

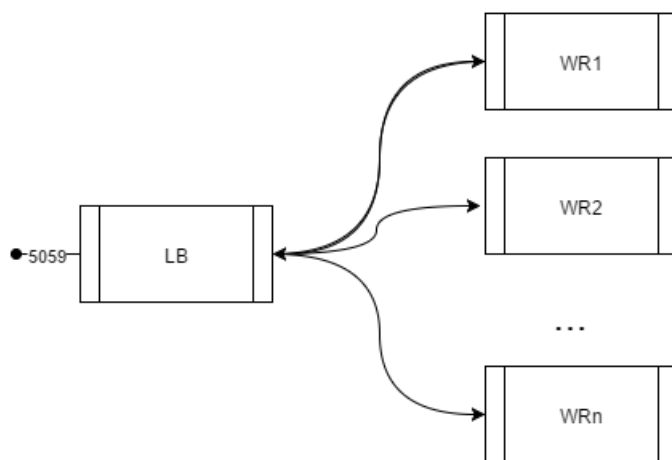
Load Balancing

Razviti servis koji vrši obradu merenja pametnih brojila. Servis se sastoji od dva tipa procesa - Load Balancer (LB) i Worker (WR). Postoji jedna instanca LB komponente koja sluša na portu 5059 i prima zahteve za računanje. Prihvaćen zahtev se prosleđuje prvom slobodnom WR. WR nakon obrade zadatka obaveštava LB da je završio sa obradom.

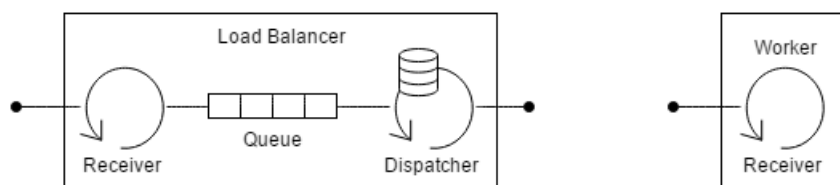
Potrebno je izmeriti propusnost sistema sa 1-10 WR uz praćenja resursa LB mašine.

LB dobija informaciju o broju WR kojima raspolaže tako što se oni prilikom startovanja registruju LB odnosno prilikom gašenja odjave.

Arhitektura servisa



Predloženi dizajn servisa



Publisher – Subscriber

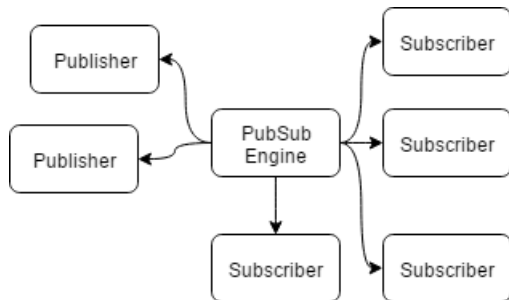
Potrebno je napraviti PubSub servis koji može da opslužuje proizvoljan broj klijenata. Servis treba da implementira sledeći interface:

1. `void Connect();`
2. `void Subscribe(void *topic);`
3. `void Publish(void * topic, void* message)`

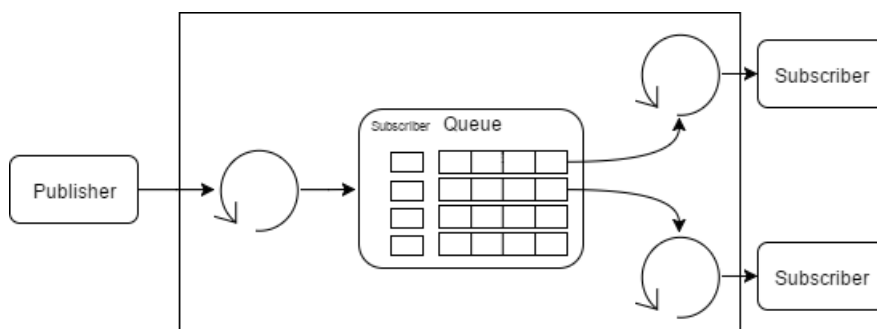
4. `bool ExistTopic(void *topic)`
5. `int numberOfSubscribers(void *topic)`

Topic treba da bude proizvoljan niz karaktera.

Arhitektura servisa



Predloženi dizajn servisa



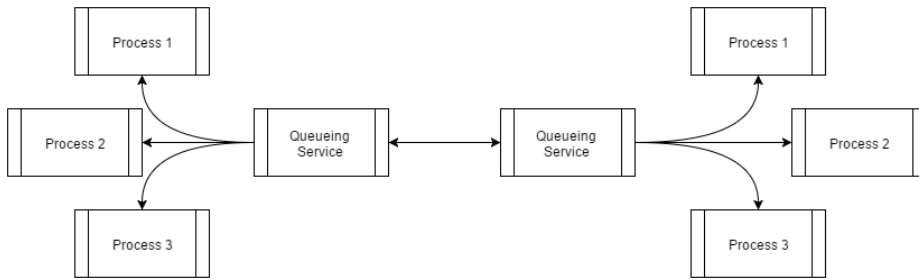
Message Queueing Service

Razviti servis za razmenu podataka preko redova za poruke (message queueing service). Servis treba da omogući razmenu poruka između dva računara. Prilikom startovanja servisa, na oba strane se zadaju imena izvornih redova. Kada se uspostavi veza između servisa, obe strane će razmeniti imena napravljenih redova. Servis implementira sledeći interface:

- `SendMessage(char* queueName, void* message, int messageSize)`

Servis treba da omogući pouzdan transport poruka bez perzistencije. Poruka će biti sklonjena sa reda tek kada je isporučena odredištu.

Arhitektura servisa



Replication

Potrebno je implementirati servis za replikaciju podataka. Servis implementira interface:

- RegisterService(int ServiceID)
- SendData(int ServiceID, void* data, int dataSize)
- Callback: void ReceiveData(void* data, int dataSize)

Servis treba da implementira asinhronu i sinhronu replikaciju. Podaci koje servis dobija na slanje treba da smešta na interni kružni red. Razmisliti kako je moguće implementirati dinamički red.

Arhitektura servisa

