Zadatak 1: Web aplikacija uključuje podršku korisnicima putem chat usluge. Kupci sami odabiru jedan od 10 repova čekanja u kojima upite poslužuje po jedan tehničar. Mjerenja pokazuju da zahtjevi prosječno dolaze 3 upita u minuti te da svaki kupac prosječno čeka 3 minute u repu i prosječno provodi 2 minute u razgovoru. *(nadogradnja primjera 6)*

1) Kakvi će biti odzivi sa 10 i 18 tehničara ako publiciranje Web stranice sa odgovorima na najčešća pitanja smanji broj upita na 2 u minuti?

$N_A = 10;\ N_B = 18$

$L = 2\ z/min$

$S = 2\ min/z$

$U = L \cdot S = 4$

$\rho_A = \frac{U}{N_A} = 0{,}4;\ \rho_B = \frac{U}{N_B} = 0{,}22$

$R_A = \frac{S}{1-\rho_A} = 3{,}33\ min/z;\ R_B = \frac{S}{1-\rho_B} = 2{,}57\ min/z$

$W_A = R_A - S_A = 1{,}33\ min/z;\ W_B = R_B - S_B = 0{,}57\ min/z$

Smanjit će se čekanje (odziv će biti brži proporcionalno smanjenju srednjeg vremena zadržavanja u sustavu i/ili povećanju broja kanala).

2) Kakve će rezultate dati smanjenje razgovora na 1.5 minutu?

