

Performance analysis of Laravel and Yii2 frameworks based on the MVC architectural pattern and PHP language

Analiza wydajności szkieletów programistycznych Laravel oraz Yii2 opartych na wzorcu architektonicznym MVC oraz języku PHP

Konrad Sławomir Węgrzecki*, Mariusz Dzieńkowski

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

Abstract

The subject of this paper is the performance analysis of two PHP programming frameworks in the latest versions - Laravel 9.6 and Yii 2.0.45. It was carried out with the help of test applications prepared for this purpose, which have identical functionalities: they generate all prime numbers from a given range and create a book ranking system based on CRUD operations. The request handling time of each application was used as a comparison criterion. To check the performance, dedicated debugbars - bottom bars displaying information about the executed request - were used. The results obtained after the tests showed that in terms of performance, Laravel is a better technology than Yii for building web applications.

Keywords: Laravel; Yii2; programming frameworks; performance comparative analysis

Streszczenie

Przedmiotem pracy jest analiza wydajnościowa dwóch szkieletów programistycznych języka PHP w najnowszych wersjach – Laravel 9.6 i Yii 2.0.45. Została ona przeprowadzona przy pomocy przygotowanych do tego celu aplikacji testowych, posiadających identyczne funkcjonalności. Aplikacje generują wszystkie liczby pierwsze z podanego zakresu oraz tworzą system rankingowy książek, oparty na operacjach CRUD. Jako kryterium porównawcze przyjęto czas obsługi żądań przez każdą aplikację. Do sprawdzenia wydajności użyto dedykowanych szkieletom tzw. debugbarów – dolnych pasków wyświetlających informację o wykonanym żądaniu. Wyniki uzyskane po przeprowadzonych badaniach wykazały, że biorąc pod uwagę wydajność, Laravel jest lepszą technologią niż Yii do budowy aplikacji internetowych.

Słowa kluczowe: Laravel; Yii2; szkielety programistyczne; analiza wydajności

*Corresponding author

Email address: konrad.wegrzecki@pollub.edu.pl (K. S. Węgrzecki)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Obecnie z oprogramowania można korzystać używając Internetu, albo posiadając zainstalowaną aplikację na urządzeniu mobilnym lub komputerze. Ostatnia opcja jest w tej chwili najrzadziej wybierana ze względu na czasochłonny proces instalacji oraz utrudnione przeniesienie danych. W związku z tym, najpopularniejszym rozwiązaniem stały się dziś aplikacje internetowe, ponieważ są one bezpieczne - nie przechowują danych na lokalnym urządzeniu oraz użytkownik nie musi ich samodzielnie instalować i modernizować.

Na rynku dostępnych jest bardzo dużo technologii umożliwiających tworzenie aplikacji internetowych. Wśród języków programowania dominującą pozycję na rynku w tym segmencie zajmują języki JavaScript, Python, Java, C# oraz PHP [1]. Na bazie języka PHP powstała największa liczba różnorodnych szkieletów programistycznych, które można zakwalifikować do jednej z trzech podgrup. Do pierwszej należą szkielety programistyczne posiadające rozbudowany zestaw narzędzi wspomagający tworzenie aplikacji oraz automatyzujący proces implementacji funkcjonalności. Drugą są szkielety, których pakiet narzędzi jest mniej rozbudowany, a programista ma większy udział w procesie

tworzenia oprogramowania. Trzecią grupę stanowią systemy zarządzania treścią CMS (ang. Control Management System). Szkielety z tej grupy różnią się od szkieletów z dwóch pierwszych grup tym, że tworzenie aplikacji odbywa się głównie za pomocą wtyczek, a nie pisania kodu. Szkielety programistyczne języka PHP wchodzące w skład dwóch pierwszych grup wykorzystują wzorce architektoniczne, dzięki którym pisanie kodu przebiega szybciej, a proces tworzenia aplikacji jest tańszy. Obecnie najpopularniejszym wzorcem, ze względu na swoją prostotę oraz szybkość działania, jest MVC (ang. Model View Controller).

W ramach niniejszej pracy została przeprowadzona analiza porównawcza dwóch technologii do budowy aplikacji webowych opartych na języku PHP. Do porównań wybrano Laravla - najpopularniejszy szkielet programistyczny należący do pierwszej oraz Yii - najpopularniejszy szkielet należący do drugiej grupy [2].

2. Przegląd literatury

Duża liczba technologii pozwalających na tworzenie aplikacji po stronie serwera doprowadza do częstych dyskusji i rozważań na temat jakości oraz użyteczności technologii. Spowodowało to powstanie wielu publika-

cji naukowych poruszających tematykę tworzenia aplikacji internetowych. Szkielety programistyczne Laravel i Yii były już analizowane w pracy *Comparison Between Yii Frameworks and Laravel in 3 Different Version for Viewing Large Data of Shipyard Industry in Indonesia*. Na potrzeby publikacji zostały wykonane aplikacje testowe bazujące na tych dwóch szkieletach programistycznych i oparte na dwóch wersjach języka PHP [3]. W badaniach pod uwagę wzięto kilka miar: czas wykonania żądania, czas reakcji serwera, rozmiar wykorzystywanej pamięci oraz liczbę przetwarzanych rekordów przez aplikację testową w ciągu jednej sekundy. Wyniki badań pokazały, że użycie wersji 7 języka PHP poprawia wydajność nawet dwukrotnie w przypadku obu szkieletów programistycznych. Poza tym okazało się, że wydajniejszym szkieletem o około 7% jest Laravel.

Publikacja *Selecting a PHP framework for a web application project-The method and case study* podejmuje tematykę analizy sześciu szkieletów programistycznych opartych na języku PHP: Laravel, Symfony, CodeIgniter, Yii2, Zend, CakePHP [4]. Autorzy pracy w swoich badaniach podeszli do tematu szeroko, zaczynając od badań popularności szkieletów, poprzez zidentyfikowanie wspieranych przez nie technologii, na testach opracowanych aplikacji kończąc. Do oceny szkieletów wykorzystano system punktowy w skali od 0 do 1. Oceny dokonano w dziewięciu kategoriach. Wyniki pozwoliły na wskazanie najbardziej uniwersalnego rozwiązania, którym okazał się zdaniem autorów Yii2. Szkielet ten uzyskał 76 procent maksymalnych punktów, dzięki czemu wypadł znacznie lepiej niż Laravel, który zdobył tylko 31 procent.

