

УДК 004.42:377

Острцова Тетяна Олександрівна

провідний концертмейстер кафедри музичного мистецтва та хореографії
Луганський національний університет імені Тараса Шевченка, Полтава, Україна
t.oxenova@gmail.com
ORCID: 0009-0001-9584-0365

Переяславська Світлана Олександрівна

доцент, кандидат педагогічних наук, викладач кафедри інформаційних технологій і систем
Луганський національний університет імені Тараса Шевченка, Полтава, Україна
pereyaslav9@gmail.com
ORCID: 0000-0001-9873-0447

Острцов Дмитро Іванович

Магістрант
Луганський національний університет імені Тараса Шевченка, Миргород, Україна
dmytroostretsov@gmail.com
ORCID: 0009-0005-6984-760X

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОДУКТИВНОСТІ ПЛАТФОРМ NODE ТА BUN В ОСВІТНІХ ВЕБДОДАТКАХ

Анотація. Стаття присвячена порівняльному аналізу продуктивності платформ Node.js та Bun у контексті розробки освітніх вебдодатків. Сучасні освітні інформаційні системи все частіше реалізуються у вигляді вебзастосунків, що потребують високої продуктивності, масштабованості та ефективного використання ресурсів. У зв'язку з появою нової платформи Bun, яка позиціонується як швидша альтернатива Node.js, виникає потреба у науковому дослідженні її можливостей, переваг і обмежень у практичному застосуванні. В роботі описується проведене експериментальне оцінювання продуктивності вебдодатка на двох платформах, обговорюється доцільність переходу з Node.js на Bun. Розроблено дорожню карту міграції вебдодатку з Node.js на Bun. Для реалізації дослідження було проведено серію навантажувальних тестувань за допомогою утиліти Bombardier. Експерименти виконувалися в хмарному середовищі Google Cloud Platform (GCP) із використанням стандартизованої тестової конфігурації, що дозволило об'єктивно оцінити продуктивність кожної платформи. Результати тестувань засвідчили значну перевагу Bun над Node.js за показником середньої кількості оброблених HTTP-запитів на секунду (RPS). Bun продемонструвала приріст продуктивності в 2,14 рази, обробляючи в середньому 8038 запитів на секунду проти 3753 запитів на секунду у Node.js. Водночас було виявлено певні проблеми сумісності Bun із деякими модулями, що вимагало внесення змін у код вебдодатку. Це свідчить про те, що хоча Bun має значний потенціал для використання у веброзробці, вона все ще перебуває у фазі активного розвитку і процес міграції на нову платформу може вимагати додаткових зусиль для адаптації коду застосунку. Отримані результати є важливими для розробників освітніх вебдодатків, оскільки вибір середовища виконання безпосередньо впливає на продуктивність застосунку, швидкість доступу та масштабованість системи. Перспективними напрямками подальших досліджень вважаємо аналіз стабільності Bun у довготривалому використанні, дослідження сумісності з існуючими модулями та оптимізації під реальні умови високого навантаження.

Ключові слова: node.js; bun; вебдодаток; освітні технології; навантажувальне тестування; цифрова трансформація

Постановка й обґрунтування актуальності проблеми. Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням обсягу даних та вимог до швидкості доступу до інформації. В цьому контексті технології створення освітніх вебдодатків набувають особливої ваги, адже переважна більшість освітніх інформаційних систем реалізується саме у вигляді вебдодатків. Дослідження цих технологій безпосередньо корелює з пріоритетними напрямками наукових досліджень і науково-технічних розробок, визначеними постановою Кабінету Міністрів України №

476 від 30 квітня 2024 р., зокрема, такими напрямками як: технологічні засоби та сервіси програмного інжинірингу, інтелектуальні інтерактивні інформаційно-аналітичні системи та національні інформаційні ресурси [1].

Стратегія цифрового розвитку інноваційної діяльності України до 2030 року, визначає бачення України як держави-лідера у сфері технологій та інновацій [2]. Цей документ окреслює напрями та завдання державної політики для стимулювання цифрової трансформації. Освітні вебдодатки є важливою складовою цифрової трансформації освітнього процесу, оскільки вони забезпечують доступ до навчальних матеріалів, інтерактивного навчання, дистанційної освіти. Розробка та впровадження таких додатків сприяє підвищенню якості освіти, розширенню доступу до знань та розвитку цифрових навичок громадян. Таким чином, дослідження технологій створення освітніх вебдодатків є актуальними, оскільки вони сприяють цифровому розвитку та інноваційній діяльності в Україні.

Однією з популярних платформ для створення освітніх вебдодатків сьогодні є Node.js. Вона забезпечує гнучкість, продуктивність та масштабованість рішень. У вересні 2023 року світ побачила нова технологія під назвою Bun, яка претендує на заміну платформи Node.js. Поява Bun ставить перед науковою спільнотою завдання детально дослідити перспективи її використання у веброботі.

Аналіз наукових досліджень. Дослідженню вебдодатків присвячені роботи таких вітчизняних науковців, як В. Зосімов, О. Войтюк, Д. Плечистий, Г. Півняк та ін. Вагомий також доробок західних дослідників, зокрема D. Brooks, L. Baresi, M. Baker, A. Hoffman, S. Fowler, D. Flanagan, V. Stanwick, E. Brown, G. Kappel та багатьох інших.

