



Diplomski studij

**Informacijska i
komunikacijska tehnologija:**

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i
informacijski sustavi

Računarska znanost

Ak.g. 2009./2010.

Raspodijeljeni sustavi

11.

Primjena alata za vrednovanje performansi raspodijeljenih aplikacija

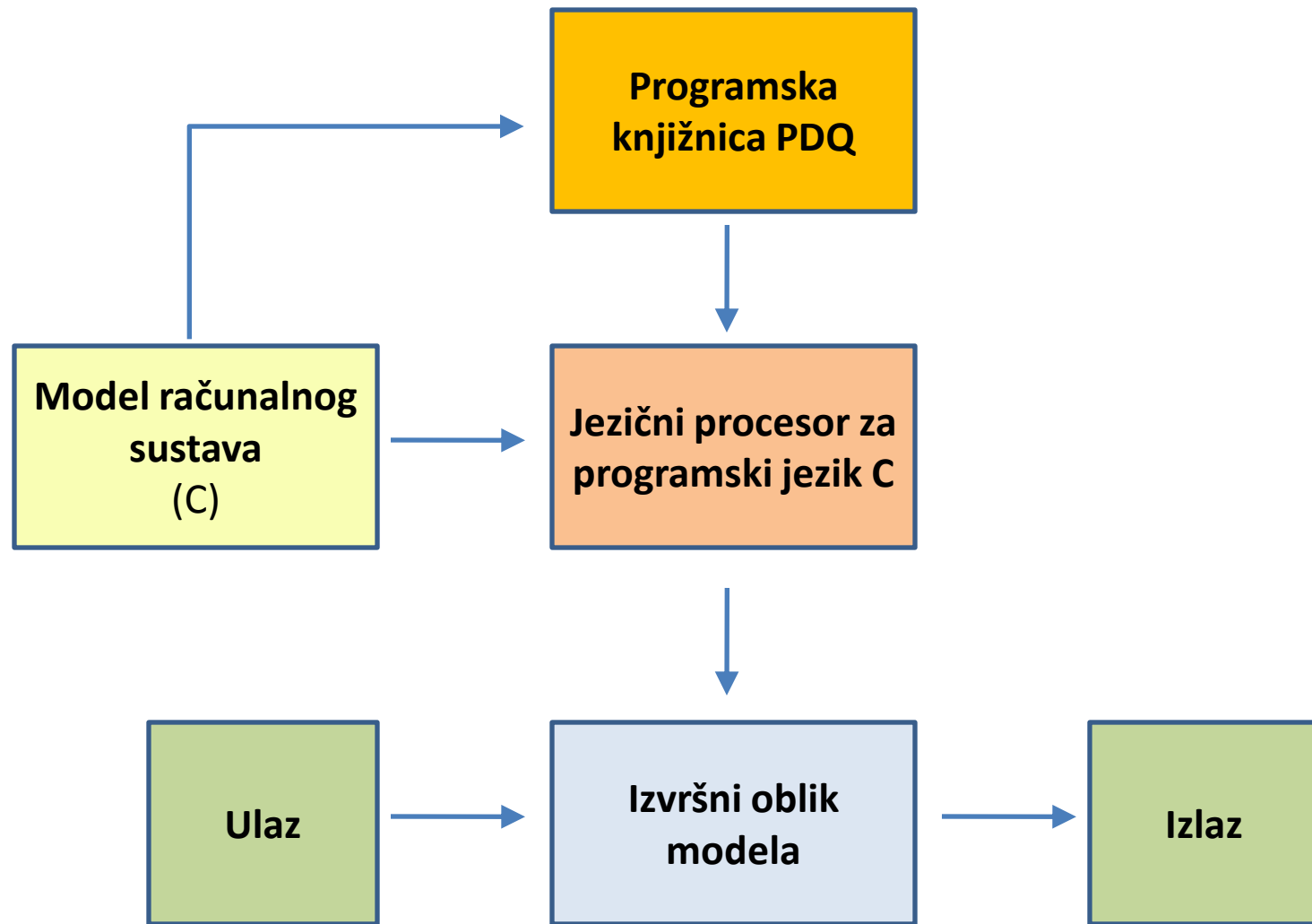
Dr. Dalibor F. Vrsalović
dalibor.f.vrsalovic@fer.hr

15.12.2009.

- ◆ **I dio: Alat PDQ (Pretty Damn Quick)**
- ◆ **II dio: Primjeri uporabe alata PDQ**
- ◆ **III dio: Primjer analize performansi web aplikacije**
- ◆ **IV dio: Domaća zadaća**

Alat PDQ (Pretty Damn Quick)

- ◆ **Omogućava izgradnju modela za vrednovanje performansi računalni sustava**
- ◆ **Modeli se grade primjenom načela teorije redova**
- ◆ **Značajke modela izračunavaju s primjenom analitičkih postupka i algoritama**
- ◆ **Dodatne informacije**
<http://www.perfdynamics.com/Tools/PDQ.html>

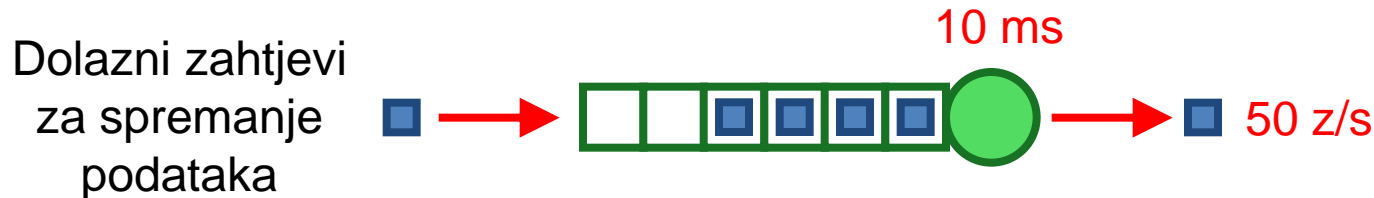


Primjeri uporabe alata PDQ

Primjer 1: Posluživanje zahtjeva na disku



- ◆ Disk za trajno spremanje podataka obrađuje **50 zahtjeva u sekundi**. Srednje vrijeme obrade zahtjeva operacija pisanja i čitanja je **10 ms**.
- ◆ Kolika je prosječna zaposlenost diska?



◆ Analitičko rješenje

◆ Propusnost sustava $X = 50 \text{ z/s}$

◆ Srednje vrijeme obrade zahtjeva $S = 10 \text{ ms/z}$

◆ Prosječna zaposlenost diska U

$$U = X * S = 50 \text{ z/s} * 0.01 \text{ s/z} = 0.5 (50 \%)$$

Primjer 1: Posluživanje zahtjeva na disku



```
main() {  
    extern int nodes, streams;  
  
    float L = 50;  
    float S = 0.01;  
  
    PDQ_Init("Diskovni podsustav");  
  
    nodes = PDQ_CreateNode("Posluzitelj", CEN, FCFS);  
    streams = PDQ_CreateOpen("Operacije", L);  
  
    PDQ_SetDemand("Posluzitelj", "Operacije", S);  
    PDQ_Solve(CANON);  
    PDQ_Report();  
}
```

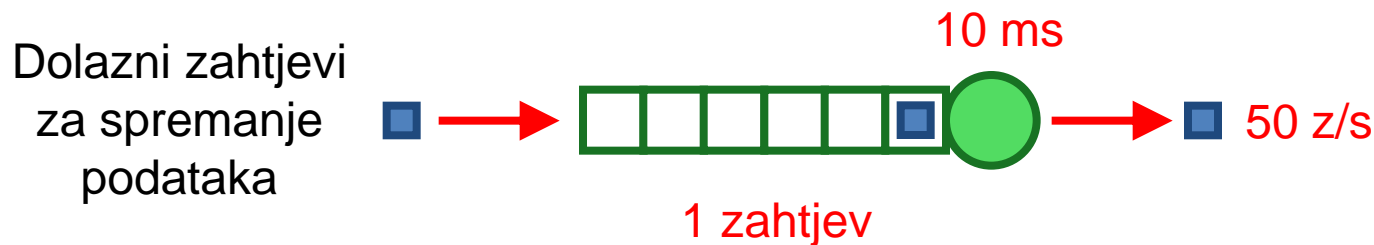


pr1.c

Primjer 2: Čekanje na posluživanje zahtjeva s diska



- ◆ Disk iz prethodnog slučaja ima prosječno **1 zahtjev u repu**
 - ◆ Koliko je prosječno vrijeme čekanja na obradu zahtjeva ?



◆ Analitičko rješenje

◆ Ulazni ritam zahtjeva $L = 50 \text{ z/s}$

◆ Broj zahtjeva u repu $Q = 1 \text{ z}$

◆ Vrijeme zadržavanja zahtjeva u sustavu R

$$R = Q/L = (1 \text{ z}) / (50 \text{ z/s}) = 20 \text{ ms}$$

◆ Vrijeme zadržavanja uključuje vrijeme čekanja u repu (W) i vrijeme obrade zahtjeva (S): $R = W + S$

◆ Vrijeme čekanja na obradu W

$$W = R - S = 20 \text{ ms} - 10 \text{ ms} = 10 \text{ ms}$$

Primjer 2: Čekanje na posluživanje zahtjeva s diska



```
main() {  
    extern int nodes, streams;  
  
    float L = 50;  
    float S = 0.01;  
  
    PDQ_Init("Diskovni podsustav");  
  
    nodes = PDQ_CreateNode("Posluzitelj", CEN, FCFS);  
    streams = PDQ_CreateOpen("Operacije", L);  
  
    PDQ_SetDemand("Posluzitelj", "Operacije", S);  
  
    PDQ_Solve(CANON);  
    PDQ_Report();  
}
```

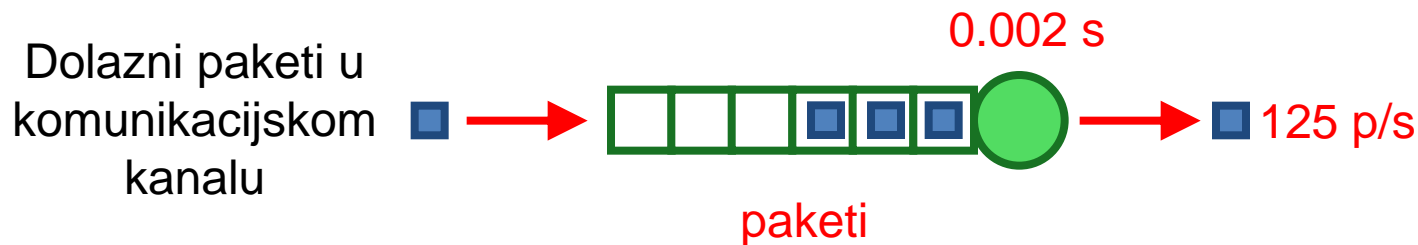


pr2.c

Primjer 3: Komunikacijski kanal



- ◆ Mjerenjem na pristupnoj točki mreže dobivamo **srednji protok od 125 paketa u sekundi** i **srednje vrijeme posluživanja 0.002 sekunde**.
- ◆ Što je sve moguće zaključiti o promatranom kanalu ?