Ostatnią pracą dotyczącą analizy szkieletów programistycznych Yii2 i Laravel jest *Porównanie możliwości tworzenia aplikacji PHP na przykładzie Yii2 i Laravel*. Na potrzeby badań zostały przygotowane aplikacje, które następnie analizowano pod względem ich struktury, łatwości i możliwości tworzenia interfejsu graficznego oraz współpracy z systemami bazodanowymi [5]. Oba szkielety były oceniane w ośmiu kategoriach w skali punktowej od 1 do 5. Końcowy wynik aplikacji opracowanej w Laravelu to 33 punkty, natomiast Yii2 - 31 punktów. Według autorki szkielet programistyczny Laravel jest bardziej zautomatyzowany i to on powinien być pierwszym wyborem dla mniej doświadczonych programistów, choć Yii2 jest jego silnym konkurentem.

Wyniki badań omówionych prac [3-5] nie pozwalają na jednoznaczne wskazanie technologii lepszej lub takiej która jest bardziej uniwersalna. Podczas zrealizowanych porównań były brane pod uwagę: wymagania technologiczne, stopień trudności tworzenia oprogramowania, obsługa systemów bazodanowych, szybkość działania aplikacji, wsparcie technologii poprzez dokumentację, popularność, użyta architektura oraz organizacja kodu. Głównym wnioskiem płynącym z tych prac jest to, iż wybór szkieletu programistycznego powinien być uzależniony od przeznaczenia tworzonej aplikacji. Laravel jest zdecydowanie prostszym szkieletem do

rozpoczęcia programowania, ponieważ struktura kodu jest bardziej zwięzła, a cały proces bardziej zautomatyzowany. Yii2 stanowi jednak poważną konkurencję jako szkielet posiadający największą liczbę wspieranych technologii.

W publikacjach [6-7] większy nacisk został położony na stronę koncepcyjną tworzenia aplikacji, czyli zastosowane rozwiązania w danym szkielecie, strukturę kodu, możliwości rozszerzania aplikacji oraz ich wsparcie poprzez dokumentację czy specjalistyczne fora dyskusyjne. Szczególną uwagę zwrócono na wsparcie dla innych technologii, takich jak systemy bazodanowe czy system szablonów szaty graficznej oraz zastosowane ułatwienia dla programisty. W zależności od pracy wyniki końcowe różnią się od siebie, głównie ze względu na przyjęty system oceniania. Powiązane jest to z tym, że część elementów była oceniana subiektywnie przez autora danej pracy.

Autorzy publikacji [8-10] porównali szkielety programistyczne na podstawie aplikacji testowych używając przyjętych przez siebie kryteriów. Dotyczyły one różnic w budowie oraz wydajności poszczególnych szkieletów programistycznych. Do głównych miar użytych w tych porównaniach należały czasy reakcji serwera, czasy wykonania zadania zleconego użytkownikom oraz czasy odpowiedzi serwera. Dodatkowo badania te często były wykonywane pod różnym wirtualnym obciążeniu aplikacji testowych, w celu sprawdzenia jej pod kątem przyszłego użytkownika. Badania prowadzone były na aplikacjach wykorzystujących bazę danych. Zazwyczaj weryfikowany był odczyt wielu rekordów oraz zapis, edycja i usuwanie pojedynczego rekordu. Testy takich aplikacji pozwoliły zbadać szkielet programistyczny w środowisku pracy przypominającym funkcjonowanie przeciętnej aplikacji internetowej.

Porównywanie technologii wytwarzania aplikacji internetowych stało się ostatnio dosyć popularne. Autorzy prac w swoich badaniach poruszają kwestie technologiczne i wydajnościowe, celem wybrania najbardziej uniwersalnego rozwiązania. Działania te mają pomóc programistom w wyborze technologii zanim przystąpią do budowy systemu. W ramach niniejszej pracy autor także zajął się problemem analizy narzędzi do tworzenia aplikacji internetowych, poprzez utworzenie, a następnie analizę działania aplikacji testowych. Tym co odróżnia zrealizowane badania od badań wykonanych przez innych autorów są użyte wersje szkieletów programistycznych, użyta wersja języka PHP oraz przygotowane aplikacje testowe zawierające niezbędne funkcjonalności, na których podstawie będzie możliwa pełna ocena porównywanych technologii.

3. Cel i hipoteza pracy

Celem pracy jest analiza porównawcza wydajności szkieletów programistycznych w ich najnowszych wersjach w momencie pisania pracy czyli Laravel 9.6 i Yii 2.0.45. Za główną miarę wydajności przyjęto średni czas realizacji zadania przez aplikację opracowaną za pomocą danego szkieletu.

Za hipotezę badawczą przyjęto stwierdzenie:

Szkielet programistyczny Laravel może być lepszym rozwiązaniem niż Yii, ze względu na jego większą wydajność oraz bardziej zautomatyzowany proces tworzenia aplikacji.

4. Metodologia badań

4.1. Aplikacje testowe

Do przeprowadzenia badań wymagane było przygotowanie dwóch aplikacji testowych posiadających identyczny zestaw funkcjonalności - jedna aplikacja wykonana była przy pomocy szkieletu Laravel oraz druga przy pomocy Yii2. Obie aplikacje korzystały z tej samej bazy danych oraz wykorzystywały wyłącznie język znaczników HTML do budowy widoków. Porównywane aplikacje testowe posiadają dwa oddzielne moduły. Pierwszy z nich służy do wyznaczania liczb pierwszych z podanego przez użytkownika zakresu. Do tego celu nie został wykorzystany żaden optymalny algorytm, a jedynie podwójna, zagnieżdżona pętla z przerwaniem, w momencie gdy dzielnik liczby zostanie znaleziony. Ten prosty, naiwny algorytm realizuje maksymalną liczbę przejść i powoduje maksymalne obciążenie. Drugim modułem jest system rankingowy książek, oparty na operacjach CRUD, realizowanych na bazie danych oraz zawierający system uwierzytelniania użytkowników z podziałem na role użytkownika oraz administratora. Moduł ten został pozbawiony atrakcyjnej warstwy wizualnej, bowiem zrealizowano ją za pomocą podstawowych znaczników HTML, które umożliwiły także interakcję z użytkownikiem.

