

RAZVOJ APLIKACIJE ZA PRAĆENJE RADA PROCESA**DEVELOPMENT OF A PROCESS MONITORING APPLICATION**Isidora Ninković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO**

Kratak sadržaj – *Ovaj rad se bavi softverskim alatom koji služi za nadgledanje i procenu rada izabranih procesa. Nadgledanje rada procesa predstavlja redovno praćenje i evidentiranje njihovih aktivnosti. Potreba za nadgledanjem rada izabranih procesa proističe iz potrebe za informisanjem o performansama rada programa kojem procesi pripadaju.*

Ključne reči: *nadgledanje procesa, performanse, evaluacija rada, izveštaj o radu*

Abstract – *Software tool for process monitoring and evaluating activities of selected processes is presented in this paper. Monitoring is the regular observation of the processes and recording of their activities. The requisit for activity observation of selected processes arises from the need for performance information about processes which belongs to programme of interest.*

Keywords: *process monitoring, performances, evaluation of work, report about work*

1. UVOD

Kvalitet proizvoda ne bi trebalo da bude kompromis između troškova i karakteristika proizvoda, već ga je nužno ostvariti kako bi proizvod bio što primenljiviji, dugotrajniji, mogao da radi što pouzdanije sa dobrim performansama, ali i kako bi njegovi korisnici bili što zadovoljniji. Ukoliko je proizvod kvalitetan na dugoročnom planu se smanjuju troškovi održavanja i stiču se lojalni korisnici [1].

Kada je reč o kvalitetu softverskog proizvoda podrazumeva se, pre svega, rad uz dobre performanse i pružanje pouzdane usluge, ali i lakoća i intuitivnost korišćenja za širi spektar korisnika, robusnost, određen nivo sigurnosti i bezbednosti. Kako bi se izgradio softverski proizvod koji karakteriše kvalitet na visokom nivou neophodno je pratiti njegov rad još od prvih stadijuma razvoja.

Nadgledanje rada procesa koji pripadaju programu od interesa obuhvata svakodnevno rutinsko prikupljanje podataka o svim aspektima njihovog rada. Prikupljene informacije su značajne na različitim poljima. Tokom faza razvoja softverskog proizvoda softver arhitekta moraju prikupiti što više informacija o radu kako bi doneli ispravne odluke o arhitekturi aplikacije, mogućim

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

izmenama koda, ali i o tome da li se aplikacija ponaša kako je predviđeno. Nadalje su prikupljene informacije od interesa menadžerima, koji na osnovu njih vrše procenu da li je proizvod u skladu sa trenutnim standardima i da li je dovoljno kompetativan na tržištu, kao i neophodnosti izmene njegove cene.

Rad procesa se može pratiti i preko grafikona, na kojima je predstavljeno kretanje određene vrednosti u vremenu. Grafikoni mogu biti u realnom vremenu ili kreirani za zadati vremenski interval, a preko njih se svakako može otkriti neobično ponašanje procesa programa [2].

Nakon prikupljanja podataka o radu može se napraviti i statistika rada procesa, koja se prikazuje u vidu izveštaja. Putem izveštaja korisnik dobija uvid u istorijske podatke, ali i izračunate podatke od interesa, i može izvršiti poređenje sa podacima iz prethodnih izveštaja.

2. SOFTVERSKI ALAT ZA NADGLEĐANJE RADA PROCESA

Softverski alat opisan u ovom radu prevashodno služi za nadgledanje aktivnosti procesa. Aktivnosti procesa se ogledaju kroz njihove osnovne karakteristike rada. Mogu se i grafički predstaviti putem grafikona, kako bi se dobio slikovit prikaz kretanja izabrane karakteristike rada za određen vremenski period.

Posebna funkcionalnost ovog softverskog alata jeste mogućnost prilagođenja tako da se ostvari povezivanje sa drugim softverskim proizvodom koji korisnik želi da nadgleda. Tačnije, može se izvršiti nadgledanje izabrane aktivnosti željenog softverskog proizvoda.

