

Д. А. Гордеев

г. Москва

Московский финансово-юридический университет МФЮА

Научный руководитель: кандидат технических наук,

старший научный сотрудник,

С. А. Амелькин

ОБЗОР СТЕКА MERN ДЛЯ РАЗРАБОТКИ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ

Аннотация. Непрерывное развитие информационных технологий и их внедрение в сферу образования определяет потребность в разработке качественных, эффективных и масштабируемых образовательных платформ. Их разработка ведется на основе современных веб-технологий, которые предоставляют возможность организации полноценного клиент-серверного взаимодействия. Рассмотрен высокоэффективный стек подобных технологий и его основные компоненты, предназначенные для обработки данных на различных уровнях реализации образовательной платформы. Определены достоинства нереляционной базы данных над классической реляционной при взаимодействии с информацией. Выявлены преимущества одностраничных приложений. Приведен анализ двух технологий с блокирующим и неблокирующим вводом-выводом, отображающий количество запросов в секунду и среднего времени выполнения этих запросов при повышении уровня параллелизма. Описано взаимодействие стека MERN с компонентами MVC архитектуры.

Ключевые слова: веб-технологии, MongoDB, Express, React, NodeJS, SPA, MVC.

Информационные технологии, используемые при разработке цифровых платформ, принято называть стеком веб-технологий. Это сочетание языков программирования, баз данных, фреймворков и библиотек, используемых для

создания веб-приложений или веб-платформ [2, с. 2]. Наиболее популярным сочетанием для создания высокоэффективных, полнофункциональных платформ является стек MERN, компоненты которого состоят из системы управления базами данных MongoDB, фреймворка Express, библиотеки React и среды выполнения программного кода NodeJS [4, с. 1]. Все компоненты стека поддерживают программирование на JavaScript, что делает процесс разработки более удобным.

MongoDB – это NoSQL или нереляционная база данных с открытым исходным кодом. Использует JSON-подобные документы, полученные в результате обработки данных фреймворком Express. Может хранить большое количество данных и обеспечивать быстрый обмен информацией между клиентом и сервером [4, с. 9-11].

Нереляционные базы данных оптимизированы для платформ, которые должны с минимальной задержкой обрабатывать большой объем данных с разной структурой [3, с. 1]. По сравнению с классическими SQL базами данных, нереляционные имеют ряд преимуществ [3, с. 1-2]:

- масштабируемость: добавление новых узлов в кластер увеличивает общую производительность системы;
- репликация: позволяет копировать обновленные данные на сторонние сервера, что обеспечивает высокую отказоустойчивость и масштабируемость системы;
- возможность хранения больших объемов информации, нет ограничений на типы данных;
- наличие облачных вычислений и хранилищ, что позволяет тестировать приложение на локальном оборудовании, а затем переносить в облако.

Применение базы данных MongoDB упростит хранение, обработку и представление данных об обучающихся при разработке платформы. База данных полностью совместима с NodeJS, что обеспечит наилучшую производительность.

Express – это серверный фреймворк для приложений NodeJS, предоставляющий множество возможностей для веб-разработки. Поддерживает HTTP методы, что делает его API чрезвычайно удобным и простым в использовании. Фреймворк получает JSON-документы и обрабатывает их для дальнейшей отправки в базу данных. Множество дополнительных возможностей позволяют получить улучшенную среду разработки при этом не снижая скорость выполнения программ NodeJS [4, с. 9].

React – это JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов [2, с. 3]. Библиотека позволяет создавать сложные интерфейсы с помощью простых компонентов, которые также возможно повторно использовать в различных частях платформы [4, с. 11-13].

Главной особенностью библиотеки React является обновление содержания веб-приложения без необходимости его перезагрузки. У таких веб-приложений есть термин – одностраничное приложение или SPA (single-page application). Благодаря одностраничным приложениям [5]:

- минимизируется нагрузка на сервер: взаимодействие с сервером происходит только в момент загрузки приложения, что уменьшает количество последующих запросов к серверу;
- улучшается UX (User Experience): нет необходимости в ожидании перезагрузки страницы при взаимодействии с её элементами;
- уменьшается объем передаваемых данных: все данные загружаются при первом посещении страницы и сохраняются за счет кэширования;
- увеличивается скорость и отзывчивость интерфейса.

Применение React библиотеки при разработке платформы не только позволит создать современный, удобный и быстрый пользовательский интерфейс, но и сформирует декларативную структуру кода, которая будет удобочитаемой, упрощенной для отладки и дальнейшей модернизации.

NodeJS – это кроссплатформенная среда выполнения JavaScript с открытым исходным кодом. Содержит встроенную библиотеку, которая позволяет приложениям выступать в качестве веб-сервера, не требуя при этом стороннего программного обеспечения, такого как Apache HTTP Server или Nginx [4, с. 5-6]. Платформа асинхронна и предназначена для создания масштабируемых сетевых приложений. Под асинхронностью понимается неблокирующий ввод-вывод, что позволяет оптимизировать скорость выполнения приложения и минимизировать работу программиста, не требуя отслеживания потоков приложения. Для сравнения приведен рисунок 1, отображающий количество запросов в секунду при повышении уровня параллелизма у NodeJS, работающего с главным потоком (main thread), и Apache, разбивающим выполнение программы на множество потоков [6].