В роботі Ю. Линник та ін. [3] досліджується розробка та інтеграція інформаційного чат-бота у вебсайт кафедри прикладної лінгвістики для потенційних та вже зареєстрованих студентів. Для створення чат-бота авторами використано мову програмування JavaScript, середовище виконання коду Node.js та бібліотеку Telegraf. Дослідники розробили інформаційну систему взаємодії чат-бота з користувачами вебресурсів за допомогою природної мови. Сформовано інформаційну структуру чат-бота, обґрунтовано інструменти реалізації, створено чат-бот для кафедри прикладної лінгвістики та протестовано його ефективність. Автори стверджують, що чат-бот ефективно інформує користувачів, однак має деякі недоліки, такі як обмежена можливість обробки складних запитів та контекстуальних нюансів мови. З огляду на швидкий розвиток технологій, можна прогнозувати, що обмеження, згадані дослідниками, можуть бути зменшені або усунені за допомогою новітніх технологій обробки природної мови та розширення функціоналу чат-ботів.

Дослідження В. Zhou та L. Muї [4] виявляє кілька можливих застосувань вебдодатків у формі чат-ботів в освіті медичних сестер. Цього разу, як віртуальних помічників для відповідей на питання студентів та надання навчальних матеріалів, як інструмент для симулювання клінічних ситуацій та відпрацювання навичок у безпечному середовищі, як джерело цілодобової допомоги та безперервного навчання, та як засіб, що зменшує навантаження на викладачів, надаючи їм більше часу для індивідуальної роботи зі студентами. Крім того, автори відзначають потенціал чат-ботів для розвитку критичного мислення у студентів.

Робота С. Sravan та ін. [5] присвячена розробці та впровадженню вебдодатку «Study Buddy» призначеного для покращення співпраці та навчання серед студентів. Він функціонує як онлайн-платформа, що дозволяє студентам обмінюватися знаннями, ставити питання, отримувати допомогу один від одного та взаємодіяти в режимі реального часу. Студенти можуть спілкуватися один з одним, ставити запитання та отримувати відповіді. Платформа дозволяє ділитися матеріалами та інформацією, що стосується навчання. Завдяки своїм функціям «Study Buddy» сприяє створенню активної

та взаємопідтримуючої навчальної спільноти. Дослідники використовували стек технологій MERN (MongoDB, Express.js, React, Node.js) для створення динамічного та адаптивного вебзастосунку. Автори зосередилися на розробці вебплатформи, яка б стимулювала співпрацю серед студентів, полегшувала обмін знаннями та забезпечувала швидкий доступ до інформації. Науковці дійшли висновку, що використання стеку MERN дозволяє створювати масштабовані та ефективні вебдодатки для освіти. Автори стверджують, що їхній додаток покращує навчальний процес та створює сприятливе середовище для навчання. Проте дослідження могло б бути покращене за рахунок більш детального аналізу продуктивності та надійності розробленого додатка. Робота обмежена описом розробки та не містить порівняльного аналізу з іншими підходами або експериментального оцінювання ефективності «Study Buddy».

S. Jaybhaue та ін. дослідили ефективність застосування технології доповненої реальності у навчанні, зокрема, у контексті покращення розуміння складних понять з фізики, хімії, біології та математики [6]. Дослідження включало в себе практичну розробку та тестування вебдодатку. Додаток дозволяє учням переглядати тривимірні моделі об'єктів, а також переглядати додаткову інформацію та відеоматеріали. Автори дійшли висновку, що використання доповненої реальності в освіті може суттєво підвищити залученість учнів та покращити розуміння складних понять. Вони також підкреслили, що інтерактивний характер робить навчання більш цікавим та ефективним. Вебплатформа, розроблена авторами, робить освіту доступнішою, оскільки не вимагає спеціального обладнання окрім смартфона або планшета та доступу до інтернету. Ми вважаємо, що було б корисно дослідити питання продуктивності та масштабування такого типу додатків для більшої кількості учнів та навчальних закладів.

Робота S. Para та ін. досліджує розробку та оцінку вебзастосунку, призначеного для навчання батьків щодо запобігання жорстокому поводженню з дітьми [7]. Автори розуміють, наскільки серйозною є проблема насильства над дітьми, і пропонують вебдодаток як один з ефективних способів боротьби з цим. Додаток має два розділи: один для батьків, а другий для адміністраторів (спеціалістів). У розділі для батьків є доступ до освітніх матеріалів на тему запобігання жорстокому поводженню з дітьми (різні види насильства, їхні наслідки, стратегії профілактики, розвиток навичок виховання дітей, законодавство тощо). Крім того, додаток дозволяє батькам спілкуватися з фахівцями, обмінюватися досвідом та отримувати персоналізовані рекомендації. Вебдодаток, безумовно, має потенціал стати корисним інструментом для навчання батьків та запобігання жорстокому поводженню з дітьми, а робота є цінним внеском у дослідження вебзастосунків для освітніх цілей та заслуговує на увагу.