◆ Analitičko rješenje

◆ Srednji protok $X = 125$ p/s

◆ Srednje vrijeme posluživanja $S = 0.002$ s/p

◆ Prosječna zaposlenost komunikacijskog sustava U

$$U = X * S = (125 \text{ p/s}) * (0.002 \text{ s/p}) = 0.25 \text{ (25 \%)}$$

◆ Srednje vrijeme zadržavanja paketa u sustavu (R)

$$R = S / (1 - U) = (0.002 \text{ s/p}) / (1 - 0.25) = 0.0026666 \text{ s}$$

◆ Srednji broj paketa u repu (Q)

$$Q = X * R = (125 \text{ p/s}) * (0.0026 \text{ s}) = 0.333 \text{ p}$$

Primjer 3: Komunikacijski kanal



```
main() {  
    extern int nodes, streams;  
  
    float L = 125;  
    float S = 0.002;  
  
    PDQ_Init("Mrezni podsustav");  
  
    nodes = PDQ_CreateNode("Posluzitelj", CEN, FCFS);  
    streams = PDQ_CreateOpen("Operacije", L);  
  
    PDQ_SetDemand("Posluzitelj", "Operacije", S);  
    PDQ_Solve(CANON);  
  
    PDQ_Report();  
}
```

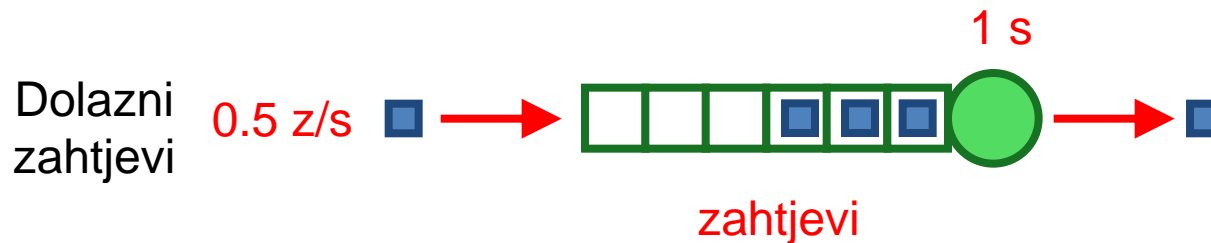


pr3.c

Primjer 4: Vrijeme čekanja i broj zahtjeva



- ◆ Sustav ima **prosječno vrijeme posluživanja 1 sekunda** i **učestalost dolazaka zahtjeva je 0.5 zadatka u sekundi**.
- ◆ Kolika je srednja vrijednost ukupnog vremena čekanja (R) i srednja vrijednost broja zahtjeva u repu (Q)?



◆ Analitičko rješenje

◆ Prosječno vrijeme posluživanja $S = 1 \text{ s/z}$

◆ Učestalost pristiglih zahtjeva $L = 0.5 \text{ z/s}$

◆ Prosječna zaposlenost sustava U

$$U = S * L = (1 \text{ s/z}) * (0.5 \text{ z/s}) = 0.5 (50 \%)$$

◆ Srednje vrijeme zadržavanja paketa u sustavu (R)

$$R = S / (1 - U) = (1) / (1 - 0.5) = 2 \text{ s}$$

◆ Srednja vrijednost broja zahtjeva u sustavu (Q)

$$Q = U / (1 - U) = 0.5 / (1 - 0.5) = 1 \text{ z}$$

Primjer 4: Vrijeme čekanja i broj zahtjeva



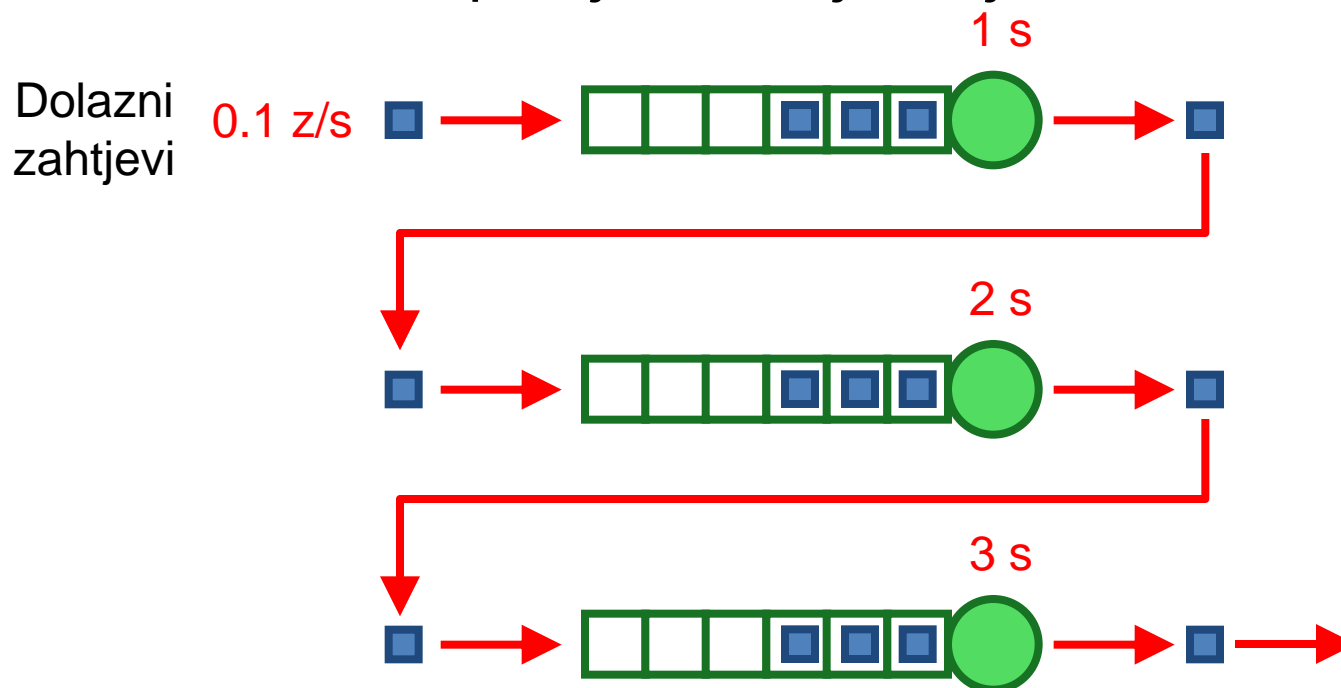
```
main() {  
    extern int nodes, streams;  
  
    float L = 0.5;  
    float S = 1.0;  
  
    PDQ_Init("Posuzitelj s repom");  
  
    nodes = PDQ_CreateNode("Posluzitelj", CEN, FCFS);  
    streams = PDQ_CreateOpen("Zadaci", L);  
  
    PDQ_SetDemand("Posluzitelj", "Zadaci", S);  
    PDQ_Solve(CANON);  
    PDQ_Report();  
}
```



pr4.c

Primjer 5: Posluživanje u seriji

- ◆ Sustav sadrži 3 serijske procesne jedinice s prosječnim vremenima posluživanja 1 s, 2 s i 3 s.
- ◆ Koliko će biti vrijeme zadržavanja u sustavu uz ulazni ritam zahtjeva od 0.1 z/s ?
- ◆ Koliki će biti prosječni broj zahtjeva u sustavu ?



◆ Analitičko rješenje

◆ Prosječna vremena posluživanja $S_1 = 1 \text{ s/z}$, $S_2 = 2 \text{ s/z}$, $S_3 = 3 \text{ s/z}$

◆ Propusnost sustava $X = 0.1 \text{ z/s}$

◆ Vremena zadržavanja $R_N = S_N / (1 - X \cdot S_N)$

$$R_1 = 1.11\text{s}, R_2 = 2.5\text{s}, R_3 = 4.29\text{s}$$

◆ Prosječni broj zahtjeva u repu Q

$$Q = X * (R_1 + R_2 + R_3) =$$

$$0.1 \text{ z/s} * (1.11 + 2.5 + 4.29) = 0.79 \text{ z}$$

Primjer 5: Posluživanje u seriji



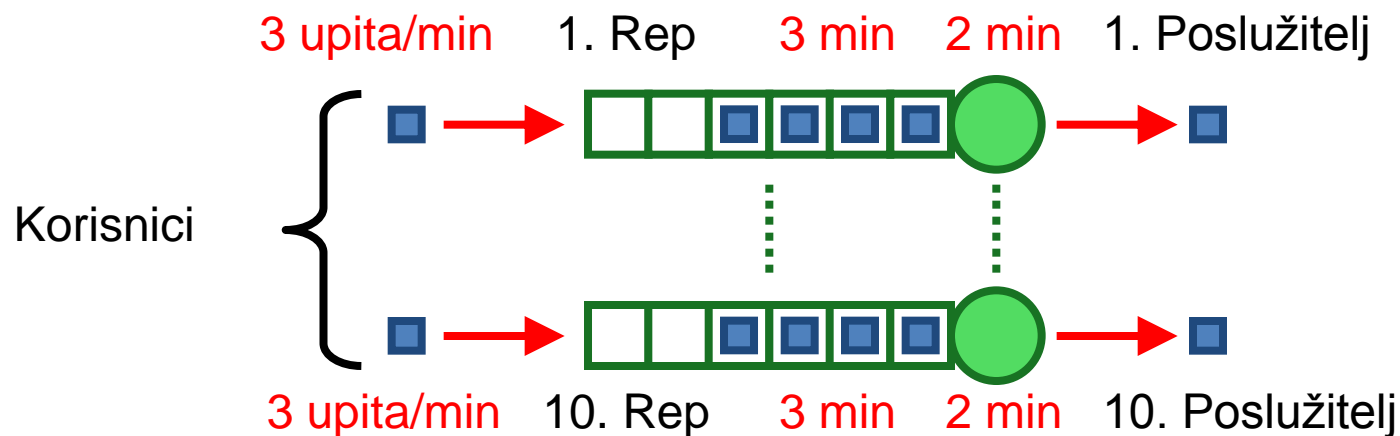
```
main() {  
    extern int nodes, streams;  
    float L = 0.1; float S1 = 1.0; float S2 = 2.0; float S3 = 3.0;  
  
    PDQ_Init("Serija tri poslužitelja");  
  
    streams = PDQ_CreateOpen("Zadaci", L);  
  
    nodes = PDQ_CreateNode("Poslužitelj1", CEN, FCFS);  
    nodes = PDQ_CreateNode("Poslužitelj2", CEN, FCFS);  
    nodes = PDQ_CreateNode("Poslužitelj3", CEN, FCFS);  
  
    PDQ_SetDemand("Poslužitelj1", "Zadaci", S1);  
    PDQ_SetDemand("Poslužitelj2", "Zadaci", S2);  
    PDQ_SetDemand("Poslužitelj3", "Zadaci", S3);  
  
    PDQ_Solve(CANON);  
  
    PDQ_Report();  
}
```



pr5.c

- ◆ Web aplikacija uključuje podršku korisnicima putem *chat* usluge. Kupci sami odabiru jedan od **10 repova čekanja**. 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 konverzaciji**.
 - ◆ Koliko bi dodatnih tehničara trebalo zaposliti da se prosječno vrijeme čekanja svede na 1 minutu ?