2.1 Skup procesa koji se prate

Softverski alat koji je predmet ovog rada karakteriše velika mogućnost prilagođenja korisniku, počev od definisanja šireg skupa procesa koji se prate. Skup procesa od interesa se nalazi u xml konfiguracionom fajlu, i korisnik ga može ručno izmeniti putem navođenja procesa koje želi da prati. Procesi se u xml konfiguracionom fajlu navode putem svog naziva, kao što je prikazano na slici 1. Xml konfiguracioni fajl sa skupom procesa koji se žele nadgledati se učitava prilikom pokretanja aplikacije, tako da je sve izmene neophodno pre pokretanja uneti.

Izgled početnog prozora koji se korisniku otvara prilikom pokretanja softverskog alata je prikazan na slici 2. U tabu *Processes* korisnik dobija pregled procesa iz unapred definisanog skupa, a svaki proces je naveden po nazivu i jedinstvenom identifikatoru (Id). Nakon naziva i Id-ja procesa su navedene osnovne karakteristika njegovog rada:

1. *CPU* – trenutno zauzeće procesora od strane procesa, izraženo u procentima,
2. *Memory* – trenutno zauzeće memorije računara od strane procesa, izraženo u MB, i
3. *RAM* – trenutno zauzeće *RAM (Random Access Memory)* memorije računara od strane procesa, izraženo u MB.

Karakteristike rada procesa se osvežavaju na svakih 1 sekund, tako da korisnik ima uvid u najsvežije podatke.

```

1 <?xml version="1.0"?>
2 <ArrayOfString xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3 <string>StabilityAnalysesService</string>
4 <string>JobScheduler</string>
5 <string>CalculationExecutor</string>
6 <string>CoreFunctionsService</string>
7 <string>PowerControlSystemService</string>
8 <string>FunctionHost</string>
9 <string>OutrageProcessingService</string>
10 <string>XACTCommandQSpooler</string>
11 <string>LocalContextManager</string>
12 <string>LoadForecastService</string>
13 <string>ModelManagerService</string>
14 <string>ModelTopologyService</string>

```

Slika 1. Xml fajl za konfiguraciju liste procesa od interesa

Name	Id	CPU [%]	Memory [MB]	RAM [MB]
DMS_StabilityAnalysesService	476	0	555.7	365.9
DMS_JobScheduler	15776	0	124.8	91.6
DMS_CalculationExecutor	25380	0	66.1	45.1
DMS_CalculationExecutor	7044	0	66.5	45
DMS_CalculationExecutor	22096	0	66.1	45
DMS_CalculationExecutor	25380	0	66.1	45.1
DMS_CalculationExecutor	7044	0	66.5	45
DMS_CalculationExecutor	22096	0	66.1	45
DMS_CalculationExecutor	25380	0	66.1	45.1

