

## EMULACIJA RAČUNALNOG KLASTERA ZA RASPOREĐIVANJE OPTEREĆENJA I POVEĆANJE DOSTUPNOSTI MREŽNIH SERVISA<sup>3</sup>

### SAŽETAK

Članak prikazuje mogućnost uspostave emulacije računalnog klastera kojim se želi povećati dostupnost i rasporediti opterećenje mrežnog servisa. Temeljni cilj uspostave emuliranoga računalnog klastera je provjera izvedivosti izgradnje laboratorija za emulaciju klastera unutar kojega bi se mogla vršiti različita testiranja sustava, analizirati tijek podataka unutar komponenti sustava te isprobavati različita rješenja prije njihove stvarne implementacije. Članak opisuje metodiku emulacije računalnog klastera primjenom emulacije računala preko računalnog programa VirtualBox, emulacije računalne mreže preko računalnog programa GNS3, te računalnog programa za raspoređivanje opterećenja HAProxy i računalnog programa za povećanje dostupnosti Heartbeat. Prikazana je praktična primjena opisane metodike na emulaciji računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti web mrežnog servisa koji je implementiran računalnim programom Apache HTTP Server. Članak detaljnije opisuje računalni klaster za povećanje dostupnosti, računalni klaster za raspoređivanje opterećenja, te njihovu kombinaciju. Detaljno se prikazuju razlozi zbog kojih je poželjno uspostaviti laboratorij za emulaciju računalnog klastera. U zaključku članka argumentira se izvedivost laboratorija za emulaciju računalnog klastera, te se detaljnije prikazuju daljnji pravci njegova razvoja.

**Ključne riječi:** emulacija, računalni klaster, mrežni servis, sustav za raspoređivanje opterećenja, sustav za povećanje dostupnosti

### 1. UVOD

Korisnici računalnih mreža u svojim se aktivnostima, neovisno o domeni kojoj pripadaju (npr. poslovanje, edukacija, istraživanje, umjetnost, sport, privatni život itd.), koriste raznovrsnim mrežnim servisima. U biti, ne postoji ljudska djelatnost u kojoj se ne mogu iskoristiti mrežni servisi. Mrežni servisi mogu se definirati kao distribuirani hardversko-softverski sustavi koji uključuju više međusobno umreženih računala i koji korisnicima pružaju podršku u njihovim aktivnostima. Neki od danas široko korištenih mrežnih servisa su: *web*, elektronička pošta, razmjena datoteka, društvene mreže, diskusijske grupe, blogovi, telekonferencije, rad na daljinu, računalni oblak itd. Najčešće se koriste mrežni servisi implementirani na mreži svih mreža - internetu, ali se mrežni servisi mogu implementirati i u privatnim

<sup>1</sup> Dr. sc., docent, profesor visoke škole, Veleučilište u Rijeci, Vukovarska 58, Rijeka, Hrvatska. E-mail: alen.jakupovic@veleri.hr

<sup>2</sup> Student, stručni prvostupnik informatike, sistemski administrator, Infobip d.o.o. Mletačka 12/III, Pula, Hrvatska. E-mail: petar.koraca@veleri.hr

<sup>3</sup> Datum primitka rada: 14. 1. 2013.; datum prihvaćanja rada: 3. 4. 2013.

lokalnim mrežama. Tijekom uporabe mrežnih servisa, uvijek se javljaju dva sustava - jedan koji ga je zatražio (klijent) i jedan koji ga pruža (poslužitelj).

U razvoju i implementaciji nekog mrežnog servisa susreću se tri osnovna izazova: skalabilnost (lakoća širenja sustava), dostupnost i niski troškovi (Fox et al., 1997). Ovi izazovi prije svega odnose se na poslužitelja, tj. sustav koji pruža mrežni servis. Kao primjer radnog okruženja mrežnog servisa može poslužiti Googleov servis elektroničke pošte - gmail. Ovaj servis pokrenut je 2004. godine. U prvom mjesecu 2012. godine servis je koristilo oko 350 milijuna korisnika. U lipnju iste godine broj korisnika povećao se na 425 milijuna. Dinamika povećanja korisnika je 15 milijuna na mjesec (D'Orazio, 2012). Pitanja koja se ovdje nameću su: kako osigurati lako širenje sustava koji će podržati ovakvu dinamiku povećanja broja korisnika, kako osigurati dostupnost sustava, bilo da se radi o povećanju brzine njegova odziva na korisničke zahtjeve ili umanjenu njegovu osjetljivosti na ispade sustava, te kako sve navedeno izgraditi uz minimalne troškove.

Jedno od rješenja problema skalabilnosti, dostupnosti i niskih troškova mrežnih servisa jesu računalni klasteri. Računalni klaster (engl. *cluster*) predstavlja vrstu paralelnoga i distribuiranoga računalnog sustava koji se sastoji od međusobno povezanih samostojećih računala koja međusobno surađuju kao jedan integralni računalni resurs (Buyya, 1999; Pfister, 1998). Gledano izvana, ovaj sustav ponaša se kao jedno računalo čije performanse premašuju performanse pojedinih računala u klasteru. Upravo su računalni klasteri omogućili poslovni uspjeh tvrtki kao što su Amazon, Yahoo!, Google, Facebook, Twitter itd., budući da se njihovi mrežni servisi koji poslužuju milijune korisnika upravo temelje na tisućama poslužitelja koji osiguravaju dostupnost i skalabilnost, jer je jeftinije i lakše širiti se horizontalno (dodavanjem novih poslužitelja) nego vertikalno (nabavom skupljih, snažnih računala) (Adelstein, Lubanovic, 2007).

Pored uvodnog dijela, članak ima dodatnih pet poglavlja. U drugom poglavlju opisuju se tri vrste računalnih klastera koji su u fokusu ovoga rada. Treće poglavlje donosi motivaciju za uspostavu laboratorija za emulaciju računalnog klastera. Četvrto poglavlje opisuje metodiku emulacije računalnog klastera, a peto prikazuje njezinu praktičnu primjenu na emulaciji računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti *web* mrežnog servisa. Članak završava zaključkom u kojem se donosi argumentacija izvedivosti laboratorija za emulaciju računalnog klastera, te daljnji pravci njegova razvoja.

## 2. RAČUNALNI KLASTER

IBM-ov inženjer Gene Amdahl (1967) prvi put formalno je oblikovao načela na kojima se temelje klaster sustavi. On je matematički opisao ubrzanje izvođenja zadatka koje se može postići njegovim razlaganjem na manje elemente koji se onda paralelno izvode na računalnom sustavu s paralelnom arhitekturom. Time je postavio Amdahlov zakon na kojemu se temelje današnji klaster sustavi visoke dostupnosti.

Razvoj računalnih klastera tekao je paralelno s razvojem računalnih mreža, budući da je to temeljna infrastruktura svakog klaster sustava. Tek nakon pojave dobro definiranih protokola komunikacije među računalima i mrežnih alata koji ih implementiraju (negdje oko 1983. godine), klaster sustavi postaju široko

dostupni. Pojavom programskog sustava otvorenoga koda naziva PVM (engl. *parallel virtual machine*) koji omogućava uspostavu klaster sustava temeljenog na TCP/IP komunikaciji, izgrađeni su klaster sustavi iz jeftinih PC-ja koji su po broju operacija u sekundi (engl. *floating point operations per seconds* - FLOPS) nadišli napredna superračunala. Ovo je bio i poticaj za razvoj grid računarstva (CARNet, 2006).

