

**wydajność**

# Agenda

1. Rodzaje i metryki testów
2. Planowanie i przebieg testów
3. Przegląd narzędzi
4. Zadanie domowe

# 1. Rodzaje i metryki testów

client-side:

- jak szybko zasoby dostarczone przez serwer są obsługiwane przez przeglądarkę?

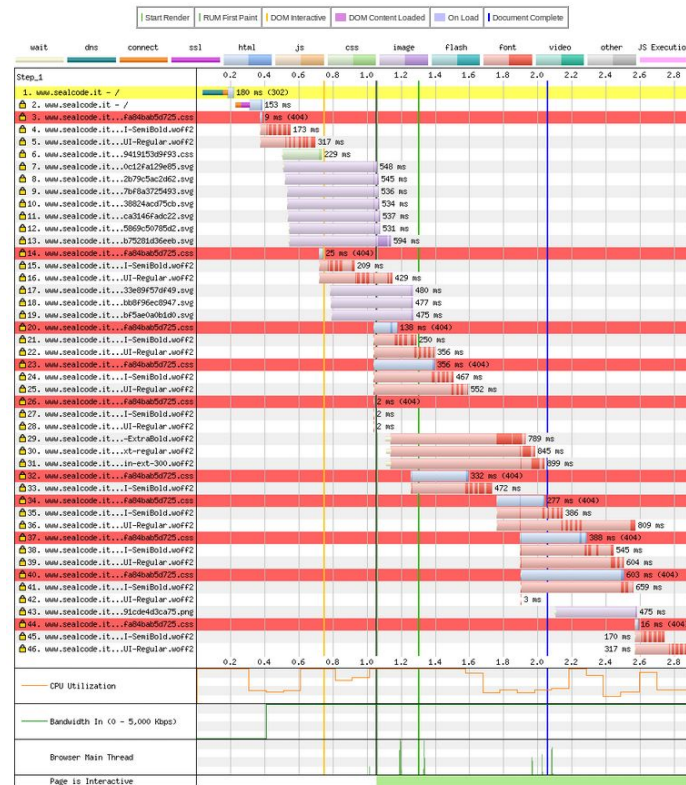
server-side:

- jak szybko serwer dostarcza zasoby?

# 1. Rodzaje i metryki testów (client-side)

## Metryki:

- first contentful   \* / meaningful   \* paint
- time to interactive   \*
- speed index   \*
- estimated input latency   \*
- first CPU idle   \*



# 1. Rodzaje i metryki testów (server-side)

Rodzaje testów:

- load testing: sprawdzamy zachowanie aplikacji w oczekiwanych warunkach
  - 200 klientów serwisu jednocześnie dokonujących operacji po zalogowaniu
- stress testing: sprawdzamy zachowanie aplikacji w ekstremalnych warunkach
  - 400 klientów serwisu jednocześnie dokonujących operacji po zalogowaniu
- endurance testing: sprawdzamy zachowanie aplikacji na dłuższą metę
  - 200 klientów serwisu jednocześnie dokonujących operacji zajmujących dłużej niż zwykle
- spike testing:
  - na przemian 40 i 400 klientów jednocześnie dokonujących operacji po zalogowaniu
- volume testing
- scalability testing

# 1. Rodzaje i metryki testów (server-side)

Metryki testów:

- transaction rate: liczba transakcji na sekundę
- availability: liczba udanych transakcji  $\div$  liczba wszystkich transakcji
- throughput: ilość danych na sekundę
- concurrency: liczba użytkowników robiących jednocześnie *różne* rzeczy w systemie

## 2. Planowanie i przebieg testów

- uwzględnienie potrzeb:
  - liczba potencjalnych użytkowników
  - lokalizacja użytkowników
  - oczekiwania użytkowników
  - platforma
  - działania marketingowe
- wyznaczenie konkretnych, wykonalnych celów
- przygotowanie rekomendacji poprawek po analizie wyników



## 2. Planowanie i przebieg testów

Co uwzględniysz w planach testów wydajności dla...

- sklepu internetowego
- strony typu brochureware
- strony ze znaczącą przewagą użytkowników mobilnych

Wybierz najważniejsze metryki i rodzaje testów, na jakich skupił(a)byś się podczas analizy wydajności tych aplikacji.



### 3. Przegląd narzędzi

Monitoring client-side performance:

- Sitespeed.io <sup>\*</sup> <sub>—</sub>
- WebPageTest <sup>\*</sup> <sub>—</sub>
- PageSpeed / Lighthouse <sup>\*</sup> <sub>—</sub>

### 3. Przegląd narzędzi: Sitespeed

1. Uruchom Sitespeeda, korzystając z pliku docker-compose. W razie wątpliwości zajrzyj do readme w repo.
2. Korzystając z przykładowej konfiguracji stwórz taką, która uruchamia 7 prób na Firefoksie na wybranych przez Ciebie 5 urlach.
3. Przeanalizuj rezultaty. Co oznaczają poszczególne metryki?

### 3. Przegląd narzędzi: Siege

Proste jak cep ;) narzędzie do load- i stress-testów.

1. Jeśli jeszcze nie zainstalowała/eś Siege'a, to zrób to. Pewnie nie obędzie się bez budowania<sup>\*</sup>.
2. Korzystając z przykładu w repo odpal Siege na 10 użytkownikach przez 20 sekund.
3. Przeanalizuj rezultaty. Co oznaczają poszczególne metryki?

### 3. Przegląd narzędzi: jMeter

Kombajn do bardziej złożonych testów server-side. Alternatywy: Gatling, Locust

1. Jeśli jeszcze nie masz jMetera, to go zainstaluj razem z wtyczką Plugin Manager.
2. Stwórz test, w którym weryfikujesz, że po przejściu z [blazedemo.com](https://blazedemo.com) na podstronę [/vacation](https://blazedemo.com/vacation) znajduje się na niej informacja o Hawajach.
3. Stwórz test, który kupuje wybrany lot z Bostonu do Nowego Jorku na 20 użytkownikach. Użyj Recordera i proxy w przeglądarce do zarejestrowania ścieżki.
4. Odpal test z punktu 3. z poziomu linii komend.

## 4. Zadanie domowe

Stwórz scenariusz zawierający co najmniej 3 różne thread groupy. Możesz użyć tej strony, postawić własną (np. gotowcem z Joomla!, Wordpressem, itp) albo zapytać kogoś w projekcie o pozwolenie na przetestowanie strony, którą właśnie tworzycie.

Przykładowe ścieżki (na podstawie strony *Computers database*):

1. wyszukaj komputer o nazwie *testemac*
2. dodaj nowy komputer
3. przejdź bezpośrednio na strony komputerów o id od 200 do 210