Optimization of Data Infrastructure in Multimodal Transport Center Based on the Finite Element Method

NATALIA MAMEDOVA, MIKHAIL AFANASEV,
SERGEY MAKARENKOV, ARKADIY URINTSOV
Basic Department of Digital Economy
Plekhanov Russian University of Economics
36, Stremyanny Lane, Moscow, 117997
RUSSIA

Abstract: - In the task of optimizing the system architecture of a multimodal transport and logistics center, the key problem is ensuring an efficient data infrastructure within a complex and heterogeneous IT landscape. This study employs the finite element method to formalize the architecture through nodes, connections, and adapted stiffness matrices, and develops resource balance equations and optimization criteria for minimizing data transmission delays. The results present scenarios for real-time data processing, with support for solutions meeting information security requirements. The scientific significance lies in the advancement of optimization methods for distributed systems, while the practical significance lies in the application of the model for deterministic data processing in the MTLC. The results are of interest to developers of industrial IIoT solutions, integrators of logistics systems, and operators of transport hubs interested in improving resource management efficiency. Research prospects are associated with expanding the model to incorporate cost criteria and integrating queue analysis methods, which will enhance forecasting accuracy under dynamic network loads.

Key-Words: - multimodal transport and logistics center, system architecture, data infrastructure, finite element method, real-time data processing, distributed system, solutions based on national standards, industrial IIoT solutions, forecasting dynamic loads, deterministic data processing

Received: April 25, 2025. Revised: June 16, 2025. Accepted: August 18, 2025. Published: December 31, 2025.

1 Introduction

A multimodal transport and logistics center (MTLC) is a multifunctional terminal complex located at a multimodal transport hub, which serves several modes of transport simultaneously, ensuring transshipment and cargo handling between them [1]. Such a center integrates various modes of transport (road, rail, sea, air) and cargo handling technologies at its terminals, allowing for the optimization of logistics processes and reduction of delivery times. Multimodality introduces significant complexity, both from the perspective of the MTLC's placement within the structure of objects in the supporting network of nodal freight multimodal transport and logistics centers in Russia [2], [3], [4], and from the perspective of the digital transformation of the MTLC as a complex managed system [5], [6].

Because the rational design of an MTLC is inextricably linked to its information support for the terminal complex's infrastructure, the design and optimization of the system architecture is therefore a relevant challenge. The system architecture as a research object is, first and foremost, the principles and methods of designing a computer system, including components (nodes, modules, subsystems). The concept also includes the distribution of functions among components (physical or software), the organization of their interaction, and the management of dynamic processes within the system [7]. The subject of this research is the tools for optimized automation of the MTLC's system architecture.

We propose a Finite Element Model (FE Model), based on the Finite Element Method (FEM), as this tool. The application of the FEM extends beyond traditional design methods based on statistical calculations and allows for accounting for peak loads and interaction modes of the model's elements. A key factor is the technological readiness of the MTLC to integrate FEM scenarios, ensured by indicators of information reliability and availability, data exchange speed, transportation costs, etc. [8]. Modern MTLCs require predictive analytics in real time, which is the basis for applying digital twin technology [9].

In this research, the FE Model allows formalizing the system architecture through nodes, connections, and stiffness matrices adapted for warehouse logistics. The research goal is to develop an FEM to ensure the correct setting of load conditions, boundary conditions when modeling the interaction of computational architecture elements, and determining the hardware equipment of computational architecture layers (cloud, edge). Key aspects of the proposed FE Model are correct discretization, accounting for boundary conditions, and model verification on real data.

Thus, the relevance of the research topic is complex and is determined by the search for industry-specific solutions to optimize the MTLC's system architecture.

2 Materials and methods

The Finite Element Method (FEM) is a powerful numerical technique traditionally used in mechanical engineering and physics to analyze complex structures by breaking them down into smaller, simpler parts called "finite elements." In this work, we adapt the FEM to model the data infrastructure of an MTLC. To draw a clear analogy: the physical components of the IT infrastructure (servers, network links, storage) are treated as the "structure," and the data flows and computational loads are treated as the "physical forces" acting upon it.

Key analogies between Mechanical FEM and our solution are described below.

Node (Finite Element) represents a physical or logical component of the system architecture (e.g., a server, a network switch, a storage array).

Connection (Element Boundary) represents a data flow or functional dependency between components (e.g., a network link, an API call).

Stiffness Matrix (K). In mechanics, this matrix defines the relationship between applied forces and resulting displacements, representing the element's resistance to deformation. In our model, the adapted stiffness matrix quantifies the resistance to data flow and computational load transfer. A "stiffer" connection (e.g., a high-bandwidth, low-latency NVLink) facilitates easier load transfer, analogous to a rigid mechanical connection, while a "softer" connection (e.g., a satellite link with high latency) imposes greater resistance.

Force Vector (f) represents the incoming computational load or data traffic assigned to a node.

Displacement Vector (u) represents the resulting state of the system under load, such as the optimal distribution of tasks (e.g., the fraction of an AI inference task processed on an edge device versus the cloud).

To visualize the complex interactions within the proposed FE Model, a schematic diagram (Figure 1) has been developed. This diagram illustrates the hierarchical relationship between the core data processing layers (Cloud, Edge, Perception) and the key FEM nodes, including integration gateways, storage systems, and peripheral devices. The data flows, represented by arrows, show the primary directions of communication, such as video streams from cameras to edge servers for initial processing, control commands from the cloud to Autonomous Guided Vehicles (AGVs), and the role of integration gateways in orchestrating data exchange with external systems (e.g., Customs, ERP). This visual representation aids in understanding the spatial distribution and logical connectivity of the MTLC's data infrastructure.

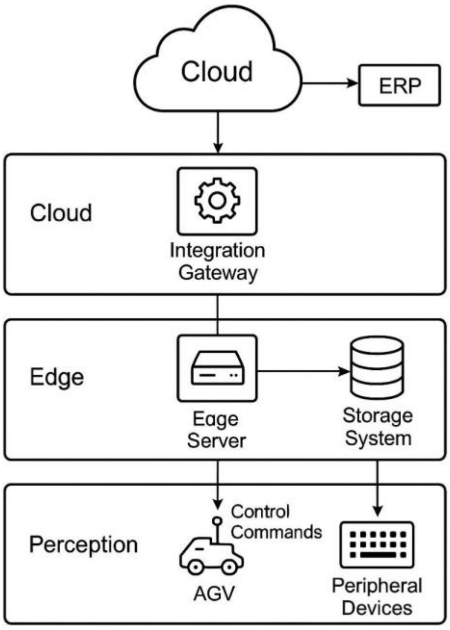

Fig. 1: Schematic diagram of the Finite Element Method for the MTLC data infrastructure
Source: created by the authors

The principles of the FEM apply to system design, and the application logic is universal. Despite several limitations (in particular, the inapplicability of the FEM to nonlinear systems), the model enables the analysis of how changes in one node propagate through the system and provides a quantitative assessment of the impact of changes, thereby allowing for the refinement of system architecture parameters.

