

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1735

**Geoprostorni sustav objavi-pretplati za
učinkovito filtriranje toka senzorskih
podataka**

Borna Radotić

Zagreb, lipanj 2018.

Zagreb, 9. ožujka 2018.

DIPLOMSKI ZADATAK br. 1735

Pristupnik: **Borna Radotić (0036470919)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi

Zadatak: **Geoprostorni sustav objavi-pretplati za učinkovito filtriranje toka senzorskih podataka**

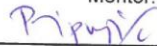
Opis zadatka:

Vaš zadatak je osmisлити, izvesti u programskom jeziku Java te testirati centralizirani geoprostorni sustav objavi-pretplati kod kojeg su objave i pretplate definirane kao geoprostorni objekti (npr. točke, linije, poligoni). Objava je senzorsko očitavanje koje se sastoji od sljedećih podataka: vrijeme očitavanja, geoprostorni objekt koji predstavlja lokaciju očitavanja, identifikator senzora, tip senzora, vrsta očitavanja, vrijednost očitavanja i jedinica očitavanja. Pretplata se sastoji od geoprostornog objekta koji predstavlja područje interesa pretplatnika te dodatnih uvjeta (npr. intervala vrijednosti očitavanja ili ključnih riječi) na osnovu kojih se vrši kontinuirana usporedba s nadolazećim objavama.

Prilikom izrade sustava iskoristite Javine programske knjižnice JTS (Java Topology Suite) i GeoTools, a objave i pretplate realizirajte u formatu GeoJSON. Definirajte neophodna sučelja i apstraktne klase koje će omogućiti podršku za numeričke i tekstualne senzore. Omogućite pretplaćivanje klijenata na sustav i objavljivanje preko TCP priključnica. Istražite geoprostorne indekse koji su podržani u navedenim Javinim programskim knjižnicama, usporedite njihove karakteristike i performanse te implementirajte jednostavnu mogućnost odabira korištenog geoprostornog indeksa u sustavu. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 16. ožujka 2018.
Rok za predaju rada: 29. lipnja 2018.

Mentor:



Izv. prof. dr. sc. Krešimir Pripuzić

Djelovođa:



Doc. dr. sc. Boris Milašinović

Predsjednik odbora za
diplomski rad profila:



Izv. prof. dr. sc. Igor Mekterović

Sadržaj

Uvod.....	1
1. Sustavi objavi-pretplati	3
2. Objave i pretplate	6
2.1 Geoprostorni objekti i geometrija	6
2.1.1 JTS Geometry klasa	7
2.1.2 GeoJSON	9
2.2 Objave u sustavu	10
2.3 Pretplate u sustavu	12
2.4 Mjerni tipovi podataka i mjerne jedinice	14
3. Arhitektura sustava	17
3.1 Dizajn sustava.....	17
3.2 Agregator	20
3.3 Sustav filtriranja.....	22
4. Strategije indeksiranja.....	26
4.1 Geoprostorne strategije indeksiranja.....	26
4.1.1 R-tree	27
4.1.2 Quadtree	29
4.2 Indeksiranje po uvjetima pretplate	31
4.2.1 Interval tree.....	31
4.2.2 Apache Lucene.....	33
4.2.3 Indeksiranje enumeracija.....	34
5. Proširivanje sustava	35
5.1 Dodavanje novog tipa podatka.....	35
5.2 Dodavanje nove geoprostorne strategije indeksiranja.....	36
6. Primjena sustava.....	38
7. Eksperimentalna evaluacija.....	40
7.1 Opis eksperimenta	40
7.2 Rezultati eksperimenta.....	41
Zaključak	46
Literatura	48
Sažetak	49
Summary.....	50

Uvod

Sustavi objavi-pretplati su primjer sustava koji su našli primjenu u različitim područjima zbog svoje skalabilnosti i mrežne dinamičnosti. To su sustavi u kojima se skupina korisnika, koji se nazivaju pretplatnici, pretplaćuje na određene karakteristike objava koje dolaze u sustav. Pretplatnicima će biti isporučene samo one objave čije karakteristike zadovoljavaju njihove pretplate. Sam rad sustava zato izrazito ovisi o vrsti objava koje mogu biti objavljene u sustavu i dopuštenim uvjetima koje pretplata može definirati nad objavama. Glavni problem takvih sustava je kako nakon dolaska objave što učinkovitije odrediti koji pretplatnici su zainteresirani za zaprimljenu objavu, odnosno čiji uvjeti pretplate su njome zadovoljeni. Ovaj zahtjev je posebno izražen u sustavima koji isporuku vrše u stvarnom vremenu te je brza obrada novih, zaprimljenih objava ključan dio rada takvog sustava. Proces pronalaska željenih pretplatnika među skupom svih pretplata naziva se filtriranje te je ključna funkcionalnost sustava objavi-pretplati. Cilj filtriranja je uz minimalno vrijeme i utrošak memorije pronaći sve pretplate čiji su uvjeti zadovoljeni, bez lažnih pozitivnih rezultata ili izostavljanja neke od pretplata.

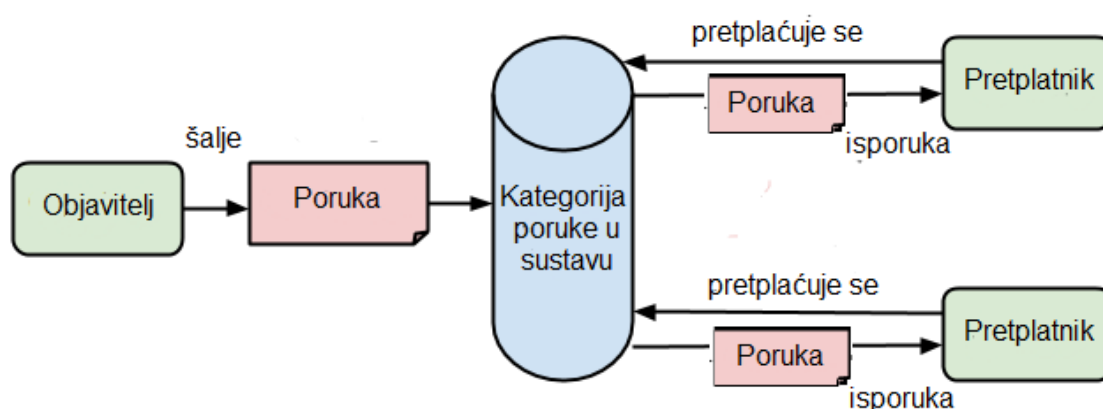
U okviru ovog rada razmatra se i implementira jedan objavi-pretplati sustav za rad u stvarnom vremenu. Glavni dijelovi objave u sustavu koji se implementira su izmjerena vrijednost, mjerna jedinica i geometrija geografskog područja za koje se mjerenje obavlja. Ovakve objave idealne su za mjerenja raznih senzora koja su obavljena na određenom geografskom području. Pretplatnici također definiraju vlastitu geometriju kojom određuju geografsko područje za koje žele primati objave i prostorni odnos prema geometriji objave. Prostorni odnos definira odnos između geometrije pretplate i objave koji mora biti zadovoljen npr. da jedna presijeka drugu, dodiruje ju, prekriva itd.. Pretplatnici tako definiraju interesno područje s kojeg žele primati obavijesti. Na primjer, pretplatnik može definirati geometriju grada Zagreba i geometrijski odnos presjeka kako bi primao sve obavijesti s područja grada. Uz to, pretplatnik može postaviti dodatno ograničenje na izmjerene vrijednosti objave. Dopuštena ograničenja ovise o tipu podataka mjerne jedinice na koju se pretplaćuju. Na primjer, ako je riječ o brojčanoj vrijednosti

objave dodatni uvjet je željeni raspon brojeva koji pretplatnik prihvaća. Ako je pak riječ o tekstualnom tipu podatka korisnik može definirati ključne riječi koje moraju biti sadržane u objavi kako bi uvjet bio zadovoljen. S time pretplatnik, uz interesno geometrijsko područje, postavlja još jedan dodatni uvjet koji treba uzeti u obzir prilikom filtriranja pretplata prilikom nailaska nove objave.

