

# Opracowanie i testowanie aplikacji utworzonych z użyciem platform Node.js oraz Laravel

Vasyl Bomba\*, Edyta Łukasik

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

**Streszczenie.** Celem pracy była analiza i badanie programowej platformy Node.js i frameworka Laravel, a także porównanie wydajności serwisów utworzonych za pomocą tych technologii. Testy wydajności aplikacji przeprowadzono za pomocą oprogramowania Apache Benchmark i Apache JMeter. Analiza obejmuje teoretyczne i praktyczne aspekty ich funkcjonowania.

**Słowa kluczowe:** Node; Node.js; Laravel; PHP; MongoDB; benchmark

\*Autor do korespondencji.

Adres e-mail: bombavasyl@gmail.com

## Development and testing of applications created with the platforms Node.js and Laravel

Vasyl Bomba\*, Edyta Łukasik

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

**Abstract.** The aim of this work was the analysis and research of software platform Node.js and the framework Laravel, and comparing the performance of the created sites using Node.js and PHP. The tests were carried out with Apache Benchmark and Apache JMeter. The analysis includes theoretical and practical aspects of their functioning.

**Keywords:** Node; Node.js; Laravel; PHP; MongoDB; benchmark.

\*Corresponding author.

E-mail address: bombavasyl@gmail.com

### 1. Wstęp

Celem artykułu jest analiza i badanie platformy programowej Node.js i frameworka Laravel oraz porównanie wydajności serwisów utworzonych za pomocą tych narzędzi. Analiza obejmuje teoretyczne aspekty ich funkcjonowania, a także badania wydajności aplikacji.

Do tworzenia i testowania aplikacji internetowych używane są następujące narzędzia:

- platforma programowa Node.js;
- framework Laravel;
- serwer MySQL;
- MongoDB;
- Bootstrap;
- Apache Benchmark;
- Apache JMeter.

### 2. Zastosowane technologie

Node (lub Node.js) – to nowoczesna platforma, która wyświetla język JavaScript poza przeglądarką i pozwala na stosowanie go w aplikacjach serwerowych. Platforma jest oparta o wyjątkowo szybki silnik JavaScript, zapożyczony z przeglądarki Chrome, V8, do której dodana została szybka i niezawodna biblioteka asynchronicznego sieciowego wejścia/wyjścia. Główny nacisk w Node położony jest na

tworzenie wysokiej jakości, dobrze skalowalnych klienckich i serwerowych aplikacji webowych [1].

Node.js działa na zasadzie asynchronicznego modelu, który zbudowany jest jak kolejka zdarzeń (event loop). W przypadku wystąpienia jakiegoś zdarzenia (przyjście żądania, odczyt pliku, odpowiedź na zapytanie z bazy danych), zostanie ono umieszczone na końcu kolejki. Strumień, który obsługuje tę kolejkę, bierze zdarzenie z początku kolejki i wykonuje związany z tym wydarzeniem kod. Dopóki kolejka nie jest pusta procesor jest zajęty pracą [2].

Laravel jest to darmowy framework, przeznaczony dla tworzenia projektów z wykorzystaniem architektury modelu MVC (ang. Model View Controller) — model-widok-kontroler).

Personal HyperText Processor (PHP) jest to skryptowy język programowania, używany po stronie serwera do dynamicznego generowania stron HTML. PHP różni się od JavaScript tym, że skrypty PHP są wykonywane na serwerze i generują kod HTML, który jest wysyłany do klienta. Zawiera on wszystkie funkcje potrzebne do pracy po stronie serwera. PHP to jeden z niewielu języków programowania, stworzonych specjalnie do wytwarzania aplikacji internetowych. Użycie języka PHP wraz ze środowiskiem Apache jako serwerem WWW oraz bazą danych MySQL określa się nazwą platforma AMP. Są to elementy niezbędne do stworzenia funkcjonalnej aplikacji internetowej[4].

W wyniku ankiety sitepoint.com w grudniu 2013 roku o najbardziej popularnych frameworkach PHP Laravel zajęł miejsce najbardziej obiecującego projektu na 2014 rok.

W 2015 roku w wyniku ankiety sitepoint.com dotyczącej wykorzystania frameworków PHP wśród programistów, PHP zajęł pierwsze miejsce w kategoriach:

- Framework klasy korporacyjnej;
- Framework dla projektów osobistych [3].

