

# Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek  
*Sveučilište u Zagrebu*  
*Fakultet organizacije i informatike*  
*Pavlinska 2, Varaždin 42000*  
**dkermek@foi.hr**

19. dio

## Mjerenje opterećenja Web mjesta

- Kako se ponaša sustav u slučajevima različitog opterećenja?
- Kako organizirati i orkestrirati korisnike da obavljaju određene aktivnosti?
- Kako osigurati okolinu koja će odgovarati stvarnoj situaciji?
- Kako osigurati opterećenje svih elemenata koje će odgovarati stvarnoj situaciji?
- Simulacijom broja korisnika, tempa njihovog rada i izbora dokumenata koje će učitavati
- Za složene situacije može se formirati grupa raspodjeljenih računala koja će provoditi aktivnosti pod zajedničkom kontrolom raspodjeljenog sustava



## Osnove testiranja opterećenja Web sustava

Testiranje performansi provodi se kroz sljedeće aktivnosti:

- **Identificiranje testne okoline** – utvrđuje se fizička okolina za testiranje i produkcijska okolina te alati i resursi koji su na raspolaganju timu za testiranje.
- **Identificiranje kriterija prihvatljivosti performansi** – utvrđuje se ciljne vrijednosti i ograničenja za vrijeme odgovora, propustnost i korištenje resursa. Vrijeme odgovora odnosi se na korisničku domenu, propustnost spada u poslovnu domenu, a korištenje resursa je sistemska domenu.
- **Planiranje i dizajn testova** – utvrđuju se ključni scenariji, određuju se varijabilnosti za predstavljenje korisnika i kako će se one simulirati, određuju se testni podaci i definiraju mjerenja koja će se prikupljati.

Prema: J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea, Performance Testing Guidance for Web Applications, Microsoft Corporation, 2007



## Osnove testiranja opterećenja Web sustava

- **Konfiguriranje testne okoline** – priprema testne okoline, alata i potrebnih resursa za izvršavanje svake strategije kako pojedine osobine i komponente sustava budu na raspolaganju.
- **Implementacija dizajna testa** – definiranje testova u skladu s dizajnom testa.
- **Izvršavanje testa** – provođenje i nadziranje testova. Slijedi validacija testova, testiranje podataka i kolekcija rezultata.
- **Analiza rezultata, izrada izvještaja, ponavljanje testa** – konsolidacija podataka iz testova, njihovo analiziranje kroz više razina i grupa, promjena prioriteta kod preostalih testova i po potrebi ponavljanje nekih testova.



## Kategorije testiranja performansi

Testiranja performansi mogu biti u jednoj od sljedećih kategorija:

- **Testiranje performansi (Performance test)** – utvrđuje se ili provjerava brzina, skalabilnost, stabilnost kao osobine sustava ili aplikacije koja se testira. Performanse koje se razmatraju su vrijeme odziva, propustnost, razina korištenja resursa koji se odnose na ciljeve performansi koji su postavljeni za sustav ili aplikaciju.
- **Testiranje opterećenja (Load test)** – utvrđuju se ili provjeravaju osobine performansi sustava ili aplikacije koja se testira, kada je podređena radnim opterećenjima korištenja i koje se očekuju za vrijeme normalnog izvršavanja poslova.



## Kategorije testiranja performansi

- **Testiranje naprezanja (Stress test)** – utvrđuju se ili provjeravaju osobine performansi sustava ili aplikacije koja se testira kada su uvjeti ispod očekivanih za normalan rad sustava (ograničena memorija, diskovni prostor, kvar poslužitelja). Utvrđuje se pod kojim uvjetima će aplikacija prestati s radom, kako će prestati i koji se indikatori mogu nadzirati kako bi se upozorilo na nadolazeći prekid.
- **Test kapaciteta (Capacity test)** – utvrđuje se koliko korisnika ili transakcija može sustav opsluživati da još uvijek da ispunjava ciljeve performansi.



## Okruženje rada Web sustava

**Web poslužitelj - čeonni dio sustava**

**Poslužitelj baza podataka - pozadinski dio sustava**

**Email poslužitelj - pomoćni dio sustava**

**LDAP poslužitelj - pomoćni dio sustava**



## Programski alati za testiranje

- JMeter, <http://jakarta.apache.org/jmeter/index.html>
- Web Application Stress Tool,  
<http://www.microsoft.com/downloads/details.aspx?FamilyID=e2c0585a-062a-439e-a67d-75a89aa36495&DisplayLang=en>
- NeoLoad, <http://www.neotys.com/>
- Badboy, <http://www.badboy.com.au/>
- Webserver Stress Tool, <http://www.paessler.com/webstress>
- WAPT, <http://www.loadtestingtool.com/>
- WebLOAD, <http://www.webload.org/>, <http://www.radview.com/>
- OpenLoad, <http://www.opendemand.com/>
- Testing Master, <http://www.siteloadtesting.com/>
- Proxy Sniffer, <http://www.proxy-sniffer.com/>



## Testni sustav

Školski primjer koji se bazira na nekoliko “skromnih” konfiguracija sustava koje studentima stoje na raspolaganju.

Za testiranje je odabran alat JMeter iz više razloga:

- ⇒ neovisan o platformi
- ⇒ otvorenog koda i besplatan.