Vidljivo je da je glavni zahtjev ovog sustava, zbog kompleksnosti uvjeta koji pretplatnik može postaviti, kako što učinkovitije filtrirati pretplate pri dolasku nove objave. Taj problem je i temelj razmatranja u ovom radu te je njegovo rješavanje glavni fokus razvijenog sustava. Razvijeni sustav rješavanju problema filtriranja pristupa kroz korištenje različitih strategija indeksiranja pretplata te se na kraju rada prikazuje evaluacija korištenih pristupa prilikom filtriranja po sadržaju objave. Uz to, sustav omogućava fleksibilan rad s različitim tipovima podataka i mjernim jedinicama te time omogućava dinamičko proširivanje vrsta objava i pretplata s kojima sustav može raditi. Rezultat ovog rada je modularan sustav koji se može prilagoditi zahtjevima različitih senzora koji šalju objave u sustav i pretplatama koje se mogu učinkovito filtrirati kod nadolazećeg toka objava.

1. Sustavi objavi-pretplati

Sustavi s arhitekturom objavi-pretplati slijede jednostavan obrazac slanja poruka u kojem pošiljatelj poruke – objavitelj ne poznaje specifičan set primatelja poruke koju šalje već isključivo formira željenu poruku i šalje ju sustavu s navedenom arhitekturom na daljnju obradu [1]. Objavitelj definira klasu ili tip poruke i sve njene atribute koji mogu varirati ovisno o specifičnostima pojedinih sustava objavi-pretplati [1]. Poruka koja je dostavljena sustavu može dalje biti isporučena jednom pretplatniku, više njih ili niti jednom pretplatniku. Pretplatnik se s druge strane također ne pretplaćuje na konkretan set objavitelja, niti je svjestan tko su konkretni objavitelji u sustavu. Pretplatnik definira kategoriju ili tip poruke za kojom izražava interes bez obzira tko je konkretno pošiljatelj te poruke. Uz navedenu pretplatu na kategoriju ili tip poruke u nekim sustavima objavi-pretplati pretplatnik može definirati interes za poruku i po nekim specifičnim atributima te poruke tzv. pretplata po sadržaju. U arhitekturi objavi-pretplati centralni sustav predstavlja posrednika između objavitelja koji ne poznaje set primatelja i pretplatnika koji ne poznaje set objavitelja. Na ovaj način objavitelji isporučuju poruku svima zainteresiranima bez brige o održavanju liste primatelja, a pretplatnici su generalno pretplaćeni na željeno područje poruka bez potrebe za brigom oko detalja objavitelja tih poruka.



Slika 1 Arhitektura sustava objavi-pretplati s primjerom pretplate na kategoriju poruke

Nakon dolaska objave u sustav objavi-pretplati potrebno je odrediti sve pretplate koje su zainteresirane za dobivenu objavu te tu objavu proslijediti do pretplatnika. Proces određivanja pretplatnika kojima je potrebno isporučiti objavu naziva se filtriranje [2]. Kako je pretplatnik ovisno o sustavu u mogućnosti definirati željeni tip poruke, njenu kategoriju ili pak specifičnosti poruke kroz njene attribute sam način filtriranja direktno ovisi o načinu na koji se korisnik pretplaćuje u sustavu. Zbog navedenih vrsta pretplata dva glavna tipa filtriranja u sustavima objavi-pretplati su filtriranje po kategoriji (tipu) poruke i filtriranje po sadržaju poruke [2]. Kod filtriranja po kategoriji unutar sustava postoje unaprijed definirane kategorije u koje poruka objavitelja može pripadati te se pretplatnici također pretplaćuju na željene kategorije. Kada objava za pojedinu kategoriju dođe u sustav, ta se objava isporučuje svim pretplatnicima unutar kategorije te objave. S druge strane kod filtriranja po sadržaju poruke korisnik sam klasificira željene poruke tako da odredi prihvatljivost atributa poruke koja dolazi u sustav. Pri dolasku poruke u sustav potrebno je odrediti sve pretplate čiji kriteriji su zadovoljeni atributima nove objave te ju sukladno filtriranju i isporučiti. Uz dva navedena načina filtriranja postoji i hibridni u kojem se pretplatnici pretplaćuju na određenu kategoriju objave a uz to postavljaju uvjet i na sam sadržaj pojedinih atributa objave. Takva hibridna implementacija je odabrana i prilikom oblikovanja sustava razvijenog u ovom radu.

Arhitektura centralnog sustava u objavi-pretplati arhitekturi također može biti različita ovisno o primjeni no generalno se dijeli u dvije skupine – centralizirana i raspodijeljena [3]. U centraliziranoj arhitekturi postoji jedan poslužitelj koji se ponaša kao posrednik i filter te se sve obavijesti šalju njemu kao i svi zahtjevi za pretplatama. Poslužitelj u takvoj arhitekturi ima kod sebe pohranjene sve pretplate te za svaku objavu nad svima radi filtriranje i zatim isporuku onim pretplatama koje se podudaraju s objavom. Raspodijeljena arhitektura sadrži skup poslužitelja od kojih je svaki dedican za pojedinu kategoriju objava. Takva arhitektura osigurava veću brzinu i bolju skalabilnost prilikom rasta sustava [3].

Prednosti arhitekture objavi-pretplati vidljive su u nepostojanju direktne veze između objavitelja i pretplatnika [1]. Iz tog razloga objavitelji i pretplatnici ne moraju znati ništa o samoj topologiji mreže i pošiljateljima/primateljima već isključivo moraju poznavati oblik poruke koji sustav prihvaća. Ovo omogućava dodavanje

novih objavitelja i pretplatnika bez većih poteškoća i potrebama za sinkronizacijom. Također, zato što ne postoji direktna veza između objavitelja i pretplatnika kao u npr. klijent-server arhitekturi, ovaj sustav je prostorno i vremenski neovisan. Prostorna neovisnost je vidljiva u činjenici da objavitelj ne poznaje identifikator pretplatnika, kao što niti pretplatnik nema identifikator objavitelja. Uz anonimnost, ovo uvelike povećava skalabilnost sustava i jednostavnost dodavanja novog klijenta u sustav. Vremenska neovisnost se odnosi na činjenicu da objavitelj i pretplatnik ne moraju biti istovremeno aktivni prilikom komunikacije. Razlog tomu je što se centralni sustav ponaša kao posrednik te može pohraniti poruku objave i isporučiti ju pretplatniku kada on postane aktivan. Uz nepostojanje direktne veze između objavitelja i pretplatnika, prednost sustava objavi-pretplati očituje se i u njegovoj skalabilnosti. Skalabilnost proizlazi iz činjenice da se neovisno o broju novih pretplatnika i objava filtriranje obavlja na definirani način po kategoriji, tipu ili sadržaju. To omogućava rast centralnog sustava kroz raspodijeljenu obradu po pojedinim domenama interesa, usmjeravanje sadržaja prema pretplatnicima s istim interesom i efikasnu pohranu interesnog sadržaja u priručna spremišta u blizini zainteresiranih pretplatnika.