Računalni klasteri prema namjeni mogu se podijeliti na: računalne klastere visoke dostupnosti (engl. *high availability* - HA), računalne klastere za raspoređivanje opterećenja (engl. *load balancing* - LB) i računalne klastere visokih performansi (engl. *high performance* - HP) (CARNet, 2006).

Računalni klaster visokih performansi koristi se za rješavanje zadataka koji se mogu razložiti u više manjih zadataka koji se potom paralelno rješavaju na više računala. Najčešće su to složeni matematički proračuni čije se vrijeme rješavanja značajno skraćuje paralelnim radom više računala (CARNet, 2006).

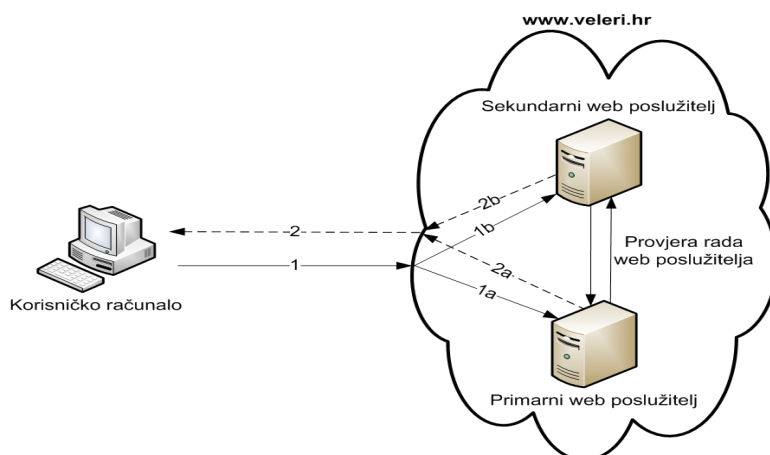
Područje interesa ovoga članka preostale su dvije vrste računalnih klastera te će oni biti dodatno opisani u poglavlju koje slijedi.

## 2.1 Računalni klaster visoke dostupnosti mrežnih servisa

Općenito se dostupnost sustava može definirati kao vjerojatnost da je u nekom trenutku on u funkciji i da je sposoban pružiti zatraženu uslugu (Sommerville, 2007). Slično se može definirati i dostupnost mrežnog servisa - vjerojatnost da je u nekom trenutku mrežni servis na poslužitelju u funkciji i da je sposoban pružiti servis koji je klijent zatražio. Povećanje dostupnosti mrežnog servisa postiže se izgradnjom sustava u kojemu postoji redundancija za sve njegove ranjive elemente. Redundancija se ogleda u tome da postoje najmanje dva ista elementa u strukturi sustava (npr. više istih poslužitelja, mrežnih preklopnika, mrežnih diskova itd.).

Naglasak u ovome članku je na redundanciji poslužitelja koji pružaju mrežni servis. Shema 1 prikazuje model povećanja dostupnosti *web* mrežnog servisa ([www.veleri.hr](http://www.veleri.hr)) koji se nalazi na dva poslužitelja.

Shema 1. Model povećanja dostupnosti web mrežnog servisa [www.veleri.hr](http://www.veleri.hr)

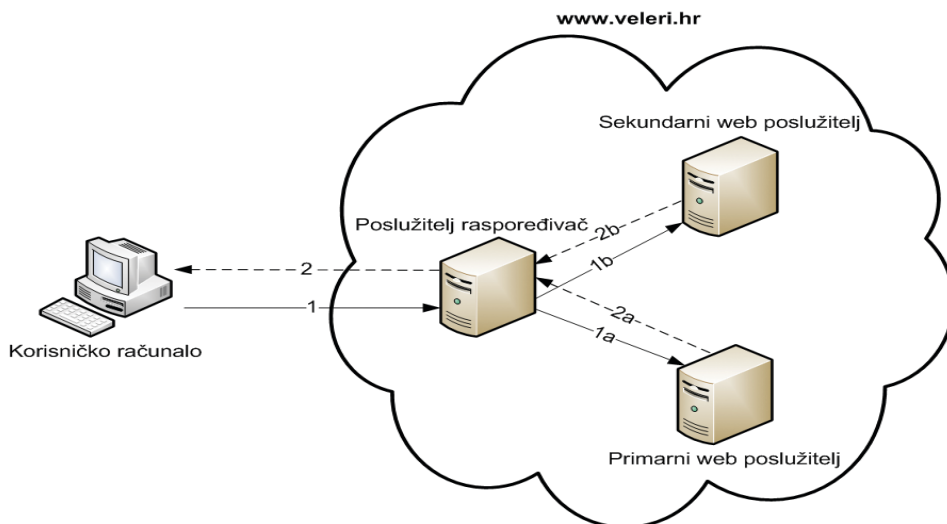


Schema 1 prikazuje dva *web*-poslužitelja (primarni i sekundarni) na kojima je implementiran mrežni servis [www.veleri.hr](http://www.veleri.hr), korisničko računalo i tokove podataka među računalima. Kada korisničko računalo zatraži servis (tok podataka 1), zahtjev prihvaća primarni *web*-poslužitelj (tok podataka 1a), te servis dostavlja tokom podataka 2a koji do korisničkog računala stiže tokom podataka 2. Između primarnog i sekundarnog *web*-poslužitelja postoje tokovi podataka kojima se provjerava rad poslužitelja. Sekundarni *web*-poslužitelj periodički provjerava rad primarnog *web*-poslužitelja i obrnuto. U slučaju da primarni *web*-poslužitelj prestane s radom, sekundarni *web*-poslužitelj to doznaje, te on preuzima posao primarnog *web*-poslužitelja. Tada bi zahtjev korisničkog računala (tok podataka 1) sada preuzeo sekundarni *web*-poslužitelj (tok podataka 1b), te bi on odgovorio na zahtjev korisničkog računala (tok podataka 2b). Primarni i sekundarni *web*-poslužitelj čine jednostavni računalni klaster za povećanje dostupnosti *web* mrežnog servisa. Vidljivo je da se takav klaster treba sastojati od najmanje dva računala.

## 2.2 Računalni klaster za raspoređivanje opterećenja mrežnih servisa

U primjeni mrežnih servisa dva su osnovna sudionika - jedan sudionik (klijent) zahtijeva servis, a drugi sudionik (poslužitelj) na zahtjev odgovara pružanjem servisa. Moguće je da više klijenata istovremeno ili u kratkom vremenskom periodu zatraže isti mrežni servis, odnosno da više zahtjeva stigne do poslužitelja. Poslužitelj će, do određenog broja zahtjeva, uspijevati odgovarati pružanjem mrežnih servisa. Nakon toga će klijenti početi osjećati značajna kašnjenja u pružanju servisa, da bi se na kraju dogodila njihova potpuna nedostupnost - kolaps mrežnog servisa. Rješenje ovoga problema leži u raspoređivanju opterećenja klijentskim zahtjevima na više poslužitelja. Schema 2 prikazuje model raspoređivanja opterećenja *web* mrežnog servisa ([www.veleri.hr](http://www.veleri.hr)).

