



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.77

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕД ИСПОЛНЕНИЯ JAVASCRIPT КОДА NODE.JS, DENO И BUN ДЛЯ РАЗРАБОТКИ СЕРВЕРНОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЯ

¹ Смирнов А.А., ² Черенков А.В., ³ Подольский Е.А.

ФГАОУ ВО "НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ (ИТМО)", Санкт-Петербург, Россия (197101, город Санкт-Петербург, Кронверкский пр-кт, д. 49 литер а), e-mail: ¹smirnov.andrew.1999@yandex.ru, ²art.cherenkov@gmail.com, ³podolskijgor51@gmail.com

Актуальность исследования обусловлена стремительным развитием веб-технологий и повсеместным использованием JavaScript не только на клиентской стороне, но и на сервере. С увеличением сложности веб-приложений и переходом к разработке fullstack, вопросы эффективности и безопасности исполнения серверного JavaScript становятся ключевыми. Анализ зарубежных и отечественных научных источников по данной теме позволит выявить современные тенденции, проблемы и подходы к оптимизации выполнения серверного JavaScript кода, а также причины появления новых сред исполнения кода, что влияет на разработку высокопроизводительных и надежных веб-приложений. Целью работы является анализ зарубежных и отечественных научных источников, связанных со средами исполнения JavaScript кода Node.js, Deno и Bun, и сравнение этих сред между собой. В результате проделанного исследования были проанализированы зарубежные и отечественные источники по теме исследования. Найдены различия между средами выполнения и какие концепции привели к этим различиям. Составлена сравнительная таблица.

Ключевые слова: Веб-приложение, серверная часть, контент, производительность.

COMPARATIVE ANALYSIS OF THE JAVASCRIPT CODE EXECUTION ENVIRONMENTS OF NODE.JS, DEMO AND BAN FOR DEVELOPING THE BACKEND OF A WEB APPLICATION

¹ Smirnov A.A., ² Cherenkov A.V., ³ Podolsky E.A.

NATIONAL RESEARCH UNIVERSITY OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS (ITMO), St. Petersburg, Russia (197101, St. Petersburg, Kronverkskiy pr-kt, 49), e-mail: ¹smirnov.andrew.1999@yandex.ru, ²art.cherenkov@gmail.com, ³podolskijgor51@gmail.com

The relevance of the research is due to the rapid development of web technologies and the widespread use of JavaScript not only on the client side, but also on the server. With the increasing complexity of web applications and the transition to fullstack development, issues of efficiency and security of server-side JavaScript execution are becoming key. An analysis of foreign and domestic scientific sources on this topic will reveal current trends, problems and approaches to optimizing the execution of server-side JavaScript code, as well as the reasons for the emergence of new code execution environments, which affects the development of high-performance and reliable web applications. The purpose of the work is to analyze foreign and domestic scientific sources related to the execution environments of the JavaScript code Node.js, Deno and Bun, and comparing these environments with each other. As a result of the research, foreign and domestic sources on the research topic were analyzed.

Differences between runtimes have been found and which concepts have led to these differences. A comparative table has been compiled.

Keywords: Web application, backend, content, performance.

Объектом исследования являются программные среды для разработки серверной части веб-приложения.

Теоретической основой исследования стали труды авторов: И. В. Родыгина, А. Наливайко, М. С. Берьянов, Р. Ф. Фархутдинов, М. Хамидуллин, I. Kniazev, A. Fitiskin, M. Abbadini, J. J. Merelo-Guervós, M. Garcia-Valdez, P. A. Castillo, А. Моисеев, F. Doglio, D. Koper, M. Woda.

Информационная база исследования: eLIBRARY, Springer, CyberLeninka, ResearchGate, Google Scholar.

Задачи исследования:

1. Поиск зарубежных и отечественных научных источников, описывающих среды исполнения серверного JavaScript кода.
2. Изучение и анализ источников.
3. Написание заключения и выводов по анализированным источникам.

В статье авторов Родыгина И.В., Наливайко А.В. «Сравнительный анализ технологий для разработки серверной части системы управления продажами» [1] говорится, что быстрая скорость выдачи контента способствует удовлетворению пользователей, что может повысить конверсию и удержание клиентов. Поэтому такая цель как повышение производительности ещё долго будет актуальной, и инновационные инструменты, предлагающие такое улучшение, имеют все шансы развиться и стать трендовыми. Здесь интересны тесты производительности Node.js в сравнении с Java и PHP. В тесте на выдачу чистого текста и в тесте сериализации Node.js занимает второе место, уступая лишь Java библиотеке Vert.X, и в десятки раз обгоняя PHP. При этом Node.js обеспечивает минимальную задержку между ответами. В тесте на множественные запросы к базе данных Node.js уступает Java с фреймворком Spring, но обгоняет PHP с Laravel.

В контексте разработки программного обеспечения на языке JavaScript существуют средства, позволяющие внедрить строгую типизацию, такие как TypeScript. Одновременно, возможность использования асинхронного кода является встроенной функциональностью JavaScript, не требующей использования сторонних библиотек. Однако, при сравнительном анализе производительности, JavaScript на платформе Node.js показывает результаты, уступающие языку Java.

