

# МЕТОД ПІДВИЩЕННЯ ШВИДКОСТІ ЗАВАНТАЖЕННЯ ВЕБ-СТОРИНОК

Вінницький національний технічний університет

## **Анотація**

*Розроблено програмний засіб для безпечного конфігурування кешування динамічного вмісту веб-сторінок, метою якого є спрощення роботи користувача в процесі конфігурування динамічного контенту, який може бути закешований, зменшення ризику кешування персональних даних, збільшення відсотку закешованих сторінок, які містять динамічний контент, зниження навантаження на основний сервер, та пришвидшення завантаження сторінок для кінцевого користувача*

**Ключові слова:** веб сайт, веб-сервіс, кешування, динамічне кешування

## **Abstract**

*Developed software for secure configuration of caching of dynamic content of web pages, which aims to simplify the user's process of configuring dynamic content that can be cached, reduce the risk of caching personal data, increase the percentage of cached pages containing dynamic content, reduce the load on the main server, and speed up page loading for the end user.*

**Keywords:** website, web service, caching, dynamic caching.

## **Вступ**

Зі збільшенням кількості користувачів глобальної мережі інтернет, проблема доставки об'ємного контенту в інтернеті стає все більш актуальною. Виникає необхідність пошуку методів оптимізації процесів обробки запитів та доставки вмісту веб сторінок. Особливо це актуально для контенту, який потрібно одночасно роздати велику кількість користувачів. Виникають нові і нові технології, такі як балансування трафіку, розподілені мережі доставки контенту, кешування та ін. Але проблема досі не вирішена. І одним з підходів для оптимізації процесу обробки та доставки контенту є пошук способу кешувати динамічно змінювані дані.

Через високу цінність динамічного кешування як у перевагах продуктивності, так і у вартості сервера, здається, що кожен веб-сайт, який існує, захоче скористатися цією практикою. Однак, оскільки документ HTML є основою всієї веб-сторінки, багато веб-сайтів вважають, що надто ризиковано кешувати його. Така ситуація існує через те, що існуючі нині методи конфігурації кешу не дозволяють належним чином налаштовувати виняткове кешування динамічного контенту та перевірити і протестувати конфігурацію кешу.

## **Основна частина**

Динамічний вміст веб-сторінок – вміст, який не є завжди однаковим. Він змінюється в залежності від користувача, для якого цей контент відображається, або з плином часу. Під час зміни контенту в залежності від користувача, контент змінюється залежно від факторів, характерних для певного користувача або групи користувачів, таких як час відвідування, місцезнаходження, пристрій, вік, стать, історія попереднього відвідування інших сторінок та інше. Такий контент також часто називають персоналізованим.

Кешування динамічного вмісту веб-сторінок зводиться до зберігання повністю готової, відрендереної веб-сторінки. На цьому етапі можна визначити що деякий контент ми не можемо кешувати ні в якому разі. Це такий контент який містить персональну інформацію користувача, стан його майнових рахунків, інформацію щодо того як він використовував веб-сайт. Прикладами таких сторінок можуть бути: особистий кабінет користувача або історія його попередніх замовлень в інтернет магазині.

Для кешування інших типів динамічного контенту треба використовувати декілька кардинально різних підходів. Так, для кешування контенту, що змінюється з часом нам достатньо налаштувати

час, протягом якого цей контент може бути актуальним і закешованим. і проводити валідацію кешу на його актуальність. А задача кешування персоналізованого контенту може зводитися до задачі кластеризації користувачів, для відображення однакового контенту певному кластеру.

### Результати дослідження

На сьогоднішній день одним з найпопулярніших підходів до управління кешуванням вмісту сторінок є заголовки запиту Http протоколу.