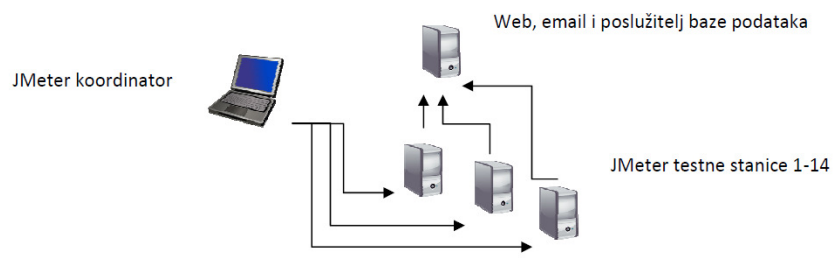


## Apache JMeter

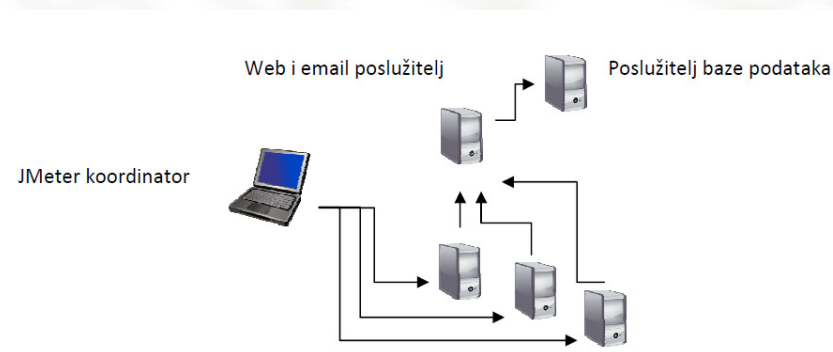
- ❑ JMeter se može koristiti za testiranje performansi statičkih i dinamičkih resursa (datoteka, servleta, skripti (ASP, JSP, Perl, PHP, ROR), Java Objekata, podatkovnih sabirnica i redova, FTP servera i sl
- ❑ Može se koristiti za simuliranje velikog opterećenja servera, mreže ili objekata za testiranje snage ili za analizu općih performansi pod različitim vrstama opterećenja.
- ❑ Može se koristiti za grafičku analizu performansi ili za testiranje ponašanja vaših poslužitelja/skripata/objekata pod velikim konkurentnim opterećenjem.



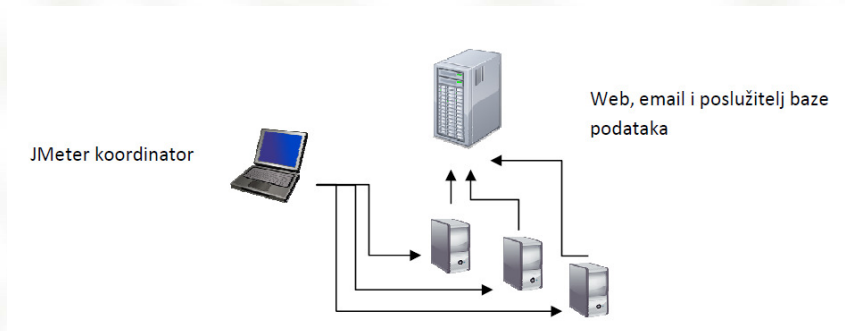
## Fizička konfiguracija broj 1



## Fizička konfiguracija broj 2



## Fizička konfiguracija broj 3



## Identificiranje kriterija prihvatljivosti performansi

Vrijeme odziva koje se odvija u širokopojasnom kao preporučenom načinu spajanja na Internet, tada se može reći da je prihvatljivo u granicama od nekoliko sekundi pri čemu se više od 10 sekundi smatra neprihvatljivo

O ovisnosti o konfiguraciji ni jedan poslužitelj ne bio smio preći granicu od 75 % korištenja procesora podataka tijekom vršnog opterećenja.

Rad sustava za upravljanje bazom podataka tijekom vršnog opterećenja mora biti takav da korištenje procesora ne pređe 50-75 % kako bi se osiguralo dovoljno vremena procesora za ostale konkurentne poslove.



## Planiranje i dizajn testa

Razrada scenarija testiranja.

Izvor podataka može biti dnevnik rada postojećeg/ih sustava iz kojeg se izvlače podaci o slijedu rada korisnika, vremenima između poziva pojedinih web stranica i sl.

Korištenjem sustava putem JMeter HTTP Proxy modula mogu se prikupiti podaci o slijedu rada korisnika, vremenima između poziva pojedinih web stranica i sl.

Razlike u radu korisnika sustava:

- ⇒ anonimni korisnik
- ⇒ prijavljeni korisnik



## Planiranje i dizajn testa

Na temelju podataka određene su vremenske granice (min-max) između poziva pojedine vrtse web stranice.

Za određene web stranice definiraju se drugi parametri (sortiranje podataka, količina podataka koja se zahtjeva (straničenje), pozicija od koje se preuzimaju podaci i sl). Svi ti parametri također se trebaju odabirati unutar određenih vrijednosti.

Kod svake aktivnosti koja se odnosi na rad Web aplikacije bilo je važno utvrditi status njenog završetka kako bi se mogla voditi evidencija uspješnosti rada. Zbog toga se analizirao odgovor svake Web stranice u kojem se tražio očekivani tekst odnosno niz riječi koji je predstavljao uspješno završenu aktivnost, odnosno pogrešku ukoliko se nije pojavio.