Різні аспекти використання платформи Node.js під час створення освітніх вебдодатків дослідили О. Ткаченко та В. Бондар [8]. Автори зосередились на аналізі сучасних технологій для розробки платформ онлайн-навчання. Науковці підсумовують наявні погляди на сучасний підхід до розробки вебсайтів з використанням Node.js, аналізують переваги та недоліки існуючих платформ для онлайн-навчання і роблять висновки про доцільність використання концепції Full Stack JavaScript для розробки сучасних вебдодатків. Для розробки інтерфейсу вони використали HTML, CSS та JavaScript, а для бекенду – Node.js з сервером Express та базу даних MongoDB. Використання концепції Full Stack JavaScript справді може спростити розробку освітніх вебдодатків. Однак, важливо враховувати й обмеження Node.js, наприклад, його нездатність ефективно обробляти складні обчислення, крім того є певні обмеження по продуктивності. Для великих та складних освітніх платформ може знадобитися більш комплексне рішення, яке поєднує в собі переваги різних технологій.

A. Turnip та ін. в своїй роботі підтверджують ефективність використання Node.js для створення легкої та продуктивної системи [9]. Їх додаток показав здатність

обробляти до 400 одночасних з'єднань без значного падіння продуктивності, успішно збираючи та обробляючи дані з датчиків. 400 з'єднань – це доволі значний показник, але для більших систем знадобиться потужніше обладнання, доопрацювання архітектури або використання альтернативних рішень. Для більш повної оцінки ефективності бажано було б бачити детальніший аналіз результатів навантажувального тестування.

Порівнянню продуктивності двох популярних бекенд-технологій для веброзробки: PHP та Node.js присвячена робота Q. Odeniran та ін. [10]. Автори прагнули визначити, яка з цих технологій є ефективнішою для вебдодатків. Для цього вони реалізували у кожній технології відомі алгоритми сортування, а потім виміряли їхню продуктивність. Дослідження показало, що Node.js перевершує PHP за показниками швидкодії. Автори пояснюють перевагу Node.js асинхронною, не блокуючою моделлю введення-виведення, яка дозволяє обробляти кілька запитів одночасно без блокування основного циклу подій. Загалом, результати дослідження свідчать про те, що Node.js є більш ефективною технологією для створення швидких та масштабованих вебдодатків, особливо тих, які потребують обробки великих обсягів даних або великої кількості запитів. Однак, варто зазначити, що вибір конкретної технології залежить від конкретних потреб проєкту, і не можна сказати, що Node.js завжди буде кращим вибором.

Пошуку шляхів оптимізації платформи Node.js присвячена робота І. Кириченко та ін. [11]. Автори опрацювали різні аспекти розробки та розгортання додатків на цій платформі, зосереджуючись на тому, як покращити їхню продуктивність та масштабованість. Важливою частиною дослідження є порівняння різних версій Node.js, яке показало, що оновлення версії впливає на продуктивність. Виявлено значне зростання продуктивності при переході до новіших версій.

І. Ревенчук та В. Стешко в своєму дослідженні архітектурних рішень та методів оптимізації для покращення продуктивності вебдодатків, розроблених з використанням Node.js доходять висновку, що розробка якісного вебдодатку потребує уважного ставлення до архітектури та оптимізації [12]. В роботі авторами детально аналізуються різні архітектурні шаблони, такі як MVC, MVVM та MVP, пояснюються їхні переваги та недоліки. Розглядаються також монолітна, мікросервісна та безсерверна архітектури, з порівнянням їх сильних та слабких сторін. На нашу думку дотримання принципів хорошої архітектури та врахування методів оптимізації є дійсно важливими факторами для успішної розробки та запуску швидкодіючих та надійних вебдодатків, особливо в освітній сфері, де швидкий доступ до інформації є критично важливим.

Аналіз існуючих досліджень виявив певну нестачу знань у порівняльній оцінці продуктивності сучасних платформ на яких працюють освітні вебдодатки. Науковці створюють нові вебдодатки та детально досліджують платформу Node.js, проте ігнорують її перспективну альтернативу – платформу Bun, яка за твердженням розробників демонструє суттєве зростання швидкодії в обробці HTTP-запитів. Існуючі роботи зосереджені на оптимізації вже певною мірою застарілих архітектур. На основі аналізу існуючих наукових доробок сформульовано наступну **гіпотезу** дослідження: платформа Bun забезпечує істотне (не менш ніж двократне) підвищення продуктивності обробки HTTP-запитів порівняно з Node.js при використанні в освітніх вебдодатках.

Мета статті: шляхом порівняльного аналізу продуктивності обґрунтувати ефективність переходу з Node.js на Bun для освітніх вебдодатків.

Завдання дослідження: розробити дорожню карту міграції існуючого вебдодатку з платформи Node.js на платформу Bun; провести експериментальне порівняння продуктивності вебдодатку на платформах Node.js та Bun.

Bun – відносно нова технологія і спільнота розробників продовжує працювати над вирішенням знайдених проблем та покращенням зручності її використання [13]. Тим не менш реліз версії платформи, яка за твердженням розробників готова до продакшн,

була запропонована широкому загалу. Bun – це новітнє середовище виконання коду JavaScript, яке набирає популярності. Воно призначене для швидкого виконання JavaScript, TypeScript, JSX, і завдяки внутрішнім оптимізаціям та архітектурним рішенням, обіцяє високу продуктивність.

Bun як і Node.js – дозволяє нам працювати з файловою системою, операційною системою, взаємодіяти з БД, відповідати на HTTP запити, створювати сервери, тощо. Проте Node.js можна порівняти з конструктором, для роботи з яким нам кожного разу потрібно підключати якісь додаткові частини, модулі для роботи з вебсокетами, інструменти для тестування, для зчитування змінних середовища тощо. Завдяки цьому конструктору ми маємо велику гнучкість системи, але з іншого боку втрачаємо швидкість. Платформа Node.js існує вже доволі давно і різні її доповнення з'являлися в різний час, тому окремі модулі недостатньо оптимізовані під спільну роботу.