S druge strane, mane arhitekture objavi-pretplati vidljive su u povećanoj latenciji prilikom isporuke poruka, nesigurnosti isporuke poruke i neprimanju objavljenih poruka [1]. Povećana latencija je posljedica uvođenja posrednika između objavitelja i pretplatnika. Centralni sustav nakon prihvata objave mora provjeriti njenu valjanost te filtrirati postojeće pretplate kako bi utvrdio zainteresirani skup pretplatnika. Iako se na ovaj način isporuka vrši do više pretplatnika, iz perspektive jedne objave i pretplatnika centralni sustav uvodi kašnjenje. Nesigurnost isporuke i neprimanje poruke su direktna posljedica ranije navedene prednosti da ne postoji direktna veza između objavitelja i pretplatnika. Objavitelj tako nakon isporuke ne može znati hoće li njegova poruka biti isporučena svima kojima mora biti. Moguće je da se nekom od pretplatnika ne uspije isporučiti objava uslijed više pokušaja i sustav isporuke odustane od daljnjih pokušaja. Također, pretplatnik nije svjestan da je došla nova obavijest i ako mu nije isporučena može ju propustiti smatrajući da do nje nije niti došlo. Vidljivo je da nepovezivanje objavitelja i pretplatnika uz prednosti, donosi i glavne mane arhitekture uslijed nepostojanja potvrde isporuke.

2. Objave i pretplate

U sustavima objavi-pretplati, uz samu arhitekturu sustava, temelj načina rada sustava kao i sam proces filtriranja određen je formatom objava koje sustav obrađuje i dopuštenim oblikom pretplata na te objave. Sam oblik objava i pretplata ovisi o željenoj funkcionalnosti sustava koji se razvija i njegovoj namjeni. Sustav razvijen u okviru ovoga rada ima za cilj podržati obradu objava senzora koji sadrže izmjerene vrijednosti različitih tipova podataka (brojčani, tekstualni, enumeracije itd.) i geoprostorne podatke o području za koje ta izmjerena vrijednost vrijedi. Sukladno tome pretplatnik može postaviti pretplatu za neko željeno geoprostorno područje i ograničiti vrijednost dodatnim uvjetima (raspon brojčanih vrijednosti, ključne riječi u tekstu itd.). Pri dolasku nove objave, ako je zadovoljen geoprostorni uvjet pretplate kao i svi dodatni uvjeti na vrijednost, obavijest će se proslijediti tom pretplatniku. Primjer moguće primjene takvog sustava su niz senzora koji mjere temperaturu na određenim točkama u gradu i nakon mjerenja odašilju svoju geolokaciju i izmjerenu vrijednost. Pretplatnici se tada mogu pretplatiti na njima zanimljiva geoprostorna područja unutar grada kao i postaviti uvjet na njima zanimljive vrijednosti temperature – npr. sve izmjerene vrijednosti iznad 20 °C na sjevernom djelu grada. Kada takva vrijednost dođe u sustav sa željenog područja pretplatnik će biti obaviješten o njoj.

2.1 Geoprostorni objekti i geometrija

Prilikom definiranja objava i pretplata vidljivo je da im je zajednička značajka geoprostorni sadržaj na kojem se temelji i većina problematike razvijenog sustava. Objave i pretplate se stoga zajednički gledaju kao geoprostorni objekti koji zatim definiraju svoje ostale nužne ili opcionalne attribute ovisno o svojim specifičnostima. Geoprostorni objekti u sustavu tako moraju definirati geometriju za koju se koristi Javina programska knjižnica JTS Topology Suite i njena klasa Geometry te ponuditi metodu za serijalizaciju objekta u GeoJSON tekstualni tip podatka (ova dva odabrana standarda su detaljnije obrađena u potpoglavljima 2.1.1 i 2.1.2). Uz to objekti imaju univerzalni jedinstveni identifikator (UUID) koji se dodjeljuje pojedinoj objavi i pretplati nakon dolaska u sustav zbog njene

identifikacije te atribut koji definira tip objekta, a koristi se kako bi se iz serijalizirane forme što jednostavnije utvrdilo o kojem je objektu riječ. Tip objekta je jednostavna enumeracija koja definira objavu i pretplatu – jedina dva geoprostorna objekta u sustavu.

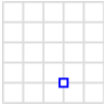
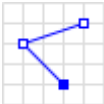
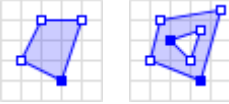
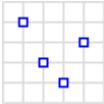
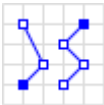
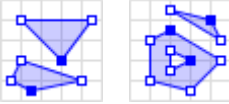


Slika 2 Dijagram apstraktne klase GeospatialObject koju nasljeđuju klase kojima se definira objava i pretplata u sustavu

2.1.1 JTS Geometry klasa

Temeljni atribut geoprostornih objekata je geometrija područja za koje su definirani. Za definiranje geometrijskog područja koristi se Geometry klasa JTS knjižnice. Navedenu klasu nasljeđuje niz klasa koje definiraju moguće geoprostorne objekte koji se mogu instancirati a to su: Point, LineString, Polygon i geometrijske kolekcije – MultiPoint, MultiLineString i MultiPolygon. Opis i prikaz navedenih objekata dan je u tablici 1. Za instanciranje tih objekata koristi se klasa tvornica GeometryFactory koja nudi odgovarajuće metode za instanciranje željenih objekata. Sama geometrija je opisana tipom objekta i nizom koordinata koji ju opisuju (klasa Coordinate). JTS knjižnica odabrana je jer nudi sve potrebne metode za pronalazak relacijskih odnosa između dvije geometrije. Relacijski odnos opisuje da li se dvije geometrije presijecaju, dodiruju, križaju, prekrivaju, sadrže jedna drugu, nalaze jedna unutar druge, ne presijecaju ili su pak jednake [4]. Navedeni odnos je potreban unutar pretplate zbog definiranja prihvatljivih geometrija. Također u kombinaciji s Geotools knjižnicom, koja također koristi JTS Geometry implementaciju, može ponuditi serijalizaciju geometrije u GeoJSON format koji je često korišten standard za serijalizaciju geometrije pa je i u ovom sustavu odabran kao format prilikom razmjene poruka.

Tablica 1 Geometrijski tipovi dostupni u Geometry tipu podatka

Naziv geometrijskog tipa	Slikovni prikaz	Opis geometrijskog tipa
Point		<p>Točka u prostoru određena s jednom koordinatom, odnosno s uređenim parom x,y.</p>
LineString		<p>Skup točaka međusobno povezan dužinama. Svaka navedena točka tvori dužinu s prethodnom i definirane točke ne zatvaraju poligon.</p>
Polygon		<p>Ravnina omeđena s nizom dužina koje su pak kao i kod LineString-a određene s nizom točaka. Nužno je da su prva i zadnja točka jednake kako bi ravnina bila omeđena. Pri definiranju poligona s rupom kao na drugoj slici potrebno je navesti dva polja koordinata od kojih jedno definira poligon a drugo rupu u poligonu.</p>
MultiPoint		<p>Više točka u prostoru, svaka određena svojom koordinatom x,y</p>
MultiLineString		<p>Skup više LineString geometrijskih tipova, svaki od njih je zasebno definiran.</p>
MultiPolygon		<p>Skup više poligona, svaki od njih je zasebno definiran.</p>

2.1.2 GeoJSON

Objave i pretplate su geoprostorni objekti s definiranim geometrijama. Za serijalizaciju objava i pretplata unutar sustava korišten je tip podataka GeoJSON. GeoJSON je format za enkodiranje različitih geografskih struktura podataka koji definira način zapisa geografskih struktura u JSON formatu [5]. GeoJSON podržava sve ranije navedene geometrijske tipove „Point”, „LineString”, „Polygon”, „MultiPoint”, „MultiLineString” i „MultiPolygon”, koji su podržani i od strane JTS-a pa tako i unutar razvijenog sustava [5]. Geometrija tipa koji se definira spremljena je kao objekt za atribut s imenom „geometry” unutar GeoJSON-a. Unutar tog objekta nalazi se atribut s ključem „type” koji sadrži naziv definiranog tipa u tekstualnom obliku npr. „Point”, „Polygon” itd.. Uz tip definiran je i atribut s ključem „coordinates” za koji je vrijednost polje koordinata koje definiraju taj geometrijski oblik. Uz geometriju u GeoJSON-u je moguće definirati i posebna svojstva objekta uz samu geometriju. Ti atributi koji su dio objekta (npr. izmjerena vrijednost i mjerna jedinica kod objave) smješteni su unutar objekta s ključem „properties”. U tom objektu moguće je definirati niz ključ-vrijednost parova koje su neovisne o geometriji a dio objekta koji se serijalizira. Opisano je najbolje vidljivo iz primjera pretplate i objave u GeoJSON formatu dane na slikama 3 i 4 koje su prema geometrijskom tipu „Point” i „LineString” te sadrže vlastita svojstva.

