

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI SEMINAR

**Postupci uravnoteživanja
opterećenja u raspodijeljenim
sustavima**

Dario Sindičić

Voditelj: izv. prof. dr. sc. Dejan Škvorc

Zagreb, svibanj 2019.

SADRŽAJ

Popis slika	iii
Popis tablica	iii
1. Uvod	1
2. Pregled metoda	2
3. Metode	3
3.1. Korisnički način	3
3.2. DNS način	4
3.2.1. TTL vrijednost	4
3.3. Pristup na temelju dispečera	5
3.3.1. Statička pravila	5
3.3.2. Dinamička pravila	6
3.4. Poslužiteljski način	6
3.4.1. HTTP preusmjerenje	7
3.4.2. Usmjeravanje paketa	8
4. Usporedbe	9
5. Budući razvoj	12
6. Zaključak	13
7. Literatura	14
8. Sažetak	16

POPIS SLIKA

3.1. Uravnoteživanje korisničkim načinom	3
3.2. Dispečerski način uravnoteživanja	5
3.3. Algoritam kružnog dodjeljivanja, 1621. (1)	6
3.4. HTTP preusmjeravanje kod poslužiteljskog načina	7
3.5. Usmjeravanje paketa kod poslužiteljskog načina	8
4.1. Performanse različitih metoda za uravnoteživanje opterećenja	11

POPIS TABLICA

4.1. Prikaz karakteristika pojedinih metoda uravnoteživanja opterećenja . .	9
---	---

1. Uvod

Internet je od svog osnutka 70-tih godina prošlog stoljeća narastao do nezamislivih granica i današnji dnevni promet lako prelazi 4 bilijuna GiB (2). Uz rast dnevnog prometa, krajnji korisnici žele doći do svog sadržaja sve brže i brže. To sve stvara veliko opterećenje na poslužitelje koji bi primali tako veliki promet, stoga je potrebno osmisliti arhitekturu koja će podnijeti današnju količinu prometa. Kako bi se riješio zadani problem, jedan od načina njegovog rješavanja je uvođenje repliciranih poslužitelja koji obrađuju korisničke zahtjeve. Replicirani poslužitelji pružaju korisniku jednaku uslugu, a njihove razlike mogu biti u arhitekturi, operacijskom sustavu ili čak u državi gdje su smješteni. Uvođenjem više poslužitelja postavlja se pitanje pravilnog raspoređivanja opterećenja na svakoga od njih. Ako u raspodjeli opterećenja (engl. *load balancing*) neki poslužitelji nisu primili predviđenu količinu prometa postavlja se pitanje njihove isplativosti. U ovom seminaru će se prezentirati načini raspodjele prometa na više repliciranih poslužitelja s ciljem što ujednačenije preraspodjele opterećenja. Drugo poglavlje bavi se metodologijom načina uravnoteživanja opterećenja te njihovom analizom kroz literaturu, dok se u trećem poglavlju pobliže opisuje svaki od predloženih načina. Četvrto poglavlje prezentira pregled svih metoda te njihovu međusobnu usporedbu. Zaključno peto poglavlje pruža uvid u budući razvoj metoda u tom području.

2. Pregled metoda

Tema uravnoteživanja opterećenja aktualna je još od 1980-tih. Tada se nije govorilo o uravnoteživanju opterećenja u web sustavima nego o najboljim načinima organiziranja poslova u paralelnim sustavima. Rad (3) je jedan od prvih radova iz prethodno navedenog područja. On je prvi uveo dinamična pravila uravnoteživanja te navodi da su ona poželjna kada vrijeme potrebno za završetak jednog zahtjeva nije unaprijed poznato. Rad (4) predlaže četiri glavne metode uravnoteživanja opterećenja u raspodijeljenim sustavima.

- Korisnički način
- DNS način
- Pristup na temelju dispečera (engl. *dispatcher-based approach*)
- Poslužiteljski način

