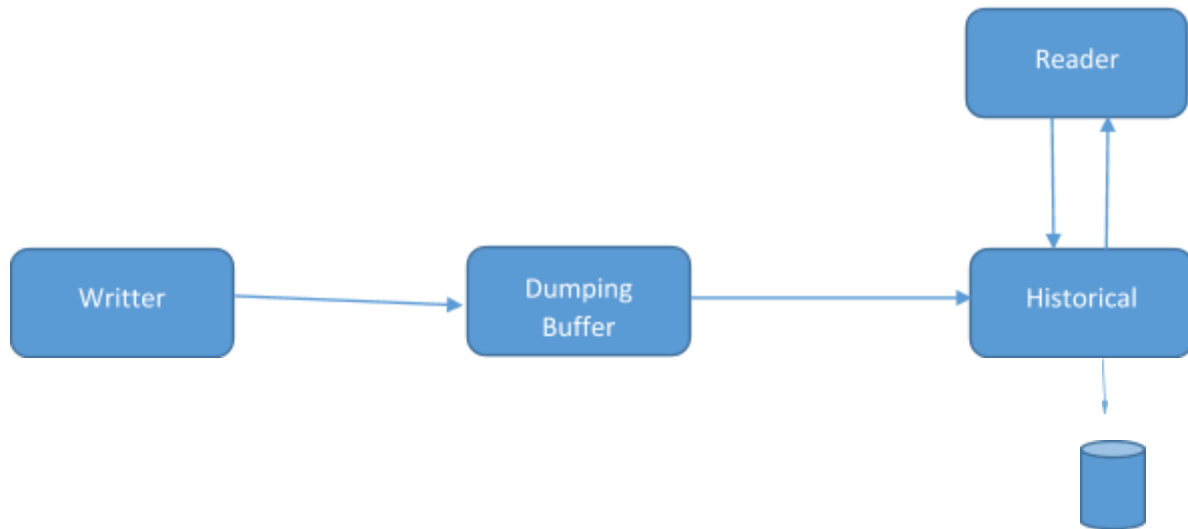


Projektni Zadatak 2

Potrebno je napraviti dizajn sistema, arhitekturu sistema, implementira i istestirati rešenje koji simulira rad i komunikaciju keš memorije sa perzistencijom podataka.

Sistem sadrži 4 komponente:

1. Dumping Buffer
2. Historical
3. Writer
4. Reader



Dumping Buffer

Dumping Buffer je komponenta koja služi za privremeno čuvanje podataka pre nego što ih prosledi Historical komponenti. Dumping Buffer prima podatke od komponente Writer, od koje dobija sve podatke.

Dumping Buffer čuva podatke u obliku kolekcije – CollectionDescription (CD).

CollectionDescription sadrži:

- ID
- Dataset
- DumpingPropertyCollection

DumpingPropertyCollection sadrži:

- Niz DumpingProperty-a

DumpingProperty sarži:

- Code
- DumpingValue

Prilikom slanja podataka Historical komponenti, Dumping Buffer kreira posebnu strukturu DeltaCD.

DeltaCD sadrži:

- Transaction ID
- CollectionDescription Add
- CollectionDescription Update
- CollectionDescription Remove

DeltaCD sadrži 3 CollectionDescription strukture, po jednu za operacije dodavanja, ažuriranja i brisanja iz skladišta.

Historical

Historical je komponenta koja služi za perzistenciju podataka dobijenih od Dumping Buffer komponente. Historical komponenta podatke dobijene od Dumping Buffer komponente snima u bazu podataka. Snimanje u bazu podataka fajl se vrši prema dataset-u koji je dobijen kao podatak od Dumping Buffer komponente. Za svaki dataset treba da postoji posebna tabela koji će čuvati izmene.

Prilikom perzistencije podataka, Historical komponenta radi sa svojim strukturama:

HistoricalProperty sadrži:

- Code
- HistoricalValue

Description sadrži:

- ID
- Lista HistoricalProperty-s
- Dataset

Lista Description-a (LD) sadrži:

- Listu Description-a

Historical komponenta radi sa LD strukturom i to skladišti.

Writer

Writer je komponenta koja služi za upisivanje novih podataka u Dumping Buffer. Writer komponenta prima nove informacije svake 2 sekunde i to prosledjuje Dumping Buffer -u. Writer ima i mogućnost

direktnog upisa u istoriju, preko Historical komponente. Writer komponenta nema svoju memorijsku strukturu sa kojom radi, već podatke direktno šalje Dumping Buffer -u. Podaci koji se šalju su:

- Code
- Value

Reader

Reader je komponenta koja služi za čitanje podataka iz .XML fajla. Reader komunicira sa Historical komponentom radi dobavljanja potrebnih podataka i ispisa tih podataka.