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      50,
      50
    ]
  },
  "properties": {
    "uuid": "a2b0460e-0bf5-4720-be48-0996bc203359",
    "objectType": "Subscription",
    "spatialRelation": "Intersects",
    "measurementTypeName": "Temperature",
    "preferredMeasurementUnitName": "Fahrenheit",
    "subscriptionCondition": "{\"dataType\":\"Numeric\", \"minimumValue\":25.0, \"isMinimumIncluded\":true, \"maximumValue\":25.0, \"isMaximumIncluded\":true}"
  },
  "id": "fid--592fe1fe_1631cf43e01_-8000"
}
```

Slika 3 GeoJSON format pretplate na „Point“ s koordinatama 50,50

```

{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        107,
        107
      ],
      [
        66,
        66
      ],
      [
        50,
        50
      ]
    ]
  },
  "properties": {
    "uuid": "694eee2e-74e0-4907-875e-a0120723c08e",
    "objectType": "Publication",
    "measurementTypeName": "Temperature",
    "measurementUnitName": "Celsius",
    "value": "25.0",
    "matchedSubscriptions": "[a2b0460e-0bf5-4720-be48-0996bc203359]",
    "timestamp": "2018-05-01T18:26:10Z"
  },
  "id": "fid--592fe1fe_1631cf43e01_-7fff"
}

```

Slika 4 GeoJSON format objave s geometrijskim tipom „LineString“

Za serijalizaciju objekta u GeoJSON format korištena je Javina knjižnica Geotools. Ekvivalent GeoJSON-a u Geotools knjižnici je klasa FeatureJSON, a sam JSON se gradi pomoću klase tvornice SimpleFeatureBuilder. Prilikom serijalizacije specificira se geometrija koja se želi serijalizirati te se dodaju sva svojstva objekta koja se žele serijalizirati u „properties“ i rezultat je željena GeoJSON reprezentacija kao na slikama.

2.2 Objave u sustavu

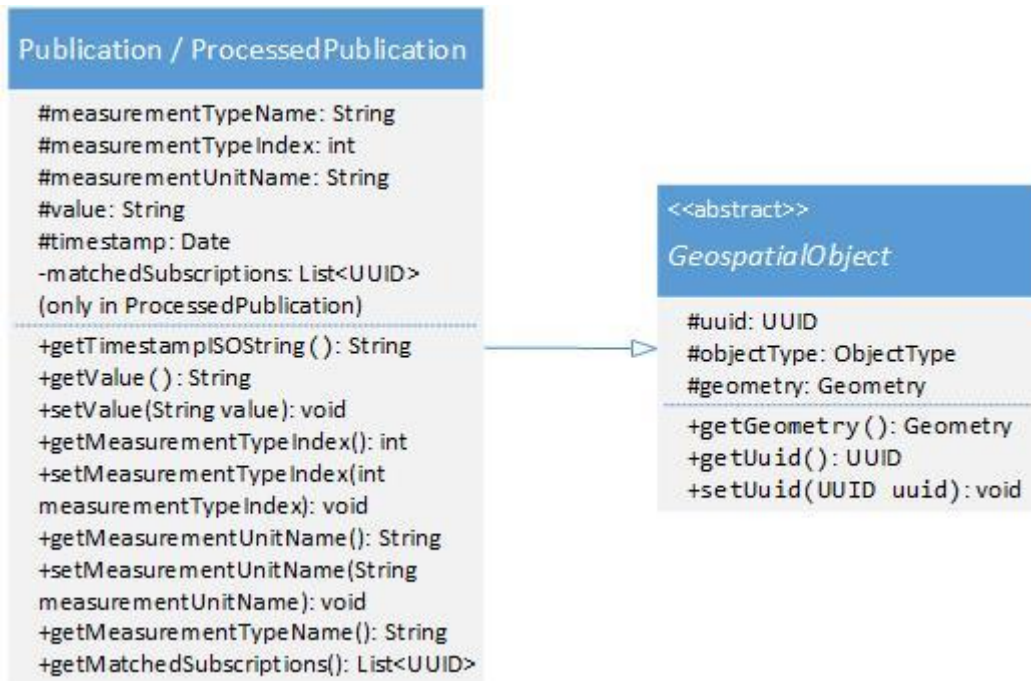
Objave u sustavu predstavljaju rezultate mjerenja senzora koji pristižu u sustav i nakon filtriranja se prosljeđuju zainteresiranim pretplatnicima. Objave su, kao što je opisano u prethodnom poglavlju, geoprostorni objekti te nasljeđujući apstraktnu klasu GeospatialObject sadrže sve njene atribute geometrije, univerzalnog jedinstvenog identifikatora, tipa objekta i implementiraju metodu

pomoću koje se serijaliziraju u GeoJSON objekt. Uz to, objave definiraju sve attribute koji su potrebni kako bi objava slala smislenu poruku. Ti atributi su ime mjernog tipa, ime mjerne jedinice, izmjerena vrijednost, vremenska oznaka i indeks mjernog tipa. Uz navedene, nakon što se objava obradi, tada sadrži i listu svih identifikatora podudarnih pretplata. To su identifikatori svih pretplata kojima se ta objava treba isporučiti.

Atributi mjernog tipa i mjerne jedinice zajednički su za objave i pretplate te su detaljnije objašnjeni u potpoglavlju 2.4. Mjerni tip predstavlja generalnu klasu izmjerene vrijednosti koja pak može biti izražena s nizom mjernih jedinica koje pripadaju toj klasi. Mjerni tipovi su klase mjerenja poput temperature, napona, jačine vjetra, pritiska, jačine zračenja itd.. Mjerna jedinica je jedinica kojom se vrijednost mjerenog tipa može izraziti. Za primjer navedenog tipa temperature mjerne jedinice su Celsius, Fahrenheit, Kelvin itd.. U valjanoj objavi objavitelj može navesti bilo koji tip koji je podržan i dostupan unutar sustava kao i neku od njegovih pripadnih jedinica. Izmjerena vrijednost je glavni dio objave i definira iznos vrijednost mjerenja za navedenu mjernu jedinicu. Izmjerena vrijednost se uvijek šalje u tekstualnom formatu, no vrijednost mora odgovarati mjernom tipu po tipu podatka i definiranim pravilima tog tipa. To znači da za ranije navedeni mjerni tip temperature koji se definira broječanim tipom podatka, sustav očekuje da tekstualnu vrijednost može parsirati u decimalnu vrijednost, dok pak tip koji sadrži enumeraciju, poput tipa koji definira jačinu zračenja, očekuje neku od vrijednosti enumeracije za taj tip. Vremenska oznaka unutar objave predstavlja vrijeme u ISO UTC formatu koje definira kada je mjerenje obavljeno. Indeks mjernog tipa se koristi interno u sustavu zbog bržeg klasificiranja objave prilikom primitka i detaljnije je objašnjen u poglavlju 3. Jednom kada je filtriranje obavljeno univerzalni identifikatori svih pretplata za koje je filtriranje utvrdilo da im je potrebno isporučiti objavu pohranjuju se u listu podudarnih pretplata unutar objave.

