

АНОТАЦІЯ

Тележенко Д. О. Методи та моделі синтезу архітектури віртуальних розподілених комп'ютерних систем. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 122 Комп'ютерні науки (Галузь знань 12 Інформаційні технології). – Харківський національний університет імені В. Н. Каразіна, Міністерство освіти і науки України, Харків, 2025.

Дисертація присвячена вирішенню актуальної наукової проблеми – розробці методів і моделей синтезу архітектури віртуальних розподілених комп'ютерних систем (ВРКС) для забезпечення їх надійності, продуктивності та адаптивності до змінних умов роботи. Основна мета дослідження полягає у створенні нових алгоритмічних підходів, які дозволять прогнозувати та мінімізувати вплив збоїв у системі, забезпечувати оптимальний розподіл ресурсів та підтримувати стабільну роботу в умовах динамічних змін.

У дисертації виконано глибокий аналіз сучасних підходів до управління ВРКС, зокрема, методів балансування навантаження, динамічного масштабування, автоматичного резервування та моніторингу. Встановлено, що основними викликами у цій сфері є відсутність ефективних засобів для проактивного виявлення та попередження збоїв, а також недостатня інтеграція інструментів машинного навчання для адаптивного управління архітектурою системи.

У першому розділі описано основи аналізу існуючих підходів до синтезу архітектури ВРКС. Зокрема, розглянуто еволюцію технологій віртуалізації, яка сприяла створенню динамічних і масштабованих обчислювальних середовищ. Підкреслено, що перехід від монолітних до мікросервісних і гібридних архітектур дозволив покращити продуктивність, гнучкість і управління ресурсами.

Також у розділі описано класифікацію архітектур ВРКС, таких як монолітні, мікросервісні, централізовані та децентралізовані моделі. Детально

розглянуто переваги та недоліки кожного типу: від простоти управління монолітних систем до складності управління, але високої адаптивності мікросервісних структур. Особливу увагу приділено впливу різних архітектур на масштабованість і оптимізацію системи.

Описано важливість впровадження методів оптимізації ресурсів у ВРКС, включаючи динамічне розміщення віртуальних машин, автоматизоване управління ресурсами та алгоритми планування завдань. Визначено, що застосування інтелектуальних підходів, таких як машинне навчання, дозволяє підвищити ефективність використання ресурсів і адаптацію системи до змін у навантаженні.

Далі у першому розділі розглянуті інструменти прогнозування та управління, які забезпечують проактивний підхід до підтримки стабільності ВРКС. Проведено аналіз існуючих алгоритмів машинного навчання для прогнозування навантаження та управління ресурсами. Визначено, що традиційні підходи не враховують динаміку сучасних систем і не забезпечують достатньої точності прогнозування.

У другому розділі описано процес розробки моделі прогнозування навантаження на сервери з використанням алгоритму LSTM. Основна увага приділяється створенню концептуальної моделі синтезу архітектури віртуальних розподілених систем, яка включає компоненти апаратного забезпечення, гіпервізора, віртуальних машин та модуля управління. Докладно розглянуто їх взаємодію та внесок у загальну продуктивність системи.

Далі у розділі досліджено методи паралельної обробки інформації, такі як метод суміщення незалежних операторів, метод конвеєрної обробки та метод декомпозиційної обробки. Ці підходи використовуються для оптимізації обчислювальних процесів у віртуальних машинах та забезпечення високої ефективності використання ресурсів. Показано, що поєднання декількох методів дозволяє підвищити продуктивність системи у динамічних умовах.

Далі у другому розділі розглянуто процес інтеграції алгоритму LSTM у архітектуру віртуальних розподілених систем. Акцент зроблено на важливості якісної підготовки даних, зокрема їх очищення, нормалізації та структуризації

для тренування моделей. Наведено підходи до збору даних у реальних умовах роботи серверів та їх перетворення для відповідності вимогам алгоритму LSTM. Окрему увагу приділено адаптації LSTM для роботи з часовими послідовностями, що характерні для прогнозування навантаження.

У третьому розділі представлено розробку і тестування модифікованого методу відновлення віртуальних розподілених систем (ВРС) після збоїв. Основна увага приділяється прогнозуванню навантаження та моделюванню поведінки системи під час аварійних ситуацій. Описано створення експериментального середовища за допомогою Docker-контейнерів, балансувальника навантаження Nginx, та інструментів моніторингу, таких як Prometheus і Node Exporter. Проведено моделювання збоїв, яке включало зупинку контейнерів та імітацію перевантаження вузлів для оцінки ефективності методу.

Далі було досліджено, як використання алгоритмів машинного навчання, зокрема LSTM, впливає на точність прогнозування збоїв вузлів. Виявлено, що LSTM забезпечує високу точність у передбаченні аварійних ситуацій, дозволяючи заздалегідь перенаправляти трафік для запобігання втратам даних і збоїв. Порівняльний аналіз показав, що підхід з використанням прогнозування дозволяє значно зменшити час реакції системи на збої та мінімізувати кількість втрачених запитів.

