UDC: 004.7-048.34(043.2) **DOI: 10.18372/2073-4751.83.20512**

Shklyar O.I.,
orcid.org/0009-0001-6524-6357
Balanyuk Y.V., D.Sc.,
orcid.org/0000-0003-3036-5804
Kudrenko S.O., Ph.D.,
orcid.org/0000-0002-0759-3908

LOAD BALANCING AND CLOUD NODE RESILIENCE ENHANCEMENT ALGORITHM BASED ON PREDICTED INTEGRAL INDEX

State University Kyiv Aviation Institute (KAI)

e-mail: oleh.shklyar@gmail.com,
e-mail: yurii.balaniuk@npp.kai.edu.ua
e-mail: stanislava.kudrenko@npp.kai.edu.ua

Introduction

IoT request flows are characterized by irregularity, stochastic nature, and the presence of short-term spikes, which complicates the maintenance of stable infrastructure functioning [10, 11]. Under such conditions, traditional load balancing mechanisms that make decisions based on current resource metric values often fail to provide the necessary level of adaptability and efficiency [1, 6, 12].

The problem is further exacerbated by the fact that cloud node overloading occurs due to sudden changes in IoT data arrival intensity, preventing balancing mechanisms from reacting in time to potential degradation in Quality of Service (QoS). Furthermore, in the context of ensuring the cyber-resilience of IoT systems, the overloading of individual nodes constitutes a critical security threat. Resource exhaustion is often accompanied not only by reduced service availability but also by increased infrastructure vulnerability to Denial-of-Service (DoS/DDoS) attacks. Under such conditions, the system loses the capability to filter malicious traffic, rendering it an easy target for attackers. Consequently, there is a need to employ methods capable of short-term forecasting of future load parameters and ensuring the preventive distribution of requests among cloud infrastructure nodes [7, 8].

In view of the above, this paper proposes a load balancing algorithm based on the predicted integral node load index. The integral index forms a generalized assessment of the state of computational resources by combining indicators of CPU load, RAM usage, disk operation intensity, and network activity [3]. Subsequent forecasting of this index allows the balancer to account for the expected node load in the immediate time interval and ensure a rational distribution of incoming requests. Thus, the application of the predicted integral node load index contributes to enhancing the reliability and efficiency of cloud system functioning under conditions of highly dynamic IoT traffic and variable resource characteristics.

Problem statement

Cloud infrastructure processing data from numerous IoT devices consists of a set of computational nodes, each possessing limited resources of CPU, RAM, disk subsystem, and network interface. Let

$$N = \{n_1, n_2, \dots, n_k\}$$

be the set of cloud platform nodes , and

$$T = \{t_1, t_2, \dots, t_m\}$$

be the set of incoming IoT requests arriving at the system in real time.

For each node n_i, an integral load index $ILI_i(t)$ is determined (calculation

provided in Formula 1) as an aggregated assessment of its resource utilization at time t [9]. Since the load generated by IoT devices is stochastic and prone to sharp fluctuations, it is necessary to consider not only the current but also the expected value of this index. It is assumed that a short-term forecasting model is available, providing an estimate of $\hat{ILI}_i(t+\Delta)$ for a certain prediction interval Δ.

Formally, the problem consists of selecting a mapping function

$$A: T \to N,$$

which satisfies the following requirements:

1. Minimization of predicted node overload (see forecasting model in Formula 2)***:***

$$min\, max_{n_i \in N} \hat{ILI}_i(t+\Delta).$$

2. Ensuring load distribution uniformity:

$$min\, Var\left(\hat{ILI}_1(t+\Delta), \dots, \hat{ILI}_k(t+\Delta)\right).$$

3. Minimization of expected request service time:

$$min \sum_{t_j \in T} p\left(t_j\right) \cdot RT\left(A\left(t_j\right)\right),$$

where:

t_j *is a specific request from set T;*

$p(t_j)$ *is the probability or frequency of this request's arrival;*

$RT\left(A\left(t_j\right)\right)$ *is the estimated response time of the node to which the request is assigned.*

