












Shopping App

- Audit de performanta -

Aplicatia dezvoltata este una de comert electronic, mai exact, un catalog de produse. Scenariul prezentat este cel al unui administrator de magazin online. Aceasta ii ofera posibilitatea sa vada toate produsele din sistem, cat si sa adauge, modifice si sa stearga produse, functii de baza al oricarei aplicatii.

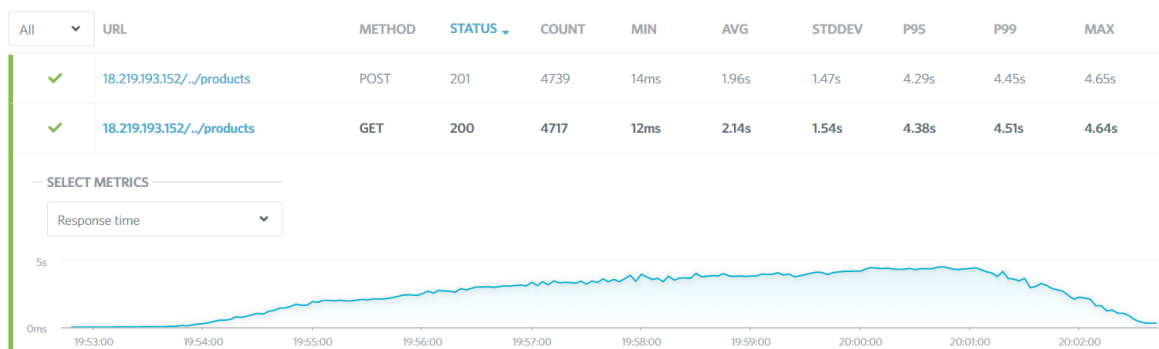
| Shopping App | | | | | ADD NEW PRODUCT |
|---------------------------------------|----------|----------|----------|---|--|
| Products | | | | | Search  |
| Name | Category | Quantity | Price | Action | |
| iPhone | Phones | 20 buc. | 4000 RON |  |  |
| iPhone | Phones | 20 buc. | 4000 RON |  |  |
| iPhone | Phones | 20 buc. | 4000 RON |  |  |
| iPhone | Phones | 20 buc. | 4000 RON |  |  |
| iPhone | Phones | 20 buc. | 4000 RON |  |  |
| Records per page: 5 ▾ 1-5 of 4757 < > | | | | | |

 Data loaded

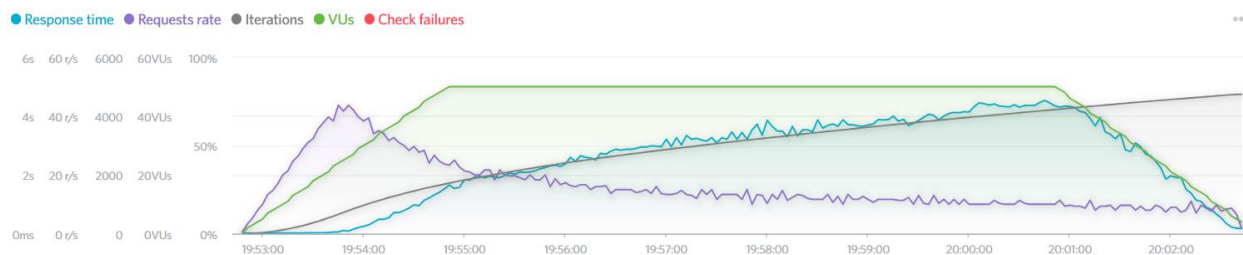
Pentru a testa performanta aplicatiei am folosit Load Impact. A fost nevoie ca aplicatia sa fie publicata pe un server web, cel de la Cloud9 fiind potrivit pentru acest lucru.

Am folosit 50 de utilizatori virtuali ai aplicatiei cu o durata de aproximativ 10 min. In acest interval de timp s-au executat aproape 9500 de request-uri catre server, jumatate din acestea fiind pentru adaugarea unui produs, iar celelalte pentru actualizarea tabelului.