$N = 10$

$L = 3\ z/min$

$S = 1{,}5\ min/z$

$U = L \cdot S = 4{,}5$

$\rho = \frac{U}{N} = 0{,}45$

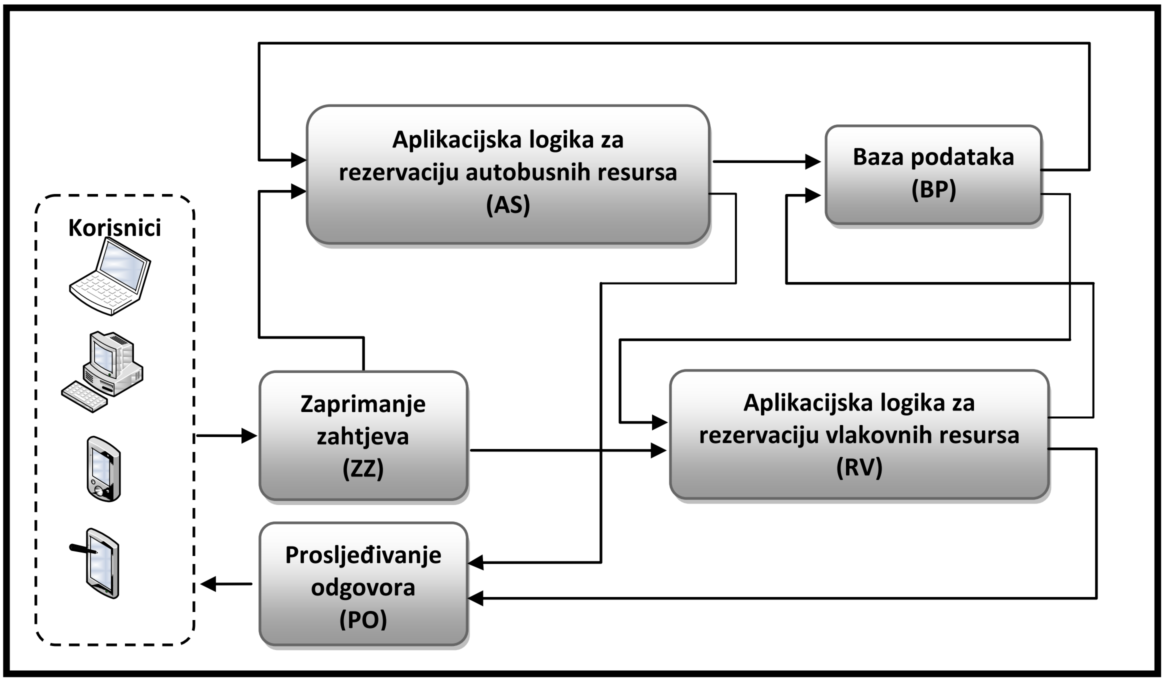
$R = \frac{S}{1-\rho} = 2{,}73\ min/z$

$W = R - S = 1{,}23\ min/z$

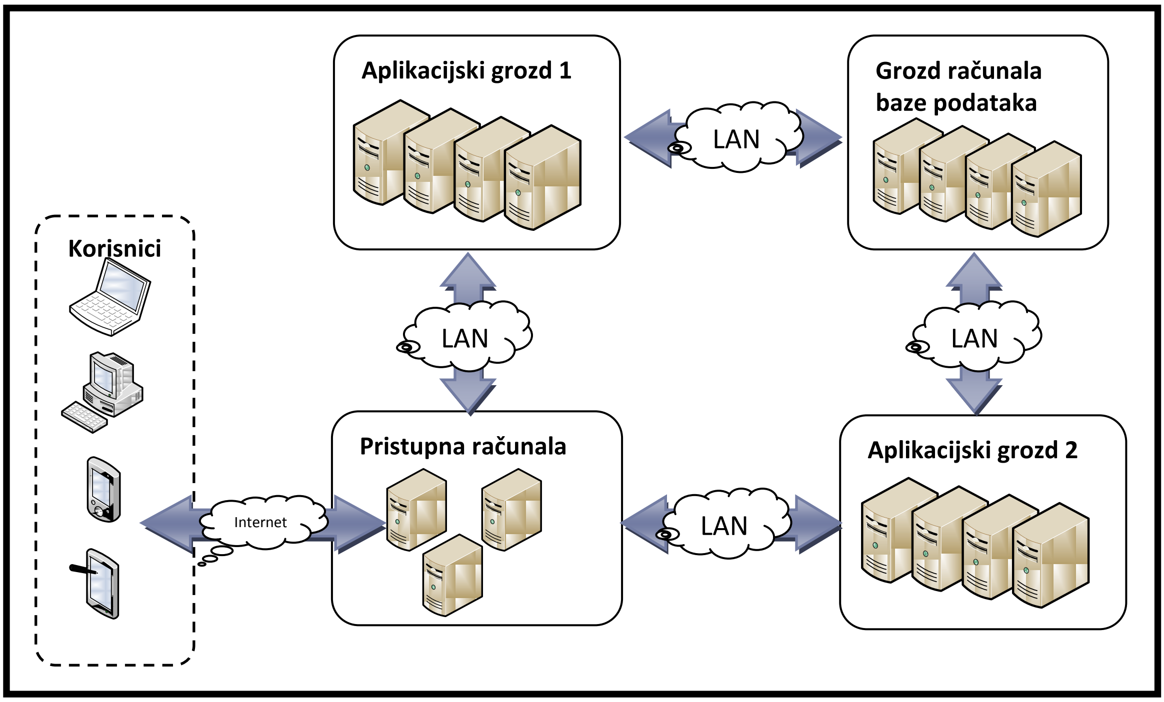
Smanjit će se čekanje (odziv će biti brži proporcionalno smanjenju srednjeg vremena zadržavanja u sustavu smanjenjem trajanja fiksne komponente).