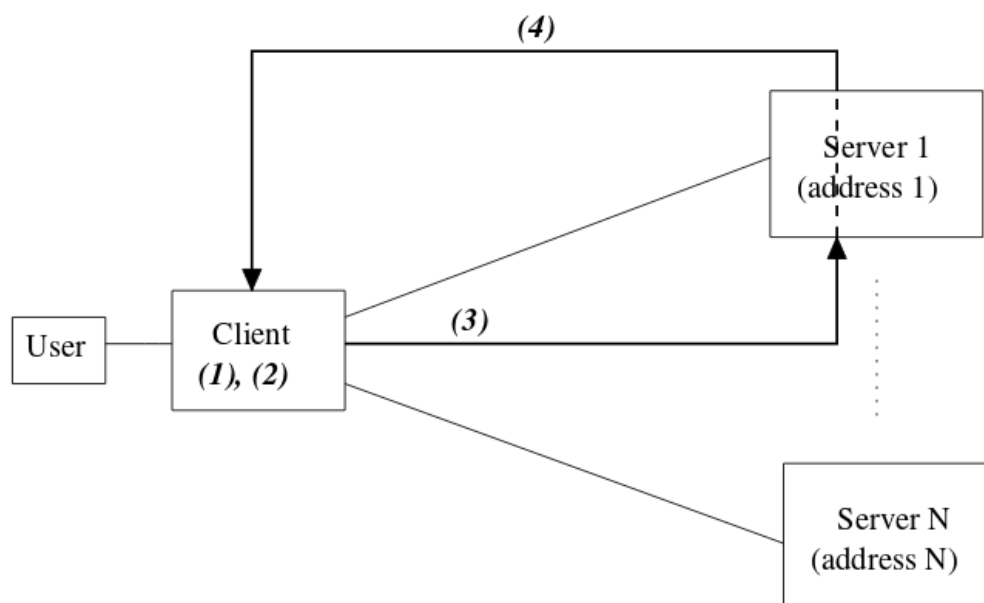
U industriji se vrlo često primjenjuju hibridna rješenja gdje su poslužiteljski, korisnički i dispečer spojeni u jedno rješenje. Takva rješenja daju još bolje rezultate jer kod svake od navedenih metoda, uravnoteživanje opterećenja se provodi na drugačijem sloju. Dok radovi (5) i (6) uvode samo dva načina koja su podvrsta dispečer pristupa, a to su statička i dinamička pravila raspoređivanja. Statičke metode kao algoritam kružnog dodjeljivanja (engl. *round-robin*), težinski algoritam kružnog dodjeljivanja (engl. *weighted round-robin*) i najmanji broj konekcija već su poznate od početka rješavanja tog problema stoga ta dva rada su veću pažnju posvetila dinamičkim metodama koje se mogu prilagoditi različitim uvjetima. Kompanija F5 Networks (7) također u svojoj dokumentaciji navodi statičke i dinamičke metode uravnoteživanja. Rad (8) predlaže uvođenje uravnoteživanja opterećenja već kod korisnikove strane te su takvim pristupom dobili najbolje performanse u promjenjivom sustavu jer se uravnoteživanje odvija kod korisnikove strane.

3. Metode

Treće poglavlje donosi pregled četiri glavne metode iznesenih u radu (4) te donosi njihove varijacije u implementaciji te arhitekturi raspodijeljenog sustava.

3.1. Korisnički način

Korisnički način je jedan od najjednostavnijih i on se ovdje navodi radi potpunosti. On uključuje da se zahtjevi već na korisnikovoj strani raspodijele prema više poslužitelja. Rad (4) navodi da se ovaj način može koristiti ako korisnik zna za postojanje repliciranih poslužitelja te tada on odmah sa svoje strane radi uravnoteživanje opterećenja poslužitelja. Slika 3.1 prikazuje korisnički način uravnoteživanja opterećenja. Korisnik može birati između N poslužitelja te onda zahtjev šalje prema njemu te od njega dobiva odgovor.



Slika 3.1: Uravnoteživanje korisničkim načinom

Netscape pristup (9) (engl. *Netscape's approach*) način pristupanja Netscape komunikacijskoj stranici preko Netscape-ovog preglednika je bio prvi primjer korisničkog načina uravnoteživanja prometa. Ovakav način nije generalno primjenjiv kako veliki broj stranica koje nema vlastiti preglednik koji bi služio distribuciji zahtjeva.

Posrednički poslužitelji (engl. *proxy server*) je još jedan dio strukture koja može pomoći u distribuciji korisničkih zahtjeva prema poslužiteljima. Posrednički poslužitelj spada pod korisnički način jer je i dalje po funkcioniranju jako sličan. Rad (10) predstavlja zanimljiv način kojim se kombinira skladištenje (engl. *caching*) i replikacija poslužitelja. Klijentski posrednički poslužitelj je prva razina skladištenja te distribuira promet na poslužitelje gornjeg nivoa te tako dalje ovisno o broj nivoa.

3.2. DNS način

DNS (engl. *Domain name system*) je servis koji služi prevođenju odnosno mapiranju alfa-numeričkih naziva u IP adresu računala (engl. *forward lookup*). U ovom slučaju zadatak uravnoteživanja preuzima DNS poslužitelj koji za isti alfa-numerički naziv zamjenjuje sa različitim poslužitelja. Njegova je zadaća da na neki način odabere neki poslužitelj te da na DNS zahtjeve odgovara s IP adresom odabranog poslužitelja te nakon nekog vremena mijenja poslužitelj. Potrebno je primijetiti da DNS ima vrlo malu kontrolu kod uravnoteživanja upravo zbog skladištenja (engl. *cache*) DNS odgovora sa korisničke strane, stoga se sljedeći dio se bavi TTL vrijednosti koja bi trebala dati veću kontrolu nad skladištenjem DNS odgovora.