Schema 2. Model raspoređivanja opterećenja *web* mrežnog servisa [www.veleri.hr](http://www.veleri.hr)



Izvor: obrada autora

Shema 2 prikazuje jedan poslužitelj raspoređivač, dva *web*-poslužitelja (primarni i sekundarni) na kojima je implementiran *web* mrežni servis [www.veleri.hr](http://www.veleri.hr), korisničko računalo i tokove podataka među računalima. Kada korisničko računalo uputi zahtjev za mrežnim servisom (tok podataka 1) prihvaća ga poslužitelj raspoređivač. Uloga ovoga poslužitelja je da pristigli korisnički zahtjev prosljedi primarnom ili sekundarnom *web*-poslužitelju. Odluku o prosljeđivanju donosi temeljem trenutne opterećenosti *web*-poslužitelja. Npr. ako primarni *web*-poslužitelj nije dostigao kritičnu razinu opterećenosti, zahtjevi se prosljeđuju njemu (tok podataka 1a). On na zahtjev odgovara mrežnim servisom (tok podataka 2a) koji kroz poslužitelj raspoređivač dolazi do korisničkog računala (tok podataka 2). U slučaju da je primarni *web*-poslužitelj dostigao kritičnu razinu opterećenosti, poslužitelj raspoređivač prosljeđuje korisnički zahtjev sekundarnom *web*-poslužitelju (tok podataka 1b) koji odgovara s mrežnim servisom (tok podataka 2b), te se on preko poslužitelja raspoređivača dostavlja korisničkom računalu. Prikazana tri poslužitelja (raspoređivač, primarni i sekundarni) čine računalni klaster za raspoređivanje opterećenja *web* mrežnog servisa. Može se zaključiti da se ovakav računalni klaster treba sastojati od najmanje tri računala.

### 2.3 Kombinacija računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti mrežnih servisa

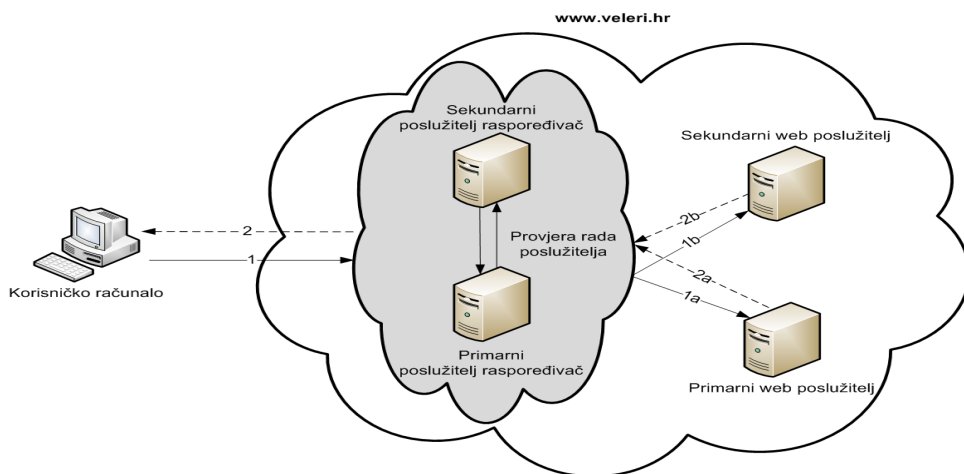
U prethodnom poglavlju pokazan je minimalni broj računala u računalnom klasteru za raspoređivanje opterećenja *web* mrežnog servisa (slika 2). Temelj takvog klastera čini poslužitelj raspoređivač koji klijentske zahtjeve raspoređuje na *web*-poslužitelje na kojima je implementiran mrežni servis. U slučaju ispada nekog *web*-poslužitelja s mrežnim servisom, raspoređivač će korisničke zahtjeve rasporediti na ostale dostupne *web*-poslužitelje. Ovime računalni klaster za raspoređivanje opterećenja djelomično pokriva svojstva klastera za povećanje dostupnosti mrežnih servisa. Naime, dok je situacija s ispadom *web*-poslužitelja s mrežnim servisom pokrivena, situacija s ispadom samog poslužitelja raspoređivača nije. Ispad poslužitelja raspoređivača uzrokovao bi potpuni ispad mrežnog servisa, stoga je potrebno dodati redundantni poslužitelj raspoređivač kako bi se dobio računalni klaster za raspoređivanje opterećenja i povećanje dostupnosti mrežnih servisa.

Dva su načina funkcioniranja redundantnih poslužitelja raspoređivača koja ovise o njihovoj konfiguraciji (Kopparapu, 2002):

- Aktivno-pasivna konfiguracija - jedan poslužitelj raspoređivač aktivno raspoređuje klijentske zahtjeve dok drugi (pasivni) čeka ispad prvoga. Dok je u pasivnom načinu rada ne odgovara na klijentske zahtjeve.
- Aktivno-aktivna konfiguracija - oba poslužitelja raspoređivača aktivno raspoređuju klijentske zahtjeve. U slučaju ispada jednoga poslužitelja, drugi preuzima njegov posao.

Shema 3 prikazuje model računalnog klastera koji u sebi kombinira klaster za raspoređivanje opterećenja *web* mrežnog servisa [www.veleri.hr](http://www.veleri.hr) (ujedno povećava i njegovu dostupnost), te klaster za povećanje dostupnosti poslužitelja raspoređivača.

Schema 3. Model računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti *web* mrežnog servisa *www.veleri.hr*



Izvor: obrada autora

Schema 3 prikazuje minimalni računalni klaster za raspoređivanje opterećenja i povećanje dostupnosti *web* mrežnog servisa. Sastoji se iz ukupno četiri poslužitelja - primarni i sekundarni poslužitelj raspoređivača, te primarni i sekundarni *web*-poslužitelj. Korisnički zahtjev (tok podataka 1) prihvaća aktivni poslužitelj raspoređivača (npr. primarni), te zahtjev proslijeđuje nekom od *web*-poslužitelja koji zatim odgovor, tj. traženi mrežni servis preko poslužitelja raspoređivača dostavlja korisniku. Vidljivo je da se unutar računalnog klastera za raspoređivanje opterećenja nalazi još jedan računalni klaster - onaj za povećanje dostupnosti poslužitelja raspoređivača (sivo obojani oblak).

