

22. TESTIRANJE PERFORMANSI

Testiranje performansi je vrsta testiranja softvera koja verifikuje da li se softver ponaša na odgovarajući način pod nekim očekivanim opterećenjem. Cilj testiranja performansi nije pronalazak novih defekata, već da se eliminišu potencijalni problemi koji utiču na performanse sistema. Testiranje performansi može da otkrije greške koje nije pokazalo funkcionalno testiranje, a to omogućava da se softver popravi i poboljša pre nego se pusti u upotrebu. Bez testiranja performansi softvera pre puštanja u upotrebu, mogu da se pojave razni problemi. Ti problemi mogu da budu spor rad softvera, u slučaju da ima više korisnika, nekonzistentnosti prilikom pristupa i sveukupne loše upotrebljivosti softvera.

Kod testiranje performansi posmatraju se:

- Brzina – proverava se brzina odziva softvera.
- Skalabilnost – testira se maksimalno korisničko opterećenje pod kojim softver može da radi.
- Stabilnost – proverava se stabilnost softvera pod različitim opterećenjima.

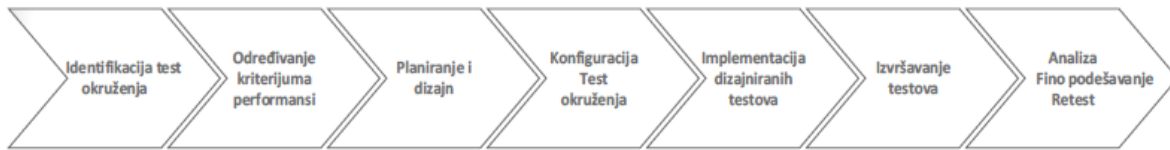
Testiranje performansi se može grupisati na više tipova:

- Testiranje opterećenja (Eng. "Load test")
- Stres test (Eng. "Stress test")
- Test izdržljivosti (Eng. "Endurance test")
- Testiranje skalabilnosti (Eng. "Scalability testing")

Najčešći problemi koji se otkrivaju testiranjem performansi:

- **Previše dugo vreme pokretanja sistema** i učitavanja aplikacije(a) – posmatra se vreme koje je potrebno za pokretanje aplikacije. Potrebno je da vreme pokretanja bude što je moguće kraće. Za kompleksne aplikacije je potrebno i više od jednog minuta za učitavanje, uopšteno je stanovište da se ovov vreme, za većinu aplikacija, treba držati u okviru od nekoliko sekundi.
- **Loše vreme odziva** – posmatra se vreme koje je potrebno, od trenutka kada korisnik izvrši unose podataka u softver ili izvrši izbor opcija i parametara, do trenutka kada sistem vrati odgovor na taj korisnički zahtev. Uopšteno gledano, ovo vreme bi trebalo da bude izuzetno kratko, jer može se desiti da korisnik izgubi interesovanje ako dugo čeka na odgovor. Ovo vreme ne bi trebalo da traje duže od nekoliko sekundi.
- **Loša skalabilnost** – ogleda se u slučajevima kada aplikacija ne može da podrži očekivani broj korisnika. Test opterećenja se koristi da bi se proverilo da li sistem može da podrži predviđeni broj korisnika.
- **Uska grla** – ograničenja sistema koja degradiraju opšte performanse celog softverskog sistema. Ponekad uzroci mogu da budu načini programiranja koja smanjuju procesorsku moć sistema pod opterećenjem. Greške mogu da budu u kodu, ali mogu da budu i hardverske (nedovoljna procesorska brzina rada, nedostatak operativne memorije, loša mrežna infrastruktura, nedostatak prostora na čvrstim diskovima i sl.).

Metodologija testiranja performansi je prikazana na sledećoj slici (Miodrag Živković – Testiranje softvera, Singidunum):



Prvo je potrebno identifikovati okruženje u kom se sprovodi test, potrebno je poznavanje detalja hardvera, softvera i mrežne konfiguracije pre početka testiranja. Zatim je potrebno definisati kriterijum prihvatljivosti za performanse, uključujući poznata ograničenja za propusnu moć, vreme odziva i alokaciju resursa. Često se kao reper koristi već uspešno realizovana slična aplikacija kao referenca za poređenje. Prilikom planiranja i dizajna testova, mora se odrediti koliko će način upotrebe aplikacije varirati među korisnicima. Nakon ovog scenarija, se kreiraju testovi i izvršavaju, uz detaljno praćenje i evidentiranje rezultata. Na osnovu rezultata se vrši analiza i uočavanje problema, koji se zatim koriguju, vrši se dodatno optimizovanje sistema i test se pokreće ponovo, da bi se uočila poboljšanja ili pogoršanja performansi. Sa podešavanjem performansi se prestaje kada procesor postane usko grlo sistema, pa je za rešavanje problema potrebno pojačanje procesorske snage.

Parametri koji se prate za vreme testiranja performansi:

- Korišćenje procesora prilikom izvršavanja operacija.
- Upotreba memorije i količina raspoložive procesoru(ima).
- Upotreba diskova, tj. vremena za koje je zauzet sa operacijama za čitanje i upis.
- Korišćenje mreže, tj. praćenje broja bitova po sekundi koje koristi mrežni interfejs.
- Broj procesorskih prekida po sekundi.
- Broj paketa koji čekaju u redu paketa na mrežnom izlazu.
- Vreme odziva od slanja zahteva od strane korisnika, do trenutka prijema odgovora.
- Propusnost sistema, tj. broj korisničkih zahteva po sekundi.
- Maksimalan broj sesija koje mogu biti aktivne u istom trenutku.
- Broj pogodaka na veb server u svakoj sekundi testa.
- Broj zaključavanja cele baze podataka ili objekata u bazi.
- Vraćanje memorije sistemu kada ona nije više potrebna aplikaciji.
- Broj SQL naredbi koje se obrade putem keširanih podataka umesto zahtevnih ulazno/izlaznih operacija.

