}, X_i^{max}, i, \gamma\} | h_{ty}^{(i)} = 1\} \in \overline{M}; \gamma = \overline{1,\xi_1}; H_\delta = \{H_i = \{h_{ty}^{(i)}, h_{ab}^{(i)}, X_i^{max}, i, \delta\} | h_{ty}^{(i)} \in \{2, 3\}\} \in \overline{D}; \delta = \overline{1,\xi_{1,2}}$ and apply to \overline{D} the function of sorting to get the ordered set: $f: \overline{D} \rightarrow \{H_\delta \in \overline{D}_{sort} | A_\delta > A_{\delta+1}\}$. As a resulting set of the nodes \overline{D}_{sort} let's determine the subsets \overline{D}_K :

$$\left\{ H_\delta \in \overline{D}_k | \sum_{r=1}^{\delta} sl_\delta \leq \zeta_k; \zeta_{k+1} = \zeta_k + \Delta; \Delta = \frac{\sum_{\delta=1}^{\xi_{1,2}} sl_\delta}{K}; \zeta_1 = \Delta; k = \overline{1,K} \right\}; \quad (20)$$

where $h_{ty}^{(i)}$ – the attribute of the node belonging to a specific type based on the classification introduced. After this, we will be analyzing the networks: $\overline{M}, \overline{D}_k$. The network \overline{M} is an open QN with the intensity of stream from the outer source $\lambda_M(t)$ (9) and one class (M) of jobs. When processing each job from the stream Φ_j (2) $\xi_1 + 2$ jobs are generated with class M, where ξ_1 of them one at a time enters each node $H_\gamma \in \overline{M}$, while the remaining jobs are distributed by the nodes H_γ , depending on the capacity and size of the node's queue. The intensity of the input stream, without taking into account the stream of resent jobs in the nodes H_γ :

$$\lambda_\gamma(t) = (1 + e_\gamma) \cdot \sum_{j=1}^{\xi_3} \lambda_j^\chi(t) \quad (21)$$

Let's define the coefficients e_γ taking into account (9) and (21) using the following equation:

$$(\xi_1 + 2) \cdot \sum_{j=1}^{\xi_3} \lambda_j^\chi(t) = \sum_{\gamma=1}^{\xi_1} \left((1 + e_\gamma) \cdot \sum_{j=1}^{\xi_3} \lambda_j^\chi(t) \right) \Rightarrow \sum_{\gamma=1}^{\xi_1} e_\gamma = 2; \quad (22)$$

Let's define e_γ as:

$$e_\gamma = \begin{cases} 2 \cdot \sqrt{(A_\gamma - \min_\gamma A_\gamma) \cdot (ml_\gamma - \min_\gamma ml_\gamma) / A_{\bar{M}} \cdot ml_{\bar{M}}}, & A_{\bar{M}} > 0 \wedge ml_{\bar{M}} > 0 \\ 2 \cdot (A_\gamma - \min_\gamma A_\gamma) / A_{\bar{M}}, & A_{\bar{M}} > 0 \wedge ml_{\bar{M}} = 0 \\ 2 \cdot (ml_\gamma - \min_\gamma ml_\gamma) / ml_{\bar{M}}, & A_{\bar{M}} = 0 \wedge ml_{\bar{M}} > 0 \\ 2 / \xi_1, & A_{\bar{M}} = 0 \wedge ml_{\bar{M}} = 0 \end{cases}; \quad (23)$$

where $A_{\bar{M}} = \max_\gamma A_\gamma - \min_\gamma A_\gamma$; $ml_{\bar{M}} = \max_\gamma ml_\gamma - \min_\gamma ml_\gamma$; $\gamma = \overline{1, \xi_1}$.