The design of the MTLC's distributed data infrastructure is guided by the CAP theorem [10]. For the MTLC, data consistency (C) is non-negotiable due to the critical nature of logistics operations (e.g., cargo tracking, inventory management), where stale or conflicting data can lead to significant operational and financial losses. While availability (A) is also crucial, the primary assumption is that a temporary, controlled unavailability during a network partition (P) is a lesser evil than operating with inconsistent data. Therefore, the system is designed as a CP system (Consistency & Partition Tolerance). In practice, this means that in the event of a network split, the system will prioritize ensuring data consistency across available nodes, potentially at the cost of making some services temporarily unavailable until the partition is resolved. This design choice directly influences the implementation of the integration layer and data replication strategies, mandating the use of consensus protocols and mechanisms to gracefully handle partition scenarios.

Besides ensuring data consistency, a key task is managing the overall complexity of the system architecture. Designing a system architecture as a complex model requires managing its complexity, achieved through abstraction and simplification. Abstraction is achieved by representing the system architecture as a structural model - a formalized hierarchy of components, connections between them, and rules of their interaction, intended for analysis, design, and optimization. The complexity of the system architecture is managed through a process of discretization. This process results in a finite set of allowable states for both the computational and engineering architectures. This simplifies system analysis by operating not with specific values describing architectural components but with ordered groups of values represented discretely in the model. The set of allowable states of the system architecture is ensured by the results of partitioning the architectures into finite elements (nodes and connections) and defining architecture detail levels.

At the same time, the nodes of the FEM - the components of the system architecture-remain unchanged, as their list, structure, and functions in computer networks are a priori known. Correct modeling of component behavior is ensured by the principles embedded in the FEM [11]. This research is based on the principles of accounting for boundary conditions (to fix scenario conditions relative to the external environment) and assembling the global stiffness matrix (by summing contributions from individual elements into the corresponding degrees of freedom).

These principles allow us to transition from a theoretical model to a practical tool for capacity planning. The process of "assembling the global stiffness matrix" (K_{global}) from individual element matrices (K^{e}) is analogous to assembling a complete structural model of a bridge from its individual beams and connectors. Each beam has its own stiffness, and when assembled, they create a global system that describes how loads (e.g., traffic) will be distributed throughout the entire structure. Similarly, our global matrix K_{global} describes how computational loads are distributed across the entire IT infrastructure based on the "stiffness" (performance) of its individual components and connections.

Based on the FEM, it is proposed to make decisions regarding planning the capacity of the computational architecture and the hardware equipment of computational architecture layers (cloud, edge). For the selected object, the MTLC, the model will ensure the choice of computational power load mode (CPU, GPU, memory, network), task distribution between cloud and edge layers, and will form the basis for determining hardware requirements. In terms of business logic, the effectiveness of the FEM is proposed to be evaluated based on several metrics influenced by the system architecture: speed of processing a cargo unit during transshipment from scanning to loading (Edge servers), vehicle utilization coefficient (IIoT gateways), network availability (hybrid architecture), and data transmission delay (GNSS track update time).

Achieving target values for these metrics is a direct function of the MTLC's IT landscape complexity. This landscape is characterized by intricate interactions between a wide array of integrated systems, applications, services, and components. This generates a set of requirements for designing the integration layer of the system architecture and places ensuring the data infrastructure is at the forefront [12]. Consequently, the FEM must include, among the allowable states, those that implement the functions of the integration layer, such as orchestrating microservice calls, transforming and routing messages between integrated systems, managing events, and asynchronous interaction. Furthermore, the data infrastructure at the level of designing the target system architecture must adhere to several key principles.

Considering the specifics of the IT landscape for the MTLC, the main principles are minimizing data movement between FEM nodes (the "data locality" principle) and minimizing data duplication and redundant computations. Thus, the list of integration layer functions and data infrastructure components is defined by the authors as a guideline for building the FEM. To implement the described integration layer functions and ensure the data infrastructure within the FEM, a list of key model nodes is presented below.

The tools for implementing the integration layer functions are ETL/ELT tools for transferring data between sources and storage, API gateways for integration with external systems and data provision, and message queues for managing data streaming. The specified toolset will be used to represent the FEM in terms of defining the list of nodes and describing the connections between them. The list of FE Model nodes is as follows: 1) Computational nodes (servers, GPU clusters, FPGAs); 2) Network nodes (routers, switches, SDN controllers); 3) Storage nodes (SSD arrays, HDD arrays, cache memory, object storage (S3)); 4) Peripheral devices (cameras, IIoT devices, edge servers). This list is standard for optimizing computational architectures but has been expanded and refined to account for the specifics of an MTLC.

The MTLC not only serves multiple modes of transport (e.g., rail, road, sea/river, and air transport) but also ensures cargo handling at its constituent terminals. To support these operations, a complex IT infrastructure is required, including GPS/GNSS trackers, IIoT sensors, routing systems, and integration with external systems (customs, port services, ERP systems).

The primary challenge in designing such an architecture, however, is enabling real-time data processing (analysis of streaming data from cameras, weight sensors, and RFID tags). This requires a hybrid architecture combining cloud resources, primary and backup high-speed data transmission means. In this regard, it is proposed to expand the list of FEM nodes for the MTLC and include: 1) Identification systems (RFID tags); 2) Analytical systems (video analytics); 3) Integration gateways (API gateways for 1C, customs systems); 4) Primary and backup communication channels (satellite communication, 5G/LTE); 5) Geo-distributed data centers (local servers in MTLC warehouses); 6) Cloud layer (clouds).

Having formed the list of FEM nodes, we present the model's connections (interconnects), which define the interaction between nodes. Based on the fact that each node represents a logical or physical component of the system architecture, a general approach is to divide connections into physical (e.g., fiber optic, Ethernet, PCIe) and logical (including virtual networks, overlay topologies). Ranking the FEM nodes is advisable based on their role in supporting the computational process, from the computation core to the periphery and integration. This approach reflects the architectural hierarchy of computational systems, where the foundation is computational resources, followed by data transmission systems, storage, periphery, integration, and supporting components. The interaction matrix of all FEM components is the starting point for defining FEM implementation scenarios.

3 Results and Discussion

Applying the finite element method and the node selection rule based on the relevance criterion, we conducted modeling for key operational scenarios of the MTLC. Considering the specifics of the MTLC's IT landscape, we consider the fixed bandwidth, constant latency, connection reliability (packet loss), and redundancy as the target metrics of the system architecture. We now focus on real-time data processing scenarios. The MTLC's data infrastructure shares common features with industrial systems [9], [12], [13], which require deterministic data transmission. In practice, this is implemented based on standards and specifications for data transmission and device interaction in industrial OPC UA and Time-Sensitive Networking (TSN).