Slika 2. Izgled početnog prozora

2.2 Istorijska baza podataka

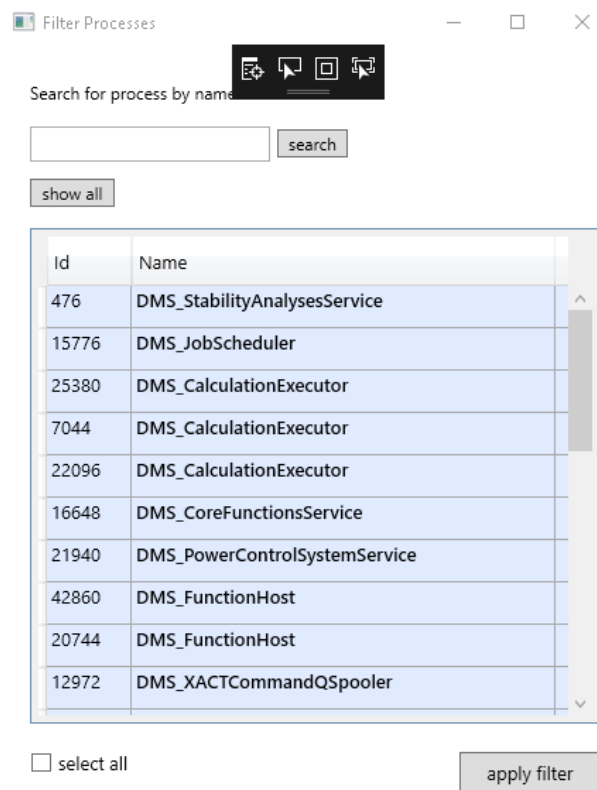
Prilikom očitavanja novih vrednosti karakteristika rada na svakih 1 sekund vrši se njihov prikaz za korisnika, ali i prethodno zapisivanje u istorijsku bazu podataka.

Za potrebe ovog softverskog alata je korišćena *SQLite* baza podataka. *SQLite* je mala, brza baza podataka koja se lako može ugraditi u aplikaciju. Posebna je po tome što čuva sve podatke u samo jednom fajlu. Ne koristi spoljne biblioteke i interfejs, osim nekoliko standardnih C biblioteka, što takođe predstavlja njenu prednost [3].

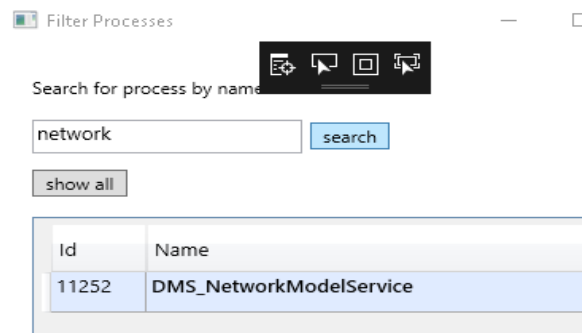
Potreba za istorijskom bazom podataka postoji zbog opcije kreiranja grafičkog prikaza koja se nudi korisniku u okviru taba *Performance charts*, što će naknadno biti opisano u poglavlju 2.3. Takođe, u sastavu posebne funkcionalnosti ovog softverskog alata se nakon praćenja određene aktivnosti procesa kreira izveštaj. Kako bi se kreirao izveštaj za vremenski period trajanja aktivnosti procesa neophodni su istorijski podaci o njegovom radu. Posebna funkcionalnost softverskog alata koji je predmet ovog rada će detaljnije biti opisana u poglavlju 2.4.

2.3 Filtriranje procesa

U okviru taba *Processes* korisniku se nudi i opcija dodatnog filtriranja skupa procesa za praćenje, putem klika na dugme *Filter processes*. Po izboru ove opcije pojavljuje se dodatan prozor sa listom svih dostupnih procesa za praćenje i mogućnošću selekcije njenog podskupa. Svi dostupni procesi za praćenje jesu procesi koje je korisnik naveo u xml konfiguracionom fajlu pre pokretanja aplikacije. Korisnik može pretražiti listu po nazivu i Id-ju procesa ukoliko je to neophodno zbog dužine liste i smanjene mogućnosti snalaženja. Nakon izvršene selekcije i klika na dugme *Apply filter* se filter primenjuje, i samo izabrani procesi nastavljaju da se prate. Nakon gašenja aplikacije filtrirana lista se čuva, tako da kada korisnik pokrene aplikaciju sledeći put i dalje će imati samo prikaz filtrirane liste procesa. Izgled prozora za filtriranje je prikazan na slici 3, dok primer pretrage je prikazan na slici 4.