В свою чергу Bun – це інструмент «все в одному». З ним розробник отримує практично все необхідне для роботи: відслідковування змін коду, HTTP та вебсокет сервер, транспіляцію, збірку проєкту, роботу з пакетами, тестування. Окрім підтримки стандартного для Node.js NPM надається і власний менеджер пакетів. Також автори наголошують на повній сумісності з Node.js, тобто будь-який проєкт, створений на Node.js, можна без проблем запускати в Bun.

Для того щоб зробити власні висновки, щодо швидкодії нової платформи, а також можливої заміни нею платформи Node ми маємо визначити та порівняти кількість HTTP-запитів на секунду, що спроможний обробити той самий вебдодаток на першій та на другій платформі. Для тестування роботи додатку на різних платформах обрано утиліту «Bombardier». Bombardier – є сучасним, популярним серед розробників інструментом для оцінки продуктивності HTTP-сервісів. Цей інструмент допомагає вимірювати продуктивність HTTP-ендпоінтів, що є критично важливим для оптимізації вебдодатків і забезпечення їх безперебійної роботи при високих навантаженнях.

Bombardier написаний на мові програмування Go і використовує в роботі бібліотеку fasthttp замість стандартної бібліотеки http Go. Це забезпечує високу швидкість і надійність в роботі, дозволяє значно зменшити витрати на обробку запитів порівняно зі стандартними інструментами, і таким чином підвищити точність. Цей інструмент розповсюджується як відкрите програмне забезпечення, що дозволяє спільноті вносити свій вклад у доробку, виправлення помилок та вдосконалення функціональності.

Критерієм оцінювання обрано середню кількість HTTP-запитів, або Request Per Second (RPS), що змогла обробити та чи інша платформа за виділений проміжок часу. Також контролювалося споживання ресурсів кожною платформою під час виконання, щоб уникнути обмежень з боку апаратного забезпечення. Для підвищення точності результатів експеримент проведено 10 разів. Для кожного експерименту виміряно кількість RPS, і на базі цих даних розраховано середні значення за формулою:

$$RPS = \frac{1}{N} \sum_{i=1}^N RPS_i, \quad (1)$$

де N – кількість експериментів,

RPS_i – кількість RPS в i -му експерименті.

Якщо для розробки та налагодження вебдодатка зазвичай цілком достатньо локальної машини, то для оцінки продуктивності доцільним є використання хмарного сервісу для більш точного і реалістичного тестування. При необхідності тестувати на багатьох конфігураціях та масштабах, локальне тестування може вимагати великої кількості ресурсів та може стати витратним. Крім того локальне тестування не дає повної

уяви про те, як вебдодаток буде поводити себе в реальному робочому середовищі. Для проведення експерименту було обрано хмарний сервіс Google Cloud Platform (GCP).

GCP – це хмарна обчислювальна платформа, що надається компанією Google. GCP пропонує масштабовану та гнучку інфраструктуру, підтримує різноманітні технології та мови програмування, і забезпечує високий рівень надійності та безпеки для додатків і даних. Можна використовувати можливості GCP згідно зі своїми потребами, сплачуючи лише за ті ресурси, які реально використовуються. Додатковою перевагою GCP є те, що цей сервіс пропонує безкоштовний рівень (Free Tier), який дозволяє користувачам випробувати та використовувати певні ресурси та послуги GCP без витрат в межах обмеженого обсягу. Free Tier надається для нових клієнтів GCP та надає можливість ознайомитися з хмарними послугами та експериментувати без фінансових витрат. Основні можливості Free Tier від GCP включають послугу Compute Engine, що закриває потреби нашого експерименту. Проте потрібно враховувати, що Free Tier надається на обмежений час та що після завершення цього періоду буде розраховано вартість використаних ресурсів, якщо їх обсяг перевищує ліміти.

В якості операційної системи для проведення експерименту обрано Debian 12 (bookworm). Ця ОС пропонується серед інших як один із можливих образів для віртуальної машини у GCP. В результаті для проведення експерименту було створене наступне середовище (таблиця 1):

Табл. 1

Компоненти середовища для проведення експерименту

Найменування	Конфігурація / Версія
Сервер	GCP Belgium (europe-west1), Тип VM: e2-highcpu-8 Intel Xeon 2.2 ГГц 4 ядра 8 потоків, 8 Гб RAM, SSD 10 Gb, ОС Debian 12
Bun	1.1.29
Node	20.17.0 (LTS)
Bombardier	1.2.6
MongoDB	GCP Belgium (europe-west1), M0 Sandbox (Shared RAM, 512 MB Storage)

Після проходження процедури реєстрації, в консолі GCP необхідно перейти в меню «Compute Engine» для створення за допомогою кнопки «Create instance» нового екземпляру серверу згідно обраних конфігурації. Про успішне створення VM система GCP сигналізує за допомогою повідомлень та відображаючи зелену позначку поряд з назвою нового серверу. Підключитися до віртуальної машини зручно за допомогою вбудованого SSH-клієнта, або будь-якого стороннього.

