

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1954

**Stvarnovremensko praćenje  
parametara ispravnosti rada u  
sustavu za raspodijeljenu obradu  
tokova podataka**

Mislav Jakšić

Zagreb, svibanj 2019.

*Umjesto ove stranice umetnite izvornik Vašeg rada.  
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

*Hvala svima!*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Sustavi za raspodijeljenu obradu tokova podataka</b>	<b>2</b>
2.1. Model objavi/pretplati . . . . .	3
2.2. Apache Kafka . . . . .	3
2.2.1. Objavljiivač . . . . .	4
2.2.2. Tema . . . . .	4
2.2.3. Posrednik . . . . .	4
2.2.4. Pretinac . . . . .	6
2.2.5. Pretplatnik . . . . .	8
2.3. Apache Pulsar . . . . .	9
2.3.1. Objavljiivač . . . . .	10
2.3.2. Tema i pretplata . . . . .	10
2.3.3. Posrednik . . . . .	12
2.3.4. Upravljana knjiga . . . . .	13
2.3.5. Pretplatnik . . . . .	13
2.4. RabbitMQ . . . . .	13
<b>3. Postojeca rjesenja</b>	<b>14</b>
<b>4. Arhitektura rjesenja</b>	<b>15</b>
<b>5. Dizajn rjesenja</b>	<b>16</b>
<b>6. Izvedba rjesenja</b>	<b>17</b>
<b>7. Rezultati</b>	<b>18</b>
<b>8. Zaključak</b>	<b>19</b>



# 1. Uvod

Raspodijeljeni sustavi su nepouzdana. Pogreška u sklopovlju, operacijskom sustavu, programu ili mreži može izazvati ispad bilo kojeg dijela sustava. Ispadi u raspodijeljenom sustavu mogu pokrenuti lanac ispada. Ako ispad ne uzrokuje lanac ispada sustav će i dalje patiti jer će program i dalje zahtijevati računalno vrijeme i memoriju, a s njima neće obavljati koristan posao. U najgorem slučaju ispad može izazvati potpuno zatajenje sustava gdje je jedini lijek iznova pokrenuti sve njegove dijelove. U najboljem slučaju ispad će samo smanjiti učinkovitost sustava. Bez pažljivog nadzora raspodijeljenog sustava teško je otkriti ispad, a još teže otkloniti izvor ispada.

Zadatak nadzora je otkriti ispad i njegov uzrok. [13] ispade dijeli na ispad procesa, pogreške u komunikaciji, vremenske pogreške, pogrešan odgovor i bizantske pogreške. Ako se ispad želi otkriti potrebno je pratiti vrijednosti koje ukazuju da se ispad dogodio. Korisne vrijednosti mogu biti zauzeće memorije, brzina obrade zahtjeva, sadržaj poruke ili duljina uspostave komunikacijskog kanala. Nadzirani program vrijednosti predaje nadzorniku koji je čovjek ili program. Ako nadzor obavlja program on treba biti pouzdaniji od nadziranog programa inače je problem nadzora udvostručen, a ne riješen.

Prvi korak u izradi pouzdanog sustava nadzora je izrada pouzdanog sakupljača vrijednosti. Prije same izrade istražene su ideje, sukobljene su arhitekture i uspoređena postojeća rješenja. Kako bi naglasak na idejama, arhitekturama i izradi rješenja bio podjednak napravljen je sakupljač za jedan raspodijeljeni sustav. Kada bi umjesto njega bio napravljen svestran sakupljač koji može sakupljati vrijednosti raznolikog sklopovlja, operacijskih sustava i programa misli o arhitekturi bile bi izražene na uštrp onima o izradi programa. Nadzirani sustav je Apache Kafka, popularni sustav za raspodijeljenu obradu tokova podataka.

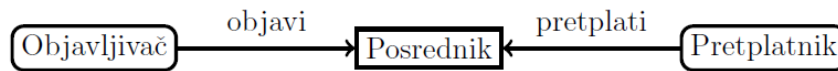
## 2. Sustavi za raspodijeljenu obradu tokova podataka

[4] razlikuje tok podataka od skupa podataka. Toku podaci dolaze stalno, u nepoznatom redoslijedu, bez znaka kada će prestat i bit će odbačeni nakon što su pročitani. Ovi sustavi su raspodijeljeni jer se tok mora obraditi brzo. Primjer problema toka podatka su kartično plaćanje, praćenje korisnika mrežnih stranica, posluživanje reklama i preporuka. Svi navedeni problemi imaju mnogo izvora i ponora podataka. Kako sakupiti, zabilježiti, te dostaviti podatke na ponor gdje će se obraditi su problemi za koje sustavi za raspodijeljenu obradu tokova podataka nude rješenje. Zato što su tokovi podatka raznovrsni razvijeno je mnoštvo alata za njihovu obradu. Alate razlikujemo po načinu sakupljanja podataka, po mogućnostima toka, po arhitekturi i načinu zapisivanja podataka.