Prilikom stvaranja objekta objave od strane objavitelja prije slanja ili prilikom deserijalizacije poruke kod primanja u centralni sustav provjerava se prisutnost svih nužnih atributa. Nužni atributi u objektu objave su uz geometriju i tip objekta još i ime mjernog tipa, ime mjerne jedinice i izmjerena vrijednost. Također, nužno je da su imena predanog tipa i mjerne jedinice valjani i postoje unutar sustava.

Prilikom stvaranja GeoJSON objekta svi dodatni atributi objave zapisuju se kao tekstualne ključ-vrijednost vrijednosti unutar „properties” objekta kako je objašnjeno u prethodnom poglavlju.



Slika 5 Dijagram klase *Publication* odnosno *ProcessedPublication* koja definira objavu koja se objavljuje u sustavu

2.3 Pretplate u sustavu

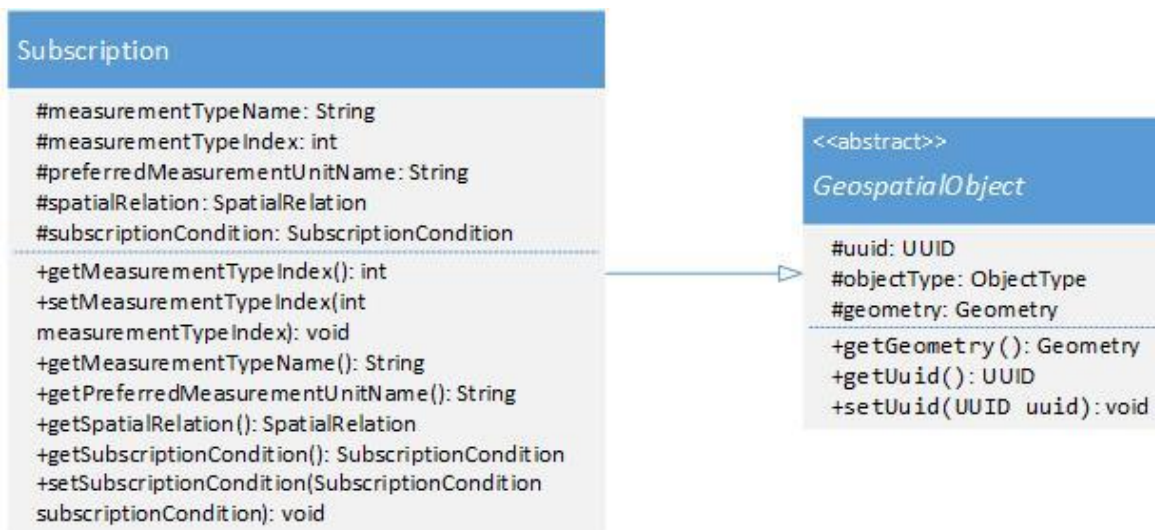
Pretplate u sustavu definiraju interes pretplatnika za pojedino geometrijsko područje, mjerni tip i vrijednosti objava koje zadovoljavaju njihov definirani kriterij. U implementaciji sustava u sklopu ovog rada ne postoji mehanizam isporuke objava pretplatnika, već je fokus samo na zaprimanju pretplata i njihovom filtriranju pri nailasku objave. Pretplate su kao i objave geoprostorni objekti te nasljeđujući klasu *GeospatialObject* sadrže obaveznu geometrijsku komponentu kao i identifikator i tip objekta. Atributi objave koji uz geometriju omogućavaju dodatno postavljanje kriterija filtriranja na zaprimljene objave su: prostorni odnos, ime mjernog tipa, ime preferirane mjerne jedinice, indeks mjernog tipa i uvjet pretplate.

Atributi imena mjernog tipa i preferirane mjerne jedinice kao i kod objave predstavljaju generalne klase vrijednosti koje se mogu mjeriti u sustavu te njihove

mjerne jedinice. Ti atributi su detaljnije objašnjeni u potpoglavlju 2.4. Mjerni tip određuje klasu objava koje interesiraju definiranu pretplatu te se pretplata kod filtriranja odmah eliminira ako nema jednak tip kao i objava. Dakle, ako je pretplata definirana za mjerni tip temperature objava koja je dostavljena za mjerni tip napona neće biti isporučena takvoj pretplati neovisno o drugim atributima. Preferirana mjerna jedinica definira u kojoj mjernoj jedinici je specificiran uvjet pretplate kao i u kojoj mjernoj jedinici pretplatnik želi dobiti isporučenu vrijednost nove objave. Atribut prostornog odnosa definira odnos između geometrije pretplate i geometrije nadolazeće objave koji ako je u procesu filtriranja zadovoljen tada je ispunjen geometrijski uvjet pretplate. Prostorni odnos je unutar sustava definiran s enumeracijom koja sadrži vrijednosti poput: „prekrivena je”, „prekriva”, „sadrži”, „presijeca”, „jednaka je”, „dodiruje” itd. [4]. Na primjer, ako je prostorni odnos postavljen kao „prekriva” (engl. covers), za definiranu geometriju pretplate, uvjet će biti zadovoljen ako geometrija pretplate prekriva geometriju objave. Indeks mjernog tipa, kao i kod objave, koristi se interno u sustavu zbog bržeg dohvata indeksnih struktura prilikom filtriranja te je kao takav pomoćni atribut objekta. Uvjet pretplate uz geometriju i klasu mjernog tipa predstavlja treći uvjet filtriranja. Ovdje je naglasak na određivanju specifičnosti objavljene vrijednosti koje moraju biti zadovoljene. Sam uvjet je definiran klasom `SubscriptionCondition` koja je apstraktna klasa i nudi metode za validaciju vrijednosti s postavljenim uvjetom. Za svaki tip podatka unutar portala – numerički, tekstualni i enumeraciju, definirana je zasebna klasa uvjeta koja nudi pretplatniku različitu mogućnost kriterija ovisno o tipu podatka. Tako korisnik za numeričku vrijednost može definirati prihvatljivi raspon vrijednosti koji se sastoji od minimalne i maksimalne dopuštene vrijednosti i granica uključenosti istih. Za tekstualni tip korisnik može navesti ključne riječi koje ako se pojave u vrijednosti tekstualne objave uvjet je zadovoljen. Enumeracija pak omogućava definiranje skupa preferiranih vrijednosti enumeracija za kojima pretplatnik izražava interes. Kako bi pretplata bila valjana definirani uvjet i njegov tip podataka moraju odgovarati mjernom tipu podataka za koji se pretplata definira. Također vrijednosti koje korisnik unosi kao uvjet, unosi ih u navedenoj preferiranoj mjernoj jedinici te se kao takve i tretiraju.

Obavezni atributi u objektu pretplate su uz geometriju i tip objekta još i ime mjernog tipa, ime preferirane mjerne jedinice i prostorni odnos. Nužno je da su

imena predanog tipa i mjerne jedinice postojeći unutar sustava. Sam uvjet pretplate nije obavezan dio i pretplatnik se može pretplatiti isključivo na mjerni tip i neku geometriju sa željenim prostornim odnosom.