## Konfiguriranje testne okoline

Testna okolina 15 stolnih računala od kojih je jedno imalo ulogu potrebnih poslužitelja, a ostala su bila angažirana kao testne stanice.

Na poslužitelj je ranije instaliran WAMP koje povezuje Apache Web poslužitelj s PHP podrškom i MySQL sustav za upravljenje bazama podataka.

Zatim je instalirana Web aplikacija, kreirana potrebna baza podataka te napunjena potrebnim podacima.

Instaliran je James (email poslužitelj) na kojem su kreirani podaci za potrebne korisnike.

Na svih 14 testnih stanica instaliran je JMeter i otvoren port 1099 kako bi se moglo provesti distribirano testiranje.



## Konfiguriranje testne okoline

Kod druge fizičke konfiguracije bilo je potrebno uključiti dodatno računalo koje je dobilo ulogu poslužitelja baza podataka te provesti njegovu pripremu za rad kao što je to opisano kod prve konfiguracije.

Kod treće konfiguracije već su ranije bili instalirani potrebni poslužiteljski programi, stoga je samo bilo potrebno instalirati Web aplikaciju, kreirati bazu podataka i napuniti ju potrebnim podacima.

Na centralnom JMeter računalu bilo je potrebno pripremiti parametre rada programa. Tako su upisane IP adrese svih testnih stanica te definirana minimalna i maksimalna veličina memorije koja stoji na raspolaganju JMeteru tijekom rada.



## Implementacija dizajna testa

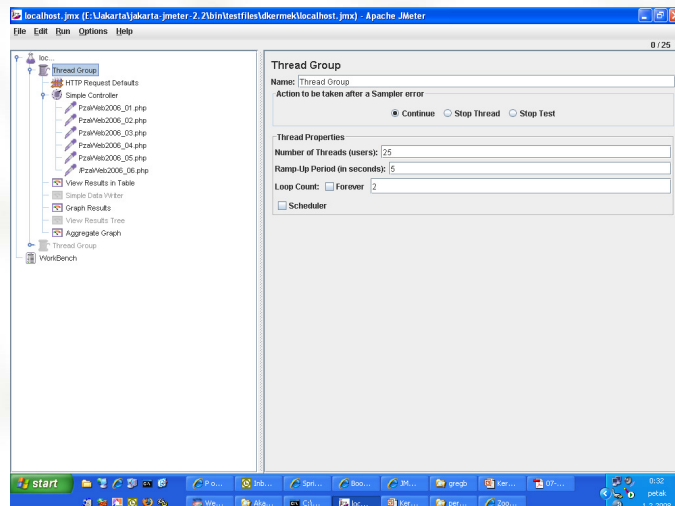
Veliki dio testiranja treba sadržavati mogućnost odabira kako između elemenata u nekom skupu ili unutar raspona vrijednosti.

Rad JMetera temelji se na pseudogeneratoru slučajnih brojeva kako bi se ostvarile one zadane osobine iz dizajna testa koje se odnose na varijabilnost korisnika, vremena, slijeda, količine podataka i sl.

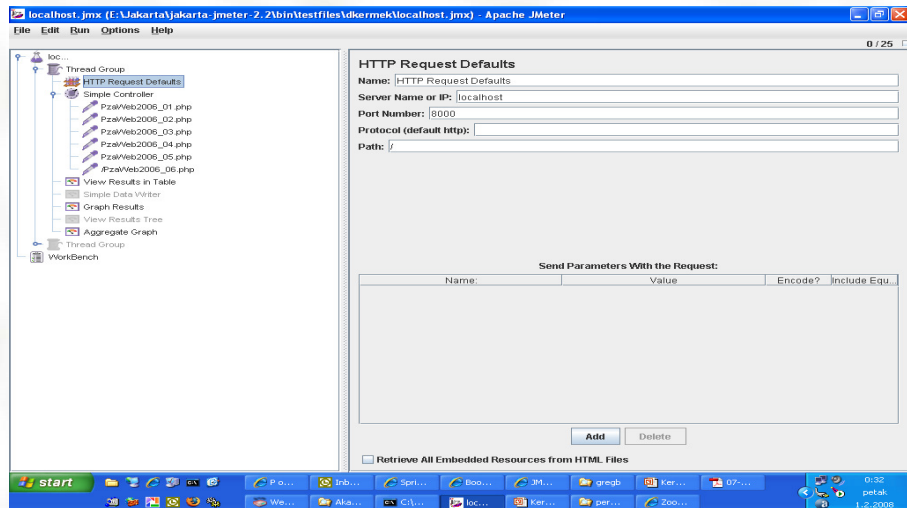
Dovršeni plan testiranja putem alata JMeter sadržavao je gotovo 5000 linija unutar XML datoteke konfiguracije.



