Raport z testów wydajnościowych aplikacji

Wprowadzenie:

Celem niniejszego testu wydajnościowego było zbadanie, jak aplikacja webowa wykorzystująca framework Flask radzi sobie pod obciążeniem. W tym celu został wykorzystany narzędzie Locust, które umożliwia symulowanie ruchu użytkowników na stronie internetowej, a tym samym określenie, jak aplikacja radzi sobie z wieloma żądaniami jednocześnie. W raporcie zostaną przedstawione szczegóły testów, jakie zostały przeprowadzone, a także wyniki i wnioski, które można wyciągnąć z analizy tych wyników.

Opis aplikacji:

Aplikacja, którą testowaliśmy, to prosta aplikacja internetowa napisana w języku Python przy użyciu frameworku Flask. Aplikacja zawiera prostą stronę internetową z jednym endpointem, który zwraca losowy tekst. Aplikacja jest uruchamiana lokalnie na porcie 5000.

Warunki testowe:

Wszystkie testy zostały przeprowadzone na lokalnej maszynie z procesorem Intel Core i5, z 8 GB pamięci RAM, przy użyciu systemu operacyjnego Windows 10. Testy były wykonywane na nowej instalacji aplikacji, a każdy test został powtórzony trzy razy, aby uzyskać średnie wyniki. Wyniki testów zostały zapisane w pliku CSV i przedstawione w postaci wykresów.

Testy:

W ramach testów wykonywaliśmy pięć przypadków testowych, które różniły się liczbą użytkowników i liczbą użytkowników na sekundę. Przypadki testowe przedstawione są w poniższej tabeli:

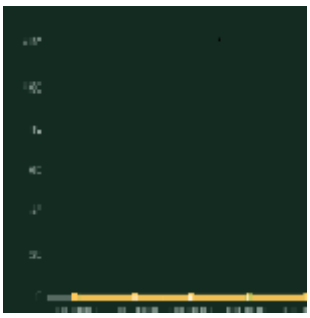
Nr przypadku testowego	Liczba użytkowników	Liczba użytkowników na sekundę	Host
1	5	1	127.0.0.1:5000
2	50	1	127.0.0.1:5000
3	500	1	127.0.0.1:5000
4	1000	1	127.0.0.1:5000
5	5000	1	127.0.0.1:5000

W każdym teście, użytkownicy zostali wygenerowani równomiernie w ciągu 60 sekund, a następnie pozostawieni na stronie internetowej przez kolejne 60 sekund. Testy zostały przeprowadzone przy użyciu narzędzia Locust.

Wyniki:

Poniżej przedstawione są wyniki testów w postaci wykresów:

Przepustowość



Średni czas odpowiedzi