Fakultet tehničkih nauka u Novom Sadu

Virtuelizacija procesa

Tehnička dokumentacija za projekat:

Čitanje i keširanje podataka o potrošnji

Profesori:
Bojan Jelačić
Zoran Janković

Asistenti:
Zorana Babić
Zoran Pajić

Projekat radili studenti:
PR 42/2020 Jovana Orestijević
PR 47/2020 Stefan Delić
PR 54/2020 Tamara Buha
PR 155/2020 Sara Simović

Sadržaj

1. Opis projektnog zadatka

Aplikacija se sastoji od servisa i od klijentske aplikacije. Servis i klijentska aplikacije komuniciraju putem WCF-a. Klijentska aplikacija je konzolna aplikacija koja ima opciju unosa datuma za koji se očekuju podaci.

Servis dobija od klijenta upit za čitanje podataka iz baze podataka, na osnovu prosleđenog datuma. Upit treba da vrati rezultat koji sadrži *Load* objekte za svaki sat za datum iz upita, ukoliko postoje u bazi podataka.
Servis pristupa bazi podataka koja se nalazi u dva oblika: XML datoteka i In-Memory baza podataka. Kada se servis pokrene, In-Memory baza podataka je inicijalno prazna.
Servis prvo pokušava čitanje iz In-Memory baze podataka. Ukoliko podaci postoje u In-Memory bazi podataka, ovi podaci se čitaju i prosleđuju se klijentskoj aplikaciji. Ukoliko podaci ne postoje u In-Memory bazi podataka, servis pokušava čitanje iz XML baze podataka. Ukoliko podaci ne postoje za prosleđeni datum ni u XML bazi podataka, kreira se novi Audit objekat sa odgovarajućom porukom, koji se upisuje u In-Memory bazu podataka i u XML bazu podataka. Ovaj Audit objekat se prosleđuje u vidu rezultata klijentskoj aplikaciji i klijentska aplikacija ispisuje tekst poruke na konzoli. Ako podaci postoje u XML bazi podataka, na osnovu tih podataka kreiraju se odgovarajući *Load* objekti. Jedan objekat klase *Load* predstavlja podatke o prognoziranoj i ostvarenoj potrošnji električne energije za jedan sat. Kreirani Load objekti upisuju se u In-Memory bazu podataka. Sledeći put kada se pojavi odgovarajući upit, ovi podaci će biti pročitani iz In-Memory baze poadataka.

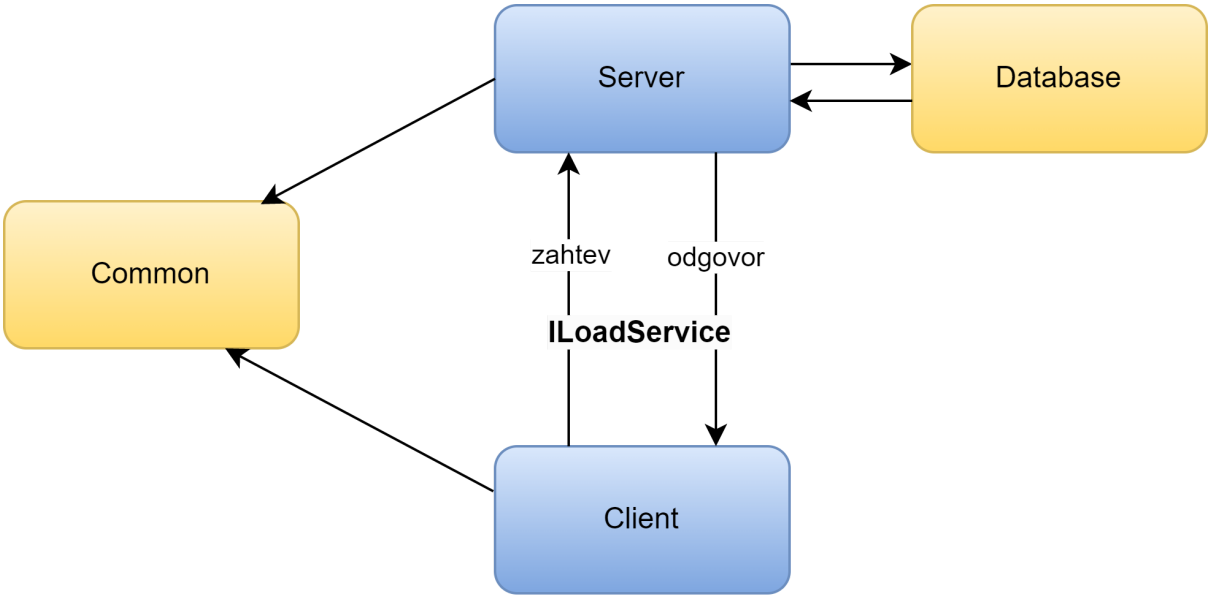
Load objekti dobijeni na osnovu upita prosleđuju se kao rezultat klijentskoj aplikaciji.
Podaci se brišu iz In-Memory baze podataka kada prođe definisano *DataTimeout* vreme. Ovo vreme se definiše kao odgovarajući broj minuta u App.config datoteci servisne aplikacije. Podrazumevani broj *DataTimeout* minuta je 15.