Wymagania funkcjonalne aplikacji testowych

- Zwykły użytkownik:
 - tworzenie konta,
 - logowanie w systemie,
 - wylogowanie,
 - przeglądanie listy książek,
 - przeglądanie rankingu książek,
 - przeglądanie szczegółów książki,
 - ocena książki.
- Administrator – oprócz funkcji użytkownika może również:
 - dodawać książki,
 - edytować książki,
 - usuwać książki.

Struktura bazy danych

Baza danych opracowana na potrzeby aplikacji testowych składa się z trzech tabel przechowujących wszystkie niezbędne dane do przeprowadzenia testów. W skład bazy danych wchodziły następujące tabele:

- *users* – przechowująca dane kont użytkowników,
- *books* – zawierająca szczegółowe dane książek,
- *rating* – zawierająca oceny książek przez użytkowników systemu.

Za wyjątkiem danych logowania wszystkie dane w bazie danych zostały wygenerowane losowo według określonego klucza.

- Tabela *books*:

- tytuł książki – ciąg losowy (15 znaków),
- imię i nazwisko autora – generowane funkcją *faker*,
- rok wydania – liczba losowa z przedziału od 1980 do 2022,
- opis – losowy ciąg o długości od 150 do 250 znaków.
- Tabela *ratings*:
 - identyfikator użytkownika – liczba losowa z przedziału od 1 do 100,
 - identyfikator książki – liczba losowa z przedziału od 1 do największego identyfikatora książki,
 - ocena książki – liczba losowa z zakresu od 1 do 10.

4.2. Scenariusze testowe

Analizę porównawczą wydajności szkieletów programistycznych Laravel i Yii2 przeprowadzono za pomocą scenariuszy testowych, podczas których dokonano pomiarów czasów potrzebnych do:

- wyznaczenia liczb pierwszych do wartości: 100, 500, 1000, 5000, 10000, 50000, 100000,
- wyświetlenia listy książek zawierającej 100, 500, 1000, 5000, 10000 pozycji,
- wyświetlenia rankingu książek zawierającego: 100, 200, 500, 1000, 2000 książek oraz odpowiednio: 500, 1000, 2500, 5000, 10000 ocen,
- wyświetlenia, dodania, edycji i usunięcia książki znajdującej się na pozycji: 100, 500, 1000, 5000, 10000 w bazie danych,
- dodania oceny książki przy wypełnieniu tabeli ocen liczbą rekordów: 500, 1000, 2500, 5000, 10000.

Podczas eksperymentu czas liczony był od momentu otrzymania żądania przez daną aplikację do zakończenia pracy i wysłania odpowiedzi do przeglądarki. W celu zobiektywizowania wyników, pomiary były wykonywane na dwóch urządzeniach, na których przeprowadzono po 5 prób każdego z testów, a następnie wyniki uśredniono dla każdego urządzenia.

Wydajność rozwiązania nie jest jedynym aspektem brany pod uwagę podczas wyboru technologii. Dodatkowymi, istotnymi kwestiami są:

- możliwości technologii – zastosowane rozwiązania, wspierane technologie zewnętrzne (np. systemy bazodanowe) oraz usprawnienia dla programisty,
- jakość dokumentacji technicznej,
- wsparcie społeczności – zrzeszanej na forach tematycznych np. Stack Exchange i ogólnych np. grupy na Facebook'u i LinkedIn,
- objętość kodu źródłowego – pisanego przez programistę oraz generowanego przez narzędzia wbudowane, niezbędnego do zapewnienia wymaganych funkcjonalności, który może być oszacowany za pomocą biblioteki *sloc* pochodzącej z repozytorium pakietów *npm* [11].

4.3. Środowisko testowe

Obie aplikacje testowe zostały uruchomione na dwóch niezależnych środowiskach testowych. Różnica w wydajności procesorów użytych komputerów w testach

syntetycznych wyniosła 30% [12]. Obydwa środowiska posiadały zainstalowany pakiet XAMPP oraz przeglądarkę Google Chrome w tych samych wersjach. Jako narzędzia testowe użyto dedykowanych debugbarów wbudowanych w porównywane szkielety. Debugbary to narzędzia deweloperskie pozwalające dokładnie sprawdzić działanie aplikacji na wszystkich jej etapach. Umożliwiają także pomiar czasu, w którym aplikacja realizuje określony scenariusz. Debugbary są narzędziami dokładnymi, odpornymi na opóźnienia oraz wpływ czynników zewnętrznych. Wersje użytych technologii oraz parametry środowisk testowych zawarte są w tabelach 1, 2 i 3.

Tabela 1: Specyfikacja użytych technologii

Wersja Google Chrome	99.0.4844.74
Wersja pakiet XAMPP	8.1.4
Wersja serwer Apache	2.4.48
Wersja PHP	8.0.7
Wersja MySQL	10.4.19

Tabela 2: Specyfikacja pierwszego środowiska testowego

Procesor	Intel Core i7-9750H
System operacyjny	Windows 10 Home 64 bit wersja 21H2
Pamięć RAM	16 GB, 2666MHz
Dysk	NVMeSKHynix_HSF256

Tabela 3: Specyfikacja drugiego środowiska testowego

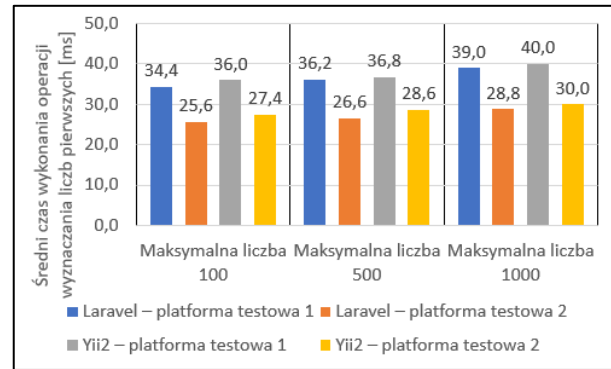
Procesor	Intel Core i7-11700K
System operacyjny	Windows 10 Pro 64 bit wersja 21H2
Pamięć RAM	16 GB, 3600MHz
Dysk	WDS100T3X0C-00SJG0