Primjer 6: Aplikacija korisničke podrške



◆ Analitičko rješenje

- ◆ Prosječno vrijeme posluživanja $S = 2 \text{ min/z}$
 - ◆ Broj pristiglih zahtjeva u jednom repu $L = 3 \text{ z/min}$
-

- ◆ Prosječna zaposlenost sustava (U)

$$U = S L = (2 \text{ min/z}) (3 \text{ z/min}) = 6$$

- ◆ Faktor iskorištenja (ro)

$$ro = U/N = 6/10 = 0.6$$

- ◆ Srednje vrijeme zadržavanja korisnika u sustavu (R)

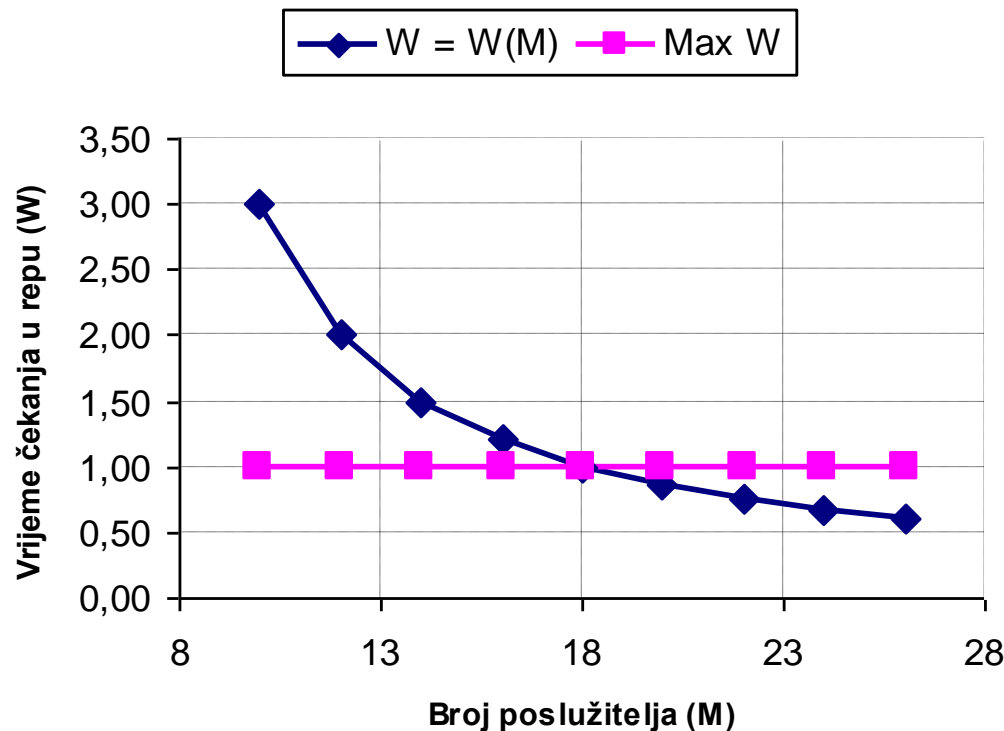
$$R = S / (1 - ro) = 2 / (1 - 0.6) = 5 \text{ min}$$

- ◆ Srednje vrijeme čekanja u repu (W)

$$W = R - S = 5 \text{ min} - 2 \text{ min} = 3 \text{ min}$$

◆ Rješenje za broj tehničara

- ◆ Za zadani sustav ne postoji analitičko rješenje. Rješenje se određuje primjenom numeričkih metoda ili primjenom metode pokušaja i promašaja.



- ◆ Kao rješenje dobije se da je potrebno 18 tehničara



[pr5.xls](#)

◆ Odabrano rješenje

- ◆ Broj poslužitelja (tehničara) $N = 18$
 - ◆ Prosječno vrijeme posluživanja $S = 2 \text{ min/z}$
 - ◆ Propusnost sustava $X = 3 \text{ z/min}$
-

- ◆ Prosječna zaposlenost sustava U

$$U = X * S = (3 \text{ z/min}) * (2 \text{ min/z}) = 6$$

- ◆ Faktor iskorištenja ro

$$ro = U/N = 6/18 = 0.33$$

- ◆ Srednje vrijeme zadržavanja korisnika u sustavu (R)

$$R = S / (1 - ro) = 2 / (1 - 0.33) = 2.985 \text{ min}$$

- ◆ Srednje vrijeme čekanja u repu (W)

$$W = 2.985 - 2 = 0.985 \text{ min}$$

Primjer 6: Aplikacija korisničke podrške



```
main() {  
    extern int    nodes;  
    extern int    streams;  
    double        L = 3;  
    double        S = 2;  
    char nName[30];  
    char cName[30];  
    int    i;  
    int    count = 10;  
  
    PDQ_Init("Aplikacija korisnicke podrške");  
  
    ...  
}
```



pr6.c

Primjer 6: Aplikacija korisničke podrške



...

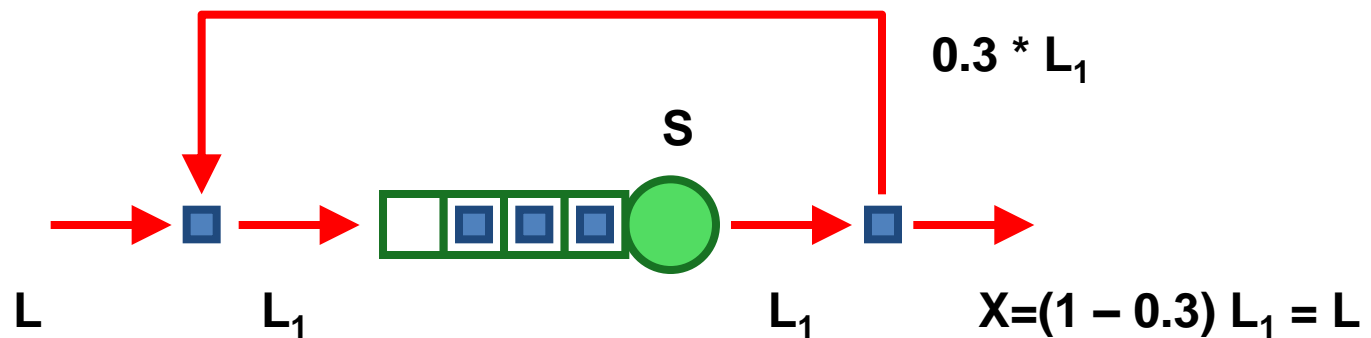
```
for( i=0; i<count; i++ ) {  
    sprintf(nName, "Serv %2d", i);  
    sprintf(cName, "CInt %2d", i);  
  
    nodes = PDQ_CreateNode(nName, CEN, FCFS);  
    streams = PDQ_CreateOpen(cName, L/count);  
}  
  
for( i=0; i<count; i++ ) {  
    sprintf(nName, "Serv %2d", i);  
    sprintf(cName, "CInt %2d", i);  
    PDQ_SetDemand(nName, cName, S);  
}  
  
PDQ_Solve(CANON);  
PDQ_Report();  
}
```



pr6.c

- ◆ Paketi dolaze u komunikacijski kanal s učestalošću 0.5 paketa u sekundi i zahtijevaju 0.75 sekundi za obradu. Za 30 % paketa dogodi se pogreška pri prijenosu i takvi paketi se umeću u rep za ponovno slanje.
 - ◆ Koliko vremena paket prosječno provede u kanalu ?

Primjer 7: Komunikacijski kanal s pogreškom



♦ Analitičko rješenje

- ♦ Broj pristiglih paketa u sekundi $L = 0.5 \text{ p/s}$
- ♦ Prosječno vrijeme obrade paketa $S = 0.75 \text{ s/p}$
- ♦ Vjerojatnost pogreške paketa pri prijenosu $p = 0.3$

- ♦ $L_1 = L / (1 - p) = 0.5 / 0.7 = 0.714 \text{ p/s}$

- ♦ Prosječna zaposlenost kanala U

$$U = L_1 * S = 0.714 \text{ p/s} * 0.75 \text{ s/p} = 0.536 \text{ (53.6 \%)}$$

- ♦ Srednje vrijeme čekanja u repu W

$$W = S * U / (1 - U) = 0.866 \text{ s/p}$$

- ♦ Srednje vrijeme zadržavanja paketa u kanalu (R_1)

$$R_1 = W + S = 0.866 \text{ s/p} + 0.75 \text{ s/p} = 1.616 \text{ s/p}$$

Prosječno vrijeme u kanalu: $R = R_1 / (1 - p) = 2.31 \text{ s}$

Primjer 7: Komunikacijski kanal s pogreškom



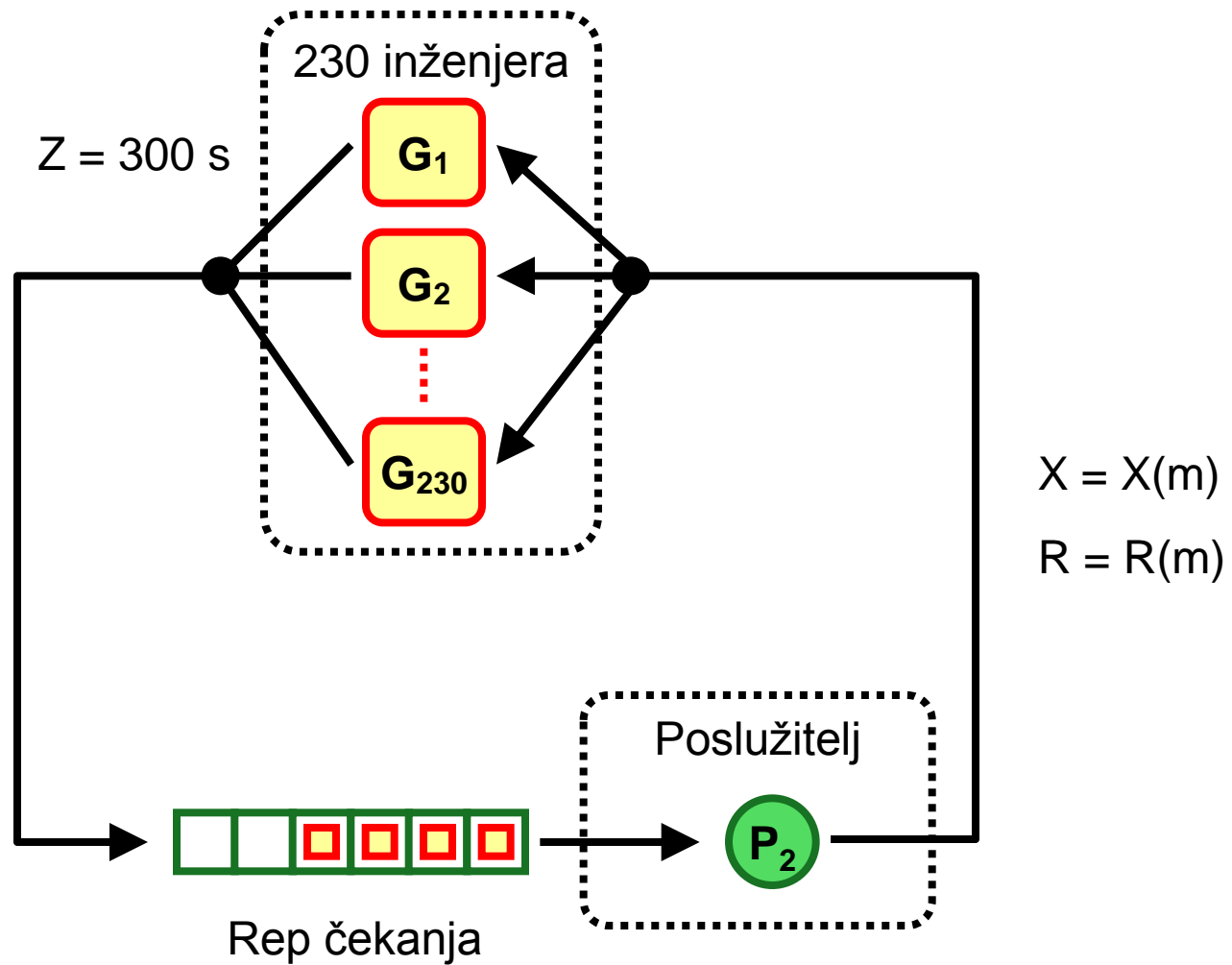
```
main() {  
    extern int      nodes, streams;  
    float  p_err = 0.30;  
    float  L    = 0.50;  
    float  S    = 0.75;  
    float  V    = 1.0 / ( 1.0 - p_err );  
  
    PDQ_Init("Posluzitelj s repom i povratnom vezom");  
  
    nodes = PDQ_CreateNode("Kanal", CEN, FCFS);  
    streams = PDQ_CreateOpen("Poruka", L);  
  
    PDQ_SetVisits("Kanal", "Poruka", V, S);  
    PDQ_Solve(CANON);  
    PDQ_Report();  
}
```



pr7.c

- ◆ Poslužitelj aplikacija omogućava skupini inženjera razvoj programa u dijeljenom vremenu. Mjerenjem su utvrđene sljedeće značajke sustava:
 - ◆ Srednji broj aktivnih razvojnih inženjera $m = 230$
 - ◆ Srednje vrijeme između kompilacija je $Z = 300$ s
 - ◆ Srednje iskorištenje poslužitelja je $U = 0.48$
 - ◆ Srednje vrijeme kompilacije $S = 0.63$ s
- ◆ Upravitelj sustava želi odrediti:
 - ◆ Propusnost sustava (X)?
 - ◆ Koliko je srednje vrijeme kompilacije (R)?





