

Все расчеты в данной работе были выполнены с использованием процессора Intel® Core™ i7-4770K Processor Turbo Core 4x3,9 GHz и оперативной памяти 16Gb.

### Литература

- 1 Дегтяренко, Н. Н. Описание программных пакетов для квантовых расчетов наносистем / Н. Н. Дегтяренко / М.: МИФИ, 2008. – 180 с.
- 2 Gurin, V. Endofullerenes M@C<sub>60</sub> with defferent monovalent metals / V. Gurin / *NANO*. – 2008. – 03. – p. 483
- 3 Варганов, С. А. Неэмпирические расчеты эндо- и экзоэдральных комплексов фуллерена C<sub>60</sub> с ионом Li<sup>+</sup> и эндоэдрального камплека C<sub>60</sub> с димером Li<sub>2</sub>/ С. А. Варганов, П. В. Аврамов, С. Г. Овчиников/ ФТТ. – 2000. – Т. 42 вып. 2. – С 378–382
- 4 Еняшин, А. Н. Моделирование структуры электронного строения конденсированных фаз фуллеренов C<sub>28</sub> и Zn@C<sub>28</sub>/ А. Н. Еняшин, В. В. Ивановская, Ю. Н. Макурин, А. Л. Ивановский /ФТТ. – 2012. Т.46, № 8 – С. 1522–1525
- 5 Назаров, А. В. Многокомпонентное 3D-проектирование наносистем / А. В. Назаров – М. : Изд-во МГТУ им. Н. Э. Баумана. – 2011. – 392 с.
- 6 Елецкий, А. В. Фуллерены и структура углерода / А. В. Елецкий , Б. М. Смирнов / УФН. – 1995. – Т.165, № 9. – С. 977–1009
- 7 Ибрагимов, И. М. Основы компьютерного моделирования наносистем / И. М. Ибрагимов, А. Н. Ковшов, Ю. Ф. Назаров / Спб.: Издательство «Лань», 2010. – 384 с.

УДК 004.7

*А. С. Пацков, Н. Б. Осипенко*

### ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ И РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ NODE.JS

*В статье описываются проведённые тестирования производительности платформы Node.js с другими серверными технологиями и их результат. По результатам тестирований можно сделать вывод о преимуществах Node.js. Также описана архитектура разработанного Web-приложения с помощью платформы и перечислены компоненты, используемые в разработке. На основе полученных результатов сделан вывод о целесообразности использования Node.js.*

Node.js является серверной технологией, которая основана на разработанном компанией Google JavaScript-движке V8. Это прекрасно масштабируемая система, поддерживающая не программные потоки или отдельные процессы, а асинхронный ввод/вывод, управляемый событиями. Она идеально подходит для Web-приложений, которые не выполняют сложных вычислений, но к которым происходят частые обращения [1].

Для того чтобы наглядно продемонстрировать производительность работы Node.js по сравнению с другими технологиями разработки Web-приложений, проведём несколько различных тестов.

Для тестирования производительности Node.js с Apache сервером проведём два тестирования. Apache HTTP-сервер является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он

позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и другое [2].

Суть первого теста будет состоять в имитации некоторого количества подключений к серверам с помощью утилиты Apache Bench. Результаты тестирования показаны в таблице 1.

Вторым тестом будем вычислять число Пи с помощью ряда Лейбница на формуле (1). Результат второго теста изображен на рисунке 1.

$$\frac{1}{1} - \frac{1}{2} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}. \quad (1)$$

Таблица 1 – Тестирование производительности Web-серверов

Количество подключений /параллельных	Технология	Попытка				
		1	2	3	4	5
10 000/10	Apache	5.45	5.37	5.38	5.25	5.61
	Node.js	2.91	2.70	2.74	2.76	2.74
10 000/100	Apache	5.71	5.66	5.67	5.72	5.54
	Node.js	3.11	2.85	2.93	2.9	2.85
10 000/1 000	Apache	9.16	7.23	7.05	8.36	7.55
	Node.js	3.21	3.21	3.27	3.28	3.15
100 000/1 000	Apache	63	62.43	62.9	64.95	67.5
	Node.js	30.7	31.1	31.3	32.5	31.5