Prije razvoja sakupljača vrijednosti potrebno je usporediti i izabrati alat za raspodijeljenu obradu podataka. Kako bi usporedba alata i izvedba sakupljača bila jednostavnija u obzir su uzeti samo javno dostupni alati. Većina takvih alata koristi model objavi/pretplati. Umjesto da se usporede svi javno dostupni alati izabran je predstavnik iz svakog važnog skupa alata:

- Apache Kafka je predstavnik popularnih alata za raspodijeljenu obradu tokova podataka
- Apache Pulsar je predstavnik modernih alata. [5] je 2016. objavio Pulsar dok je [1] objavio Kafku 2011. godine
- RabbitMQ je predstavnik alat koji potiskuju podatke prema korisnicima. Za razliku od RabbitMQ korisnici Kafke i Pulsara moraju povlačiti podatke

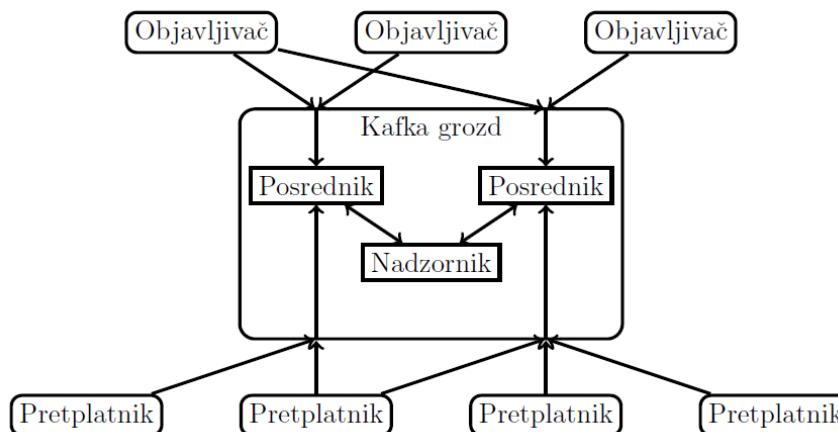
## 2.1. Model objavi/pretplati



Slika 2.1: Model objavi/pretplati

Apache Kafka, Apache Pulsar i RabbitMQ koriste model objavi/pretplati za razmjenu podataka između procesa. 2.1 prikazuje nužne dijelove modela objavi/pretplati. [13] navodi da se model objavi/pretplati sastoji od objavljiivača koji šalju poruke, pretplatnika koji čitaju poruke i posrednika koji razmjenjuje poruke između njih. Posrednik zapisuje objavljene poruke i dopušta pretplatnicima da ih čitaju. Prednost modela objavi/pretplati nad izravnim razgovorom među procesima je što objavljiivači i pretplatnici ne trebaju znati jedan za drugoga.

## 2.2. Apache Kafka



Slika 2.2: Apache Kafka grozd

Apache Kafka je popularni alat za raspodijeljenu obradu tokova podataka. 2.2 prikazuje Kafka grozd. Svaki grozd sastoji se od ZooKeeper nadzornika i barem jednog posrednika. Nadzornik usklađuje posrednike. Objavljiivači šalju poruke u temu grozda. Posrednici zapisuju poruke u pretinac teme. Pretplatnici čitaju poruke iz teme grozda i obrađuju podatke. [1] [2] opisuju primjenu, arhitekturu i izvedbu Kafka pretplatnika, pretinca, posrednika, teme i objavljiivača.



### **2.2.1. Objavljivač**

Objavljivač je korisnički program koji šalje poruke u temu grozda. Poruka se nužno sastoji od sadržaja, teme na koju se objavljuje, oznake pretinca, odmaka u pretincu i vremena objave. Poruke se šalju u skupini nakon sto prođe određeno vrijeme ili se nakupi dovoljno poruka. [11] navodi kako poruke sažeti prije slanja. Tema je podijeljena na pretince. Pretinac je dnevnik u koji posrednik zapisuje poruke. Posrednik je poslužitelj koji zapisuje poruke u pretinac. Objavljivač ili ključem izabere pretinac u koji želi zapisati poruke ili se poruke šalju u svaki pretinac jednoliko. Posrednik će poslati potvrdu kada se poruke zapišu u pretinac vođe i izabrani broj usklađenih sljedbenika. Prije slanja sljedećeg skupa poruka objavljivač može pričekati potvrdu posrednika.

Ako se dogodi ispad objavljivača ili posrednika, objavljivač će ponovno poslati poruke. Objavljene poruke biti će dostavljene posredniku barem jednom. Ako je uvišestručavanje poruka nedopustivo, objavljivač može koristiti transakcijski način rada.