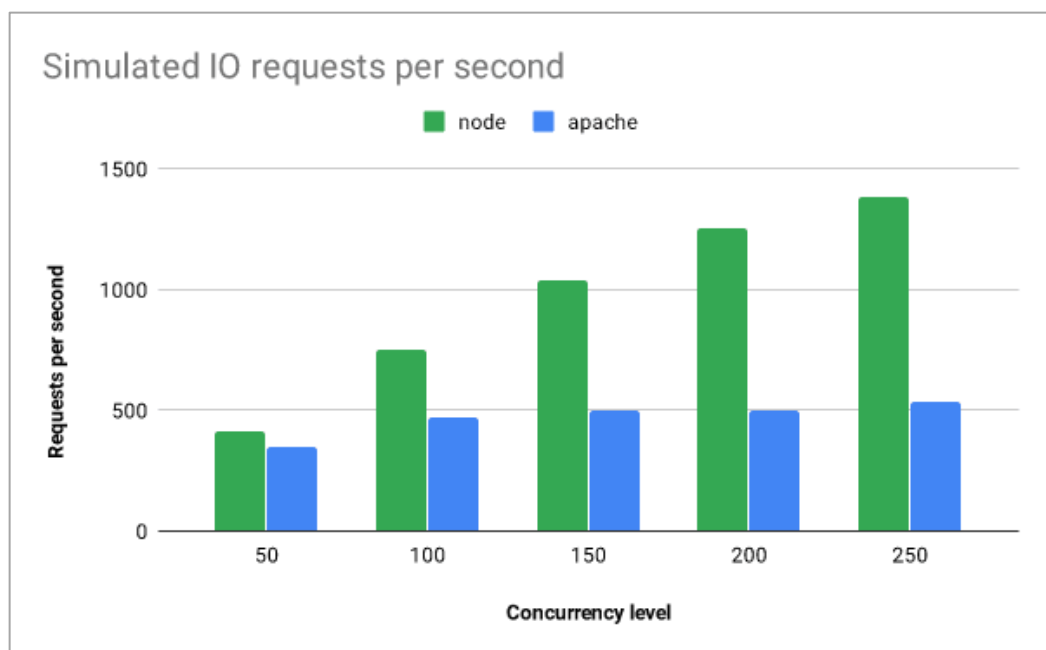


Рисунок 1. Сравнение количества запросов в секунду у платформы NodeJS и Apache при повышении уровня параллелизма

Благодаря асинхронной, неблокирующей модели, среда выполнения NodeJS позволяет выполнять большее число запросов и отлично подходит для управления ими в главном потоке.

При обращении к данным на рисунке 2, можно сделать вывод, что асинхронный подход оказывает существенное влияние на среднее время выполнения запросов [6].

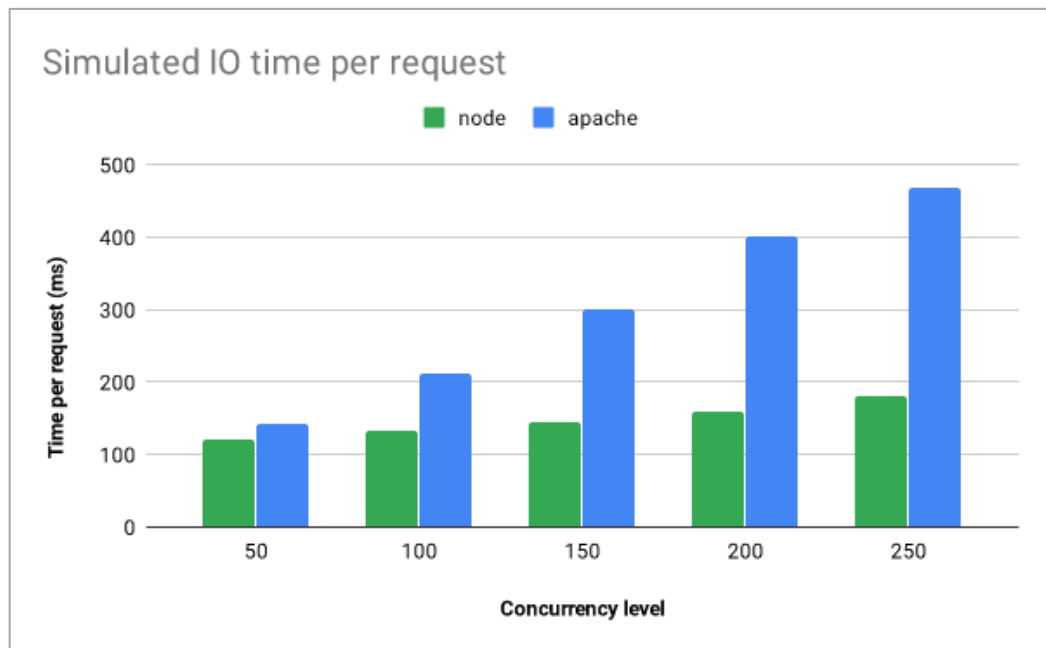


Рисунок 2. Сравнение среднего времени выполнения запросов у платформы NodeJS и Apache при повышении уровня параллелизма

Применение среды выполнения NodeJS позволит увеличить производительность и стабильность работы образовательной платформы при большом объеме серверных запросов.

