INDUSTRIJSKI KOMUNIKACIONI PROTOKOLI

DOKUMENTACIJA PROGRAMA LOAD BALANCER

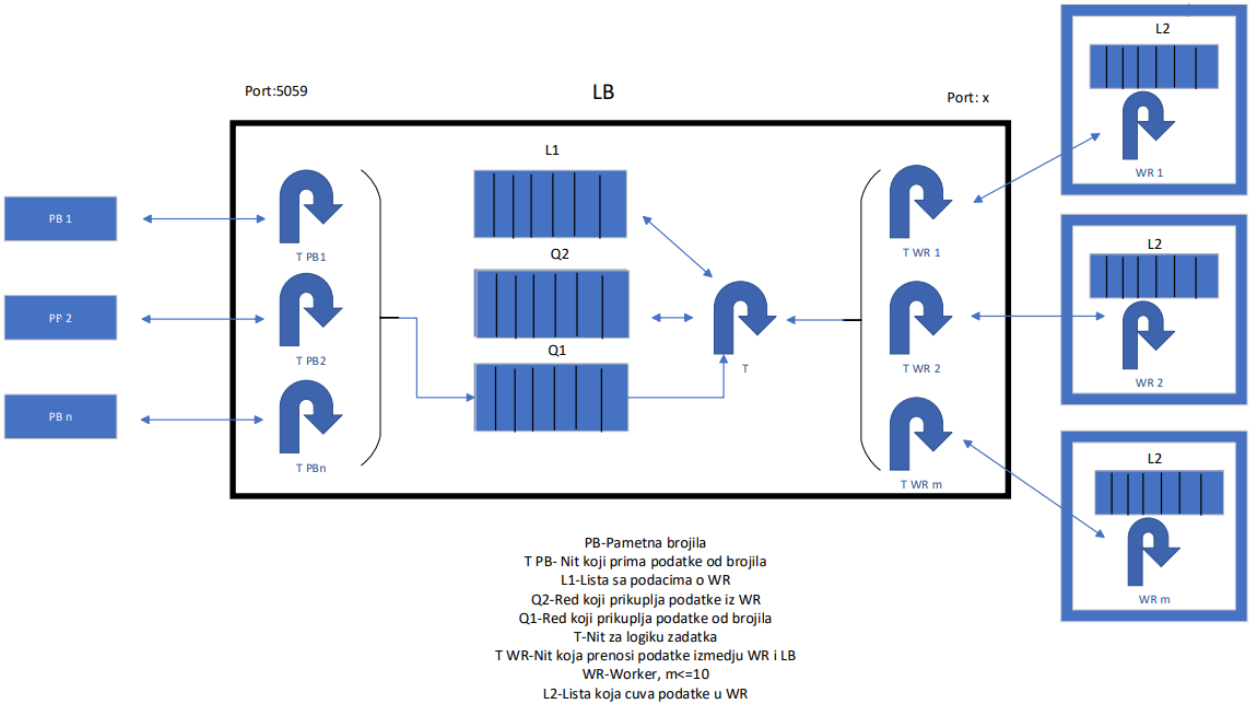
PR87/2020 Nevena Ćulibrk

PR97/2020 Milorad Maksić

1. UVOD

Potrebno je 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.

2. DIZAJN



PB-Pametna brojila
T PB- Nit koji prima podatke od brojila
L1-Lista sa podacima o WR
Q2-Red koji prikuplja podatke iz WR
Q1-Red koji prikuplja podatke od brojila
T-Nit za logiku zadatka
T WR-Nit koja prenosi podatke izmedju WR i LB
WR-Worker, m<=10
L2-Lista koja cuva podatke u WR