### **2.2.2. Tema**

Tema je tok poruka, apstraktna umotvorina u koju objavljivači šalju poruke, a iz koje pretplatnici čitaju poruke. Posrednici se brinu da poruke budu zapisane. Biti pretplaćen na temu znači napraviti podtok poruka. Poruke objavljene u temu se ravnomjerno raspoređuju u podtokove. Svaki podtok ima pokazivač kojim pretplatnik čita poruke. Ako pokazivač dođe do kraja podtoka, pretplatnik se blokira dok se ne objavi nova poruka. Količina nepročitanih poruka u temi ne utječe na protok poruka kroz temu. Poruke u temi se brišu tek nakon određenog vremena. Tema je podijeljena na pretince s kojih posrednici čitaju i u koje pišu poruke. Kako bi poruke u pretincu bile dostupne nakon ispad posrednika, pretinac je moguće umnožiti.

### **2.2.3. Posrednik**

Posrednik je poslužiteljski program koji zapisuje poruke objavljivača i poslužuje poruke pretplatnicima. Svaki posrednik dio je samo jednog Kafka grozda. Posrednici koriste Apache ZooKeeper za izvršavanje sporazumnog algoritma, pamćenje tema, njihovih pretinaca i usklađenih posrednika. Posrednik će poruke zapisati u pretinac onim redoslijedom kojim je objavljivač poslao poruke, ne redoslijedom kojim je posrednik primio poruke. Pretplatnici će s posrednika čitati poruke onim redoslijedom kojim su zapisane u pretinac.

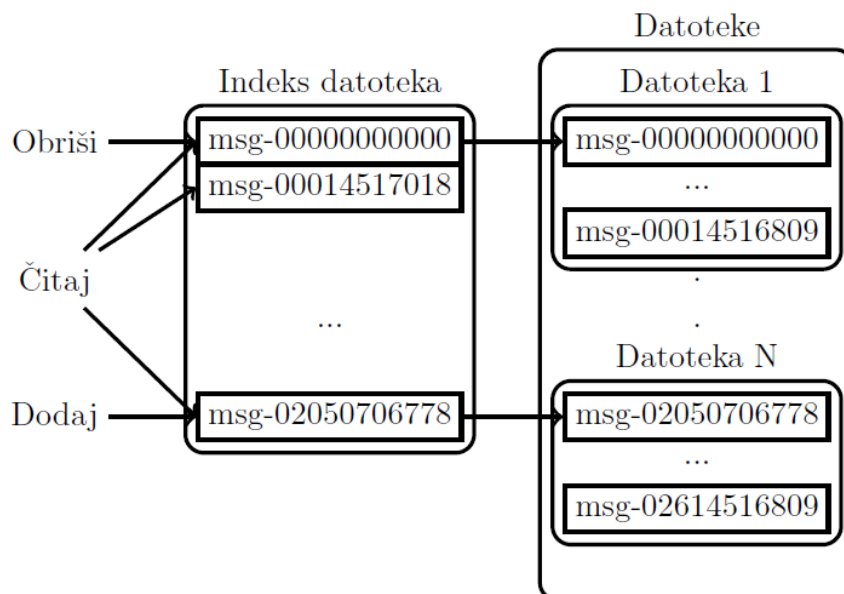
Kada objavljivač objavi poruke na temu posrednik će zapisati poruke u određeni

pretinac. Posrednik koristi sustav straničenja i tvrdi disk za zapisivanje poruka opisano u [6]. Umjesto da posrednik koristi memoriju procesa za priručnu memoriju posrednik poruke predaje sustavu straničenja operacijskog sustava. Operacijski sustav će poruke zapisati u stranice i pohraniti u nedodijeljene dijelove radne memorije. Poruke se zapisuju na tvrdi disk samo kada operacijski sustav želi osloboditi memoriju sustava straničenja. Opisano gospodarenje memorijom dopusta posredniku da čuva veliku količinu poruka bez gubitka učinkovitosti.

Pretplatnici često dohvaćaju uzastopne poruke pa se one čitaju iz priručne memorije umjesto iz tvrdog diska. Zahvaljujući sustavu straničenja priručna memorija sa zapisanim porukama će postojati neko vrijeme nakon ispada posrednika. Korištenjem sustava straničenja za zapisivanje poruka izbjegava se korištenje sakupljača smeća Java virtualnog stroja. Umjesto da se poruke zapisu dva puta, jednom u dodijeljenu memoriju procesa i jednom u sustav straničenja poruke se zapisu samo jednom u sustav straničenja.

Kada pretplatnik želi pročitati poruke posrednik će pronaći, pročitati i poslati poruke pretplatniku. [7] [12] opisuju kako posrednik s `sendfile` i `zero-copy` funkcijama čita poruke iz radne memorije i šalje u memoriju mrežne kartice u jednom koraku. Kako bi se dodatno ubrzao rad posrednici, objavljiivači i pretplatnici grupe poruka čitaju, zapisuju i šalju u istom binarnom obliku. Prije slanja poruka pretplatnicima posrednik može sažeti poruke.