◆ Rješenje

- ◆ Broj generatora zahtjeva **$m=230$**
- ◆ Srednje vrijeme između kompilacija je **$Z = 300$ s**
- ◆ Srednje iskorištenje poslužitelja je **$U = 0.48$**
- ◆ Srednje vrijeme kompilacije **$S = 0.63$ s/kom**

◆ Propusnost sustava (X)

$$X = U/S = 0.48 / 0.63 \text{ s} = \mathbf{0.7619 \text{ kom/s}}$$

◆ Srednje vrijeme zadržavanja u sustavu (R)

$$R = m / X(m) - Z$$

$$R = (230 \text{ kom} / 0.7636 \text{ kom/s}) - 300 \text{ s} = \mathbf{1.21 \text{ s}}$$

◆ Proširivanje sustava iz prethodnog primjera

- ◆ Što će se dogoditi sa sustavom ako poduzeće zaposli novih 200 programera?

Odgovor: $U = 0.88$; $R = 5.009$ s

- ◆ Koliko će se situacija popraviti ako se poslužitelju doda drugi procesor sa istim značajkama?

Odgovor korištenjem *repair.c*:

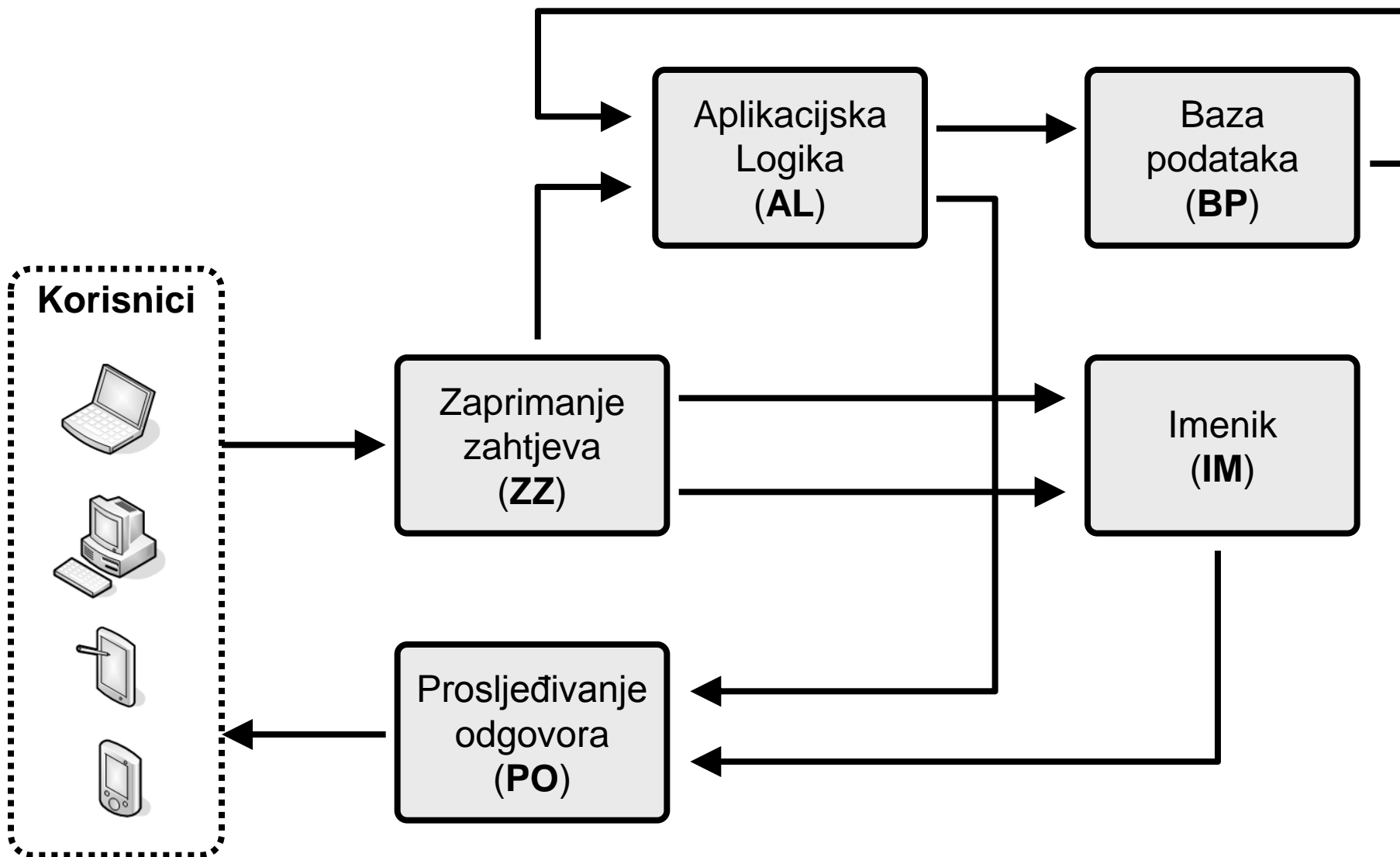
- ◆ Praksa je pokazala da se u multi-procesorskim sistemima postoji dodatni teret zbog sinkronizacije procesora. Uobičajeni faktor je 3 - 5% tj. u našem slučaju uzmimo da se S rate se povećava na $\sim 0.66s$
- ◆ Program daje slijedeće rezultate:
 - ◆ $U = 0.47$; $R = 0.8465s$



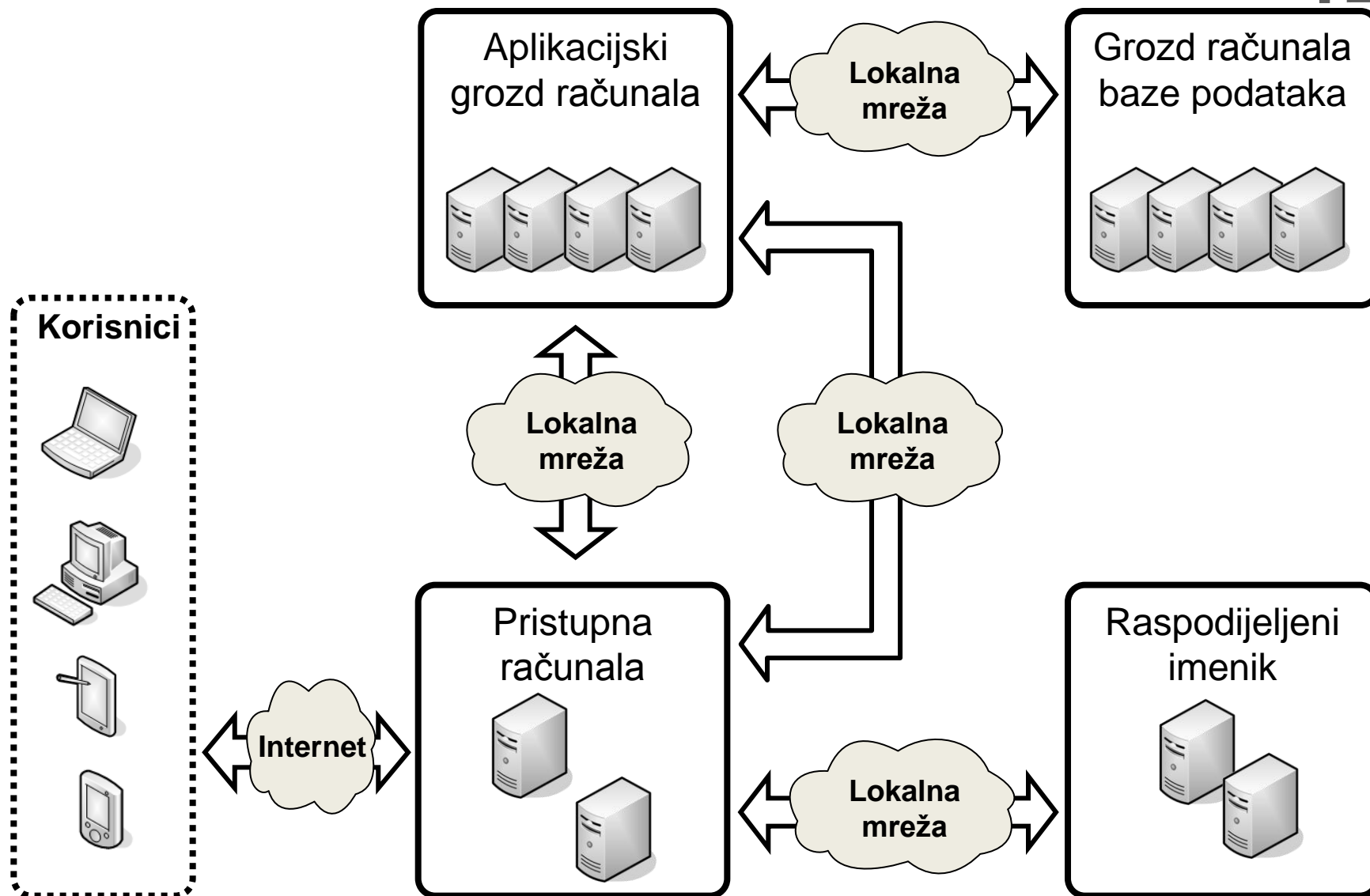
repair.c

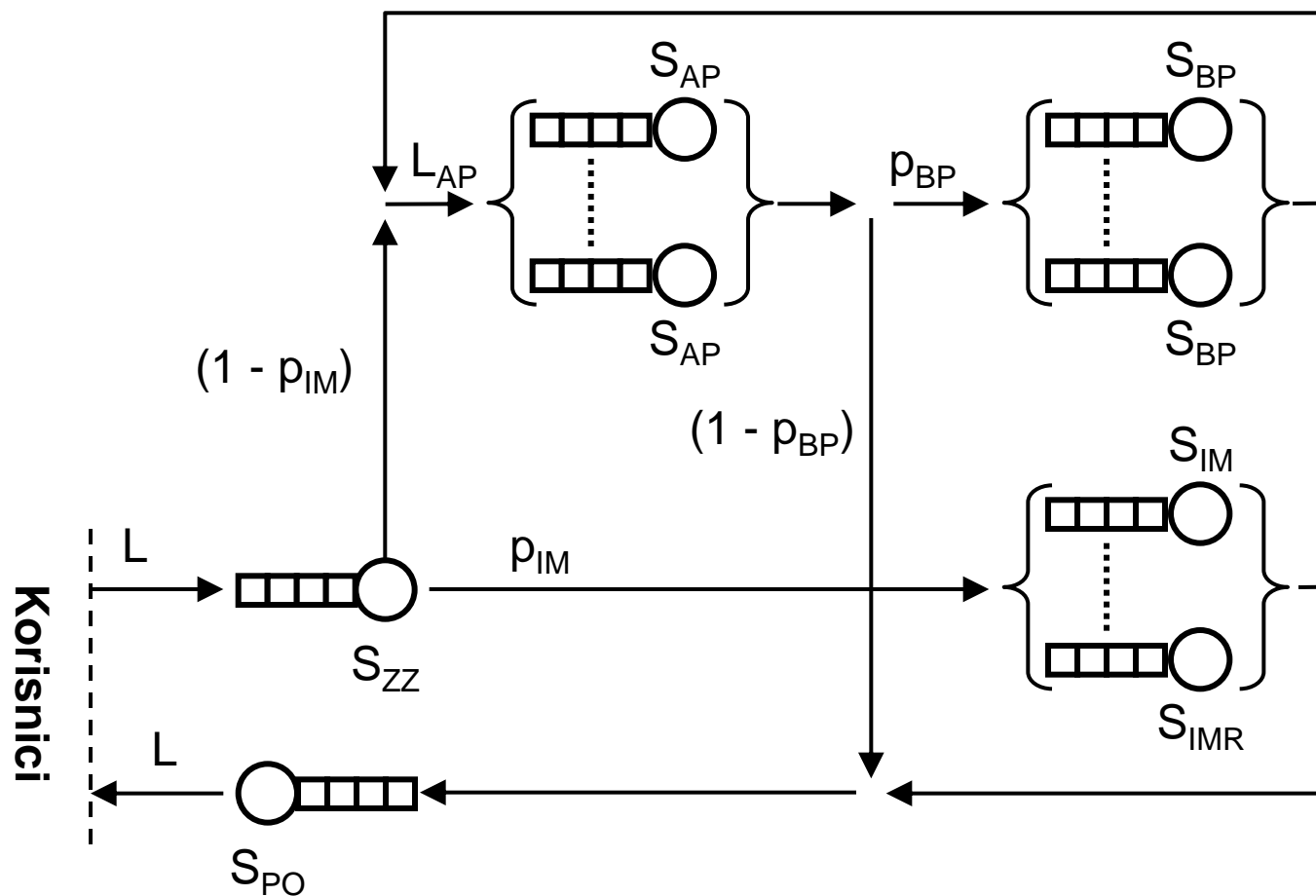
Primjer analize performansi web aplikacije

