

ДОСЛІДЖЕННЯ ЗАСОБІВ ТА ТЕХНОЛОГІЙ СТВОРЕННЯ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СПІЛЬНОГО РЕДАГУВАННЯ ДОКУМЕНТІВ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Лютий Д. А.¹, Дончик Т. О.², Лемко Р. В.³, Ревзон Н. Д.⁴

^{1,2,3,4} Національний університет кораблебудування імені адмірала Макарова;

^{1,2,3,4} Україна, Миколаїв

¹ljutyjj.dima@gmail.com; ²tetyana.donchuk@nuos.edu.ua; ³lemko_r.v.@gmail.com;

⁴natalya.revzon@nuos.edu.ua

Анотація. Розглянуто трирівневу модель "клієнт-сервер", для розробки розподіленої інформаційної системи спільного редагування документів у режимі реального часу. Наведено порівняльний аналіз технологій розробки програмного забезпечення за всіма рівнями.

Ключові слова: трирівнева модель "клієнт-сервер"; Node.JS.

Сьогодні існує велика кількість технологій та методів розробки програмного забезпечення. Тому треба ретельно підходити до вибору задовільних технологій та методів розробки, які за менший час дадуть кращий результат. Тому треба проаналізувати, що краще всього підійде для розробки розподіленої системи спільного редагування документів у режимі реального часу.

Розробка сервера є найважливіша частина. Сервер буде опрацьовувати усі запити від клієнтів. Для розробки високонавантаженого сервера використовується трирівнева модель "клієнт-сервер", отже запити оброблюються одразу декількома серверами и це дозволяє розподілити та значно зменшити навантаження на сервер (рис.1).

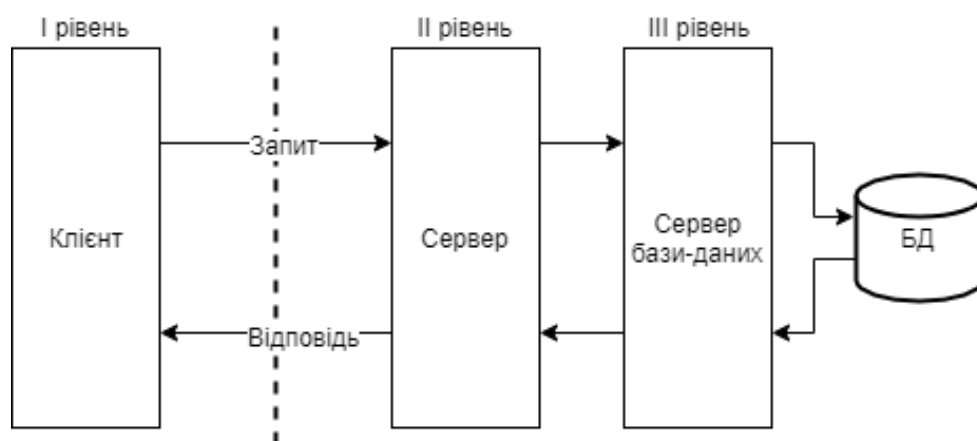


Рисунок 1 – Трирівнева модель "клієнт-сервер"

Для розробки сервера існують дві популярні технології – Node.JS та PHP. Але у Node.JS є переваги у продуктивності (рис.2).

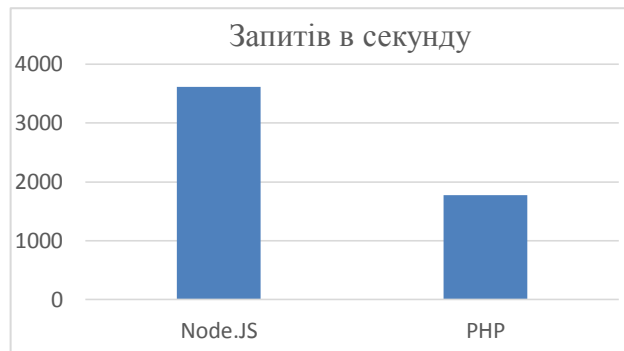


Рисунок 2 – Скільки запитів оброблюється в секунду

Також використання однакових мов програмування на сервері та клієнті веде к пришвидшенню розробки додатків, і як наслідок використовуються однакові бібліотеки. Також є можливість використовувати однакові моделі даних на клієнті та сервері, це не потребує переформатування даних, що не потребує додаткових обчислень.