### 3. MOTIVACIJA ZA EMULACIJU RAČUNALNOG KLASTERA ZA RASPOREĐIVANJE OPTEREĆENJA I POVEĆANJE DOSTUPNOSTI MREŽNIH SERVISIA

Prikazani minimalni računalni klasteri za raspoređivanje opterećenja i povećanje dostupnosti sastoje se iz dva, tri ili četiri međusobno umrežena računala (ovisno o namjeni klastera). Moguća je uspostava računalnih klastera i s većim brojem računala. Osim potrebe umrežavanja računala u klasteru uz primjenu niza drugih mrežnih uređaja (npr. mrežnih kabela, preklopnika, usmjerivača, pristupnih točki itd.), na navedena računala je potrebno izvršiti implementaciju određenoga mrežnog servisa, te implementaciju računalnih programa koji će osigurati raspoređivanje opterećenja i povećanje dostupnosti. Zatim, jednom uspostavljeni računalni klaster treba proći niz provjera te eventualnih konfiguracijskih izmjena dok se ne izgradi konačan računalni klaster koji je moguće pustiti u pogon. Za potrebe provjere računalnog klastera potrebna su dodatna računala koja će s njime biti umrežena.

Pitanje koje se može postaviti je kako ostvariti praktičnu edukaciju iz domene računalnih klastera budući da su i za najmanji računalni klaster potrebna dva umrežena računala, odnosno

još jedno dodatno za njegovu provjeru (ukupno tri umrežena računala). Problem je još izraženiji ako se promatraju računalni klasteri s više računala. Ovaj problem javlja se neovisno izvodi li se edukacija u nekom poduzeću ili obrazovnoj instituciji. Primjena fizičkih računala i fizičkim mrežnih uređaja povlači za sobom potrebu za prostorom gdje bi se takva oprema skladištila, prostorom gdje bi polaznici edukacije mogli računala i mrežne uređaje međusobno umrežiti, na njih izvršiti instalacije potrebnih računalnih programa, izvršiti konfiguriranje i konačno provjeru uspostavljenog klastera. Značajan je i trošak nabave potrebnih računala i mrežnih uređaja, posebno ako ih se želi osigurati u većem broju za svakog polaznika. Značajan problem je i rastavljanje računalnog klastera i dovođenje računala i mrežnih uređaja u prvobitno stanje (rastavljanje mreže, uklanjanje instaliranih računalnih programa itd.), a kako bi ih se moglo iskoristiti za uspostavu novih klastera.

Slična problematika postoji i u poduzećima koja implementiraju računalne klastere. Naime, puno je lakše implementaciju izvršiti nakon što je ona isprobana u testnoj okolini. Ako bi se za testnu okolinu koristila fizička računala i mrežni uređaji, ponovno se dolazi do već opisanih problema.

Osim probne implementacije računalnih klastera, u testnoj okolini se mogu isprobavati i testirati različiti računalni programi potrebni za uspostavu klastera i mrežnog servisa, različite konfiguracije sustava, novi računalni programi, proizvodi i usluge, te postojeći i novorazvijeni komunikacijski protokoli koje koriste računala u klasteru.

Sve do sada navedeno upućuje na snažnu potrebu za uspostavom laboratorija za emulaciju računalnih klastera za raspoređivanje opterećenja i povećanje dostupnosti mrežnih servisa koji neće zahtijevati potrebu za većim brojem fizičkih računala i mrežnih uređaja. Laboratorij bi trebao osigurati primjenu samo jedne radne stanice za emulaciju cjelokupnog računalnog klastera, ali i korisničkih računala. Trebao bi omogućiti povezivanje takvoga klastera u stvarnu fizičku mrežu, te omogućiti uspostavu različitih topologija (struktura) računalnih mreža povezanih s klasterom. Pored navedenoga, laboratorij bi trebao osigurati mogućnost uspostave različitih mrežnih servisa, mogućnost analize komunikacijskih protokola, te jednostavno rastavljanje sustava i dovođenje njegovih elemenata (računala i mrežnih uređaja) u prvobitno stanje. Laboratorij bi trebao omogućiti spremanje mrežnih topologija i računalnih klastera, te njihov jednostavni prijenos na druga računala.

#### **4. METODIKA EMULACIJE RAČUNALNOG KLASTERA ZA RASPOREĐIVANJE OPTEREĆENJA I POVEĆANJE DOSTUPNOSTI MREŽNIH SERVISA**

U prethodnom poglavlju naveden je niz zahtjeva koje bi trebao zadovoljiti laboratorij za emulaciju računalnih klastera. Jedan od zahtjeva je da se cjelokupni računalni klaster može izgraditi unutar samo jedne radne stanice. Budući da se računalni klaster sastoji od više međusobno umreženih računala na koja su instalirani različiti računalni programi, to se unutar radne stanice mora moći prikazati više neovisnih računala na koja je moguće izvršiti instalaciju računalnih programa, te se mora moći izvršiti njihovo umrežavanje.



Ovo se može postići metodom emulacije računala i računalne mreže. Emulacija (oponašanje) može se definirati kao metoda kojom se oponaša funkcioniranje jednoga sustava (sustav gost, engl. *guest*) preko nekog drugog sustava (sustav domaćin, engl. *host*). Kako bi se oponašao sustav gost, sustav domaćin mora posjedovati emulator - računalni program i/ili sklopovlje koje omogućava oponašanje funkcioniranja sustava iz stvarnog svijeta. Temeljno svojstvo emulacije je da vanjski promatrač doživljava emulirani sustav kao da se radi o stvarnom sustavu - među njima ne zamjećuje razlike (Jakupović, Knežević, 2012).

Metodom emulacije moguće je unutar jednog fizičkog računala pokrenuti više međusobno neovisnih drugih računala s drugim operacijskim sustavima i instaliranim računalnim programima. U tu svrhu mogu se koristiti različiti besplatni emulatori računala. Za potrebe ovoga članka koristit će se emulator VirtualBox (<https://www.virtualbox.org/>). Treba naglasiti da svako emulirano računalo zahvaća resurse fizičkoga računala (npr. radnu memoriju) te je time ograničen broj emuliranih računala koji se mogu pokrenuti istovremeno. Također, moguće je, preko sučelja fizičkoga računala, u emulirano računalo uvući periferne jedinice (npr. CD/DVD čitač, USB memoriju itd.).

Na jednom fizičkom računalu, metodom emulacije moguće je prikazati računalnu mrežu koja se sastoji od emuliranih računala i drugih mrežnih uređaja. Postoji niz besplatnih emulatora računalne mreže, a za potrebe članka koristit će se emulator GNS3 (<http://www.gns3.net/>). Emulator računalne mreže omogućava njezino povezivanje s emuliranom mrežom na drugom fizičkom računalu, te sa stvarnom fizičkom mrežom i stvarnim fizičkim računalima i mrežnim uređajima. Također je moguće da se u emuliranu računalnu mrežu implementiraju mrežni servisi. Emulator GNS3 omogućava analizu mrežnoga prometa, odnosno komunikacijskih protokola (Jakupović, Knežević, 2012). U (Zec, 2012) može se pronaći više o simulaciji i emulaciji računalnih mreža kao i o komparativnoj analizi različitih virtualizacijskih tehnika.

Jakupović i Knežević (2012) detaljno su prikazali praktičnu primjenu emulatora VirtualBox i GNS3 na složenoj topologiji računalne mreže koja se sastojala od ukupno osam emuliranih računala s operacijskim sustavom Linux TinyCore. Topologija je sadržavala dva korisnička računala, tri mrežna preklopnika, po dva mrežna usmjerivača, DHCP i *web*-poslužitelja, te tri DNS poslužitelja. Dakle, unutar mrežne topologije, pored korisničkih računala i mrežnih uređaja, implementirana su i tri mrežna servisa (DHCP, WEB i DNS).

