

Razvoj alata AnaLog za praćenje i analizu performansi realnog web baziranog sistema

Miloš Micić, Dragan Okiljević, Jelica Protić

Sadržaj — Predložen je metodološki pristup analizi performansi realnog Web baziranog sistema realizovanog u troslojnoj arhitekturi (Oracle DBMS/J2EE). Pristup analizi bazira se na definisanju uskih grla sistema i njihovom nadgledanju kroz logove operativnog sistema, baze podataka i aplikativnog servera. Realizovan je i opisan alat koji se koristi za analizu pomenutih logova. Rezultati dobijeni korišćenjem realizovanog alata na realnom produkcionom sistemu prikazani su grafički. Predložena metoda se može koristiti u praćenju, očuvanju i unapređenju performansi produkcionih Web baziranih sistema.

Ključne reči — analiza performansi, DBMS, J2EE.

I. UVOD

Ogroman potencijal koji sa sobom nosi razvoj Interneta u poslednje dve decenije ispoljava se u vidu nastajanja novih tehnologija i aplikacija koje koriste prednosti distribuiranih web okruženja. Praktične prednosti koje pružaju ovakvi sistemi u odnosu na centralizovane dovele su kako u svetu, tako i kod nas do primetnog trenda porasta njihovog razvoja i primene za podršku radu u komercijalnom, bankarskom i ostalim sektorima savremenog poslovanja [1]. Razvoj web baziranih sistema je postao široko prihvatljiv i pristupačan između ostalog i kao posledica inovacija u domenu programskih jezika, baza podataka, aplikativnih servera i razvojnih alata koji u mnogome olakšavaju njihovo realizovanje. Osnovni zahtevi koje ovi sistemi treba da ispune jesu robustnost, skalabilnost, dostupnost i sigurnost, ali ipak pre svega kvalitet servisa [2]. Skalabilnost obezbeđuje mogućnost proširenja i povećanja kapaciteta sistema u slučajevima kada je to potrebno, pri čemu se vodi računa o odnosu cene i kvaliteta, tj. performansi [3]. S obzirom da je korišćenje većine web sistema stohastički proces i da ima dinamičku prirodu, potrebni kapacitet mora da se predvidi i po potrebi prilagodi novom opterećenju u cilju očuvanja kvaliteta servisa. Samim tim, javlja se potreba za razvojem metoda analize i praćenja performansi web baziranih sistema.

Modelovanje performansi web sistema je posebna problematika kojoj se može pristupiti na više načina. Jedan

od pristupa je čisto teorijski i bazira se na definisanju matematičkog modela sistema, najčešće primenom redova za čekanje, ili u skorije vreme upotrebom novih modela kao što su na primer empirijski, statistički i modeli komponenti [4]. Korak dalje predstavlja verifikacija modela u realnim situacijama, ali kako bi se domen posmatranja problema sveo u što jasnije okvire, najčešće se umesto realnih sistema koriste njihove simulacije. Ovako dobijeni modeli performansi se mogu u određenom nivou primeniti u cilju uspešnog dizajniranja web sistema u početnim fazama razvoja. Međutim, simulatori koji se koriste u verifikacijama modela najčešće ne odgovaraju ponašanju stvarnih sistema koji imaju veoma dinamičke karakteristike i koji su podložni širokom spektru promena tokom vremena, koje se odnose na: navike korisnika sistema, izdavanje novih verzija sistema, količine podataka sa kojima sistem funkcioniše i sl. Može se reći da je modelovanje performansi bitno u prvoj fazi razvoja projekta, ali se ipak javlja potreba za definisanjem metodološkog pristupa za praćenje i analizu performansi sistema u produkciji.