## Apache JMeter



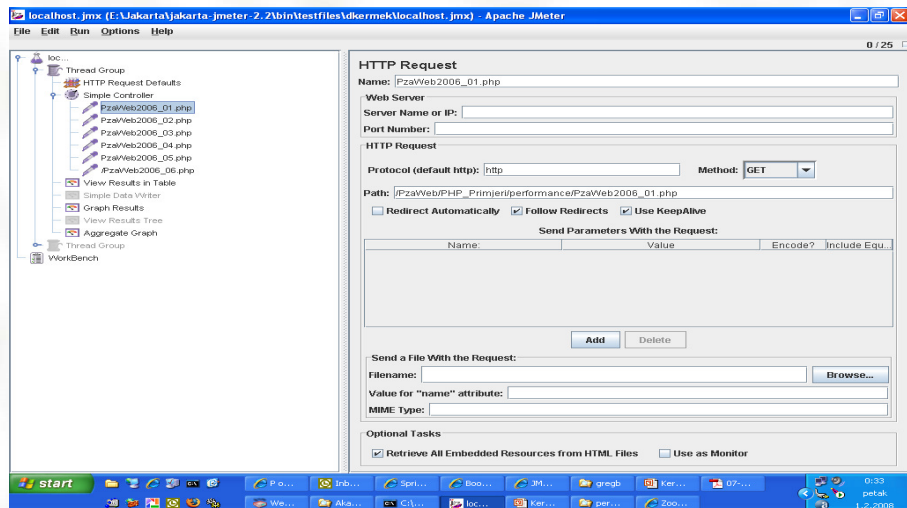
# Apache JMeter



21



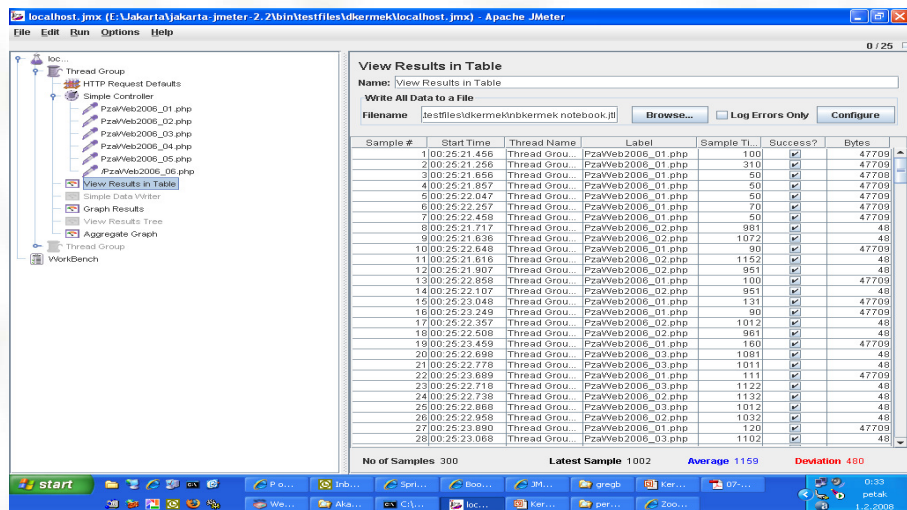
# Apache JMeter



22



# Apache JMeter

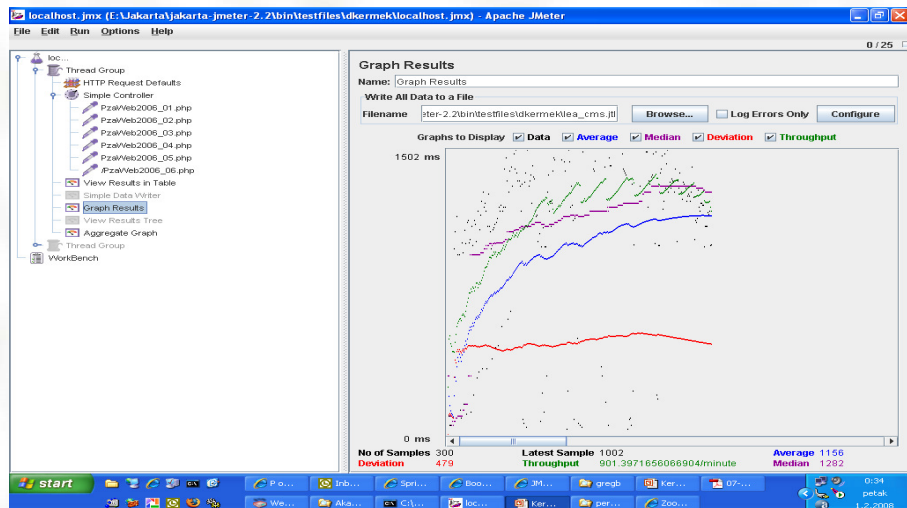


23

Web dizajn i programiranje - FOI  
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



# Apache JMeter

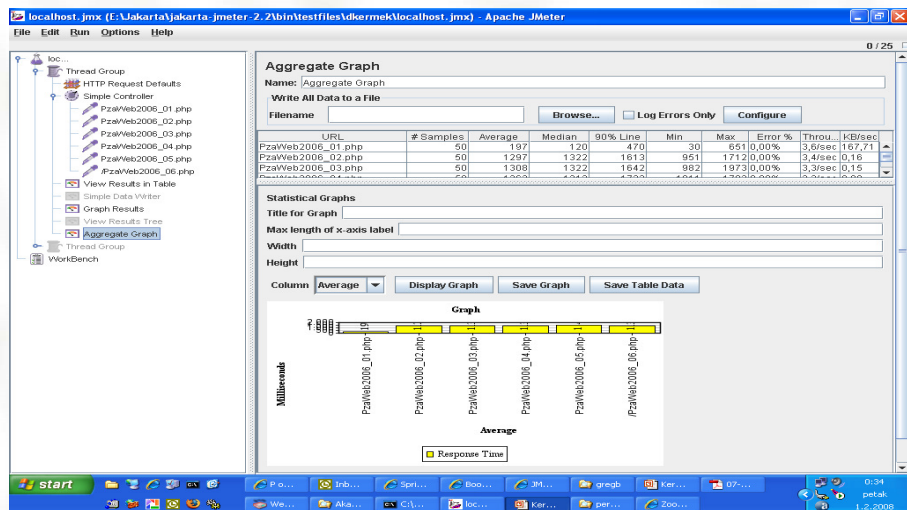


24

Web dizajn i programiranje - FOI  
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