Для того чтобы увеличить конкурентоспособность языка JavaScript в этой области, были созданы новые инновационные решения, такие как Bun и Deno. Эти платформы интегрируют поддержку TypeScript "из коробки", заявляют о значительном улучшении производительности по сравнению с Node.js и предлагают инновационные решения некоторых проблем, рассматриваемых далее в работе.

Берьянов М. С. в статье «Исследование платформы Deno в сравнении с Node.js» [2] рассматривает вхождение Node.js в предпочтительный стек технологий для разработки приложений, обозначая его как ведущий тренд, который завоевал популярность, в значительной степени благодаря массовому сообществу разработчиков. В данной статье

уделяется внимание Deno как потенциальному конкуренту Node.js. Проводится краткий анализ двух платформ (таблица 1). В большей степени рассматриваются интересные аспекты инноваций, предложенных в Deno, особенно в сравнении с Node.js.

Таблица 1 – Сравнение Node.js и Deno при анализе [2]

	Node.js	Deno
Совместимость с браузером	В значительной степени	Полная
Поддержка TypeScript	Необходимо устанавливать библиотеку	Встроенная
Обеспечение безопасности	Не настраивается	Выдаются разрешения при запуске кода
Управлении модулями	В файле package.json задаются все зависимости и загружаются в папку node_modules	Зависимости задаются непосредственно в коде и загружаются при первом запуске

Инновационность в безопасности относительно Node.js проявляется в выполнении в Deno кода в «песочнице». Это означает, что у среды выполнения по умолчанию нет доступа к файловой системе, сети, выполнению других скриптов и к переменным окружения. Разрешения на эти действия выдаются при запуске с помощью флагов, например, флаги --allow-write и --allow-read позволяют более детально контролировать, к каким частям файловой системы может получить доступ процесс.

В работе авторов Фархутдинов Р. Ф., Хамидуллин М. Р. «PHP и NODE.JS в веб-разработке, анализ преимуществ, сравнение и пути развития» [3] предполагается, что Node.js вероятно будет развиваться в сторону увеличения производительности и улучшения работы на многоядерных машинах.

Чепегин, И.Д. в статье «Серверный JavaScript - преимущества и недостатки Node.js» [4] выделяет главное преимущество не столько Node.js, сколько JavaScript – использование единого языка как на стороне клиента, так и сервера (таблица 2). Преимуществом считается уже конкретно Node.js пакетный менеджер NPM, причём именно количество доступных через него модулей и пакетов, то есть огромное сообщество. Отмечено, что в NPM есть недостаток в виде сильной зависимости от пакетов, так как удаление зависимости может привести в нерабочее состояние уже разработанные сервисы. Недостатки Node.js покрываются решением другой платформы.

Таблица 2 – Сравнение Node.js с другими технологиями при анализе [4]

	JavaScript (Node.js)	Другие языки программирования
Использование единого языка для написания клиента и сервера при разработке веб приложения	Присутствует	Отсутствует у большинства
Пакетный менеджер	NPM с миллионами библиотек	Зависит от языка программирования

Тихонов Д.С., Черенков А.В., Долгов А.В. в работе «Сравнительный анализ технологий серверной разработки на платформах DENO и BUN» [5] проводят сравнительный анализ сред исполнения JavaScript кода Deno и Bun (таблица 3). Помимо общей информации о средах проводится ряд экспериментов, в которых сравнивается производительность технологий. На основании полученных данных можно сделать следующий вывод: Bun выделяется в производительности при обработке массивов и рекурсивных вычислениях, что делает его идеальным для сценариев, где требуется высокая скорость обработки данных и интенсивные вычисления.

Таблица 3 – Сравнение Deno и Bun при анализе [5]

	Deno	Bun
Рекомендуемый вариант использования	Веб-серверы или микросервисы	Обработка массивов и рекурсивных вычислениях

С другой стороны, Deno обеспечивает лучшую производительность в асинхронных задачах, что особенно важно для приложений, зависящих от асинхронного ввода-вывода, таких как веб-серверы или микросервисы.

В результате анализа установлено, что новые среды исполнения (Deno, Bun) являются не просто копиями Node.js, а каждая из них нацелена решить определённые проблемы и предоставить разработчикам альтернативы, в которых реализованы инновационные идеи (Таблица 4).