U II delu ovog rada je opisan web bazirani sistem, predmet dalje analize, koji se aktivno koristi u realnom produkcionom okruženju [5]. III deo daje detalje o metodološkom pristupu prikupljanja ciljanih podataka o performansama sistema koji se koriste u daljoj obradi, kao i opis načina funkcionisanja alata AnaLog koji predstavlja realnu primenu metodologije. IV deo prikazuje rezultate merenja i diskusiju, dok je u zaključku dat osvrt na dalju upotrebu opisanog pristupa, kao i mogućnosti za proširenje domena njegovog korišćenja.

II. OPIS ANALIZIRANOG REALNOG WEB BAZIRANOG SISTEMA

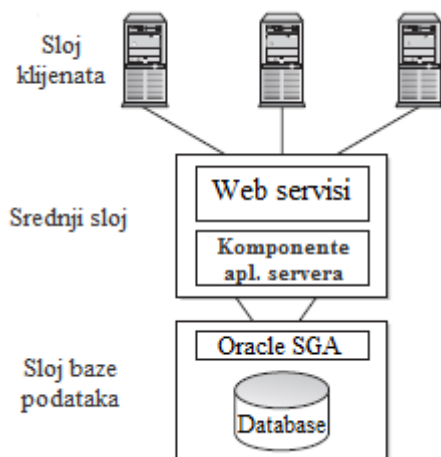
Pod web baziranim sistemom podrazumeva se aplikacija kojoj klijenti pristupaju putem TCP/IP protokola. Svi zahtevi se upućuju aplikativnom serveru. Vreme odziva se definiše kao vreme koje protekne od prihvatanja zahteva do slanja odgovora klijentu koji predstavlja HTML stranicu u konkretnom slučaju. Posmatrani sistem se sastoji od ukupno 10 aplikativnih servera (Oracle Application Server 10.1.3.1) od kojih je na njih devet postavljeno devet instanci aplikacije A, dok se na desetom obrađuju zahtevi upućeni aplikaciji B. U cilju analize, odabran je jedan od devet podsistema aplikacije A koji obrađuje najviše zahteva i sistem aplikacije B. Zahtevi koji pristižu od korisnika svakoj od instanci sistema A i B su potpuno nezavisni i možemo smatrati da sve ove instance

M. Micić, Računarski centar Univerziteta u Beogradu, Kumanovska bb, 11000 Beograd, Srbija (telefon: 381-64-3072209, e-mail: milos.micic@rcub.bg.ac.rs)

D. Okiljević, Računarski centar Univerziteta u Beogradu, Kumanovska bb, 11000 Beograd, Srbija (telefon: 381-63-8694836, e-mail: dragan.okiljevic@rcub.bg.ac.rs)

J. Protić, Prof. dr, Elektrotehnički fakultet u Beogradu, Bulevar kralja Aleksandra 73, 11120 Beograd, Srbija; (e-mail: jeca@sezampro.rs).

funkcionišu kao odvojene celine troslojne arhitekture (Sl. 1). Sličnosti između A i B su tehnološka platforma i poslovni procesi što ih čini pogodnim za poređenje.

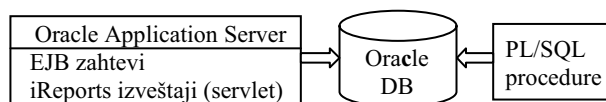


Sl. 1. Troslojna arhitektura sistema A i B.

Aplikativni serveri sistema A i B pokreću J2EE (Java 2 Enterprise Edition) aplikacije modelovane u MVC (Model View Control) arhitekturi. One su napisane u Java programskom jeziku pri čemu su korišćene i sledeće tehnologije: EJB (Enterprise Java Beans) v3.0, Hibernate, iReports, JSF (Java Server Faces), Oracle ADF (Application Development Framework) v10g, JavaScript i druge. Hardware na kome se pokreće svaki od aplikativnih servera sastoji se od dvojezgarinih Intel Xeon 5160 procesora frekvencije 3.00GHz i 3.25GB RAM memorije. Sekundarna memorija su hard diskovi sa brojem obrtaja 7200rpm kojima se pristupa putem SATA interfejsa. Operativni sistemi na ovim serverima su 32-bitni Windows 2003 Release 2 Standard Edition SP2.