4. Adherence to resource constraints:

$$CPU_i(t+\Delta) \le CPU_{i,max},$$
$$RAM_i(t+\Delta) \le RAM_{i,max},$$
$$\forall n_i \in N.$$

Thus, the formulation of the load balancing problem in an IoT cloud environment reduces to selecting an optimal request routing strategy based on the predicted integral node load index, which allows for accounting for the dynamic nature of IoT traffic and ensuring efficient computational resource utilization [2, 5]. In this context, the goal is to determine an optimal routing function $A: T \to N$*that assigns each incoming request to a specific cloud node in a way that minimizes overload risks, balances the predicted load across nodes, and reduces the expected service time. Such formulation explicitly defines the decision variable and optimization objectives, thereby completing the formal problem statement.].*

Main material

In cloud systems, the operational state of a node is determined simultaneously by several parameters, each affecting performance in its own way: processor resources dictate computation speed, RAM determines data volumes processable without disk access, disk operation intensity affects data access latency, and network bandwidth influences the timeliness of information exchange. Consequently, analyzing each metric separately does not yield a holistic assessment of the node's real state.

Therefore, an **Integral Load Index** (ILI) is introduced, which combines all these parameters into a single numerical criterion. It serves as a unified generalized characteristic of node loading and reflects its capacity to process additional requests in the nearest time interval. Such an index is a convenient tool in load balancing tasks, as it enables node comparison via a single numerical indicator, provides a comprehensive resource state assessment, is suitable for subsequent short-term forecasting by time-series models, and can serve as a formal criterion for selecting the optimal node for request routing. The integral load index is defined as:

$$ILI_i(t) = w_1 \cdot CPU_i(t) + w_2 \cdot RAM_i(t) + w_3 \cdot IO_i(t) + w_4 \cdot NET_i(t), \quad (1)$$

where:

$CPU_i(t)$ - processor load of node n_i;

$RAM_i(t)$ - RAM usage of node n_i;

$IO_i(t)$ - disk operation intensity of node n_i;

$NET_i(t)$ - network load of node n_i;

w_1, w_2, w_3, w_4 - importance weighting coefficients. The weighting coefficients w_1, w_2, w_3, w_4 reflect the sensitivity of the integral index to variations in individual resource metrics. They can be determined in two ways: (1) manually, based on expert assessment and the dominant workload type, or (2) automatically—through optimization procedures, for example by minimizing the mean-squared error of the load prediction model. In both cases, normalization is applied so that $\sum w_i = 1$. Such parameterization allows the model to adapt to different types of IoT traffic and improves the accuracy of the integral index forecasting.

It is worth noting that the use of the predicted integral index also enables the resolution of related cybersecurity tasks. Anomalous load growth scenarios, identified via deviations of ILI(t) from typical patterns, often correlate with the aberrant behavior of compromised IoT devices (e.g., within botnets) or direct hostile activity. Thus, analyzing the index dynamics creates a foundation for early threat detection, allowing for the integration of load balancing mechanisms with security monitoring systems.

Since the load in cloud systems oriented towards processing IoT data streams changes with high frequency and has a pronounced stochastic character, it is advisable not only to measure the integral index ILI but also to forecast it over a short time interval. The predicted index value allows accounting for expected load dynamics and forming preventive decisions regarding request routing. For this purpose, a sequential forecasting model is used, which operates on time series of resource metrics and provides an estimate of the future node state.

Let $ILI_i(t)$ - be the current value of the integral index of node n_i. Then the forecast for interval Δ is defined as:

$$\hat{ILI}_i(t+\Delta) = f_{pred}(ILI_i(t), ILI_i(t-1), \dots), \quad (2)$$

where f_{pred} - is the short-term forecasting model, specifically based on a Recurrent Neural Network (RNN) of the LSTM type [4]. The obtained predicted index value is used in the balancing algorithm as a key criterion for determining the node's capacity to process additional requests in the immediate time interval.

The algorithm utilizes predicted values of the integral index to determine a node's capacity to accept new requests. The lower the $\hat{ILI}_i(t+\Delta)$, the more suitable the node is. Request distribution is performed proportionally to the "availability" of nodes, determined by the function:

$$Score(n_i) = \frac{1}{\hat{ILI}_i(t+\Delta)}. \quad (3)$$

The graphical interpretation of the proposed algorithm's logic is presented in Fig. 1.