Таблица 4 – Итоговое сравнение Node.js, Deno и Bun [6-8]

	Node.js	Deno	Bun
Как проводится обновление среды исполнения	Сторонними инструментами, например, NVM	Встроенными инструментами	Встроенными инструментами
Обеспечение безопасности	Не настраивается	Выдаются разрешения при запуске кода	Не настраивается
Компиляция кода в исполняемый файл	Отсутствует	Присутствует	Присутствует
Совместимость с браузером	В значительной степени	Полная	В значительной степени
Поддержка TypeScript	Необходимо устанавливать библиотеку	Встроенная	Встроенная
Наличие дополнительных инструментов для разработки	Необходимо устанавливать библиотеки	Инструменты для сборки, тестирования, обновления	Инструменты для сборки, тестирования, обновления
Пакетный менеджер	NPM	Не нуждается в пакетном менеджере	NPM
Рекомендуемые модели облачного сервиса	Виртуальные машины	Виртуальные машины	Бессерверные вычисления
Стандартная библиотека	Часто требуются внешние библиотеки	Обширная	Часто требуются внешние библиотеки
Работа с пакетами	В файле package.json задаются все зависимости и загружаются в папку node_modules	Зависимости задаются непосредственно в коде и загружаются при первом запуске	В файле package.json задаются все зависимости и загружаются в папку node_modules
Используемый движок	V8	V8	JavaScriptCore
Дата выпуска версии 1.0.0	2015-01-14	2020-05-14	2023-09-08

Заключение

Deno будет лучшим выбором при акценте на безопасность, полной совместимости с браузером и удобстве работы с библиотеками.

Vip же стоит выбрать при необходимости быстрой сборки, а также при выполнении затратных синхронных вычислений, и в бессервисных вычислениях.

Node.js стоит выбрать для крупных проектов, в которых важны стабильность работы, проверенная временем, и наличие специальных, узконаправленных библиотек. Также намного больше разработчиков знакомо с Node.js, нежели чем с двумя другими средами.

Список литературы

1. Родыгина И. В., Наливайко А. В. Сравнительный анализ технологий для разработки серверной части системы управления продажами: статья в журнале - научная статья//Известия ЮФУ. Технические науки. №4(221). — 2021. — С. 256-266
2. Берьянов М. С. Исследование платформы Deno в сравнении с Node.js//Студенческий: электрон. научн. журн. 2022. № 40(210). — С. 45-48.
3. Фархутдинов Р. Ф., Хамидуллин М. Р. PHP и NODE.JS в веб-разработке, анализ преимуществ, сравнение и пути развития//Материалы 12-й Международной молодежной научной конференции. В 4-х томах. Том 3. Отв. редактор М.С. Разумов. Курск, 2022. — С. 122-124.
4. Чепегин, И. Д. Серверный JavaScript - преимущества и недостатки Node.js//Вестник науки и образования. – 2020. – № 12-1(90). – С. 18-20.
5. Тихонов Д.С., Черенков А.В., Долгов А.В. сравнительный анализ технологий серверной разработки на платформах DENO и BUN//Научно-технические инновации и веб-технологии. - 2023. - №2. - С. 55-59.
6. Kniazev I., Fitiskin A. Choosing the right javascript runtime: an in-depth comparison of NODE.JS and BUN//Norwegian Journal of Development of the International Science. — 2023. — С. 72-84.
7. Abbadini M. et al. NatiSand: Native Code Sandboxing for JavaScript Runtimes//Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses. – 2023. – С. 639-653.
8. Koper D., Woda M. Performance Analysis and Comparison of Acceleration Methods in JavaScript Environments Based on Simplified Standard Hough Transform Algorithm //International Conference on Dependability and Complex Systems. – Cham: Springer International Publishing, 2022. – С. 131-142.

References

1. Rodygina I. V., Nalivaiko A.V. Comparative analysis of technologies for the development of the server part of the sales management system: an article in the journal - a scientific article // News of the Southern Federal University. Technical sciences. №4(221). — 2021. — pp. 256-266
2. Baranov M. S. Research of the Deno platform in comparison with Node.js // Student: electron. scientific Journal. 2022. № 40(210). — pp. 45-48.

3. Farkhutdinov R. F., Khamidullin M. R. PHP and NODE.JS in web development, analysis of advantages, comparison and development paths // Materials of the 12th International Youth Scientific Conference. In 4 volumes. Volume 3. Editor M.S. Razumov. Kursk, 2022. — pp. 122-124.
 4. Chepegin, I. D. Server-side JavaScript - advantages and disadvantages Node.js // Bulletin of Science and Education. – 2020. – № 12-1(90). – pp. 18-20.
 5. Tikhonov D.S., Cherenkov A.V., Dolgov A.V. comparative analysis of server development technologies on the DENO and BUN platforms // Scientific and technical innovations and web technologies. - 2023. - No. 2. - pp. 55-59.
 6. Kniازهv I., Fitiskin A. Choosing the right javascript runtime: an in-depth comparison of NODE.JS and BUN // Norwegian Journal of Development of the International Science. — 2023. — pp. 72-84.
 7. Abbadini M. et al. NatiSand: Native Code Sandboxing for JavaScript Runtimes //Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses. – 2023. – pp. 639-653.
 8. Koper D., Woda M. Performance Analysis and Comparison of Acceleration Methods in JavaScript Environments Based on Simplified Standard Hough Transform Algorithm //International Conference on Dependability and Complex Systems. – Cham: Springer International Publishing, 2022. – pp. 131-142.
-