However, the logistics domain also presents unique characteristics. In particular, seasonal loads impose scalability requirements, and higher equipment mobility is combined with mandatory integration of GPS/GNSS trackers. The FEM scenarios, besides Ethernet networks (according to the TSN standard) guaranteeing timestamped data delivery, include PROFINET IRT networks for critical control loops (e.g., sorting lines for cargo of different modalities) and 5G URLLC, if the scenario includes mobile and wireless edge devices (loaders, drones) among the peripheral devices. It is also important to consider the unique aspects of modern MTLCs implementing IIoT solutions based on protocols such as Zigbee 3.0 (for WPAN networks used for personalized tracking and monitoring of room parameters) and LoRaWAN for LPWAN (monitoring remote warehouse areas, GPS trackers) and 5G mMTC for connecting a large number of IIoT devices under conditions of their high geographical distribution. Thus, the described FEM implementation scenarios should be aimed at ensuring deterministic, time-sensitive data transmission.

The real-time data processing scenarios for the FEM are as follows:

- Analysis of AI inference task distribution between edge devices and the cloud (general scenario);
- Analysis of cloud and edge layer load for video analytics.

There are integration points between the scenarios, the most significant for the data infrastructure in the MTLC's IT landscape being: the use of TSN for temporal synchronization of AGVs and video analytics (a unified timestamp for metrics, traffic prioritization using the DSCP mechanism); OPC UA for end-to-end data transmission from sensors to the cloud. Furthermore, work on the FEM scenarios considers only solutions based on national standards, the use of domestic cryptographic standards and hardware (ElIoT, SKIF, Elbrus, e.g., sensors and processors) certified by the Russian Ministry of Industry and Trade. This is mandated by information security requirements for enterprises operating in the field of warehouse logistics.

The description of the scenarios presented later in this section specifies their implementation conditions. The following logic was used when indicating the main and additional conditions. Both conditions relate to components of the MTLC's IT infrastructure and are related as follows: the main condition can be optimally implemented only if the additional condition is met, the fulfillment of which is determined by the content of the main condition. Data from the Prometheus monitoring system was used to collect initial data and validate modeling results.

To enhance the utility of the proposed scenarios for the field of warehouse logistics, distribution, and supply, numerical solutions were checked against a set of measurable metrics for compliance with business logic operations, such as:

- management of autonomous loading equipment (Automated Guided Vehicle (AGV)/ Autonomous Mobile Robots (AMR)),
- monitoring cargo condition (vibration, temperature, seal integrity),
- predictive maintenance system for vehicles (scheduled preventive maintenance system),
- coordination of multimodal transportation (real-time tracking of cargo transshipment).

The business logic operations described above are specific to warehouse logistics and are performed based on physical components of the system architecture. This focus enhances the relevance of the proposed scenarios for MTLCs, albeit at the cost of broader universality for other industrial systems.

Specific modeling results for each of the listed scenarios, including the obtained metric values and optimized configurations, are presented below.

3.1 Scenario 1. Optimization of AI Inference Task Distribution between Edge Devices and the Cloud

In the first scenario, for the main condition "Optimization of AI inference task distribution between edge devices and the cloud," the additional condition is "Optimization of edge computing (CPU/NPU + network)." Edge devices (cameras, sensors, smartphones, industrial controllers) perform AI inference locally, i.e., directly at the data collection site. Model training, global analytics, long-term storage, and information processing occur in the cloud. The mathematical model for implementing the scenario using the finite element method is presented below.

The mathematical model comprises the following elements and connections: 1) the connection "high-speed inter-server data transfer" between the nodes "computational nodes" and "cloud layer"; 2) the connection "hybrid edge-to-cloud connection" between the nodes "cloud layer" and "peripheral devices (Edge servers)". The system is represented as a continuous computational environment, consisting of a cloud domain. Ω_c (GPU cluster) and a peripheral domain Ω_e (edge devices), connected by network interfaces $\Gamma = \Gamma\_fast \cup \Gamma\_hybrid$, where Γ_fast: NVLink (900 GB/s), InfiniBand (400 GB/s), Ethernet (100 Gbit/s), Γ_hybrid: 5G/LTE/Wi-Fi 6/Fiber optic/Satellite.

The main idea of the model is to discretize the system into finite elements, where each element represents a computational node with corresponding network connections. Mathematically, this is expressed as: $\Omega_c = \bigcup_{e=1}^{N} \Omega_e$, (1)

In this equation Ω_e^i describes an individual computational element $\{node\_i,\ connection\_ij\}$ with its characteristics. Each element is characterized by computational power C_i (TFLOPS), connection delay τ_{ij}, and throughput B_{ij}.

The goal of the model is to minimize the functional $J(u)$ under the condition of optimal load distribution $u(x,t)$, where x is the spatial coordinate, t is the data transmission time between Ω_c and Ω_e. To minimize the functional $J(u)$ with respect to the vector u, it is necessary to solve the system of equations obtained from the stationarity condition $\delta J/\delta u = 0$.

The core of our model is the computational load balance equation for each element: $K^e u^e = f^e$, (2)

In this equation:

f^e (the "force" vector) is the incoming computational load demanding resources.

u^e (the "displacement" vector) is the solution we seek - the optimal distribution of this load across the system.

K^e (the "stiffness" matrix) is the crucial component that defines how the load is distributed. In mechanical terms, a stiff spring transmits force efficiently with little deformation. In our network terms, a "stiff" connection (low delay, high throughput) allows for efficient load transfer. The coefficients in the K^e matrices are therefore formulated as functions of network metrics like inverse delay ($1/\tau$) and throughput (B). A high value of $1/\tau$ (e.g., for NVLink) means high stiffness, promoting load sharing across that channel, much like a rigid beam efficiently transfers stress in a structure. For different types of connections, the stiffness matrix has a different form: high-speed connections K_{det} and hybrid connections K_{hybrid}.

For high-speed connections (NVLink/InfiniBand), a deterministic component (3) is used.

$$K_{det}^{e} = \begin{pmatrix} \frac{1}{\tau_{nv}^{ii}} & -\frac{1}{\tau_{nv}^{ij}} \\ -\frac{1}{\tau_{nv}^{ji}} & \frac{1}{\tau_{nv}^{jj}} + \frac{1}{\tau_{ib}^{jj}} \end{pmatrix}, \tag{3}$$

The positive diagonal terms represent a node's inherent "stiffness" or capacity to handle load based on its local high-speed connections. The negative off-diagonal terms represent the coupling between nodes; the higher the quality of the connection, the stronger the coupling, enabling more effective offloading of tasks from one node to another. Superscript denotes the connection from node to node, ensuring consistent indexing across all matrix equations. This is directly analogous to how a stiff mechanical connection allows one part of a structure to support another.

The matrix describes how data transmission delays τ_{nv} = 0.5 ms (NVLink delay) and τ_{ib} = 2 ms (InfiniBand delay) affect load balancing between nodes. The smaller the delay, the larger the corresponding coefficient, which increases the connection "stiffness" and promotes transferring more load through this channel. The form of the matrix (3) is derived from the condition of minimizing the system's potential energy, taken as the total delay of computational load propagation in the network. For a pair of nodes connected simultaneously by NVLink and InfiniBand, the resulting stiffness matrix is calculated as the sum of the matrices corresponding to each connection type. For TCP/IP connections, the delay model is:

$$\tau_{tcp} = \tau_{mean} + \xi, \tag{4}$$

where $\xi \sim N(0, \sigma^2)$, σ =1ms. This is accounted for by introducing a stochastic component with a normal jitter distribution (σ = 1 ms) into the system.