5. Wyniki badań

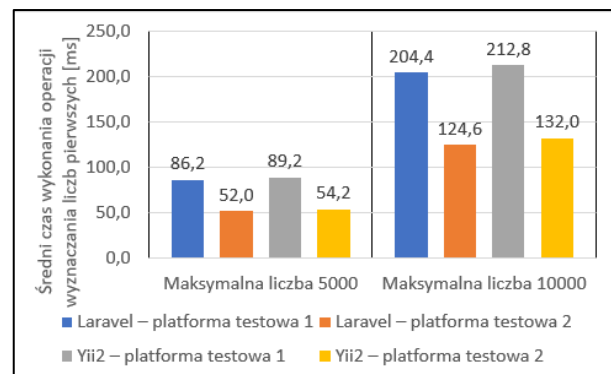
5.1. Eksperyment 1 - pomiar czasów wyznaczania liczb pierwszych

Uśrednione wyniki czasów wykonywania operacji wyznaczania liczb pierwszych, wyrażone w milisekundach, przedstawiają rysunki 1-3. Wykresy pokazują, że wyniki dla obu testowanych aplikacji są porównywalne niezależnie na jakiej platformie badawczej były wykonane.

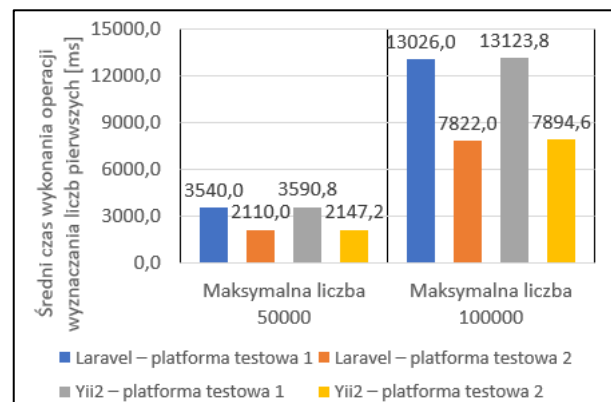
We wszystkich siedmiu przebadanych przypadkach szkielet programistyczny Laravel okazał się rozwiązaniem wydajniejszym, choć różnice były niewielkie. Na pierwszej platformie testowej maksymalna zaobserwowana różnica w wydajności między szkieletami to 5%, a na drugiej 8%. Wyznaczanie liczb pierwszych powyżej liczby 5000 powoduje, że różnice w czasach wyraźnie rosną.



Rysunek 1: Wyniki pomiarów wyznaczania liczb pierwszych dla wartości 100, 500 i 1000.



Rysunek 2: Wyniki pomiarów wyznaczania liczb pierwszych dla wartości 5000 i 10000.



Rysunek 3: Wyniki pomiarów wyznaczania liczb pierwszych do wartości 50000 i 100000.

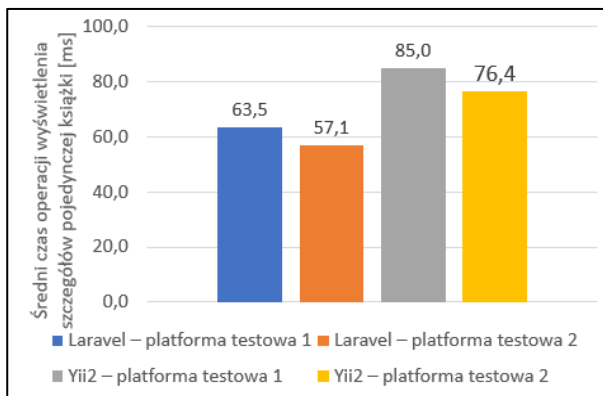
Porównując platformy testowe podczas wyznaczania liczb pierwszych na platformie 2 (mocniejszej) do wartości 100, 500 i 1000 wydajność Laravla wzrosła o 26,13% natomiast Yii do 23,73%. W przypadku wyznaczania liczb pierwszych dla zakresu 5000, 10000, 50000, 100000 nastąpił w równym stopniu wzrost wydajności dla obu platformy - dla Laravla o 39,8%, a dla Yii2 o 39,32%.

5.2. Pomiar czasów operacji CRUD

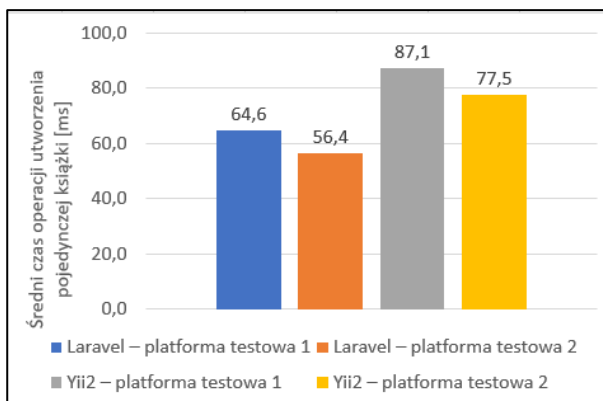
Operacje wykonywane przez moduł rankingowania książek podzielono na te, które były przeprowadzane na jednym rekordzie oraz te, które były realizowane na wielu rekordach.