- ◆ Logička arhitektura raspodijeljene aplikacije
- ◆ Fizička arhitektura raspodijeljene aplikacije
- ◆ Model raspodijeljene aplikacije
- ◆ Vrednovanje značajki performansi aplikacije



Fizička arhitektura aplikacije





- ◆ Svaki od podsustava aplikacije na jednom računalu
- ◆ Učestalost dolazaka zahtjeva na podsustave

- ◆ $L_{IM} = p_{IM}L, v_{IM} = p_{IM}$

- ◆ $L_{AP} = p_{BP}L_{AP} + (1 - p_{IM})L \Rightarrow$

$$L_{AP} = [(1 - p_{IM})/(1 - p_{BP})]L$$

$$v_{AP} = [(1 - p_{IM})/(1 - p_{BP})]$$

- ◆ $L_{BP} = p_{BP} L_{AP} \Rightarrow$

$$L_{BP} = p_{BP} [(1 - p_{IM})/(1 - p_{BP})] L$$

$$v_{BP} = p_{BP} [(1 - p_{IM})/(1 - p_{BP})]$$

- ◆ $L_{ZZ} = L, v_{ZZ} = 1$

- ◆ $L_{PO} = L, v_{PO} = 1$

◆ Skalirana vremena posluživanja

$$◆ D_{IM} = v_{IM} S_{IM} = p_{IM} L$$

$$◆ D_{AP} = v_{AP} S_{AP} = [(1 - p_{IM})/(1 - p_{BP})] L \Rightarrow$$

$$◆ D_{BP} = v_{BP} S_{BP} = p_{BP} [(1 - p_{IM})/(1 - p_{BP})] S_{AP}$$

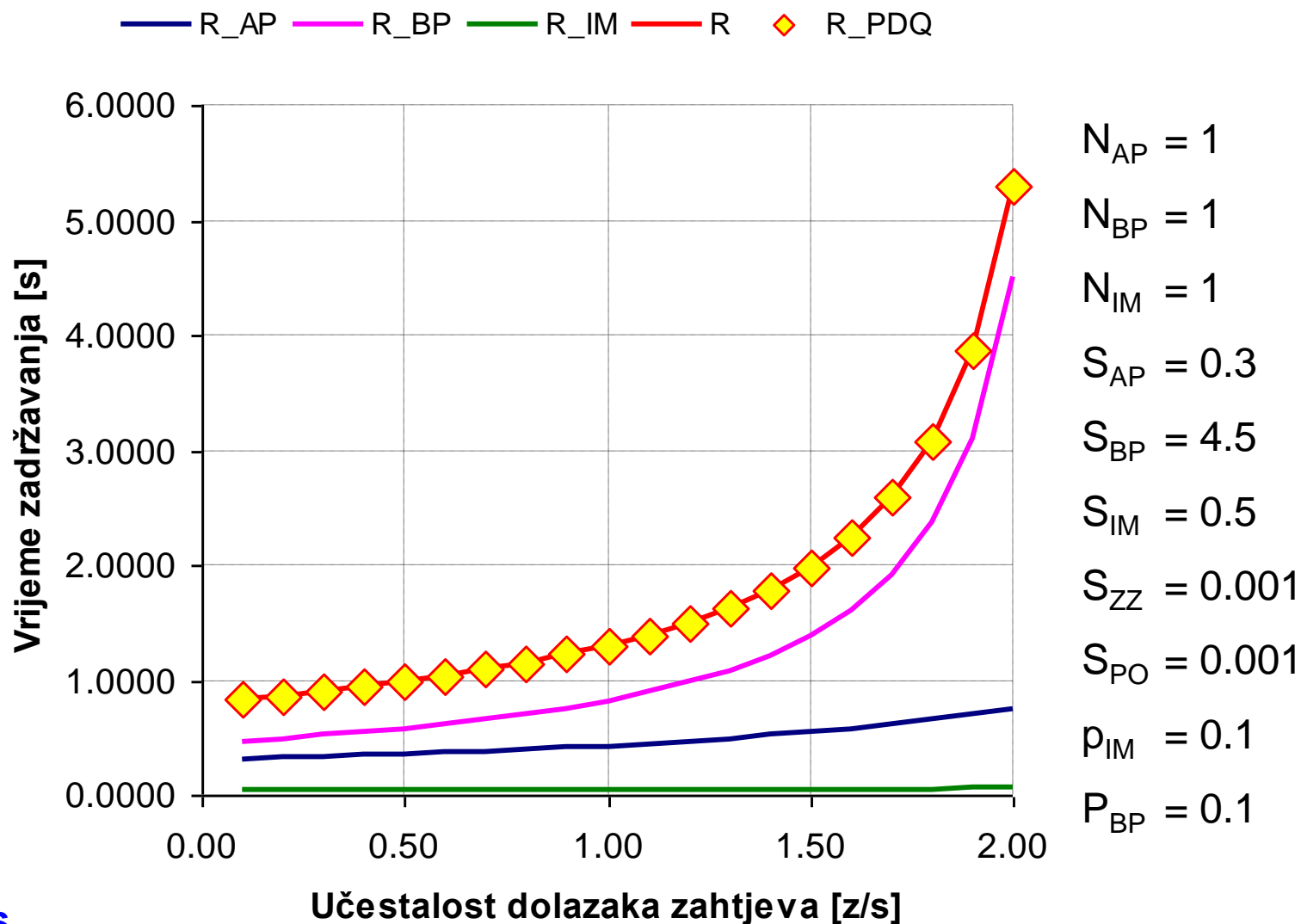
$$◆ D_{ZZ} = v_{ZZ} S_{BP} = 1 S_{BP}$$

$$◆ D_{PO} = v_{PO} S_{BP} = 1 S_{PO}$$

◆ Vrijeme zadržavanja zahtjeva

$$◆ R = D_{IM}/(1 - L D_{IM}) + D_{AP}/(1 - L D_{AP}) + D_{BP}/(1 - L D_{BP}) + \\ D_{ZZ}/(1 - L D_{ZZ}) + D_{PO}/(1 - L D_{PO})$$