Server baze podataka se nalazi u odvojenom sloju od aplikativnog servera. Prednost ovakvog pristupa je u tome da povećanje procesorske aktivnosti bilo na aplikativnom ili sloju baze podataka ne utiče na performanse ostalih komponenti u sistemu. Druga prednost je u tome što je moguće dodati još jednu instancu Oracle aplikativnog servera u slučaju povećanja opterećenja tog dela sistema. Baza podataka je Oracle 10g Enterprise Edition Release 10.2.0.1.0. Hardver na kome se pokreće je sličan onom koji koristi aplikativni server, pri čemu je jedina razlika u količini RAM memorije: 8GB i verziji operativnog sistema na kome se izvršava: 64-bitni Windows 2003.

Oracle baza podataka podržava PL/SQL procedure. Iscrpne obrade, koje su vremenski zahtevne, obavljaju se na sloju baze podataka putem izvršavanja poslova kreiranih pomoću ovih procedura. Na ovaj način se distribuira obrada sa aplikativnog na sloj baze. Proceduralne obrade zauzimaju veliku količinu resursa i direktno utiču na performanse celokupnog sistema i potrebno je voditi računa da se tokom njihovog izvršavanja opslužuju i ostali zahtevi. Svi mogući tokovi pristizanja zahteva bazi podataka prikazani su na Sl. 2.



Sl. 2. Generisanje opterećenja nad bazom podataka.

III. METODOLOGIJA MERENJA I PRAĆENJA PERFORMANSI SISTEMA U PRODUKCIJI

Osnova metodologije modelovanja performansi sistema zasniva se na shvatanju ključnih faktora koji utiču na performanse. Sledeći korak jesu merenja i razumevanje opterećenja sistema. Na osnovu rezultata merenja se definiše odgovarajući analitički model koji je potrebno potvrditi u praksi, a u cilju predikcije i optimizacije performansi u sistemu. Ove faze se mogu ponavljati sa svakim novim ciklusom razvoja aplikacije [6]. Ovakav pristup ima veliku podršku i široku primenu prilikom ispunjavanja početnih zahteva pre puštanja aplikacije u produkciju. Predmet ovog rada se bazira na modifikaciji ovog pristupa u slučaju realnog sistema koji već ispunjava trenutne zahteve vezane za performanse. Ideja je da se prati funkcionisanje sistema nadgledanjem ključnih faktora koji utiču na performanse. Potom se uočavaju pravila i zavisnosti koja postoje u sistemu na osnovu dobijenih rezultata merenja. Na osnovu učenog trenda kretanja posmatranih rezultata merenja moguće je izvršiti određene predikcije i optimizacije.

Na osnovu analize troslojne arhitekture sistema, definišu se tačke u kojima se mere parametri potrebni za dalju analizu. Moguće je grupisati ih u sledeće kategorije: podaci aplikativnog servera, podaci o serveru baze podataka i parametri performansi funkcionisanja operativnih sistema koji su podrška ovim servisima. Njih je moguće prikupiti iz logova aplikativnog, servera baze podataka i sistemskih logova.

A. Logovi Oracle 10g aplikativnog servera

OHS (Oracle HTTP server) je važan deo Oracle 10g aplikativnog servera. OHS-u se pristupa najmanje dva puta prilikom svake transakcije. Usled toga, Oracle je obezbedio mehanizam za praćenje OHS komponenti putem generisanja logova korišćenjem alata za dinamički monitoring, tzv. DMSTool (Dinamic Monitoring Service Tool) [7]. Ovaj alat predstavlja program koji se pokreće iz komandne linije i koristi se za prikaz akumulativnih metrika performansi različitih komponenti aplikativnog servera kao što je prikazano u Tabeli 1.

