

1.Uvod

Potrebno je napraviti Publisher servis i Subscriber servis koji mogu da opslužuju proizvoljan broj klijenata. Servisi treba medjusobno da komuniciraju tako sto Publisher servis samo prima poruke od Publisher-a i zatim pošalje poruke Subscriber servisu. Nakon prijema Subscriber servis salje poruke primljene od Publisher servisa svi trenutnim Subscriber-ima na odrednjenu temu.

2.Dizajn resenja

Radi se o klijent-server arhitekturi. Postoje dva server, Publisher server koji publisher-ima (njegovim klijentima) pruza servis preko kog mogu da objave poruku na odredjenu temu kao i Subscriber server koji subscriber-ima (njegovim klijentima) pruza servise preko kojih mogu da se pretplate na poruke na odredjene teme.

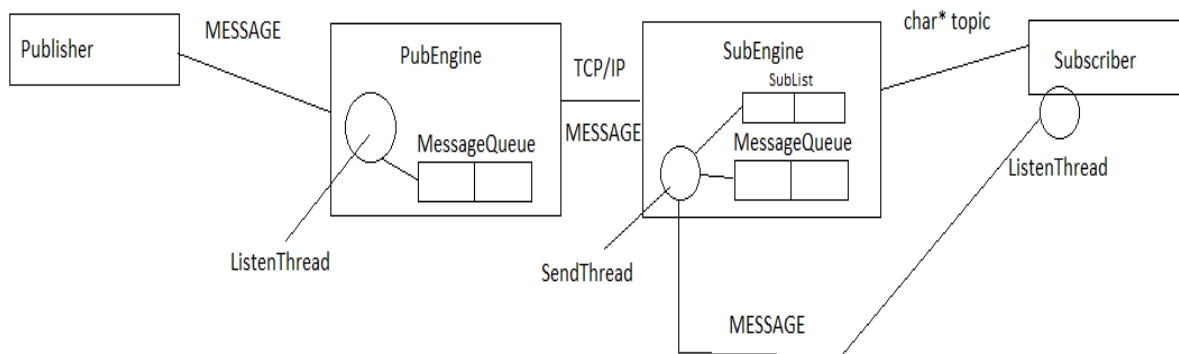
-Publisher servis ima na sebi dve niti, jedna koja prima klijente i slusa na poruke na njim(multipleksiranje se koristi kako bi se procesor rasteretio) i po prijemu je smesta na red i obavestava drugu nit da moze da pocne sa radom. Druga nit kada primi signal (preko semafora) skine sa reda poruku i prosledi je Sub servisu.

-Subscriber servis takodje ima dve niti, prva slusa na poruke od Pub servisa preko soketa koji je u blokirajucem modu kako nit ne bi radila u prazno. Kada primi poruku ona je stavlja u red i obavestava drugu nit da moze da krene sa radom. Druga nit skida sa reda poruku, proverava da li ima subscriber-a temu na koju je objava pisana i prosledi je njima.

-Subscriber Client isto ima dve niti. Jedna nit slusa na odredjenom portu za objave na temu na koju je pretplacena dok druga komunicira sa Sub servisom kom. Salje mu informacije na kom portu je otvorena prva nit kako bi on mogao tu da salje objave i preko nje se pretplacuje na servis.

Za komunikaciju izmedju klijenata i servisa kao i izmadju servisa medjusobno koristi se iskljucivo TCP protokol.

