

ДОСЛІДЖЕННЯ, ПРОЕКТУВАННЯ ТА РОЗРОБКА REAL-TIME WEB-ЗАСТОСУНКУ З ВИКОРИСТАННЯМ PHP, ASP.NET, HTML5 ТА WEBSOCKET

Запорізька державна інженерна академія, кафедра ПЗАС

У зв'язку з постійним зростанням кількості користувачів мережі Інтернет і розширенням використання таких технологій як AJAX (Asynchronous JavaScript And XML) значно зростає навантаження серверів постійним потоком запитів, тому необхідні нові підходи до побудови застосунків, які працюють в режимі реального часу.

Будучи синхронним протоколом, HTTP працює за принципом “запит-відповідь”. Тому для отримання свіжої інформації треба постійно оновлювати сторінку, а це, незважаючи на незначний час обробки одного запиту, призводить до суттєвого збільшення навантаження на сервер і робить цей протокол непридатним для вирішення багатьох сучасних задач.

Одним з вирішень цієї проблеми став протокол WebSocket.

Метою даної роботи є дослідження та порівняльний аналіз технологій створення застосунків реального часу та створення тестових real-time web-застосунків без використання протоколу WebSocket на основі PHP/MySQL та з використанням WebSocket на основі HTML5, ASP.NET/SignalR.

Важливістю створення web-застосунків real-time з використанням технології WebSocket є забезпечення безпечної та швидкої передачі даних. Ця технологія зустрічається в додатках Магазину Windows, Google Analytics, іграх в реальному часі, продажу білетів, миттєвих повідомленнях соціальних мереж, відображенні свіжої інформації про курс акцій або погоду, а також в інших додатках.

До останнього часу для відтворення режиму real-time використовувались такі технології[3]:

- 1) Polling (опитування) – постійне опитування сервера Ajax-запитами. *Переваги*: простота реалізації, підтримка у всіх сучасних браузерах. *Недоліки*: затримка в результатах, при зменшенні затримки істотно збільшується навантаження на сервер.
- 2) Long Polling (довгі опитування) – Ajax-запити, які йдуть один за одним, але кожен запит тримається відкритим протягом декількох хвилин. *Переваги*: знижене навантаження на сервер порівняно із звичайним Polling, зменшений трафік, підтримка у всіх сучасних браузерах. *Недоліки*: більше одночасно відкритих з'єднань, тому що кожен запит живе довше.
- 3) Server-Sent Event (передані сервером події) – новий стандарт HTML5, що працює поверх HTTP. Дозволяє створювати довготривалі з'єднання з сервером, щоб сервер міг відправляти дані клієнту. *Переваги*: немає необхідності постійно з'єднуватися з сервером, немає змін на стороні сервера. тому працює на всіх сучасних веб-серверах. *Недоліки*: не підтримується в ІЕ (навіть у ІЕ10), працює тільки в напрямку сервер -> клієнт (на сервер можна відправляти звичайні Ajax-запити).

З появою WebSocket протоколу з'явилися нові можливості. Проведено дослідження та аналіз технології WebSocket. Протокол є досить новим, але, незважаючи на це, отримав широку популярність, включений до специфікації HTML5.

WebSocket - це протокол повнодуплексного двобічного зв'язку поверх TCP-з'єднання, призначений для обміну даними між браузером і веб-сервером у режимі реального часу. Це стало можливим за рахунок того, що тепер немає поняття «клієнт» і «сервер», а є два рівноправних учасника обміну даними[2].

Виявлено переваги протоколу:

- WebSocket-специфікація, розроблена в рамках ініціативи HTML5, представляє WebSocket JavaScript інтерфейс, який визначає повнодуплексне односокетне з'єднання в якому повідомлення можуть бути відправлені між клієнтом і сервером.
- WebSocket забезпечує величезне зниження непотрібного мережевого трафіку і затримок.
- WebSocket являє собою наступний еволюційний крок у веб-зв'язку в порівнянні з Comet і Ajax.

Головна проблема попередніх технологій, велике навантаження на серверне обладнання, затримки, для того щоб з'явилися нові дані необхідно кожного разу оновлювати сторінку у браузері.

Рішенням цих проблем є використання технології WebSocket. Вона дозволяє створювати стійке з'єднання між клієнтом і сервером, зменшуючи навантаження на сервер порівняно з іншими методами.

У якості першого веб-застосунку було розроблено веб-застосунок (чат), який використовує базу даних MySQL, мову веб програмування PHP та відтворення режиму real-time за допомогою JavaScript. В процесі розробки було використано протокол HTTP, який має недолік, а саме те, що він працює за принципом "запит - відповідь", що неприпустимо для веб-додатків реального часу, також виявлено що такий спосіб відтворення режиму реального часу є не ефективним, тому що для відображення нових даних необхідно чекати певний час та кожного разу оновлювати сторінку у браузері[2].

Другий застосунок (у якості порівняння також це Чат) був розроблений за технологією ASP.NET з використанням бібліотеки SignalR та технології WebSocket. *SignalR* - це бібліотека для створення багатокористувацьких real-time ASP.NET (і не тільки) додатків. SignalR управляє з'єднаннями автоматично, і надсилає повідомлення всім підключеним клієнтам одночасно, як в чаті. Можливо також відсилати повідомлення тільки певним клієнтам. З'єднання між клієнтом і сервером постійне, на відміну від класичного HTTP з'єднання, яке повторно встановлює зв'язок для кожного підключення. SignalR підтримує функціональність "server push" (штовхни сервер), коли код на сервері може відправити повідомлення у браузери, використовуючи Remote Procedure Calls (RPC), це швидше працює, ніж звичайна модель "запит-відповідь", існуюча в Інтернеті сьогодні[1].

Порівнюючи ці два застосунки, можна зробити висновки, що новий протокол WebSockets має ряд переваг над протокол HTTP. Проаналізовано технології, які використовувалися до останнього часу для відтворення роботи веб-застосунку у реальному часі.

Висновки:

- вивчені існуючі підходи рішення проблем режиму реального часу;
- досліджено роботу протоколів HTTP та WebSockets;
- оглянуті існуючі реалізації WebSockets протоколу;
- розроблено два веб-застосунки, які відтворюють роботу у режимі реального часу;
- використано бібліотеку SignalR для розробки веб-застосунків;
- виконано порівняння двох веб-застосунків, які були розроблені за різними технологіями, виявлено недоліки та переваги.

Література

1. *Ingebrigtsen E.*, SignalR: Real-time Application Development, 2013.
2. *Lengstorf J., Leggetter P.*, Realtime Web Apps: With HTML5 WebSocket, PHP, and jQuery, 2013.
3. Лабберс П., Олберс Б., Салим Ф., HTML5 для профессионалов. Мощные инструменты для разработки современных веб-приложений, 2011.