TABELA 1: KOMPONENTE KOJE PRIKAZUJE DMSTOOL.

Komponenta	Opis
OC4J	Oracle AS kontejner za J2EE – prikazuje vreme potrebno za parsiranje dolazećih zahteva kao i količina zauzete JVM (Java Virtual Machine) memorije.
Servlet kod	DMS metrika Java koda
OHS	Aktivni zahtevi upućeni Oracle HTTP serveru

Generišu se metrike u razmaku od jednog minuta pri čemu se one snimaju u fajl smešten na lokalnom disku u cilju dalje analize. Svi podaci koje generiše DMSTool se

prikazuju akumulativno u odnosu na vreme od kada je startovan aplikativni server. Ovako prikupljeni podaci su preobimni i nepodesni za dalju analizu bez dodatne aplikativne podrške.

U cilju analize generisanih logova putem korišćenja DMSTool-a osmišljen je alat AnaLog napisan u Java programskom jeziku. Osnovna namena ovog alata jeste da parsira veliku kolekciju generisanih logova i agregira dobijene podatke. Tako organizovani podaci se grafički prikazuju grupisani po kategorijama i moguće je definisati međusobne relacije podataka i prikazati ih vizuelno u okviru iste aplikacije. Rezultat toga je grafički prikaz agregiranih podataka, što pojednostavljuje dalju analizu, a osim toga podržano je i automatsko izveštavanje u vidu prikaza komponenti sistema koje imaju najviše udela u opterećenju aplikativnog servera (Sl. 6-9).

B. Logovi baze podata

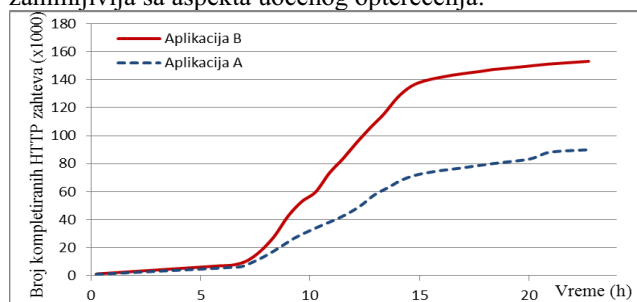
Oracle 10g DBMS (Database Management System) nudi brojna postojeća rešenja za snimanje i analizu performansi Oracle baze podataka. U našem slučaju, koriste se rešenja uvedena sa pojavom verzije 10g: ASH (Active Session History) i AWR (Automatic Workload Repository). ASH beleži detaljne informacije o svim događajima relevantnim za performanse baze podataka sa rezolucijom od jedne sekunde. AWR beleži isti tip informacija sa rezolucijom od 10 sekundi, ali za duži vremenski period (podrazumevano je nedelju dana). Informacije potrebne za merenja su ekstrahovane upitom u ASH i AWR sistemske tabele u kojima je logovana aktivnost baze u posmatranom periodu [8].

C. Sistemski logovi

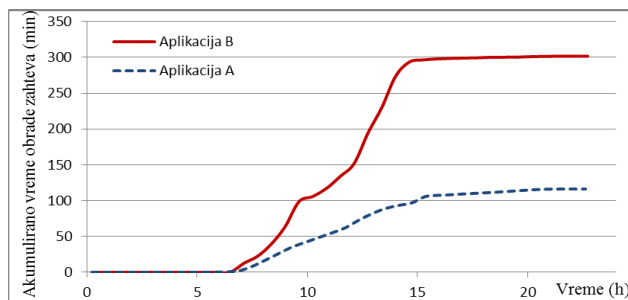
Sistemske performanse mašina na kojima su smešteni aplikativni i serveri baza podataka aplikacija A i B su takođe logovane i ekstrahovane za dalju analizu. Pod sistemskim performansama se podrazumeva praćenje sledećih parametara: iskorišćenje CPU, performanse diskova, opterećenje memorije i mrežnih interfejsa. S obzirom da su operativni sistemi nad kojima su merenja obavljena Windows 2003, korišćeni su ugrađeni brojači performansi operativnog sistema pomoću "Performance monitor" alata u intervalima od jednog minuta, a rezultati su prebačeni u bazu podataka u cilju lakše dalje analize.