## 2.2.4. Pretinac



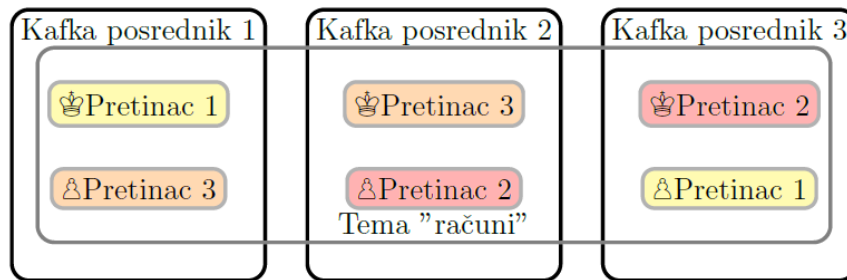
Slika 2.3: Kafka dnevnik

Pretinac je logički dnevnik i najmanja gradivna jedinica teme. Kada objavljiivač objavi poruku u temu grozda posrednik poruke zapiše u pretinac teme. 2.3 prikazuje dnevnik izveden kao skup datoteka iste veličine. Kada posrednik zapiše poruke u pretinac on doda poruke na kraj datoteke. Datoteke se predaju sustavu straničenja operacijskog sustava tek nakon određenog vremena ili broja dodavanja. Svaka poruka ima jedinstvenu oznaku koja je ujedno i odmak unutar datoteke. Odmak sljedeće poruke računa se kao zbroj odnaka i veličine ranije poruke. Zato su oznake poruka jedinstvene i strogo rastuće ali nisu uzastopne.

Protok poruka kroz temu može biti toliki da izazove preopterećenje posrednika. Zato je temu moguće podijeliti na više pretinaca. Ako je B broj posrednika u Kafka grozdu onda tema može biti podijeljena na najviše B pretinaca. Svaki pretinac teme mora biti dodijeljen različitom posredniku.

Ako se dogodi ispad posrednika, poruke u pretincu postat će nedostupne. Ako poruke moraju biti dostupne čak i tijekom ispada posrednika pretinac se mora umnožiti. Svaki pretinac može biti umnožen najviše B puta ako je B broj posrednika u Kafka grozdu. Svaki umnoženi pretinac bit će dodijeljen različitom posredniku jer dodjela istom ne povećava dostupnost uslijed ispada. Ako se pretinac umnoži B puta onda će poruke u pretincu postati nedostupne tek ako se dogodi strogo više od B-1 ispada posrednika.

Kada su pretinci umnoženi potrebno je ujednačiti poruke u svakom umnoženom pretincu. Zato će jedan posrednik biti vođa pretinca, a ostali posrednici će biti pratitelji pretinca. Svaki posrednik može biti vođa najviše jednog pretinca po temi. Vođa pretinca je jedini posrednik koji smije čitati ili pisati poruke u pretinac. Zadaća pratitelja je preusmjeriti korisnike na vođu pretinca i uskladiti svoj pretinac s pretincem vođe. Pratitelji mogu preusmjeriti korisnike na vođu pretinca tako da pitaju ZooKeepera tko je vođa kojeg pretinca. Svaki pratitelj ima pretplatnika koji čita poruke iz pretinca vođe i zapisuje poruke u umnoženi pretinac.



**Slika 2.4:** Pretinci vođe i pretinci sljedbenici

2.4 prikazuje tri posrednika u grozdu. Tema "računi" je podijeljena na najveći mogući broj pretinaca: broj posrednika u grozdu. Kako se poruke ne bi izgubile zbog ispada posrednika svaki pretinac teme je umnožen jednom. Vođa žutog pretinca je nasumično posrednik jedan. Vođa crvenog pretinca će biti ili posrednik dva ili tri jer se vodstvo pretinaca mora jednoliki raspodijeliti. Ako vođa crvenog pretinca postane posrednik tri, onda će vođa narančastog pretinca postati posrednik dva. Pretinac pratitelji se ravnomjerno rasporede po posrednicima tako da pratitelji nikad nisu zajedno s vođom u istom posredniku.

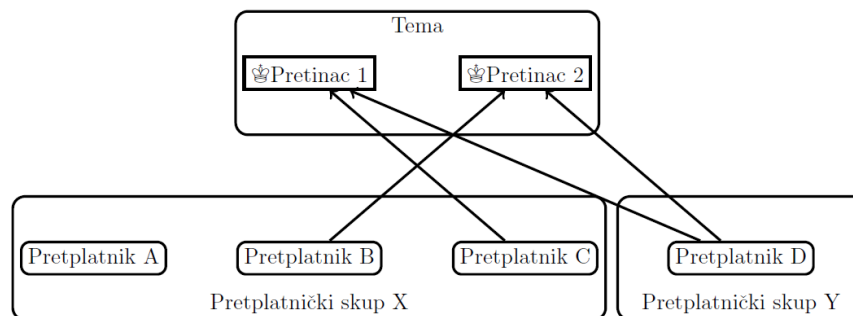
Osim sto vođa pretinca je jedini posrednik koji može čitati ili pisati poruke iz pretinca, on se mora brinuti o pratiteljima. Vođa smatra da je pratitelj živ samo ako je prijavljen u Kafka grozd i ako ne zaostaje s čitanjem poruka iz pretinca vode. Ako zaostaje, vođa će pratitelja izbaciti iz skupa usklađenih pratitelja (ISR). Kada vođa primi poruke od objavljiivača on će poruke zapisati u svoju particiju i čekat će potvrde pratitelja. Tek kada svi pratitelji pročitaju i zapišu poruke u svoj pretinac će vođa pretinca poslati potvrdu objavljiivaču. Pretplatitelji mogu čitati samo one poruke koje su i vođa i pratitelji zapisali u svoje pretince.