Zadatak 2: Oblikovati proizvoljnu raspodijeljenu aplikaciju i ostvariti analizu performansi ostvarene aplikacije

1) Definirati logičku i fizičku arhitekturu aplikacije



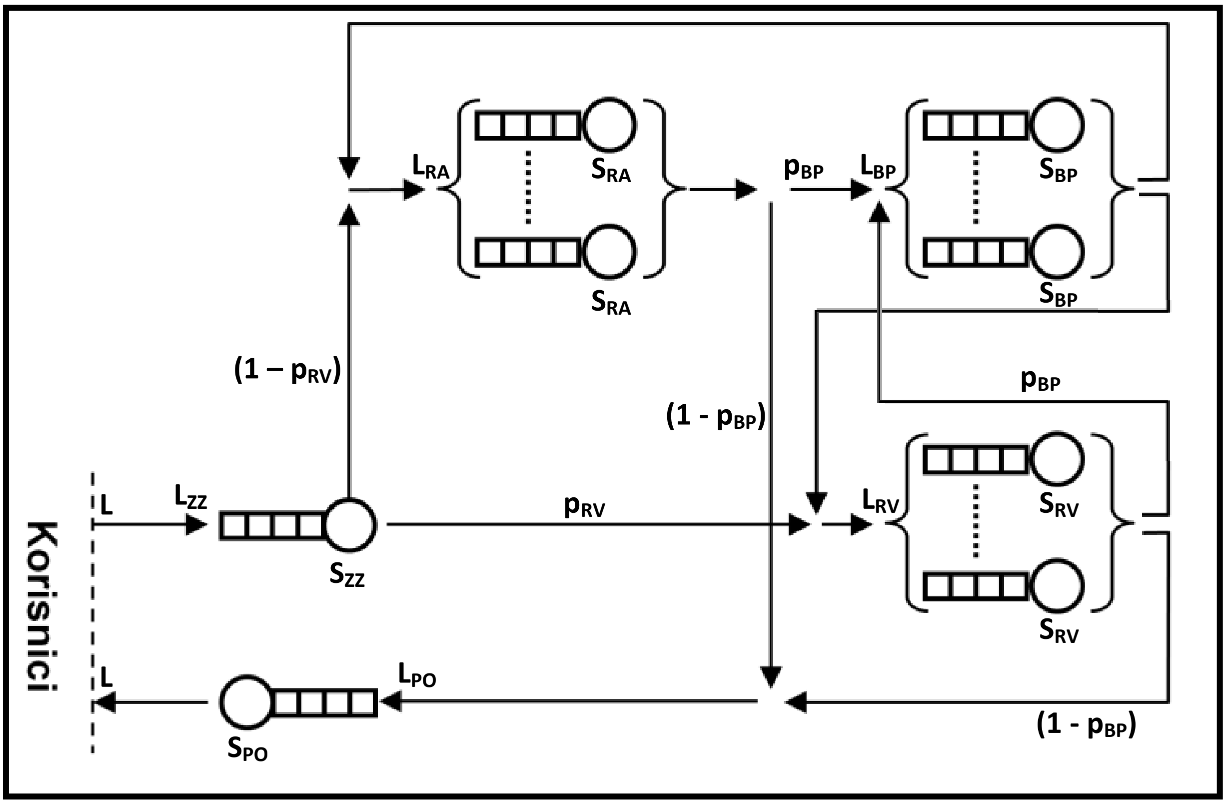
Slika 1 Logička arhitektura aplikacije



Slika 2 Fizička arhitektura aplikacije

2) Izgraditi model aplikacije primjenom teorije repova

- Odrediti analitičko rješenje funkcije zadržavanja zahtjeva u aplikaciji R=f(L)



Slika 3 Model aplikacije

Analiza:

$$L_{RV} = p_{BP} \cdot L_{RV} + p_{RV} \cdot L = \frac{p_{RV}}{1 - p_{BP}} \cdot L \qquad v_{RV} = \frac{p_{RV}}{1 - p_{BP}}$$

$$L_{RA} = p_{BP} \cdot L_{RA} + (1 - p_{RV}) \cdot L = \frac{1 - p_{RV}}{1 - p_{BP}} \cdot L \qquad v_{RA} = \frac{1 - p_{RV}}{1 - p_{BP}}$$