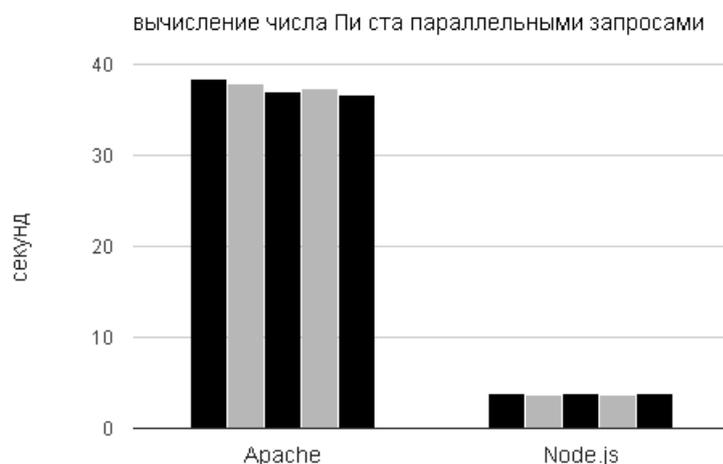


Рисунок 1 – Вычисление числа Пи

В следующем тестировании мы рассмотрим всевозможные комбинации работы технологий PHP и Node.js с базами данных MySQL и MongoDB.

PHP – скриптовый язык программирования общего назначения, интенсивно применяемый для разработки Web-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических Web-сайтов. Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом [3].

MySQL – свободная система управления базами данных (СУБД). MySQL является собственностью компании Oracle Corporation, получившей её вместе с поглощённой Sun Microsystems, осуществляющей разработку и поддержку приложений [4].

MongoDB – это система управления базами данных, которая значительно отличается от MySQL. Основная разница заключается в том, что в MySQL запросы пишутся на языке SQL, а в MongoDB на BSON (бинарный JSON). Это значит, что работа с этой системой может осуществляться в основном через JavaScript выражения [5].

В тестировании будет принимать участие простое приложение, которое делает запрос к базе данных и извлекает случайную запись, в случае MySQL, или документ, в случае MongoDB, из ста тысяч существующих. Результат тестирования представлен на рисунке 2.

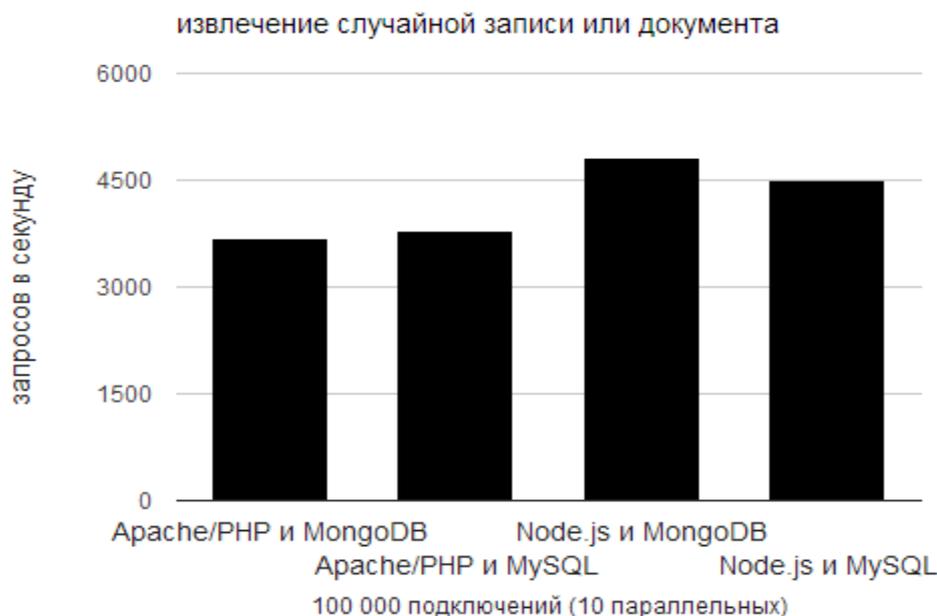


Рисунок 2 – Результат тестирования с базами данных

Современные Web-приложения используют большое количество файлов, получаемых и отдаваемых пользователям. Поэтому быстрота их чтения играет важную роль для общей производительности. Тестирование будет проводиться между технологиями Node.js, Ruby и PHP.

Ruby — интерпретируемый язык программирования высокого уровня. Обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, «сборщиком мусора» и многими другими возможностями, поддерживающими много разных парадигм программирования, прежде всего классово-объектную [6].