Hybrid edge-to-cloud connections are modeled using an adaptive matrix:

$$K_{hybrid}^{e} = \alpha K_{5G} + (1-\alpha) K_{sat}, \alpha \in [0,1], \tag{5}$$

where the coefficient α is controlled by a Markov process, ensuring optimal switching between different connection types (5G/LTE/Wi-Fi 6/fiber optic/satellite). The process is defined by three states. For example, State 1 (excellent 5G coverage), State 2 (poor 5G coverage), and State 3 (satellite backup), and their dynamics are governed by the transition probability matrix. The probabilities were estimated from historical network data to reflect realistic state transitions, with values chosen to promote system stability by favoring the best available connection.

The form of the adaptive matrix K_{hybrid}^{e} for hybrid connections is justified by the need for dynamic load redistribution between heterogeneous communication channels (5G/LTE/Wi-Fi 6, fiber optic, satellite, etc.), whose characteristics (delay, throughput) can change significantly and unpredictably over time. K_{5G} and K_{sat} are conductivity matrices, similar in structure to matrix (3), but parameterized by the corresponding delays and throughput of the 5G and satellite channels.

$$\text{For example, } K_{5G} = \begin{pmatrix} \frac{1}{\tau_{5g}^{ii}} & -\frac{1}{\tau_{5g}^{ij}} \\ -\frac{1}{\tau_{5g}^{ji}} & \frac{1}{\tau_{5g}^{jj}} \end{pmatrix}, \tag{6}$$

With the current estimate of the delay in the 5G network. The adaptation coefficient α (where $\alpha \in [0,1]$) controls the contribution of each channel to the total conductivity. A value of α =1 corresponds to full use of the 5G channel, α =0 - the satellite channel. Intermediate values implement weighted load distribution.

The coefficient α is controlled using a Markov process, which allows modeling random and time-dependent transitions between network states (e.g., "excellent 5G coverage", "poor 5G coverage", "satellite backup"). Transition probabilities between states and the value of α in each state can be estimated based on historical link quality data or set by QoS policy.

Thus, the adaptive matrix (5) provides a mathematical basis for optimizing load distribution under unstable network infrastructure conditions.

The global system of equations for the AI inference task distribution in the hybrid edge-cloud system is formulated as:

$$(K_{det} + K_{stoch}) u = f, \tag{7}$$

The model incorporates boundary conditions, which are crucial for obtaining a unique and physically meaningful solution. Formula (8) denotes a fixed percentage of processing on peripheral devices.

$$u_e = g_e(t), \tag{8}$$

This is an analogue of the "fixed end" (fixed support) in a mechanical beam. The node cannot move freely (i.e., its u_e state is rigidly fixed by an external requirement). For example, a law or security policy may require that certain data be processed only locally and not transferred anywhere.

$\frac{\partial u_c}{\partial t}=0$ - stable load on the cloud. This is an analogue of a "pinned support" or a symmetry condition. It does not capture the state of the cloud completely, but restricts its behaviour, indicating that its load is constant over time in a steady state. This simplifies the model by assuming that cloud resources are scaled so as to absorb the load without changing their internal state from the point of view of the external system.

The optimization criterion is presented in formula (9). It is expressed by the functionality that we strive to minimize:

$$J(u)=\frac{1}{2}u^T Ku+\lambda_1 \left|\left|u-u_{target}\right|\right|^2+\lambda_2 R(u), \quad (9)$$

The physical interpretation of this function is directly related to mechanics. The term $\frac{1}{2}u^T Ku$ from (9) represents the "strain energy" of the system. In mechanics, it is the energy stored in a deformable body. In our model, this is the generalized "cost" of data transmission and distribution of computing load. Minimizing this term forces the system to find a state u in which data is transmitted through the most "rigid" (high-performance) channels, minimizing overall delays and losses, just as a physical structure minimizes strain energy.

The term $\lambda_1\left|\left|u-u_{target}\right|\right|^2$ from (9) ensures that the solution is close to the target distribution of u_{target} (for example, preference for local processing on edge devices).

This is an analogue of the application of "elastic force" in mechanics, which seeks to return the system to the desired position. The term $\lambda_2 R(u)$ from (9) is a penalty for undesirable effects such as high-power consumption.

Thus, minimizing the $J(u)$ functional is not just an abstract mathematical procedure, but a search for a system configuration that minimizes the overall "energy cost" of data transmission while observing the specified constraints.

The model validation employed the Prometheus monitoring system over a 14-day operational period, capturing both normal and peak load conditions. Specific parameters measured included: computational metrics (CPU/GPU utilization, memory allocation per node), network metrics (end-to-end latency, jitter, packet loss at each connection point), and application metrics (AI inference completion time, video frame processing rate).

The numerical solution of the system of equations was implemented based on a custom Python solution using NumPy and included the following steps: discretization by splitting into nodes and connections; local optimization is solved by formula (2) for each element; global assembly by constructing. K_{global}. The numerical solution also included network topology adaptation by recalculating α for hybrid network connections. The element size was taken as 1 computational node (1 computational module and related interfaces), the time step was 10ms for dynamic balancing, the convergence criterion was a relative change in the functional <0.1%, and the stopping criterion: $\frac{\Delta J}{J}$ <0.1%.

The global stiffness matrix K_{global} The hybrid network was obtained using the standard finite element assembly procedure, which combines deterministic components (high-speed connections) and adaptive stochastic components (hybrid connections) governed by a Markov process. This approach to hybrid system modeling is well-established and widely used in the literature [14]. The system stability analysis under dynamic load variations was performed using a Lyapunov function, ensuring convergence to an equilibrium state.

Model verification included checking the convergence condition (10) checking the compliance of calculated and measured values (error <5%), as well as fulfilling SLA for delay (<100 ms), jitter (<15 ms), and availability (>99.95%).

$$\left|\left|Ku-f\right|\right|<\varepsilon, \quad (10)$$

Solving the optimization problem under the given boundary conditions yielded an optimal system configuration. This configuration includes specific node parameters and defines the operational boundaries. The mathematical apparatus for verifying the compliance of the FEM connection with the declared target metrics of the MTLC system architecture is also specified.

3.1.1 Interpretation of the results of scenario 1

The optimization of AI inference task distribution between edge devices and cloud infrastructure is implemented through a structured finite element model (FEM) consisting of two critical interconnection types with distinct characteristics and performance parameters.