Далі у третьому розділі досліджено процес збору даних, який включав генерацію реалістичних сценаріїв навантаження за допомогою Apache Benchmark та моніторинг ключових показників роботи системи, таких як CPU, RAM, затримка відповіді та кількість оброблених запитів. Для навчання моделі були використані нормалізовані часові ряди, які дозволили моделі LSTM ефективно ідентифікувати аномалії та тренди у поведінці системи.

У четвертому розділі описано процес розробки моделі прогнозування навантаження на сервери у віртуальних розподілених системах з використанням алгоритму LSTM. Основна увага приділяється методології збору, обробки та нормалізації даних, які є ключовими для забезпечення якості моделі. Для дослідження було використано відкриті набори даних з платформи Kaggle, які включають показники використання CPU, RAM, дискових і мережевих ресурсів,

що дозволяє враховувати різноманітність параметрів і забезпечити комплексний аналіз.

Також було досліджене налаштування архітектури LSTM-моделі, яка включала декілька рівнів обробки: вхідний шар для нормалізації даних, LSTM-шари для аналізу часових залежностей, Dropout-шар для запобігання перенавчанню та Dense-шар для отримання остаточного прогнозу. Було визначено оптимальну кількість нейронів для кожного шару та параметри тренування, такі як навчальна швидкість, кількість епох та розмір батчу.

Далі у розділі розглянуто процес навчання та валідації моделі. Використання крос-валідації дозволило мінімізувати ризик перенавчання, а метрики оцінки, зокрема MSE, MAE і R^2 , забезпечили кількісну оцінку точності моделі. Результати тестування підтвердили здатність моделі LSTM ефективно прогнозувати навантаження на сервери навіть за наявності аномалій у даних.

В наступній частині четвертого розділу було проаналізовано ефективність моделі у реальних умовах роботи серверів. Виконані експерименти показали, що впровадження LSTM дозволяє зменшити затримки, мінімізувати втрати запитів і підвищити загальну стабільність системи. Було створено гістограми розподілу затримок, які продемонстрували значне зниження частоти високих затримок у системі після впровадження прогнозування.

В останньому розділі обговорено практичну значущість отриманих результатів. Використання LSTM у прогнозуванні навантаження сприяє оптимізації розподілу ресурсів, що є критично важливим для забезпечення продуктивності та надійності віртуальних розподілених систем. Отримані результати підтверджують ефективність підходу, забезпечуючи фундамент для подальшого розвитку моделей прогнозування у динамічних обчислювальних середовищах.

Сукупність результатів, викладених у дисертації, демонструє ефективність запропонованих методів і моделей для синтезу архітектури віртуальних розподілених систем. Запропонований прогнозно-адаптивний підхід із використанням алгоритму LSTM забезпечує високу точність прогнозування навантаження, оптимізацію розподілу ресурсів та мінімізацію наслідків збоїв у

системах. Практична значущість отриманих результатів підтверджується їхньою здатністю підвищувати продуктивність і надійність віртуальних систем у динамічних умовах. Це відкриває нові перспективи для впровадження інтелектуальних рішень у хмарних середовищах, IoT та інших масштабованих обчислювальних платформах, роблячи внесок у розвиток сучасних інформаційних технологій.

Ключові слова: віртуальні розподілені системи, синтез архітектури, прогнозування навантаження, алгоритм LSTM, машинне навчання, оптимізація ресурсів, інформаційні технології, модифікація, комп'ютерні системи, інтернет речей (IoT), паралельна обробка, моделювання збоїв, моніторинг систем, інтелектуальні обчислювальні платформи.

ABSTRACT

Telezhenko D. Methods and models for the synthesis of architecture in virtual distributed computer systems. – Qualification scholarly paper: a manuscript.

The dissertation submitted for obtaining the Doctor of Philosophy degree in Information Technology: Speciality 122 Computer science. V. N Karazin Kharkiv National University, Ministry of Education and Science of Ukraine, Kharkiv, 2025.

The dissertation addresses a pressing scientific problem: the development of methods and models for the synthesis of architecture in virtual distributed computer systems (VDCS) to ensure their reliability, performance, and adaptability to changing operational conditions. The primary objective of the research is to create new algorithmic approaches that enable the prediction and mitigation of system failures, ensure optimal resource allocation, and maintain stable operation under dynamic changes.

The dissertation includes a comprehensive analysis of contemporary approaches to VDCS management, particularly methods for load balancing, dynamic scaling, automated redundancy, and monitoring. It has been established that the main challenges in this field include the lack of effective tools for proactive failure detection and prevention, as well as insufficient integration of machine learning tools for adaptive system architecture management.

The first chapter outlines the fundamentals of analyzing existing approaches to the synthesis of VDCS architectures. Specifically, it examines the evolution of virtualization technologies, which have facilitated the creation of dynamic and scalable computing environments. It emphasizes that the transition from monolithic to microservice and hybrid architectures has improved performance, flexibility, and resource management.

The text then delves into the classification of VDCS architectures, including monolithic, microservice, centralized, and decentralized models. The advantages and disadvantages of each type are discussed in detail, ranging from the simplicity of managing monolithic systems to the complexity but high adaptability of microservice

structures. Special attention is given to the impact of various architectures on scalability and system optimization.