Determination of the coefficients e_γ in the form (23) not only satisfies the equation (22), but it also redistributes the load to the most productive nodes of the network \bar{M} . Each node of the network \bar{M} is a QS of G|G|1| ml_γ type as per Kendall's classification with the queueing discipline FCFS [4]. The node $H_\gamma \in \bar{M}$ at any moment of time t can be in the condition $S_{\gamma,z}$, $z = \overline{0, ml_\gamma + 1}$, where $z = 0$ – the condition where the number of jobs in the queue $L_{qw,\gamma} = 0$, where $0 < z \leq ml_\gamma$ $L_{qw,\gamma} = z$, and $z = ml_\gamma + 1$ means that the node is overloaded and servicing of the job is denied. Probability distribution $P_{\gamma,z}(t)$ of the conditions $S_{\gamma,z}$ is established by the system of Kolmogorov differential equations [2,4]:

$$\begin{aligned} \frac{dP_{\gamma,0}(t)}{dt} &= -\lambda_\gamma(t) \cdot P_{\gamma,0}(t) + \mu_\gamma(t) \cdot P_{\gamma,1}(t); \\ \frac{dP_{\gamma,z}(t)}{dt} &= \lambda_\gamma(t) \cdot P_{\gamma,z-1}(t) - (\lambda_\gamma(t) + \mu_\gamma(t)) \cdot P_{\gamma,z}(t) + \mu_\gamma(t) \cdot P_{\gamma,z+1}(t); \end{aligned} \quad (24)$$

with the starting condition $P_{\gamma,0}(0) = 1$, the normalization requirement $\sum_{z=0}^{ml_\gamma+1} P_{\gamma,z}(t) = 1$ and the intensity of output stream $\mu_\gamma(t) = \begin{cases} A_\gamma, & \lambda_\gamma(t) \geq A_\gamma \\ \lambda_\gamma(t), & \lambda_\gamma(t) < A_\gamma \end{cases}$. Solution of the system (24) is, in general, possible using numerical techniques [5, 6], for example, by way of using the computational procedure proposed in [10, 11]. If we know the probability distribution $P_{\gamma,z}(t)$ taking into account the ordinary, homogeneous and asymptotically Poisson nature of the input stream of request, we can determine distribution of the average number of jobs in the queue of the node H_γ as [2]: $\bar{L}_{qw,\gamma}(t) = \sum_{z=1}^{ml_\gamma} z \cdot P_{\gamma,z}(t)$. Then distribution of the virtual time for processing of the job taking into account (4), (19): $\bar{t}_{wa,\gamma}(t) = \frac{\bar{L}_{qw,\gamma}(t)}{\lambda_\gamma(t)} + t_{\alpha,\gamma} + t_{vm}$. It should be additionally noted that the node H_γ generates a stream of denials with the intensity: $\lambda_{reg,\gamma}(t) = P_{\gamma,ml_\gamma+1}(t) \cdot \lambda_\gamma(t)$. The jobs denied must be submitted for processing to the available nodes of the network \bar{M} based on the queue size ml_γ . Here the stream of jobs resent to the nodes H_γ , taking into account the function of availability (12), will be as follows: $\lambda_{rep,\gamma}(t) = e_{rep,\gamma}(t) \cdot \sum_{\gamma=1}^{\xi_1} \lambda_{reg,\gamma}(t)$. Let's define the coefficients $e_{rep,\gamma}$ based on the equation:

$$\sum_{\gamma=1}^{\xi_1} e_{rep,\gamma}(t) = 1.$$

Let's define $e_{rep,\gamma}(t)$ as:

$$E = mc^2, \quad (1)$$

$$\begin{cases} \sqrt{(A_\gamma - A_{rmi,\gamma}(t)) \cdot (ml_\gamma - ml_{rmi,\gamma}(t)) / A_{\bar{M}}(t) \cdot ml_{\bar{M}}(t)}, & A_{\bar{M}}(t) > 0 \wedge ml_{\bar{M}}(t) > 0 \\ (A_\gamma - A_{rmi,\gamma}(t)) / A_{\bar{M}}(t), & A_{\bar{M}}(t) > 0 \wedge ml_{\bar{M}}(t) = 0 \\ (ml_\gamma - ml_{rmi,\gamma}(t)) / ml_{\bar{M}}(t), & A_{\bar{M}}(t) = 0 \wedge ml_{\bar{M}}(t) > 0 \\ 1 / \xi_{rep,1}, & A_{\bar{M}}(t) = 0 \wedge ml_{\bar{M}}(t) = 0 \end{cases},$$

