

Uvod

Današnje veb stranice i aplikacije su značajno kompleksnije nego što su bile pre deset godina.

Softverski inženjeri koriste više JavaScript, više slika i dodaju kompleksne slojeve koji jednostavno nisu bili razmatrani sredinom 2000-tih.

Način na koji koristimo veb i komuniciramo sa aplikacijama se takođe promenio, zahvaljujući upotrebi mobilnih uređaja. Upotreba se više ne svodi na korišćenje desktop računara – sada nam mobilni uređaji daju slobodu pristupa internetu sa bilo kojeg mesta, u bilo koje vreme. Uporedo sa promenom tehnologije i ponašanja korisnika, promenila su se i očekivanja u smislu korisničkog iskustva. U današnje vreme od softvera se očekuje samo da radi, a preduzeća se na njega oslanjaju kako bi isporučila dobra i usluge koje kupci očekuju.

Pomenutoj složenosti dodaje se neumorni porast broja internet korisnika širom sveta. To sigurno predstavlja ogromnu priliku za poslovanje – više klijenata, korisnika web sajtova i aplikacija, nova tržišta za osvajanje. Ali to takođe predstavlja osnovu za brigu o performansama kao rezultatu globalnog rasta korisnika i aplikacija, kao i kompleksnosti veb sajtova.

Srećom, na preseku između tehnologije i performansi, moguće je identifikovati uska grla i osmisliti nove strategije za ublažavanje njihovog uticaja na performanse aplikacija.

Otkrijmo kako



U periodu između marta 2007. i decembra 2017. broj korisnika interneta širom sveta **skočio je za 37%**, sa samo 1.1 milijardi do čak 4,2 milijarde

Primarno “Front-end” usko grlo

Jedno od primarnih ograničenja pre pet godina bila je mreža. Optimizacija se fokusirala na isporuku sadržaja od servera do klijenta na najbrži i najefikasniji mogući način. Komprimirali smo informacije poslate preko mreže, objedinjavali zahteve kako bismo izbegli višestruke HTTP zahteve i omogućili “Keep-Alive” da bi se održala ista TCP konekcija.

U današnje vreme stvari su se promenile. Mreža nije ona koju krivimo za većinu uskih grla u performansama – to je CPU. Ova promena nastala je usled dva ključna pomeranja: Mobilni uređaji su postali dominantan način komunikacije i drugo - performanse na uređaju su postale primarni fokus developera. Do tog momenta se podrazumevalo da ljudi koriste laptop računare za pristup internetu. Širokim usvajanjem mobilnih uređaja, međutim, ljudi su imali moćan ručni računar koji je bio produžetak mreže, ali sa različitim ograničenjima u pogledu brzine obrade, memorije i drugih faktora. Performanse uređaja su brzo postale nešto što treba istražiti detaljnije.

Sa vebom vršimo zamenu “thick” i “thin” klijentske infrastrukture i trenutno smo definitivno u “thick” klijentskoj fazi. Kao rezultat toga, prešli smo na izgradnju naših veb lokacija sa puno Java Script-a. Sve moćniji standardi omogućuju nam da radimo stvari koje pre samo nekoliko godina nismo mogli ni da sanjamo. Spuštamo mnogo skripti na ove “thick” klijente. HTTP arhiva, na primer, sadrži oko 370 kilobajta kompresovanog Java Script-a – ili otprilike dva megabajta koda, koji, kada je dekompresovan na uređaju, klijent mora da raščlani, kompajlira i izvrši. Nije iznenađujuće što sve to uzima danak na CPU i stvarno usporava stvari. Takođe šaljemo teže slike za koje je potrebno više memorije, kao i napredne crteže, očekujući da će se uređaj pobrinuti za sve. Kao rezultat toga, mreža je postala manji problem.



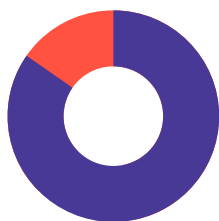
2018.godine, **66% osoba** u ključnim državama je posedovalo pametan telefon, više od 58% u 2016.