3. REŠENJE

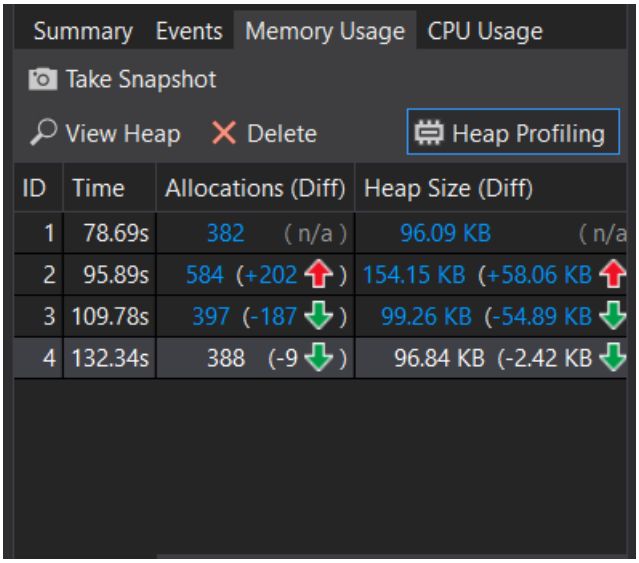
Aplikacije funkcioniše tako što se pametna brojila prijavljuju na servis preko porta 5059 i šalju podatke fiksne dužine. Workeri se prijavljuju na LoadBalancer preko porta 4000 i tom prilikom ostavljaju svoje podatke (kritične sekcije, accepted socket, broj primljenih podataka) koji se čuvaju u listi L1. Pristigle podatke od strane pametnih brojila LoadBalancer smešta u red “Q1”. Nit “T” proverava da li u privremenoj memoriji (Lista L2) ima nekih podataka, u slučaju da nema izvršava logiku distribuiranja na osnovu podataka iz liste L2, uzima podatke iz reda i preko niti “TW” šalje Workeru koji je najmanje opterećen podacima. Prilikom odgovora Workera podaci o stanju skladišta u listi L1 se ažuriraju. Kada se prijavi novi Worker vrši se balansiranje skladišta svih Workera tako što na osnovu logike sistema svi Workeri pošalju odredjeni broj podataka koji se smešta u privremenoj memoriji, Lista L2, i zatim se ti podaci pošalju novom Workeru. Isti princip se izvodi ukoliko se neki Worker odjavi sa LoadBalancera.

4. STRUKTURE PODATAKA

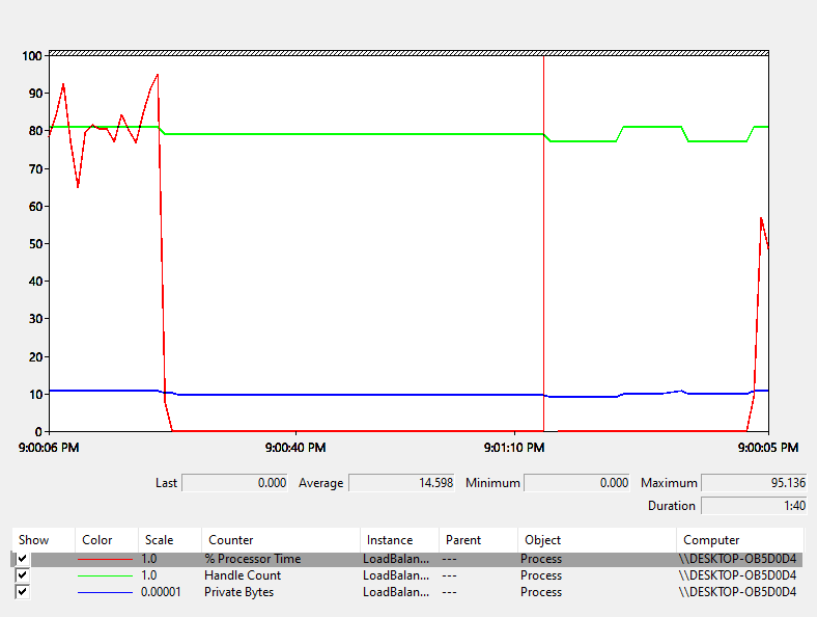
Redovi (Q1, Q2) se koriste za skladištenje privremenih podataka jer je to struktura koja omogućuje lako i brzo ubacivanje i izbacivanje podataka, kao i zato što predstavlja red čekanja za podatke. Lista sa head i tail pokazivačima se koristi na mestima gde nije bitna brzina smeštanja podataka već sama organizacija istih. Listu koristimo kako bi lakše podatke prebacivali sa kraja na početak i obrnuto, što nam je ključno za distribuciju podataka.

5. TESTIRANJE

U okviru testiranja projekta odradjen je Stress Test koji je implementiran tako da brojilo u sekundi šalje 200 poruka i opterećuje redove za skladištenje podataka. Na priloženim slikama je prikazano stanje heap-a prilikom testiranja projekta kao i korišćenje procerskog vremena, handle-ova i memorije računara.



Snapshot 1. – Pokrenut program.
Snapshot 2. – Početak stress testa, poslato 200 poruka sa brojila.
Snapshot 3. – Pokrenut worker, odradjena distribucija.
Snapshot 4. – Kraj programa.



6. ZAKLJUČAK

Rezultati testova su skoro očekivani. Na početku programa je zauzeto više memorije nego što je očekivano, sam rad programa ima dobru kontrolu memorije, pravovremeno oslobadja zauzete resurse. Na kraju programa ostane zauzeto više memorije nego što je to bio slučaj na samom početku programa, a to je zbog handle-ova koji kada se zatvore ne briše se njihova memorija.