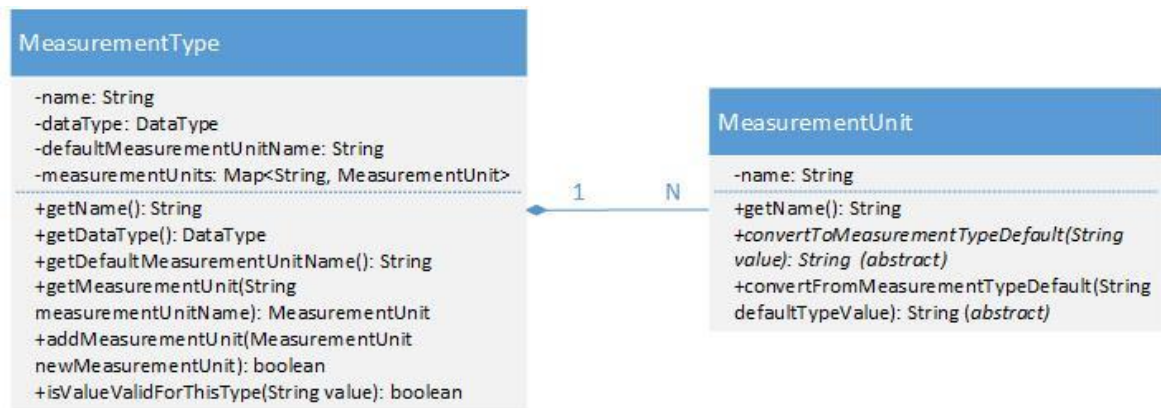
Slika 6 Dijagram klase Subscription koja definira pretplatu u sustavu

2.4 Mjerni tipovi podataka i mjerne jedinice

Mjerni tipovi i mjerne jedinice su obavezni atributi svake objave i pretplate. Mjerni tip predstavlja svojevrsnu logičku grupu, odnosno klasu mjerenja koja je moguća unutar sustava. Primjeri mogućih mjernih tipova su temperatura, pritisak, vlaga, novosti, indeks zračenja itd.. Objava kao i pretplata mogu sadržavati samo jedan mjerni tip. Kod objave navedeni tip definira na što se točno odnosi vrijednost objave (što je mjereno) dok kod pretplate određuje za koji mjerni tip odnosno skupinu objava je pretplatnik zainteresiran. Svaki mjerni tip ima definiranu jednu ili više mjernih jedinica. Mjerna jedinica je dogovorena vrijednost fizikalne veličine s kojom se izražava vrijednost izmjerene objave za pojedini mjerni tip. Mjerna jedinica za temperaturu tako može biti neka od poznatih skala: Celzijev stupanj, Kelvin ili pak Fahrenheit. Svaki mjerni tip ima definiranu standardnu mjernu jedinicu s kojom se obavljaju usporedbe unutar sustava objavi-pretplati prilikom filtriranja te svaka definirana mjerna jedinica mora ponuditi metodu kojom se njena vrijednost preračunava u vrijednost standardne mjerne jedinice i nazad.

Mjerni tip je u sustavu predstavljen klasom `MeasurementType`. Mjerni tip definira vlastito ime koje mora biti jedinstveno u sustavu, tip podatka, standardnu mjernu jedinicu i sve mjerne jedinice koje pripadaju tom mjernom tipu. Ime mjernog tipa mora biti jedinstveno jer služi kao indikator mjernog tipa u sustavu te se i unutar objave i pretplate mjerni tip definira isključivo svojim imenom. Nadalje, mjernom tipu je definiran tip podataka (enumeracija `DataType` unutar sustava) koja određuje da li se radi o tipu čije se mjerne jedinice izražavaju preko numeričkog tipa, tekstualnog ili pak enumeracije. Tip podataka je definiran kako bi sustav na jednostavan način mogao provjeriti da li je vrijednost objave koja dolazi u sustav valjana za određeni mjerni tip kao i da li je uvjet nove pretplate u skladu sa zadanim mjernim tipom. Mjerni tip tako nudi metodu s kojom validira zaprimljenu vrijednost objave i koja standardno koristi provjeru koju definira tip podataka, no za potrebu pojedinog mjernog tipa ova metoda se može nadjačati. Posljednje, mjerni tip definira sve mjerne jedinice kao i standardnu mjernu jedinicu. Sve mjerne jedinice su zbog jednostavnosti pristupa njima organizirane u mapu ime mjerne jedinice – mjerna jedinica te je moguće dohvatiti pojedinu jedinicu za neki mjerni tip. Standardna mjerna jedinica je definirana mjerna jedinica čije se ime određuje u konstruktoru mjernog tipa a određuje mjernu jedinicu u koju se sve vrijednosti unutar sustava preračunavaju i spremaju zbog brže usporedbe prilikom filtriranja.

Mjerna jedinica se definira kroz klasu `MeasurementUnit`. Kao i kod mjernog tipa mjerna jedinica se definira isključivo imenom koje mora biti jedinstveno unutar pojedinog mjernog tipa. Jedini atribut mjerne jedinice je tako njeno ime koje daje semantičko značenje vrijednosti objave i time korisniku daje do znanja što zapravo dobivena vrijednost znači. Uz ime mjerne jedinice, glavne metode koje svaka mjerna jedinica mora ponuditi su metoda pretvorbe vrijednosti u standardnu mjernu jedinicu mjernog tipa kojem jedinica pripada kao i pretvorba iz standardne mjerne jedinice u tu mjernu jedinicu. Funkcionalnost pretvorbe je važna jer kod dolaska nove objave u sustav, prije filtriranja potrebno je pretvoriti vrijednost objave u standardnu mjernu jedinicu kako bi se mogla usporediti s uvjetima pretplate koji su isto tako pretvoreni u standardnu mjernu jedinicu. Pretvorba iz standardne jedinice koristi se kod isporuke objave pretplati kako bi se vrijednost objave mogla pretvoriti u preferiranu mjernu jedinicu pretplatnika.



Slika 7 Dijagram klasa *MeasurementType* i *MeasurementUnit* kojima se u sustavu modelira mjerni tip i mjerna jedinica