# Apache JMeter



25



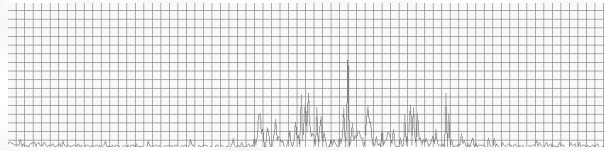
# Podaci iz testova

Test	Broj novih korisnika	Broj starih korisnika	Broj test stanica	Ukupno korisnika	Redni broj konfiguracije	Broj zahtjeva	% izvršenja	Broj pogrešaka	Prosječna propust ost u sec	Prosječno vrijeme odziva	90% linija	KB/s
1.	10	20	1	30	1	416	99,76	1	1,3	77	189	19,5
2.	10	20	14	420	1	6291	87,11	809	19,2	4531	10000	570,1
3.	20	40	1	60	1	1174	98,83	1	3,6	106	267	184,4
4.	20	40	14	840	1	11384	59,50	4600	34,7	5901	5471	701,9
5.	20	40	1	60	2	1136	99,82	1	3,6	120	279	172,4
6.	20	40	14	840	2	11401	63,17	4202	32,9	5725	10049	692,6
7.	20	40	1	60	3	979	100,00	0	3,1	73	160	101,2
8.	20	40	14	840	3	13240	100,00	0	40,3	538	1104	1209,5

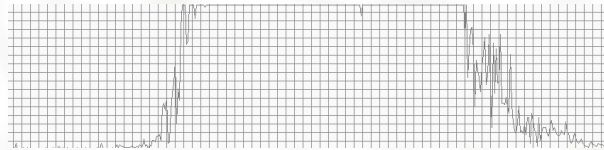
26



## Analiza rezultata



Stanje na procesoru tijekom 1. testa



Stanje na procesoru tijekom 2. testa

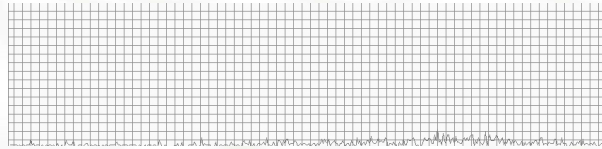


## Stanje na SUBP

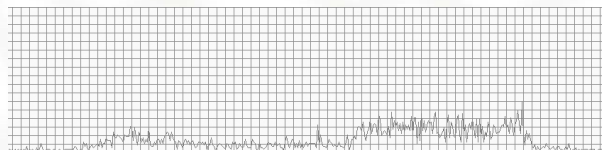
PID	User	Host	Time	State	Info
2	root	localhost:1160	0		SHOW FULL PROCESSLIST
3	root	localhost:1161	1		
11805	tester	D19RM1:4431	23	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11808	tester	D19RM1:4434	22	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11813	tester	D19RM1:4442	22	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11815	tester	D19RM1:4445	21	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11834	tester	D19RM1:4487	19	Sorting result	select zip, city, state, county from zip_codes order by city, state, county, zip limit 4
11849	tester	D19RM1:4510	17	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11857	tester	D19RM1:4538	15	Sending data	select * from (select zip, city, state, county, round(RAND() * (select county) FROM
11859	tester	D19RM1:4530	15	Sorting result	select zip, city, state, county from zip_codes order by city, state, county, zip limit 2
11860	tester	D19RM1:4535	15	Sorting result	select zip, city, state, county from zip_codes order by city, state, county, zip limit 2
11868	tester	D19RM1:4558	13	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11870	tester	D19RM1:4561	13	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 4
11871	tester	D19RM1:4562	13	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 4
11876	tester	D19RM1:4571	12	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 3
11877	tester	D19RM1:4575	12	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 3
11884	tester	D19RM1:4589	11	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 1
11887	tester	D19RM1:4602	11	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 2
11892	tester	D19RM1:4609	10	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11893	tester	D19RM1:4611	10	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 4
11894	tester	D19RM1:4614	10	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, zip limit 2
11895	tester	D19RM1:4615	10	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, zip limit 2
11900	tester	D19RM1:4613	8	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11907	tester	D19RM1:4643	8	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 4
11908	tester	D19RM1:4646	8	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 4
11911	tester	D19RM1:4650	7	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE
11912	tester	D19RM1:4652	7	Sorting result	select zip, city, state, county from zip_codes order by zip, state, county, city limit 2
11916	tester	D19RM1:4658	6	Sending data	select zip, city, state, county from zip_codes where state in (select state from (SELE