where

$A_{\overline{M}}(t) = A_{rma,\gamma}(t) - A_{rmi,\gamma}(t)$; $ml_{\overline{M}}(t) = ml_{rma,\gamma}(t) - ml_{rmi,\gamma}(t)$; $\gamma = \overline{1, \xi_1}$;
 $A_{rma,\gamma}(t) = \max_{\gamma}\{A_{\gamma} \in H_{\gamma}: h_{Ram}^{(\gamma)}(t) = 1\}$; $A_{rmi,\gamma}(t) = \min_{\gamma}\{A_{\gamma} \in H_{\gamma}: h_{Ram}^{(\gamma)}(t) = 1\}$;
 $ml_{rma,\gamma}(t) = \max_{\gamma}\{ml_{\gamma} \in H_{\gamma}: h_{Ram}^{(\gamma)}(t) = 1\}$; $ml_{rmi,\gamma}(t) = \min_{\gamma}\{ml_{\gamma} \in H_{\gamma}: h_{Ram}^{(\gamma)}(t) = 1\}$.
 Taking into account the intensity of the stream of denials it would be fair to record $\lambda_{\gamma}(t)$ (21) as:

$$E\lambda_{\gamma}(t) = (1 + e_{\gamma}) \cdot \left(1 + e_{rep,\gamma}(t) \cdot \sum_{s=1}^{\xi_1} P_{s,ml_s+1}(t) \right) \cdot \sum_{j=1}^{\xi_3} \lambda_j^{\chi}(t); \quad (25)$$

Considering the functional model of the network \overline{M} (21) – (25) let's define the concluding virtual distribution of time for execution of the jobs $\Phi_{j,q}$ by the network \overline{M} :

$$T_{M,\Phi}(t) = \max_{\gamma} \bar{t}_{wa,\gamma}(t) = \frac{\sum_{z=1}^{ml_{\gamma}} z \cdot P_{\gamma,z}(t)}{\lambda_{\gamma}(t)} + t_{\alpha,\gamma} + t_{vm} \quad (26)$$

The network \overline{D}_{sort} is an open QN with two classes (D1, D0) of jobs and intensity of the streams from the external source $\lambda_{D0}(t)$ and $\lambda_{D1}(t)$, divided into the subnetworks \overline{D}_k (20). Taking into account the nature of the networks \overline{D}_k arranged by capacity of the nodes H_{δ} let's divide the stream D1 between the nodes of the networks \overline{D}_K with the intensities: $\lambda_{D1,k,\delta}(t) = e_{k,\delta} \cdot \lambda_{D1}(t)/K$, where $e_{k,\delta} = \delta / \sum_{\delta=q_k+1}^{q_k+cord(\overline{D}_k)} \delta$; $\sum_{\delta=q_k+1}^{q_k+cord(\overline{D}_k)} e_{\delta} = 1$; $q_k = \sum_{c=1}^{k-1} cord(\overline{D}_c)$; $c = \overline{1, k-1}$, $q_1 = 0$. The stream D0 is divided by the nodes: H_{δ} of the network \overline{D}_{sort} : $\lambda_{D0,\delta}(t) = e_{\delta} \cdot \lambda_{D0}(t)$. As subjective estimation of a system's processing speed is defined by the speed of reading, and \overline{D}_{sort} is the multitude arranged by capacity of the nodes H_{δ} , let's define e_{δ} as: $e_{\delta} = \delta / \sum_{\delta=1}^{\xi_{1,2}} \delta$; $\sum_{\delta=1}^{\xi_{1,2}} e_{\delta} = 1$. Then the intensity of the input stream of the node H_{δ} without taking into account the resent jobs:

$$\lambda_{\delta}(t) = \lambda_{D0,\delta}(t) + \lambda_{D1,k,\delta}(t); \quad (27)$$