The high-speed inter-server data transfer connection between computational nodes and the cloud layer (GPU cluster) employs multiple high-performance technologies, including NVLink 3.0 (900 GB/s), InfiniBand HDR (400 GB/s), and TCP/IP (100 Gbit/s). This connection exhibits varying latency characteristics: 0.5 ms for NVLink, 2 ms for InfiniBand, and 10 ms for Ethernet. The model combines deterministic behavior for NVLink and InfiniBand connections with stochastic elements for

TCP/IP, the latter featuring normal jitter distribution with σ=1 ms.

The hybrid edge-to-cloud connection linking the cloud layer with peripheral devices (Edge servers) integrates diverse communication technologies, including 5G/LTE/Wi-Fi 6 and fiber optics, operating within the 3.4-3.8 GHz frequency range (according to RTRS regulations). The system implements the GOST R 53113.1-2008 standard for channel protection and employs VLAN segmentation by device types. Critical processes are handled through local edge computing capabilities, with the "Gonets" satellite system providing channel redundancy. Performance parameters include bandwidth of 50-1000 Mbps, latency ranging from 10-100 ms, jitter below 15 ms, and 99.95% availability. This adaptive stochastic model utilizes Markov processes for state transitions, enabling dynamic optimization of connection parameters based on network conditions and operational requirements.

The integrated system represents a sophisticated approach to AI workload distribution, combining high-performance cloud infrastructure with adaptable edge connectivity to ensure optimal processing efficiency while maintaining stringent performance and reliability standards (Figure 2).

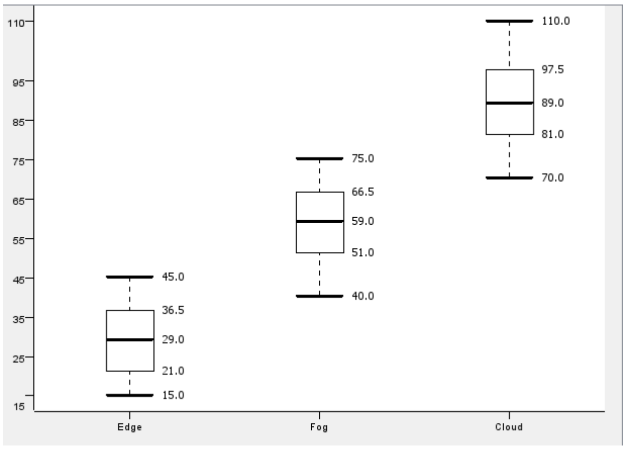

Fig. 2: Latency distribution across network layers under dynamic load conditions

Source: created by the authors

Implementing this scenario, considering the presented parameters and boundary values, enables the MTLC's system architecture to adapt to changing network conditions and ensure compliance with QoS requirements. When working with test metrics, special attention was paid to processing hybrid connections and dynamic load redistribution when network conditions change, which made it possible to refine the boundary conditions.

3.2 Scenario 2. Optimization of Cloud and Edge Layer Load for Video Analytics

For this scenario, the proposed finite element model formalizes video analytics as a task of optimal data flow distribution in a multi-level environment. The following are sequentially described: the system of elements, the balance equation, stiffness matrices for various types of connections, the optimization functional, and the numerical solution algorithm. The main condition of the model "Optimization of cloud and edge layer load for video analytics" is supplemented by the condition "Data processing on Edge servers". The mathematical model for implementing the FEM scenarios was developed for the following system of elements: 1) between the nodes "geo-distributed data center" and "peripheral devices (local server, camera)" there is a connection "high-availability video traffic"; 2) between the nodes "geo-distributed data center" and "peripheral devices (camera, Edge server)" there is a connection "control video traffic"; 3) between the nodes "cloud layer (GPU cluster)" and "peripheral devices (camera, Edge server)" there is a connection "video stream + control commands"; 4) between the nodes "cloud layer (GPU cluster)" and "peripheral devices (camera, Edge server)" there is a connection "video analytics pipeline".

Let us consider the video analytics system as a continuous environment, consisting of: peripheral devices. Ω_e (cameras, Edge servers, local servers), geo-distributed data centers Ω_n, cloud GPU cluster Ω_p, network interfaces $\Gamma = \Gamma_1$ (control traffic) $\cup\ \Gamma_2$ (video streams). Finite element discretization made it possible to represent the system, where each element includes:

- a node (camera/server) with parameters: resolution 4K/1080p; codecs H.265/H.264; bitrate 2-50 Mbps;
- connections with characteristics: transmission delay <100 ms (E2E); jitter <15 ms; frame loss <0.1%.

The video stream balance equation is a direct application of the law of conservation, analogous to Kirchhoff's current law in electrical circuits or mass balance in fluid dynamics. Imagine node I as an interchange in a pipeline system. F_i, (Mbps) is the total volume of water (data) entering the junction.

The equation: $\sum_j B_{ij} u_{ij} + \sum_k B_{ik} u_{ik} = F_i$, (11)

ensures that the entire incoming data stream is distributed losslessly among all available outgoing channels. The term $\sum_j B_{ij} u_{ij}$ from (11) is the total flow sent from the junction through one set of pipes (for example, through j-type channels leading to edge servers).

The term $\sum_k B_{ik}u_{ik}$ from (11) is the total flow directed through another set of pipes (for example, through channels of type k leading to the cloud).

The "stiffness" parameters in the matrix K determine which fraction of the flow (u_{ij} and u_{ik}) will be directed through each of the channels, depending on their "throughput" (B) and "quality" (τ, J).

Matrix representation for the entire system:

$$[K] \cdot u = F, \tag{12}$$

where $[K]$ is the global stiffness matrix of the system, composed of elementary matrices. Here, the stiffness matrix K characterizes the system's resistance to the smooth propagation of video streams. The "load" u is the video traffic itself. We define the "stiffness" of a video channel not just by its delay (τ), but by the stability of that delay, factoring in jitter (J). The term $\tau^2 + J^2$ acts as a composite metric for signal distortion and increases the "penalty" for instability (J) compared to constant delay (τ). Thus, the matrix coefficient (13) implies that a video channel's effective "stiffness" (its ability to carry a load) is directly proportional to its bitrate (B) and inversely proportional to the square of its transmission imperfections.

$$B/(\tau^2 + J^2), \tag{13}$$

A clean, stable, high-bandwidth link is a "stiff" pipe for video data, minimizing distortion under load, similar to how a stiff structural member minimizes deflection under force.

Elementary stiffness matrices included a matrix for the "camera-edge" and a matrix for the "edge-cloud".

Matrix for the "camera-edge" connection:

$$K^e = \begin{pmatrix} \frac{B_i}{\tau_i} & -\frac{B_i}{\tau_i} \\ -\frac{B_j}{\tau_i} & \frac{B_j}{\tau_i} + \frac{B_j}{\tau_j} \end{pmatrix}, \tag{14}$$

where τ_i is the transmission delay (<100 ms).

The matrix in (14) for the "camera-edge" connection is built on the principle of the generalized Ohm's law: the channel conductivity (stiffness) is directly proportional to the bitrate (B_i, B_j) and inversely proportional to the delay (τ_i, τ_j), which reflects the channel's ability to transmit data and the resistance it offers.