Обмін інформацією повинен здійснюватися в реальному часі, для цього потрібно використовувати протокол WebSocket. Для більш зручної розробки є обгортка над цим протоколом – Socket.IO. Ця бібліотека дозволяє створити виділений TCP-канал між клієнтом та сервером та обмінюватися інформацією. Обмін повинен здійснюватися зашифрованими даними.

Для шифрування краще використовувати асиметричні алгоритми шифрування, але вони потребують великих обчислювальних потужностей при кожній операції обміну даних. Щоб уникнути цих обчислювань використовується гібридна криптосистема, де асиметричний алгоритм застосовується тільки для обміну ключем для симетричного алгоритму шифрування. Потім сервер та клієнт обмінюється даними зашифрованими симетричним алгоритмом. В даній роботі використовується асиметричний алгоритм – RSA та симетричний алгоритм – AES. Краще використовувати готові бібліотеки алгоритмів, що допоможе уникнути помилок. Також треба звертати увагу, що бібліотека використовує AES-NI набір команд, це значно прискорює шифрування та дешифрування AES (рис. 3). В Node.js є модуль Crypto, який є обгорткою над бібліотекою OpenSSL.

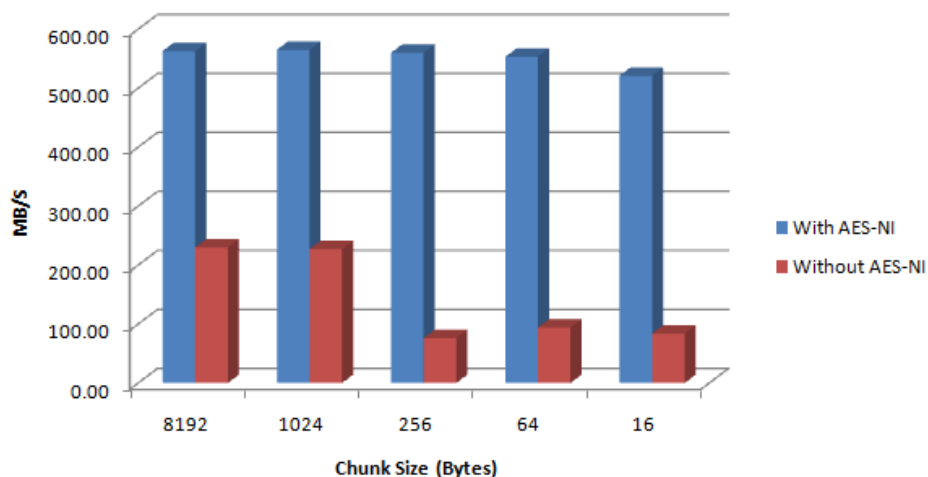


Рисунок 3–Швидкість обробки блоків з використанням AES-NI та без нього

Третій рівень моделі – це сервер бази-даних та база даних. Існує два поширених варіанта – це використання реляційних баз-даних та NoSQL. NoSQL бази даних не мають відношень, як у реляційних базах, замість них використовуються масиви або інші об'єкти. Інформація зберігається у BSON-форматі. Цей формат є бінарним представленням JSON-формату, який є основний формат обміну даними у Node.JS. Реляційні бази діляться на відношення, які мають зв'язки між собою. Це дозволяє створювати більш складні умови вибору даних з таких баз. Реляційні бази працюють швидше на вибірку даних, але NoSQL швидше для оновлення та додавання нових даних (рис. 4). Так як спільне редагування документів потребує швидкого збереження або оновлення існуючих даних, коли як вибірка буде тільки при відкритті існуючих документів, NoSQL бази даних більш доречні.