Jakupović i Pirić (2013) su, prema prikazu primjene emulatora VirtualBox i GNS3 u (Jakupović, Knežević, 2012) prikazali uspostavu laboratorija za emulaciju računalnih mreža, te njegovu primjenu u praktičnoj nastavi. Prikazani su primjeri nastavnih sadržaja i ishoda učenja koji su podržani laboratorijem, opisan je način primjene laboratorija u provjeri ishoda učenja, te je prikazana mogućnost njegove primjene u realizaciji praktičnih studentskih projekata uspostave mrežnih servisa.

Za potrebe emulacije računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti mrežnih servisa, emulator VirtualBox koristit će se za emulaciju računala u klasteru, a emulator GNS3 za njihovo međusobno umrežavanje, te umrežavanje korisničkih računala s uspostavljenim klasterom. Emulator GNS3 omogućit će povezivanje računalnog klastera sa stvarnom fizičkom



mrežom, te analizu mrežnog prometa. Na računala emulirana s emulatorom VirtualBox moći će se instalirati potrebni operacijski sustavi i računalni programi.

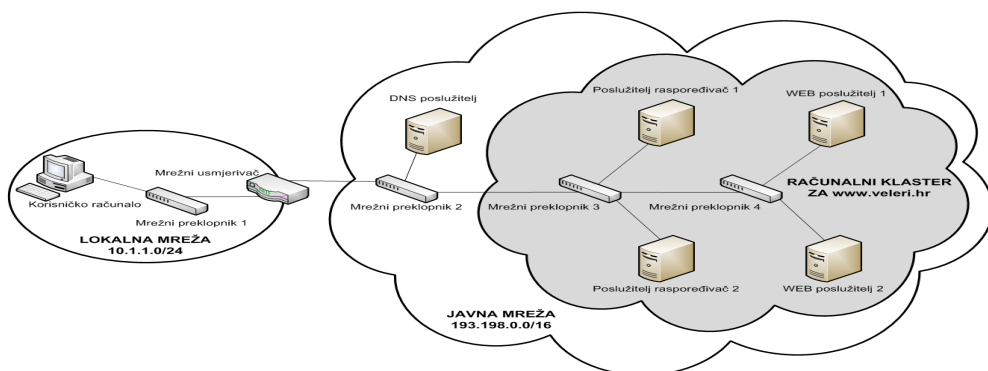
Uspostava računalnog klastera zahtijeva dva posebna računalna programa - jedan koji će implementirati raspoređivanje opterećenja i drugi koji će implementirati povećanje dostupnosti. Raspoređivanje opterećenja podržat će se besplatnim računalnim programom HAProxy (<http://haproxy.1wt.eu/>). On implementira raspoređivanje opterećenja mrežnih servisa temeljenih na TCP i HTTP protokolu, te je pogodan za *web* mrežni servis na čijem primjeru će se i uspostaviti računalni klaster. Povećanje dostupnosti implementirat će se besplatnim računalnim programom Heartbeat (<http://linux-ha.org/wiki/Heartbeat>). Oba navedena programa bit će instalirana unutar operacijskog sustava Linux Debian.

Emulacija računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti bit će izvedena za *web* mrežni servis koji će se implementirati besplatnim računalnim programom Apache HTTP Server na računalu s operacijskim sustavom Linux TinyCore. Za druge mrežne servise (npr. DNS, FTP, *e-mail* itd.) potrebno je izabrati i instalirati odgovarajuće računalne programe. No treba voditi računa o tome da se mrežni servisi temelje na protokolu TCP i HTTP, budući da to zahtijeva HAProxy - korišteni računalni program za raspoređivanje opterećenja.

## 5. REZULTATI EMULACIJE RAČUNALNOG KLASTERA ZA RASPOREĐIVANJE OPTEREĆENJA I POVEĆANJE DOSTUPNOSTI WEB MREŽNIH SERVISA

Kako bi se provjerila primjenjivost opisane metodike emulacije računalnog klastera, izrađena je emulacija računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti *web* mrežnih servisa. Shema 4. prikazuje emuliranu topologiju računalne mreže unutar koje se nalazi računalni klaster.

Shema 4. Topologija emulirane računalne mreže s računalnim klasterom za raspoređivanje opterećenja i povećanje dostupnosti *web* mrežnog servisa [www.veleri.hr](http://www.veleri.hr)



Izvor: obrada autora

Emulirana topologija računalne mreže s računalnim klasterom (shema 4) sastoji se od dviju međusobno povezanih mreža (povezuje ih mrežni usmjerivač). U lokalnoj mreži se nalazi jedno korisničko računalo, dok se u drugoj mreži nalazi ukupno pet poslužitelja (jedan DNS poslužitelj i po dva poslužitelja za raspoređivač i web mrežni servis). Sljedeća tablica detaljno opisuje zahtjeve koje treba zadovoljiti prikazana topologija računalne mreže.

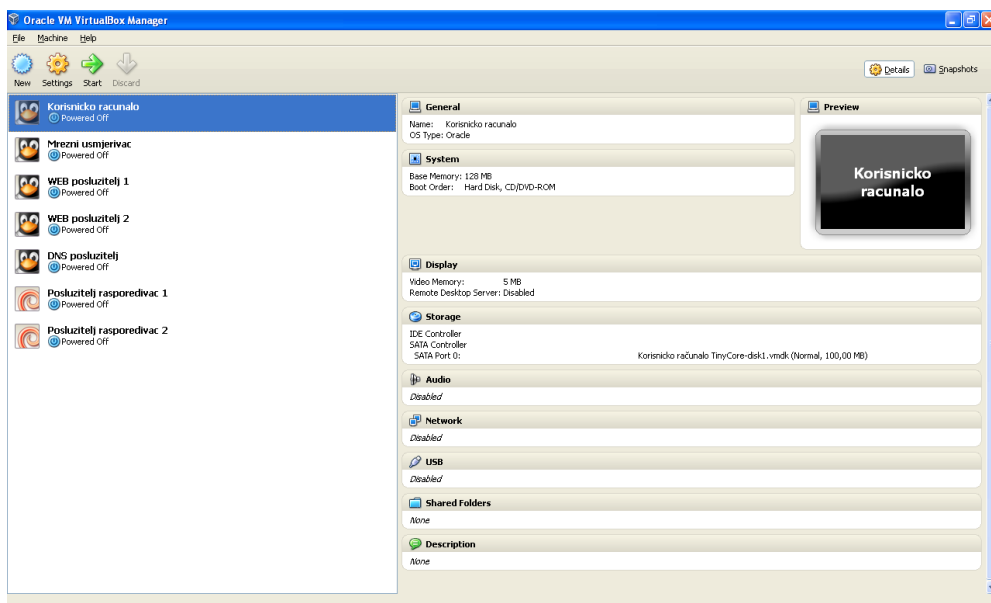
Tablica 1. Zahtjevi za topologiju računalne mreže s računalnim klasterom