Šta predstavlja problem?

Činjenica da jeftini Android uređaji pokreću rast interneta globalno. Ako vaše preduzeće teži da dosegne to tržište, potrebno je da obratite pažnju na performanse brauzera na mobilnim uređajima, posebno onima bez puno memorije ili moćnog procesora. Štaviše, današnje veb stranice i aplikacije su mnogo zahtevnije u pogledu performansi nego što su bile. Sadrže više Java Script-a nego ikada, više slika i CSS (Cascading Style Sheet). Oni među nama koji kreiraju te veb stranice i aplikacije često previdaju činjenicu da koristimo najsavremeniju tehnologiju da bismo to uradili – i da to može proizvesti značajne “mrtve uglove” u pogledu onoga što gradimo. Inženjeri mogu potrošiti koliko god je potrebno da nabave najnoviji MacBook, a drugi developer se može razmetati vrhunskim Surface Book-om. Ali posetioци vašeg sajta i korisnici aplikacija verovatno koriste nešto manje moćno da bi koristili ono što napravite. Moguće je čak da oni koriste nešto nižeg ranga, kao Chromebook.

Apsolutno je presudno testirati na širokom spektru uređaja kako bi se osigurala performanse aplikacije.

Ovaj korak je posebno važan ako vaše ciljno tržište uključuje regije u razvoju u kojima su slabo razvijeni mobilni uređaji i nekonzistentna pokrivenost signalom norma. Dakle, uzmite taj jeftini pametni telefon, zađite usred šume i pokušajte da koristite svoju veb stranicu ili aplikaciju. Tokom testiranja koristite WebPagetest za simulaciju zaista loše mreže, a Chrome DevTools za regulisanje procesora. I nemojte ga štedeti (nemojte biti nežni). Guranjem do krajnjih granica – gušenjem procesora, mreže i uopšte stvaranjem zaista groznog okruženja - otkrićete slabosti vaše aplikacije ili sajta. Ovi testovi otpornosti na stres takođe će olakšati pronalaženje smetnji u performansama.



IDC prognozira da će u 2022.
Android telefoni iznositi
85,3% globalnih isporuka.

Izaberite pravi alat za posao

Tokom godina programerima je bila potrebna pomoć u izgradnji sve složenijih sajtova i aplikacija sa većom funkcionalnošću. Popularni alati poput React-a, Vue and jQuery su im znatno olakšali život, nudeći ugrađene metode, strukture i biblioteke koji se bave velikim delom ove složenosti. U nekim slučajevima pisanje nečeg kompleksnog postalo je jednostavno poput uvlačenja druge datoteke ili nekoliko linija koda. Ali ovakav pristup podrazumeva pravljenje kompromisa. Jedan koji se obično zanemaruje jeste to da obično nasleđujemo prtljag koji dolazi sa ovim metodologijama. React je, na primer, originalno napravljen da reši probleme programerima na Facebook-u, što je kasnije JavaScript biblioteku učinilo javno dostupnom. Facebook, međutim, ima veoma različite probleme i izazove od većine nas. React je uspeo kod njih, ali možda nije dobar za nas ostale, jer uglavnom pruža gomilu stvari koje nam nisu potrebne. Isti kompromis važi i za druge tehnologije kao što su Vue i jQuery. Dobra strana mu je što je razvoj lakši sa bibliotekama zasnovanim na komponentama u Angular-u, React-u i Vue-u. Dobijamo dobru organizaciju koda, postojeću dokumentaciju i uključenu zajednicu. Samo zapamtite da ovi alati pružaju mnogo dodatnih funkcionalnosti koje vam verovatno nikada neće trebati niti ćete ikada ući u njih. Ako postoji "lakša" alternativa, potražite je. Na primer, umesto React-a, mnogo ljudi koristi Preact, koji predstavlja samo deo cele React biblioteke. Preact uklanja mnoštvo funkcionalnosti koje većini ljudi nisu potrebne. U najmanju ruku, ako gradite nešto što je teško za JavaScript, pokušajte da imate dobar proces za analizu JavaScript paketa koje kreirate. Muče vas duplikati koda? Pouzdan alat i dobar postupak izrade identifikovaće i pomoći u uklanjanju duplikata koda, omogućavajući vam da isporučite nešto što je moguće lakše.



