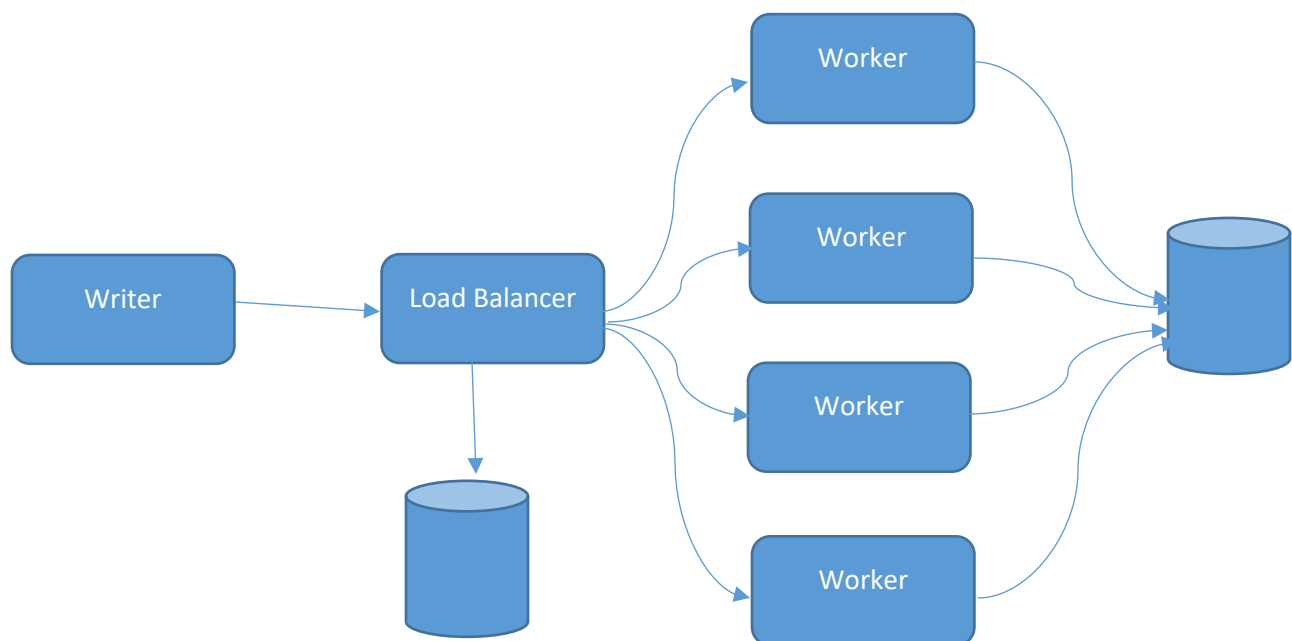


Projektni Zadatak 3

Potrebno je napraviti dizajn sistema, arhitekturu sistema, implementirati i istestirati rešenje koji simulira rad i komunikaciju Load Balancer modula.

Sistem sadrži 4 komponente:

1. Load Balancer
2. Worker
3. Writer
4. Reader



Load Balancer

Load Balancer je komponenta koja služi za ravnomerno raspoređivanje posla. Posao raspoređuje Workerima. Load Balancer prima podatke od Writer komponente i prosleđuje ih nekom od slobodnih Workera na obradu. Load Balancer privremeno smešta kod sebe pre slanja.

LB salje podatke putem Description strukture:

Item sadrži:

- Code
- Value

Description sadrži:

- ID
- Lista Item -a

- Dataset

Lista Description-a (LD) sadrži:

- Listu Description-a

LB vrši perzistenciju podataka po Dataset-u.

Worker

Worker je komponenta koja služi za obradu podataka dobijenih od Load Balancer komponente i perzistenciju tih podataka. Worker komponenta podatke dobijene od Load Balancer komponente snima u bazu. Snimanje u bazu se vrši prema dataset-u. Za svaki dataset treba da postoji posebna tabela/fajl koji će čuvati izmene.

Worker prima podatke u LD strukturi koju treba da prepakuje u svoju strukturu za rad.

Prilikom obrade i perzistencije podataka Worker komponenta radi sa svojom interneom strukturom – CollectionDescription (CD):

CollectionDescription sadrži:

- ID
- Dataset
- HistoricalCollection

HistoricalCollection sadrži:

- Niz WorkerProperty -a

WorkerProperty sadrži:

- Code
- WorkerValue

Writer

Writer je komponenta koja služi za upisivanje novih podataka u Load Balancer komponentu. Writer komponenta šalje nove podatke svake 2 sekunde Load Balancer komponenti. Writer takođe ima opciju paljenja i gašenja Workera sa kojima Load Balancer radi. Paljenje i gašenje Workera se radi ručno. Podaci koje Writer šalje su:

- Code
- Value

Reader

Reader je komponenta koja služi za čitanje podataka iz baze. Reader komunicira sa Workerima radi dobavljanja potrebnih podataka i ispisa tih podataka.

Scenario rada aplikacije

Writer komponenta radi sledeće:

1. Šalje podatke na svake 2 sekunde Load Balancer komponenti
2. Inicira paljenje novih Workera
3. Inicira gašenje postojećih Workera

Load Balancer prima podatke od Writera i smešta ih kod sebe u bafer. Podaci se iz bafera skidaju i šalju Workerima na obradu. Load Balancer vodi računa o ravnomernoj raspodeli posla među Workerima koji su upaljeni. Raspoređivanje posla treba da bude po Round-Robin raspoređivanju. Load Balancer treba da vodi evidenciju koji od Workera je upaljen, a koji ugašen. Takođe treba da primi zahtev od Writera da upali novi Worker ili ugasi postojeći.

Worker komponenta prima sadržaj od Load Balancer komponente i pakuje taj sadržaj u svoju internu strukturu – CD. Worker komponenta treba da vodi računa o dataset-u prilikom obrade, da se podaci pakuju u određeni dataset.

Worker inicira upis u bazu samo kad je dataset popunjen, da su obe vrednosti u strukturi pre snimanja u bazu.

Worker komponenta treba da proveriti da li dobijeni podaci treba da budu upisani u bazu. Uslov da se podatak upiše u bazu je da izlazi iz Deadband-a.

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

Deadband iznosi 2% i to znači da ukoliko pristigli podatak, već postoji u bazi, ukoliko je njegova nova vrednost različita više 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.

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, Worker komponenta će generisati timestamp sa vremenom upisa tog podatka, i timestamp takođe treba da bude upisan u bazu zajedno sa podatkom koji se upisuje.

Reader komponenta komunicira sa Workerom radi dobavljanja poslednjih vrednosti izabranih Code-ova.

Reader komponenta treba da iščita istorijske vrednosti iz nekog od Worker komponenti po vremenskom intervalu za traženi Code.

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

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

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