Testiranje opterećenja (eng. Load testing)

Testiranjem opterećenja određuju se performanse sistema u okviru realnih uslova upotrebe. Ovo testiranje spada u testiranje nefunkcionalnih zahteva. Određuje se ponašanje sistema pod normalnih opterećenjima, kao i pod najvećim očekivanim opterećenjem. Identifikuje se maksimalni operativni kapacitet, uska grla ako postoje i koja komponenta izaziva opadanje performansi. U slučaju kada se opterećenje podigne iznad razumnog nivoa, test opterećenja postaje stres test. Ovaj oblik testiranja najčešće se primjenjuje za klijent/server veb aplikacije.

Test opterećenja određuje:

- Maksimalni operativni kapacitet sistema.
- Da li postojeća infrastruktura zadovoljava potrebe sistema.
- Održivost sistema u slučaju povećanog korisničkog opterećenja (eng. peak load).
- Broj konkurentnih korisnika koje sistem može da izdrži.

Koliko je testiranje opterećenja bitno, najbolje ilustruju veliki i popularni sajtovi (poput Amazona, juli 2018., veći broj sajtova avio kompanija, enciklopedija Britanica itd.) koji su duži vremenski period bili oboreni zbog ogromnog obima saobraćaja, najčešće zbog promocija koje su nudili, a nisu mogli da izdrže neočekivano visok broj zahteva korisnika.

Stres testiranje (Eng. “Stress testing”)

Stres test je vrsta testiranja u kom se **sistem stavlja pod ekstremna opterećenja i posmatra se njegovo ponašanje** za vreme ogromnog broja zahteva ili obrade podataka. Cilj stres testa je da sistem izbacivan normalnog maksimalnog operativnog kapaciteta i dovede do tačke pucanja. Prilikom određivanja tačke pucanja, određuju se i sigurnosna ograničenja sistema, koja se onda upoređuju sa zahtevima iz specifikacija. Na taj način se određuje stabilnost i pouzdanost sistema, kao i da li sistem ispunjava zahteve iz specifikacije sistema.

Ovde je vrlo važno da se utvrdi kada sistem tačno “puca”, odnosno da se utvrdi oblik otkaza sistema. Sistem ne bi trebalo da u potpunosti otkáže ni u slučaju ekstremnog opterećenja. Često, ni u slučaju ekstremnog opterećenja, sistem ne bi trebao da otkáže i prestane sa radom, već da prikaže odgovarajuću poruku o grešci.

Svaki stres test treba da se fokusira na sledeće:

- Provera da li sistem radi pod ekstremnim opterećenjima.
- Provera da li sistem prikazuje odgovarajuće poruke u slučaju kada je opterećenje toliko da ne može da odgovori na zahtev korisnika.
- Ako sistem potpuno otkáže, dolazi do gubitka podataka, novca i poverenja korisnika.

U stres testiranju, analizira se i ponašanje sistema nakon otkaza. Za uspešno izvršen stres test, potrebno je da sistem ispiše poruku o grešci i da bude bez gubitka kritičnih podataka. U ovoj vrsti testiranja, tester koristi masivne setove podataka kako bi izazvali ekstremno opterećenje. Nakon otkaza sistema, potrebno je i verifikovati uspešan oporavak sistema.

Instalaciono testiranje

Tip testiranja koji osigurava to da su sve mogućnosti i opcije pravilno instalirane, proverava da li su sve komponente aplikacija instalirane. Kod većih sistema postoji nekoliko načina za instalaciju, dok se kod manjih instalacija radi direktno. Ukoliko se želi postojeći sistem zameniti novim, instalacija novog sistema se vrši u paralelnom modu, koji omogućava da oba sistema rade istovremeno u jednom periodu. Prati se rad korisnika i kad se steknu uslovi, tj. korisnik se navikne na novi sistem, stari sistem se deinstalira ili briše. Većina kompanije postavljaju nekoliko servera (multi – server) na jedan sistem, koji su nezavisni jedan od drugog. Dobar način za instaliranje sistema je probna instalacija na jednom odabranom serveru, posle provere dužine trajanja, instalira se na drugom serveru.

Instalaciono testiranje treba da obezbedi sledeće:

- Provera verzije istog softvera, prethodna verzija ne bi trebala biti postavljena posle nove verzije.
- Instalater bi trebao da da podrazumevanu putanju instalacije.
- Instalater treba da dopusti korisniku instalaciju na drugoj lokaciji, drugačijoj od podrazumevane.
- Provera da li se proizvod može instalirati preko mreže.
- Omogućiti autorun aplikacije kada se postavi na CD/DVD ROM.
- Omogućiti korisniku opciju REMOVE.
- Pri deinstaliranju proveriti da li su svi folderi, dinamičke biblioteke, delovi registra, prečica, activeX komponente uklonjene iz operativnog sistema.
- Pokušati instaliranje softvera bez administrativnih prava.
- Pokušati instaliranje na različitim verzijama operativnog sistema.
- Pokušati instalaciju na slabijem računaru koji ima manje operativne memorije, prostor na diskovima i sl.