3.2.1. TTL vrijednost

TTL (engl. *time to live*) je vremenski period koji se izražava u sekundama, a predstavlja period koliko bi dugo trebao biti DNS rezultati spremljeni za daljnje korištenje. TTL vrijednost može biti statična, unaprijed definirana ili može biti dinamična.

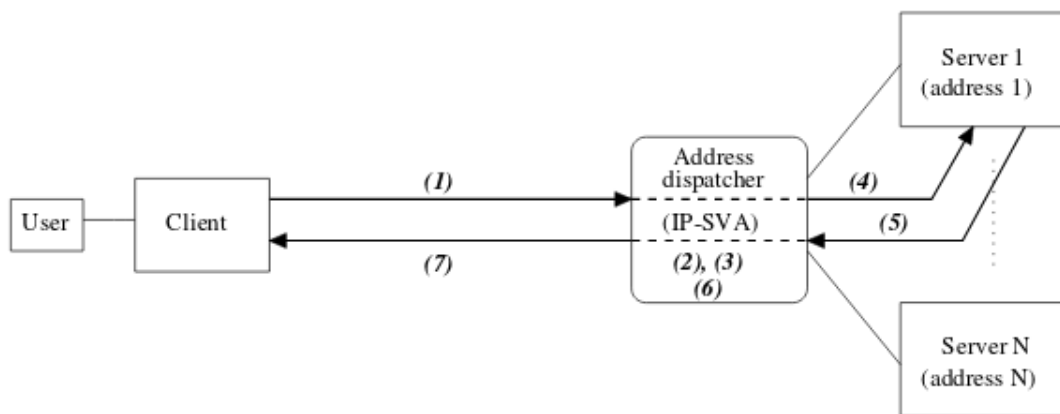
Dinamičan TTL

Kako bi balansirao promet na više web poslužitelja, DNS način ima mogućnost podešavanja politike postavljanja poslužitelja koji će biti u DNS odgovoru te odabir TTL vrijednost zapisa. Rad (11) je predložio dinamično mijenjanje TTL vrijednosti. U tom radu je pokazano da ipak smanjenje TTL vrijednosti kako bi se dobila veća disperzija zahtjeva nije najbolja opcija, nego je svakom zahtjevu potrebno postaviti drugu TTL

vrijednost. Kao način odabira poslužitelja najboljim se pokazao algoritam kružnog dodjeljivanja.

3.3. Pristup na temelju dispečera

Pristup na temelju dispečera (engl. *dispathcer*) je drugačiji od prethodna dva pristupa. U prošlim pristupima se od web poslužitelja moglo dolaziti preko njihovih javnih IP adresa, no u ovom slučaju samo jedan poslužitelj ili dispečer ima javnu adresu dok ostali koji poslužuju sadržaj imaju privatne adrese koje nisu dostupne vanjskim korisnicima. Dispečer ima popis privatnih adresa poslužitelja s kojima se nalazi u mreži i on njima prosljeđuje promet. Takav princip je prikazan na slici 3.2.



Slika 3.2: Dispečerski način uravnotežavanja

U tom slučaju se radi modifikacija adrese odredišta u adresu poslužitelja koga je odabrao dispečer, pri čemu dispečer provodi NAT postupak (engl. *Network address translation*). Naravno, moguća je i drugačija arhitektura. Ako svaki poslužitelj ima javnu IP adresu, tada korak (5) sa slike 3.2 ne ide iz poslužitelja 1 prema dispečer nego direktno prema korisniku. Dispečer periodično šalje ICMP poruke poslužiteljima da ustanovi koji su od njih trenutno aktivni te samo njima prosljeđuje zahtjeve.

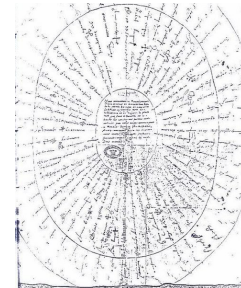
3.3.1. Statička pravila

Statička pravila se podešavaju kod inicijalizacije dispečera te ostaju trajna cijelo vrijeme dok dispečer radi. Statičkih pravila ima mnogo, no samo neka od njih imaju veću primjenu u industriji kao što su :

- algoritam kružnoga dodjeljivanja

- broj aktivnih veza
- slučajno

Algoritam kružnog dodjeljivanja (12) (franc. *ruban rond*) je tehnika koja potječe iz 17 stoljeća. Takozvano round-robin pismo (engl. *round robin letter*) predstavljalo je potpise velikog broja ljudi za određenu peticiju o kružnom obliku kako se na kraju ne bi znalo tko je pokrenuo potpisivanje. Algoritam kružnog dodjeljivanja preusmjerava promet naizmjenično s prvog prema drugom, ..., $N-1$ prema N -tom poslužitelju i tako ukруг.