Отримавши доступ до консолі сервера необхідно виконати певні оновлення списків доступних пакетів програмного забезпечення з офіційних репозиторіїв, а потім встановити ці оновлення. Для цього послідовно виконуємо в консолі команди:

```
sudo apt update
sudo apt upgrade
```

Після успішного оновлення необхідно встановити наступні програми та утиліти, що допоможуть у проведенні експерименту:

- git – система менеджменту версій ПЗ
- curl – популярна службова програма, що дозволяє взаємодіяти з серверами, наприклад створювати запити
- unzip – архіватор
- nvm – Node Version Manager, утиліта, що дозволяє встановлювати, переключати та видаляти різні версії Node.js. Дає можливість тримати на одній машині будь-яку кількість версій Node
- Node.js – програмна платформа

Для встановлення цих програм послідовно виконуємо в консолі наступні команди:

```
sudo apt install git -y
sudo apt install curl -y
sudo apt install unzip -y
curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
source ~/.bashrc
nvm install 20.17.0
```

Наступним етапом встановлюємо платформу Bun:

```
curl -fsSL https://bun.sh/install | bash -s "bun-v1.1.29"
source ~/.bashrc
```

Перевірити коректність роботи встановлених платформ можна за допомогою команд `node -v` та `bun -v`. Також необхідно завантажити та розпакувати архів з мовою програмування Go:

```
wget https://dl.google.com/go/go1.22.0.linux-amd64.tar.gz
sudo tar -C /usr/local -xzf go1.22.0.linux-amd64.tar.gz
export PATH=$PATH:/usr/local/go/bin
```

Інтерпретатор Go потрібен в системі для запуску тестової утиліти Bombardier, останню версію якої необхідно встановити командою:

```
go install github.com/codesenberg/bombardier@latest
export PATH=$PATH:/home/<ваша домашня директорія>/go/bin
```

На цьому етапі підготовку середовища експерименту можна вважати завершеною. Тепер ми маємо встановити на сервер тестовий вебдодаток та перевірити його правильну роботу. В нашому випадку вебдодаток завантажувався з серверу GitHub, після чого інстальовалися необхідні залежності вебдодатку командою:

```
npm ci
```

Запуск вебдодатку відбувався в SSH-консолі віртуальної машини командою:

```
node index.js
```

або у випадку використання замість Node.js платформи Bun:

```
bun index.js
```

При першому запуску вебдодатку виявилось, що він повністю працездатний на платформі Node.js, проте при спробі запуску того самого вебдодатку на платформі Bun виникає помилка. Цей факт свідчить про відсутність анонсованої розробниками повної сумісності проєктів, написаних і працюючих під Node.js з новою платформою. В інформаційному повідомленні про помилку йшлося про те, що система не може знайти модуль `mongodb-extjson`. Проте як виявилось, ланцюгом додається ще низка помилок які після кількох спроб не вдалося виправити.

Для продовження експерименту та перевірки швидкодії Bun було відключено частину функціоналу додатку, що була пов'язана з модулем `mongoose-morgan`, який використовується для запису журналу дій в БД MongoDB. Цей функціонал був скоріше додатковою можливістю вебдодатку.

Видалення несумісного з Bun модулю привело до відновлення працездатності вебдодатку. Після цього запуск сервера як в середовищі Node.js так і на платформі Bun відбувався без помилок.

Для проведення експерименту в одному терміналі запускався сервер вебдодатку, а в іншому стартувала утиліта Bombardier. Подальший алгоритм дій був наступним:

1. В першому терміналі запускаємо сервер вебдодатку командою:

```
node index.js
```

2. В другому терміналі за допомогою утиліти curl створюємо тестовий запит до ендпоінту:

```
curl localhost:8080/test
```

3. Переконавшись, що сервер відповідає на запит без помилок (надсилає у відповідь об'єкт JSON), запускаємо утиліту Bombardier. В параметрах утиліти вказуємо, що тестування має проходити 180 секунд, з використанням 125 одночасних з'єднань, з максимальним таймаутом 5 секунд. Також в параметрах вказуються метод запиту та адреса на яку будуть відправлятися запити:

```
bombardier -c 125 -d 180s --method=GET --timeout=5s localhost:8080/test
```

4. Після завершення роботи утиліти Bombardier фіксуємо середню кількість Requests per second (RPS), тобто запитів на секунду та вносимо результати до таблиці.

5. Зупиняємо роботу сервера в першому терміналі.

6. Запускаємо сервер в першому терміналі проте тепер вже в середовищі Bun командою:

```
bun index.js
```

7. Повторюємо пункти 2–5. В результаті отримуємо таблицю, в якій обчислюємо середню кількість оброблених запитів на секунду для платформи Node та платформи Bun.

З метою отримання більш точних даних було проведено 10 циклів тестування. Як видно з таблиці результатів (таблиця 2) платформа Bun продемонструвала підвищення продуктивності в 2,14 раз у порівнянні з Node.js, обробляючи в середньому 8038 запитів на секунду. Це доволі вражаючий результат, незважаючи на те, що ми не підтвердили анонсоване розробниками платформи Bun підвищення продуктивності у 4,8 рази. Зазвичай розробники створюють особливі умови, що демонструють їх рішення в більш вигідних позиціях. Навпаки, наш експеримент проводився в наближених до реальної роботи умовах, без певних оптимізацій.

