

асинхронно подієвим JavaScript оточення, котре дозволяє програмувати клієнтську та серверну частини однією мовою. До того ж ця технологія забезпечує майже на чверть вищу швидкість у порівнянні з PHP [3].

ВИСНОВКИ

Отже: функціонал мови програмування JavaScript в поєднанні з HTML5 дають можливість створювати клієнтські веб-застосунки будь-якої складності, а при додаванні технології Node.js, котра призначена для створення високопродуктивних мережових додатків та надає можливості використання операцій неблокуючого операційного введення/виведення інформації, можливо створити будь-який клієнт-серверний веб-додаток.

ЛІТЕРАТУРА

1. Про безпечність Flash Player [Електронний ресурс] – Режим доступу до ресурсу: <https://ichip.ru/tekhnologii/pravda-ili-mif-rochemu-flash-player-nebezopasen-225982>.
2. Офіційний сайт Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org>.
3. PHP vs Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/273259/>.

Пузіно М. В.

Криворізький національний університет

Рибальченко О. Г.

старший викладач, Криворізький національний університет

АВТОМАТИЗАЦІЯ РОЗГОРТАННЯ WEB-ДОДАТКІВ З ВИКОРИСТАННЯМ КОНТЕЙНЕРІВ DOCKER

У доповіді були порівняні основні методи розгортання веб-додатків та висвітлені переваги автоматизації цього процесу за допомогою програмного забезпечення Docker.

Розвиток технологій створення веб-додатків, їх супроводу та необхідність автоматизації цих процесів призвели до виникнення низки актуальних питань, пов'язаних з розробкою, доставкою та експлуатацією готового продукту. Основною з них є проблема передачі додатку клієнту та користувачеві, а також забезпечення його

широкої експлуатації. Саме задля її вирішення був створений Docker – технологія та програмне забезпечення з відкритим кодом для автоматизації розгортання додатків з підтримкою контейнеризації.

Контейнеризація – це підхід до розробки програмного забезпечення, при якому додаток і його конфігурація (абстрактні файли маніфесту розгортання) упаковуються разом в образ контейнера. Контейнерний додаток може тестуватися як єдине ціле і розгортатися як екземпляр образу контейнера в операційній системі (ОС) вузла. Тобто контейнер – це стандартна одиниця програмного забезпечення, в якій упакований додаток з усіма необхідними для його роботи залежностями – кодом програми, середовищем запуску, системними інструментами, бібліотеками і налаштуваннями. Контейнеризація програмного забезпечення дозволяє розробникам та ІТ-фахівцям розгортати його в різних середовищах без будь-яких змін або з мінімальними змінами.

Docker запускає свої процеси у власній, заздалегідь налаштованій операційній системі. Треба зазначити, що при цьому всі процеси Docker працюють на фізичному host-сервері, поділяючи всі процесори і всю доступну пам'ять з іншими процесами, запущеними в host-системі. Контейнери надають такі переваги як ізоляція, кросплатформеність, гнучкість, масштабованість і контроль, протягом усього життєвого циклу програми. Найважливішою перевагою є ізоляція середовища розробки від робочого середовища.

Розглянемо інші можливі шляхи розгортання веб-додатків. Установчий скрипт дозволяє розгорнути все необхідне для роботи проекту на потрібних серверах. Недоліком цього способу є нестійкість до помилок, які можуть виникнути під час запуску, та неможливість повернення до початкової точки в разі їх виникнення. Інший спосіб розгортання – це хмарні сервіси, на які розробник власноруч установлює всі необхідні програми, інструменти, бібліотеки, а потім виконує клонування проекту задану кількість разів. Проте, через використання хмарних сервісів запуск рішення за межами хмари стає неможливим та спостерігається різке зниження швидкості роботи. Останній спосіб розгортання – це віртуальні машини, образи яких використовуються із заздалегідь встановленим та налаштованим додатком. Недоліками такого підходу є розмір образу, що може

бути занадто великим, а також необхідність повторного завантаження і встановлення образу при будь-якій зміні в додатку.

Наразі Docker стає все більш популярним завдяки простоті використання та надійності. Контейнери на базі Docker стають фактичним стандартом в цій області і підтримуються більшістю постачальників в екосистемі Windows і Linux, такими як Майкрософт, Amazon AWS, Google і IBM. Дуже скоро Docker, ймовірно, буде широко використовуватися як в хмарних, так і локальних центрах обробки даних. Статистика поширення використання Docker приведена на Рис.1.

