

УДК 004.72

ВИБІР АРХІТЕКТУРИ ДОДАТКА ПРИ РОБОТІ З ВЕЛИКИМИ НАВАНТАЖЕННЯМИ

Марковець Н.С., Гаркавий С.Р.

Науковий керівник – к.т.н., доц. Губаренко Є.В.

Харківський національний університет радіоелектроніки, каф. СТ
м. Харків, Україна

тел. +38 (066) 948-41-99, e-mail: nazar.makrovets@nure.ua,

тел. +38 (096) 193-26-61, e-mail: stanislav.harkavyi@nure.ua.

This work is devoted to assessing the efficiency of the right approach to develop a high loaded computer system, especially applications for customers to meet their needs. There are the considered solutions that allow to increase and decrease performance. There were developed three applications with different architecture, which had been relevant at different times. Database indexes impact on data loading is shown. It has been found out that using modern framework versions has more advantages both for performance and the implementation.

В процесі функціонування електронної комерції завжди були актуальними ідеї використання програмних продуктів для реалізації збуту власного продукту та послуг. Останнім часом набагато менше уваги приділяється додаткам для персональних комп'ютерів, через більшу вірогідність платного доступу та вимоги встановлення додатку до локального сховища пристрою, відсутності мобільності. Саме через це, цільовими пристроями сучасних додатків є мобільні пристрої.

Під час розробки додатку для мобільного пристрою, який позбавлений великої потужності та можливості своєчасного охолодження, головним питанням залишається адаптивність застосунку та час його відгуку. Виходячи з приведеного вище, можна сказати, що кінцевими критеріями є: зовнішній вигляд, зручність та швидкодія. Однак, перші два критерія нівелюються у разі відсутності швидкодії, яка неопосередковано залежить від архітектури. Доказом цьому твердженню слугує експеримент, проведений за допомогою двох застосунків з різною архітектурою. Архітектура першого мала актуальність в 2015 році, архітектура другого – сьогодення.

Створено зразок блогу, де розміщено 10 000 статей. Статті розміщені у чотирьох категоріях, кожна стаття має одну або кілька категорій. Дві з цих категорій будуть використовуватись у експерименті. Для того щоб зробити експеримент максимально точним і справедливим, кожен веб-додаток завантажуватиме 10 статей з першої категорії при ініціалізації. Після переходу за посиланням – завантажиться 10 статей із другої категорії.

Додатки викликають одну функцію для отримання даних та використовують одну базу даних для вилучення статей. Розташування застосунку та бази даних знаходиться в однаковому обчислювальному оточенні.

Нижче приведені зображення експериментів. Зліва розташоване зображення ініціалізації сторінки, справа – запит на вилучення даних.

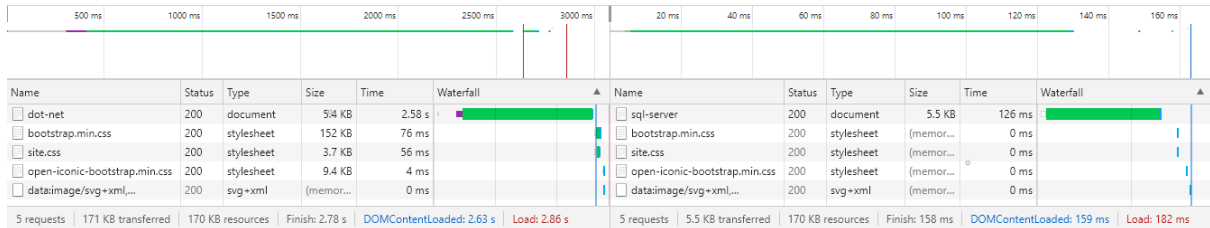


Рисунок 1 – Запити першого застосунку з найстарішим та найгіршим архітектурним рішенням

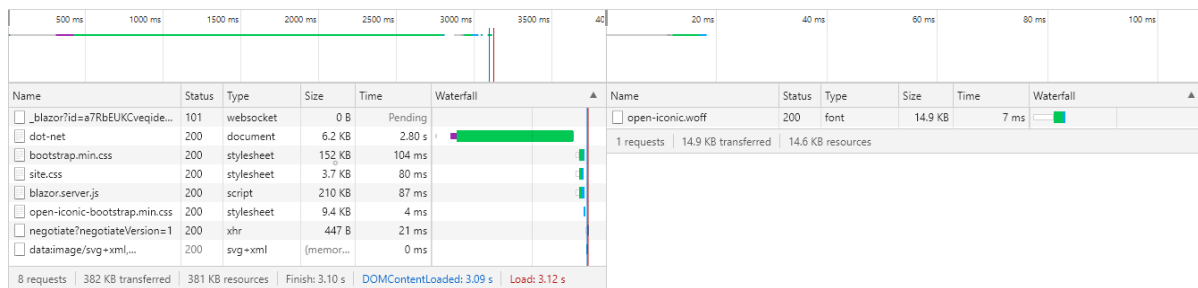


Рисунок 2 – Запити другого застосунку з актуальним та вдалим архітектурним рішенням

Виходячи з наведених результатів можна зробити висновки. Перший застосунок робить на 4 запити більше, після першого завантаження сторінки. Перший застосунок потребує 170 KB на кожний запит, другий застосунок потребує 382 KB на перший запуск (кожен наступний займає 14-20 KB). Сумарно, через 10 запитів до першого застосунку користувач витратить 1.7 MB, до другого ~517 KB. Сумарний час очікування клієнта після 100 запитів до першого застосунку 20.8 секунди, до другого 3.6 секунди.

Список використаних джерел:

1. Alan McSweeney (2019). Introduction to Solution Architecture. https://www.researchgate.net/publication/333295098_Introduction_to_Solution_Architecture.
2. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. (2002). Pattern Oriented Software Architecture. Wiley.
3. L. Bass, P. Clements, R. Kazman. (2003). Software Architecture in Practice. 2-nd edition. Addison-Wesley.