## Analiza rezultata



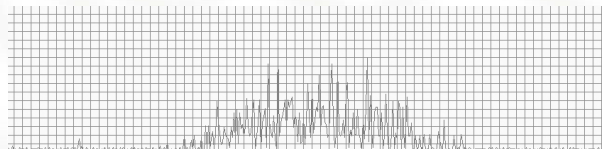
Stanje na procesoru Web i email poslužitelja tijekom 5. testa



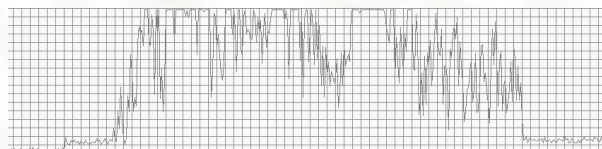
Stanje na procesoru Web i email poslužitelja tijekom 6. testa



## Analiza rezultata



Stanje na procesoru poslužitelja baze podataka tijekom 5. testa

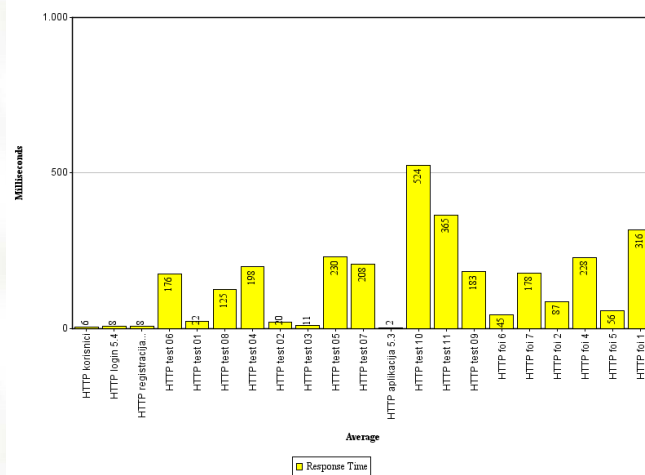


Stanje na procesoru poslužitelja baze podataka tijekom 6. testa





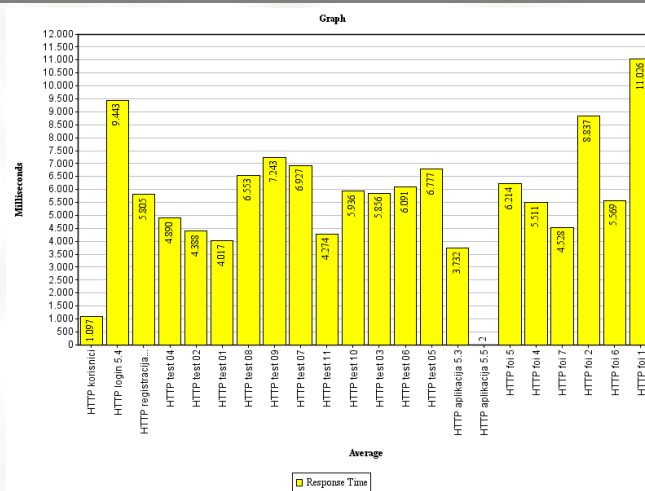
## Analiza rezultata – vrijeme odziva



Prosječno vrijeme odziva po pojedinim zahtjevima za 5. test



## Analiza rezultata – vrijeme odziva

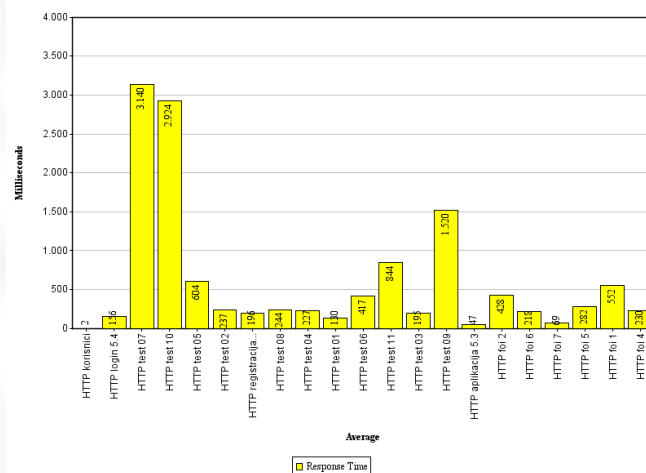