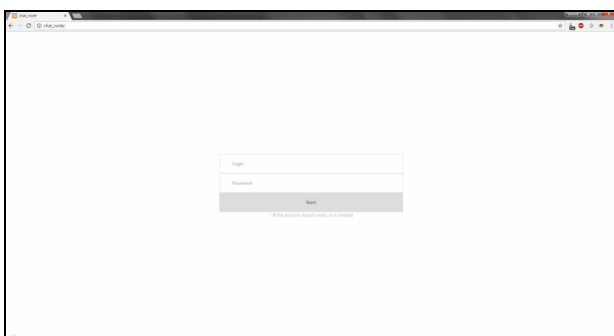
### 3. Opis aplikacji

Zostały utworzone dwie aplikacje realizujące wymianę informacji tekstowej pomiędzy osobami. Jedna z nich używa Node.js, a druga frameworka Laravel i języka programowania skryptowego PHP. Mają one bardzo podobną funkcjonalność:

- zapisywanie wiadomości w bazie danych;
- autoryzacja użytkownika;
- przesyłanie wiadomości;
- powiadomienie podłączonych klientów, gdy ktoś wyszedł z czatu lub wszedł na czat;
- lista użytkowników online czatu;
- połączenie z serwerem za pomocą WebSocket.

Każda z nich dostosowuje się wyglądem strony do rozmiaru okna przeglądarki, w której jest uruchamiana.

Aby wejść na czat konieczne jest przejście przez narzędzie autoryzacji. Zrzut ekranu tego okna pokazano na rys. 1. Należy w nim wypełnić 2 pola: login i hasło. W przypadku, gdy użytkownik pozostawi jedno z pól puste, pojawia się komunikat o błędzie. Jeśli użytkownika o podanym loginie nie istnieje w bazie danych, zostanie on automatycznie rejestrowany w niej z wprowadzonymi danymi.



Rys. 1. Autoryzacja na czacie Node.js

Po tym, jak użytkownik pomyślnie przeszedł autoryzację pojawia się okno czatu pokazane na rys. 2.

W oknie czatu po lewej stronie dostępny jest spis otrzymanych wiadomości, które uporządkowane są chronologicznie według pojawiania się. Po stronie prawej okna znajduje się lista aktywnych w danym momencie czasu użytkowników. Oprócz tego wyświetlana jest informacja z licznikiem użytkowników zalogowanych do aplikacji w danej chwili. W dolnej części okna aplikacji jest miejsce na wpisanie informacji tekstowej, która ma być przesłana. Wysłanie odbywa się po wciśnięciu klawisza „enter”.

W prawym dolnym rogu znajduje się przycisk umożliwiający zamknięcie czatu.



Rys. 2. Główne okno czatu Node.js

W stworzonych aplikacjach pojawia się dźwięk informujący o otrzymaniu wiadomości.

### 4. Badania wydajności

Głównym zadaniem każdego testowania wydajności witryny jest poznanie jego odporności na obciążenia, które mogą pojawiać się nie tylko z powodu dużej ilości użytkowników podłączonych jednocześnie, ale i być konsekwencją nieprawidłowej konfiguracji serwera, nieprawidłowego działania skryptów lub ataków typu DOS.

Do testów został użyty komputer posiadający:

- Procesor: Intel Core i5-3210M @ 3.10 GHz (4 Cores);
- RAM: 8 GB.

Aby przeprowadzić testy z taką dużą ilością użytkowników należy zmienić ustawienia w systemie operacyjnym Ubuntu:

- zwiększenie limitu deskryptorów plików. Zmiana wartości opcji `nofile` w pliku `limits.conf`, co zostało pokazane na listingu 1;
- zmiana wartości parametru `net.ipv4.netfilter.ip_conntrack_max` w pliku `sysctl.conf`, co pokazano na listingu 2;
- zmiany ograniczeń systemowych na otwarte pliki - poprzez edycję parametru `ulimit -n`.

Listing 1. Zwiększenie limitu deskryptorów plików w pliku `limits.conf`

```
#/etc/security/limits.conf  
nofile 1048576
```

Listing 2. Zwiększenie limitu deskryptorów plików w pliku `sysctl.conf`

```
#/etc/sysctl.conf  
net.ipv4.netfilter.ip_conntrack_max = 1048576
```

Apache Benchmark jest to jednowątkowa aplikacja wywoływana z wiersza poleceń służąca do pomiaru wydajności HTTP serwerów WWW. Pierwotnie opracowany został on do testowania serwera HTTP Apache, ale w zasadzie nadaje się do testowania dowolnego serwera WWW. Narzędzie to jest dostarczane ze standardową dystrybucją

Apache i jak sam Apache, jest darmowe i rozpowszechniane na Licencji Apache [5].

Do zbierania informacji o ustawieniach serwera takich jak CPU i RAM, był zainstalowany moduł node-millennium.

Serwery były testowane przez 20 sekund dla różnej liczby HTTP GET żądań. Żądania te były generowane poprzez Apache Benchmark. Liczba użytkowników oraz czas testu były podawane jako parametry wejściowe dla przeprowadzanych testów. Wyniki testów z żadaniami przedstawiono w tabeli 1 i 2.