Ako se posredniku koji je vođa pretinca dogodi ispad, nitko neće moći čitati ili pisati poruke u pretinac. Zato će pratitelji glasati tko će od pratitelja postati novi vođa pretinca. Samo pratitelji koji su u skupu ukradenih pratitelja mogu postati novi vođa. U

iznimnom slučaju kada ne postoji usklađeni pratitelj neusklađeni pratitelj može postati novi vođa pretinca.

Posrednika, tema, pretinaca i umnoženih pretinaca je puno. Kako posrednici ne bi glasali za novog vođu pretinca za svaki pretinac zasebno jedan od posrednika je zadužen da bude nadglednik glasanja. Ako se dogodi ispad posrednika, nadglednik glasanja će ubrzati glasanje novog vođe pretinca.

### 2.2.5. Pretplatnik



**Slika 2.5:** Kafka pretinci i skup pretplatnika

Pretplatnik je korisnički program koji čita poruke iz teme. 2.5 prikazuje odnos pretplatnika i teme. Svaki pretplatnik je član samo jedne skupine pretplatnika. Skupina pretplatnika sastoji se od barem jednog člana. Skupina pretplatnika zajedno čita poruke iz teme. Neki pretplatnici u skupu pretplatnika biti će zaduženi za čitanje poruka iz teme. Svaki zaduženi pretplatnik ima barem jedan pretinac iz kojeg jedino on može čitati poruke. Skupina pretplatnika može istovremeno napraviti najviše  $P$  čitanja ako je  $P$  broj pretinaca i u skupini pretplatnika je barem  $P$  članova. Poruke se mogu pročitati jedino iz pretinca vođe. Ako pretplatnik uputi zahtjev za čitanje pratitelju pretinca on će pretplatnika preusmjeriti na vođu pretinca.

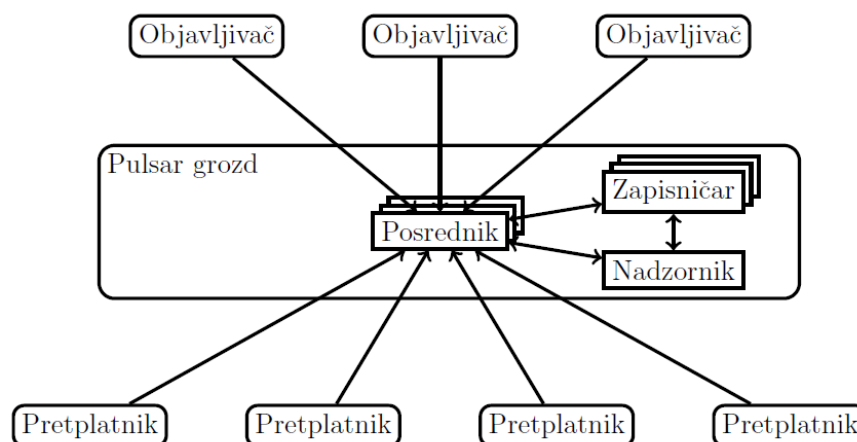
Pretplatnik u zahtjevu za čitanje navodi odmak zadnje pročitane poruke i koliko poruka želi pročitati. Posrednik će dostaviti sve poruke od odmak do tražene količine ili dok ne dođe do kraja pretinca. Odmak poruke je i jedinstvena oznaka poruke i mjesto u datoteci posrednika gdje se poruka nalazi. Posrednik ne pazi koje je poruke pretplatnik pročitao. Pretplatnik je zadužen za rukovanje odmakom. Odmak zadnje pročitane poruke pretplatnik šalje u posebnu temu posrednika. Ako se dogodi ispad pretplatnika, drugi pretplatnik će pročitati odmak iz teme i nastaviti s čitanjem poruka iz pretinca.

Posrednik će jednom pretplatniku po skupini pretplatnika dostaviti poruku barem jednom. Ako se dogodi ispad pretplatnika ista poruka se može dostaviti više puta. Ako je nedopustivo istu poruku pročitati više puta onda korisnik mora napraviti vlastiti algoritam ili koristiti transakcijskog pretplatnika. Pretplatnik čita poruke onim redoslijedom kojim su poruke zapisne u pretinac. Posrednici vremenski ne uređuju dostavu poruka iz svih pretinaca, ali su poruke vremenski uređene u svakom pretincu pojedinačno. Ako je potrebno vremenski urediti sve poruke u svim pretincima onda je potrebno razviti vlastiti algoritam ili napraviti temu sa samo jednim pretincem.