Kada je klijentska aplikacija primila rezultate sa Load objektima, na osnovu njih kreiraju se CSV datotekae kojae se upisuju upisuje na lokaciju definisanu u konfiguracionoj datoteci klijentske aplikacije App.config. Takođe, na konzoli klijentske aplikacije ispisuje se poruka o kreiranoj datoteci. Ova poruka sadrži i podatke o putanji i imenu datoteke.

2. Arhitektura projekta sa tokom podataka

Kao što je opisano u prethodnom poglavlju, aplikacija se sastoji iz servisa i klijentske aplikacije. Klijentska aplikacija putem WCF tehnologije šalje zahteve servisu, dok servis generiše odgovore za pristigle zahteve. Za WCF, komunikacija je definisana u okviru *contract*-a *ILoadService*.

Na slici 1 su prikazane komponente sistema. Plavom bojom su označene konzolne aplikacije, dok su žutom *class librery*. *Common* projekat referenciraju servis i klijentska aplikacija i u njemu se nalaze zajedničke klase i *ILoadService* interfejs. Projekat *Database* obezbeđuje podatke servisu tako što proveri da li ima keširanih podataka koje dostavlja ako postoje, u suprotnom podatke uzima iz XML baze.



Slika 1 - Arhitektura aplikacije

3. Opis interfejsa sa opisom osnovnih funkcionalnosti

Aplikacija se sastoji iz dve konzolne aplikacije, serverske i klijentske. Klijentska aplikacija sadrži korisnički interfejs uz pomoć kojeg korisnici mogu da šalju zahteve servisu. Korisnik ima mogućnost unosa datuma za koji želi da dobije CSV datoteku. Ukoliko unese datum koji nije u validnom formatu, aplikacija će zatražiti unos nove vrednosti. Za validno uneti datum se sa servera dobavljaju load i audit objekti. Audit objekti se ispisuju na konzoli dok se load objekti čuvaju na putanji definisanoj u konfiguraciji.

3.1. Konfiguracija

Klijentska konfiguracija sadrži sledeće paramere:

1. Service model konfiguraciju za WCF klijenta
2. *folderPath* - putanja do foldera u kojem se čuvaju CSV fajlovi

Serverska konfiguracija sadrži sledeće paramere

1. Service model konfiguraciju za WCF servisa
2. *DateTimeout* - broj minuta koliko traju keširani podaci
3. Putanje do XML fajlova: *loadsPath* i *auditPath*

3.2. Starovanje aplikacije

Neophodno je prvo startovati instancu servisa (Server projekat) nakon čega je moguće startovanje klijentske aplikacije (Client projekat).

4. Opis tehnologija koje su korišćene

Za izradu aplikacije korišćen je .NET Framework. NET Framework je softverski paket (ponekad se koristi pojmovi: tehnologija, platforma), koji je dizajniran za razvoj programa i aplikacija. Glavna karakteristika paketa je da će različite usluge i programi pisani na različitim programskim jezicima biti kompatibilni.

Za komunikaciju između aplikacija korišćena je tehnologija WCF. **Windows Communication Foundation** (skraćeno **WCF**), ranije nazivan Indigo, je servisno orijentisani model razmene poruka, koji omogućava programima da komuniciraju preko računarske mreže ili lokalno, na sličan način na koji se povezuju i veb servisi. WCF je alat koji u sebi uključuje set biblioteka razvijenih za distribuirano programiranje. Microsoft ga je uveo sa verzijom .NET Framework 3.0

5. Zaključak sa mogućim pravcima budućeg istraživanja i proširenjem zadatka

Kreirana aplikacija omogućava kreiranje CSV datoteka na efikasan način. Keširanjem podataka se dobilo ubrzanje serverske aplikacije, ali je i obezbeđeno da se podaci ne gomilaju u radnoj memoriji upotrebom pozadinskog Taska koji briše stare podatke.

Dalje unapređenje je moguće na polju boljeg korisničkog interfesa i podrške za bolje filtriranje podataka. Unapređenje bi sadržalo mogućnost izlistavanja svih dostupnih datumai sadržalo opciju unosa nekog drugog kriterijuma osim datuma poput zahteva za izvajanje Load objekata koji imaju prognoziranu grešku (apsolutna vrednost razlike između forecast i measured value) veću od definisane u korisničkom interfejsu.