Slika 3. Izgled prozora za filtriranje



Slika 4. Primer pretrage

2.4 Grafici performansi

Drugi tab – *Performance charts* korisniku nudi mogućnost kreiranja grafikona za izabrani proces i vremenski period, prikazujući kretanje vrednosti izabrane karakteristike rada (*CPU*, *Memory*, *RAM*) u izabranom vremenskom periodu. Grafički prikaz je često dobar izbor jer korisnici na osnovu njega na intuitivan način mogu doći do zaključaka. Na slici 5 je prikazan tab *Performance charts*. Pri vrhu prozora se nalaze polja putem kojih korisnik vrši izbor parametara za kreiranje grafikona. Polja za izbor vremenskog perioda su posebno detaljna, i korisniku omogućavaju unos tačnog datuma i vremena u formatu HH:mm:ss (sat: minut:sekund format). Nakon klika na dugme *show chart* započinje proces kreiranja grafikona. Kako bi grafikon bio kreiran neophodno je iščitati podatke o radu procesa iz istorijske baze podataka. Jedan primer kreiranja grafikona je dat na slici 5.

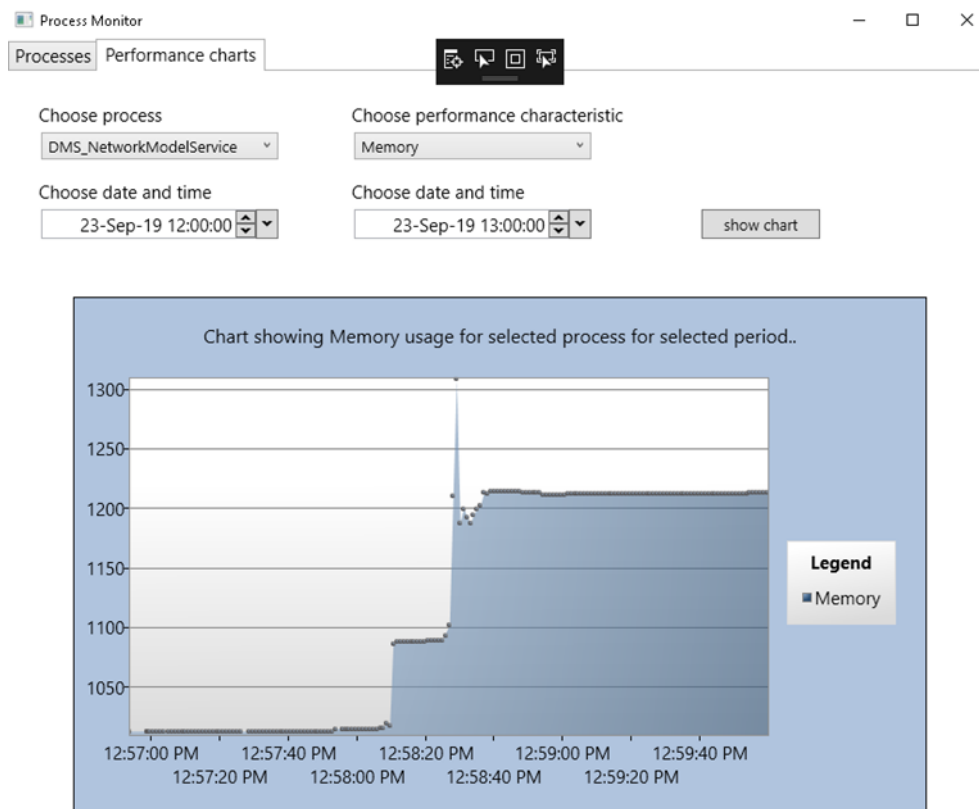
2.5 Praćenje izabrane aktivnosti

Softverski alat za nadgledanje rada procesa nudi korisniku posebnu funkcionalnost – povezivanje sa željenim programom i praćenje jedne ili više njegovih aktivnosti. Kako bi se ova funkcionalnost ostvarila neophodno je u manjoj ili većoj meri prilagođenje ovog softverskog alata. Program od interesa, čija se jedna ili više aktivnosti prati,

može biti posmatran kao “server” koji izvršava željenu operaciju na zahtev. Aplikacija koja je predmet ovog rada predstavlja “klijenta” koji zahteve inicira, tačnije šalje zahtev “serveru” i pokreće operaciju čije će izvršavanje naknadno pratiti.