Više skupina pretplatnika može istovremeno citati poruke iz iste teme i pretinaca. Posrednik će odaslati istu poruku svim pretplaćenim pretplatnicima. Posrednici će skupu pretplatnika omogućiti čitanje poruka samo ako su svi pratitelji u skupu usklađenih pratitelja i vođa pretinca zapisali poruke u pretinac. Nakon što pretplatnik primi poruke on će u temi odmak a ažurirati odmak do kojeg je pročitao poruke.

Model povlačenja poruka pretplatnicima dopušta čitanje poruka brzinom koja njima odgovara. Model dopušta i učinkovito razaslanje iste poruke na više pretplatnika. Ako pretplatnik želi ponovno pročitati poruku on samo treba poslati odmak pročitane poruke. Kako pretplatnik ne bi zaglavio u petlji ako u pretincu nema novim poruka on se može blokirati dok ne dođu nove poruke. Posrednik nikad ne zna kada su svi pretplatnici pročitali poruke i zato ih ne može obrisati nakon čitanja. Posrednici su zato napravljeni da s povećanjem nepročitanih poruka njihova učinkovitost ne opada. Posrednik će poruke izbrisati nakon zadanog vremena.

## 2.3. Apache Pulsar



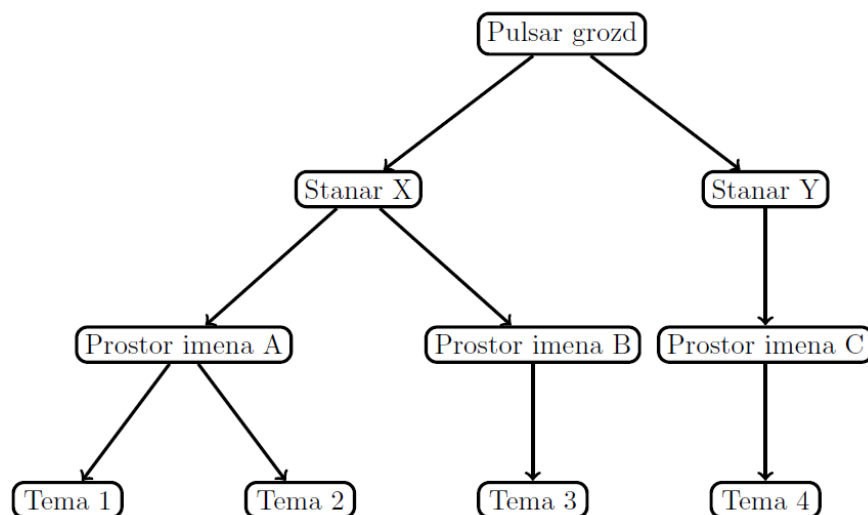
Slika 2.6: Apache Pulsar grozd

Apache Pulsar je moderni alat za raspodijeljenu obradu tokova podataka. 2.6 prikazuje Pulsar proces koji se sastoji od barem jednog Pulsar grozda. Pulsar grozd sastoji se od barem jednog Pulsar posrednika, od barem jednog Apache BookKeeper zapisnicara i Apache ZooKeeper nadzornika. Objavljiivači šalju poruke u temu grozda. Posrednici primaju i poslužuju poruke, a zapisnicari ih zapisuju. Pretplatnici čitaju poruke iz teme grozda i obrađuju podatke. [3] [8] [9] [10] opisuju primjenu, arhitekturu i izvedbu Pulsar pretplatnika, upravljane knjige, posrednika, teme, pretplate i objavljiivača.

### 2.3.1. Objavljiivač

Objavljiivač je korisnički program koji šalje poruke u temu. Poruke se nužno sastoje od sadržaja, oznake objavljiivača, jedinstvene oznake poruke i vremena objave. Poruke se mogu slati u skupini. Prije slanja, poruke se mogu sažeti. Tema može biti podijeljena na pretince. Objavljiivač može usmjeriti poruke u određeni pretinac. Poruke se mogu usmjeriti na jedan nasumični pretinac, na točno određene pretince koristeći ključ ili se mogu slati jednoliko na sve pretince. Objavljiivači ili čekaju potvrdu dok posrednik zapisuje poruke ili nastave s radom i naknadno provjere je li su poruke primljene i zapisane.

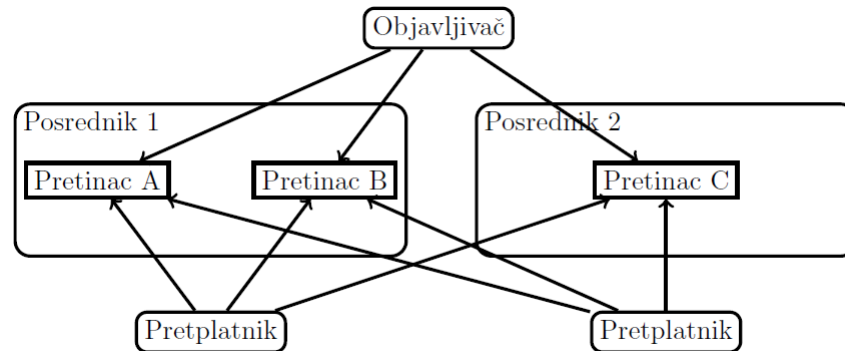
### 2.3.2. Tema i pretplata



**Slika 2.7:** Pulsar teme, prostori imena i stanari

Tema je imenovani tok poruka koji prenosi poruke od objavljiivača do pretplatnika kroz posrednika. Tema je izgrađena kao poveznica. 2.7 prikazuje stupnjevanje tema,