Slika 3.3: Algoritam kružnog dodjeljivanja, 1621. (1)

Broj aktivnih veza označuje da će dispečer proslijediti korisnički zahtjev prema poslužitelju s najmanje trenutno aktivnih veza te slučajno označuje da dispečer ne zna ništa o poslužiteljima te slučajno odabire poslužitelj koji će obraditi taj zahtjev.

3.3.2. Dinamička pravila

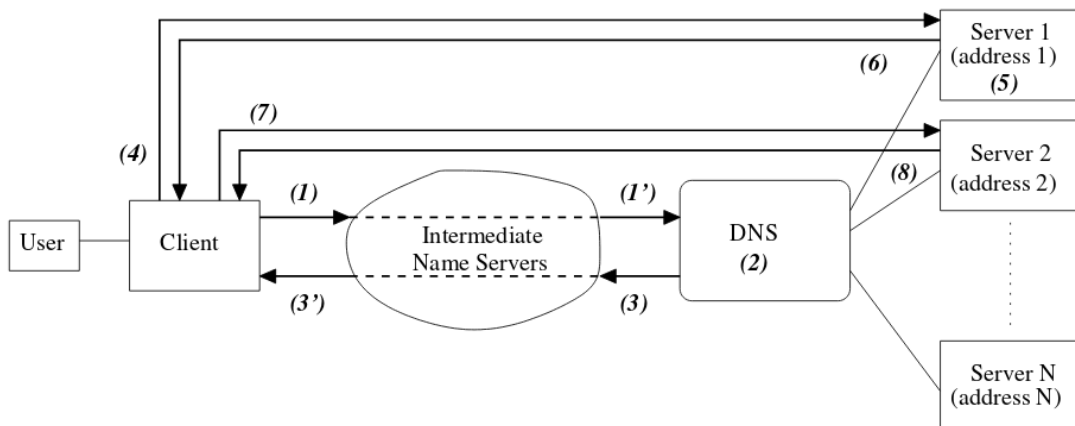
Dinamička pravila označavaju da se politika prosljeđivanja zahtjeva prema poslužiteljima mijenja. Pokazuje se potreba za time jer se stanje sustava mijenja kroz vrijeme pa tako i pogodna politika razmještanja. Tehnika za mijenjanje politika ima mnogo, no svoj procvat doživljavaju 2016. pojavom strojnog učenja (engl. *machine learning*). Potrebno je naglasiti da dinamička pravila zahtjevaju od dispečera da konstantno prima stanja poslužitelja što može dovesti do mrežne zalihosti te što složenija pravila koristi dispečer, to mu je više vremena potrebno da proslijedi zahtjev prema određenom poslužitelju što dovodi do povećanja RTT. Takve metode će se pobliže pojasniti u petom poglavlju.

3.4. Poslužiteljski način

Poslužiteljski način koristi dvije razine uravnoteživanja opterećenja. Korisnički zahtjevi se prvo prosljeđuju prema poslužitelju kojeg je odredio DNS server iz DNS grupe (engl. *DNS cluster*) kao u DNS načinu. Nakon toga svaki poslužitelj može korisnički zahtjev preusmjeriti prema nekom drugom poslužitelju iz grupe ako nije u mogućnosti na njega odgovoriti. Odluka poslužiti zahtjev ili ga proslijediti može ovisiti o mnogo