Matrix for the "edge-cloud" connection:

$$K^e = \begin{pmatrix} \frac{B_i}{\tau_{ij}^2 + J_{ij}^2} & -\frac{B_i}{\tau_{ij}^2 + J_{ij}^2} \\ -\frac{B_j}{\tau_{ij}^2 + J_{ij}^2} & \frac{B_j}{\tau_{ij}^2 + J_{ij}^2} + \frac{B_j}{\tau_j} \end{pmatrix}, \tag{15}$$

where J_i is the jitter (<5 ms), B_i is the bitrate from node i (edge) to node j (cloud); B_j is the bitrate processed at node j (cloud); τ_{ij}, J_{ij} are the delay and jitter of the channel from i to j; τ_j is the internal processing delay in the cloud node j.

This form of the matrix (15) for the hybrid "edge-cloud" connection combines two principles. The off-diagonal elements, responsible for load distribution between edge and cloud, depend on the quality of the "edge-cloud" channel (τ_{ij}, J_{ij}). The diagonal element $\frac{B_j}{\tau_j}$ adds the accounting of the internal delay (τ_j) in the cloud node when processing the incoming stream. The element $K^e[2,2]$ describes the total "resistance" of the cloud node, which is the sum of the resistance of the incoming channel ($B_j/(\tau_{ij}^2 + J_{ij}^2)$) and the resistance of its own processing (B_j/τ_j).

Stiffness matrix for control traffic - the diagonal structure reflects the deterministic nature of the traffic:

$$K_{ctrl}^e = \begin{pmatrix} \frac{1}{\tau_{ctrl}} & 0 \\ 0 & \frac{1}{\tau_{ctrl}} \end{pmatrix}, \tag{16}$$

where τ_{ctrl} <50 ms is the control traffic delay.

The diagonal form of the matrix (16) is because control traffic is deterministic and not intended for distribution between nodes. Zero off-diagonal elements reflect the absence of mutual influence of nodes on the control channel. Stiffness depends only on the delay in each specific section.

Stiffness matrix for video streams (RTSP/H.265) – stiffness depends on the ratio of bitrate to the quadratic norm of delay and jitter:

$$K_{video}^e = \begin{pmatrix} \frac{B_i}{\tau_{ij}^2 + J_{ij}^2} & -\frac{B_i}{\tau_{ij}^2 + J_{ij}^2} \\ -\frac{B_j}{\tau_{ij}^2 + J_{ij}^2} & \frac{B_j}{\tau_{ij}^2 + J_{ij}^2} \end{pmatrix}, \tag{17}$$

where: B_i is the video stream bitrate (2-50 Mbps); τ_{ij} is transmission delays (<100 ms); J_{ij} is jitter values (<5 ms).

The form of the matrix (17) for video streams is justified by the fact that the generalized channel resistance for multimedia traffic is determined not simply by the delay τ, but by its variations (jitter J). The quadratic norm $\tau^2 + J^2$ is an integral quality metric characterizing the distortion of the temporal characteristics of the flow. Stiffness (conductivity) is directly proportional to the bitrate B (volume of transmitted data) and inversely proportional to this quality metric, which reflects the channel's ability to transmit video information with minimal distortions.

Quality of Service (QoS) equations included: delay constraint (18) and loss constraint (19).

$$\tau_{\mathrm{i}} = \tau^0 + \sum(u_{\mathrm{i}}\square/B_{\mathrm{i}}\square) \leq \tau_{max}(100\text{ms}), \tag{18}$$

$$P_{loss} = \exp(-\alpha * \mathrm{B}_{\mathrm{i}}\square/\mathrm{u}_{\mathrm{i}}\square) \leq 0.01\%, \tag{19}$$

where α is a coefficient determining the system's response rate to changes in network conditions.

The optimization functional of the model is represented by minimization:

$$minJ(u) = \tfrac{1}{2}u^{\mathrm{T}}Ku - F^{\mathrm{T}}u + \lambda\left|\left|u - u_{QoS}\right|\right|^2 + \mu R(B), \quad (20)$$

where: $R(B) = \sum\left(B_i - B_{target}\right)^2$ - describes bitrate stabilization, and u_{QoS} = (0.99, 0.01) - target distribution (99% availability), under the constraints: $\sum_j u_{ij} = 1$, $0 \le \mathrm{u}_{\mathrm{i}}$⍰ ≤ 1; $B_{min} \le B_{\mathrm{i}}$⍰ $\le B_{max}$; $\tau_{\mathrm{i}} \le \tau_{max}$; $P_{loss} \le P_{max}$.

The numerical solution of the system of equations was implemented using Python with NumPy. Data for model validation, boundary conditions, and real-time SLA compliance monitoring were collected and analyzed in the Prometheus monitoring system. For applied scenarios related to AGV/AMR management and cargo monitoring, the numerical solution was implemented using an iterative scheme.

To solve the system of equation (12), considering constraints and the quality functional, a gradient method was applied. At each iteration step, the calculation of a new approximation of the load vector u was performed according to the following scheme, where the gradient of the functional $J(u)$ is approximated by the expression $\left[K \cdot u - F + \lambda\left(u - u_{QoS}\right)\right]$:

$$u^{(n+1)} = u^{(n)} - \gamma \cdot \left[K \cdot u^{(n)} - F + \lambda\left(u^{(n)} - u_{QoS}\right)\right], \quad (21)$$

where γ is the iteration step (~0.01), the element size is defined as 1 camera + associated Edge servers, time step: 33 ms (corresponds to 30 fps).

Model adaptation included traffic prioritization via the CRISP protocol and dynamic recalculation of B_i when τ_{ij} changes.

Calculations were performed according to the bitrate equations (22) and for load coefficients (23).

$$B_{ij}^{(n+1)} = B_{ij}^{(n)} + \beta \cdot \left(\tau_{max} - \tau_i^{(n)}\right), \quad (22)$$

Δu_{i}⍰ $= \alpha \cdot (\partial J/\partial u_{\mathrm{i}}$⍰$) + \eta \cdot N(0,1)$, (23)

where β, α, η are adaptation rate coefficients determining the system's speed of response to changes in network conditions.

For replacing matrices K for the "Elbrus" processor (optimization for the AVX-512 instruction set), equations for hardware accelerators were used:

$$\mathrm{K}_{\mathrm{el}brus} = \mathrm{K} \odot M_{ij}, \text{где } M_{ij} = \sigma\left(B_{ij} - B_0\right), \quad (24)$$

where σ is the sigmoid function, B_i is the video stream bitrate (in Mbps), 20 is the bitrate threshold value (optimization for 4K). Thus, for B_i > 20 Mbps: Mij ≈ 1 (maximum performance), for B_i < 20 Mbps: Mij ≈ 0 (load reduction).