Табл. 2
Результати тестування швидкодії платформ утилітою Bombardier

Спроба, №	Node, RPS	Bun, RPS
1	3718,61	8006,35
2	3750,95	7988,08
3	3672,13	8131,81
4	3816,12	8047,98
5	3700,46	7942,80
6	3761,59	8023,33
7	3761,15	8064,75
8	3851,21	7995,45
9	3821,45	8101,45
10	3677,75	8075,46
В середньому, RPS:	3753,14	8037,75

Слід також зазначити, що анонсована розробниками Bun повна підтримка додатків, що працюють на платформі Node.js в ході дослідження виявила певні обмеження. Вебдодаток, що працював під Node.js не запустився в Bun, з повідомленням про помилку в модулі і почав працювати лише після виключення цього модулю з коду. Проте можна стверджувати, що використання платформи Bun в розробці освітніх вебдодатків може суттєво підвищити їх продуктивність. Bun демонструє значно кращі результати навіть на неоптимізованому під неї проєкті. Це свідчить про наявний потенціал Bun особливо для проєктів, де велике навантаження є критичним аспектом.

Той факт, що підтримка в Bun коду розробленого для Node.js на даному етапі не повна, вказує на те, що платформа все ще знаходиться в процесі вдосконалення. Існування таких помилок може призвести до додаткових витрат часу на коригування та тестування коду розробниками, що мають бажання перейти з Node.js на Bun. Проте, з огляду на значне підвищення продуктивності системи, тимчасові незручності можна вважати виправданим компромісом.

Обмеження дослідження. Незважаючи на отримані позитивні результати щодо продуктивності платформи Bun порівняно з Node.js, дослідження має певні обмеження, які необхідно враховувати при інтерпретації результатів:

1. Обмежений масштаб тестування. Експерименти проводились на одному типі вебдодатку, що може не відображати повний спектр можливих сценаріїв використання в освітній сфері. Різні типи освітніх додатків можуть демонструвати відмінні показники продуктивності на досліджуваних платформах.
2. Специфіка тестового середовища. Тестування проводилось у хмарному середовищі Google Cloud Platform з конкретною конфігурацією обладнання, що може не відповідати умовам реальної експлуатації в навчальних закладах з різними інфраструктурними можливостями.
3. Проблеми сумісності. Виявлені під час дослідження проблеми сумісності Bun з деякими модулями Node.js (зокрема, mongoose-morgan) вказують на недостатню зрілість платформи Bun, що може обмежувати її застосування в проєктах з широким використанням сторонніх бібліотек.
4. Відсутність довготривалого тестування. Дослідження не включало оцінку стабільності платформ при тривалому навантаженні, що є важливим аспектом для освітніх систем, які повинні працювати безперебійно протягом навчального процесу.
5. Обмежений набір метрик. Основною метрикою оцінювання була кількість оброблених HTTP-запитів на секунду (RPS), тоді як інші важливі показники, такі як використання пам'яті, енергоефективність не були детально проаналізовані.
6. Відсутність оптимізації під Bun. Тестовий вебдодаток був спочатку розроблений для Node.js і адаптований для Bun без спеціальної оптимізації під особливості цієї платформи, що могло вплинути на результати порівняння.

Врахування цих обмежень є важливим для об'єктивної оцінки результатів дослідження та визначення напрямків подальших наукових розвідок у сфері оптимізації продуктивності освітніх вебдодатків. Втім, результати, отримані в ході дослідження, ясно показують, що Bun має значний потенціал для майбутнього веброботки. Зі зростанням навантаження на мережеві ресурси та збільшенням потреб у швидкісному доступі до даних, платформи, здатні обробляти велику кількість запитів за одиницю часу, стануть незамінними. У той же час, розвиток та вдосконалення Bun мають продовжуватися, зокрема, шляхом поліпшення сумісності з розробленими під Node модулями та виправленням існуючих обмежень.

На основі отриманих результатів можна сформулювати практичні рекомендації щодо застосування Bun в освітніх вебдодатках. Перехід на Bun найбільш доцільний для

високонавантажених систем дистанційного навчання з великою кількістю одночасних користувачів (понад 1000), платформ для проведення онлайн-тестування та систем з інтенсивним обміном даними в реальному часі. Кількісними критеріями для прийняття рішення про міграцію можуть слугувати: час відповіді понад 200 мс при стандартному навантаженні або регулярні відмови системи при пікових навантаженнях. Технічними передумовами успішної міграції є модульна архітектура додатку з мінімальною залежністю від специфічних для Node.js модулів, використання стандартних API та можливість тимчасового паралельного функціонування обох версій системи. Економічно перехід обґрунтований потенційною економією на серверних ресурсах (приблизно 40-50%), яку слід співвіднести з витратами на адаптацію коду та тестування. Процес міграції доцільно розпочинати з аудиту сумісності модулів, після чого здійснювати поетапний перехід компонентів системи з пріоритизацією найбільш навантажених ділянок коду, супроводжуючи кожен етап ретельним тестуванням продуктивності.