Naziv mrežnog uređaja	Zahtjev
Mrežni preklopnik 1, 2, 3 i 4	Ethernet mrežni preklopnik
Korisničko računalo	Ima stalnu IP adresu 10.1.1.100/24, Gateway 10.1.1.1 i DNS 193.198.97.4
Mrežni usmjerivač	Usmjerava promet između lokalne i javne mreže. Prema lokalnoj mreži ima stalnu IP adresu 10.1.1.1/24. Prema javnoj mreži ima stalnu IP adresu 193.198.235.206/16
DNS poslužitelj	Ima stalnu IP adresu 193.198.97.4/16 i Gateway 193.198.235.206. Rješava adresu www.veleri.hr u 193.198.97.6/16.
Poslužitelj raspoređivač 1	Ima stalnu IP adresu 193.198.97.1/16 i Gateway 193.198.235.206. Njegova VIP (virtualna IP, engl. virtual IP) adresa je 193.198.97.6/16
Poslužitelj raspoređivač 2	Ima stalnu IP adresu 193.198.97.2/16 i Gateway 193.198.235.206. Njegova VIP (virtualna IP, engl. virtual IP) adresa je 193.198.97.6/16
web-poslužitelj 1	Ima stalnu IP adresu 193.198.97.10/16.
web-poslužitelj 2	Ima stalnu IP adresu 193.198.97.11/16.

Izvor: obrada autora

Za emulaciju topologije računalne mreže s računalnim klasterom, potrebno je ukupno sedam emuliranih računala (slika 1).

Slika 1. Emulirana računala u emulatoru VirtualBox



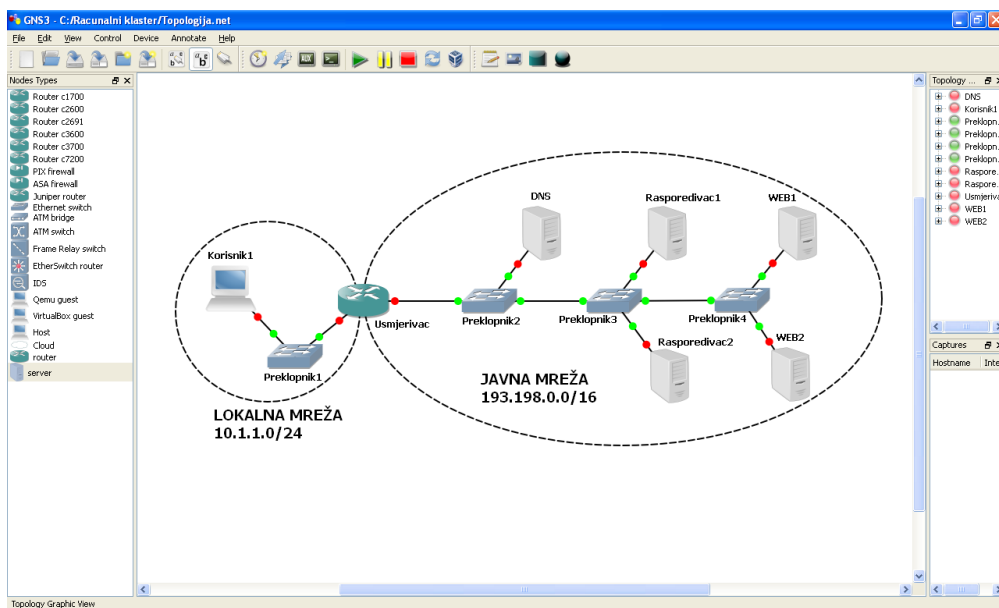
Izvor: obrada autora

korisničko računalo je emulirano s operacijskim sustavom Linux TinyCore na kojem se nalazi internetski preglednik Mozilla FireFox. Mrežni usmjerivač također je emuliran s operacijskim sustavom Linux TinyCore unutar kojega je uključena opcija IP prosljeđivanja (engl. *IP forwarding*) definirana tablicom usmjeravanja (engl. *IP routing table*).

DNS poslužitelj implementiran je preko računalnog programa Dnsmasq koji je instaliran na emulirano računalo s Linux TinyCore operacijskim sustavom. Isti operacijski sustav imaju i računala koja predstavljaju *web*-poslužitelje koji su implementirani s računalnim programom Apache HTTP Server. Poslužitelji raspoređivači emulirana su računala s Linux Debian operacijskim sustavom i instaliranim računalnim programima HAProxy i Heartbeat.

Emulirana topologija računalne mreže u emulatoru GNS3 prikazana je na slici 2. Crvene točke na krajevima nekih veza među mrežnim uređajima ukazuju na to da ti mrežni uređaji nisu pokrenuti. Pokretanje emulirane računalne mreže (posebna ikona u alatnoj traci), uzrokuje podizanje emuliranih računala. Nakon podizanja, u pojedinom emuliranom računalu postavljaju se odgovarajuće IP adrese (tablica 1) kako bi potrebna računalna mreža bila uspostavljena. U DNS poslužitelju postavlja se rješavanje adrese *www.veleri.hr* u IP adresu 193.198.97.6. Posebnu pažnju treba usmjeriti na postavljanje poslužitelja raspoređivača i vezano uz njih *web*-poslužitelja. Također treba biti pažljiv u postavljanju sustava za povećavanje dostupnosti poslužitelja raspoređivača. Postavljanje raspoređivača i povećanje njegove dostupnosti detaljnije se opisuje u posebnom poglavlju.

Slika 2. Emulirana topologija računalne mreže u emulatoru GNS3



Izvor: obrada autora

Kada se na korisničkom računalu u internetskom pregledniku zahtijeva *web* mrežni servis [www.veleri.hr](http://www.veleri.hr), dostavlja se zahtjev DNS poslužitelju za rješavanje te adrese u IP adresu. DNS poslužitelj vraća IP adresu 193.198.97.6 na koju se zatim upućuje zahtjev za *web* mrežnim servisom. Ovaj zahtjev prihvaća aktivni poslužitelj raspoređivač, te ga dostavlja nekom od *web*-poslužitelja. *Web*-poslužitelj dostavlja zatraženi *web* mrežni servis poslužitelju raspoređivaču, koji ga prosljeđuje korisničkom računalu. Zbog postavljenog sustava povećanja dostupnosti na poslužiteljima raspoređivačima, u slučaju ispada aktivnog raspoređivača, njegovu ulogu preuzima sekundarni raspoređivač. U slučaju ispada jednog *web*-poslužitelja, aktivni raspoređivač će sve zahtjeve usmjeriti na preostali *web*-poslužitelj.

### 5.1 Konfiguracija poslužitelja raspoređivača i sustava za povećanje njihove dostupnosti

Za uspostavu poslužitelja raspoređivača, implementiranog računalnim programom HAProxy, potrebno je pripremiti konfiguracijsku datoteku `/etc/haproxy/haproxy.cfg`. Tablica 2 prikazuje sadržaj primijenjene konfiguracijske datoteke.