Scenario rada aplikacije

Writer komponenta prosleđuje podatke na 2 načina:

1. Podatke šalje na svake 2 sekundi Dumping Buffer komponenti (WriteToDumpingBuffer)
2. Podatke šalje direktno za skladištenje Historical komponenti (ManualWriteToHistory)

Dumping Buffer komponenta dobijene podatke prepakuje u svoju strukturu sa kojom rukuje, pri tome dodeljuje jedinstven ID. Dataset se dodeljuje prema Code-u podataka koji su primljeni.

Dumping Buffer komponenta u svojoj strukturi skladišti podatke u memoriji (CD struktura). Ukoliko od Writer komponente pristignu podaci koji imaju isti Code kao i oni što se nalaze u Dumping Buffer komponenti, njihova vrednost (Value) u Dumping Buffer komponenti treba da se ažurira na novu vrednost koja je stigla od Writer-a.

Kad Dumping Buffer u svojoj kolekciji nakupi 2 različite vrednosti u okviru istog dataset-a, tada su podaci spremni za slanje i pakuju se u DeltaCD komponentu, zatim se oslobađa CD struktura za prijem novih elemenata. Posle pristiglih 10 vrednosti iz Writer komponente DeltaCD se prosleđuje Historical komponenti (WriteToHistory). Tom prilikom ukoliko DeltaCD nema nijedan od Collection Description objekata (Add, Update ili Remove) slanje će se odložiti za još 10 prijema podataka od Writer komponente. Ukoliko pre slanja DeltaCD komponente, Dumping Buffer nakupi opet 2 različite vrednosti u okviru istog dataset-a, tada će ti podaci čekati, i ukoliko je potrebno ažurirati se sa Writer-a dok se ne pošalje DeltaCD komponenta i tada je moguće upisati nove elemente.

Za potrebe DeltaCD komponente potrebno je voditi računa koji elementi su novi i stižu za Add, koji za Update, a koji za Remove u Dumping Bufferu. Iz razloga što Writer ima mogućnost manualnog upisa u istoriju, Dumping Buffer u nekim slučajevima ne može da zna da li postoji element u istoriji i pri tome Historical treba da čuva informaciju da je podatak stigao preko manualnog upisa, i da će jedino validno biti da se ažurira preko CollectionDescription Add iz Dumping Buffera, dok nije validno da Dumping Buffer pošalje Update i Remove za taj element jer nema informaciju da li je upisan.

Historical komponenta prima sadržaj od Dumping Buffer komponente i prepakuje strukturu od Dumping Buffer komponente, DeltaCD, u svoju internu strukturu – LD, tom prilikom se mapira DumpingValue na HistoricalValue.

Historical komponenta treba da proveri da li su podaci validni – da li su dataset-ovi odgovarajući i u skladu sa Code-ovima koji su prosleđeni u okviru dataset-a.

Historical komponenta treba da proveri da li dobijeni podaci treba da budu upisani u .XML fajl. Uslov da se podatak upiše u .XML fajl je da izlazi iz Deadband-a.

Deadband predstavlja uslov da li je potrebno podatak upisati u bazu podataka.

Deadband iznosi 2% i značiće da ukoliko pristigli podatak, već postoji u bazi, ukoliko nova njegova vrednost je veća od 2% od stare vrednosti, tada će biti upisana nova vrednost. Ukoliko nova vrednost ne izlazi iz okvira od 2% od stare vrednosti tada nova vrednost ne treba da bude upisana u bazu podataka.

Jedini izuzetak iz Deadband-a je Code – CODE_DIGITAL, za ovaj Code se uvek upisuje prosleđena vrednost i ne proverava se Deadband.

Prilikom upisa podatka u bazu, Historical komponenta će generisati timestamp sa vremenom upisa tog podatka.

Reader komponenta treba da iščita vrednosti iz Historical komponenti po vremenskom intervalu za traženi Code (GetChangesForInterval).

Implementirati Logger, koji će beležiti sve aktivnosti koje se dešavaju u ostalim komponentama.

Lista Code-ova:

1. CODE_ANALOG
2. CODE_DIGITAL
3. CODE_CUSTOM
4. CODE_LIMITSET
5. CODE_SINGLENODE
6. CODE_MULTIPLENODE
7. CODE_CONSUMER
8. CODE_SOURCE
9. CODE_MOTION
10. CODE_SENSOR

Spisak DataSet-ova po Code-ovima:

- DataSet = 1 – CODE_ANALOG, CODE_DIGITAL
- DataSet = 2 – CODE_CUSTOM, CODE_LIMITSET
- DataSet = 3 – CODE_SINGLENODE, CODE_MULTIPLENODE
- DataSet = 4 – CODE_CONSUMER, CODE_SOURCE
- DataSet = 5 – CODE_MOTION, CODE_SENSOR

*Value ima strukturu:

- Timestamp (datum plus vreme)
- ID geografskog područja
- Potrošnja u mW/h

Kriterijum ocenjivanja

1. Dizajn I arhitektura rešenja
2. Korišćenje Scrum metodologije razvoja – definisanje User Story-a i taskova, planiranje i estimacija
3. Implementacija rešenja
4. CI ciklus
 - a. Build
 - b. UnitTestovi
 - c. Pokrivenost koda testovima