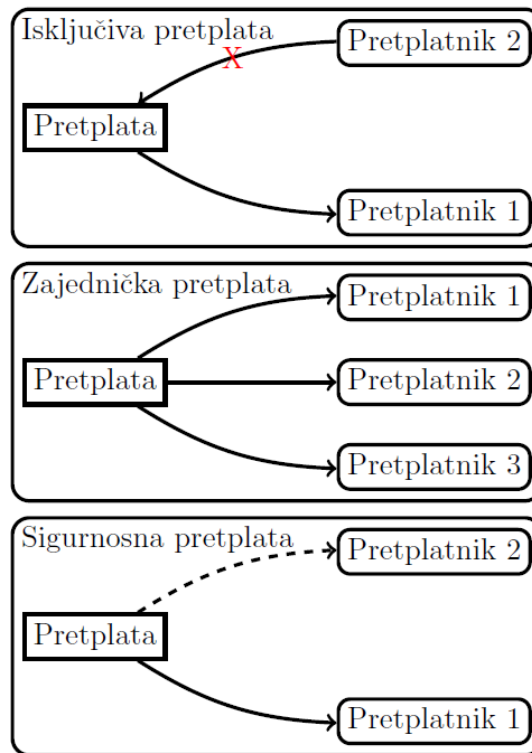
prostor imena i stanara. Svaka tema pripada jednom prostoru imena. Prostor imena je upravna jedinka kojom se mijenjaju postavke tema. Svaki prostor imena pripada jednom stanaru. Stanar određuje način pristupa, korištenje resursa, brisanje poruka i izolaciju prostora imena. Stanari pripadaju Pulsar grozdu.



**Slika 2.8:** Pulsar pretinci

Moguće je napraviti bezmemorijske teme koje poruke čuvaju do slanja pretplatnicima ili ispada posrednika. Bezmemorijske teme objavljene poruke potiskuju prema pretplatnicima. Prednost bezmemorijskih tema je brzina. Količina nepročitanih poruka u temi ne utječe na protok poruka kroz temu. Poruka se briše iz teme kada svi pretplatnici pročitaju poruku ili kada je poruka pročitana i starija od zadane vrijednosti ili kada je nepročitana i starija od zadane vrijednosti. Tema se može podijeliti na više podtema zvanih pretinac. 2.8 prikazuje odnos objavljiivača i pretplatnika prema temi koja je podijeljena na pretince. Pretinci su ravnomjerno raspodijeljeni po posrednicima. Broj istovremenih pretplatnika odnosno čitanja nije ograničeno brojem pretinaca.





**Slika 2.9:** Pulsar pretplate

Kako objavljiivači usmjeravaju dostavu poruka na pretince tako pretplatnici čitaju poruka koristeći pretplate. 2.9 prikazuje tri vrste pretplate. Isključiva pretplata pravo čitanja poruka daje samo jednom pretplatniku. Ako drugi pretplatnik pokuša čitati iz isključive pretplate on će biti odbijen. Zajednička pretplata jednoliko dostavlja poruke pretplatnicima. Svaka poruka dostavit će se samo jednom pretplatniku. Zajednička pretplata ne podržava skupnu potvrdu dostave poruka niti pazi na vremensko uređenje dostave poruka. Sigurnosna pretplata pravo čitanja daje samo jednom pretplatniku dok se pretplatniku ne dogodi ispad. Kada se dogodi ispad drugotni pretplatnik će nastaviti čitanje poruka od mjesta ispada.

### 2.3.3. Posrednik

Posrednik je program bez stanja koji se sastoji od dva dijela. Prvi dio je poslužitelj s REST sučeljem za upravljanje i pretraživanje tema, a drugi dio je otpravnik za prijenos podataka. Umjesto da objavljiivači i pretplatnici izravno razgovaraju s posrednikom mogu se spojiti na zastupnika koji će preusmjeriti njihove zahtjeve posrednicima. Posrednik može odbaciti udvostručene poruke tako da ih ne proslijedi zapisničarima.

Pulsar grozdovi mogu umnožiti poruke ako pripadaju istom procesu. Zadaća posrednik je poslužiti poruke pretplatnicima iz upravljane knjige ili BookKeeper zapis-

ničara, dok je ZooKeeper nadzornik zadužen za čuvanje podata o grozdu. Zapisničar zapisuje poruke koje su poslane posredniku.