Tablica 2. Sadržaj konfiguracijske datoteke `haproxy.cfg`

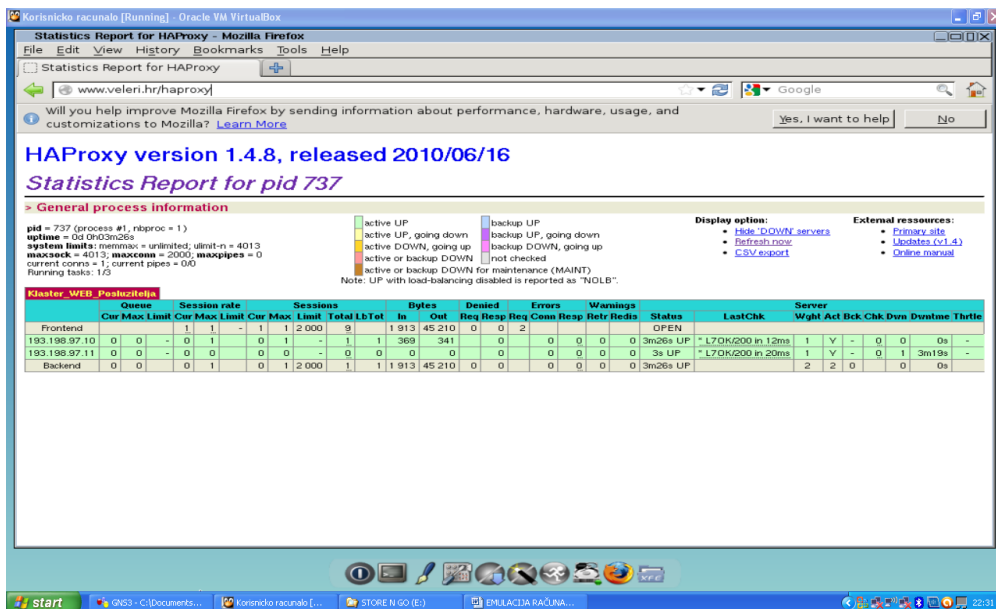
R.Br.	Sadržaj
1	global
2	daemon
3	
4	defaults
5	mode                http
6	retries3
7	option redispatch
8	maxconn2000
9	contimeout        5000
10	clitimeout        50000
11	srvtimeout        50000
12	
13	listen Klaster_WEB_Poslužitelja        193.198.97.6:80
14	stats                enabled
15	stats uri            /haproxy
16	balanceroundrobin
17	cookie JSESSIONID prefix
18	option httpclose
19	option forwardfor
20	option httpchk HEAD /haproxytest.txt HTTP/1.0
21	server 193.198.97.10 193.198.97.10:80 cookie A check
22	server 193.198.97.11 193.198.97.11:80 cookie B check

Izvor: obrada autora

U liniji 13 konfiguracijske datoteke navodi se naziv klastera (Klaster\_WEB\_Poslužitelj), te virtualna IP adresa (193.198.97.6) preko koje će korisnička računala pristupati *web* mrežnom servisu. U

liniji 14 i 15 navodi se je li dostupna statistika poslužitelja raspoređivača i kako se kroz internetski preglednik dolazi do *web*-stranice s osnovnim statističkim podacima o *web*-poslužiteljima. Prema prikazanoj postavki, do statističkih podataka dolazi se tako da se adresi mrežnog servisa doda /haproxy (odnosno [www.veleri.hr/haproxy](http://www.veleri.hr/haproxy)) - slika 3. Moguće je uključiti opciju obavezne korisničke prijave za pregled statističkih podataka.

Slika 3. Statistički podaci za uspostavljeni računalni klaster



Izvor: obrada autora

Linija 20 konfiguracijske datoteke prikazuje naziv datoteke preko koje poslužitelj raspoređivač vrši provjeru je li neki *web*-poslužitelj dostupan. Ova datoteka se mora nalaziti unutar mape *web*-poslužitelja gdje se nalazi početna *web*-stranica. Tako je i za prikazani primjer izrađena datoteka *haproxytest.txt* unutar oba *web*-poslužitelja. Linije 21 i 22 prijavljuju *web*-poslužitelje koje raspoređivač treba provjeravati i koristiti.

Kako bi se računalni program koji implementira poslužitelj raspoređivač pokrenuo pri pokretanju računala, potrebno je u datoteci */etc/default/haproxy* postaviti *ENABLED=1*. Ponovno pokretanje računalnog programa HAProxy (bez ponovnog pokretanja računala) vrši se s */etc/init.d/haproxy restart*. Budući da poslužitelji raspoređivači trebaju imati dvije IP adrese na mrežnom sučelju (stvarnu i virtualnu) u datoteci */etc/sysctl.conf* potrebno je dodati *net.ipv4.ip\_nonlocal\_bind=1*.

Za postavljanje povećane dostupnosti poslužitelja raspoređivača (implementirano s računalnim programom Heartbeat), na njima je potrebno konfigurirati tri datoteke: */etc/ha.d/authkeys*, */etc/ha.d/haresources* i */etc/ha.d/ha.cf*.

Tablica 3. Sadržaj konfiguracijske datoteke *authkeys*

R.Br.	Sadržaj		
1	auth	1	
2	1	md5	VELERI

Izvor: obrada autora

Konfiguracijska datoteka *authkeys* koristi se za autentifikaciju računala u klasteru. Sadržaj ove datoteke prikazan je u tablici 3. U liniji 2 definira se algoritam kriptiranja ključa koji se koristi prilikom komunikacije između dvaju računala (u prikazanom primjeru između dva poslužitelja raspoređivača), te sam ključ. Oba poslužitelja raspoređivača moraju imati identičnu konfiguracijsku datoteku. Ona treba biti čitljiva samo *root* korisniku i zbog toga joj treba promijeniti svojstva naredbom *chmod 600 /ect/ha.d/authkeys*.

Tablica 4. Sadržaj konfiguracijske datoteke *haresources*

R.Br.	Sadržaj	
1	rasporedivac1	193.198.97.6

Izvor: obrada autora

Tablica 4 prikazuje sadržaj konfiguracijske datoteke *haresources* koja mora biti identična na oba poslužitelja raspoređivača. Njezina svrha je definiranje preferiranog (primarnog) računala koje će pružati mrežni servis na određenoj IP adresi. U primjeru je to poslužitelj raspoređivač 1. To znači da, kada su oba poslužitelja u funkciji, na zahtjeve korisničkih računala koji su poslani na IP adresu 193.198.97.6 odgovara samo raspoređivač 1.

Tablica 5. Sadržaj konfiguracijske datoteke *ha.cf*

R.Br.	Sadržaj	
1	keepalive	2
2	deadtime	10
3	bcast	eth0
4	node	rasporedjivac1
5	node	rasporedjivac2

Izvor: obrada autora

Linija 1 konfiguracijske datoteke *ha.cf* (tablica 5) definira vrijeme između dvije provjere računala (u primjeru, poslužitelj raspoređivač 1 provjerava svake dvije sekunde rad poslužitelja raspoređivača 2). Linija 2 definira koliko dugo će se čekati drugo računalo prije nego se odluči da se dogodio njegov ispad (u primjeru, poslužitelj raspoređivač 1 čeka deset sekundi odgovor poslužitelja raspoređivača 2, nakon toga smatra da se dogodio njegov ispad). Linija 3 definira mrežno sučelje preko kojega će se vršiti difuzija. Posljednje dvije linije konfiguracijske datoteke definiraju (preko naziva) koja računala su u računalnom

klasteru. Budući da se koriste nazivi računala, oni trebaju biti definirani u konfiguracijskoj datoteci */etc/hostname* za svaki poslužitelj raspoređivač. Konfiguracijska datoteka *ha.cf* treba se nalaziti na svakom raspoređivaču.