Vrijeme zadržavanja zahtjeva

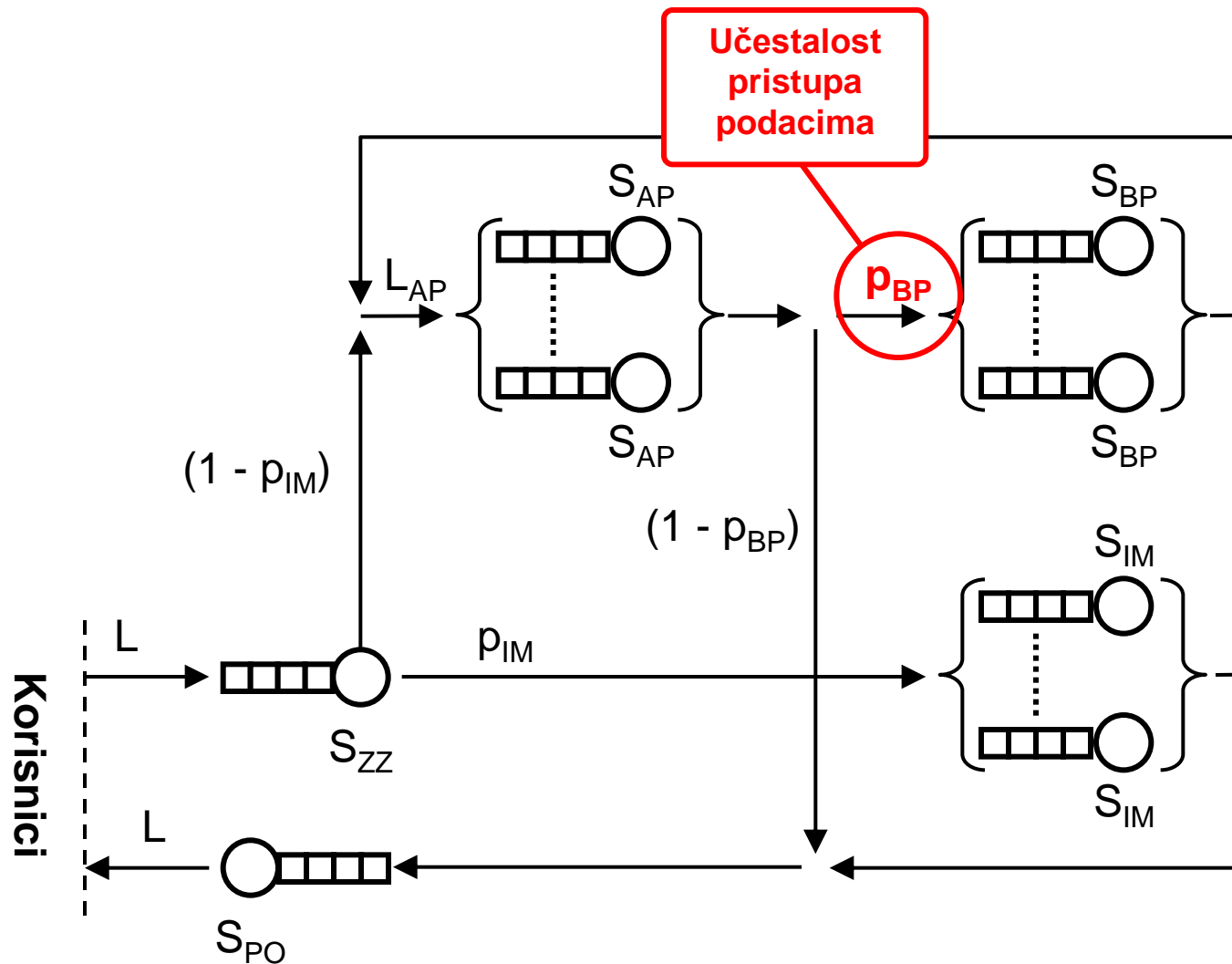


ap-Zad.c

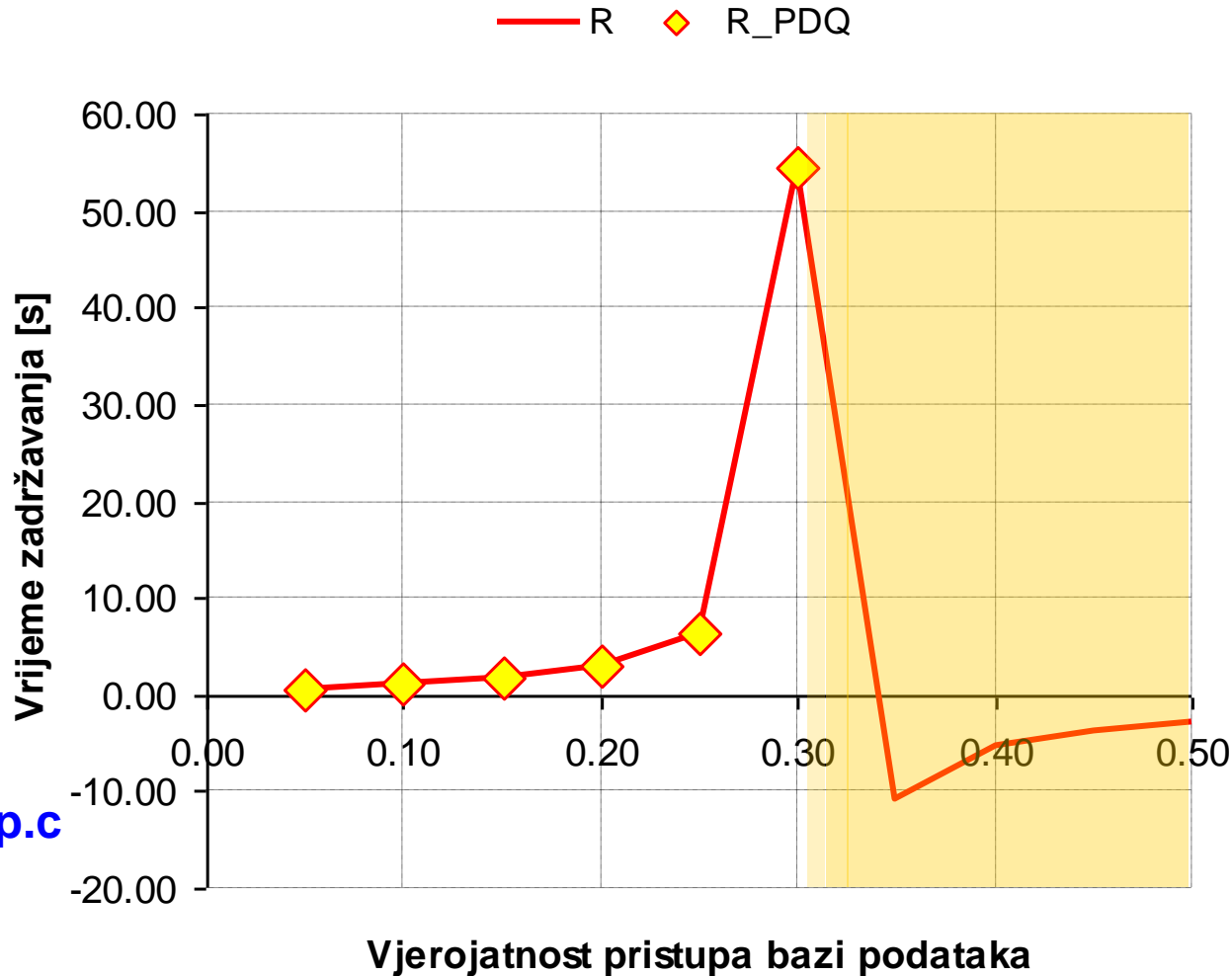


Ap_Zad.xls

- ◆ Učestalosti pristupa podacima
- ◆ Veličina grozda baze podataka
- ◆ Promjena organizacije podataka
- ◆ Promjena stupnja sigurnosti