IV. REZULTATI MERENJA I DISKUSIJA

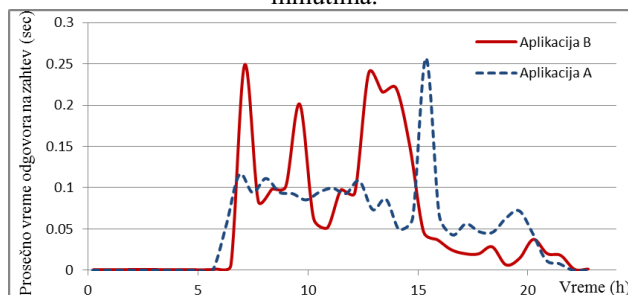
Analizom rezultata merenja na nivou od jednog dana dobijeni su grafici za sve navedene tipove merenja obe aplikacije. U daljem tekstu će biti prikazani rezultati analize obe aplikacije sa akcentom na aplikaciju B koja je zanimljivija sa aspekta uočenog opterećenja.



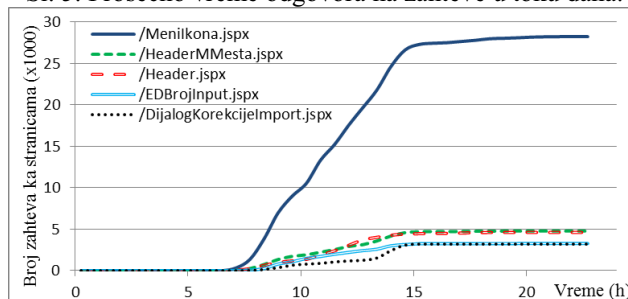
Sl. 3. Broj kompletiranih HTTP zahteva od početka dana.



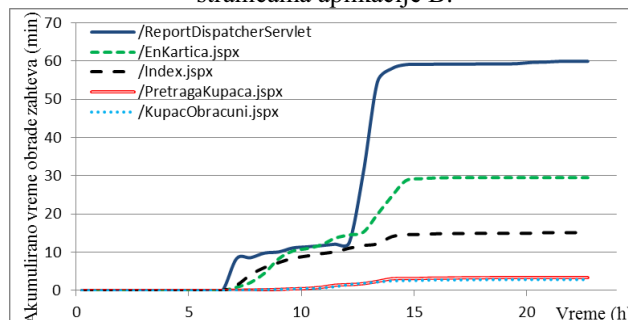
Sl. 4. Akumulativno vreme obrade HTTP zahteva u minutima.



Sl. 5. Prosečno vreme odgovora na zahteve u toku dana.



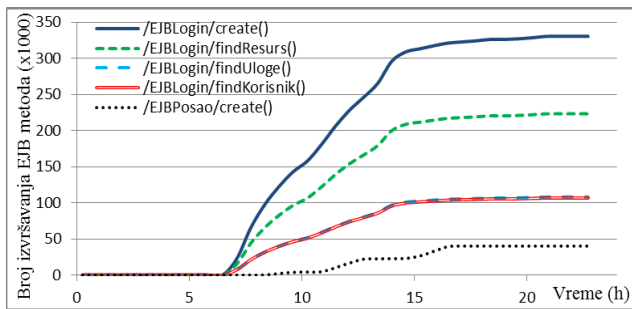
Sl. 6. Akumulativan broj zahteva ka najčešće pozivanim stranicama aplikacije B.