Prvi korak u ostvarenju ove funkcionalnosti svakako jeste povezivanje “klijentske” i “serverske” aplikacije, što predstavlja i najveći izazov u njenom ostvarenju. Nakon toga sledi implementacija slanja zahteva za izvršenje određene operacije. Ovaj korak može biti odrađen kao u potpunosti spreman zahtev za slanje, ili da se korisniku ostavi određen nivo slobode da u zahtevu šalje testne podatke koje želi.

Svrha funkcionalnosti praćenja izabrane aktivnosti željenog programa jeste prikupljanje podataka o radu programa u toku izvršenja aktivnosti, kao i kreiranje statistike o njegovom radu u toku izvršenja aktivnosti. Zato je u okviru ovog softverskog rešenja implementirano i kreiranje izveštaja nakon što “serverska” strana završi posao. U izveštaju se nalaze opšti podaci o izvršenoj aktivnosti i grafički prikazi kretanja vrednosti karakteristika rada procesa koji je aktivnost izvršavao. U izveštaj se mogu dodati još neki podaci ukoliko to korisnik želi.



Slika 5. Primer kreiranja grafikona

2.6 Primer praćenja izabrane aktivnosti

Kao jedan primer „servera” za potrebe testiranja aplikacije je uzeta aplikacija iz oblasti elektroenergetike, koja služi za prikaz i upravljanje elektroenergetskom mrežom. Operacija koja je bila interesantna za praćenje jeste *Apply Delta*, tačnije operacija koja služi za dodavanje novog modela mreže nad komom će se raditi.

Komunikacija ostvorena između “klijentske” i “serverske” aplikacije je *WCF* komunikacija. Korisniku je ostavljena sloboda u većoj meri prilikom testiranja ove funkcionalnosti “klijentske” aplikacije time što korisnik bira model/modele mreže koji će biti prosleđeni u sklopu zahteva. Iz tog razloga je bilo neophodno veće prilagođenje aplikacije, kako bi korisniku izbor modela mreže bio dat.

Korisnik bira jedan ili više modela, koji su u formatu CIM/XML fajla. Sledi njihova konverzija u *Delta* format i slanje "serverskoj" aplikaciji. Nakon izvršenja operacije korisniku izlazi *message box* sa osnovnim informacijama o uspešnosti izvršenja operacije i sa mogućnošću kreiranja izveštaja.

Prvi deo izveštaja sadrži osnovne informacije o izvršenju operacije: kada je započela i kada se završila, koliko je trajala, da li je uspešno izvršena, kao i podatke o tome koja *Delta* je primenjena i koliko elemenata je putem nje dodato u mrežu (slika 6).

Date: 10/7/2019

Process name: DMS_NetworkModelService

Operation: ApplyDelta

Delta file name(s): 41.xml,

Total elements inserted: 71

Result type: Succeeded

Start time: 10/7/2019 3:53:41 PM

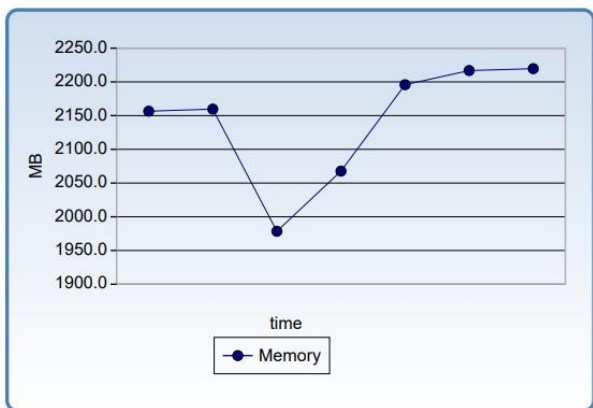
End time: 10/7/2019 3:53:47 PM

Duration: 5.98 seconds

Slika 6. Statističke informacije iz primera izveštaja

Nakon toga slede grafikoni koji predstavljaju kretanje vrednosti korišćenja *CPU*, *Memory* i *RAM* resursa računara od strane procesa koji je operaciju izvršavao. Izgled grafikona iz izveštaja je dat na slikama 7 i 8.