$$L_{BP} = p_{BP} \cdot (L_{RV} + L_{RA}) = \frac{p_{BP}}{1 - p_{BP}} \cdot L \qquad v_{BP} = \frac{p_{BP}}{1 - p_{BP}}$$

$$L_{ZZ} = L \qquad v_{ZZ} = 1$$

$$L_{PO} = L \qquad v_{PO} = 1$$

$$D_{RV} = v_{RV} \cdot S_{RV} = \frac{p_{RV}}{1 - p_{BP}} \cdot S_{RV}$$

$$D_{RA} = v_{RA} \cdot S_{RA} = \frac{1 - p_{RV}}{1 - p_{BP}} \cdot S_{RA}$$

$$D_{BP} = v_{BP} \cdot S_{BP} = \frac{p_{BP}}{1 - p_{BP}} \cdot S_{BP}$$

$$D_{ZZ} = v_{ZZ} \cdot S_{ZZ} = S_{ZZ}$$

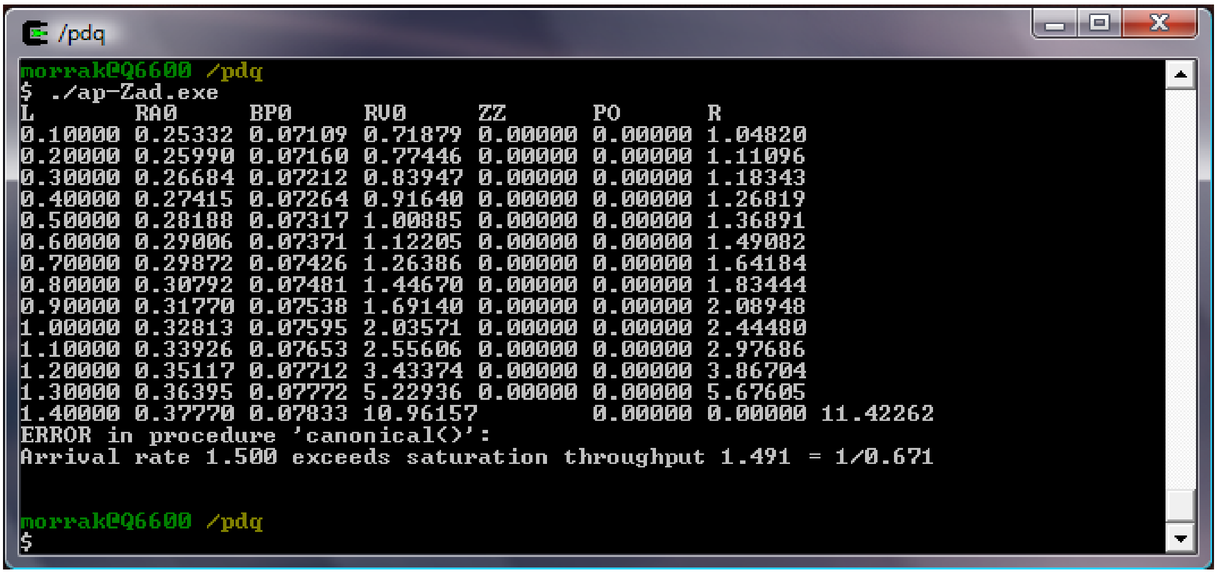
$$D_{PO} = v_{PO} \cdot S_{PO} = S_{PO}$$

$$R = \frac{D_{RV}}{1 - L \cdot D_{RV}} + \frac{D_{RA}}{1 - L \cdot D_{RA}} + \frac{D_{BP}}{1 - L \cdot D_{BP}} + \frac{D_{ZZ}}{1 - L \cdot D_{ZZ}} + \frac{D_{PO}}{1 - L \cdot D_{PO}}$$

3) Izgraditi model aplikacije za alat PDQ

- Primjenom izgrađenog modela odrediti vrijednosti funkcije zadržavanja zahtjeva R=f(L) u nekoliko točaka