Optimizacija za globalno tržište

Kompanijama je sve potrebnije da dosegnu globalnu publiku. Razvijena tržišta na koja se obično fokusiramo – mesta poput Evrope i Severne Amerike – su visoko zasićena u pogledu korišćenja interneta. I, prema skorašnjoj studiji Consumer Technology Association, zastupljenost smart telefona u domaćinstvima u Sjedinjenim Državama je dostigla 87%. Tržišta zemalja u razvoju kao što su Indija, Afrika i Kina, s druge strane, imaju daleko veći razvojni potencijal. Posebno Kina koja uvodi dodatni sloj složenosti zahvaljujući svom projektu Zlatni štit – koji se često naziva “Veliki Kineski zaštitni zid” – lavirint zakonskih akcija i tehnologija koje blokiraju pristup nekim stranim veb sajtovima i usporavaju prekogranični internet saobraćaj. Tržišta u razvoju donose različita mrežna okruženja sa mnoštvom novih ograničenja. Kašnjenje je veliki problem u nekim od ovih regiona. U Africi, na primer, mnoge mreže za isporuku sadržaja (CDN) imaju slabu pokrivenost, što obično znači da korisnik mora pristupati serveru na drugom kontinentu. Mreže malog propusnog opsega, velikog kašnjenja, specifična ograničenja u izradi jeftinijih Android uređaja – sve su to velika pitanja na tržištima u razvoju, ali uglavnom nisu aktuelne teme za konferencije ili članke. Ove teme se mogu pojaviti povremeno, naravno, ali razgovori o proksi uslugama i pretraživačima nije ni približno uobičajen kao svakodnevne rasprave o stvarima kao što su React ili najnovija JavaScript metodologija ili veb standard. Ovo nije previše iznenađujuće jer većina programera nije potrošila mnogo vremena razmišljajući o mrežnim pitanjima i upotrebi uređaja u zemljama u razvoju. Ali trebalo bi. Zato što kao programer morate poznavati vaše tržište – koje mreže i uređaje ljudi koriste tamo i sa kojim problemima i ograničenjima ćete morati da se nosite. Bez razumevanja ovih dimenzija projekta, pucate u prazno i borite se sa problemima u performansama, a da zaista ne znate sa čime se suočavate.

87%

Zastupljenost smart telefona u domaćinstvima u Sjedinjenim Državama je **dostigla 87%**.

Rad unutar “Velikog Zaštitnog Zida” i zašto vam je potrebna defanzivna strategija razvoja

Narodna Republika Kina blokira pristup informacijama, podacima i uslugama koje smatra nepovoljnim ili nepoželjnim. Taj zabranjeni saobraćaj uključuje mnoge veb sajtove i usluge koje dominiraju našom društvenom komunikacijom: stvari kao Twitter, Snapchat i mnogi Google servisi kao pretraga i YouTube. Ako vaš sajt ili aplikacija pokušava da pristupi blokiranom domenu iz Kine, to takođe može uticati na njene performanse, zbog čega će biti neupotrebljiva nekoliko sekundi pre nego što se nešto okonča i stvari nastave dalje. Unutar Velikog Zaštitnog Zida, odbrambena razvojna strategija je od ključne važnosti. Vi morate vaš sajt ili aplikaciju napraviti tako da sve što povučete izvan svog domena – ukoliko propadne – ne upropasti i vašu uslugu. Ovo je dobar razvojni savet generalno, naravno, ali posebno relevantan u zemlji u kojoj sadržaj može biti blokiran. Možda ćete i sami blokirati svoje domene. Mnogo je faktora, kako tehničkih, tako i političkih, zbog kojih je teško znati koji servisi će dobro funkcionisati u ovom okruženju. Nemojte samo uzeti prvi CDN (Content Delivery Network) sa police. Umesto toga, uradite svoj domaći zadatak. Pre ulaska na tržište obratite se svojoj potencijalnoj publici i pitajte ih kakva je njihova pokrivenost internetom i steknite mnogo različitih gledišta na tu temu.

