

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СРЕД ИСПОЛНЕНИЯ JAVASCRIPT ДЛЯ BACKEND-РАЗРАБОТКИ: ПРАКТИЧЕСКИЙ ПОДХОД К ВЫБОРУ ОПТИМАЛЬНОЙ ПЛАТФОРМЫ

Черенков А.В. (ИТМО)

Научный руководитель – кандидат педагогических наук, доцент Государев И.Б.

(ИТМО)

Введение. Серверная разработка на JavaScript стала революционным этапом после появления Node.js в 2009 году, что позволило использовать один язык для разработки как клиентской, так и серверной части приложений [1]. Эволюция экосистемы привела к появлению новых платформ, таких как Deno (2018) и Bun (2023), каждая из которых решает определенные проблемы Node.js – от безопасности до производительности и удобства работы с TypeScript. В условиях быстрого развития технологий и растущих требований к масштабируемости и эффективности работы серверных приложений становится крайне актуальной задача определения оптимальной среды исполнения в зависимости от конкретных сценариев использования.

Основная часть. Цель работы заключается в комплексном сравнительном анализе производительности сред исполнения JavaScript кода (Node.js, Deno и Bun) для выявления оптимальных сценариев применения. Исследование направлено на определение, какая платформа демонстрирует наилучшие результаты при выполнении различных типов задач, включая стандартные операции CRUD, обработку HTTP-запросов, вычислительно-интенсивные задачи, холодный старт бессерверных функций, а также специфические сценарии, такие как кодирование видео и задачи машинного обучения. Для достижения цели исследования реализованы следующие эксперименты:

- **Вычислительные операции** Измерение времени выполнения задач, требующих интенсивных вычислений, таких как сортировка массивов или матричные операции [2]. Оценивает эффективность платформы в обработке ресурсоёмких вычислений.
- **CRUD-тесты и обработка HTTP-запросов.** Оценка скорости и стабильности выполнения операций создания, чтения, обновления и удаления данных, а также обработки HTTP-запросов. Определяет пригодность платформы для типичных веб-приложений, где важны быстрая реакция на запросы и надёжная работа с данными.
- **Тест холодного старта.** Измерение времени, необходимого для инициализации новой серверной функции или приложения с нуля. Критично для бессерверных архитектур и приложений с переменной нагрузкой, где требуется быстрое масштабирование.
- **Специфические нагрузки (кодирование видео и машинное обучение).** Тестирование производительности при выполнении задач, требующих значительные вычислительные ресурсы, таких как обработка видео или выполнение алгоритмов машинного обучения. Оценивает способность платформы эффективно обрабатывать сложные вычислительные задачи, что важно для приложений с большой нагрузкой на CPU.
- **Анализ влияния аппаратной мощности.** Проведение тестов на серверах с различными конфигурациями (например, разным объёмом оперативной памяти, количеством ядер и разной тактовой частотой процессора) для оценки влияния аппаратных ресурсов на производительность. Помогает понять, как платформа масштабируется и адаптируется к изменениям в доступных ресурсах, что важно при планировании инфраструктуры и оптимизации затрат [3].

Группы экспериментов проводились на машинах с различной аппаратной мощностью, но в идентичных условиях с воспроизводимой конфигурацией, а результаты доступны в открытом доступе для независимой проверки.

Заключение. В рамках данной работы был проведён комплексный сравнительный анализ трёх современных сред исполнения JavaScript (Node.js, Deno и Bun) с целью определения их оптимальной применимости для backend-разработки. На основе полученных данных сформированы объективные рекомендации для разработчиков, позволяющие выбрать оптимальную платформу в зависимости от конкретных задач и условий эксплуатации. Новизна работы заключается в интеграции широкого спектра тестовых сценариев, включая уникальный анализ влияния аппаратной мощности на производительность, что даёт возможность адаптировать рекомендации под реальные условия эксплуатации и масштабирования приложений.

Список использованных источников:

1. Басуматари Б., Агнихотри Н. Преимущества и проблемы использования NodeJS // Международный журнал инновационных исследований в области компьютерных наук и технологий. 2022. DOI: 10.55524/ijirest.2022.10.3.13.
2. Сорокин С. А. и др. Методы оценки производительности вычислительных комплексов // Экономика. Информатика. – 2017. – Т. 42. – №. 9 (258). – С. 89-96.
3. Ирвинг С., Ли Б., Чен Ш., Пэн Л., Чжан В., Дуань Л. Сравнение компьютеров в условиях вариаций производительности // Frontiers of Computer Science. 2018. Т. 14. С. 21–41. DOI: 10.1007/s11704-018-7319-2.