The importance of implementing resource optimization methods in VDCS is described, including dynamic virtual machine placement, automated resource management, and task scheduling algorithms. It is established that the application of intelligent approaches, such as machine learning, enhances resource utilization efficiency and system adaptation to workload changes.

The chapter also explores prediction and management tools that support a proactive approach to maintaining VDCS stability. An analysis of existing machine learning algorithms for load prediction and resource management is presented. It is noted that traditional approaches fail to account for the dynamics of modern systems and do not provide sufficient prediction accuracy.

The second chapter describes the process of developing a server load forecasting model using the LSTM algorithm. The focus is on creating a conceptual model for synthesizing the architecture of virtual distributed systems, which includes components such as hardware, hypervisor, virtual machines, and a management module. The interaction between these components and their contributions to the overall system performance are examined in detail.

The chapter also explores methods of parallel data processing, such as the method of overlapping independent operators, pipeline processing, and decomposition-based processing. These approaches are employed to optimize computational processes within virtual machines and ensure high resource utilization efficiency. It is demonstrated that combining multiple methods can enhance system performance under dynamic conditions.

Furthermore, the second chapter investigates the integration of the LSTM algorithm into the architecture of virtual distributed systems. Emphasis is placed on the importance of high-quality data preparation, including data cleaning, normalization, and structuring, for training the models. Approaches to data collection under real-world server operating conditions and its transformation to meet the requirements of the LSTM algorithm are presented. Special attention is given to adapting LSTM to handle time-series data, which is characteristic of load forecasting.

The third chapter presents the development and testing of a modified method for restoring virtual distributed systems (VDS) after failures. The primary focus is on load forecasting and modeling system behavior during failure scenarios. The creation of an experimental environment is described, utilizing Docker containers, the Nginx load balancer, and monitoring tools such as Prometheus and Node Exporter. Failure modeling included stopping containers and simulating node overloads to evaluate the method's effectiveness.

The study further investigates how machine learning algorithms, particularly LSTM, influence the accuracy of predicting node failures. It was found that LSTM provides high accuracy in forecasting failure scenarios, enabling proactive traffic redirection to prevent data loss and system downtime. Comparative analysis revealed that the predictive approach significantly reduces system response time to failures and minimizes the number of lost requests.

Additionally, the third chapter examines the data collection process, which involved generating realistic load scenarios using Apache Benchmark and monitoring key system performance metrics, such as CPU, RAM, response latency, and the number of processed requests. Normalized time-series data were used for model training, enabling the LSTM model to effectively identify anomalies and trends in system behavior.

The fourth chapter describes the development of a server load forecasting model for virtual distributed systems using the LSTM algorithm. The primary focus is on the methodology for data collection, processing, and normalization, which are crucial for ensuring model quality. Open datasets from the Kaggle platform were used for the study, including metrics on CPU, RAM, disk, and network resource usage. These datasets provided a diverse range of parameters and enabled comprehensive analysis.

The chapter then explores the configuration of the LSTM model architecture, which included multiple processing layers: an input layer for data normalization, LSTM layers for analyzing temporal dependencies, a Dropout layer to prevent overfitting, and a Dense layer for generating the final forecast. The optimal number of neurons for each layer and training parameters, such as learning rate, epochs, and batch size, were determined.

The process of training and validating the model is discussed further. Cross-validation was used to minimize the risk of overfitting, while evaluation metrics such as MSE, MAE, and R^2 provided a quantitative assessment of the model's accuracy. Testing results confirmed the LSTM model's ability to effectively predict server load even in the presence of data anomalies.

In the next section of the chapter, the model's efficiency in real-world server operation conditions was analyzed. Experiments demonstrated that implementing the LSTM model reduced latencies, minimized request losses, and improved the overall system stability. Histograms of latency distribution revealed a significant decrease in the frequency of high latencies in the system after incorporating load forecasting.

The final part of the chapter discusses the practical significance of the obtained results. The use of LSTM for load forecasting contributes to resource allocation optimization, which is critical for ensuring the performance and reliability of virtual distributed systems. The results validate the effectiveness of this approach, providing a foundation for further development of forecasting models in dynamic computing environments.

The body of results presented in the dissertation demonstrates the effectiveness of the proposed methods and models for synthesizing the architecture of virtual distributed systems. The proposed predictive-adaptive approach using the LSTM algorithm ensures high accuracy in load forecasting, resource allocation optimization, and minimization of failure impacts within the systems.

The practical significance of the obtained results is confirmed by their ability to enhance the performance and reliability of virtual systems under dynamic conditions. This opens new prospects for implementing intelligent solutions in cloud environments, IoT, and other scalable computing platforms, contributing to the advancement of modern information technologies.

Keywords: *virtual distributed systems, architecture synthesis, load forecasting, LSTM algorithm, machine learning, resource optimization, information technologies, modification, computer systems, Internet of Things (IoT), parallel processing, failure modeling, system monitoring, intelligent computing platforms.*