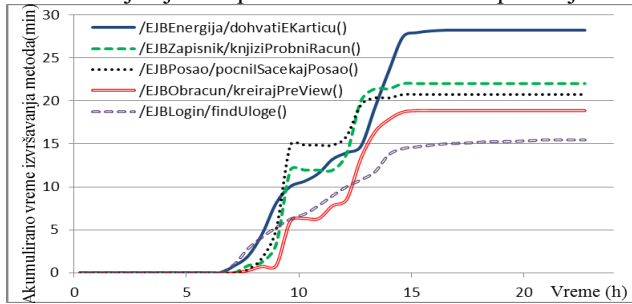
Sl. 7. Akumulativno vreme obrade zahteva ka servletima i stranicama koje su se najduže obrađivale na aplikaciji B.

Na Sl. 7 se primećuje da se u toku rada aplikacije B dnevno 60 minuta obrađuju zahtevi za štampu. Osim toga, na grafiku se može uočiti skok potrebnog vremena obrade /ReportDispatcherServlet zahteva što sugeriše da je u tom trenutku moglo da nastupi zagušenje u sistemu. U cilju sprečavanja zagušenja, sledeći korak bi bila optimizacija koda dela sistema koji je prouzrokovao skok ukoliko je to moguće, redefinisane poslovnog procesa kako bi se iscrpne obrade obavljale u rasterećenom periodu ili povećanje broja potrebnih resursa.

Obrada zahteva upućenih aplikativnom serveru enkapsulira u sebi obraćanje bazi putem EJB metoda. Grafički prikaz najčešće korišćenih i najduže obrađivanih EJB metoda i servleta predstavlja sledeći korak u analizi u cilju detektovanja potencijalnih uskih grla.

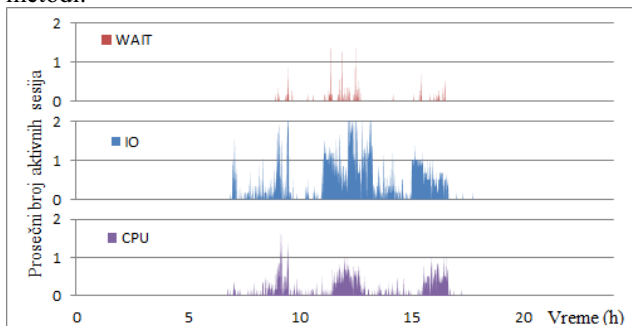


Sl. 8. Broj najčešće pozivanih EJB metoda aplikacije B.



Sl. 9. Akumulativno vreme izvršavanja najduže opsluživanih EJB metoda na aplikaciji B.

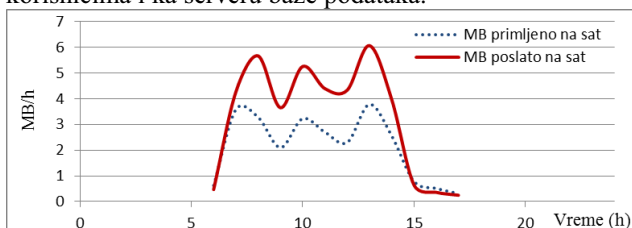
Analizom rezultata dobijenih na Sl. 7 i 9 može se uočiti korelacija između poziva stranice /EnKartica.jspx i EJB metode /EJBEnergiya/dohvatiEKarticu(). Ovo sugeriše da celokupno opterećenje sistema može da se ublaži redizajniranjem programskog koda koji se koristi u ovoj metodi.



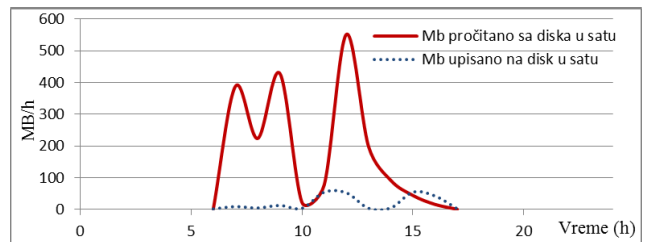
Sl. 10. Ukupan broj aktivnih sesija nad IO, CPU ili nekim drugim resursom (WAIT) baze podataka aplikacije B.