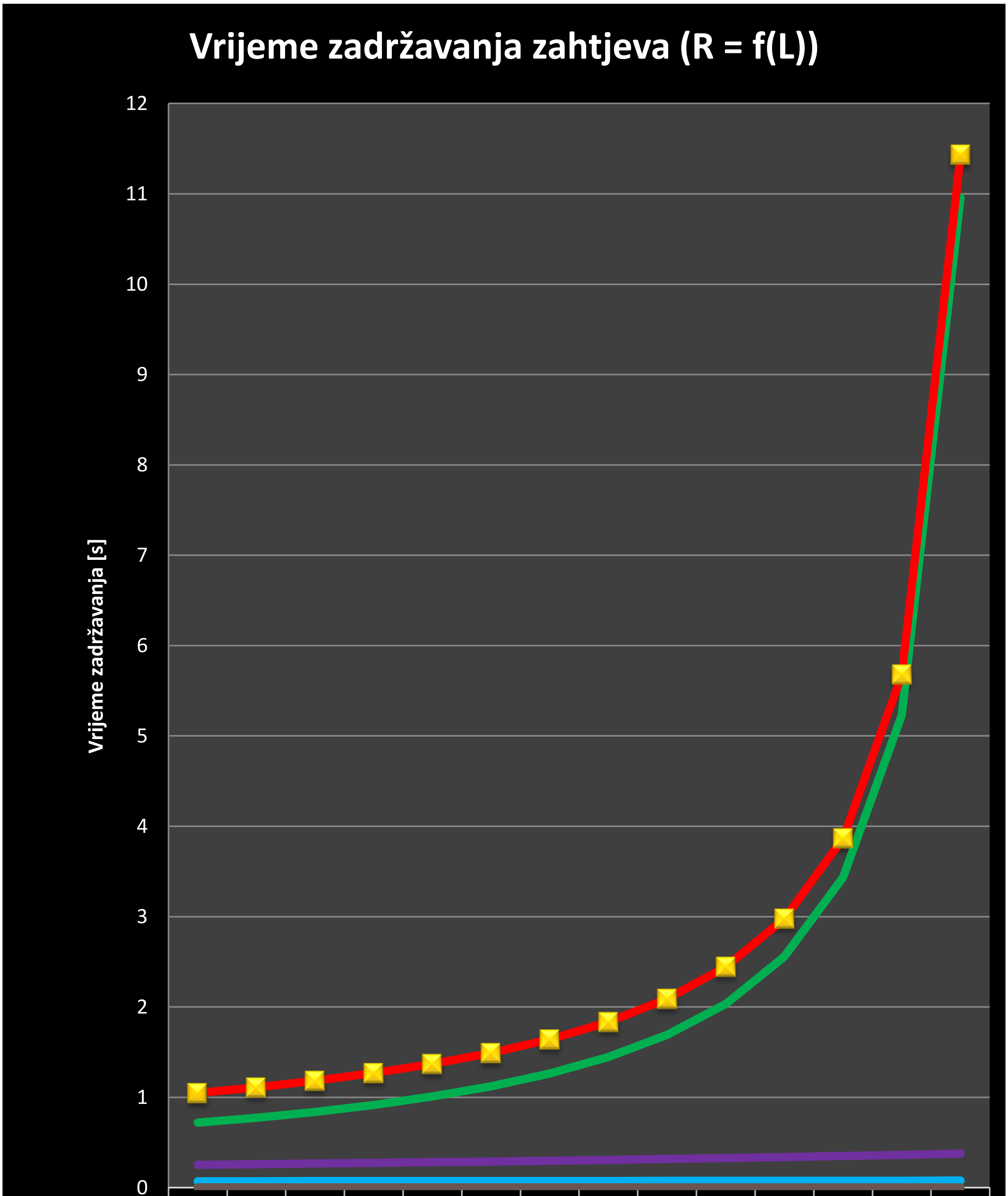
Implementacija ranije navedenog modela aplikacije za PDQ nalazi se u prilogu A. Na Slika 4 je prikazan ispis izvođenja u emulatoru Cygwin, dok su na Graf 1 prikazani rezultati tablično i grafički. Prati se pojedino (po podsustavima) i sumarno srednje vrijeme zadržavanja u sustavu u ovisnosti o učestalosti zahtjeva. Tablica sadrži i Sumarno srednje vrijeme zadržavanja u sustavu dobiveno analitičkim putem.



```
/pdq
morrak@Q6600 /pdq
$ ./ap-Zad.exe
L         RA0       BP0       RU0       ZZ        PO        R
0.10000 0.25332 0.07109 0.71879 0.00000 0.00000 1.04820
0.20000 0.25990 0.07160 0.77446 0.00000 0.00000 1.11096
0.30000 0.26684 0.07212 0.83947 0.00000 0.00000 1.18343
0.40000 0.27415 0.07264 0.91640 0.00000 0.00000 1.26819
0.50000 0.28188 0.07317 1.00885 0.00000 0.00000 1.36891
0.60000 0.29006 0.07371 1.12205 0.00000 0.00000 1.49082
0.70000 0.29872 0.07426 1.26386 0.00000 0.00000 1.64184
0.80000 0.30792 0.07481 1.44670 0.00000 0.00000 1.83444
0.90000 0.31770 0.07538 1.69140 0.00000 0.00000 2.08948
1.00000 0.32813 0.07595 2.03571 0.00000 0.00000 2.44480
1.10000 0.33926 0.07653 2.55606 0.00000 0.00000 2.97686
1.20000 0.35117 0.07712 3.43374 0.00000 0.00000 3.86704
1.30000 0.36395 0.07772 5.22936 0.00000 0.00000 5.67605
1.40000 0.37770 0.07833 10.96157        0.00000 0.00000 11.42262
ERROR in procedure 'canonical()':
Arrival rate 1.500 exceeds saturation throughput 1.491 = 1/0.671


morrak@Q6600 /pdq
$
```