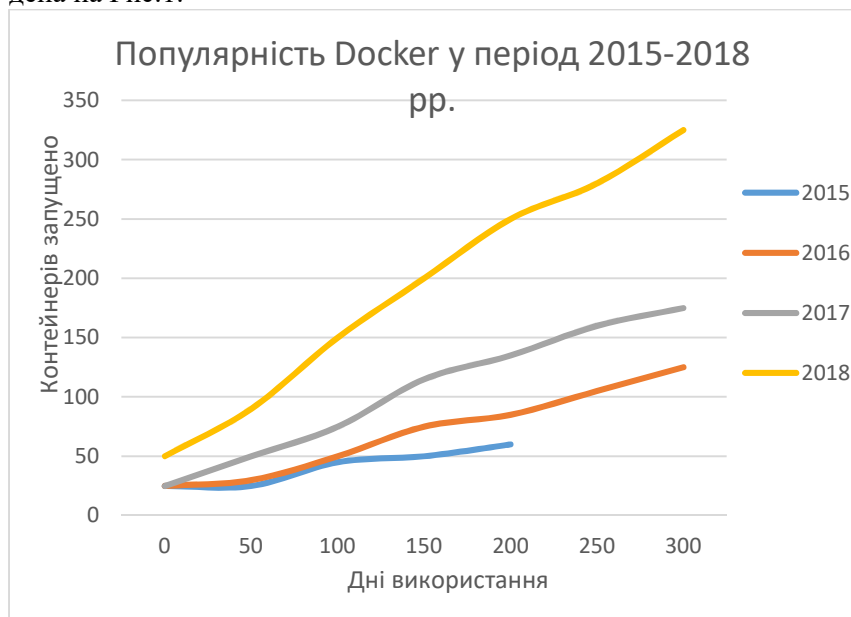


Рис.1. Популярність Docker у період 2015-2018 рр.

ВИСНОВКИ

Після порівняння шляхів автоматизації розгортання web-додатків було зроблено висновок, що Docker є зручнішим для вирішення проблем передачі серверного проекту клієнту, забезпечення тиражування серверів та оновлення раніше створених конфігурацій. Рішення на основі контейнерів надають важливі переваги з боку скорочення витрат, тому що дозволяють усувати проблеми, викликані відсутністю залежностей у робочому середовищі.

ЛІТЕРАТУРА

1. Docker Customers [Електронний ресурс] / Режим доступу: <https://www.docker.com/customers>
2. Docker [Електронний ресурс] / Режим доступу: <https://ru.wikipedia.org/wiki/Docker>
3. Docker. Зачем и как [Електронний ресурс] / Режим доступу: <https://habr.com/ru/post/309556/>
4. Usage Among Adopters [Електронний ресурс] / Режим доступу: <https://www.datadoghq.com/docker-adoption/>

*Вайдалаускас П. В.,
Державний університет «Житомирська політехніка»
Вакалюк Т. А.,
д.п.н., доцент, професор,
Державний університет «Житомирська політехніка»*

ВИКОРИСТАННЯ HIBERNATE ДЛЯ РОЗРОБКИ ПРОГРАМНИХ ДОДАТКІВ

Дано визначення технології Hibernate. Описано архітектура фреймворку та візуально представлено компоненти, що входять до складу технології. Проаналізовано переваги використання Hibernate технології для створення великих enterprise додатків.

Hibernate - це інструмент реляційного відображення об'єктів з відкритим кодом (ORM), який забезпечує основу для представлення об'єктно-орієнтованих моделей доменів у реляційних базах даних для веб-додатків.[1]

Об'єктивне реляційне відображення базується на контейнеризації об'єктів та абстракції, що забезпечує цю ємність. Абстракція дає змогу звертатися до об'єктів та керувати ними, не враховуючи, як вони пов'язані зі своїми джерелами даних. Hibernate ORM керує представленням класів Java до таблиць баз даних та типів даних Java на типи даних SQL та забезпечує запити та пошук.

Будь-які внесені зміни інкапсульовані у самому джерелі даних, так що коли ці джерела чи їх API змінюються, додатки, які використовують ORM, не повинні вносити зміни або навіть усвідом-