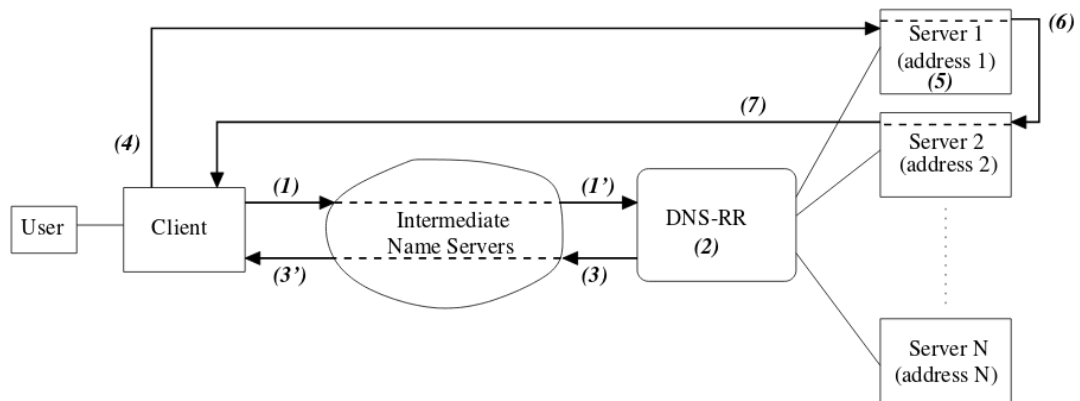
faktora. Rad (13) uvodi sinkrone (periodične) i asinkrone odluke o prosljeđivanju zahtjeva. Periodični označuje da se svaki N-ti zahtjev preusmjeri prema nekom drugom poslužitelju. Asinkroni način označuje da će se prosljeđivanje događati kada stvarno poslužitelj neće moći odgovoriti na zahtjev. Za razliku od DNS načina ili dispečera, u ovom načinu u raspodijeli sudjeluju svi poslužitelji. Najčešće razlike u implementaciji ove metode je upravo u načinu preusmjeravanja zahtjeva. Seminar će pojasniti dvije najpopularnije. Prva je preusmjeravanje pomoću HTTP protokola, dok je druga preinaka zaglavlja paketa. Ostalo je nerazriješeno kako poslužitelj odlučuje kojem će poslužitelju preusmjeriti zahtjev. Postoje statička i dinamička pravila kao i kod pristupa temeljenog na dispečeru. Dinamička i neka statička pravila najčešće uključuju dodatnu komunikaciju između poslužitelja s ciljem znanja stanja cijelog sustava.

3.4.1. HTTP preusmjeravanje



Slika 3.4: HTTP preusmjeravanje kod poslužiteljskog načina

Slika 3.4 predstavlja poslužiteljski način uravnoteživanja prometa pomoću HTTP preusmjeravanja. Korisnik prvo razrješuje nazivlje stranice u IP adresu te dobiva odgovor (1, 1', 2, 3, 3'). Zahtjev ide prema dobivenom poslužitelju (4) te poslužitelj zaključuje da ne može poslužiti trenutno taj zahtjev (5) te s HTTP 301 statusnim kodom (*moved permanently*) vraća IP adresu nekog drugog poslužitelja za kojega misli da može odgovoriti na zahtjev(6). Korisnik nakon toga šalje zahtjev prema poslužitelju kojega je dobio u odgovoru (7) te dobiva odgovor(8) od njega.



Slika 3.5: Usmjeravanje paketa kod poslužiteljskog načina

3.4.2. Usmjeravanje paketa

Slika 3.5 predstavlja poslužiteljski način usmjeravanja prometa pomoću preinaka zaglavlja paketa. Korisnik dobiva IP adresu poslužitelja preko DNS grupe (1, 1', 2, 3, 3'). Onda zahtjev šalje prema tom poslužitelju (4), ako on ne može odgovoriti (5) prosleđuje zahtjev prema drugom poslužitelju (6) te odgovori prema korisniku idu odmah prema njemu bez potrebe preusmjeravanja preko prethodnog poslužitelja (7). Metoda usmjeravanja paketa je brža od HTTP preusmjeravanja zbog manjeg broja paketa iako je bitno za primjetiti da arhitektura sustava ne ovisi o odabiru neke od ponuđenih metoda usmjeravanja.

4. Usporedbe

Četvrto poglavlje predstavlja usporedni prikaz gore navedenih metoda s cilj da se pokažu najbolje metode za pojedine situacije te također da se dodatno pojašne gore navedene implementacije metoda. Prezentirani rezultati te usporedbe preuzeti su iz rada (4). Na početku važno je napomenuti da svaka od gore navedenih metoda ima prednosti i mane te ne postoji jedna koja je bolja od svih ostalih. Tablica 4.1 prikazuje karakteristike svake od pojedinih metoda navedenih u trećem poglavlju.

Naziv metode	Vrlina	Mana
Korisnički način	Nije potrebna promjena arhitekture poslužitelja	Korisnik mora biti svjestan raspoloživih IP adresa
DNS način	Potrebna je samo modifikacija DNS servera	Pričuvna memorija DNS rezultata može rezultirati nejednakim opterećenjem
Pristup temeljen na dispečeru	Politika raspodjeljivanja se vrlo lako mijenja	Nužna je promjena arhitekture
Poslužiteljski način	Svi poslužitelji sudjeluju u uravnoteživanju opterećenja.	Poslužitelj može poslati zahtjev prema drugom poslužitelju.