Vrijeme zadržavanja zahtjeva



$$N_{AP} = 1$$

$$N_{BP} = 1$$

$$N_{IM} = 1$$

$$S_{AP} = 0.3$$

$$S_{BP} = 4.5$$

$$S_{IM} = 0.5$$

$$S_{ZZ} = 0.001$$

$$S_{PO} = 0.001$$

$$p_{IM} = 0.0$$

$$L = 0.5$$

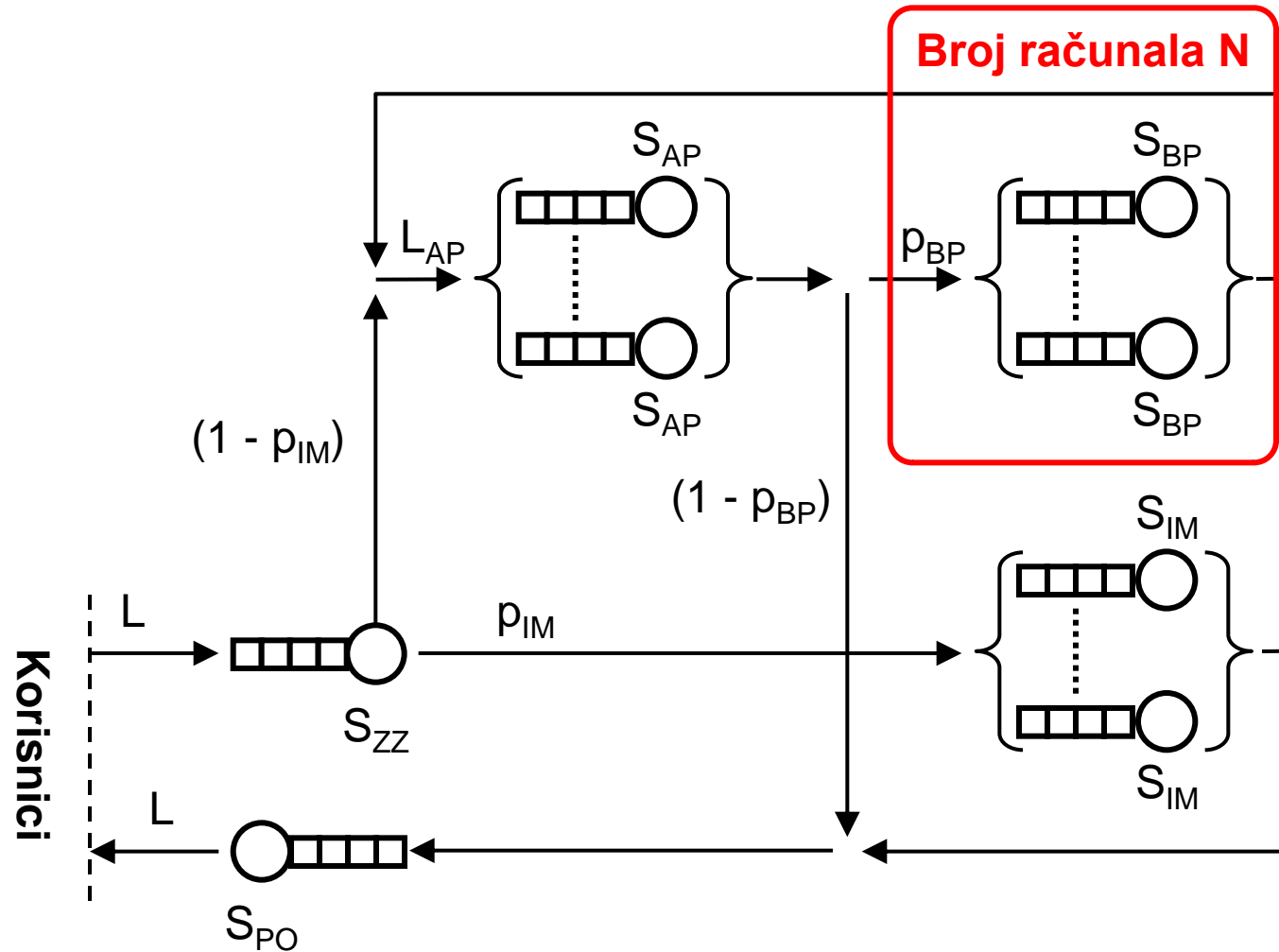


ap-Pristup.c



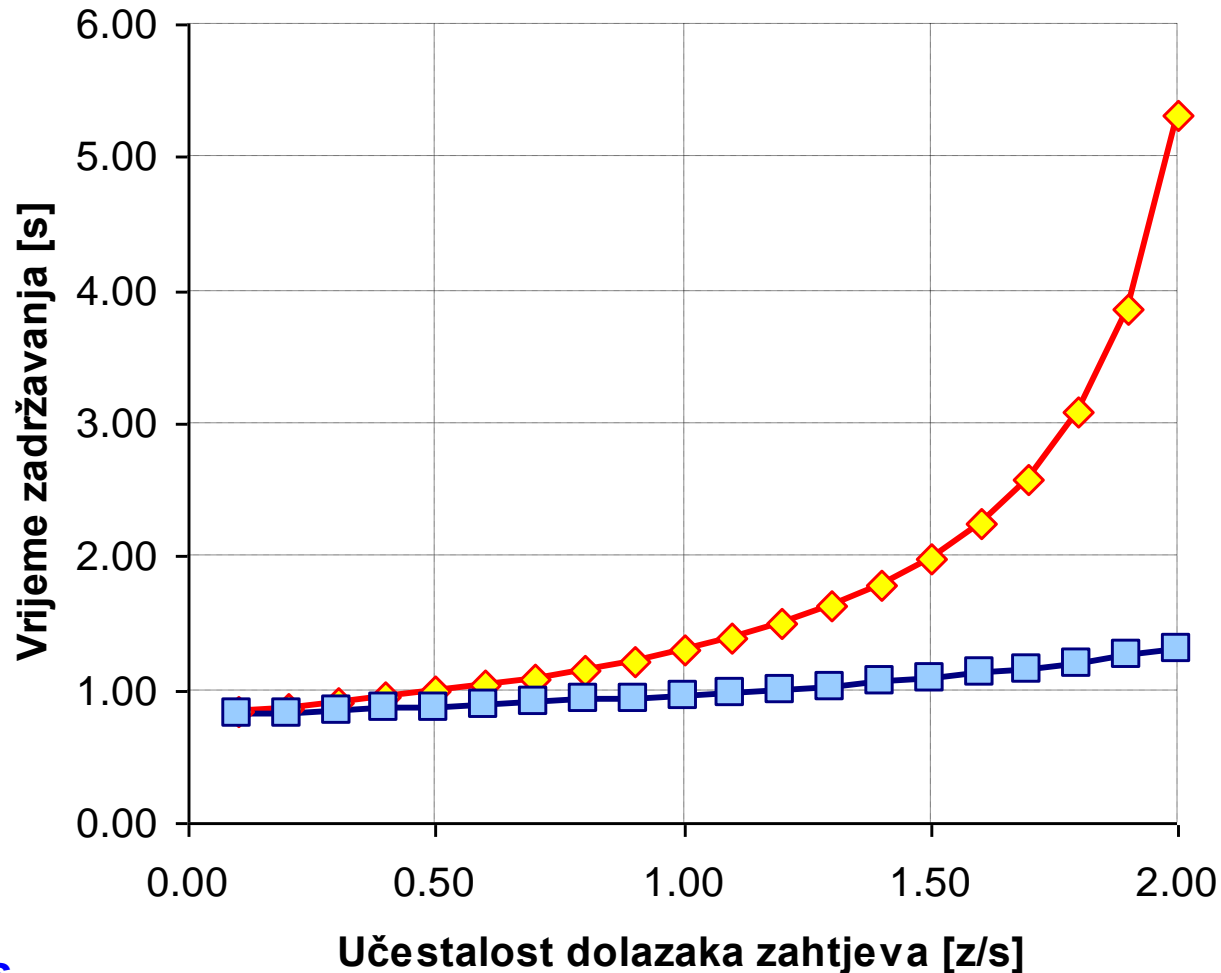
Ap_Prist.xls

Veličina grozda baze podataka



Vrijeme zadržavanja zahtjeva

—◆— $N = 1$ —■— $N = 10$



$S_{AP} = 0.3$
 $S_{BP} = 4.5$
 $S_{IM} = 0.5$
 $S_{ZZ} = 0.001$
 $S_{PO} = 0.001$
 $\rho_{IM} = 0.1$
 $P_{BP} = 0.1$



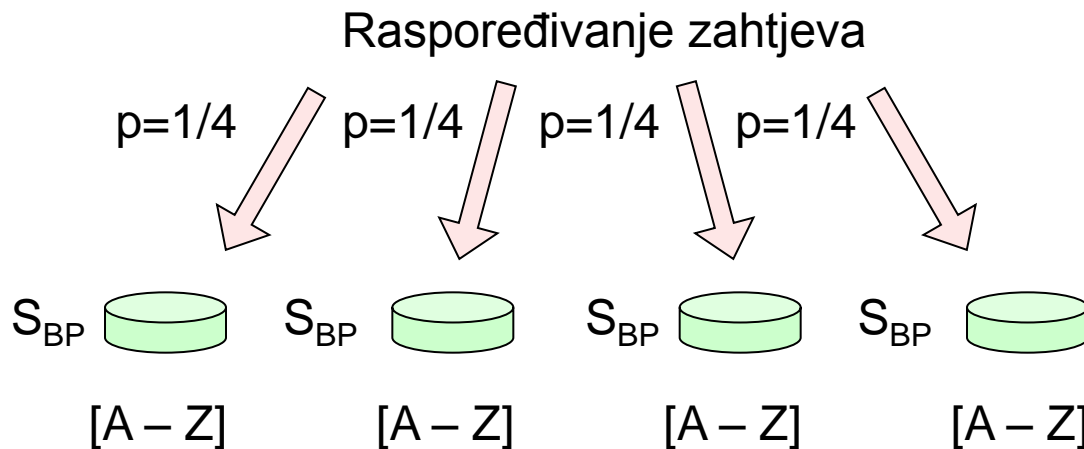
ap-Zad.c



Ap_Rac.xls

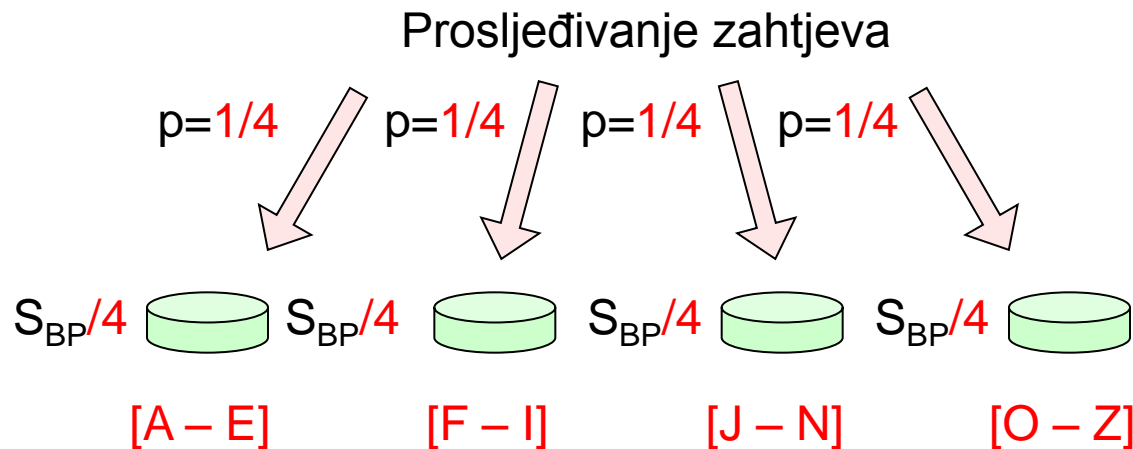
◆ Replikacija podataka

- ◆ Skup računala od kojih svako u spremniku sadrži kopiju cijele baze podataka
- ◆ Zahtjevi se **raspoređuju** na računala s ciljem raspoređivanja opterećenja



◆ Segmentacija podataka

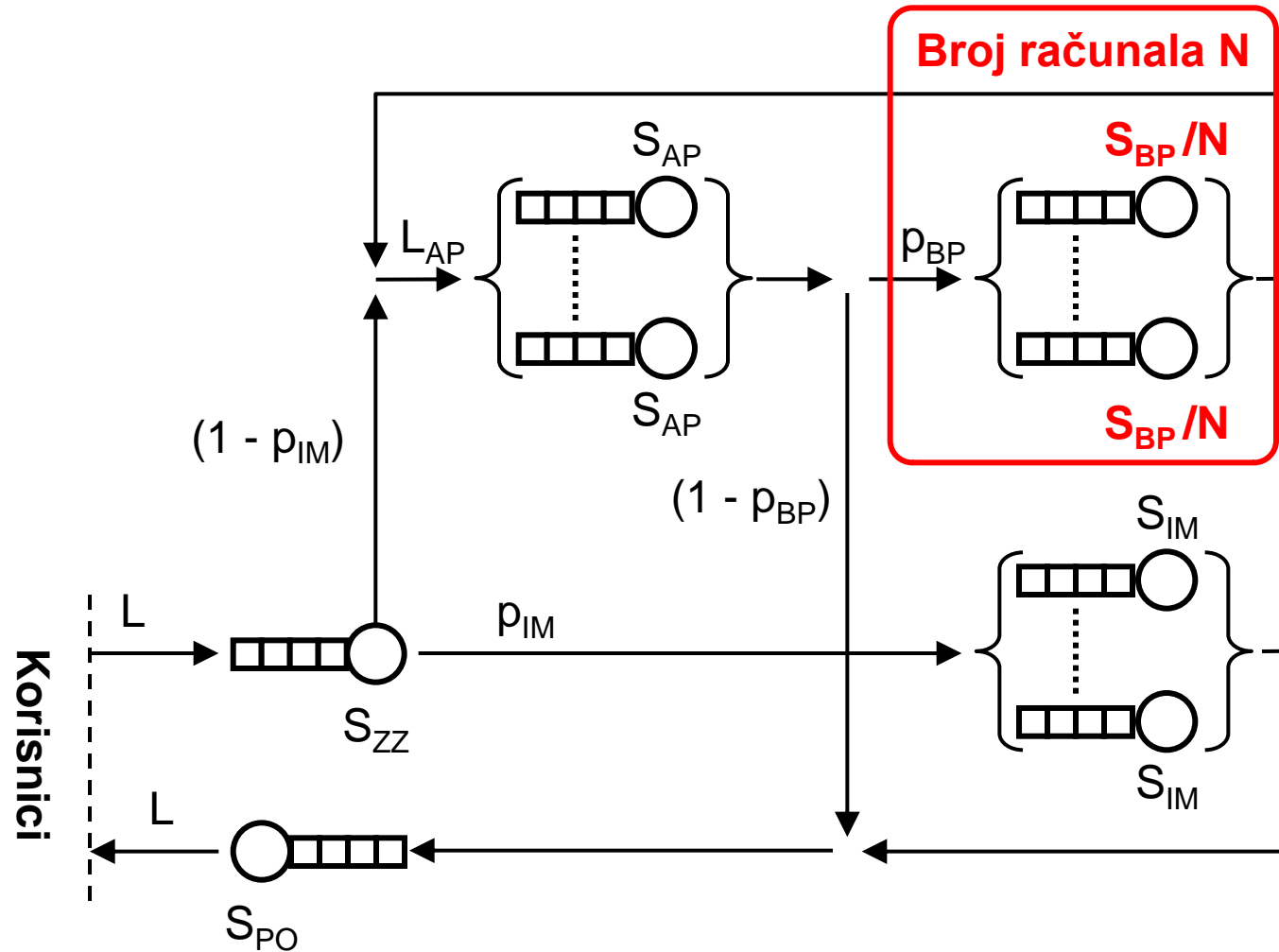
- ◆ Skup računala od kojih svako u spremniku sadrži dio cijele baze podataka
- ◆ Zahtjevi se **prosljeđuju** prema računalu s traženim zapisima



◆ Pretpostavke

- ◆ Uniformna raspodjela zahtjeva na zapise
- ◆ Linearna složenost obrade zahtjeva o količini zapisa

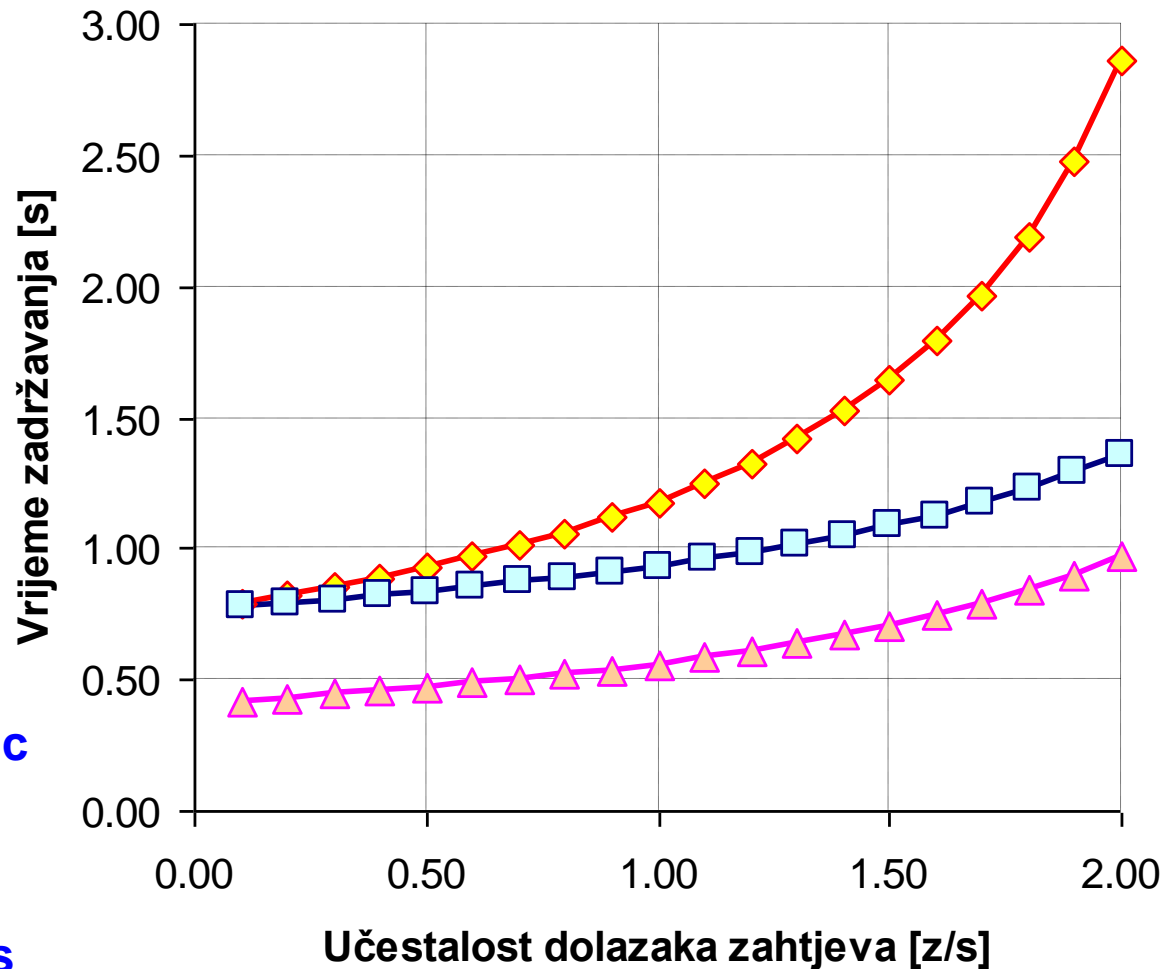
Promjena organizacije podataka



Vrijeme zadržavanja zahtjeva



—◆— $N = 1$ (REP) —■— $N = 10$ (REP) —▲— $N = 10$ (SEG)