## 6. ZAKLJUČAK

Opisana metodika emulacije računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti mrežnih servisa uspješno je testirana na praktičnoj emulaciji računalnog klastera za *web* mrežni servis. Provjerena je funkcionalnost raspoređivanja opterećenja, te funkcionalnost povećane dostupnosti *web*-poslužitelja i poslužitelja raspoređivača. Također se provjerila mogućnost analize paketa koji kolaju računalnom mrežom, te mogućnost spremanja izrađene emulacije i njezinog ponovnog pokretanja. Temeljem rezultata koji su proizašli iz emuliranoga računalnog klastera može se zaključiti da je izrada laboratorija za emulaciju računalnog klastera za raspoređivanje opterećenja i povećanje dostupnosti izvediva, ali samo onih za *web* mrežni servis.

Postoji stoga potreba za daljnjom provjerom metodike kroz praktičnu uspostavu računalnih klastera za niz drugih mrežnih servisa, kao što su FTP, DNS, *e-mail*, VPN, VoIP itd., ali i onih koji su izrađeni za neko specifično aplikacijsko područje (npr. *web* aplikacija, mrežni servisi za neku poslovnu funkciju i sl.).

Nakon provjere izvedivosti emulacije temeljne infrastrukture računalnoga klastera za razne mrežne servise, treba uslijediti provjera njegove primjenjivosti u edukaciji, te izgradnji testne okoline koju bi poduzeća mogla koristiti za provjeru implementacije računalnog klastera, isprobavanje novih konfiguracija, novorazvijenih računalnih programa i protokola, te novih proizvoda i usluga.

Posebnu pažnju treba usmjeriti na dobro definiranje potrebnih emuliranih računala koja trebaju imati svojstvo da zahtijevaju minimalne resurse računala domaćina (prije svega radne memorije). Isto tako, pažnju treba usmjeriti i na mogućnost povezivanja emuliranog računalnog klastera sa stvarnom fizičkom računalnom mrežom, ali i onom koja je također emulirana. Temeljni element laboratorija za emulaciju računalnog klastera je računalo domaćin - računalo na kojem se odvija cjelokupna emulacija. Stoga je važno analizirati koja svojstva treba imati ovo računalo, kako bi na njemu implementiran laboratorij imao zadovoljavajuće performanse.

U prikazanoj emulaciji računalnog klastera za raspoređivanje opterećenja i povećanje pouzdanosti za *web* mrežni servis korišteni su računalni programi VirtulaBox, GNS3, HAProxy, Heartbeat i Apache HTTP Server. Pitanje koje se postavlja je postoji li alternativa za ove računalne programe, čime se otvara pravac za njihovu komparativnu analizu. Ta alternativa treba biti besplatna i eventualno treba biti otvorenog koda. Općenito, računalni programi otvorenoga koda pružaju mogućnost prilagodbe posebnim primjenama, pa se time otvara i pravac istraživanja prilagodbe računalnih programa za emulaciju.



## LITERATURA

- Adelstein, T., Lubanovic, B. (2007) *Administriranje Linux sustava*, Zagreb: Dobar Plan
- Amdahl, G. (1967) "Validity of the single processor approach to achieving large scale computing capabilities". U: AFIPS '67 (Spring) *Proceedings of the April 18-20, 1967, spring joint computer conference*. New York: ACM, p. 483-485
- Buyya, R. (1999). *High Performance Cluster Computing: Architectures and Systems*. New York: Prentice Hall
- CARNet (2006) "Cluster sustavi", <http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2006-12-176.pdf> (24. 12. 2012.)
- D'Orazio, D. (2012) "Gmail now has 425 million active users", <http://www.theverge.com/2012/6/28/3123643/gmail-425-million-total-users> (24. 12. 2012.)
- Fox, et al. (1997) "Cluster-Based Scalable Network Services", *ACM SIGOPS Operating Systems Review*, 31/5, p. 78-91.
- Jakupović, A., Knežević, M. (2012). "Emulacija računalne mreže primjenom računalnih programa GNS3 i VirtualBox". U: *KOM 2012 - Elektroničke komunikacijske tehnologije i norme u informatici* 26.11. - 27.11.2012. Rijeka: CASE, p. 105-112.
- Jakupović, A., Pirić, E. (2013). "Laboratorij za emulaciju računalnih mreža i primjena u praktičnoj nastavi", u postupku objave na MIPRO 2013.
- Kopparapu, C. (2002). *Load Balancing Servers, Firewalls, and Caches*. New York: John Wiley & Sons
- Pfister, G. F. (1998) *In Search of Clusters*. New York: Prentice Hall
- Sommerville, I. (2007) *Software Engineering*. Harlow: Pearson Education
- Zec, M. (2012) "Simulacija i emulacija računalnih mreža i raspodijeljenih sustava primjenom virtualizacijskih mehanizama", [http://www.fer.unizg.hr/\\_download/repository/Zec\\_KDI.pdf](http://www.fer.unizg.hr/_download/repository/Zec_KDI.pdf) (24. 12. 2012.)

## EMULATION OF THE LOAD BALANCING AND HIGH AVAILABILITY COMPUTER CLUSTER FOR NETWORK SERVICES<sup>3</sup>

### ABSTRACT

*The paper presents the possibility of establishing an emulation of the load balancing and high availability computer cluster for network service. The basic objective of establishing the emulated computer cluster is checking the feasibility of building laboratory for the cluster emulation. This laboratory can be used for different system testing, for analysing the system components data flows, and for trying out different solutions prior to their actual implementation. This paper describes the methodology of computer cluster emulation using computer emulation via software VirtualBox, computer network emulation via software GNS3, load balancing software HAProxy and high availability software Heartbeat. The paper shows the practical application of the described methodology in emulation of load balancing and high availability computer cluster for web network service which is implemented with software Apache HTTP Server. The paper also describes in detail a high availability computer cluster, load balancing computer cluster and their combination and shows the reasons why it is desirable to establish a laboratory for the emulation of computer cluster. In conclusion, the paper argues the feasibility of laboratory for the emulation of computer cluster, and indicates further development.*

**Key words:** emulation, computer cluster, network service, load balancing system, high availability system

<sup>1</sup> PhD, Assistant Professor, Principal Lecturer, Polytechnic of Rijeka, Vukovarska 58, Rijeka, Hrvatska. E-mail: alen.jakupovic@veleri.hr

<sup>2</sup> Student, BSc of Informatics, System Administrator, Infobip d.o.o. Mletačka 12/III, Pula, Hrvatska. E-mail: petar.koraca@veleri.hr

<sup>3</sup> Received: 14. 1. 2013; accepted: 3. 4. 2013