The node of the network $H_{\delta} \in \overline{D}_{sort}$ is a QS of G|G| ml_{δ} type with the intensity of the input steam $\lambda_{\delta}(t)$ and at any specific time t it can be in the condition $S_{\delta,z}$, $z = \overline{0, ml_{\delta} + 1}$, where $z = 0$ is the condition where the number of jobs in the queue $L_{qw,\delta} = 0$, where $0 < z \leq ml_{\delta}$ $L_{qw,\delta} = z$, a $z = ml_{\delta} + 1$ means that the node is overloaded and the servicing of the job is denied. The probability distribution $P_{\delta,z}(t)$ of the conditions $S_{\delta,z}$ was defined by the system of Kolmogorov differential equations [2, 4]:

$$\begin{aligned} \frac{dP_{\delta,0}(t)}{dt} &= -\lambda_{\delta}(t) \cdot P_{\gamma,0}(t) + \mu_{\delta}(t) \cdot P_{\delta,1}(t); \\ \frac{dP_{\delta,z}(t)}{dt} &= \lambda_{\delta}(t) \cdot P_{\delta,z-1}(t) - (\lambda_{\delta}(t) + \mu_{\delta}(t)) \cdot P_{\delta,z}(t) + \mu_{\delta}(t) \cdot P_{\delta,z+1}(t); \end{aligned} \quad (28)$$

with the initial condition $P_{\delta,0}(0) = 1$, the normalization requirement $\sum_{z=0}^{ml_{\delta}+1} P_{\delta,z}(t) = 1$, and intensity of the output stream $\mu_{\delta}(t) = \begin{cases} A_{\delta}, & \lambda_{\delta}(t) \geq A_{\delta} \\ \lambda_{\delta}(t), & \lambda_{\delta}(t) < A_{\delta} \end{cases}$. Let's define distribution of the average number of jobs in the queue of the node H_{δ} as [2]: $\bar{L}_{qw,\delta}(t) = \sum_{z=1}^{ml_{\delta}} z \cdot P_{\delta,z}(t)$. Then distribution of the virtual time for execution of the job taking into account (4), (19): $\bar{t}_{wa,\delta}(t) = \frac{\bar{L}_{qw,\delta}(t)}{\lambda_{\delta}(t)} + t_{\alpha,\delta} + t_{vd}$. The node H_{δ} generates the streams of denials of classes D0 and D1 with the intensities: $\lambda_{reg,D0,\delta}(t) = P_{\delta,ml_{\delta}+1}(t) \cdot \lambda_{D0,\delta}(t)$ and $\lambda_{reg,D0,\delta}(t) = P_{\delta,ml_{\delta}+1}(t) \cdot \lambda_{D1,k,\delta}(t)$. The jobs denied must be submitted for processing to nodes of the network \overline{D}_{sort} which are available based on the size of the queue ml_{δ} . Here the stream of resent jobs sent to the nodes H_{δ} taking into account the function of availability (12) will be as follows: $\lambda_{rep,D0,\delta}(t) = e_{rep,D0,\delta}(t) \cdot \sum_{\delta=1}^{\xi_{1,2}} \lambda_{rep,D0,\delta}(t)$ and $\lambda_{rep,D1,k,\delta}(t) = e_{rep,D1,k,\delta}(t) \cdot \sum_{\delta=1}^{\xi_{1,2}} \lambda_{rep,D1,k,\delta}(t)$ where $e_{rep,D0,\delta}(t) = \delta \left(h_{Ram}^{(\delta)}(t) \right) / \sum_{\delta=1}^{\xi_{1,2}} \delta \left(h_{Ram}^{(\delta)}(t) \right)$, $\sum_{\delta=1}^{\xi_{1,2}} e_{\delta} = 1$, $e_{rep,D1,k,\delta}(t) = \delta \left(h_{Ram}^{(\delta)}(t) \right) / \sum_{\delta=q_k+1}^{q_k+cord(\overline{D}_k)} \delta \left(h_{Ram}^{(\delta)}(t) \right)$,