Definiranje svih postojećih mjernih tipova i pripadnih mjernih jedinica obavlja se u klasi *AvailableMeasurementTypes*. U navedenoj klasi prilikom pokretanja sustava učitaju se svi definirani tipovi te se kroz statičke metode korisnicima nudi pristup svim definiranim mjernim tipovima i jedinicama. Arhitektura je takva da ako se želi definirati novi tip potrebno ga je dodati u navedenu klasu i ponovno pokrenuti sustav. Razlog tome je što je glavna namjena ovog rada testiranje učinkovitosti filtriranja u definiranom sustavu a u budućoj arhitekturi sustava potencijalno bolji i fleksibilniji pristup bi bio mjerne tipove i jedinice pohraniti u bazu podataka iz koje bi se dinamički učitavali na zahtjev. Uz definiranje samih mjernih tipova, klasa određuje i indekse mjernih tipova gdje za svaki definirani mjerni tip određuje broj unutar polja indeksa koji će sustav filtriranja koristiti za brzi pristup (detaljnije u poglavlju 3). Prilikom definiranja objava od strane senzora odnosno pretplata od strane korisnika nužno je da se dohvate dostupni mjerni tipovi i jedinice te da korisnici navedu stvarno definirano ime jer u protivnom pretplata odnosno objava koju definiraju neće biti valjana.

3. Arhitektura sustava

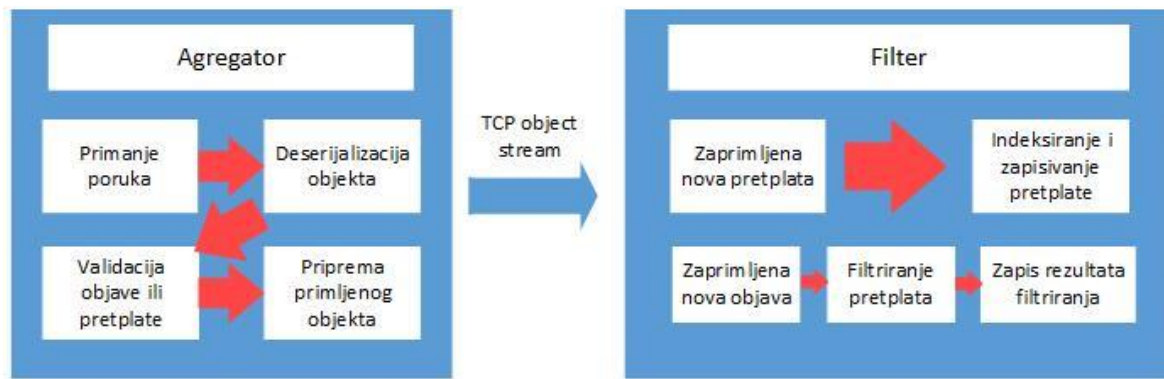
3.1 Dizajn sustava

Razvijeni sustav objavi-pretplati mora omogućiti obradu objava i pretplata. Korisnici serijaliziraju objave i pretplate u GeoJSON format (definiran u potpoglavlju 2.1.2) te ih šalju centralnom sustavu na obradu. Obrada se sastoji od deserijalizacije i provjere valjanosti korisničkih zahtjeva te njihovog izvršavanja u slučaju valjanosti. Prilikom dolaska nove valjane pretplate potrebno je obaviti pripremu pretplate kako bi se proces filtriranja što učinkovitije mogao izvesti te pohraniti pretplatu na disk kako bi se u slučaju prestanka rada sustava zaprimljene pretplate očuvale i mogle ponovno učitati. Kod dolaska valjane objave potrebno je profilirati sve pretplate unutar sustava te listu identifikatora tih pretplata zajedno sa samom objavom pohraniti na disk kao konačni rezultat koji kasnije sustav isporuke može obraditi i proslijediti objavu u željenom formatu do pretplatnika.

Prilikom prvog dizajna sustava razvijen je osnovni, najjednostavniji model centralnog sustava i njegovog filtera. Za komunikaciju s klijentom odabran je protokol TCP te je nakon što se primi i obradi objava ili pretplata od korisnika u GeoJSON formatu server davao odgovor o uspješnosti obrade zahtjeva. Sama obrada se radila u jednoj dretvi te je sustav slijedno obrađivao zahtjev po zahtjev a to je proces koji uključuje deserijalizaciju poruke, validaciju poruke, filtriranje svih pretplata ako je riječ o objavi odnosno dodavanje nove pretplate ako se radi o pretplati. Samo filtriranje se radilo slijedno u procesu provjeravanja svake zaprimljene objave sa svim pretplatama u sustavu. U navedenoj arhitekturi postoji nekoliko problema. Proces filtriranja u kojem se slijedno provjerava sve pretplate je neučinkovit i sam sustav bi postao neupotrebljiv jako brzo s porastom broja pretplatnika. Drugi problem je vidljiv u činjenici da se zaprimljeni zahtjevi nepotrebno obrađuju slijedno. Svaki zahtjev, bio on nova objava ili pretplata je u obradi neovisan o drugima neobrađenim zahtjevima te se obrada može jednostavno paralelizirati. Zadrje, korišteni TCP protokol i slanje odgovora klijentima uvodi dodatnu neželjenu latenciju. Za svaki korisnički zahtjev potrebno je stvoriti novu TCP vezu, što uvodi dodatno kašnjenje. Uz to, kako je riječ o arhitekturi objavi-pretplati pozitivan odgovor korisniku, u slučaju slanja objave,

znači samo da je objava uspješno zaprimljena u centralni sustav. Ova informacija nije garancija korisniku da će objava biti uspješno dostavljena pretplatnicima. S druge strane, uspostava veze i odgovor uvodi neželjeno kašnjenje pa ako ne donosi dovoljnu vrijednost predstavlja nepotrebnii dio arhitekture.

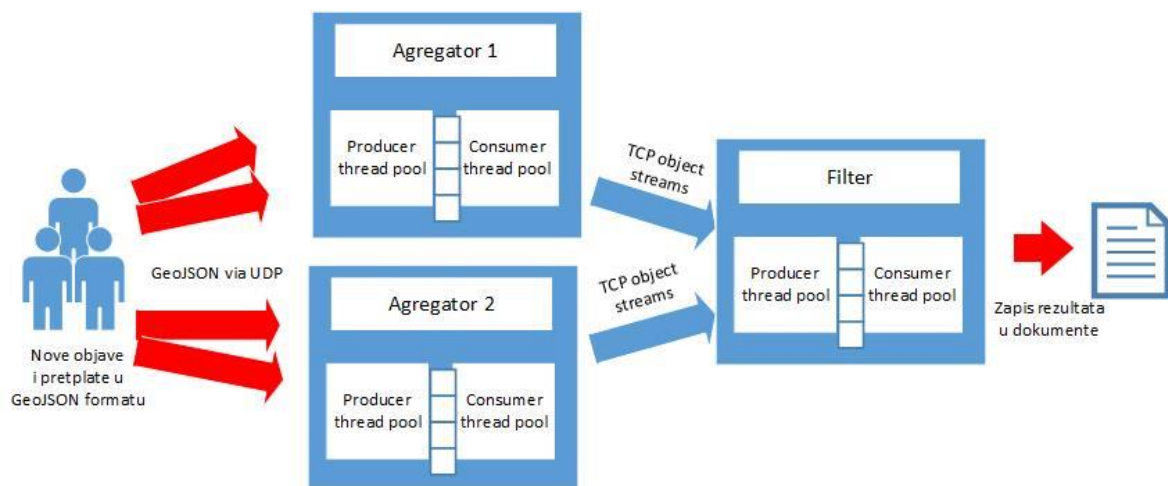
Kako bi se otklonili navedeni problemi i minimiziralo vrijeme obrade osmišljena je nova arhitektura koja je odabrana i kao konačna. Za slanje zahtjeva od klijenta do centralnog sustava u obliku novih objava i pretplata koristi se UDP protokol bez slanja potvrde. Sadržaj objava i pretplata šalje se u serijaliziranom GeoJSON obliku te se kod primanja deserijalizira i obrađuje. U obradi je iskorištena činjenica da se novi zahtjevi mogu paralelno obrađivati te se obrađuju u dretvama radnicima čiji broj korisnik sustava može podesiti. Kako bi se obrada dodatno ubrzala razdvojena je u dvije cjeline koje obrađuju dvije komponente – agregator i filter. Komponente agregatora i filtera te njihove osnovne funkcionalnosti prikazane su na slici 8. Zadaća agregatora je primati zahtjeve te ih deserijalizirati, validirati i pripremiti za filtriranje. Ovo obavlja u nizu neovisnih dretva radnika