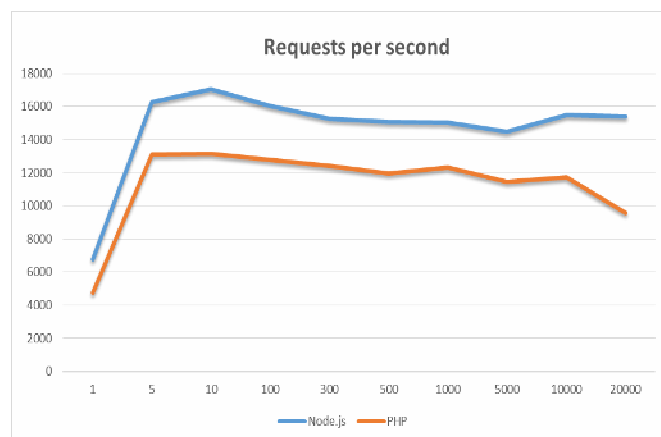
Tabela 1. Testy aplikacji napisanej z użyciem Node.js

Liczba użytkowników	Liczba żądań	Liczba żądań / sekunda	Średnie CPU [%]	RAM [MB]
1	135037	6751.85	27	24
5	325700	16284.9	84.6	29
10	340912	17045.6	91.7	37
100	321004	16049.2	92.4	59
300	305202	15256.9	91.4	91
500	301954	15093.9	90.3	105
1000	300325	15012.9	90.1	107
5000	289842	14489.4	78.9	168
10000	310211	15508.9	81.5	215
20000	308017	15399.7	86	232

Tabela 2. Testy aplikacji napisanej z użyciem PHP i Laravel

Liczba użytkowników	Liczba żądań	Liczba żądań / sekunda	Średnie CPU [%]	RAM [MB]
1	94678	4733.9	33	26
5	261759	13087.9	95.7	31
10	263042	13152.1	97.7	80
100	255957	12797.2	97.6	86
300	248093	12404.1	94.6	140
500	239068	11952	93.3	151
1000	246103	12304.8	95.4	169
5000	228845	11441	92	194
10000	234412	11719.3	93.7	299
20000	191469	9573.42	92	330

Na podstawie przeprowadzonych badań zostały zbudowane porównawcze wykresy przedstawione na rys. 3. Na podstawie wyników można stwierdzić, że bardziej wydajny okazał się Node.js w zakresie zużywanych zasobów procesora oraz pamięci. Liczba obsłużonych żądań na sekundę dla każdej liczby użytkowników była większa dla aplikacji napisanej z użyciem Node.js. Dla liczby użytkowników od 1 do 5 ta różnica była niewielka, natomiast dla zakresu od 5 do 10000 wynosiła ona nie mniej niż 4000. Po przekroczeniu liczby 10000 użytkowników różnica ta zaczęła się powiększać, a dla 20000 użytkowników osiągnęła 6000.



Rys. 3. Liczba żądań dla HTTP GET żądań Node.js i PHP

Program Apache JMeter jest to narzędzie do przeprowadzenia testowania obciążenia, stworzone przez Apache Software Foundation. Choć początkowo program JMeter został zaprojektowany jako narzędzie do testowania aplikacji WWW, obecnie jest on w stanie przeprowadzić testy dla JDBC połączeń, FTP, LDAP, SOAP, JMS, POP3, IMAP, HTTP i TCP [6].

Celem skorzystania z tego oprogramowania było zasymulowanie wysyłania wiadomości do wielu użytkowników jednocześnie. W ten sposób można było zbadać działanie aplikacji z bazą danych.

Ponieważ obciążenie serwera zapytaniami i równoległe przetwarzanie ich wyników programem Apache JMeter wymagała znacznych zasobów systemu, do tego celu użyto komputera o parametrach:

- CPU: Intel Core i7-4790K @4.00 GHz;
- RAM: 16 GB.

Wiadomości wysłane za pośrednictwem gniazd miały przekazywane parametry ustawione odpowiednio na: "type": "message" i "message": "test message". Czynności wykonywane były jednocześnie dla różnej ilości wirtualnych użytkowników w ciągu 20 sekund. Takie warunki maksymalnie zbliżyły testy do rzeczywistych warunków pracy aplikacji realizujących czaty.

Wyniki testów dla generowanych automatycznie wiadomości wysyłanych do wszystkich podłączonych użytkowników czatu przedstawiono w tabeli 3 i 4. Zarówno

liczba użytkowników jak i liczba wysyłanych wiadomości były ustawiane w programie Apache JMeter.