$$\sum_{\delta=q_k+1}^{q_k+cord(\bar{D}_k)} e_{\delta} \left(h_{Ram}^{(\delta)}(t) \right) = 1, \quad q_k = \sum_{c=1}^{k-1} cord(\bar{D}_c), \quad q_1 = 0, \quad \delta \left(h_{Ram}^{(\delta)}(t) \right) = \delta \cdot h_{Ram}^{(\delta)}(t).$$

Together with the intensity of the stream of denials it will be fair to record $\lambda_{\delta}(t)$ (27) as:

$$\lambda_{\delta}(t) = \lambda_{D0,\delta}(t) + \lambda_{D1,k,\delta}(t) + \lambda_{rep,D0,\delta}(t) + \lambda_{rep,D1,k,\delta}(t); \quad (29)$$

Taking into account the functional model of the network \bar{D} (20), (27) – (29) and concurrent processing of the jobs of classes D0 and D1 as generated by SDEM from the job $\Phi_{j,q}$, let's define the final virtual distribution of time for execution of the jobs $\Phi_{j,q}$ by the network \bar{D} :

$$T_{D,\phi}(t) = \frac{\sum_{\delta=1}^{\xi_{1,2}} \bar{t}_{wa,\delta}(t)}{\xi_{1,2}}; \quad (30)$$

Then the virtual distribution of time for execution of the jobs $\Phi_{j,q}$ as generated by the node G_j of the system on the basis of (7), (28), (41):

$$T_{\chi_j}(t) = \max\{T_{M,\phi}(t), T_{D,\phi}(t)\} + \bar{t}_j^{cr}; \quad (31)$$

4. Some Modeling Results

Based on the model (31) and using Python programming language [12] the software designed for the simulation of operation of a corporate computer network was realized. Using it, operation of the network was simulated using the example previously reviewed in our work [1]. The network comprises $\xi_3 = 500$ nodes of four hardware types (Table 1) which vary in terms of capacity of their disk subsystems and amounts of RAM (which characterizes the maximum queue size) and which are interconnected by Gigabit Ethernet network. The capacity of the file subsystem was defined as nonterminating. Unlike [1] there are no servers in the network ($\xi_2 = 0$). Reading/writing operations are performed in the emulation. The network for data processing comprises 200 nodes ξ_3 , and divided in accordance with the model into $K = 4$ subsystems, 50 nodes each. Actually utilized are 50 nodes per each hardware type. The JSG network comprises $\xi_3 = 500$ nodes.

Table 1

Hardware types of the nodes – members of the network

Hardware Type	Quantity	Capacity of the Disk Subsystem (kB/s)	Amount of RAM (kB)
1	50	61.440	8.388.608
2	50	61.440	4.194.304
3	350	61.440	2.097.152
4	50	409.600	16.777.216

The following values of the model parameters (31) were used in the computations: The size of data packet was within the range of $\alpha_d \in [100 \text{ kB}, 2000 \text{ kB}]$; the size of metadata packet was $\alpha_m = 2 \text{ kB}$; parameters of the job stream Φ_j for two experiments (Table 2). Parameters of the first experiment simulate peak activity in the test net when processing multimedia data (opening, editing, and reporting video-, audio- or any other graphic data) by real users. As part of the second experiment an extreme situation is simulated, where each node of the network at any specific time reads out from distributed storage or transfers to distributed storage super large amounts of data in automatic mode, for instance, it executes a queue of jobs for copying multimedia data or graphic data. The generalized stream of traffic coming from JSG network is determined by the sum (10).