Tablica 4.1: Prikaz karakteristika pojedinih metoda uravnoteživanja opterećenja

Korisnički način

Korisnički način smanjio je količinu rada koje bi inače u drugim metodama morali napraviti poslužitelji. Nažalost, takva metoda ne garantira savršeno uravnoteživanje te korisnik mora biti svjestan disprezije poslužitelja. Ako ga uspoređujemo s ostalim metodama, u njegovoj arhitekturi nema uskog grla. Još jedna zanimljiva karakteristika ovog načina naspram ostalih je da poslužitelji ne moraju biti replicirani, nego svaki

može biti specijaliziran za obradu određenih zahtjeva.

DNS način

Ako se brzina DNS načina uspoređuje s korisničkim načinom, ona je naravno sporija jer je u tom slučaju potrebno napraviti pretvorbu iz imena stranice u njezinu IP adresu. No DNS način ima mogućnost boljeg raspoređivanja zahtjeva po poslužiteljima. Ako je potrebno dodati novi poslužitelj, to je lakše napraviti u DNS načinu dodavanjem zapisa u DNS server nego u korisničkom načinu.

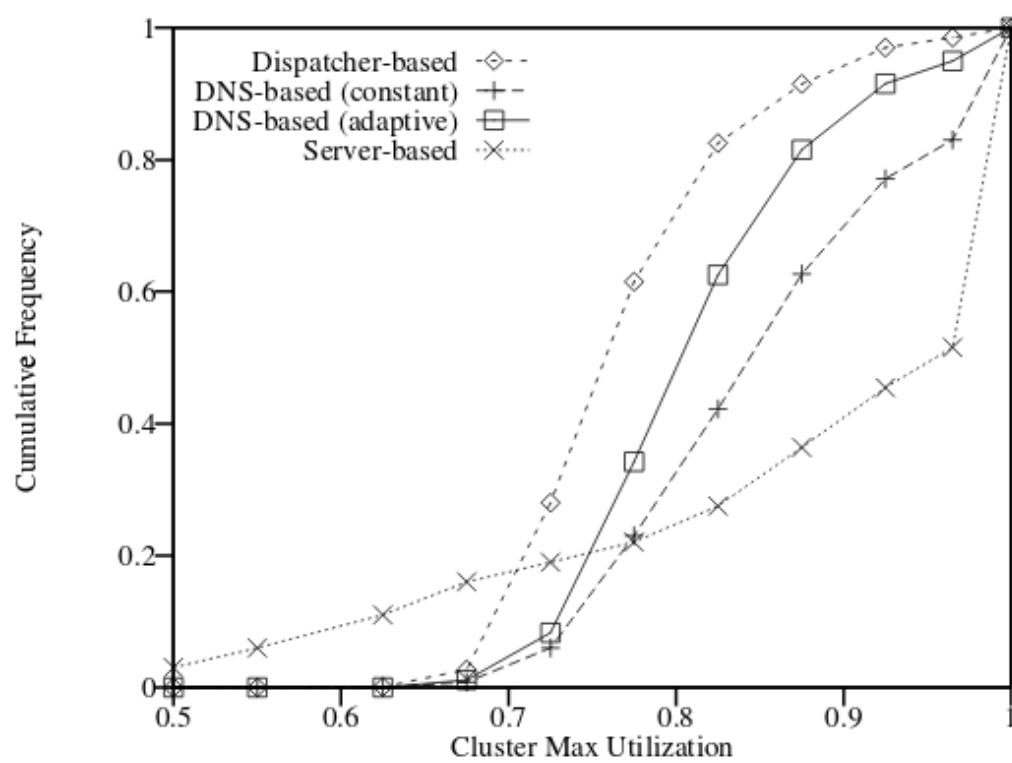
Pristup temeljen na dispečeru

Pristup temeljen na dispečeru ima najviše varijacija od svih ostalih metoda, no dispečer je usko grlo što taj način ne čini pouzdanim. Varijacije pravila raspoređivanja ovom pristupu daju mogućnost da se prilagodi raznim situacijama, ukoliko se koriste dinamična pravila raspoređivanja. Ako se koriste statična pravila to ne daje nego veću razliku naspram DNS načina.

Poslužiteljski način

Poslužiteljski način uvodi više dispečera te time se rješava problema uskog grla, no kod njega je problem vrijeme obrade zahtjeva. Ukoliko su poslužitelji opterećeni, onda zahtjev može duže vrijeme putovati do onoga koji je sposoban obraditi zahtjev. To se može riješiti dodatnom komunikacijom između svih poslužitelja no to onda uvodi dodatno mrežno opterećenje. Također, performanse poslužiteljskog načina ovise o odabiru načina preusmjerenja.