#### **2.3.4. Upravljana knjiga**

BookKeeper grozd je raspodijeljeni zapisivač koji se sastoji od barem jednog zapisničara. Zapisničar zapisuju poruke koje mu posrednici pošalju. Poruke se mogu umnožiti i zapisati u više knjiga odjednom. Svaka tema sastoji se od barem jedne knjige. Kapacitet poruka koji može se povećati dodavanjem zapisničara. Zapisničari mogu istovremeno čitati i pisati poruke.

Knjiga je struktura podatka u koju se poruke mogu dodati samo na kraj. Nakon što se knjiga zatvori ona se jedino može čitati. Ako se zapisničaru dogodi ispad, knjiga se zatvori. Kada se ispad otkloni zapisničar će ustanoviti u kojem je stanju knjiga i ustanovljeno stanje poslati ostalim zapisničarima u grozdu.

Upravljana knjiga je skup BookKeeper knjiga u koju se upisuju poruke koje pripadaju jednoj temi. Iako se poruke mogu zapisati u samo jednu knjige više BookKeeper knjiga olakšava brisanje i pisanje poruka. Upravljana knjiga sastoji se skupa tokova podata koji se zapisuju u knjigu s jednim pisačem i skupa pokazivača koji prate koje poruke su pretplatnici pročitali. Zapisivač prije pisanja poruke u upravljanoj knjigi poruke zapisuje u dnevnik.

#### **2.3.5. Pretplatnik**

Pretplatnik je korisnički program koji se pretplaćuje na pretplatu teme. Postoje tri vrste pretplate. Svaka pretplata određuje način na koji pretplatnik čita poruke. Pretplatnik može ili biti blokiran dok posrednik ne dobije poruku ili nastaviti s radom i dobiti budućnosnicu kojom će čitati poruke. Pretplatnik može pojedinačno ili skupno potvrditi poruke.

Ako korisnik nije zadovoljan izvedenim pretplatnikom on može koristiti sučelje čitača. Sučelje čitača je biblioteka koja omogućuje ručno potvrđivanje poruka, ponovno čitanje poruke i odbacivanje umnoženih poruka.

### **2.4. RabbitMQ**

### **3. Postojeca rjesenja**

## **4. Arhitektura rjesenja**

## **5. Dizajn rjesenja**

## **6. Izvedba rjesenja**

## **7. Rezultati**

## **8. Zaključak**

Zaključak.



# LITERATURA

- [1] *Kafka: a Distributed Messaging System for Log Processing*, June 2011. Apache Foundation. URL <https://www.microsoft.com/en-us/research/wp-content/uploads/2017/09/Kafka.pdf>. ACM 978-1-4503-0652-2/11/06; NetDB workshop 2011.
- [2] *Official Kafka Documentation*. Apache Foundation, 2019. URL <https://kafka.apache.org/documentation/>.
- [3] *Official Pulsar Overview*. Apache Foundation, 2019. URL <https://pulsar.apache.org/docs/en/concepts-overview/>.
- [4] B. Babcock, S. Babu, M. Datar, R. Motwani, i J. Widom. Models and issues in data stream systems. Technical Report 2002-19, Stanford InfoLab, 2002. URL <http://ilpubs.stanford.edu:8090/535/>.
- [5] J. Francis i M. Merli. Open-sourcing pulsar, pub-sub messaging at scale, September 2016. URL <https://yahooeng.tumblr.com/post/150078336821/open-sourcing-pulsar-pub-sub-messaging-at-scale>.
- [6] P.-K. Kamp. Notes from the architect, 2014. URL <http://varnish-cache.org/docs/trunk/phk/notes.html>.
- [7] Michael Kerrisk. Linux sendfile, September 2017. URL <http://man7.org/linux/man-pages/man2/sendfile.2.html>.
- [8] M. Merli i K. Ramasamy. Why apache pulsar? part 1. <https://streaml.io/blog/why-apache-pulsar>, August 2017.
- [9] M. Merli i K. Ramasamy. Why apache pulsar? part 2. <https://streaml.io/blog/why-apache-pulsar-part-2>, August 2017.

- [10] M. Merli i K. Ramasamy. Introduction to the apache pulsar pub-sub messaging platform. <https://streaml.io/blog/intro-to-pulsar>, August 2017.
- [11] N. Narkhede. Kafka compression, February 2017. URL <https://cwiki.apache.org/confluence/display/KAFKA/Compression>.
- [12] S. Palaniappan i P. Nagaraja. Linux sendfile, September 2008. URL <https://developer.ibm.com/articles/j-zero-copy/>.
- [13] I. P. Žarko, K. Pripužić, I. Lovrek, i M. Kušek. *Raspodijeljeni sustavi*, 1.3 izdanju, 2013.

**Stvarnovremensko praćenje parametara ispravnosti rada u sustavu za  
raspodijeljenu obradu tokova podataka**

**Sažetak**

Sažetak na hrvatskom jeziku.

**Ključne riječi:** Ključne riječi, odvojene zarezima.

**Real-Time Health Monitoring in Distributed Data Stream Processing System**

**Abstract**

Abstract.

**Keywords:** Keywords.