Table 2

Parameters of the job stream from one node of JSG network

Parameter	Experiment 1	Experiment 2
$\lambda_j^{\chi}(t)$ (jobs/sec.)	1	1
M_{χ_j} (kB)	[300 kB; 2000 kB]	[3000 kB; 20000 kB]

In order to estimate the outgoing job stream (the results of execution of Φ_j) the following statistical values were used [13, 14]: Statistical expectation M_{out} (kB/s) and variance S_{out} (kB/s) of the speed of data processing $V_{out} = \chi_j/T_{\chi_j}(t)$ (kB/s) which is defined as its mean-square deviation. Additionally received was the number of L_{out} of the packets α_d unserved as of the end of the experiment (estimated as a number of packets).

Based on the results of operation in Experiment 1 mode (Figure 1:) it is apparent that M_{out} grows purely and linearly as M_{χ_j} grows, and it practically does not depend on the size of the packet α_d . Certain correlation of the value α_d , M_{χ_j} with S_{out} can be simultaneously observed. We did not demonstrate the chart L_{out} as there were no service denials in Experiment 1 mode. This behavior indicates sufficiency and even certain redundancy in terms of capacity of the distributed data storage system built using the model proposed for traffic with the input parameters, equaling the values of Experiment 1 mode (Table 2).

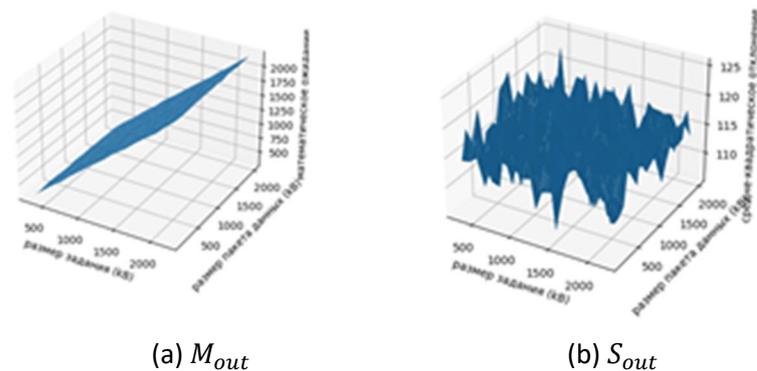


Figure 1: Simulation Results in Experiment 1 Mode

Based on the results of operation in Experiment 2 mode (Figure 2:) it is apparent that M_{out} grows purely and linearly as M_{χ_j} grows to the value $M_{\chi_j} \approx 7000 \text{ kB}$, starting from which, we observe a clearly-defined productivity dip, which testifies to the initial stage of the system overload and accumulation of queues at the nodes of the data processing network, and there can be observed a meaningful dependency on the size of the packet α_d . Starting with the values $M_{\chi_j} > 7000 \text{ kB}$ the variance S_{out} begins to grow substantially. Despite the obvious overload, the denials L_{out} are not present, except for the area $\{\alpha_d \in [100 \text{ kB}, 500 \text{ kB}]; M_{\chi_j} \in [11000 \text{ kB}; 20000 \text{ kB}]\}$. Thus, the distributed data storage system built using the model proposed for traffic with the input parameters equaling the values in Experiment 1 mode (Table 2) demonstrates acceptable productivity, except for the values $\{\alpha_d \in [100 \text{ kB}, 500 \text{ kB}]; M_{\chi_j} \in [11000 \text{ kB}; 20000 \text{ kB}]\}$.