Kako Veliki Zaštitni Zid radi:

- URL filtriranje
- Trovanje DNS-om
- Samo-cenzura
- Ručno izvršenje
- VPN blokiranje

Kada propisi mogu pomoći performansama

Evropski (GDPR) General Data Protection Regulation je doneo najveće promene u zakonu o privatnosti podataka. Za veb programere koji uklanjaju sadržaj nezavisnih proizvođača sa svojih veb sajtova kako bi se pridržavali strogih GDPR ograničenja u razmeni podataka, promene bi mogle biti pozitivne sa stanovišta performansi. Arhitekta veb performansi Pol Kalvano u tvitu iz maja 2018.godine pružio je sjajan primer za to. Američka verzija sajta novinske agencije USA Today, primetio je Kalvano, imala je veličinu 5.5 megabajta, sa više od 800 zahteva za informacijama u vezi sa oglasima, učitanih sa 188 različitih domena. Ali EU verzija, koje je imala veličinu manju od pola megabajta i nije sadržala sadržaj trećih proizvođača, učitala se mnogo brže i pratila daleko manje podataka. GDPR je fascinantna studija slučaja jer stavlja brojna ograničenja na sadržaj trećih strana koji ubija performanse, upravo ono na šta su se programeri žalili duži vremenski period. Biće interesantno videti kako ovi oslabljeni veb sajtovi utiču na krajnje korisnike. Videli smo mnoštvo dokaza godinama unazad da kada se performanse poboljšaju, povećava se aktivnost korisnika. Više ljudi koristi veb sajtove, troše više, čitaju više i sl. Kao što je veb dizajner Ethan Marcotte rekao pre više godina, “performanse su poslovni problem”. Potrebne su vam solidne performanse da bi vaše poslovanje radilo dobro online, ali mnogo oglasa i analitike koji prave novac, negativno utiču na performanse. U krajnjoj liniji, nije jednostavno. GDPR može primorati kompanije da malo više eksperimentišu i nađu nove, kreativne načine da poboljšaju i performanse i prihod u isto vreme. Bolje performanse mogu imati demonstrativno pozitivan uticaj i na metriku korisnika i na metriku poslovanja.

Vodite računa o serverskoj strani

Mnogo smo govorili o performansama sa korisničke strane, koje su sigurno veoma važne i gde se dešava mnogo problema u funkcionisanju. Ali ako nadgledate samo korisnički interfejs, odnosno korisničku stranu aplikacije, a ne poklanjate pažnju šta se dešava sa vašom aplikacijom na serveru, vi u tom slučaju ignorišete ogroman deo vašeg sajta ili aplikacije – i potencijalno se izlažete velikim nevoljama. Postoji opšteprihvaćeno pravilo da se najviše problema sa performansama dešava na klijentu. Ako dođe do problema na serveru, rešenje je obično izuzetno skupo i od one vrste koju želite da otklonite brzo. Kada nešto sa vašom aplikacijom pođe po zlu na serverskoj strani, to nije mala stvar. Jer, pored brzog odgovora koji mora da obezbedi klijentu, server mora da podnese i opterećenje. A kada stvari prođu, ne želite da vaš server ostane paralisan i u stanju da ne može da odgovori. Postoji mala razlika između teške JavaScript “front-end” arhitekture i sloja Node.js koji se izvodi na serveru. Takođe ima puno potencijala za front-end programere da iskoriste činjenicu da sada možemo pisati JavaScript na serveru. Mnoge metodologije počinju da se ugrađuju u (ili bar pružaju dodatke za) sloj Node.js koji pre-kompajlira JavaScript, kreirajući HTML i prosleđujući ga nazad (u osnovi ga unapred generišući). Potencijal Node.js je bio u ovome: “Hej, front-end programeri, sada imamo ovaj srednji sloj koji vam dopušta da proširite ono što radite u JavaScript-u na serveru.” A opšti odgovor je bio: “To je sjajno! Hajde da nadogradimo još stvari na klijentu.” Dakle, postoji ovaj ceo nedostajući sloj koji nismo dovoljno iskoristili, ili bar ne još uvek. Iz JavaScript perspektive, ista uska grla ili greške u performansama na klijentskoj strani, mogu se pojaviti i na serveru, što morate nadgledati pažljivo kako biste bili sigurni da sve radi kako treba. Softver za nadgledanje performansi aplikacija (APM) vam može pomoći da nadgledate i detektujete anomalije pre nego što uzrokuju veći problem. Po pitanju APM softvera, mišljenja front-end i back-end programera se značajno razlikuju. Manje je verovatno da će sajtovi sa JavaScript teškim front-endom koristiti APM, dok kompanije koje primenjuju tradicionalniji pristup koji podrazumeva odgovor servera klijentu, ili oni sa visoko kompleksnim back-endom, verovatnije već koriste APM. U stvari, oba tabora zaslužuju u potpunosti integrisano rešenje za nadgledanje, ono koje povezuje ono što korisnik doživljava na front-endu sa bilo kakvom obradom koja se odvija u pozadini, a to se često radi kroz koncept Poslovne Transakcije (Business Transaction).

Prednosti APM softvera:



Mapiranje aplikacija



Dinamičko osnovno
povezivanje



Dijagnostika na nivou koda

Izbegnite sklonost ka familijarnosti

Svi mi imamo sklonost ka onome što poznajemo i što nam odgovara. To je u ljudskoj prirodi. Čuo sam mnoge JavaScript programere kako govore kako bi bilo dobro kada nikada više ne bi morali da dotaknu server, kada nikada više ne bi morali da pokrenu bilo šta na serveru, nego da umesto toga pokreću kod isključivo na klijentu. Budući da poznaju front-end i da se tu osećaju komforno, tu dolazi do izražaja njihova sklonost ka familijarnosti. Međutim, to ne znači da oni ignorišu probleme na serverskoj strani. Kada se problem pojavi, oni ga iskopaju i reše. Ali, opet, veća je verovatnoća da će obratiti pažnju na front-end, na koji su se fokusirali većim delom svoje karijere. Kao rezultat, server verovatno nije najvažniji. Zbog toga je od ključne važnosti omogućiti praćenje i nadgledanje nečega što se dešava po definiciji na serverskoj strani. Sklonost ka familijarizaciji može takođe dovesti do scenarija "igre optuživanja", gde jedna strana krivi drugu nakon što se dogodi nešto loše. Da biste rešili problem i poboljšali performanse moraćete da okupite ceo tim. Želite da u vlasništvu vaših front-end ljudi budu performanse koje se dešavaju na front-endu, a da ljudi iz pozadine obraćaju pažnju na ono što se dešava sa aplikacijom na serverskom nivou. Takođe želite da vaši dizajneri razmišljaju o performansama kada kreiraju oznake i stranice. To je jedna od oblasti koju mi kao programeri ne razmatramo ozbiljno, ali koja vrši ogroman uticaj. Radio sam sa dizajnerima koji su veoma orjentisani ka performansama i kada oni dobro odrade posao, sve krene kako treba.

Fokusirajmo se na zajednički cilj

Izvanredne performanse su u najboljem interesu svih i svaki član vašeg tima mora da deli ovaj zajednički cilj. Kao lider, morate jasno definisati ovaj cilj tako da svi znaju šta je to i da se trude da ga postignu. Pristup “Prebaci preko zida”- gde se svaki tim fokusira na svoj uski skup zadataka, ignorišući one za koje nije direktno odgovoran – stvara okruženje u kojem je teško postići dobre performanse i otpornost, jer vašem timu nedostaje etos “svi za jednog”.

Želite li da ostvarite vidljivost performansi u Kubernetes-u i svim mikroservisima koje razvijate?

Treba da razgovaramo.