При разработке веб-приложений, в основе которых лежат технологии стека MERN, задействована MVC архитектура, разделяющая структуру приложения на три компонента: модель (model), представление (view) и контроллер (controller) [1, с. 2].

Компонент модели отвечает за логическую структуру данных приложения, реализация которой в стеке MERN ложится на базу данных MongoDB.

Компонент представления отвечает за визуальную составляющую приложения и реализуется через библиотеку React.

Контроллер связывается с базой данных и получает информацию, необходимую для отправки в компонент представления. Таким образом контроллер взаимодействует как с моделью, так и с представлением, используя технологии NodeJS и Express. На рисунке 3 представлен схематичный процесс взаимодействия стека MERN в MVC архитектуре.

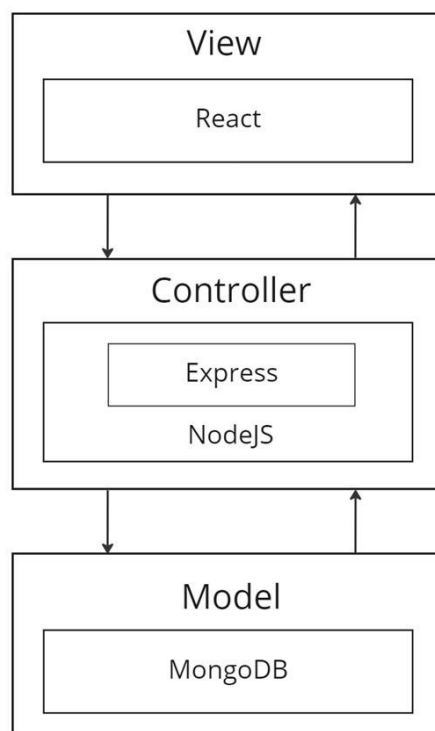


Рисунок 3. Схема взаимодействия MERN технологий в MVC архитектуре

Такой подход позволяет отделить логику от визуального представления, что обеспечивает единую концепцию архитектуры приложения, делает его масштабируемым и упрощает механизм отладки [1, с. 1].

В заключении можно сделать вывод, что стек веб-технологий MERN позволяет создавать полнофункциональные веб-приложения. База данных MongoDB предоставляет возможность работы практически с любой структурой данных, обеспечивает масштабируемость. Программная платформа NodeJS и фреймворк Express позволяют легко и быстро реализовать клиент-серверное взаимодействие на платформе. Библиотека React обеспечивает декларативное представление кода, повторное использование компонентов, стабильность и

скорость пользовательского интерфейса. React реализация одностраничного приложения позволяет обновлять структурные элементы страницы без необходимости её перезагрузки, а применение MVC архитектуры структурирует логику платформы, разделяя ее на компоненты.

Список литературы и источников

1. Бахтин И.В. Главные принципы MVC и смысл использования в разработке программных продуктов // Форум молодых ученых. 2020. №1 (41). – URL: <https://cyberleninka.ru/article/n/glavnye-printsipy-mvc-i-smysl-ispolzovaniya-v-razrabotke-programmnyh-produktov> (дата обращения: 01.03.2023).
2. Давыдовский М.А. Выбор веб-стека для реализации цифровой среды предоставления транспортных услуг // Образовательные ресурсы и технологии. 2019. № 4 (29). – URL: <https://vestnik-muiv.ru/article/vybor-veb-steka-dlya-realizatsii-tsifrovoy-sredy-predostavleniya-transportnykh-uslug/> (дата обращения: 15.01.2023).
3. Шахрай А.В. Преимущества и недостатки нереляционных баз данных [Электронный ресурс]. – URL: https://libeldoc.bsuir.by/bitstream/123456789/39180/1/Shakhray_Preimushchestva.pdf (дата обращения: 23.02.2023).
4. Quang N. M. E-commerce Application using MERN stack [Electronic resource]. – URL: https://www.theseus.fi/bitstream/handle/10024/349838/NhatMai_Thesis.pdf?sequence=2&isAllowed=y (date of treatment: 15.01.2023).
5. Mallick B. What Is a Single-Page Application? Architecture, Benefits, and Challenges. [Electronic resource]. – URL: <https://www.spiceworks.com/tech/devops/articles/what-is-single-page-application/> (date of treatment: 30.02.2023).
6. Spatola E. NodeJS vs Apache performance battle [Electronic resource]. – URL: <https://dev.to/emiliosp/nodejs-vs-apache-performance-battle-for-the-conquest-of-my-5c4n> (date of treatment: 24.02.2023).