Він містить поле Cache-Control що використовується для завдання інструкцій по механізми кешування як в запитах, так і у відповідях. Застосовується для завдання політик кешування, серед яких:

- Повна відсутність кешування
- Параметри no-cache, no-store, must-revalidate
- Кешувати, але перевіряти актуальність
- Параметри: no-cache
- Приватний (private) і загальний (public) кеш
- Термін дії (Expiration)
- Перевірка актуальності - параметер "must-revalidate"
- Заголовок Pragma

Даний метод управління кешем є досить складним, для його грамотного використання треба мати неабі які технічні знання, і він не є достатньо гнучким для використання його в управлінні кешуванням динамічного контенту. Для цього необхідний інший підхід, який буде більш інтуїтивно зрозумілий та гнучкий.

Тому для програмного засобу управління кешуванням динамічного вмісту веб-сторінок можна виділити наступні критерії:

- Зручний графічний інтерфейс
- Інтуїтивно зрозумілий та структурований покроковий алгоритм налаштувань
- Можливість детального налаштування кешування для кожної сторінки, або групи сторінок
- Широкий вибір параметрів персоналізації та коректної валідації кешу
- Простий спосіб інтеграції програмного засобу з більшістю платформ та методологій побудови веб-сайтів.

Окрім того необхідна розробка інтелектуального модуля аналізу параметрів та налаштувань для оцінювання можливості кешування та правильного способу кешування веб сторінки, що містить динамічний контент. Загальна схема роботи такого програмного забезпечення зображена на рис. 1.

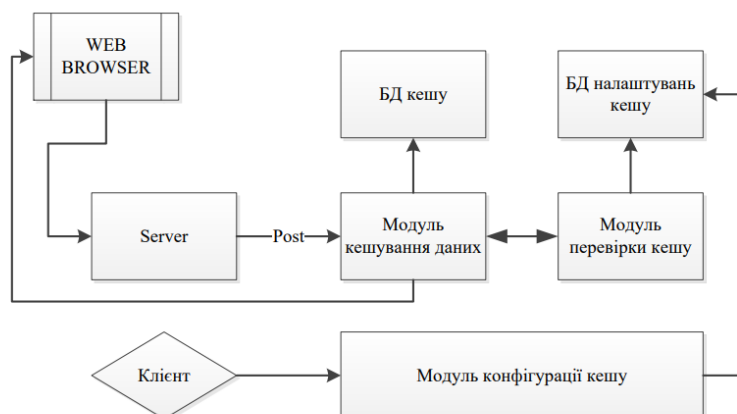


Рисунок 1 – Загальна схема роботи програмного засобу управління кешуванням динамічного вмісту веб-сторінок

Клієнт, або власник веб-сайту за допомогою «Модулю конфігурації кешу» та зручного інтерфейсу налаштовує як він хоче кешувати кожну окрему сторінку сайту і чи хоче взагалі. Далі ці дані потрапляють до «БД налаштувань кешу», де міститься інформація по налаштуванням для кожної сторінки.

Коли веб-браузер звертається до веб-серверу, той, в свою чергу, відправляє запит до «модулю кешування даних». Цей модуль відповідає за збереження та отримання сторінок з кешу. Крім того він звертається до «Модулю перевірки».

Модуль перевірки необхідний для того, щоб визначити, чи можна кешувати сторінку за поданим запитом. Якщо можна, тоді модуль кешування дістає сторінку з кешу. Якщо не можна – сервер самостійно збирає сторінку. А якщо можна, але такої сторінки в кеші поки що немає, то сервер самостійно збирає сторінку, і перед відправленням її користувачу, передає її до «модулю кешування даних» і той кешує її.

Модуль перевірки кешу (рис. 2) отримує запит у форматі Json. Далі перевіряє чи запитується саме html сторінка, а не статичний ресурс чи щось інше, чи можна кешувати сторінку принаймні за деяких обставин, чи немає в запиті параметрів, що забороняють кешування, чи є кеш валідним, і вкінці кінців повертає позитивний або негативний результат.