Operacje wykonywane na jednym rekordzie

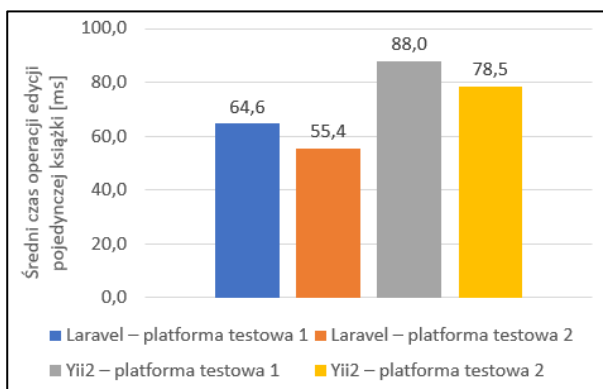
Operacje dodawania, wyświetlania, edycji i usuwania książki oraz dodawania oceny sprawdzono przy różnych poziomach zapelnienia bazy danych. Miało to jednak niezauważalny wpływ na czas realizacji zadania przez aplikację. W przypadku szkieletu programistycznego Laravel maksymalne odchylenie od średniej wynosiło 3%, a Yii2 2%. Wyniki tych operacji, wyrażone w milisekundach, przedstawiają rysunki 4-8.



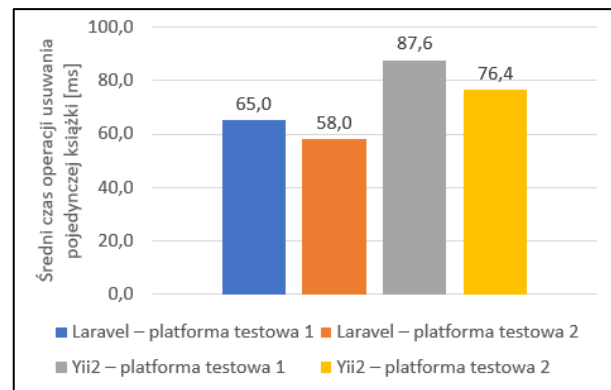
Rysunek 4: Wyniki pomiarów wyświetlenia książki.



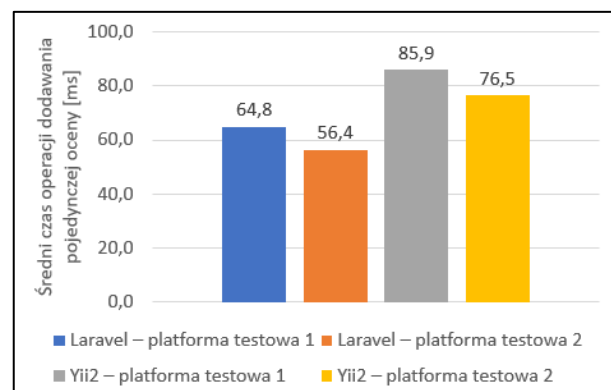
Rysunek 5: Wyniki pomiarów dodania książki.



Rysunek 6: Wyniki pomiarów edycji książki.



Rysunek 7: Wyniki pomiarów usunięcia książki.

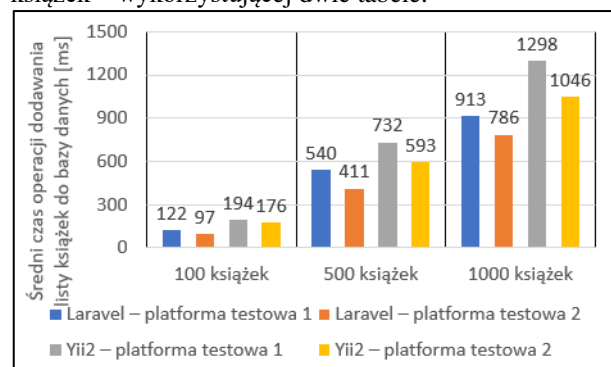


Rysunek 8: Wyniki pomiarów dodania oceny.

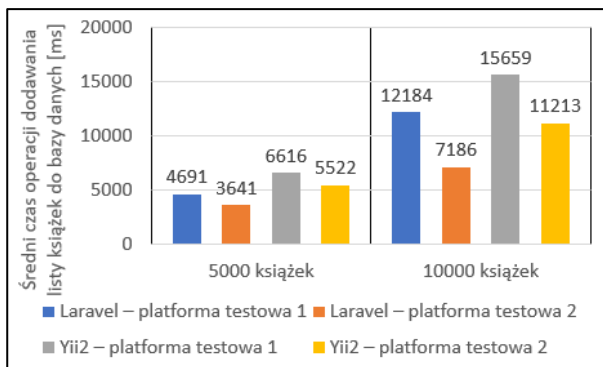
Wyniki badań pokazują większą wydajność aplikacji zbudowanej na bazie szkieletu Laravel w każdej z badanych operacji. W przypadku pierwszego stanowiska testowego średnia różnica czasu to około 38%, a w przypadku drugiego - 35% na korzyść Laravel'a. Prawdopodobnie jest to spowodowane zastosowaniem ORM (ang. Object-Relation Mapping) Eloquent, który jest znacznie wydajniejszy niż ActiveRecord stosowany w Yii2 od pierwszej jego publikacji. Zastosowanie wydajniejszej platformy testowej w tym wypadku działa na korzyść Laravla, który notuje 13% wzrost wydajności, a Yii2 11%.

Operacje wykonywane na wielu rekordach

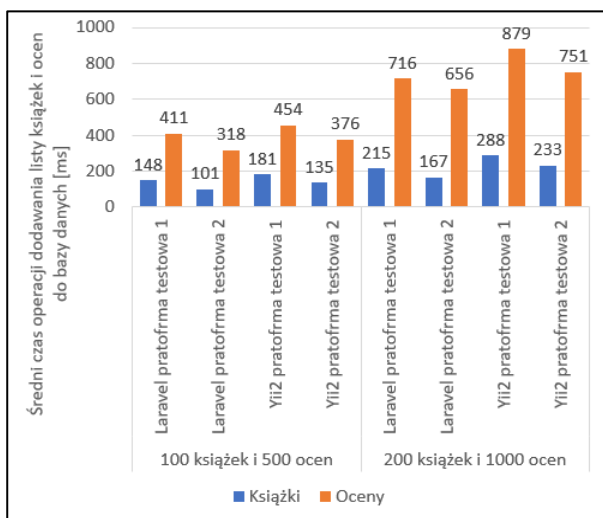
Operacjami wykonywanymi na wielu rekordach były zapis oraz wyświetlanie. Dotyczą one listy książek – operacji na jednej tabeli w bazie danych oraz rankingu książek – wykorzystującej dwie tabele.