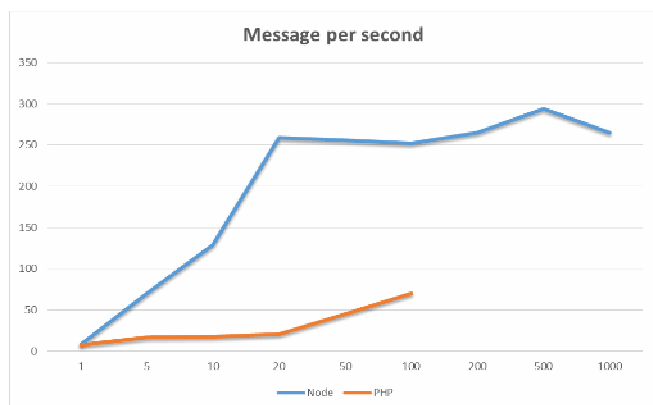
Tabela 3. Test wysyłania wiadomości - aplikacja z użyciem Node.js

Liczba użytkowników	Liczba wiadomości	Liczba żądań / sekunda	Błąd [%]	Średnie CPU [%]
1	184	9.2	0	11
5	1406	70.3	0.4	21
10	2586	129.3	1.7	26
20	5185	259.3	14.3	40
50	5137	256.9	23	42
100	5043	252.2	25.3	50
200	5301	265	24.7	58
500	5876	293.8	23.66	60
1000	5301	265	25.37	81

Tabela 4. Test wysyłania wiadomości - aplikacja z użyciem Lavarem

Liczba użytkowników	Liczba wiadomości	Liczba żądań / sekunda	Błąd [%]	Średnie CPU [%]
1	161	8.1	0	29
5	340	17	12.6	41
10	362	18.1	19	45
20	429	21.5	27	51
50	904	45.2	35.8	61
100	1399	70	46	63

Na podstawie wyników przeprowadzonych badań zostały zbudowane porównawcze wykresy, przedstawione na rys. 4.



Rys. 4. Liczba żądań na sekundę dla aplikacji w Node.js i PHP

Wynik testu pokazał, że aplikacja realizująca czat napisana z użyciem Node.js radzi sobie z wiadomościami kilka razy

szybciej niż aplikacja napisana w wykorzystaniu frameworka Laravel i języka PHP. Przy ilości użytkowników powyżej 100, serwer aplikacji, napisany przy użyciu PHP i Laravel, nie radzi sobie z takim obciążeniem. Serwer Node.js wytrzymuje obciążenie ponad 1000 równoległych strumieni.

W każdym z przeprowadzonych testów widoczna jest różnica w wydajnościach stworzonych aplikacji. W testach pustymi zapytaniami różnica ta wynosiła ok. 25%, natomiast w testowaniu wysyłania wiadomości znaczną przewagę miała aplikacja stworzona z pomocą Node.js. Była ona około 4 razy szybsza.

## 5. Wnioski

Przeprowadzone zostały badania dotyczące wydajności dwóch bardzo zbliżonych funkcjonalnie aplikacji stworzonych w życiu Node.js oraz frameworka Laravel i PHP. Z pomocą obu technologii utworzono 2 czaty. Wielowątkowy czat na frameworku Laravel z bazą danych MySQL oraz asynchroniczny czat na platformie Node.js z bazą danych MongoDB.

Przeprowadzona analiza teoretyczna związana z funkcjonowaniem badanych technologii programistycznych wskazała podobieństwa oraz różnice w działaniu obu szkieletów projektowych. Omawiając równocześnie mocne i słabe strony każdej z technologii w sytuacjach typowych dla działania aplikacji internetowych, w analizie zwrócono uwagę na najważniejsze aspekty związane z działaniem badanych technologii.

Dla obu aplikacji przeprowadzono badania za pomocą specjalnych programów: Apache Benchmark (test pustymi zapytaniami) i Apache JMeter (test wysyłania wiadomości). W każdym z testów lepiej wypadła aplikacja stworzona z użyciem Node.js.

Przedstawione porównanie powinno ułatwić programiście dokonanie wyboru między dwoma alternatywnymi rozwiązaniami.

## Literatura

- [1] Пауэрс III.: Изучаем Node.js, O'REILLY, 2012
- [2] <https://habrahabr.ru/post/150788/> [1.9.2016]
- [3] <https://laravel.ru/> [8.9.2016]
- [4] <http://php.net/manual/ru/intro-what-is.php> [12.9.2016]
- [5] <http://httpd.apache.org/docs/current/programs/ab.html> [18.9.2016]
- [6] <http://www.sitepoint.com/sitepoint-smackdown-php-vs-node-js/> [21.9.2016]