In graficul de mai jos se poate observa cum numarul de utilizatori ai aplicatiei influenteaza viteza de raspuns a server-ului. Initial cat baza de date era goala, se poate vedea ca cererile sunt procesate mai rapid, fata de spre final unde desi sunt acealasi numar de utlizatori virtuali, cererile dureaza mai mult.



Testul realizat a fost unul de tip Endurance Test, unul dintre cele mai cunoscute si mai folosite, avand ca scop incarcarea unor utilizatori si realizarea de actiuni asupra serverului pe o perioada indelungata de timp cu acestia, pentru a vedea daca serverul poate rezista sub o astfel de incarcare. Graficul are forma asteptata, cu un numar de utilizatori care creste pana la un punct, dupa care intampina intarzieri de raspuns din partea serverului. Timpul de raspuns este corelat cu numarul de produse din baza de date (iteratii), iar rata de cerere invers proportional dupa ce se atinge numarul de 50 de utilizatori virtuali.

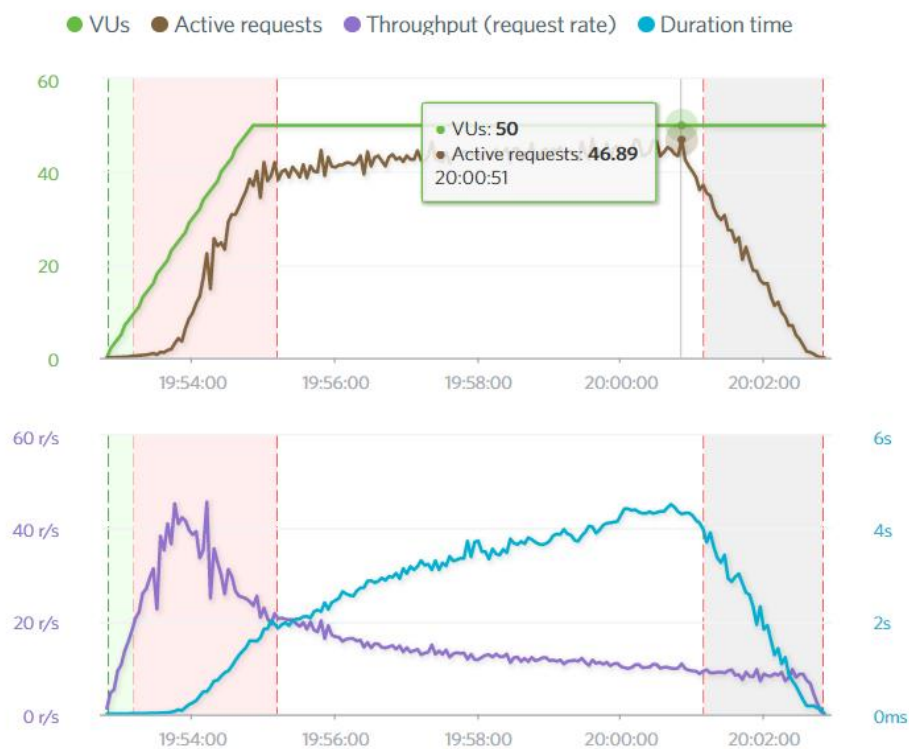


O limita de tranzitare a fost atinsa.

Numarul de request-uri active continua sa creasca, la fel ca numarul de utilizatori virtuali, in timp ce rata request-urilor finalizate stagneaza, ce indica faptul ca sistemul in timpul acestui test este supraincarcat, determinand un timp de raspuns mare.

In acest caz, este recomandat sa se coreleze acest blocaj in performanta cu datele obtinute in urma monitorizarii serverului.

In grafic se poate observa ca in prima parte a testului rata de raspuns este mult mai mare comparativ cu durata unei cereri. Odata ce numarul de utilizatori ajunge la 50, rata incepe sa scada treptat, pana la 20 r/s, iar durata atinge chiar si 4s.



Ca si recomandari exista cateva solutii. Faptul ca javascript este single-threaded vine un dezavantaj si anume limitarea folosirii unui singur nucleu din procesor, ceea ce aduce limitari de performanta. Una din solutii ar fi crearea mai multor instante ce servesc clientii si sunt conectate in sincron la baza de date pentru a asigura corectitudinea si securitatea datelor existente.