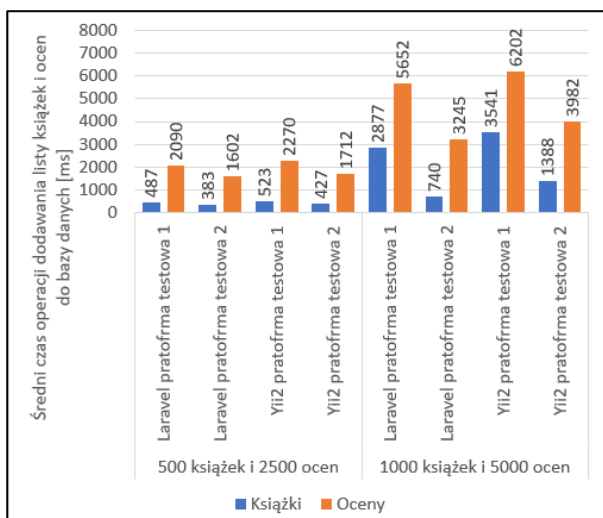
Rysunek 9: Wyniki pomiarów dodania 100, 500 i 1000 książek.



Rysunek 10: Wyniki pomiarów dodania 5000 i 10000 książek.



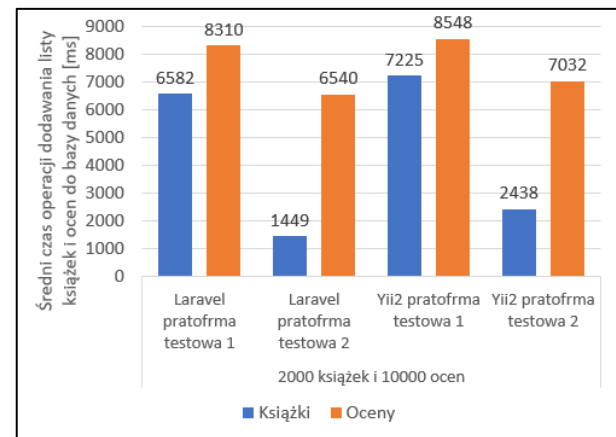
Rysunek 11: Wyniki pomiarów dodania 100 i 200 książek oraz odpowiednio 500 i 1000 ocen.



Rysunek 12: Wyniki pomiarów dodania 500 i 1000 książek oraz odpowiednio 2500 i 5000 ocen.

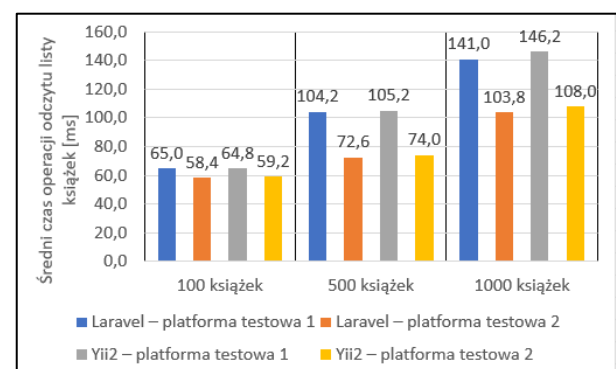
W przypadku zapełniania tabel użyto funkcji *random* i *faker* dla obydwu szkieletów programistycznych. Pozwalają one na generowanie wartości, które zostały zapisane w bazie danych. W przypadku Laravel’a wykorzystano dedykowane narzędzie do zapełniania bazy danych – *seeder*. Yii2 nie posiada jednak podobnego

pakietu. Z tego względu przygotowana została metoda o identycznej funkcjonalności jak w szkielecie Laravel z użyciem kontrolera. Wyświetlanie pozycji odbywa się w ten sam sposób niezależnie od szkieletu. W przypadku rankingu poza samym pobraniem danych z bazy jest również liczona średnia ocen dla każdej książki. Wyniki czasów wyrażone w milisekundach zapisu danych zawierają rysunki 9-13 natomiast odczytu 14-17.

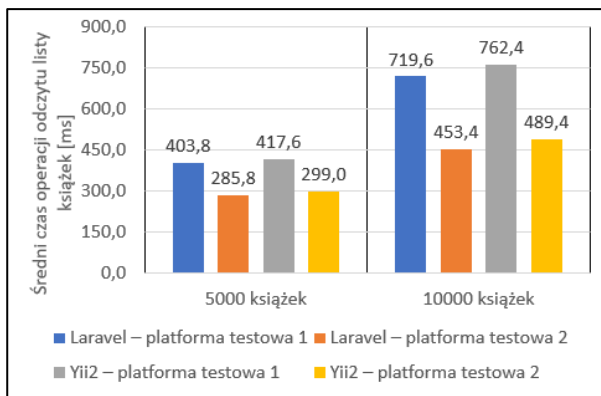


Rysunek 13: Wyniki pomiarów dodania 2000 książek i 10000 ocen.

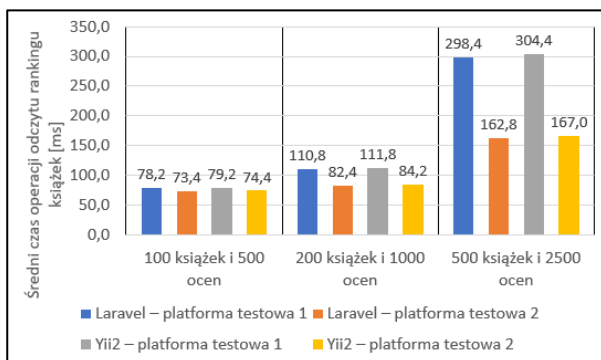
Wyniki tej części badań ponownie wykazują większą wydajność po stronie szkieletu programistycznego Laravel. Rezultaty zapełniania tabeli książek na pierwszej platformie testowej ukazują maksymalną różnicę w wydajności między szkieletami na poziomie 59%, a na drugiej platformie nawet 81%. W przypadku zapisywania danych do dwóch tabel różnice nie są tak duże, chociaż wyraźnie. Wyniki badań przeprowadzonych na urządzeniu pierwszym wykazują różnice między szkieletami Laravel – Yii2 od 7% do 25% dla tabeli książek i od 3% do 19% dla tabeli ocen. Dla drugiej platformy testowej różnice wydajności zwiększają się względem pierwszej i oscylują pomiędzy 10% a 47% dla tabeli książek oraz 6% do 19% dla tabeli ocen. W tym wypadku znaczna większość straty wydajności spowodowana jest zastosowaniem ORM Eloquent dla Laravel’a. Drugą mniej znaczącą różnicą jest dedykowane rozwiązanie do zapełniania bazy danych w szkielecie Laravel – *seeder*.