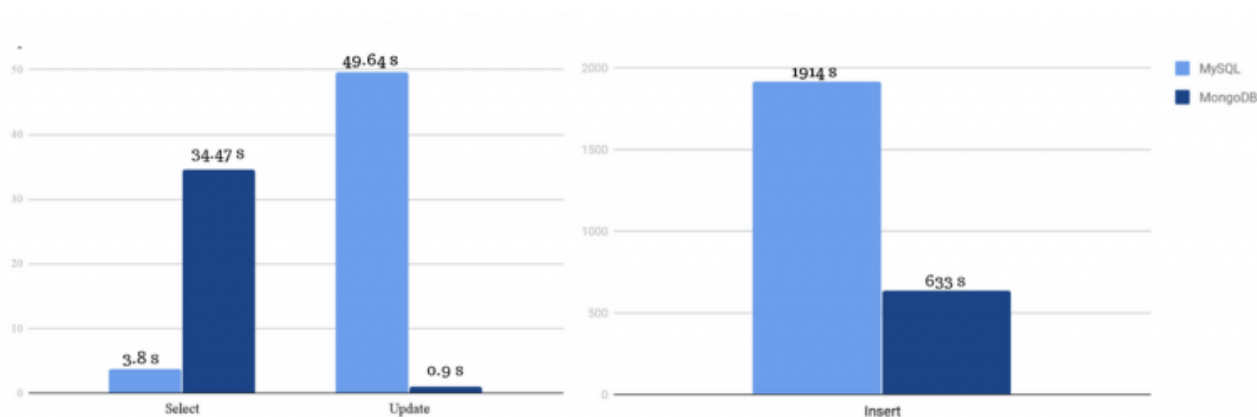


Рис. 4–Порівняння швидкості вибірки, оновлення та вставлення для 1000000 записів

Висновки. Використанням новітніх технологій: Node.JS, Crypto, Socket.IO та MongoDB для створення системи спільного редагування документів у реальному часі забезпечує низьку пов'язаність, здатність до горизонтального та вертикального масштабування. А також дає змогу легкої інтеграції нових методологій розробки програмного забезпечення сім'ї Agile, які збільшують якість та швидкість розробки.

REFERENCES

- [1] Sanchez, R. (2018, May 31). *Comparing Node.js vs PHP Performance*. Retrieved from <https://www.hostingadvice.com/blog/comparing-node-js-vs-php-performance/>.
- [2] Technology, W. (2017, December 28). *Encrypted client-server communication (protection of privacy and integrity with AES and RSA in...*. Retrieved from https://medium.com/@weblab_tech/encrypted-client-server-communication-protection-of-privacy-and-integrity-with-aes-and-rsa-in-c7b180fe614e.
- [3] Yurichev, D. (2019, September 26). *"Reverse Engineering for Beginners" free book*. Retrieved October 2, 2019, from <https://beginners.re/>.
- [4] AES-NI: Hardware Encryption in your Processor. (2011, September 7). Retrieved from <https://datacenteroverlords.com/2011/09/07/aes-ni-pimp-your-aes/>.
- [5] Shah, M. (2019, November 11). *MongoDB vs. MySQL - DZone Database*. Retrieved from <https://dzone.com/articles/comparing-mongodb-amp-mysql>.

Liutyi D. A., Donchyk T. O., Lemko R. V., Revzon N. D.

Methods And Technologies Research Of Creating Distributed Information System Of Co-Editorial Documents In Real-Time

***Annotation.** The three-leveled model "client-server" has been considered for development of distributed information system co-editorial documents in real time. Comparative analysis of development technologies has been given for each level*

***Keywords:** a three-tier client/server architecture; Node.JS.*

Лютый Д. А., Дончик Т. А., Левмко Р. В., Ревзон Н. Д.

Исследование средств и технологий создания распределенной информационной системы совместного редактирования документов в режиме реального времени

***Аннотация.** Рассмотрена трехуровневая модель "клиент-сервер" для разработки распределенной информационной системы совместного редактирования документов в режиме реального времени. Проведен сравнительный анализ технологий разработки программного обеспечения по всем уровням.*

***Ключевые слова:** трехуровневая модель "клиент-сервер"; Node.JS.*