УДК 004.75:004.89

К.т.н. Морозов К.В, магістрант Пшеничний С.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД ІНТЕЛЕКТУАЛЬНОГО РОЗПОДІЛУ НАВАНТАЖЕННЯ У ВЕБ-СЕРВІСАХ НА ОСНОВІ ПРОГНОЗУВАННЯ ТРАФІКУ

Abstract

Morozov Kostiantyn, PhD; Pshenychnyj Serhij, student
Method of intelligent load distribution in web services based on traffic forecasting

The paper presents a hybrid intelligent method for dynamic load balancing in web services based on traffic forecasting. The proposed approach combines a long short-term memory (LSTM) neural network to predict future workload fluctuations and a rule-based decision module to proactively redistribute client requests among servers. The system continuously collects runtime metrics such as CPU, RAM, latency and number of active sessions, and forecasts resource utilization for each node within a short horizon (30–60 s). Based on predicted values, the decision engine dynamically adjusts request routing in Nginx and Docker-based environments. Experimental evaluation demonstrates a reduction in average response latency by 25–35 % and an increase in SLA stability by 10–18 %. The approach can be integrated into cloud infrastructures such as Kubernetes and extended with reinforcement learning models for adaptive scaling.

Вступ

Сучасні веб-сервіси функціонують у середовищі з динамічно змінним навантаженням, що часто ускладнює забезпечення стабільної продуктивності. Кількість користувачів, рекламні кампанії чи зовнішні події можуть спричиняти короткочасні пікові перевантаження серверів, які призводять до зростання часу відповіді та зниження рівня доступності.

Традиційні методи балансування навантаження — *Round Robin*, *Least Connections* або *Weighted Fair Queuing* — застосовують реактивні стратегії, які не враховують прогноз майбутніх змін трафіку. Через це сервери можуть залишатися перевантаженими до моменту повторного перерозподілу запитів, що знижує ефективність системи.

Нещодавні дослідження [1; 2] демонструють перспективність використання методів машинного навчання для прогнозування інтенсивності запитів і побудови адаптивних балансувальників навантаження. Такі підходи дозволяють здійснювати проактивне

управління потоками запитів, зменшуючи час відповіді системи та підвищуючи стабільність роботи у хмарних середовищах.

У зв’язку з цим актуальним є створення інтелектуального методу балансування навантаження у веб-сервісах, який поєднує прогнозування трафіку та адаптивне керування розподілом запитів між серверами.

Постановка задачі

Проблема балансування навантаження у веб-сервісах полягає в тому, щоб рівномірно розподіляти потік користувацьких запитів між кількома серверами таким чином, аби забезпечити мінімальний час відповіді та стабільне виконання угод рівня обслуговування. У більшості реальних систем балансувальники приймають рішення на основі поточних показників — кількості активних підключень, середнього часу обробки або рівня використання ресурсів. Такий підхід є реактивним і не враховує очікувані зміни навантаження, що часто призводить до короткочасного перевантаження окремих вузлів.

У цій роботі задача формулюється як мінімізація середнього часу відповіді системи за рахунок динамічного коригування розподілу запитів між серверами з урахуванням прогнозу трафіку на найближчий часовий інтервал.

Таким чином, мета полягає у створенні методу, який поєднує прогнозування майбутнього навантаження та адаптивне управління потоками запитів у режимі реального часу.

Термінологія

Traffic forecasting (прогнозування трафіку**)** — оцінювання майбутньої інтенсивності запитів на основі аналізу часових рядів попередніх звернень до сервісу. У даному дослідженні прогнозування реалізується за допомогою моделей машинного навчання.

LSTM (Long Short-Term Memory) — тип рекурентної нейронної мережі, призначеної для моделювання часових залежностей у послідовних даних. Цей підхід застосовується для короткострокового прогнозування змін навантаження.

SLA (Service Level Agreement) — угода рівня обслуговування, що визначає гарантовані показники якості сервісу, зокрема максимально допустимий час відповіді системи.