Для данного теста на файловой системе размещается большой текстовый файл, в котором содержится около семидесяти семи тысяч строк, общим весом около сорока семи мегабайт. Задача состоит в разбивке текста на слова и подсчёт количества каждого слова в тексте. Результаты теста иллюстрированы на рисунке 3. Быстрее всех оказался Node.js, за ним – Ruby, у PHP в этом тестировании самая низкая производительность.

Разработанное Web-приложение обладает таким функционалом, как регистрация, авторизация, сессии, работа с базой данных, полнодуплексная связь клиента с сервером, одним из примеров которой является сервис мгновенного обмена сообщениями, создание модели, валидация данных, обслуживание статических файлов и другая работа с файловой системой. Схема работы компонентов, задействованных при разработке, отображены на рисунке 4. С помощью фреймворка Express был создан начальный макет приложения. Затем был подключен шаблонизатор для отображения HTML страниц. Следующим

этапом была настройка маршрутизации. Для работы с базой данных MongoDB был подключён и настроен драйвер mongoose и спроектированы модели данных пользователя, сообщений, групп. После этого были реализованы регистрация и авторизация пользователей. Затем были подключены и настроены сессии. Для реализации сервиса мгновенного обмена сообщениями использовались сокеты, которые позволяют без перезагрузки страницы обмениваться сообщениями между клиентом и сервером. Передача аудиофайлов была организована с помощью дополнительных модулей Node.js.

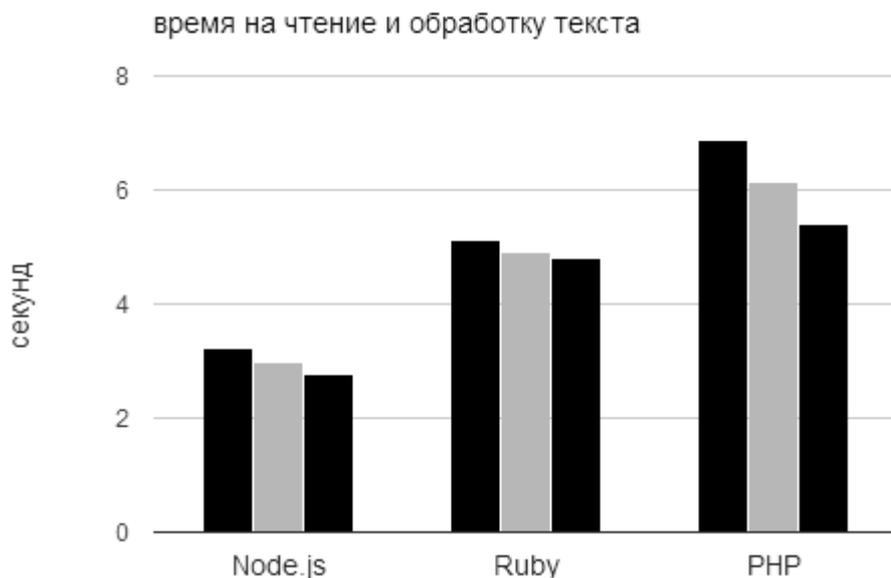
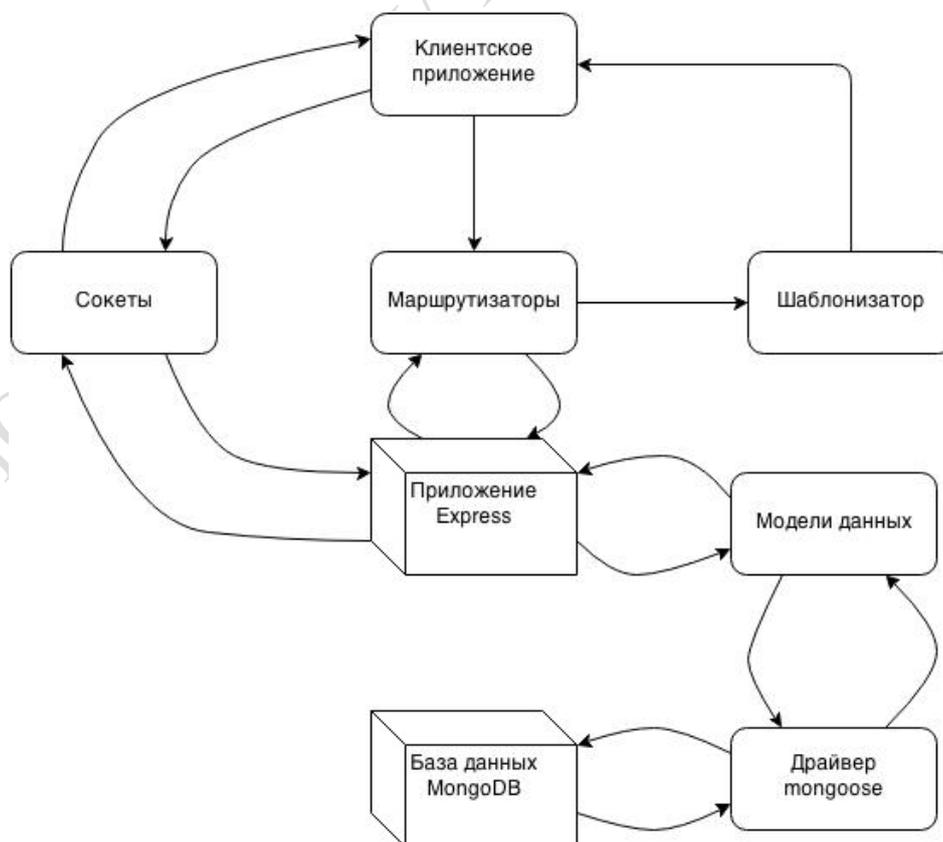