Grafik na osnovu AWR i ASH logova (Sl. 10) prikazuje odnos ukupnog vremena koje su procesi na bazi proveli u izvršavanju, ili čekajući na CPU, I/O i druge resure i vremena koje je zaista proteklo (Oracle Average Active Sessions metrika [9]).

Podaci od interesa generisani na osnovu sistemskih logova su protok na mrežnom interfejsu aplikativnog servera (Sl. 11) i količina upisa i čitanja podataka sa diskova (Sl. 12). Pri tome treba imati u vidu da mrežni saobraćaj aplikativnog servera može da bude usmeren i ka korisnicima i ka serveru baze podataka.



Sl. 11. Mrežni protok na interfejsu aplikativnog servera B.



Sl. 12. Količina podataka upisana i pročitana sa diska baze podataka B.

V. ZAKLJUČAK

Značaj opisanog postupka za nadgledanje i analizu performansi realnog web baziranog sistema leži u mogućnosti detektovanja konkretnih problema i uzroka potencijalnih uskih grla u sistemu. Promene u izvršavanju određenih metoda, nastale usled dinamičke prirode realnih sistema, moguće je na vreme otkriti i pravovremenim akcijama otkloniti. Dalji razvoj sistema AnaLog kreće se u smeru automatizovane predikcije potencijalnih pretnji koje mogu uticati na performanse sistema i kvalitet servisa.

LITERATURA

- [1] D. A. Menasce, D. Barbara, R. Dodge, "Preserving QoS of e-commerce sites through self-tuning: A performance model approach," in *Proc. 3rd ACM conference on Electronic Commerce table of contents*, Tampa, Florida, USA, pp. 224-234, 2001.
- [2] C. U. Smith, L. G. Williams, "Best Practices for Software Performance Engineering," in *Proc. CMG*, Dallas, Dec. 2003.
- [3] C. U. Smith, L. G. Williams, "Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software," Addison-Wesley, 2002.
- [4] D. Okiljević, M. Micić, Ž. Stanisavljević, J. Protić, "Klasifikacija modela za analizu performansi web baziranih sistema," YU Info, Kopaonik, 2010.
- [5] G. Prelić, Z. Bolić, "Implementacija softverskog paketa SONEE u Elektro distribuciji Beograd," prezentacija Oracle korisnika, SMOUG, Beograd, 2009.
- [6] D. A. Menasce, V. A.F. Almeida, "Capacity planning for Web Services : metrics, models, and methods," Prentice Hall, Upper Saddle River, NJ, 2002.
- [7] J. Garmany, D. K. Burlson, "Oracle Application Server 10g Administration Handbook," Oracle Press, McGraw-Hill, 2004.
- [8] T. Kyte, "Expert Oracle Database Architecture: Oracle Database 9i, 10g, and 11g Programming Techniques and Solutions," Second Edition, Oracle Corporation, Apress, 2005.
- [9] G. Wood, U. Shaft, J. Beresiewicz, "Performance Fundamentals for Oracle Database 10g and 11g," Oracle Corporation, 2008.

ABSTRACT

The paper presents methodology for performance analysis of a 3-tier Web-based systems (Oracle DBMS/J2EE architecture). The approach is based on defining and monitoring system bottlenecks using operating system, database and application server logs. The tool for analyzing these logs is developed and used on real, production system. The results gathered using this tool are presented graphically and analyzed. The presented method can be used for monitoring, improving and maintaining performance of production Web-based systems.

DEVELOPMENT OF TOOL "ANALOG" FOR PERFORMANCE MONITORING AND ANALYSIS OF A REAL WEB-BASED SYSTEM

Miloš Micić, Dragan Okiljević, Jelica Protić