Prosječno vrijeme odziva po pojedinim zahtjevima za 6. test





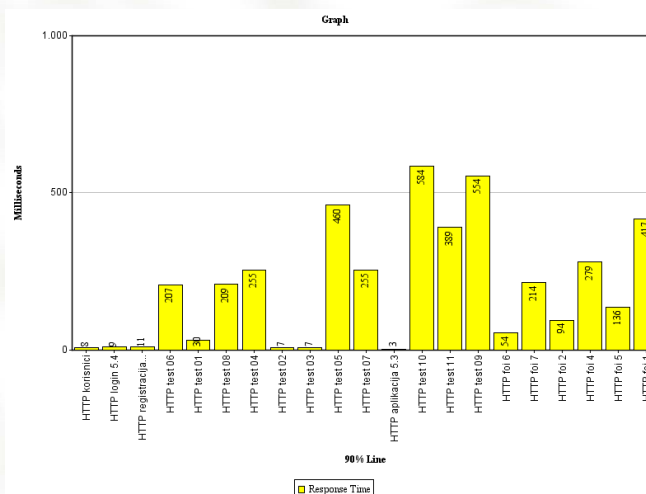
## Analiza rezultata – vrijeme odziva



Prosječno vrijeme odziva po pojedinim zahtjevima za 8. test



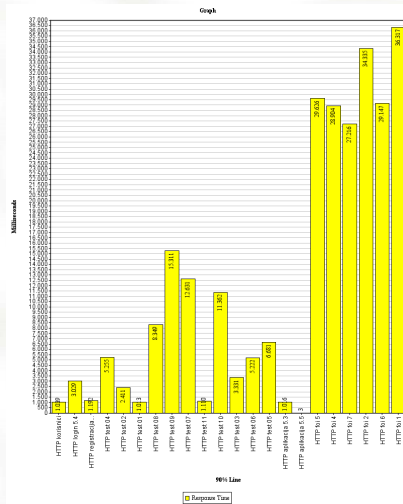
## Analiza rezultata – 90% linija



90% linija po pojedinim zahtjevima za 5. test



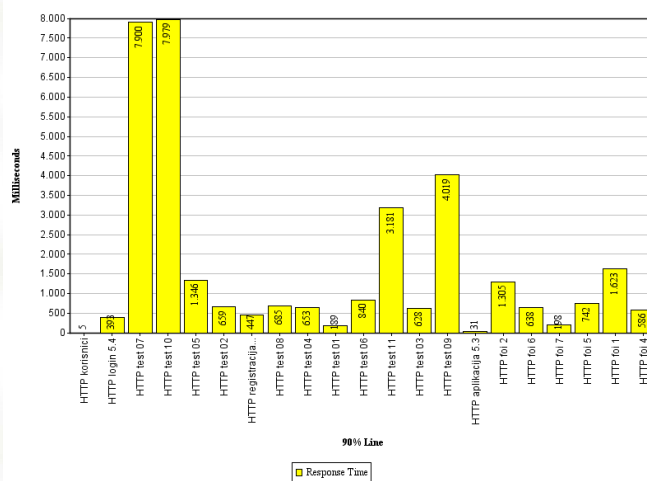
## Analiza rezultata – 90% linija



**90% linija po pojedinim zahtjevima za 6. test**



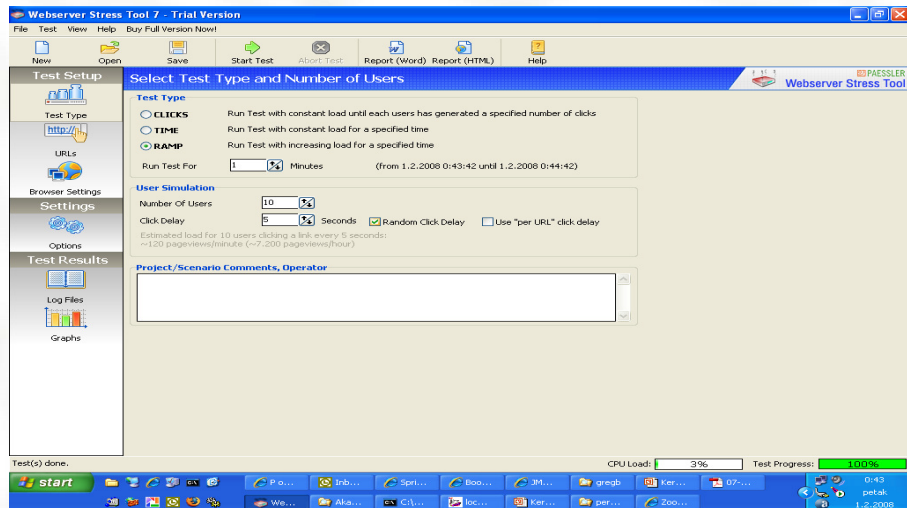
## Analiza rezultata – 90% linija



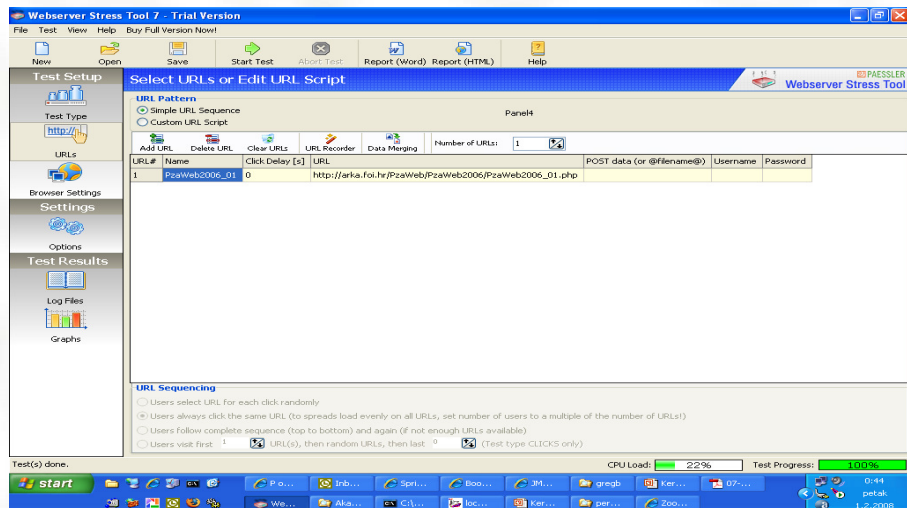
**90% linija po pojedinim zahtjevima za 8. test**