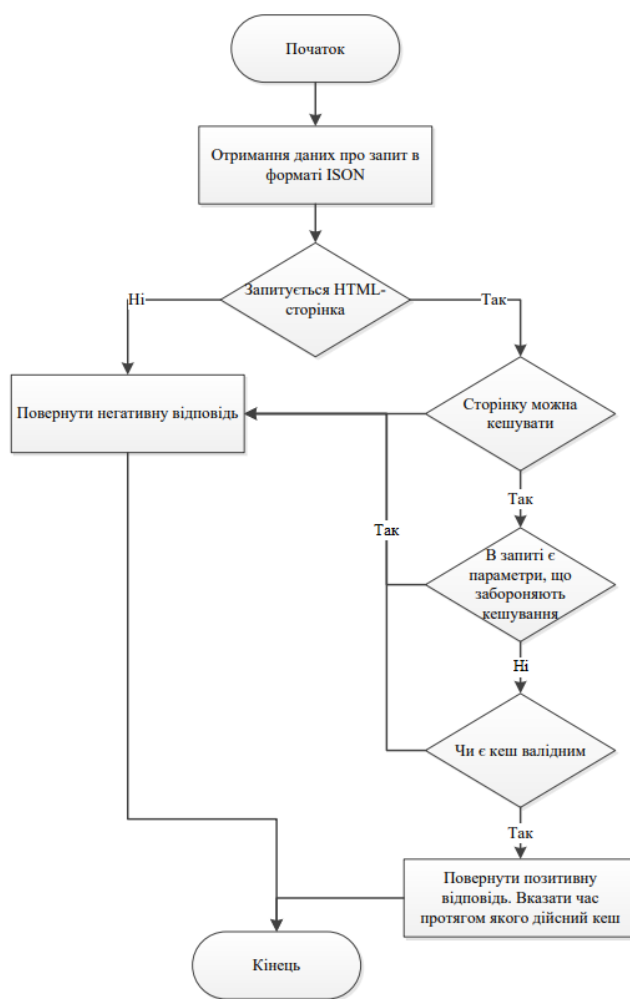


Рисунок 2 – Алгоритм роботи модулю перевірки кешу

## Висновки

Представлений підхід щодо прискорення відображення динамічного контенту відбувається за рахунок створення власної розподіленої системи для налаштування кешування динамічного вмісту веб сторінок, на основі хмарних технологій. Дана система дозволяє безпечно кешувати можливості використання HTML сторінки, що значно знижує навантаження на вихідний сервер, в наслідок чого зме-

ншується вартість хостингу веб-сайтів. За рахунок використання хмарних технологій, розроблений додаток є більш гнучким, легко масштабованим та надійним, внаслідок чого витрати на обслуговування додатку є також значно зниженими.

#### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Adrian Popescu, David Erman, Dragos Ilie, Doru Constantinescu: Internet Content Distribution: Developments and Challenges, 2018.

2. Т. М. Боровська, І. С. Колесник, В. А. Северілов: Оптимізація стратегій розвитку розподілених виробничих систем на базі агрегування виробничих функцій, Вісник Вінницького політехнічного інституту, 2010.

3. Pallis G. and Vakali A., Insight and Perspectives for Content Delivery Networks, Communications of the ACM, Vol 49, No 1, January 2006.

**Курко Владислав Сергійович** – студент групи 2КІ-20м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, Хмельницьке шосе 95, e-mail: v.kurko2012@gmail.com

**Колесник Ірина Сергіївна** - к.т.н, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця, Хмельницьке шосе 95

**Kurko Vladyslav Sergeevich** - student of group 2KI-20m, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, Khmelnytske shosse 95, e-mail: v.kurko2012@gmail.com

**Kolesnyk Iryna Sergeevna** - Cand. Sc. (Eng.), Assistant Professor, Assistant Professor of the Department of Computer Engineering, Vinnytsia National Technical University, Vinnytsia, Khmelnytske Shosse 95