Аналіз існуючих підходів та алгоритмів

Існуючі системи балансування навантаження у веб-середовищах можна умовно поділити на статичні, динамічні та інтелектуальні. Статичні методи, такі як *Round Robin* або *Weighted Round Robin,*

розподіляють запити між серверами у фіксованій послідовності або пропорційно до їхньої обчислювальної потужності. Вони прості у реалізації, проте не враховують поточного стану системи, тому можуть створювати ситуації, коли окремі вузли перевантажені, а інші — майже не використовуються.

Динамічні алгоритми — наприклад, *Least Connections* або *Least Response Time* — приймають рішення на основі актуальних показників, зокрема кількості активних сесій або середнього часу відповіді. Вони краще адаптуються до змін трафіку, але залишаються реактивними: перерозподіл запитів здійснюється лише після того, як перевантаження вже відбулося. У хмарних середовищах також застосовуються механізми автоматичного масштабування, як-от *Horizontal Pod Autoscaler* у Kubernetes, однак і вони не є миттєвими через часові затримки між збором метрик і розгортанням нових екземплярів сервісів.

Інтелектуальні або ML-based підходи пропонують прогнозувати поведінку трафіку на коротких часових горизонтах та проактивно керувати балансуванням навантаження. Для цього використовують методи машинного навчання — регресійні моделі, дерева рішень або нейронні мережі типу *LSTM*. Такі системи здатні враховувати історію запитів, сезонність і раптові пікові зміни. Основна перевага інтелектуальних методів полягає у зниженні середньої затримки відповіді та підвищенні стабільності роботи сервісу навіть під час високої інтенсивності запитів. Недоліком залишається складність побудови моделі прогнозу та потреба в додаткових обчислювальних ресурсах для її навчання та інференсу.

Пропонований метод

Запропонований метод балансування навантаження ґрунтується на гібридному підході, який поєднує прогнозування трафіку за допомогою моделі машинного навчання та адаптивні правила прийняття рішень у системі розподілу запитів. Основна ідея полягає в тому, щоб здійснювати проактивне балансування, коли система змінює маршрутизацію запитів не після виникнення перевантаження, а заздалегідь — на основі прогнозу майбутнього навантаження.

Архітектура методу складається з чотирьох основних компонентів:

1. Модуль збору метрик, який періодично отримує дані про кількість активних запитів, середній час відповіді, використання процесора та пам’яті на кожному сервері.
2. Модуль прогнозування, у якому використовується модель *Long Short-Term Memory (LSTM)* для оцінювання майбутньої інтенсивності запитів $\hat{y}(t+1)$ на інтервалі 30–60 секунд. Модель навчається на

історичних даних трафіку з урахуванням сезонності та короткострокових трендів.

3. Модуль прийняття рішень, який розраховує оптимальний розподіл запитів між вузлами кластера залежно від прогнозованих навантажень і поточних метрик.
4. Балансувальник запитів, який оновлює таблиці маршрутизації у *Nginx* або іншому програмному балансувальнику.

Для кожного сервера s_i визначається інтегральна оцінка навантаження:

$$Score_i = \alpha \cdot \hat{y}_i(t) + \beta \cdot L_i(t) + \gamma \cdot U_i(t),$$

де $\hat{y}_i(t)$ — прогнозована кількість запитів, $L_i(t)$ — середній час відповіді, $U_i(t)$ — рівень використання ресурсів (CPU, RAM), а коефіцієнти α, β, γ визначають вагу кожного параметра.

Після обчислення оцінок $Score_i$ виконується нормалізація:

$$\omega_i(t) = \frac{1/Score_i}{\sum_{j=1}^{n}(\frac{1}{Score_j})},$$

де $\omega_i(t)$ — частка запитів, яку балансувальник спрямовує на сервер s_i. Таким чином, менше значення $Score_i$ відповідає кращому стану вузла, і він отримує більшу частину навантаження.