Algorithm Steps:

Input

1. Set of nodes $N = \{n_1, n_2, \dots, n_k\}$

2. Stream of requests $T = \{t_1, t_2, \dots, t_m\}$

3. Forecast time interval Δ

Step 1. Telemetry Collection

4. For each node n_i:

5 Read $CPU_i(t)$, $RAM_i(t)$, $IO_i(t)$, $NET_i(t)$;

6. Calculate current $ILI_i(t)$.

Step 2. Forecasting

7. For each node n_i:

8. Based on history $ILI_i(t)$, $ILI_i(t-1)$, ...;

9. Obtain forecast using LSTM model;

10. $\hat{SILI}_i = ILI_i(t+\Delta)$.

Step 3. Candidate Evaluation

11. For each node n_i:

12. Calculate node suitability as:

13. $core(n_i) = 1/\hat{SILI}_i$.

Step 4. Request Distribution

14. For each new request $t_j \in T$:

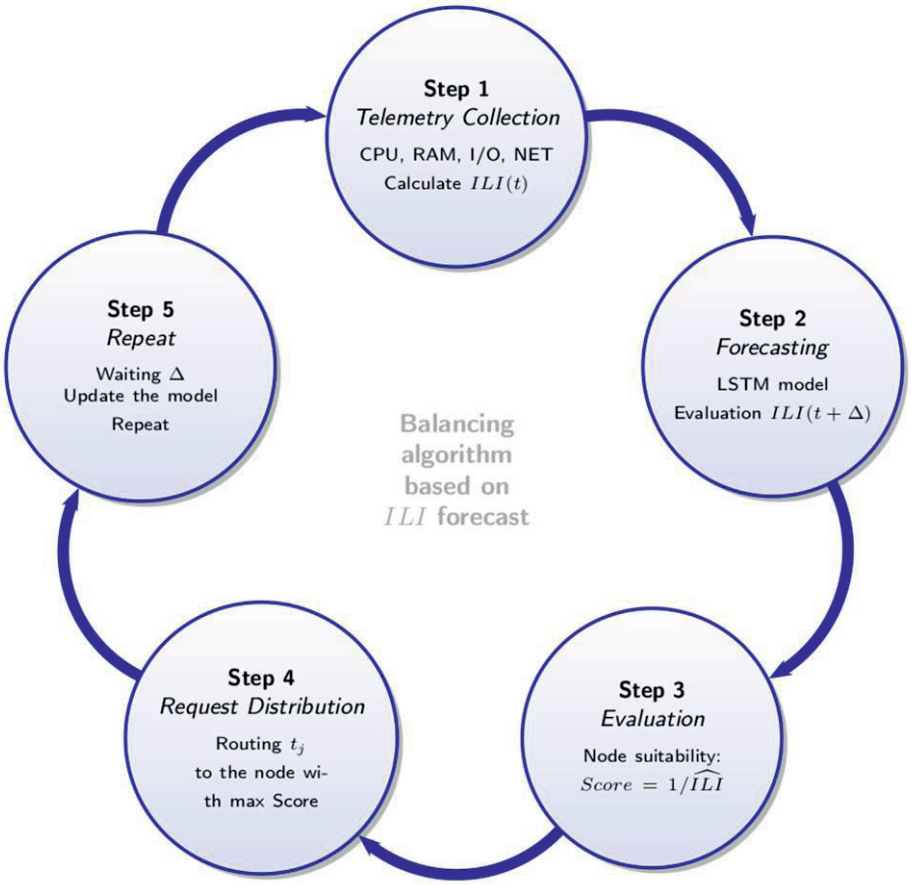

Fig. 1: Cyclic scheme of the request routing algorithm

15. Select node n with maximum Score(n), calculated according to formula (3);

16. Assign $t_j \rightarrow n$.

Step 5. Repeat

17. After interval Δ update telemetry;
18. Retrain/update model if necessary;

19. Repeat Steps 1–4.

Conclusions

The paper develops and formalizes a load balancing algorithm in cloud environments oriented towards processing IoT traffic. The proposed approach differs from traditional methods by utilizing a predicted integral load index, enabling a transition from reactive resource management to preventive management.