Slika 4 Primjer izvođenja modela u programu PDQ



	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	1,1	1,2	1,3	1,4
RV	0,253	0,259	0,266	0,274	0,281	0,290	0,298	0,307	0,317	0,328	0,339	0,351	0,363	0,377
RA	0,071	0,071	0,072	0,072	0,073	0,073	0,074	0,074	0,075	0,075	0,076	0,077	0,077	0,078
BP	0,718	0,774	0,839	0,916	1,008	1,122	1,263	1,446	1,691	2,035	2,556	3,433	5,229	10,96
ZZ	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002
PO	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003	0,003
R	1,048	1,110	1,183	1,268	1,368	1,490	1,641	1,834	2,089	2,444	2,976	3,867	5,676	11,42
R(PDQ)	1,048	1,110	1,183	1,268	1,368	1,490	1,641	1,834	2,089	2,444	2,976	3,867	5,676	11,42

Graf 1 Vrijeme zadržavanja zahtjeva u sustavu

4) Usporediti i obrazložiti dobivene rezultate

Rezultati su (na 4 decimale) identični što pokazuje kako je PDQ model ispravno zadan i izveden. Prema rezultatima za raspodijeljeni sustav kojem je svaka komponenta jedno računalo zaključujem kako dani sustav uz navedene parametre prihvatljivo funkcionira do učestalosti zahtjeva od 1,1 s^{-1}. Od tada pa do granice od 1,4 zahtjeva u sekundi performanse eksponencijalno padaju, dok nakon navedene granice priljev zahtjeva zasićuje sustav te ga isti više nije u stanju obrađivati u realnom vremenu. Sustav bi tada trebalo nadograditi kako bi se povećala propusnost i omogućio rad pri povećanom naporu.