This approach allows adapting the computational load to the architecture of the "Elbrus" processor. The mask M, applied through element-wise multiplication ($\odot$), acts as a software-hardware regulator. For streams with a low bitrate (B_{ij} < 20 Mbps), where processing costs may exceed the useful effect, the mask value tends to zero, excluding them from intensive computations supported by AVX-512. For high-bitrate streams (4K and above), the mask M tends to unity, ensuring maximum use of vector instructions for their processing. Thus, the model not only optimizes load distribution in the network but also efficiently distributes CPU computational resources.

Model verification was performed according to the convergence criterion (25).

$$\left|\left|u^{(n+1)} - u^{(n)}\right|\right| / \left|\left|u^{(n)}\right|\right| < \varepsilon,\ \varepsilon = 0.001, \quad (25)$$

The QoS criterion max $\max(\tau_{\mathrm{i}}) \le 100\mathrm{ms}$, $\max(\mathrm{J_i}) \le$ms. SLA checking was performed in parallel with optimization and included transmission delay (E2E Latency), considering the constraint (26) and frame loss (27).

$$\tau i = \tau\text{processing} + \tau\text{network} < \tau\max, \tau\max = 0.1 \text{ sec}, \quad (26)$$

$$P_{loss} = 1 - N_{received}/N_{sent} < 0.01, \quad (27)$$

where N_{sent} and $N_{received}$ are the number of sent and received frames. The final system of model conditions included checking availability, jitter, and bitrate; metrics $video\_latency\_seconds$ and $video\_frame\_loss$ were exported to Prometheus from each node, and threshold settings were configured in alerting rules (Prometheus Alerting Rules).

3.2.1 Interpretation of the results of scenario 2

The optimization of computational load distribution between cloud and edge layers for video analytics is implemented through a structured finite element model (FEM) comprising critical nodes and their interconnections. This system is designed to handle high-bandwidth, low-latency video streams while ensuring reliability and compliance with strict service level agreements (SLAs).

The connection between geo-distributed data centers and peripheral devices (local servers, cameras) is designated as high-availability video traffic. It utilizes industrial standard protocols, including RTSP, ONVIF, and GB28181, supporting modern codecs such as H.265, H.264, and MJPEG. The video stream operates at 4K resolution (3840×2160 pixels at 30 frames per second) with an adaptive bitrate ranging from 4 to 50 Mbps. Physical interfaces include Power over Ethernet++ (IEEE 802.3bt) and USB3 Vision. This connection is bound by stringent boundary conditions: end-to-end transmission delay must not exceed 100 ms, frame loss is maintained below 0.01%, and system availability is guaranteed at 99.99%. The hardware is rated for a mean time between failures (MTBF) exceeding 100,000 hours and operates within an extended temperature range of -40°C to +60°C. The hybrid model governing this connection combines deterministic video encoding processes with stochastic handling of network delays.

A separate control video traffic link connects geo-distributed data centers to peripheral devices (cameras → Edge servers). This channel also supports 4K/1080p resolution using H.265/H.264 codecs over RTSP/ONVIF transport, with a bitrate range of 2-50 Mbps. It similarly employs PoE++ and USB3 Vision interfaces. The performance requirements include <100 ms end-to-end delay, jitter below 5 ms, and 99.98% availability. Cameras maintain an MTBF exceeding 100,000 hours. This adaptive-hybrid model manages deterministic encoding aspects while accounting for stochastic elements in the Power over Ethernet network infrastructure.

The core video analytics pipeline connects the cloud layer (GPU cluster) with peripheral devices (cameras, Edge servers). This high-capacity channel handles RTSP/H.265 streams at 4K/8K resolution, utilizing WebRTC and MQTT protocols alongside GOST R ISO/IEC 24730-1-2017 standard as a replacement for REST API. Traffic prioritization is implemented through the CRISP protocol, with bitrates of 5-50 Mbps per stream. The system employs domestic hardware accelerators, Elbrus and SKIF, instead of NVIDIA DeepStream or Intel OpenVINO solutions. Performance metrics include 50 Mbps per stream capacity, 100 ms round-trip time, QoS DSCP 46 classification, <200 ms end-to-end delay, <15 ms jitter, 99.95% availability, and <0.1% frame loss. This deterministic-stochastic model operates with a normal delay distribution (σ=5 ms) while maintaining jitter under 15 ms and packet loss below 0.1%.

The integrated system represents a sophisticated approach to video analytics load distribution, combining high-performance hardware, adaptive protocols, and rigorous quality standards to ensure reliable operation under demanding conditions.

Implementing this scenario, considering the presented parameters and boundary values, combines optimal distribution of video load (4K/1080p streams) with quality of service control. The proposed system of equations describes support for domestic processors and protocols (SKIF, CRISP) in compliance with GOST R ISO/IEC 24730-1-2017. The monitoring panel, based on the results of measuring real-time management metrics via Prometheus, can display a heatmap of delays between nodes, a graph of load distribution changes (share of processing on edge/cloud) in real time, and data on QoS violations.

The computational complexity of the proposed FEM approach was analyzed and compared with the basic models. Our method demonstrates a time complexity of $O(n^{1,5})$ and a spatial complexity of O(n) for n nodes of the system. The experimental test (Figure 3 and Figure 4) demonstrated a significant increase in performance: a 38% reduction in computing time compared to static load balancing methods, a 35% reduction in end-to-end latency, and 2.1 times faster convergence than using traditional optimization approaches.

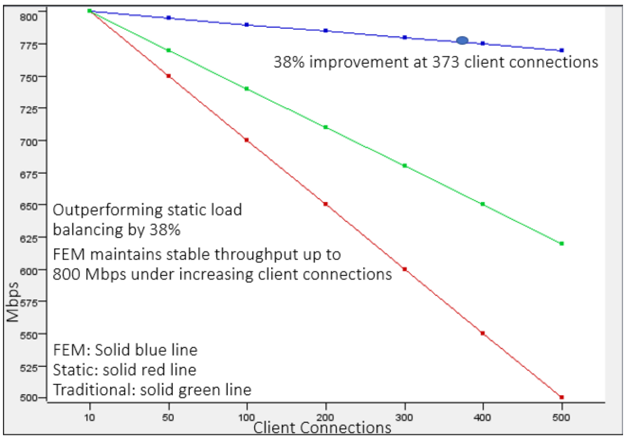

Fig. 3: Throughput scalability analysis

Source: created by the authors

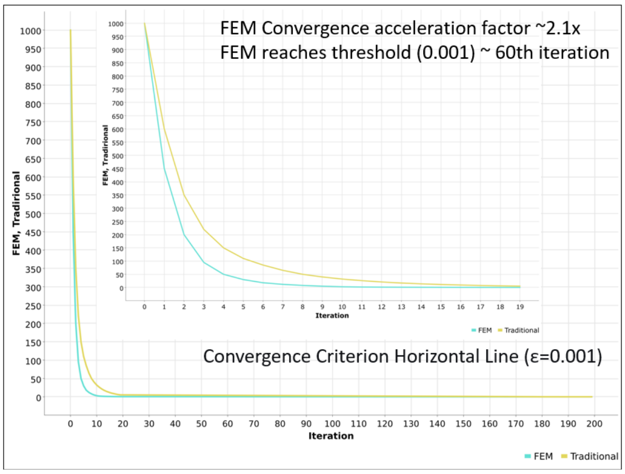

Fig.4: Convergence Behavior of the Functional

Source: created by the authors

A 60% increase in memory efficiency was achieved by optimizing matrix storage and localized data processing.