Main research results:

1. Metric for node state assessment developed. The concept of an integral index is introduced, aggregating CPU, RAM, disk operations, and network activity indicators, providing a comprehensive assessment of the node's request processing capacity.

2. Forecasting mechanism integrated. The use of a short-term forecasting model (specifically LSTM) allows for accounting for the stochastic nature of IoT traffic and mitigating the impact of sudden peak loads on system stability.

3. Routing strategy optimized. The formalized problem of minimizing expected service time and load variance ensures uniform resource distribution and adherence to QoS requirements.

4. Cloud environment fault tolerance enhanced. It is demonstrated that the

proposed prediction-based routing approach mitigates the risks of service degradation caused by both technical faults and intentional interference, providing an additional layer of infrastructure protection.

Directions for further research include:

- Adaptation of weighting coefficients: Development of a mechanism for dynamic weight change ($w_1 \ldots w_4$) in the ILI calculation formula depending on the type of incoming tasks (e.g., increasing CPU weight for computational tasks or I/O for data processing tasks).

- Comparative analysis of forecasting models: Investigation of the efficiency of other neural network architectures (e.g., GRU or Transformer) to improve forecast accuracy over longer time intervals.

- Experimental validation: Implementation of the proposed algorithm in a real cluster (e.g., Kubernetes) to assess forecasting overhead and measure real system energy consumption.

References

1. Kunwar V. et al. Load Balancing in Cloud — A Systematic Review. *Advances in Intelligent Systems and Computing*. 2018. DOI: 10.1007/978-981-10-6620-7_56.

2. Ameen J.N., Begum S.J. Evolutionary Algorithm Based Adaptive Load Balancing (EA-ALB) in Cloud Computing Framework. *Intelligent Automation & Soft Computing*. 2022. 34(2). P. 1281-1294. DOI: 10.32604/iasc.2022.025137.

3. Lilhore U.K. et al. A multi-objective approach to load balancing in cloud environments integrating ACO and WWO techniques. *Scientific Reports*. 2025. Vol. 15, 12036. DOI: 10.1038/s41598-025-96364-1.

4. Zhang B et al. SWT-CLSTM: A hybrid model for cloud workload prediction combining smooth wavelet transform and contrastive learning. *Journal of King Saud University Computer and Information Sciences*. 2025. 37. DOI: 10.1007/s44443-025-00316-8.

5. Almezeini N.A., Hafez A. Task Scheduling in Cloud Computing using Lion Optimization Algorithm. *International Journal of Advanced Computer Science and Applications*. 2017. 8(11). DOI: 10.14569/IJACSA.2017.081110.

6. Fang Y., Wang F., Ge J. A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. *Lecture Notes in Computer Science*. 2010. 6318:271-277. DOI: 10.1007/978-3-642-16515-3_34

7. Batahari M. et al. Dynamic Load Balancing in Cloud Computing Using Machine Learning. *Conference: 3rd International conference on business analytics for technology and security*. 2025.

8. Bansal S., Kumar M. Deep Learning-based Workload Prediction in Cloud Computing to Enhance the Performance. *Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*. 2023. DOI: 10.1109/ICSCCC58608.2023.10176790Corpus

9. Mohanty S. et al. A Novel Meta-Heuristic Approach for Load Balancing in Cloud Computing. *International Journal of Knowledge-Based Organizations*. 2018. 8(1):29-49. DOI: 10.4018/IJKBO.2018010103.

10. Ranesh Naha R. et al. Deadline-based dynamic resource allocation and provisioning algorithms in Fog-Cloud environment. *Future Generation Computer Systems*. 2019. 104. DOI: 10.1016/j.future.2019.10.018.

11. Mahmud R., Kotagiri R., Buyya R. Fog Computing: A Taxonomy, Survey and Future Directions. *Internet of Everything. Internet of Things (Technology, Communications and Computing)*. 2017. P. 103-130. DOI: 10.1007/978-981-10-5861-5_5.