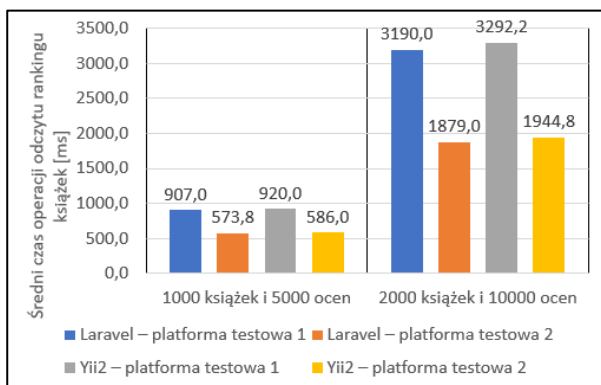
Rysunek 14: Wyniki pomiarów wyświetlenia 100, 500 i 1000 książek.



Rysunek 15: Wyniki pomiarów wyświetlenia 5000 i 10000 książek.



Rysunek 16: Wyniki pomiarów wyświetlenia 100, 200 i 500 książek oraz odpowiednio 500, 1000 i 2500 ocen.



Rysunek 17: Wyniki pomiarów dodania 1000 i 2000 książek oraz odpowiednio 5000 i 10000 ocen.

W przypadku masowego odczytu danych wydajniejszy był Laravel. Różnica uzyskanych czasów była jednak mniejsza niż dla masowego zapisu danych. W przypadku wyświetlania 100 książek Yii2 uzyskał średni czas lepszy o 0,2 milisekundy. Dla wszystkich innych pomiarów wydajność szkieletu Laravel jest wyższa o 6% na pierwszej platformie testowej oraz 7% na drugiej. Różnica zwiększa się w przypadku obydwu platform testowych im więcej danych jest przetwarzanych. Rozbieżność wydajności dla rankingu jest mniejsza i oscyluje pomiędzy 1% a 3% niezależnie od platformy testowej. Mniejsza dysproporcja wynika ze sposobu w jaki zastosowane ORM'y pracują na danych. ActiveRecord pod względem odczytu dużej ilości danych częściowo dorównuje Eloquent. Dysproporcja uzyska-

nych wyników zwiększa się na niekorzyść Yii2 im większa jest baza danych.

5.3. Poza wydajnościowe kryteria oceny szkieletów programistycznych

Głównym celem w pracy jest badanie wydajności szkieletów programistycznych, jednak istotne są też inne kryteria. W związku z tym podrozdział ten został poświęcony analizie takich kryteriów jak: możliwości technologii, dokumentację techniczną, wsparcie społeczności oraz objętość kodu źródłowego. Parametry te pomimo braku wpływu na wydajność mają duże znaczenie podczas wyboru technologii.

Możliwości technologii

Szkielet programistyczny Yii2 cechuje się większą swobodą podczas doboru technologii wspomagających. Pozwala on na zastosowanie większej liczby systemów bazodanowych, technologii tworzenia warstwy wizualnej oraz wspieranych narzędzi do pisania testów jednostkowych. Instalacja jest możliwa przy zastosowaniu starszej wersji serwera PHP. Elementy takie jak domyślny system szablonów, ORM oraz sposób debugowania różnią się w tych szkieletach jednak nie zapewniają dodatkowych funkcjonalności. Yii2 posiada generator kodu Gii, który pozwala wygenerować modele, kontrolery a nawet kompletne metody CRUD wraz z widokami. W szkielecie Yii2 bardziej rozbudowane elementy nie posiadają znaczących usprawnień. Laravel pomimo braku tak rozbudowanych udogodnień wprowadza usprawnienia dla programisty na każdym etapie pisania aplikacji.

Dokumentacja techniczna

Dokumentacje obydwu szkieletów programistycznych pod względem struktury są zbudowane bardzo podobnie. W przypadku Yii2 dokumentacja wyjaśnia szczegółowo tylko główne metody szkieletu. Dla bardziej szczegółowych danych przedstawiona jest tylko logika funkcji bez dokładniejszego wyjaśnienia oraz przykład kodu źródłowego pozwalającego osiągnąć określoną funkcjonalność. Dokumentacja szkieletu programistycznego Laravel zawiera szczegółowe wyjaśnienia oraz wiele przykładów użycia kodu. Dla każdej funkcji i metody można wybrać pożądaną wersję szkieletu co zwiększa wygodę pracy użytkownika. Wartą uwagi zaletą szkieletu Yii2 jest to, że posiada on dokumentację w języku polskim i innych 11 językach, natomiast Laravel ma dokumentację jedynie w języku angielskim.

Wsparcie społeczności

Największa platforma zrzeszająca wiele różnych forów tematycznych – Stack Exchange pozwala sprawdzić liczbę zapytań posiadających określoną frazę. Na dzień 22.05.2022 roku Laravel posiadał 544878 wątków na wszystkich forach a Yii2 43359. Podobne dysproporcje wykazuje najpopularniejsze internetowe narzędzie interakcji międzyludzkich w Polsce – Facebook. Na dzień 22.05.2022 roku polskojęzyczna grupa Laravel'a posiada 1286 członków a Yii2 10. Na grupach anglojęzycznych jest to 59982 użytkowników Laravla i 3440 w przypadku Yii2. Podobna sytuacja występuje w przy-

padku LinkedIn – serwisu społecznościowego dedykowanemu kontaktom zawodowym. Na dzień 22.05.2022 roku grupa Laravel zrzeszała około 18000, a Yii2 1100 członków.