Slika 4.1 prikazuje performanse različitih metoda za uravnoteživanje opterećenja. X os predstavlja maksimalno opterećenje poslužitelja, a y os udio najvećeg prometa kojeg mogu primiti. Pristup temeljen na dispečeru je pokazao najbolje rezultate, dok je poslužiteljski način pokazao najgore. Također, DNS način s promjenjivim TTL je pokazao bolje rezultate od statičkog načina. Sukladno tome, DNS način s promjenjivim načinom je vrlo dobra alternativa pristupu s dispečerom jer on nema usko grlo u svojoj arhitekturi. Važno je napomenuti da je poslužiteljski način bio implementiran tako da se zahtjev proslijeđuje prema poslužitelju s najmanjim opterećenjem. Takva politika radi dobro, no postaje neprihvatljiva kada je opterećenje u prošlosti loše korelirano sa budućim što se događa kada ima jako puno zahtjeva.



Slika 4.1: Performanse različitih metoda za uravnoteživanje opterećenja

5. Budući razvoj

Peto poglavlje predstavlja nova istraživanja i nove rezultate na području uravnoteživanja opterećenja. Mnoge novije metode baziraju se na strojnom učenju te će peto poglavlje pobliže predstaviti prednosti i mane takvih metoda te što bi od njih moglo očekivati u budućnosti.

Podsjetimo se rada (3) koji navodi da je poželjno poznavati vrijeme obrade svakog zahtjeva unaprijed. Ako znamo to vrijeme, onda je uravnoteživanje vrlo jednostavno te se bez problema mogu koristiti statička pravila (najčešće algoritam kružnog dodjeljivanja). Određivanje trajanja obrade zahtjeva je (14) odredio pomoću neuronske mreže. Neuronska mreža je učila na skupu za učenje. Kako je sustav obrađivao zahtjeve, njihova trajanja su se davala neuronskoj mreži kako bi ona dobivala nova trajanja te u konačnici davala što bolju predikciju za buduće zahtjeve. U konačnici su dobili bolje performanse, no ipak nisu uvijek u mogućnosti točno odrediti trajanje svakog zahtjeva zbog dinamičnosti sustava. Mana ovakve metode je ta što ukoliko trajanje jednog zahtjeva se promijeni naglo, neuronskoj mreži treba dugo vremena da ga nauči. Rad (15) umjesto da odredi vrijeme obrade zahtjeva, pokušava odrediti buduće opterećenje svakog poslužitelja te na temelju tog znanja raspoređivati zahtjeve. Rad koristi metodu predviđanja temeljenu na eksponencijalnom izgladivanju (engl. *Exponential smoothing forecasting method*).

$$x_{t+1} = \alpha x_t + (1 - \alpha) \hat{x}_t \quad (5.1)$$

U izrazu 5.1 starije vrijednosti kako vrijeme ide dobivaju sve manji značaj, dok novije vrijednosti imaju veći doprinos. Navedena metoda dobro je predviđala buduće opterećenje, no ona ovisi o odabiru hiperparametra α koji može varirati u različitim sustavima.

6. Zaključak

Uravnoteživanje opterećenja u raspodijeljenim sustavima je od kritičnog značaja za velike web sustave kako bi bez problema odgovarali na zahtjeve korisnika. Prikazano je da postoji veliki broj različitih metoda kojima se opterećenje poslužitelja u raspodijeljenim sustavima može ujednačiti. Primjena pojedine metode ovisi o velikom broju faktora, ponajviše arhitekturi mreže te o tome što korisnik smije znati o sustavu. Ukoliko korisnik smije znati za dva ili više izoliranih dijelova aplikacije, onda se može koristiti korisnička metoda. Ukoliko se želi dodati uravnoteživanje opterećenja bez prevelike promjene arhitekture, DNS način je dobar odabir. Izbor između pristupa temeljenog na dispečeru i poslužiteljskog načina koji u sebi uključuje DNS način ovisi ponajviše o tome koliko je na raspolaganju javnih IP adresa. Drugi važni faktor je koliko se može vjerovati dispečeru jer je on "usko grlo" te njegovim padom pada cijeli sustav. Poslužiteljski način ima više dispečera te bi padom jednog od njih DNS grupa mogla saznati za to periodičnim slanjem ICMP poruka te promet uopće ne slati prema njemu. Metode bazirane na pokušaju predikcije trajanje obrade zahtjeva ili opterećenja poslužitelja pokazale su dobre performanse u početnim uvjetima sustava, no kako su promjene bivale sve veće, moć predikcije je padala te su se sve teže prilagođavale. Iako su rezultati pokazali da čak i takve predikcije daju bolje rezultate od na primjer statičkih pravila razmještanja. U budućnosti se možemo nadati dodatnom razvoju u tom području koji bi mogao pratiti razvoj umjetne inteligencije. Seminar nije razmatrao pametne usmjeritelje (engl. *smart routers*) niti softverski definirane mreže (engl. *software defined network*) iako i oni na svoj način mogu pomoći uravnoteživanju opterećenja u raspodijeljenim sustavima.