$N_{AP} = 1$
 $N_{BP} = 1$
 $N_{IM} = 1$
 $S_{AP} = 0.3$
 $S_{BP} = 2.5$
 $S_{IM} = 0.5$
 $S_{ZZ} = 0.001$
 $S_{PO} = 0.001$
 $p_{IM} = 0.1$
 $p_{BP} = 0.15$



ap-PodOrg.c



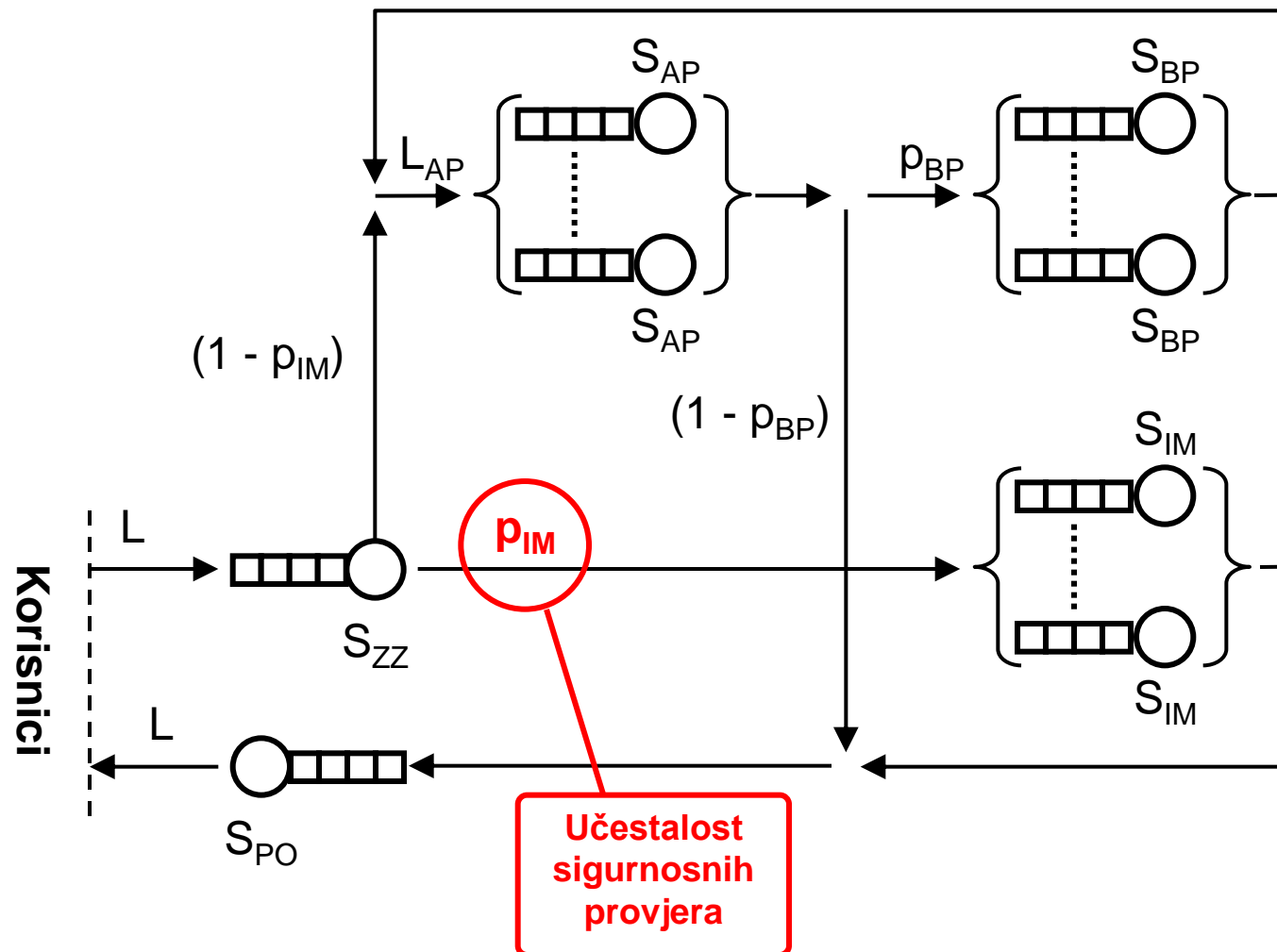
Ap_Pod.xls

- ◆ **Imenik aplikacije sadrži informacije o korisnicima**
 - ◆ Korisnički identiteti
 - ◆ Korisnička prava pristupa

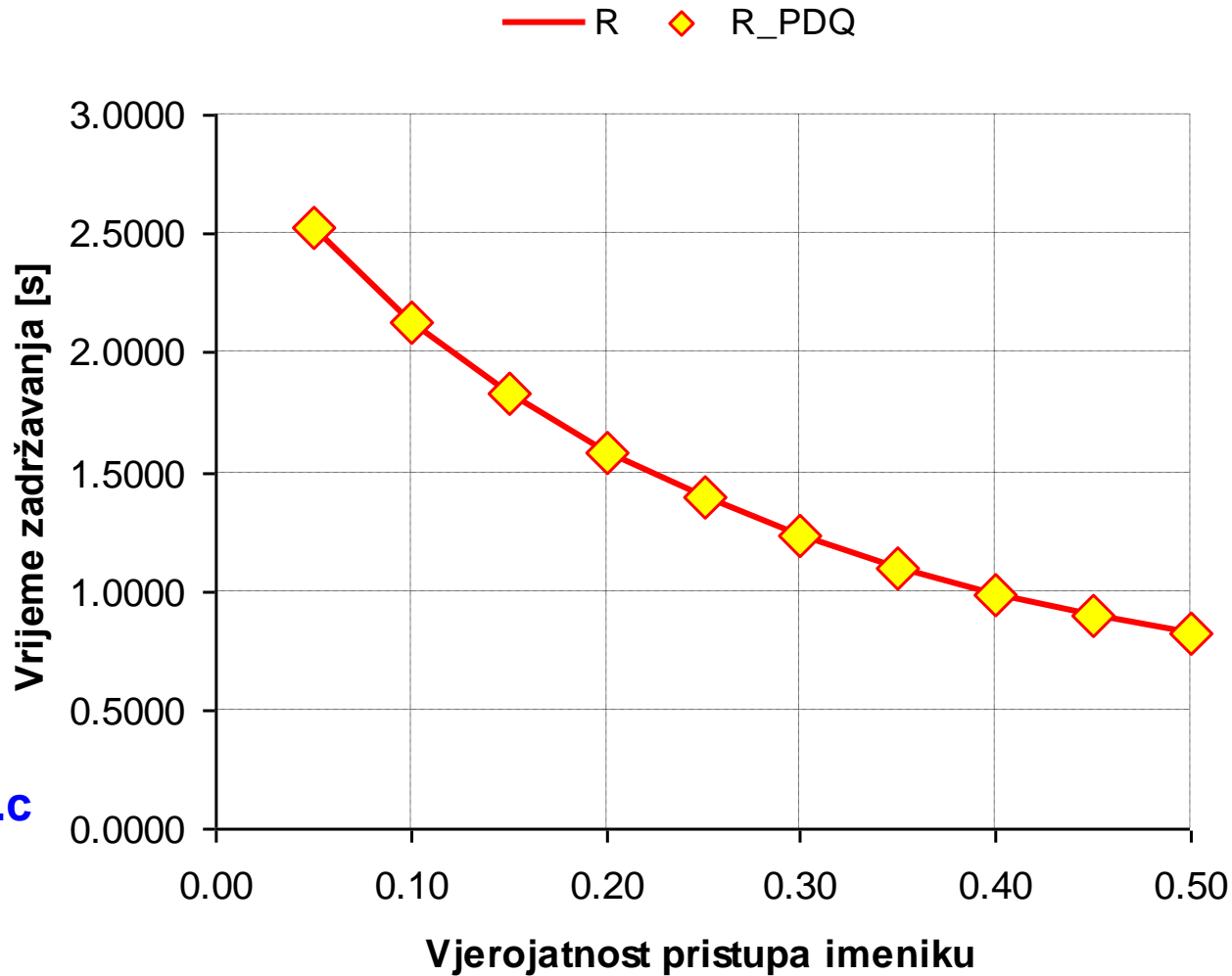
- ◆ **Sigurnosna značka**
 - ◆ Određuje sigurnosne postavke korisnika aplikacije
 - ◆ Značka se dohvaća iz imenika

- ◆ **Životni vijek sigurnosne značke**
 - ◆ Ograničeni broj pristupa
 - ◆ Zadano vrijeme korištenja
 - ◆ Ostali sigurnosni modeli

Promjena stupnja sigurnosti



Vrijeme zadržavanja zahtjeva



$$N_{AP} = 1$$

$$N_{BP} = 1$$

$$N_{IM} = 1$$

$$S_{AP} = 0.3$$

$$S_{BP} = 4.5$$

$$S_{IM} = 0.5$$

$$S_{ZZ} = 0.001$$

$$S_{PO} = 0.001$$

$$p_{BP} = 0.2$$

$$L = 0.5$$



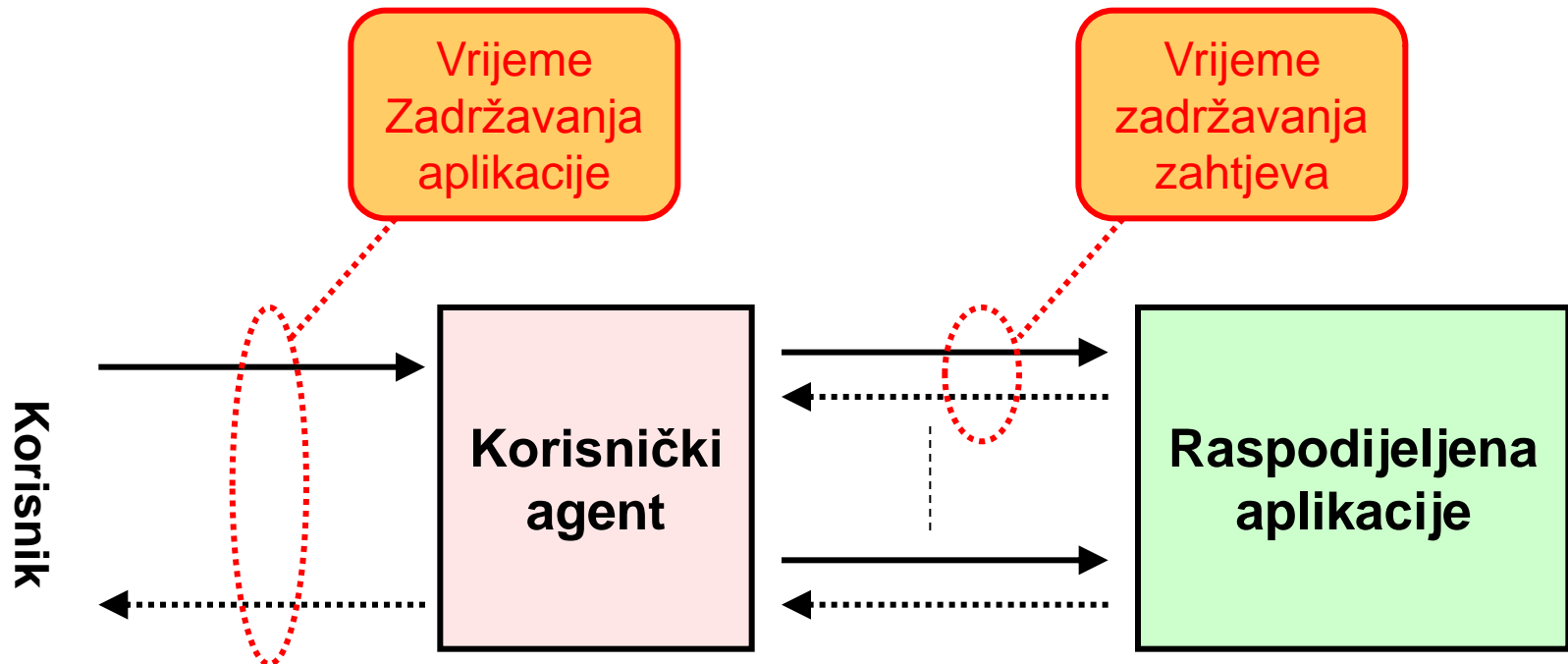
ap-Sig.c



Ap_Sig.xls

◆ Zašto vrijeme zadržavanja opada?

- ◆ Modelirano je vrijeme zahtjeva zadržavanja ali ne i ukupno vrijeme zadržavanja aplikacije koje doživljava korisnik



Domaća zadaća

♦ **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?

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

- ◆ **Zadatak 2:** Oblikovati proizvoljnu raspodijeljenu aplikaciju i ostvariti analizu performansi ostvarene aplikacije
 - 1) Definirati logičku i fizičku arhitekturu aplikacije**
 - 2) Izgraditi model aplikacije primjenom teorije repova**
 - Odrediti analitičko rješenje funkcije zadržavanja zahtjeva u aplikaciji $R=f(L)$
 - 3) Izgraditi model aplikacije za alat PDQ**
 - Primjenom izgrađenog modela odrediti vrijednosti funkcije zadržavanja zahtjeva $R=f(L)$ u nekoliko točaka
 - 4) Usporediti i obrazložiti dobivene rezultate**