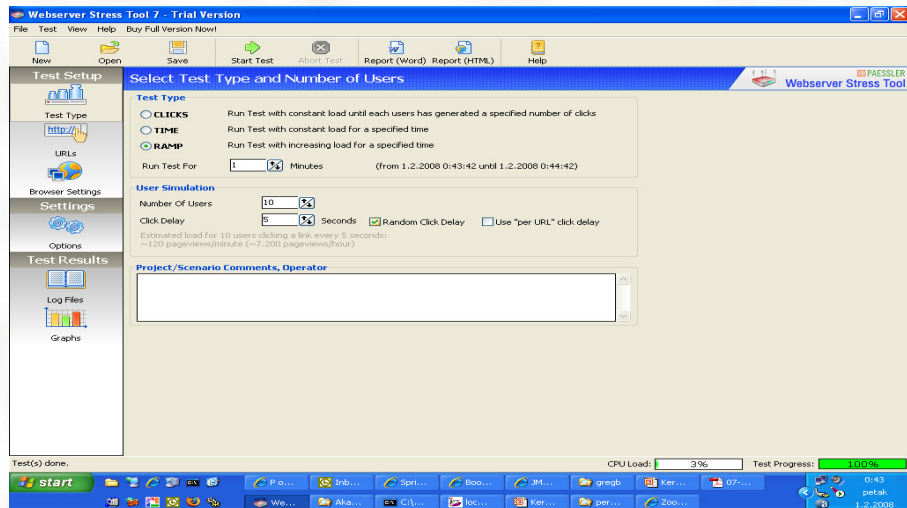
## Webserver Stress Tool



## Webserver Stress Tool



## Webserver Stress Tool

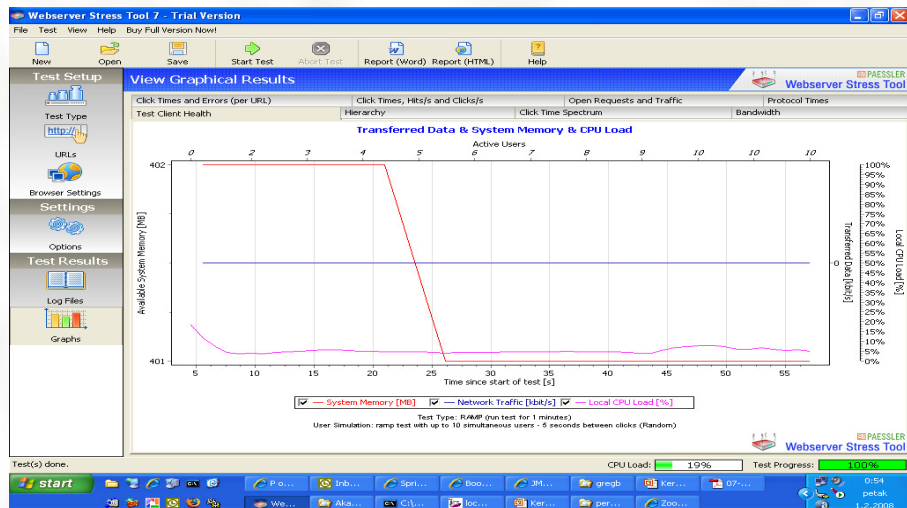


39

Web dizajn i programiranje - FOI  
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



## Webserver Stress Tool



40

Web dizajn i programiranje - FOI  
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



## Optimizacija Web sustava

- Dodavanje poslužitelja za bazu podataka
- Podešavanje parametara poslužitelja baze podataka
- Reorganizacija sheme baze podataka
- Redundacija podataka kako bi se smanjilo ili ukinulo povezivanje tablica prilikom SQL upita.
- Optimizaciju programskog koda Web aplikacije kao što je Rasmus Lerdorf prikazao na primjeru PHP.
- Preporuke za prezentacijski dio Web aplikacija tj. na osobine HTTP protokola, HTML jezika, smještaj JavaScript i CSS datoteka, interno međuspremanje podataka kod AJAX-a i sl.
- Aktivno-redundantni elementi, za raspored opterećenja, specijalizirani Web i aplikacijski poslužitelji, zajedničko spremište podataka, replikacija podataka i sl.



## Korištena i dodatna literatura

- J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea, Performance Testing Guidance for Web Applications, Microsoft Corporation, 2007.  
<http://www.codeplex.com/PerfTestingGuide>
- Response Times: The Three Important Limits,  
<http://www.useit.com/papers/responsetime.html>
- Web Performance Matters - Journal - Acceptable Response Times,  
<http://www.webperformancematters.com/journal/2007/7/10/acceptable-response-times.html>
- Rasmus Lerdorf, Web 2.0 and PHP 5, Fakultet organizacije i informatike, Apr.19, 2006. Varaždin, <http://talks.php.net/show/varazdin>
- Steve Souders, High-Performance Web Sites, Communications of the ACM, vol. 51, no. 12, 2008.
- Paul Dyson, Andrew Longshaw, Architecting Enterprise Solutions: Patterns For High-capability Internet-based Systems, John Wiley & Sons, 2004.