Design



3. Struktura podataka

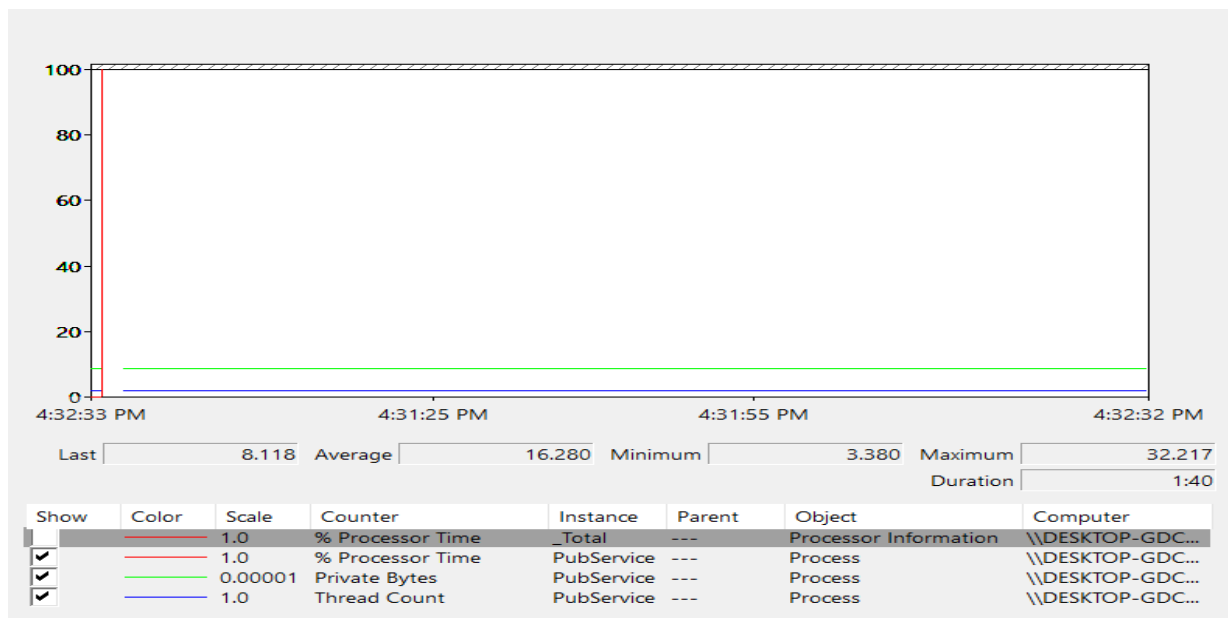
Struktura "MESSAGE" se koristi kako bi predstavili jednu objavu subscribera u sistemu. Ona se sadrzi od polja "char* topic" koje predstavlja temu na koju se objava pise i polje "char* message" u kojem se sadrzi preuzeta poruka od publisher-a.

Struktura "SUB_LIST" se koristi kako bi u njoj cuvali koji subscriber-i su na koji topic pretplaceni. Ona sadrzi u sebi polje "int numSubs" koje oznacava broj subscriber-a na polje "char* topic" koje predstavlja temu na koju je on pretplacen. Trece polje ove strukture "short subPorts[MAX_CLIENTS]" je niz portova preko kojih se vrši slanje na klijente nakon sto pristigne poruka na datu temu. Takodje postoje polja "SUB_LIST* next" koji pokazuje na sledeci clan liste i polje "CRITICAL_SECTION cs" koje se koristi kako bi se obezbedila konzistencija podataka.