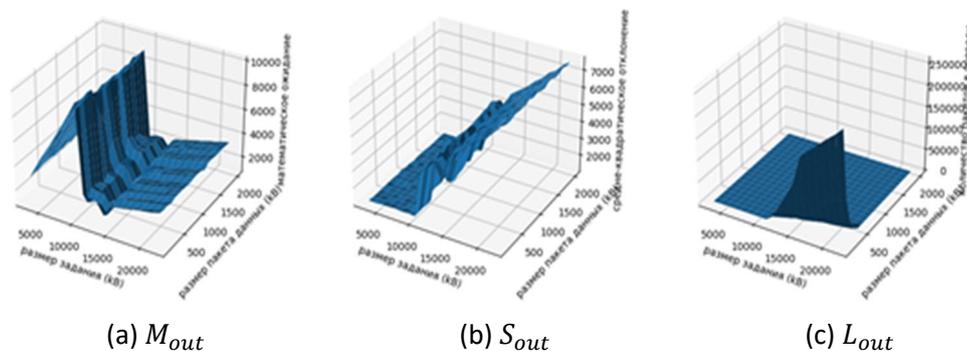


Figure 2: Simulation Results in Experiment 2 Mode

5. Conclusion

Testing of the model proposed in two fairly hard operating modes, at peak loads, was performed. The results of testing suggest that the productivity of the distributed data storage systems built as per the mathematical model proposed, despite the absence of high-end server hardware in the network and rather mediocre hardware parameters of its nodes, is sufficiently high and comparable with the productivity of operation of centralized data storage systems designed and built with high-end and expensive server hardware.

6. References

- [1] A. Nenashev, V. Khryashchev, "The Economics of Introducing the Peer-to-peer System of Storage and Processing of Protected Information at an Enterprise," 2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP), Samara, Russia (2019) 769-772. doi: 10.1109/CSCMP45713.2019.8976720.
- [2] L. Kleinrock, *Theory of Queueing Systems: Translation from English*. Translated by I.I. Glushko; edited by V.I. Neiman, Moscow: Mashinostroenie, p. 432, 1979.
- [3] V. B. Marakhovsky, L. Ya. Rosenblum, A. V. Yakovlev, *Concurrent Processes Modeling, Petri nets*, Saint Petersburg: Professional Literature, p. 400, 2014.
- [4] V. M. Vishnevsky, *Basics of Computer Networks Design*, Moscow: Tekhnosfera, p. 512, 2003.
- [5] I. S. Berezin, N. P. Zhidkov, *Computing Techniques*, volume 1, Moscow: State Publishing House of Physico-mathematical Literature, p. 464, 1962.
- [6] A. A. Samarsky, *Introduction to Calculus of Approximations*, Moscow: Science, p. 271, 1982.
- [7] A. Ya. Khinchin, *On Poisson Streams of Accidental Events*, *Probability Theory and Its Applications*, volume 1, Issue 3, pp. 320–327, 1956.
- [8] A. Ya. Khinchin, V.A. Steklov, *Mathematical Methods of Queueing Theory*, *Mathematical Institute at the Academy of Sciences of the USSR*, volume 49, pp. 1–123, 1955.
- [9] A. B. Markhasin, *Asymptotic model of accumulation of unsteady flows of like events with carry over effect in multiple access wireless networks*, *Herald of Siberian State University of Telecommunications and Information Science* 4 (2011) 19-31.
- [10] M. Yu. Livshits, A. V. Nenashev, Yu. E. Pleshivtseva, *Computing algorithm for optimal control of an object with distributed constants in imperfect areas of end states*, *Herald of South Ural State University, Series: Mathematical Simulation and Programming* 12(4) (2019) 41-51.
- [11] M. Yu. Livshits, A. V. Nenashev, "Efficient computational procedure of alternance optimization method," *Herald of Samara State Technical University, Physico-mathematical Sciences Series*, 23(2) (2019) 361–377.
- [12] Official website of Python programming. URL: language <https://www.python.org>.
- [13] I. K. Tsybriy, *Statistical Analysis of Observed Data: Textbook*, Rostov-on-Don: DGTU Publishing Center, p. 147, 2010.
- [14] W. Ledermann, E. Lloyd. *Handbook of Applicable Mathematics, Statistics*. New York: Wiley, 1984, p. 511