Робота системи відбувається циклічно: після збору нових метрик прогноз оновлюється, розраховується новий розподіл, і балансувальник адаптує потоки запитів. Завдяки такому підходу система не лише реагує на поточний стан, а й випереджає пікові навантаження, перерозподіляючи запити до їхнього фактичного зростання.

Описану архітектуру реалізовано у вигляді прототипу, що складається з окремих контейнерів Docker. Один контейнер відповідає за моніторинг метрик, другий — за прогнозування навантаження, третій — за обробку запитів балансувальником *Nginx*. Така модульність дозволяє легко інтегрувати рішення у хмарні середовища або розширити його функціональність за допомогою алгоритмів навчання з підкріпленням (*reinforcement learning*).

Результати експериментальних досліджень

Для оцінювання ефективності запропонованого методу було розгорнуто тестовий кластер із трьох веб-серверів у середовищі Docker, де балансування запитів виконувалося за допомогою *Nginx*. Система збирала телеметрію про навантаження (кількість запитів, час відповіді, використання CPU та пам’яті) з інтервалом 5 секунд.

Модель *LSTM* навчалася на даних трафіку веб-додатка за достатній період (10 000 вимірювань). Середня абсолютна похибка прогнозу (MAE) становила близько **7 %**, що забезпечує достатню точність для короткострокового передбачення змін навантаження.

Порівняно з класичним методом *Round Robin*, запропонований підхід зменшив середній час відповіді системи на 25–30 % та підвищив стабільність виконання SLA із 78 % до 91 %. Алгоритм продемонстрував здатність до швидкої адаптації: перерозподіл запитів відбувався до настання пікових перевантажень.

Розроблений метод може бути використаний у комерційних веб-платформах для стабільної роботи під час пікових навантажень (наприклад, під час розпродажів), у потокових сервісах для рівномірного розподілу запитів або у корпоративних аналітичних панелях у режимі реального часу.

Висновки

Дана робота присвячена розробці методу інтелектуального балансування навантаження у веб-сервісах на основі прогнозування трафіку. Запропонований підхід поєднує машинне навчання та адаптивний розподіл запитів, що дозволяє підвищити швидкодію та стабільність веб-додатків без збільшення кількості серверів.

Ідея дослідження полягає у використанні моделі *LSTM* для короткострокового прогнозування інтенсивності запитів і динамічного коригування таблиць маршрутизації в балансувальнику. Проведені експерименти показали зменшення середнього часу відповіді системи на 25–30 % та підвищення рівня виконання *SLA* до 90%.

Подальших досліджень потребує розробка алгоритмів автоматичного підбору параметрів моделі та інтеграція методу з механізмами масштабування у хмарних платформах, зокрема *Kubernetes*.

Література

1. Li, Y., & Wang, X. Intelligent load balancing algorithm for cloud web services based on LSTM prediction [Text] // *IEEE Access*. — 2022. — Vol. 10. — P. 73452–73461.
2. *Nguyen, H., Pham, Q., & Kim, J. Proactive load balancing in Kubernetes clusters using machine learning-based traffic forecasting [Text] // Future Generation Computer Systems. — 2023. — Vol. 141. — P. 482–494.*
3. *Al-Fuqaha, A., & Guizani, M. Machine learning for network traffic prediction and management: A survey [Text] // IEEE Communications Surveys & Tutorials. — 2021. — Vol. 23, № 3. — P. 1629–1658.*
4. *Bendechache, M., & Kechadi, T. Dynamic resource allocation and load balancing in cloud systems using predictive models [Text] // Journal of Cloud Computing. — 2020. — Vol. 9, № 1. — Article 56.*

5. *Liu, J., & Chen, Y. Hybrid traffic forecasting and adaptive routing for web applications [Text] // Applied Soft Computing. — 2021. — Vol. 111. — Article 107681.*
6. Li, S., Zhang, H., & Zhao, K. A reinforcement learning approach to intelligent load balancing in cloud platforms [Text] // *Sensors*. — 2024. — Vol. 24, № 2. — P. 718–729.