7. Literatura

- [1] “Round-robin (document) - Wikipedia.” [Online]. Available: [https://en.wikipedia.org/wiki/Round-robin_\(document\)](https://en.wikipedia.org/wiki/Round-robin_(document))
- [2] “Internet Live Stats - Internet Usage & Social Media Statistics.” [Online]. Available: <http://www.internetlivestats.com/>
- [3] M. A. Iqbal, J. H. Saltz, and S. H. Bokhart, “Performance tradeoffs in static and dynamic load balancing strategies,” mar 1986. [Online]. Available: <https://ntrs.nasa.gov/search.jsp?R=19860014876&hterms=Performance+Tradeoffs+Static+Dynamic+Load+Balancing+Strategies&qs=N%D0%2526Ntk%253DAI%2526Ntt%253DPerformance%252520Tradeoffs%252520in%252520Static%252520and%252520Dynamic%252520Load%252520Balancin>
- [4] V. Cardellini, M. Colajanni, and P. Yu, “Dynamic load balancing on Web-server systems,” *IEEE Internet Computing*, vol. 3, no. 3, pp. 28–39, 1999. [Online]. Available: <http://ieeexplore.ieee.org/document/769420/>
- [5] Z. Duan and Z. Gu, “Dynamic Load Balancing in Web Cache Cluster,” in *2008 Seventh International Conference on Grid and Cooperative Computing*. IEEE, oct 2008, pp. 147–150. [Online]. Available: <http://ieeexplore.ieee.org/document/4662856/>
- [6] A. Naaz, Sameena and Meftah Alrayes, Mohamed and Alam, “A Dynamic Load Balancing Algorithm For Web Applications.” [Online]. Available: https://www.researchgate.net/publication/228416308_A_Dynamic_Load_Balancing_Algorithm_For_Web_Applications
- [7] “Ask F5 | Manual Chapter: Pools.” [Online]. Available: https://techdocs.f5.com/kb/en-us/products/big-ip/_htm/manuals/product/lm-concepts-11-5-0/5.html

- [8] S. Wee and H. Liu, "Client-side load balancer using cloud," in *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10*. New York, New York, USA: ACM Press, 2010, p. 399. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1774088.1774173>
- [9] D. Mosedale, W. Foss, and R. McCool, "Lessons learned administering Netscape's Internet site," *IEEE Internet Computing*, vol. 1, no. 2, pp. 28–35, 1997. [Online]. Available: <http://ieeexplore.ieee.org/document/601086/>
- [10] M. Baentsch, L. Baum, G. Molter, S. Rothkugel, and P. Sturm, "Enhancing the Web's infrastructure: from caching to replication," *IEEE Internet Computing*, vol. 1, no. 2, pp. 18–27, 1997. [Online]. Available: <http://ieeexplore.ieee.org/document/601083/>
- [11] M. Colajanni, P. Yu, and V. Cardellini, "Dynamic load balancing in geographically distributed heterogeneous Web servers," in *Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No.98CB36183)*. IEEE Comput. Soc, pp. 295–302. [Online]. Available: <http://ieeexplore.ieee.org/document/679729/>
- [12] "Round Robin - phrase meaning and origin." [Online]. Available: <https://www.phrases.org.uk/bulletinboard/15/messages/643.html>
- [13] V. Cardellini, M. Colajanni, and P. Yu, "Redirection algorithms for load sharing in distributed Web-server systems," in *Proceedings. 19th IEEE International Conference on Distributed Computing Systems (Cat. No.99CB37003)*. IEEE Comput. Soc, pp. 528–535. [Online]. Available: <http://ieeexplore.ieee.org/document/776555/>
- [14] "Using Machine Learning to Load Balance Elasticsearch Queries - Meltwater Engineering Blog." [Online]. Available: <https://underthehood.meltwater.com/blog/2018/09/28/using-machine-learning-to-load-balance-elasticsearch-queries/>
- [15] X. Ren, R. Lin, and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*. IEEE, sep 2011, pp. 220–224. [Online]. Available: <http://ieeexplore.ieee.org/document/6045063/>

8. Sažetak

U ovom seminaru napravljena retrospektiva i podjela vrsta uravnoteživanja opterećenja u raspodijeljenim sustavima. Specificirana je potreba njihovog uvođenja te je prikazana glavna podjela metoda uravnoteživanja opterećenja. Prikazana je arhitektura mreže svake metode te su pojašnjene podmetode i njihove implementacije. Prikazani su i komentirani usporedni rezultati preuzeti iz drugih radova. Pokazani su neki nedavni radovi koji uvode predikciju kako bi poboljšali svoje performanse te su njihovi rezultati podrobnije pojašnjeni.