4 Conclusion

The conducted research successfully achieved its goal of developing a finite element model for optimizing the system architecture of a multimodal transport and logistics center, ensuring the formalization of the system architecture through nodes, connections, and adapted stiffness matrices. The applied mathematical apparatus of the finite element method made it possible to correctly discretize the system, consider critical boundary conditions, and verify the model on real data, which is especially important for the complex IT landscape of the MTLC with its requirements for data infrastructure and the integration layer.

Mathematical models of resource balance equations (processor performance, memory, network) and optimization criteria (minimizing delays, energy consumption) were used to describe the scenarios. Among the adopted limitations, it should be noted that cost optimization criteria were not considered during the implementation of the scenarios, and queueing theory models (task distribution between layers) were not used as a mathematical apparatus.

The practical significance of the work is manifested in the creation of a unified model covering both technical scenarios (optimization of AI inference task distribution, edge computing, and video analytics) and applied tasks (AGV/AMR management, cargo monitoring, and multimodal coordination), with an emphasis on solutions based on national standards. The integration of TSN for temporal synchronization and OPC UA for end-to-end data transmission, as well as the use of Prometheus tools, ensured deterministic real-time data processing.

The FEM approach presented in this work offers a complementary perspective to traditional optimization frameworks in distributed systems. Unlike queuing theory models that excel at analyzing stochastic arrival processes, the FEM provides a spatial-temporal understanding of resource distribution and load propagation. Similarly, while neural network models offer powerful pattern recognition for predictive analytics, they often lack the interpretability and deterministic guarantees of the FEM. The particular strength of our FEM implementation lies in its ability to model the physical architecture of the IT infrastructure as a continuous medium, providing clear engineering insights into how localized changes affect global system behavior. Future work will explore hybrid approaches that combine FEM's structural optimization with queueing theory for workload characterization and machine learning for adaptive parameter tuning.

Research prospects are associated with the further development of the model towards improving the accuracy of load forecasting, expanding support for new industrial standards, which will allow creating a more flexible and scalable system for dynamically developing logistics transport centers. Special attention in future work should be paid to in-depth analysis of the energy efficiency of edge devices and the development of adaptive load balancing algorithms considering seasonal traffic fluctuations, which is critically important for ensuring the stable operation of the MTLC under conditions of constantly growing data volumes and requirements for their processing speed.

References:

[1] A. Shmatko, S. Barykin, S. Sergeev, and A. Thirakulwanich, "Modeling a Logistics Hub Using the Digital Footprint Method – The Implication for Open Innovation Engineering," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 7, no. 1, p. 59, 2021. doi: 10.3390/joitmc7010059.

[2] V. Fialkin and E. Veremeenko, "Characteristics of Traffic Flow Management in Multimodal Transport Hub (by the Example of the Seaport)," *Transportation Research Procedia*, vol. 20, pp. 205–211, 2017. doi: 10.1016/j.trpro.2017.01.053.

[3] Y. Zhou, T. Kundu, M. Goh, and J. Sheu, "Multimodal transportation network centrality analysis for Belt and Road Initiative," *Transportation Research Part E: Logistics and Transportation Review*, vol. 149, p. 102292, 2021. doi: 10.1016/j.tre.2021.102292.https://doi.org/10.1016/j.tre.2021.102292

[4] K. X. Li, M. Li, Y. Zhu, K. F. Yuen, H. Tong, and H. Zhou, "Smart port: A bibliometric review and future research directions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 174, p. 103098, 2023. doi: 10.1016/j.tre.2023.103098.

[5] E. Palkina, "Transformation of business models of logistics and transportation companies in digital economy," *Transportation Research Procedia*, vol. 63, pp. 2130–2137, 2022. doi: 10.1016/j.trpro.2022.06.239.

[6] S. Pettit, Y. Wang, and A. Beresford, "The impact of digitalization on contemporary and future logistics," in *The Digital Supply Chain*, B. L. MacCarthy and D. Ivanov, Eds. Elsevier, 2022, pp. 111–125. doi: 10.1016/B978-0-323-91614-1.00007-1.

[7] E. Tanenbaum, N. Feemster, and D. Weatherall, *Computer networks*, 6th ed. [In

Russian]. St. Petersburg: Peter, 2023, ser. Classics of Computer Science. ISBN: 978-5-4461-1766-6.

[8] A. Hakiri, A. Gokhale, S. Ben Yahia, and N. Mellouli, "A comprehensive survey on digital twin for future networks and emerging Internet of Things industry," *Computer Networks*, vol. 244, p. 110350, 2024. doi: 10.1016/j.comnet.2024.110350.

[9] Y. H. Pan, T. Qu, N. Q. Wu, M. Khalgui, and G. Q. Huang, "Digital twin-based real-time production logistics synchronization system in a multi-level computing architecture," *Journal of Manufacturing Systems*, vol. 58, no. B, pp. 246–260, 2021. doi: 10.1016/j.jmsy.2020.10.015.

[10] E. Brewer, "CAP twelve years later: How the 'rules' have changed," *Computer*, vol. 45, no. 2, pp. 23–29, 2012. doi: 10.1109/MC.2012.37.

[11] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, "Chapter 1 - The standard discrete system and origins of the finite element method," in *The Finite Element Method: its Basis and Fundamentals*, 7th ed. Oxford: Butterworth-Heinemann, 2013, pp. 1–20. doi: 10.1016/B978-1-85617-633-0.00001-0.

[12] R. Ebera and S. Schwarzera, "Toolbox for Efficient Implementation of Digitalization in Logistics," *Procedia CIRP*, vol. 130, pp. 701–704, 2024. doi: 10.1016/j.procir.2024.10.151.

[13] S. Winkelhaus and E. H. Grosse, "Logistics 4.0: a systematic review towards a new logistics system," *International Journal of Production Research*, vol. 58, no. 1, pp. 18–43, 2020. doi: 10.1080/00207543.2019.1612964.

[14] R.David and H. Alla, "On Hybrid Petri Nets," *Discrete Event Dynamic Systems*, no 11, pp. 9–40, 2001. doi: 10.1023/A:1008330914786.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Mamedova N.A. – creation of the research model, activities for creating metadata, accumulating research data, preparation and creation of the manuscript draft, obtaining project financial support; Afanasev M.A. – conducting the research process, editing the article; Makarenkov S.A. – data verification, editing the article; Urintsov A.I. – application of formal methods for analysis and synthesis of research data, providing computational resources.

Sources of Funding for Research Presented in a Scientific Article or the Scientific Article Itself

The research was funded by the grant
Russian Science Foundation № 24-21-20089.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)