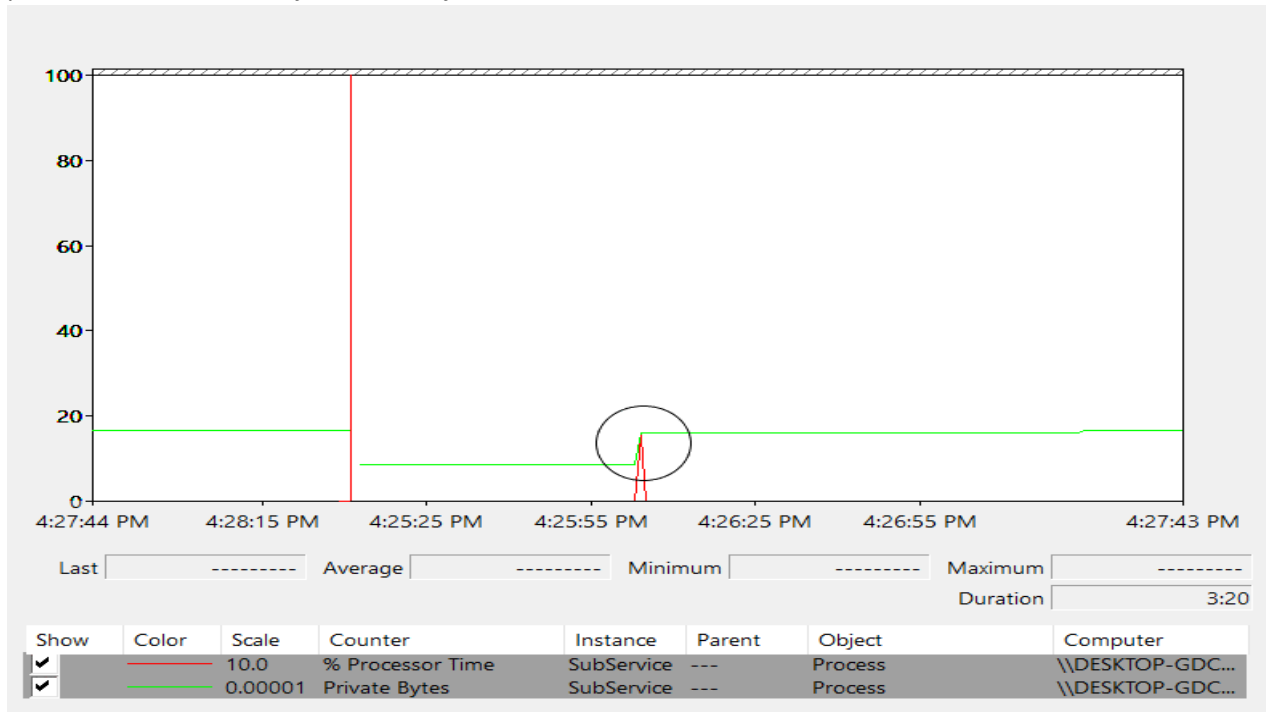
Pored ove lista koriste se i redovi podataka tipa "MESSAGE" u kome se cuvaju poruke koje cekaju obradu kako bi moglo vise zahteva da se primi od klijenata i kako bi sinhronizacija bila olaksana.

4. Rezultati testiranja

-Publisher servis je testiran sa jednim, maksimalnim brojem kao i vecim brojem klijenata od dozvoljenog pri cemu je sve radilo optimalno i bez ikakvih nepredvidjenih dogadjaja. Takodje je testirano slanje poruka sa istih kao i opterecenje queue, i ove funkcionalnosti su radile kako je predvidjeno.



-Subscriber servis je testiran na isti nacin kao I Subscriber servis pri cemu je primecena samo jedna greska koja I dalje nije resena. Na vecem broju klijenata proces pocne da se ponasa nepredvidjeno, skoci procesorsko vreme I broj zauzetih bajtova.



-SubClient aplikacija je takodje testirana jer ima dodatne listen threadove I ona takodje radi kako treba.

5.Zakljucak

Publisher servis radi optimalno, ne nigde ne trosi dodatno procesorsko vreme ili memoriju za obradu novih klijenata . Subscriber servis nije prosao sve testove I postoje optimalnija resenja.

6.Unapredjenja

Na subscriber servisu umesto reda objava bi bilo bolje da se koristi "thread pool" I za svaku pristiglu objavu da se uzme nit iz pool-a koja ce obraditi tj poslati tu objavu svim sub-ovima. Ubacivanje novih sub-va na temu(jedino sto se u listi menja pored brojaca subova) je vec zasticen kritcnim sekcijama tako da je konzistensnost podataka vec obezbedjena. Ovo bi ubrzalo proces slanja I prividno ga cak paralelizovalo.