Dokumentacija projektnog zadatka

PS - G12 - Publisher - Subscriber

Studenti: Nemanja Malinović, PR108/2019

Jovana Lažetić, PR111/2019

Link do github repozitorijuma: *https://github.com/nemanjaa21/IKP_PUBSUB_G12*

Uvod

Projektni zadatak je baziran na implementaciji ***Publisher*** – ***Subscriber*** servisa koji može opsluživati proizvoljan broj klijenata. Klijenti mogu biti ***Publisheri*** ili Subscriberi. ***Subscriber*** može da se poveže na servis - ***PubSubEngine*** i da izabere temu - topic na koju će biti pretplaćen. Nakon toga treba da prima poruke za datu temu. ***Publisher*** treba da ima mogućnost povezivanja sa servisom. Takođe treba da izabere temu na koju će slati poruke i da pošalje samu poruku.

Dizajn

Sistem se sastoji od tri komponente. Serverska komponenta je ***PubSubEngine*** i ona je zadužena za komunikaciju sa klijentima. Klijentske komponente su ***Publisher*** i ***Subscriber. Publisher*** je korisnik koji poseduje metodu *SendMessage* (tj. publish metodu) kojom šalje poruke na izabranu temu. ***Subscriber*** je korisnik koji ima metodu *ReceiveMessage* (tj. subscribe metodu) kojom se pretplaćuje na izabranu temu i dobija poruke koje ***Publisher*** objavljuje na tu temu.

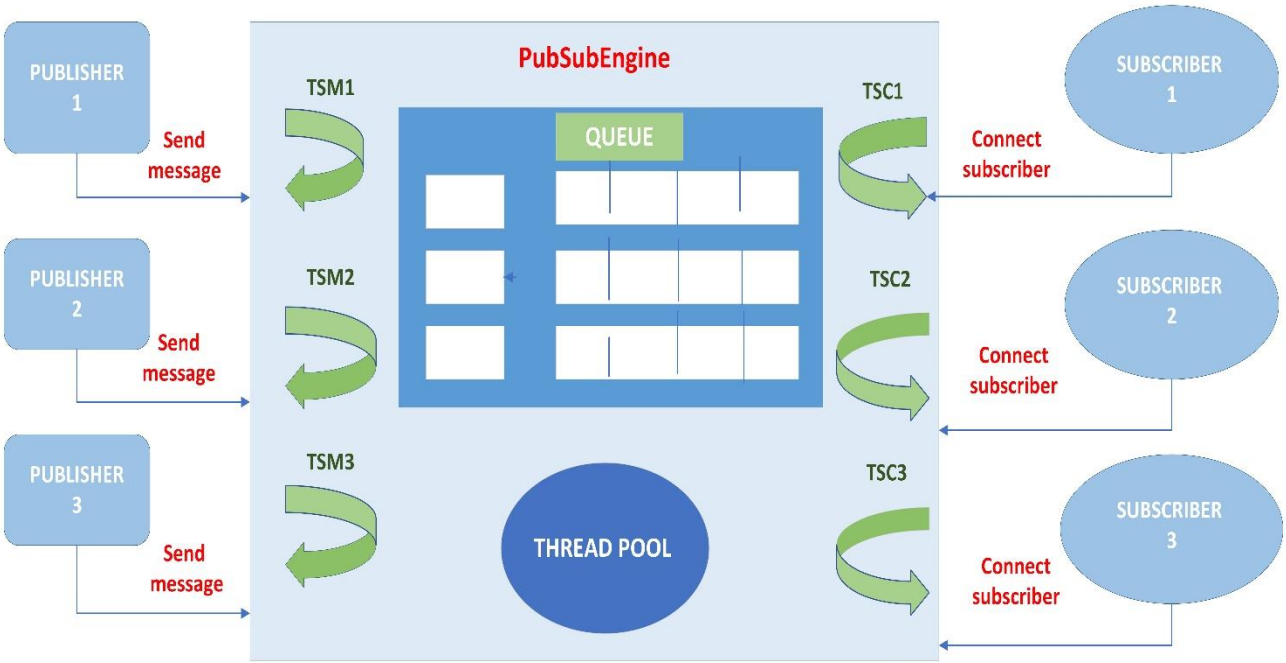
Tip komunikacije između komponenti je neblokirajući TCP. On garantuje isporuku poruka i bolju iskorišćenost performansi zbog neblokirajućeg režima. Takođe servis podržava povezivanje sa više klijenata i obradu njihovih zahteva. To je omogućeno uz pomoć paralelnog rada više niti - thread na samom servisu, pa tako imamo sledeće niti:

- Nit koja sluša konekcije ***Publishera***

- Nit koja prima poruke i upisuje ih u ***Queue***.
- Nit koja sluša konekciju ***Subscribera***.

Takođe koristimo ***Thread pool*** koji se sastoji od više niti koje imaju mogućnost prosleđivanja poruka. Niti su prethodno kreirane, pa ne gubimo vreme za kreiranje i uništavanje svake pojedinačne niti u toku izvršavanja programa.

Dijagram komponenti:



Slika 1. *Dizajn projekta*

TSM → Thread Send Message

TSC → Thread Subscriber Connect

Zakljucak:

U zadatku je bilo potrebno omogućiti opsluživanje većeg broja klijenata u cilju razmene poruka. To smo postigli upotrebom većeg broja niti, kao i struktura podataka kao što je npr. *Queue*.

Moguća unapredjenja:

“Usko grlo” programa je uočeno prilikom obrade velikog broja poruka na servisu. Potencijalno, ali ne i najbolje rešenje je proširenje prijemnog buffera. Njegovim proširenjem , moguće je skladištiti znatno veći broj poruka, ali brzina obrade ostaje nepromenjena, te postoji mogućnost da se u izvesnom trenutku buffer opet prepuni, ukoliko servis primi jako veliki broj poruka u kratkom vremenskom intervalu. Zbog toga je najbolje rešenje da se pored povećanja buffera poveća i brzina obrade podataka. To se može postići dodavanjem Thread pool-a, čiji će thread-ovi parsirati pristigle poruke i prosleđivati zahteve za slanje poruka subscriber-ima. U tom slučaju, jedan thread bi bio zadužen samo za prijem poruka, dok bi Thread pool bio zadužen za njihovu obradu.