12. Al-Arasi R.A., Saif A. Task scheduling in cloud computing based on metaheuristic techniques: A review paper. *EAI Endorsed Transactions on Cloud Systems*. 2018. 6(17):162829. DOI: 10.4108/eai.13-7-2018.162829.

Shklyar O.I., Balanyuk Y.V., Kudrenko S.O.

LOAD BALANCING AND CLOUD NODE RESILIENCE ENHANCEMENT ALGORITHM BASED ON PREDICTED INTEGRAL INDEX

The paper addresses the problem of load balancing in cloud environments designed for processing stochastic IoT traffic. It is established that traditional reactive methods are insufficiently effective under conditions of sharp fluctuations in request intensity. An adaptive routing algorithm based on the predicted integral node load index is proposed. This index aggregates CPU, RAM, disk I/O, and network activity metrics into a single criterion. An LSTM recurrent neural network model is used to forecast node states. The problem of minimizing expected service time and overload is formalized. The implementation of the proposed approach enables preventive resource distribution, thereby enhancing system stability. Furthermore, the proposed approach significantly contributes to the cybersecurity and resilience of cloud infrastructure. Within IoT ecosystems, node saturation frequently leads to compromised service availability and heightened susceptibility to Denial-of-Service (DoS) attacks. Leveraging the predicted integral load index facilitates not only the optimized allocation of computational resources but also the early detection of load anomalies attributed to aberrant IoT device behavior or hostile actions. This fosters the integration of load balancing with security monitoring frameworks, bolstering cloud fault tolerance and minimizing service degradation risks arising from both systemic failures and malicious intent.

Keywords: *cloud computing; Internet of Things (IoT); load balancing; integral index; forecasting; security monitoring; LSTM; neural network; resource optimization; response time; stochastic traffic; QoS; routing.*

Шкляр О.І., Баланюк Ю.В., Кудренко С.О.

АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ТА ПІДВИЩЕННЯ СТІЙКОСТІ ХМАРНИХ ВУЗЛІВ НА ОСНОВІ ПРОГНОЗОВАНОГО ІНТЕГРАЛЬНОГО ІНДЕКСУ

У роботі розв'язано задачу балансування навантаження у хмарних середовищах, орієнтованих на обробку стохастичного IoT-трафіку. Встановлено, що традиційні реактивні методи є недостатньо ефективними в умовах різких коливань інтенсивності запитів. Запропоновано адаптивний алгоритм маршрутизації, що базується на прогнозованому інтегральному індексі навантаження вузла. Цей індекс агрегує метрики CPU, RAM, дискових операцій та мережевої активності в єдиний критерій. Для передбачення стану вузлів використано модель рекурентної нейронної мережі LSTM. Формалізовано задачу мінімізації очікуваного часу обслуговування та перевантаження. Використання запропонованого підходу дозволяє забезпечити превентивний розподіл ресурсів, підвищуючи стабільність системи. Додатковою перевагою запропонованого підходу є його релевантність до забезпечення кібербезпеки та стійкості хмарної інфраструктури. У контексті IoT-систем перевантаження окремих вузлів часто супроводжується зниженням рівня доступності сервісів та підвищенням уразливості до атак типу відмова в обслуговуванні. Використання прогнозованого інтегрального індексу навантаження дає змогу не лише оптимізувати розподіл обчислювальних ресурсів, а й своєчасно виявляти аномальні сценарії зростання навантаження, що можуть бути пов'язані з некоректною поведінкою IoT-пристроїв або ворожою активністю. Це створює підґрунтя для інтеграції механізмів балансування навантаження з системами моніторингу безпеки, підвищуючи відмовостійкість хмарного середовища та зменшуючи ризики деградації сервісів під дією як техногенних, так і зумисних впливів.

Ключові слова: *хмарні обчислення; Інтернет речей (IoT); балансування навантаження; інтегральний індекс; прогнозування; моніторинг безпеки; LSTM; нейронна мережа; оптимізація ресурсів; час відгуку; стохастичний трафік; QoS; маршрутизація.*