Висновки. Дослідження показало, що вебдодаток перенесений на платформу Bun, може продемонструвати значний приріст продуктивності порівняно з його роботою на Node.js. Це може свідчити про ефективність та оптимізацію, які забезпечує платформа Bun, що робить її привабливим варіантом для розробки високонавантажених вебдодатків. Результати експерименту є специфічними для даної реалізації вебдодатку. При виборі стеку технологій для веброзробки слід враховувати потреби та характеристики конкретного проєкту, а також здатність розробників адаптувати та оптимізувати код під конкретну платформу. Платформу Bun можна розглядати як перспективну альтернативу Node.js, зокрема для проєктів, де вимоги до продуктивності відіграють ключову роль. Таким чином, нові комбінації технологій, такі як BEAM (Bun, Express.js, Angular, MongoDB), BERM (з React) – можуть виявитися привабливими для розробників, які шукають оптимізовану та ефективну альтернативу у своїх проєктах, не зважаючи на можливі труднощі та ризики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 31 грудня року, наступного після припинення або скасування воєнного стану в Україні: Постанова Каб. Міністрів України від 30.04.2024 № 476. URL: <https://zakon.rada.gov.ua/laws/show/476-2024-п> (дата звернення: 07.03.2025).
2. Про схвалення Стратегії цифрового розвитку інноваційної діяльності України на період до 2030 року та затвердження операційного плану заходів з її реалізації у 2025-2027 роках: Розпорядж. Каб. Міністрів України від 31.12.2024 № 1351-р. URL: <https://zakon.rada.gov.ua/laws/show/1351-2024-р> (дата звернення: 07.03.2025).
3. Lynnyk Y., Krestyanpol L., Rozvod E. Development of a natural language chatbot interface for website users. Eastern-European Journal of Enterprise Technologies. 2024. Vol. 1, №2 (127). P. 35-44. URL: <https://doi.org/10.15587/1729-4061.2024.299200> (дата звернення: 07.03.2025).
4. Zhou B., Mui L. G. Utilising chatbots in clinical nursing education: Application and obstacles. Journal of Clinical Nursing. 2024. URL: <https://doi.org/10.1111/jocn.17089> (дата звернення: 07.03.2025).
5. Constructing a Study Buddy Using MERN (MongoDB, Express.js, React, Node.js) Stack Technologies. C. Sravani et al. Engineering Proceedings. 2024. Vol. 66(1). P. 27. URL: <https://doi.org/10.3390/engproc2024066027>.

6. TeachAR-Augmented reality-based Education application / S. M. Jaybhaye et al. Journal of Physics: Conference Series. 2023. Vol. 2601. №1. P. 012012. URL: <https://doi.org/10.1088/1742-6596/2601/1/012012> (дата звернення: 07.03.2025).
7. Design and evaluation of child abuse web-based application for parent education & strengthen / S. Para et al. BMC Public Health. 2024. Vol. 24. №1. URL: <https://doi.org/10.1186/s12889-024-18248-9> (дата звернення: 07.03.2025).
8. Ткаченко О., Бондар В. Деякі аспекти використання Node.js та MongoDB під час створення освітніх вебзастосунків. Цифрова платформа: інформаційні технології в соціокультурній сфері. 2023. № 6(1). С. 76-89. URL: <https://doi.org/10.31866/2617-796x.6.1.2023.283958> (дата звернення: 07.03.2025).
9. Design of smart farming communication and web interface using MQTT and Node.js / A. Turnip et al. Open Agriculture. 2023. Vol. 8, №1. URL: <https://doi.org/10.1515/opag-2022-0159> (дата звернення: 07.03.2025).
10. Odeniran Q., Wimmer H., Rebman C. Node.js or PHP? Determining the better website server backend scripting language. Issues in Information Systems. 2023. URL: https://doi.org/10.48009/1_iis_2023_128 (дата звернення: 07.03.2025).
11. Кириченко І., Назаренко А., Попов Р. Оптимізація та масштабування Node.js додатків. Біоніка інтелекту. 2020. Т.2. № 95. С. 28-31. URL: [https://doi.org/10.30837/bi.2020.2\(95\).04](https://doi.org/10.30837/bi.2020.2(95).04) (дата звернення: 07.03.2025).
12. Ревенчук І., Стешко В. Архітектурні рішення і методи оптимізації для підвищення продуктивності додатків на Node.js та Vue.js. Біоніка інтелекту. 2022. Т. 1. № 98. С. 64-69. URL: [https://doi.org/10.30837/bi.2022.1\(98\).08](https://doi.org/10.30837/bi.2022.1(98).08) (дата звернення: 07.03.2025).
13. Bun. Bun – A fast all-in-one JavaScript runtime. URL: <https://bun.sh/> (дата звернення: 07.03.2025).

*Матеріал надіслано до редакції 09.03.2025 р.
Затверджено до друку 24.04.2025 р.*

COMPARATIVE ANALYSIS OF THE PERFORMANCE OF NODE AND BUN PLATFORMS IN EDUCATIONAL WEB APPLICATIONS

Tetiana Ostretsova

Chief Accompanist of the Department of Musical Art and Choreography at the Institute of Arts
Luhansk Taras Shevchenko National University, Poltava, Ukraine
t.axenova@gmail.com
ORCID: 0009-0001-9584-0365

Svitlana Pereiaslavska

PhD (Pedagogical Sciences), Associate Professor at the Department of Information Technologies and Systems
Luhansk Taras Shevchenko National University, Poltava, Ukraine
pereyaslav9@gmail.com
ORCID: 0000-0001-9873-0447

Dmytro Ostretsov

Master's Student, Luhansk Taras Shevchenko National University, Poltava, Ukraine
dmytroostretsov@gmail.com
ORCID: 0009-0005-6984-760X