Objętość kodu źródłowego

Dla obydwu aplikacji testowych zliczona została liczba linii kodu. Zostało zastosowane jednakowe formatowanie kodu w celu otrzymania możliwie najdokładniejszych wyników. Do zaimplementowania funkcjonalności opisanych w rozdziale 4.1 Laravel wymagał napisania 743 linii kodu a Yii2 812. Dla testowanych aplikacji Laravel wymagał o 8,5% mniej linii kodu. W przypadku Yii2 znacznie większa część została wygenerowana.

6. Wnioski

Analiza wydajności szkieletów programistycznych Laravel i Yii2 wykazała wyraźną przewagę pierwszego z nich. W 31 na 32 przypadki testowe wydajniejszy okazał się Laravel. Operacje wykonywane bez wykorzystania bazy danych wymagały zbliżonych czasów z przewagą Laravel'a o maksymalnie 8%. Większą dysproporcję wydajności wykazują operacje CRUD na pojedynczych rekordach gdzie różnica wydajności wynosi nawet 26%. Masowy odczyt danych wykazuje różnice do 7% wydajności podczas wyświetlania listy książek i 3% dla rankingu. Zapis danych do bazy wynika z opóźnienia technologicznego szkieletu Yii2. Różnice w wydajności wzrastają do poziomu maksymalnego 81% dla pojedynczej tabeli oraz 47% w przypadku dwóch tabel. Ponadto niemal w każdym wypadku różnice wydajności wzrastają w przypadku zastosowania wydajniejszej platformy testowej. Wyniki pozwalają stwierdzić zdecydowaną przewagę szkieletu programistycznego Laravel pod kątem wydajności.

Analiza dodatkowych parametrów oceny wydajności nie pozwala jasno wskazać lepszego narzędzia. Zaletą Yii2 jest większa dowolność podczas wyboru technologii wspierających np. system bazodanowy. W przypadku zastosowania rozwiązań konkurencyjnych np. ORM większą wydajnością cechują się te zastosowane w szkielecie Laravel. Jakość dokumentacji technicznej w obydwu przypadkach jest bardzo wysoka. Większą przejrzystość dla niedoświadczonego programisty zapewnia szkielet programistyczny Laravel. Posiada też on zdecydowanie większą społeczność co widać na przykładzie grup Facebookowych oraz liczby zapytań na stronie Stack Exchange. Parametrem, którego nie można określić jako jednoznacznie lepszy jest objętość kodu źródłowego. W przypadku szkieletu Yii2 linii kodu jest więcej, jednak moduł Gii wygenerował znaczną jego większość. Szkielet Laravel również pozwala generować fragmenty kodu, jednak nie posiada dedykowanego rozwiązania. Cechuje się on jednak bardziej zwięzłą strukturą kodu źródłowego. Podsumowując dodatkowe parametry oceny można stwierdzić, że Yii2 jest odpowiednim wyborem dla prostych aplikacji pisanych przez programistów z doświadczeniem w pisaniu aplikacji. Laravel jest odpowiednio przygotowany do tworzenia rozbudowanych aplikacji oraz dla mniej doświadczonych programistów.

Uzyskane wyniki oraz analiza dodatkowych parametrów potwierdzają postawioną na początku pracy hipotezę, że Laravel wydajnościowo jest lepszym rozwiązaniem. Szkielet Yii2 może być konkurencyjny tylko w wypadku prostych aplikacji realizujących podstawowe operacje CRUD dzięki modułowi generowania kodu Gii, gdzie wydajność nie jest głównym kryterium podczas wyboru technologii. W każdym innym aspekcie lepszą technologią jest Laravel ze względu na swoją wydajność oraz zautomatyzowany proces tworzenia funkcjonalności i to on powinien być pierwszym wyborem podczas doboru technologii przez programistę.

Literatura

- [1] Dane statystyczne wykorzystywania języków programistycznych w 2021 roku, <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>, [22.05.2022].
- [2] Dane statystyczne wykorzystywania szkieletów programistycznych języka PHP w 2021 roku, <https://www.jetbrains.com/lp/devecosystem-2021/php/>, [22.05.2022].
- [3] U. K. Latif, T. F. Kusumasari, Comparison Between Yii Frameworks and Laravel in 3 Different Version for Viewing Large Data of Shipyard Industry in Indonesia, *International Journal of Innovation in Enterprise System* 2 (2018) 13-18.
- [4] A. Zurkiewicz, M. Milosz, Selecting a PHP framework for a web application project-The method and case study. In *Conference: 9th International Technology, Education and Development Conference* (2015).
- [5] O. Sydoruk, Porównanie możliwości tworzenia aplikacji PHP na przykładzie Yii2 i Laravel, *Journal of Computer Sciences Institute* 11 (2019) 125-130.
- [6] K. Benmoussa, M. Laaziri, S. Khoulji, K. M. Larbi, A. El Yamami, A new model for the selection of web development frameworks: application to PHP frameworks, *International Journal of Electrical and Computer Engineering (IJECE)* 9(1) (2019) 695 – 703.
- [7] M. Laaziri, K. Benmoussa, S. Khoulji, K. M. Larbi, A. El Yamami, A comparative study of laravel and symfony PHP frameworks. *International Journal of Electrical and Computer Engineering (IJECE)* 9(1) (2019) 704-712.
- [8] K. Kuflewski, M. Dzieńkowski, Symfony and Laravel - a comparative analysis of PHP programming frameworks. *Journal of Computer Science Institute* 21 (2021) 367-372.
- [9] D. Drabik, Comparison of new ways of creating PHP applications using Laravel and CodeIgniter example. *Journal of Computer Sciences Institute* 10 (2019) 71-76.
- [10] N. Prokofyeva, V. Boltunova, Analysis and practical application of PHP frameworks in development of web information systems, *Procedia Computer Science* 104 (2017) 51-56.
- [11] Opis biblioteki SLOC, <https://www.npmjs.com/package/sloc>, [22.05.2022].
- [12] Porównanie wydajności platform testowych, <https://cpu.userbenchmark.com/Compare/Intel-Core-i7-9750H-vs-Intel-Core-i7-11700K/m766364vs4107>, [22.05.2022].