Chart representing Memory usage

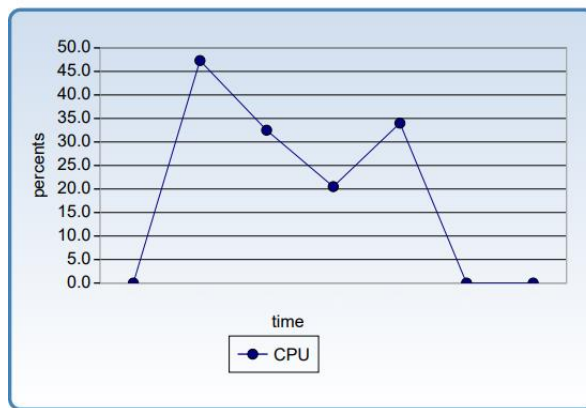


Slika 7. Grafikon iz primera izveštaja koji predstavlja iskorišćenje *memory* resursa računara tokom procesa primene delte

Na osnovu grafikona prikazanog na slici 7 se vidi da je maksimalna vrednost zauzeća memorije računara od strane procesa: 2220 MB. Početna vrednost zauzeća memorije iznosi: 2150 MB, dok krajnja vrednost iznosi: 2220 MB.

Nakon dodavanja novog modela je zauzeće memorije od strane *Network Model Service* procesa povećano za 70 MB.

Chart representing CPU usage



Slika 8. Grafikon iz primera izveštaja koji predstavlja iskorišćenje *cpu* računara tokom procesa primene delte

Zaključuje da je maksimalna vrednost zauzeća procesora od strane nadgledanog procesa iznosila 47 %.

3. ZAKLJUČAK

Primene aplikacije za nadgledanje rada procesa su višestruke, počev od čisto informativnog povremenog nadzora, uveravanja u kvalitet proizvoda, preko analize performansi radi izmene arhitekture aplikacije tokom razvoja ili poboljšanja njenog učinka. Mogućnost povezivanja sa drugom izabranom aplikacijom čije operacije mogu biti nadgledane je čini posebnom i dodatno je prilagođava potrebama korisnika. Time dobija širok spektar korisnika. Softverske arhitekture i korisnici koji se bave razvojem svog proizvoda, kao i menadžeri predstavljaju glavnu ciljnu grupu korisnika. Kreiranje izveštaja najviše koristi prilikom dalje analize i izvođenja zaključaka, a izveštaji sačuvani na računaru korisnika predstavljaju jednu vrstu baze podataka.

4. LITERATURA

- [1] Krishnan M. S., Kriebel C. H., Kekre S., & Mukhopadhyay T. *An empirical analysis of productivity and quality in software products*. Management science, 2000.
- [2] Woodside M., Franks G., & Petriu D. C. *The future of software performance engineering*. Future of Software Engineering. IEEE Computer Society, 2007.
- [3] Owens M. *The definitive guide to SQLite*. Apress, 2006.

Kratka biografija:



Isidora Ninković rođena je u Novom Sadu 1995. god. Fakultet tehničkih nauka, Departman za Elektrotehniku i računarstvo upisala je 2014. god. Odbrana master rada na Fakultetu tehničkih nauka iz oblasti Elektrotehničkog i računarskog inženjerstva – Razvoj aplikacije za praćenje rada procesa, očekuje se u oktobru 2019. god.

kontakt: isidora_ninkovic@hotmail.com