Рисунок 3 – Производительность работы с файловой системой



#### Рисунок 4 – Схема работы компонентов Web-приложения

Таким образом, Node.js на данный момент уже является достаточно сформировавшейся платформой для разработки Web-приложений. Кроме своей производительности технология привлекает разработчиков интересным процессом разработки, а также унификацией языка программирования на всех этапах разработки.

#### Литература

- 1 Пауэрс, Ш., Изучаем Node.js / пер. с англ. – СПб.: Питер, 2013. – 400 с.
- 2 Apache.RU – Документация [Электронный ресурс]. – 1999. – URL: <http://www.apache.ru/docs/> (дата обращения 27.04.2014).
- 3 Коггзолл, Д., PHP 5. Полное руководство / пер. с англ. – СПб.: Питер, 2010. – 752 с.
- 4 Маклафин, Б., PHP и MySQL – Исчерпывающее руководство / пер. с англ. – СПб.: Питер, 2013. – 508 с.
- 5 MongoDB документы [Электронный ресурс]. – 2011. – URL: <http://docs.mongodb.org/manual/> (дата обращения 27.04.2014).
- 6 Википедия – свободная энциклопедия [Электронный ресурс]. – 2001. – URL: <http://wikipedia.org> (дата обращения 27.04.2014).

УДК 539.126

*Н. С. Потинко*

#### РАСЧЕТ СПЕКТРА МАСС КВАРКОНИЯ В НЕРЕЛЯТИВИСТСКИХ МОДЕЛЯХ

*Данная работа посвящена численному расчету спектра масс кваркония. Для этих целей решалось радиальное уравнение Шредингера с запирающим потенциалом. Решение реализовывалось с помощью спектрального метода в среде программирования Mathcad. Полученные результаты были проанализированы, построены графики решений.*

В физике элементарных частиц мезоны рассматривают как связанное состояние системы кварк - антикварк. В нерелятивистском приближении для описания данной системы используется радиальное уравнение Шредингера [1,2]

$$-\frac{\hbar^2}{2m} \frac{d^2 R(r)}{dr^2} + U(r)R(r) = ER(r) \quad (1)$$

где  $R(r) = r\psi(r)$ ,  $\hbar = 1$ ,  $2m = 1$ ,  $l = 0$ .

Для решения уравнения (1) используем спектральный метод [3,4]. В данном методе волновая функция представляется в виде ряда Фурье:

$$R(r) = \sum_{k=0}^{\infty} A_k g_k(r) \quad (2)$$

по базисным функциям на участке  $0 \leq r \leq r_{\max}$ .

$$g_k(r) = \sqrt{\frac{2}{r_{\max}}} \sin\left(\frac{k\pi r}{r_{\max}}\right). \quad (3)$$

Важным свойством базисных функций является их полнота и ортогональность: