SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Zoran Antolović**

WEB SUSTAVI VISOKIH PERFORMANSI BAZIRANI NA PHP-u

DIPLOMSKI rad

Varaždin, 2017.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Zoran Antolović

Matični broj: 43579/14–R

Studij: Informacijsko i programsko inženjerstvo

WEB SUSTAVI VISOKIH PERFORMANSI BAZIRANI NA PHP-u

DIPLOMSKI rad

Mentor:

Prof. dr. sc. Dragutin Kermek

Varaždin, kolovoz 2017.

**Sadržaj**

1. Uvod 1

2. World Wide Web 3

2.1. Nastanak i razvoj World Wide Weba 3

2.1.1. HTTP - Hypertext Transfer Protocol 3

2.1.2. Web i Web 2.0 5

2.2. Web sustavi 5

2.2.1. Web stranice 6

2.2.2. Web servisi 6

2.2.3. Web aplikacije 6

2.3. Web tehnologije 7

2.3.1. Razvoj web sustava na strani poslužitelja - Backend tehnologije 7

2.3.2. Razvoj web sustava na strani klijenta - Frontend tehnologije 7

3. PHP programski jezik 8

3.1. Razvoj PHP programskog jezika 8

3.2. Razvoj suvremenih web sustava u PHP programskom jeziku 13

3.2.1. Razvojni ekosustav 13

3.2.2. On-line zajednica 13

3.2.3. Kritike programskog jezika PHP 13

4. Performanse web sustava 14

4.1. Web sustavi visokih performansi 14

4.2. Performansne karakteristike web sustava 15

4.2.1. Propusnost web sustava 15

4.2.2. Vrijeme odziva web sustava 15

4.2.3. Dostupnost web sustava 15

4.3. Testiranje performansi web sustava 15

4.3.1. Testiranje web sustava pod opterećenjem 15

4.3.2. Testiranje granica web sustava 15

4.3.3. Regresijsko testiranje web sustava 15

5. Razvoj web sustava visokih performansi 16

5.1. Tijek razvoja web sustava visokih performansi 16

5.2. Mjerenje i optimizacija performansi 16

5.2.1. Optimizacija aplikacijskog sloja web sustava 16

5.2.1.1. Pronalazak uskih grla i optimizacija koda 16

5.2.1.2. Pohrana podataka u privremenu memoriju 16

5.2.2. Optimizacija podatkovnog sloja web sustava 16

5.2.2.1. Optimizacija upita prema bazi podataka 16

5.2.2.2. Prilagodba strukture baze podataka s ciljem optimizacije sustava 16

5.2.2.3. Pohrana rezultata u privremenu memoriju 16

5.2.3. Optimizacija infrastrukturnog sloja web sustava 16

5.2.3.1. HTTP zaglavlja i pohrana podataka u privremenu memoriju 16

5.2.3.2. Kompresija podataka i pohrana podataka u privremenu memoriju preglednika 16

6. Praktičan rad – web sustav visokih performansi 17

6.1. Opis web sustava 17

6.1.1. Struktura web sustava 17

6.1.2. Infrastruktura 17

6.1.3. Model baze podataka 17

6.1.4. Koraci razvoja i optimizacije web sustava 17

6.2. Optimizacija i mjerenje performansi web sustava 17

6.2.1. Bez korištenja privremene memorije i indeksa u bazi podataka 17

6.2.2. Uz korištenje indeksa u bazi podataka 17

6.2.3. Uz korištenje pohrane podataka u privremenu memoriju na strani aplikacije 17

6.2.4. Uz korištenje pohrane rezultata u privremenu memoriju na strani HTTP servera za privremenu pohranu 17

7. Literatura 18

8. Izvorni kod praktičnog primjera 20

1. Uvod

World Wide Web (WWW, web) nastao je 1989. godine kada je Tim Berners-Lee u CERN-u izradio prijedlog sustava za upravljanje i dijeljenje informacija kako bi kolegama znanstvenicima omogućio jednostavniju razmjenu informacija kako unutar institucije tako i između znanstveno-istraživačkih institucija[[1]](#footnote-1). Prva je web stranica objavljena 20. prosinca 1990. godine, a kao prvi web poslužitelj (eng. web server) Berners-Lee je iskoristio svoje NeXT računalo. Ta je web stranica opisivala koncept i osnovne funkcionalnosti weba, sadržavala je upute za pristupanje dokumentima drugih osoba te upute za postavljanje osobnog servera[[2]](#footnote-2). Izvorni kod World Wide Web-a javno je objavljen 1993. godine.

27 godina od objave prve web stranice, Internet broji više od 3.5 milijardi korisnika, 966 milijuna web stranica, a gotovo 50% svjetske populacije ostvaruje pristup Internetu uz značajnu korelaciju (0.87) stupnja razvijenosti države i mogućnosti pristupa Internetu (Poushter, 2016; Internet World Stats, 2017; VPN Mentor, 2017).

Razvojem weba i porastom broja korisnika weba značajno je prošireno područje informatičke i računalne industrije, marketinga, prodaje ali i drugih industrija kao što su industrija zabave i turizam. Stvorena su nova zanimanja poput web dizajnera i programera, upravitelja društvenim mrežama (eng. *community manager*, *social media manager*), a sam web preuzeo je ulogu vodećeg medija za distribuciju informacija. Iako primarno razvijen u akademske svrhe, web je danas centralno mjesto komunikacije i socijalizacije (npr. društvene mreže, platforme za diskusije i instant dopisivanje), poslovanja (digitalne trgovine, internet bankarstvo, poslovne aplikacije) ali i zabave (internet portali, mrežne igrice, platforme za multimediju). Web se pokazao kao pouzdan medij u kriznim situacijama, a trenutačna razmjena informacija kroz društvene mreže omogućila je transparentniju i distribuciju informacija bez cenzure.

Performanse web stranica (web sustava) značajno utječu na korisničko iskustvo, preferencije korisnika i njihovo ponašanje na webu, pogotovo kod korisnika koji webu pristupaju s mobilnih uređaja. Primjerice, ukoliko učitavanje sadržaja potraje više od 3 sekunde 53% korisnika koji pristupaju webu s mobilnog uređaja istu napušta. Brzina učitavanja sadržaja (eng. load time, render time) izravno utječe na efikasnost prikazivanja oglasa (tj. zarade od internet oglasa), korisničko zadržavanje na web mjestu i sklonost napuštanja web mjesta sa prve posjećene stranice (eng. bounce rate) (Shellhammer, 2017).

Ovaj rad donosi osnovne informacije o povijesti i razvoju weba, web sustavima i njihovu kategorizaciju te osnovne informacije o PHP programskom jeziku. Naglasak je stavljen na web sustave i njihove performanse, definiranje metrika, mjerenje performansi te primjenu principa pohrane podataka u privremenu memoriju (eng. *caching*) s ciljem optimizacije web sustava. Praktični dio rada prikazuje optimizaciju fiktivnog web portala razvijenog u PHP programskom jeziku, mjerenje njegovih performansi, identifikaciju uskih grla, optimizaciju te usporedbu performansi prije i nakon optimizacije s primjerima koda.

1. World Wide Web

Internet je često krivo poistovjećen s World Wide Webom (u nastavku rada web) koji je, iako najpopularniji, samo jedan njegov aspekt. Internet podrazumijeva "mrežu međusobno povezanih prostorno distribuiranih računala" (Collins, 2012), dok je web "prostor informacija unutar kojeg se interesni pojmovi nazivaju resursima i identificiraju globalnim oznakama nazvanim Uniform Resource Identifiers (URI)" (Berners-Lee *et al.*, 2004). Web možemo gledati kao uslugu razmjene informacija u obliku dokumenata tj. web stranica. Dakle, web je usluga nastala nad Internetom baš kao što je to i elektronska pošta (eng. email) ili *peer-to-peer* razmjena podataka.

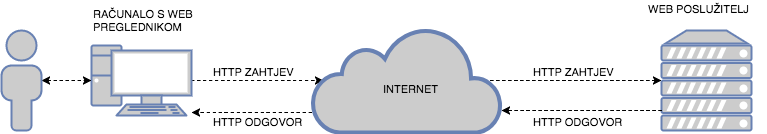
Ovo se poglavlje bavi nastankom i razvojem World Wide Weba, nastankom i razvojem HTTP protokola te kategorizacijom web sustava i web tehnologija.

* 1. Nastanak i razvoj World Wide Weba
     1. HTTP - Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) protokol je aplikacijskog sloja ISO-OSI konceptualnog modela komunikacijskih sustava kojim se najčešće opisuju Internet sustavi. HTTP je definiran kao generički, višestruko primjenjiv, objektno orjentirani protokol bez stanja (eng. stateless). Protokol definira pravila komunikacije distribuiranih sustava za razmjenu informacija odnosno strukturu i semantiku digitalnih poruka (HTTP zahtjeva i odgovora) i temelj je infrastrukture World Wide Weba (Berners-Lee *et al.*, 1997).

Razvoj HTTP-a započeo je Tim Berners-Lee 1989. godine, a danas njegov razvoj nadziru organizacije Internet Engineering Task Force (IETF)[[3]](#footnote-3) i World Wide Web Consortium (W3C)[[4]](#footnote-4). Tim Berners-Lee danas je direktor W3C organizacije. Najraširenija verzija HTTP protokola, verzija HTTP/1.1, prvi je put opisana u dokumentu [RFC 2068](https://tools.ietf.org/html/rfc2068) 1997. godine, a kasnije je zamijenjen dokumentom [RFC 2616](https://tools.ietf.org/html/rfc2616) 1999. godine. Grupa dokumenata [RFC 7230](https://tools.ietf.org/html/rfc7230) (RFC 7230, RFC 7231, RFC 7232, RFC 7233, RFC 7234, RFC 7235) 2014. godine zamijenjuju i nadopunjuju HTTP protokol 2014. godine.

HTTP protokol temelji se na parovima poruka zahtjev i odgovor. Klijent (najčešće web preglednik) generira zahtjev za željeni resurs (npr. web stranicu, multimedijski resurs i sl.) te u zahtjev uključuje različite metapodatke u obliku zaglavlja HTTP zahtjeva (npr. identifikator preglednika, podržane načine enkodiranja sadržaja, upravljačke direktive za kontrolu pohrane podataka u privremenu memoriju i sl.).



Slika : Vizualizacija principa rada HTTP protokola

HTTP zahtjevi koriste različite HTTP metode (npr. GET, POST, HEAD, PUT, DELETE, ...) koje semantički određuju zahtjev, što je posebno važno kod REST web servisa. HTTP odgovor sadrži statusni kod koji semantički određuje status nakon obrade zahtjeva, HTTP zaglavlja koja sadrže dodatne informacije te tijelo poruke (eng. message body) u nekom formatu (npr. HTML, XML, JSON). Standardizirani HTTP statusni kodovi troznamenkasti su brojevi raspoređeni u grupe:

* 1xx – Informativni odgovori
* 2xx – Uspješno obrađeni odgovori
* 3xx – Preusmjerenja (eng. redirections)
* 4xx – Greška na strani klijenta
* 5xx – Greška na strani servera

Brojni se servisi i aplikacije poput web preglednika, web pretraživača i robota za analizu sadržaja oslanjaju na semantiku HTTP statusnih kodova. Primjerice, roboti za indeksiranje sadržaja najpopularnije web tražilice Google različito tretiraju statusne kodove unutar iste grupe statusnih kodova. Tako Google-ovi roboti neće pokušati ponovno dohvatiti sadržaj koji je prethodno poslužen s HTTP statusom 301 (Moved Permanently) već će zapamtiti da je preusmjerenje trajno (permanentno). S druge strane, roboti će redovito ponavljati dohvaćanje sadržaja koji je poslužen s HTTP statusom 302 (Moved Temporarily). Pravilna uporaba i poštivanje semantike HTTP statusnih osigurava bolju poziciju na rezultatima pretrage, smanjuje rizik od negativnog bodovanja ocjene stranice kao i rizik od neočekivanih ponašanja web preglednika.

U prosincu 2014. godine radna skupina IETF organizacije pod nazivom "Hypertext Transfer Protocol working group httpbis" (bis dolazi iz latinskog i znači drugi) predstavila je odboru grupe Internet Engineering Steering Group (IESG)[[5]](#footnote-5) prijedlog prve nove verzije HTTP protokola nakon HTTP 1.1 pod nazivom HTTP/2. IESG je prijedlog odobrio i objavio ga kao predloženi standard 17. veljače 2015. godine. Specifikacija protokola HTTP/2 objavljena je u svibnju 2015. godine u obliku dokumenta RFC 7540. Razvoj HTTP/2 protokola temeljen je na radu Google-ovih inženjera i njihovom eksperimentalnom protokolu SPDY (Belshe, Thomson and Peon, 2014; Nottingham *et al.*, 2015). U kolovozu 2017. godine 16.3% svih web sustava podržava protokol HTTP/2 (W3Techs, 2017b).

Jedna od ključnih novosti koje donosi HTTP/2 protokol jest "Server Push" mehanizam koji omogućava poslužitelju posluživanje resursa koje klijent još nije zatražio, a za koje poslužitelj sigurno zna da će biti zatraženi u nekom od idućih HTTP zahtijeva. Novi je protokol u potpunosti kompatibilan s prethodnom verzijom te je moguće migrirati web sustave koji su do sada radili s protokolom HTTP 1.1. na HTTP/2 bez izmjena u kodu samog sustava. Radna skupina navodi sljedeće ciljeve razvoja novog protokola:

* Razvoj mehanizma pregovaranja koji bi klijentu i serveru omogućio odabir protokola (HTTP 1.1, 2.0 ili drugog protokola)
* Zadržati kompatibilnost s protokolom HTTP 1.1 (HTTP metode, statusne kodove te većinu zaglavlja)
* Optimizirati i poboljšati performanse web sustava kroz uvođenje kompresije HTTP zaglavlja, HTTP/2 "Server Push" mehanizam, ulančavanje HTTP zahtjeva, multipleksiranje više HTTP zahtjeva preko jedne TCP veze i druga poboljšanja.
  + 1. Web i Web 2.0

**@todo**

* 1. Web sustavi

Web sustavima nazivamo aplikacije, platforme, stranice i servise temeljene na HTTP protokolu odnosno one sustave koje resurse distribuiraju kroz web. Ovisno o razini dinamike sadržaja, razni interakcije korisnika sustava sa sustavo te o minimalno potrebnim tehnologijama za razvoj, web sustave dijelimo na web aplikacije, web servise i web stranice. Tablica u nastavku prikazuje osnovnu podjelu web sustava u navedene kategorije dok se detaljnija kategorizacija nalazi u nastavku rada.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Web stranice** | **Web servisi** | **Web aplikacije** |
| **Tehnologije izrade** | HTML, JavaScript, CSS | Programski jezici (PHP, Java, Ruby, C#, JavaScript, ...), online alati | Programski jezici (PHP, Java, Ruby, C#, JavaScript, ...), online alati |
| **Ciljani korisnici** | Ljudi | Programi | Ljudi i programi |
| **Razina moguće interakcije** | Niska | Srednja | Visoka |
| **Primjer** | Prezentacijska web stranica restorana | REST servisi za podataka vremenske prognoze | Društvena mreža |

Tablica : Kategorizacija web sustava

U nastavku poglavlja detaljnije je objašnjena podjela web sustava te razlika između pojedinih kategorija.

* + 1. Web stranice

Web stranice jednosmjerno pružaju statičke informacije. Sadržaj stranica ne mijenja se dinamički već je svaka promjena ručno inicirana (npr. autor web stranice promijeni sadržaj). Web stranice su najraniji i najjednostavniji oblik web sustava kod kojih je korisnička interakcija svedena na minimum – osim pasivnog konzumiranja sadržaja korisnici nemaju mnogo mogućnosti. Takvi su web sustavi uglavnom prezentacijskog karaktera i predstavljaju svojevrsnu digitalnu "posjetnicu" svojih vlasnika. Web stranice su bile iznimno popularne krajem 20. stoljeća. Tehnologije za razvoj statičnih web stranica osnovne su web tehnologije koje bi svaki web programer morao temeljito poznavati (Flanagan, 2011), a služe za prezentaciju statičnih podataka u standardiziranom obliku koji web preglednici znaju prikazati:

* Cascading Style Sheets (CSS)
* Hypertext Markup Language (HTML)
* JavaScript (JS)

Detaljnije o web tehnologijama slijedi u zasebnom poglavlju o web tehnologijama.

* + 1. Web servisi

Web servisima nazivamo one web sustave čiji su primarni korisnici računala i drugi web sustavi. Takvi web sustavi sadržaj ne prezentiraju u vizualnom obliku već u strukturiranom obliku i standardiziranim formatima kao što su XML, JSON i dr.. Web servisi najčešće predstavljaju javno dostupni dio nekog zatvorenog sustava (npr. web servisi za rezervaciju karata avioprjevoznika ili sustavi za plaćanja kreditnim karticama) odnosno sučelje za programsku interakciju web sustava i okoline.

S ciljem povećanja interoperabilnosti različitih web sustava i servisa razvijeni su i standardizirani protokoli poput "Simple Object Access Protocol" (SOAP), "Representational state transfer" (REST) i sl.. Takvi standardi apstrahiraju implementacijske razlike web sustava i određuju protokol komunikacije sustava na njihovim programskim sučeljima (eng. *application programming interface, API*). Web servisi gotovo su beskorisni bez popratne dokumentacije koja opisuje na koji se način web servisi mogu koristiti, koji su prihvatljivi formati poruka i druga ograničenja web servisa.

Početkom 21. stoljeća razvijen je "Web Service Definition Language" (WSDL), jezik za dokumentiranje i opisivanje dostupnih usluga web servisa. Definicija web servisa u WSDL formatu omogućila je razvojnim programerima korištenje širokog spektra različitih web servisa na jednostavniji način – učitavanjem WSDL definicije u razvojno okruženje programer je udaljene web servise koristio na isti način na koji bi koristio funkcije i procedure razvijene unutar svog web sustava (Christensen *et al.*, 2001).

* + 1. Web aplikacije

Web aplikacije najsloženiji su oblik web sustava koji osim distribucije informacija korisniku pružaju dodanu vrijednost kroz svoje funkcionalnosti. Društvene mreže (eng. *social networks*), sustavi za upravljanje sadržajem (eng. *content management system, CMS*),sustavi za rezervaciju i Internet bankarstvo samo su neki od primjera web aplikacija s kojima je prosječni korisnik web-a upoznat. Web aplikacije obično se sastoje od javnog (eng. *public*) dijela koji je javno dostupan te privatnog dijela sustava koji je rezerviran za korisnike s aktivnim korisničkim profilom. Osim autentikacijskog i autorizacijskog sloja, web aplikacije obično implementiraju različite oblike obavijesti korisnika (eng. *notifications*), dvosmjerne kanale komunikacije u obliku razmjena poruka ili diskusije, mogućnost promjene korisničkih postavki i sl..

Web aplikacije često su sastavljene od nekolicine web servisa koji u pozadini prikupljaju podatke, obrađuju ih i pripremaju za prezentaciju i korištenje. Uzmemo li za primjer web sustav PayPal, najpoznatiji sustav za digitalni prijenos novca, možemo identificirati web aplikaciju - sučelja za prijavu, upravljanje računima, plaćanja i sl. te web servise koji na javno objavljenim adresama (eng. *endpoint*) omogućuju programsko korištnje usluga (npr. integraciju plaćanja PayPal-om u druge web aplikacije).

* 1. Web tehnologije
     1. Razvoj web sustava na strani poslužitelja - Backend tehnologije
     2. Razvoj web sustava na strani klijenta - Frontend tehnologije

1. PHP programski jezik
   1. Razvoj PHP programskog jezika

Današnji programski jezik PHP (PHP je rekurzivni akronim od eng. *Hypertext Preprocessor*) naslijednik je jezika PHP/FI (eng. *Personal Home Page/Forms Interpreter*). Rasmus Lerdorf 1994. godine razvio je prvu verziju PHP/FI jezika kako bi javno objavio, i pratio posjete svom digitalnom životopisu te je skupinu programskih skripti grupirao pod nazivom "*Personal Home Page Tools*" ili skraćeno "*PHP Tools*". Lerdorf je nastavio s razvojem svojih skripti te je u lipnju 1995. javno objavio izvorni kod prve verzije PHP-a s ciljem popularizacije novostvorenog programskog jezika, ali i suradnje te poboljšanja koje bi eventualno drugi programeri izradili (The PHP Group, 2017a). Sam Lerdorf izjavio je kako u svemu tome nije planirao razvoj novog programskog jezika:

"*Nisam znao kako stati, nije bilo namjere razviti novi pgorgramski jezik... Niti ne znam kako razviti programski jezik, samo sam dodavao elemente koji su mi se logički činili potrebnima*" (IT Conversations, 2003).

PHP je već u svojim prvim godinama doživio nekoliko velikih revizija i izmjena iz temelja, ubrzo su dodane funkcionalnosti potrebne za komunikaciju s bazom podataka, a već krajem 1998. godine PHP je brojao nekoliko tisuća korisnika u svijetu. Istraživanje Netcraft-a pokazalo je kako je u svibnju 1998. godine više od 60 000 ili 1% ukupnog broja web stranica na svijetu bilo pokretano na poslužiteljima koji podržavaju PHP (The PHP Group, 2017a).

PHP verzija 3.0 najbliža je PHP-u kakvog danas poznajemo. Programeri Andi Gutmans i Zeev Suraski radili su na razvoju sustava elektronske trgovine (eng. *ecommerce*) baziranog na PHP-u te su uočili brojne nedostatke prethodne verzije jezika PHP/FI 2.0. Kontaktirali su Lerdorfa i digitalnim kanalima raspravljali o poboljšanjima na kojima su Gutmans i Suraski već radili. Konačno, dogovorena je suradnja trojice programera te su zajedno započeli rad na razvoju novog i neovisnog programskog jezika pod novim nazivom "*PHP: Hypertext Preprocessor*" kako bi se rješili ograničavajućeg prizvuka korištenja isključivo za osobne potrebe. PHP 3.0 privukao je brojne programere koji dali svoj doprinos razvojem različitih programskih modula. Na vrhuncu popularnosti verziju PHP 3.0 podržavalo je približno 10% svih web poslužitelja na svijetu (The PHP Group, 2017a).

U zimu 1998. godine, nedugo nakon službene objave PHP 3.0, Gutmans i Suraski započeli su rad na poboljšanju jezgre PHP-a s ciljem optimizacije performansi kompleksnih aplikacija te poboljšanje modularnosti izvornog koda PHP-a. Razvili su novi sustav koji je pokretao PHP (eng. engine) i nazvali ga "*Zend Engine*" (prema kombinaciji imena autora, Zeev i Andi). PHP 4.0 temeljen na novom sustavu "*Zend Engine*" objavljen je u svibnju 2000. godine a nova je verzija podržavala različite web poslužitelje, HTTP sesije, nove programske konstrukte te poboljšane sigurnosne mehanizme (The PHP Group, 2017a).

PHP 5 objavljen je u srpnju 2004. godine nakon nekoliko iteracija razvoja i objavljenih pred-verzija. Nova je verzija PHP-a bila pokretana sustavom "*Zend Engine 2.0*" (The PHP Group, 2017a). Nova je verzija programskog jezika donijela brojne funkcionalnosti i poboljšala podršku za objektno-orjentirano programiranje u PHP-u, "PHP Data Objects" (PDO) ekstenzija omogućila je jednostavnije i sigurnije pristupanje bazama podataka a performansna poboljšanja omogućila su razvoj kompleksnih web sustava u PHP-u (Trachtenberg, 2004). U kolovozu 2014. godine objavljena je verzija PHP 5.6 koja je donijela značajna poboljšanja i pristupačniju sintaksu a u kolovozu 2017. godine PHP 5.6 je najkorištenija verzija PHP programskog jezika s gotovo 30% tržišnog udjela (W3Techs, 2017a).

PHP je često bio na meti kritika zbog nedostatka podrške za *unicode* način enkodiranja tekstualnih podataka (PHP je podržavao samo tzv. *byte strings*). Andrei Mievski započeo je rad na uvođenju nativne podrške za *unicode* način enkodiranja 2005. godine kroz ugradnju "International Components for Unicode" (ICU) biblioteke i enkodiranje tekstualnih podataka UTF-16 načinom kodiranja. Značajnost promjene koju bi objava nove verzije programskog jezika donijela nalagala je da se izmjena uvede kroz novu verziju programskog jezika – PHP 6 (tzv. *major release*). Nedostatak programera koji su prepoznali važnost predloženih promjena i performansni problemi nastali prilikom promjene načina kodiranja tekstualnih podataka doveli su do zastoja u razvoju što je rezultiralo objavom verzije PHP 5.3 2009. godine koja je donijela funkcionalnosti prethodno planirane za objavu u sklopu verzije PHP 6. Razvoj verzije PHP 6 napušten je u ožujku 2010. godine a verzija PHP 5.4 donijela je preostale funkcionalnosti planirane za napuštenu verziju (Zmievski, 2005, 2011).

Tijekom 2014. i 2015. godine razvijena je nova značajna (eng. *major*) verzija PHP programskog jezika – PHP 7. Imenovanje nove verzije uzrokovalo je brojne debate. Postojale su dvije struje, jedna koja se zalagala za objavu nove verzije pod nazivom PHP 6 budući da nije postojala službena verzija s tim nazivom i druga koja se zalagala za preskakanje verzije i objavu nove verzije programskog jezika pod nazivom PHP 7. Radna verzija PHP 6 iako nikada nije bila javno objavljena, bila je dostupna u repozitoriju koda i poznata (i očekivana) u programerskim krugovima. Zanimljiva je činjenica da su izdavači stručnih knjiga već objavili nekoliko izdanja koja su referencirala PHP verziju 6, što bi objavom nove verzije pod nazivom PHP 6 stvorilo nesporazum. Na kraju je za službeni naziv verzije odabran PHP 7 a diskusija oko konačne odluke traje i danas (Sturgeon, 2014). Tim programera Dmitry Stogov, Xinchen Hui i Nikita Popov razvili su novu verziju programskog jezika sa značajno poboljšanim performansama koje su postigli doradom sustava "Zend Engine". Performnansa mjerenja (eng. *benchmark*) pokazala su 100% poboljšanje performansi u novoj verziji PHP 7 u odnosu na prethodnu verziju. Izmjene u sustavu "Zend Engine" objavljene su pod nazivom "Zend Engine 3" koji je naslijedio "Zend Engine 2" korišten u verziji PHP 5. Osim performansnih poboljšanja i dorada izvornog koda, PHP 7 donio je nove programske konstrukte u PHP programski jezik kao što su definiranje povratnog tipa podataka funkcije i definiranje tipa podataka argumenta funkcija (Stogov and Suraski, 2014).

U nastavku je napravljen pregled do sada objavljenih verzija programskog jezika PHP s kratkim opisom novosti pojedine verzije te stanjem podrške za pojedinu verziju.

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum objave** | **Podrška do** | **Opis** |
| ***Legenda****: ■ Neaktivno izdanje, ■ Stabilno izdanje s doradama sigurnosnih propusta, ■ stabilno izdanje s doradama bugova i sigurnosnih propusta, ■ planirano (buduće izdanje)* | | | |
| 1.0 ■ | 8. lipnja 1995. | *Nema podataka* | Prvi puta korišten naziv PHP – službeni naziv: "Personal Home Page Tools (PHP Tools)" |
| 2.0 ■ | 1. studenoga 1997. | *Nema podataka* | Službeni naziv "PHP/FI 2.0". Prva verzija samostalnog programskog jezika u punom smislu, s funkcionalnostima od kojih su neke zadržane do danas |
| 3.0 ■ | 6. lipnja 1998. | 20. listopada 2000. | Proširenje razvojnog tima - Zeev Suraski i Andi Gutmans dorađuju temelje nove verzije |
| 4.0 ■ | 22. svibnja 2000. | 23. lipnja 2001. | Dodan je sustav obrade i izvođenja – tzv. Zend engine |
| 4.1 ■ | 10. prosinca 2001. | 12. ožujka 2002. | Uvedene su superglobalne varijable |
| 4.2 ■ | 22. travnja 2002. | 6. rujna 2002. | Sigurnosna poboljšanja |
| 4.3 ■ | 27. prosinca 2002. | 31. ožujka 2005. | Uvedeno je konzolno sučelje CLI |
| 4.4 ■ | 11. srpnja 2005. | 7. kolovoza 2008. | Ispravljen je problem u radu s memorijom |
| 5.0 ■ | 13. srpnja 2004. | 5. rujna 2005. | Uveden je sustav Zend Engine 2 s novim objektnim modelom |
| 5.1 ■ | 24. studenoga 2005. | 24. kolovoza 2006. | Performansna poboljšanja, uvođenje PHP Data Objekata (PDO) kao konzistentnog sučelja za rad s bazama podataka |
| 5.2 ■ | 2. studenoga 2006. | 6. siječnja 2011. | Nativna podrška za JSON |
| 5.3 ■ | 30. lipnja 2009. | 14. kolovoza 2014. | Podrška za *namespace* kasno statičko povezivanje (eng.  *late static binding*), anonimne funkcije, PHP arhive i druge dorade. |
| 5.4 ■ | 1. ožujka 2012. | 3. rujna 2015. | Podrška za *trait* mehanizam, sintaksne dorade, ugrađeni web poslužitelj, performansna poboljšanja. |
| 5.5 ■ | 20. lipnja 2013. | 21. srpnja 2016. | Uvedeni su generatori, poboljšano je upravljanje iznimkama |
| 5.6 ■ | 28. kolovoza 2014. | 31. prosinca 2018. | Poboljšane konstante, funkcije s neograničenim brojem argumenata, operator potenciranja, novi razvojni alati i druge dorade. |
| 6.x | *Nije objavljeno* |  | Napuštena verzija koja je trebala uvesti podršku za unicode način kodiranja |
| 7.0 ■ | 3. prosinca 2015. | 3. prosinca 2018. | Uveden je novi sustav Zend Engine 3 sa značajnim poboljšanjem performansi, novom sintaksom, novim operatorima i drugim doradama. |
| 7.1 ■ | 1. prosinca 2016. | 1. prosinca 2019. | Uvođenje nedefiniranog (eng. *void*) povratnog tipa, bolje upravljanje iznimkama i druge dorade. |
| 7.2 ■ | 30. studenoga 2017 | 30. studenoga 2020. |  |

Tablica : Povijest verzija programskog jezika PHP prema (The PHP Group, 2017b, 2017c).

* 1. Razvoj suvremenih web sustava u PHP programskom jeziku
     1. Razvojni ekosustav

Frameworks, composer, git, server support

* + 1. On-line zajednica
    2. Kritike programskog jezika PHP

1. Performanse web sustava

Osnovna je svrha svih web sustava zaprimanje i obrada HTTP zahtjeva te generiranje HTTP odgovora, odnosno posluživanje resursa koji su bili zathjevani. Brzina kojom web sustav može obraditi korisnički zahtjev najvažnija je njegova karakteristika. Web sustav koji brže obrađuje zahtjeve može podnijeti veći promet i ne zahtijeva značajnija ulaganja u infrastrukturu. Ukoliko web sustav doživi ekspanziju i prosječan se broj korisničkih zahtjeva u minuti (eng. *request per minute, rpm*) značajno poveća, potrebno je pratiti i optimizirati sustav, ukloniti uska grla, poboljšati infrastrukturu i na druge načine poboljšati performanse kako bi web sustav mogao služiti svojoj svrsi. Navedene aktivnosti nazivamo skaliranjem web sustava, a kako bi se performanse web sustava mogle poboljšati, prvo ih je potrebno identificirati i izmjeriti.

U nastavku poglavlja postavljena je ljestvica za kategorizaciju web sustava prema posjećenosti, opisane su performansne karakteristike web sustava uz primjere metrika za svaku o performansnih karakteristika.

* 1. Web sustavi visokih performansi

Za razliku od aplikacija na stolnim računalima (eng. *desktop applications*) koji su smješteni (instalirani) na korisničkom računalu, web sustavi podatke prenose s (često i stotinama kilometara) udaljenih poslužitelja. Korisnici ne mare previše za infrastrukturne karakteristike i razlike sustava koje koriste, zanima ih ono što vide odnosno "osjete" prilikom korištenja, stoga su metrike za kvalitetu web sustava jednake kao i za sve druge računalne sustave.

Ukoliko računalni sustav na korisnički zahtjev odgovori unutar 100 milisekundi, kod korisnika je stvoren dojam "instantnosti" – rezultat je prikazan u vremenu u kojem korisnik nije stigao osjetiti da se nešto događa odnosno nije primjećen zastoj rada sustava. Ukoliko korisnik čeka na odgovor sustava više od 1 sekunde, on gubi osjećaj izravnog rada s podacima i osjeti usporenje u radu sustava ali i dalje zadržava fokus. Sustavi kojima je potrebno 10 sekundi ili više za generiranje odgovora na korisničku interakciju ne mogu računati s koncentracijom korisnika – u tom vremenu korisnici se obično okreću drugim zadacima dok čekaju da sustav završi obradu. Takvi sustavi obavezno moraju korisnicima knstantno pružati povratnu informaciju o napretku (eng. *progress*) obrade (npr. instalacija zahtjevnih računalnih programa i traka napretka koja prikazuje postotke dovršenosti.) (Miller, 1968).

Autori web sustava moraju uzeti u obzir nekoliko performansno nepovoljnih okolnosti: vrijeme prijenosa podataka korisnik percipira kao vrijeme obrade, korisnici često pristupaju web sustavima s mobilnih ili drugih uređaja s ograničenom brzinom pristupa Internetu, svi resursi (multimedijske datoteke i druge datoteke, artefakti) putuju prema korisniku kroz mrežu i vrijeme njihovog prijenosa ne smije se zanemariti, više korisnika može istovremeno pristupati web sustavu i na taj ga način opteretiti. Imajući u vidu navedena ograničenja pažljivi čitatelj može zaključiti da web sustavi često moraju raditi mnogo brže kako bi kompenzirali nedostatke i ograničenu brzinu prijenosa podataka.

Čak i kada su svi podaci uspješno preneseni na korisnikovo računalo, web sustav nije završio s radom – reprezentacija pristiglih podataka, dohvaćanje povezanih resursa kao i obrada na strani klijenta samo su neke od operacija koje web preglednici obavljaju nakon što su podaci uspješno preneseni. Gotovo 80-90% (Souders, 2007) posla otpada na obradu i prezentaciju podataka na strani klijenta (tzv. frontend) što ostavlja vrlo malo prostora za optimizacijski manevar na strani poslužitelja (tzv. backend).

Iako se većina optimizacija web sustava svodi na minimiziranje podataka koji se prenose kroz mrežu (pohranom u privremenu memoriju, minifikacijom i kompresijom datoteka ili drugim tehnikama), optimizacija na strani poslužitelja važna je zbog smanjenja troškova i ulaganja u infrastrukturu te potrošnje energije (Souders, 2007).

* 1. Performansne karakteristike web sustava

Performansne su karakteristike web sustava one osobine koje opisuju kojom brzinom web sustav može obraditi određenu količinu zahtjeva, koju količinu prometa web sustav može podnijeti prije nego postane nedostupan te koliko je jednostavno (ili kompleksno) sustav skalirati za nekoliko redova veličine prometa.

* + 1. Propusnost web sustava
    2. Vrijeme odziva web sustava
    3. Dostupnost web sustava
  1. Testiranje performansi web sustava
     1. Testiranje web sustava pod opterećenjem
     2. Testiranje granica web sustava
     3. Regresijsko testiranje web sustava

1. Razvoj web sustava visokih performansi
   1. Tijek razvoja web sustava visokih performansi
   2. Mjerenje i optimizacija performansi
      1. Optimizacija aplikacijskog sloja web sustava
         1. Pronalazak uskih grla i optimizacija koda
         2. Pohrana podataka u privremenu memoriju
      2. Optimizacija podatkovnog sloja web sustava
         1. Optimizacija upita prema bazi podataka
         2. Prilagodba strukture baze podataka s ciljem optimizacije sustava
         3. Pohrana rezultata u privremenu memoriju
      3. Optimizacija infrastrukturnog sloja web sustava
         1. HTTP zaglavlja i pohrana podataka u privremenu memoriju
         2. Kompresija podataka i pohrana podataka u privremenu memoriju preglednika
2. Praktičan rad – web sustav visokih performansi
   1. Opis web sustava
      1. Struktura web sustava
      2. Infrastruktura
      3. Model baze podataka
      4. Koraci razvoja i optimizacije web sustava
   2. Optimizacija i mjerenje performansi web sustava
      1. Bez korištenja privremene memorije i indeksa u bazi podataka
      2. Uz korištenje indeksa u bazi podataka
      3. Uz korištenje pohrane podataka u privremenu memoriju na strani aplikacije
      4. Uz korištenje pohrane rezultata u privremenu memoriju na strani HTTP servera za privremenu pohranu
3. Literatura

Belshe, M., Thomson, M. and Peon, R. (2014) ‘Hypertext Transfer Protocol Version 2 (HTTP/2)’. Available at: https://tools.ietf.org/html/rfc7540 (Accessed: 20 August 2017).

Berners-Lee, T. *et al.* (1997) ‘RFC2068 - Hypertext Transfer Protocol -- HTTP/1.1’. Available at: https://tools.ietf.org/html/rfc2068 (Accessed: 18 August 2017).

Berners-Lee, T. *et al.* (2004) ‘Architecture of the World Wide Web, Volume One’, *W3C Recommendation 15 December 2004*. Available at: https://www.w3.org/TR/webarch/ (Accessed: 20 August 2017).

Christensen, E. *et al.* (2001) *Web Service Definition Language (WSDL)*. Available at: https://www.w3.org/TR/wsdl (Accessed: 20 August 2017).

Collins, W. (2012) *Collins English Dictionary - Complete & Unabridged 2012 Digital Edition*. HarperCollins Publishers.

Flanagan, D. (2011) *JavaScript: The Definitive Guide, 6th Edition - O’Reilly Media*. 6th edn. O’Reilly Media. Available at: http://shop.oreilly.com/product/9780596805531.do (Accessed: 20 August 2017).

Internet World Stats (2017) *World Internet Users Statistics and 2017 World Population Stats*. Available at: http://www.internetworldstats.com/stats.htm (Accessed: 18 August 2017).

IT Conversations (2003) ‘Rasmus Lerdorf, Senior Technical Yahoo: PHP, Behind the Mic’. Available at: https://web.archive.org/web/20130728125152/http://itc.conversationsnetwork.org/shows/detail58.html (Accessed: 19 August 2017).

Miller, R. B. (1968) ‘Response time in man-computer conversational transactions’, in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS ’68 (Fall, part I)*. New York, New York, USA: ACM Press, p. 267. doi: 10.1145/1476589.1476628.

Nottingham, M. *et al.* (2015) *Hypertext Transfer Protocol (httpbis)*, *IETF Datatracker*.

Poushter, J. (2016) ‘Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies’, *Pew Research Center*, pp. 1–45. doi: 10.1017/CBO9781107415324.004.

Shellhammer, A. (2017) *The need for mobile speed: How mobile latency impacts publisher revenue*. Available at: https://www.doubleclickbygoogle.com/articles/mobile-speed-matters/ (Accessed: 18 August 2017).

Souders, S. (2007) *14 Rules for Faster-Loading Web Sites*. Available at: http://stevesouders.com/hpws/ (Accessed: 20 August 2017).

Stogov, D. and Suraski, Z. (2014) *PHP RFC: Move the phpng branch into master*. Available at: https://wiki.php.net/rfc/phpng (Accessed: 20 August 2017).

Sturgeon, P. (2014) *The Neverending Muppet Debate of PHP 6 v PHP 7 | Phil Sturgeon*. Available at: https://philsturgeon.uk/php/2014/07/23/neverending-muppet-debate-of-php-6-v-php-7/ (Accessed: 20 August 2017).

The PHP Group (2017a) *History of PHP and Related Projects*. Available at: http://php.net/manual/en/history.php.php (Accessed: 19 August 2017).

The PHP Group (2017b) *PHP: Supported Versions*. Available at: http://php.net/supported-versions.php (Accessed: 20 August 2017).

The PHP Group (2017c) *PHP: Unsupported Branches*. Available at: http://php.net/eol.php (Accessed: 20 August 2017).

Trachtenberg, A. (2004) *Upgrading to PHP 5*. O’Reilly Media. Available at: http://shop.oreilly.com/product/9780596006365.do (Accessed: 20 August 2017).

VPN Mentor (2017) *Internet Trends 2017. Stats & Facts in the U.S. and Worldwide*. Available at: https://www.vpnmentor.com/blog/vital-internet-trends/ (Accessed: 18 August 2017).

W3Techs (2017a) *Historical trends in the usage of PHP version 5, August 2017*. Available at: https://w3techs.com/technologies/history\_details/pl-php/5 (Accessed: 20 August 2017).

W3Techs (2017b) *Usage Statistics of HTTP/2 for Websites, August 2017*. Available at: https://w3techs.com/technologies/details/ce-http2/all/all (Accessed: 20 August 2017).

Zmievski, A. (2005) *[PHP-DEV] PHP Unicode support design document*. Available at: https://marc.info/?l=php-internals&m=112365908921757&w=1 (Accessed: 20 August 2017).

Zmievski, A. (2011) *The Good, the Bad, and the Ugly: What Happened to Unicode and PHP 6*. Available at: https://www.slideshare.net/andreizm/the-good-the-bad-and-the-ugly-what-happened-to-unicode-and-php-6 (Accessed: 20 August 2017).

1. Izvorni kod praktičnog primjera

1. <https://home.cern/topics/birth-web> Službeni podaci o rođenju weba [↑](#footnote-ref-1)
2. <http://info.cern.ch/hypertext/WWW/TheProject.html> Prva web stranica koju je Tim Berners-Lee objavio 1990. godine [↑](#footnote-ref-2)
3. <https://www.ietf.org/> - Internet Engineering Task Force organizacija [↑](#footnote-ref-3)
4. <https://www.w3.org/> - World Wide Web Consortium (W3C) konzorcij [↑](#footnote-ref-4)
5. <https://www.ietf.org/iesg/> - službena web stranica IESG grupe zadužene za tehničko upravljanje aktivnosti IETF te donošenje Internet standarda. [↑](#footnote-ref-5)