Abstract. This article presents a comparative analysis of the performance of Node.js and Bun platforms in the context of developing educational web applications. Modern educational information systems are increasingly implemented as web applications, requiring high performance, scalability, and efficient resource utilization. With the emergence of Bun, a new platform positioned as a faster alternative to Node.js, there is a need for scientific investigation of its capabilities, advantages, and limitations in practical application. This study describes an experimental evaluation

of a web application's performance on both platforms and discusses the feasibility of migrating from Node.js to Bun. A roadmap for migrating a web application from Node.js to Bun is also developed. The research involved a series of load tests conducted using the Bombardier utility. Experiments were performed in the Google Cloud Platform (GCP) cloud environment using a standardized test configuration, enabling an objective performance assessment of each platform. The test results demonstrated a significant advantage of Bun over Node.js in terms of the average number of HTTP requests processed per second (RPS). Bun showed a 2.14-fold performance increase, processing an average of 8038 requests per second compared to 3753 requests for Node.js. However, certain compatibility issues between Bun and some modules were identified, necessitating modifications to the web application's code. This suggests that while Bun holds considerable potential for web development, it is still under active development, and the migration process might require additional effort to adapt the application code. The obtained results are valuable for developers of educational web applications, as the choice of runtime environment directly impacts application performance, access speed to materials, and system scalability. Promising areas for future research include analyzing Bun's stability in long-term use, its compatibility with existing modules, and optimization under real-world high-load conditions.

Keywords: node.js; bun; web application; educational technology; load testing; digital transformation

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Resolution of the Cabinet of Ministers of Ukraine (2024). On Approval of the List of Priority Thematic Directions for Scientific Research and Scientific-Technical Developments until December 31 of the year following the cessation or cancellation of martial law in Ukraine #476.
<https://zakon.rada.gov.ua/laws/show/476-2024-п> (in Ukrainian).
2. Order of the Cabinet of Ministers of Ukraine (2024). On Approval of the Strategy for Digital Development of Innovation Activities in Ukraine for the Period until 2030 and Adoption of the Operational Action Plan for Its Implementation in 2025-2027, #1351-r.
<https://zakon.rada.gov.ua/laws/show/1351-2024-п> (in Ukrainian).
3. Lynnyk, Y., Krestyanpol, L., & Rozvod, E. (2024). Development of a natural language chatbot interface for website users. *Eastern-European Journal of Enterprise Technologies*, 1(2 (127)), 35-44.
<https://doi.org/10.15587/1729-4061.2024.299200>
4. Zhou, B., & Mui, L. G. (2024). Utilising chatbots in clinical nursing education: Application and obstacles. *Journal of Clinical Nursing*.
<https://doi.org/10.1111/jocn.17089>
5. Sravani, C., Kumar, P., Priya, S., Yadav, S. K., Rao, M. J., & Prasan, U. D. (2024). Constructing a Study Buddy Using MERN (MongoDB, Express.js, React, Node.js) Stack Technologies. *Engineering Proceedings*, 66(1), 27.
<https://doi.org/10.3390/engproc2024066027>
6. Jaybhave, S. M., Natekar, D., Nayakodi, P., Raut, N., & Jahagirdar, O. (2023). TeachAR-Augmented reality-based Education application. *Journal of Physics: Conference Series*, 2601(1), 012012.
<https://doi.org/10.1088/1742-6596/2601/1/012012>
7. Para, S., Shahrokhi, H., Maserat, E., & Mohammadzadeh, Z. (2024). Design and evaluation of child abuse web-based application for parent education & strengthen. *BMC Public Health*, 24(1).
<https://doi.org/10.1186/s12889-024-18248-9>
8. Tkachenko, O., & Bondar, V. (2023). Some Aspects of Using Node.js and MongoDB in Creating Educational Web Applications. *Digital Platform: Information Technologies in Sociocultural Sphere*, 6(1), 76-89.
<https://doi.org/10.31866/2617-796x.6.1.2023.283958> (in Ukrainian).

9. Turnip, A., Pebriansyah, F. R., Simarmata, T., Sihombing, P., & Joelianto, E. (2023). Design of smart farming communication and web interface using MQTT and Node.js. *Open Agriculture*, 8(1).
<https://doi.org/10.1515/opag-2022-0159>
10. Odeniran, Q., Wimmer, H., & Rebman, C. (2023). Node.js or PHP? Determining the better website server backend scripting language. *Issues in Information Systems*.
https://doi.org/10.48009/1_iis_2023_128
11. Kyrychenko, I. , Nazarenko, A., & Popov, R. (2020). Optimization and scaling Node.js apps. *Bionics of Intelligence*, 2(95), 28-31.
[https://doi.org/10.30837/bi.2020.2\(95\).04](https://doi.org/10.30837/bi.2020.2(95).04) (in Ukrainian).
12. Revenchuk, I., & Steshko, V. (2022). Architectural solutions and optimization methods to improve the performance of node.js and vue.js applications. *Bionics of Intelligence*, 1(98), 64-69.
[https://doi.org/10.30837/bi.2022.1\(98\).08](https://doi.org/10.30837/bi.2022.1(98).08) (in Ukrainian).
13. Bun. (n.d.). Bun – A fast all-in-one JavaScript runtime.
<https://bun.sh/>