i po obradi šalje filteru. Filter pri dolasku nove objave filtrira pretplate i sprema rezultat. U slučaju pretplate komponenta filtra priprema pretplatu za dolazak buduće objave. Proces filtriranja je poboljšan. Za razliku od slijednog prolaska svim pretplatama korištene su indeksne strukture. Pretplate se indeksiraju s geoprostornim indeksima po definiranoj geometriji te se kod nailaska nove objave gleda uz pomoć indeksa koje sve pretplate geometrija objave presijeca a zatim se za njih slijedno provjerava valjanost definiranog prostornog odnosa. Filtriranje se obavlja za svaki mjerni tip zasebno, odnosno za svaki mjerni tip se definira zaseban indeks. U ovom slučaju mjerni tipovi predstavljaju klase objava te se pretplate time efektivno pretplaćuju na klasu – na pojedini mjerni tip. Uz spomenuto geoprostorno indeksiranje korištene su i odgovarajuće indeksne strukture za svaki od mogućih uvjeta pretplate. Tako numerički tip uvjeta pretplate koji sadrži numeričke raspone koristi strukturu Interval tree za efikasno indeksiranje, dok tekstualni tip uvjeta pretplate koji sadrži ključne riječi koristi Apache Lucene knjižnicu s njenim mogućnostima brze pretrage teksta za takve slučajeve. Zbog evaluacije sustava zadržana je i linearna implementacija kao i mogućnost odabira različitih indeksnih struktura kako bi se odredilo koja arhitektura kao i koji proces filtriranja su najbolji za rad ovakvog sustava.



Slika 8 Prikaz osnovnih komponenta arhitekture agregatora i filtra s osnovnim funkcionalnostima

Proces obrade novog zahtjeva u sustavu započinje s definiranjem objekta pretplate odnosno objave od strane korisnika sustava. Objekti klase Subscription i Publication opisuju pretplate i objave te korisnik (senzor ili pretplatnik) definira sve potrebne attribute u objektima. Nakon toga se definirana objava odnosno pretplata serijalizira u poruku GeoJSON tipa. Kako su i objave i pretplate geoprostorni objekti nude odgovarajuću metodu serijalizacije u GeoJSON tip te se poruka u tom formatu preko UDP protokola šalje centralnom sustavu. Komponenta koja u centralnom sustavu prima poslano poruke je agregator. Nakon primanja poruke agregator ju delegira jednoj od dretva proizvođača (engl. producer thread) koja zatim obavlja deserijalizaciju, validaciju i pripremu objekta za daljnju obradu. Nakon završetka obrade dretva proizvođač stavlja objekt u red čekanja iz kojeg pak jedna od dretva potrošača (engl. consumer thread) čita objekt i šalje ga komponenti filtra. Slanje objekata od agregatora do filtra obavlja se preko TCP protokola te na strani filtra jedna od dretvi proizvođača u komponenti filtera prima zahtjeve od agregatora. Objekti se prenose u toku podataka te se jednom ostvarena veza između potrošača agregatora i proizvođača filtera održava sve do prestanka veze gašenjem neke od komponentata ili greške. Na kraju, dretva potrošač na strani filtera čita objekte zaprimljene od agregatora koji su unaprijed pripremljeni za obradu te ih obrađuje i zapisuje u datoteku. U trenutnoj arhitekturi proces završava zapisom u datoteku, no ideja je da se obrada dalje nastavlja u komponenti koja bi vršila isporuku do pretplatnika.

U razvijenoj arhitekturi izvođenje jedne instance komponente agregatora i filtera na istom serveru predstavlja dodatno nepotrebno opterećenje jer u tom slučaju proces slanja objekata TCP protokolom između komponenta nema smisla i mogao se riješiti direktnom komunikacijom. Ideja ove arhitekture je u tome da je moguće pokrenuti više instanca agregatora za jednu instancu komponente filtra te tako kritična točka obrade ostaje filtriranje. U integriranju ovog sustava u veću cjelinu tako postoji mogućnost postavljanja reverznog proxy poslužitelja ispred skupa instanci agregatora na koje se zatim delegira posao obrade zahtjeva sukladno njihovoj opterećenosti a zatim svi oni šalju pripremljene i validne objekte centralnoj kritičnoj komponenti filtra. Upravo iz tog razloga i odabrane arhitekture vidljiv je razlog naglaska ovog rada na odabiru efikasnog filtriranja kako bi se time poboljšalo vrijeme izvođenja cijelog sustava.



Slika 9 Proces obrade korisničkog zahtjeva od slanja do rezultata

3.2 Agregator

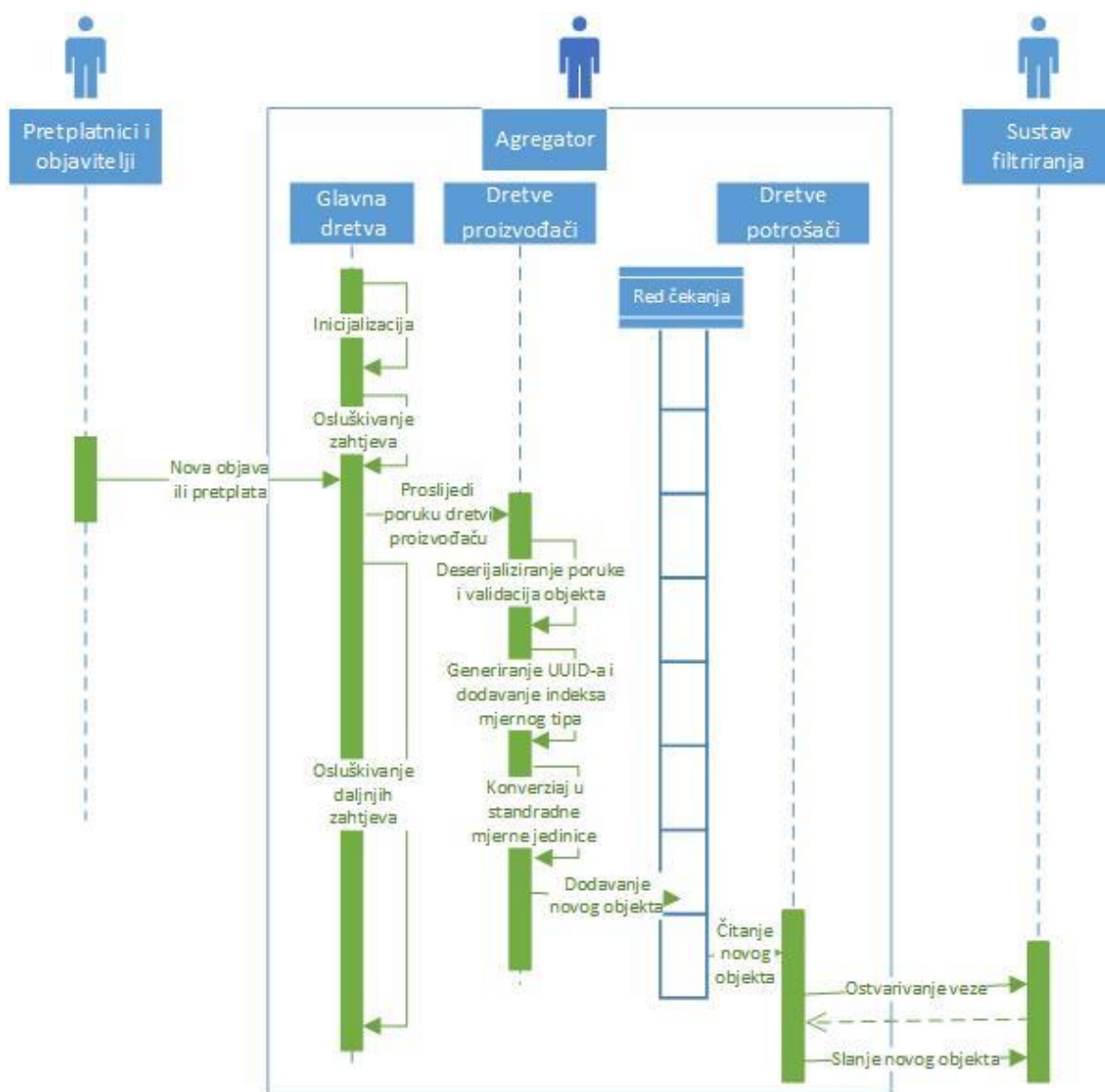
Zadaća agregatora u centralnom sustavu je primiti serijalizirane objave i pretplate korisnika sustava te ih pripremiti za daljnju obradu. Sustav može imati jednu ili više instanci agregatora. Nakon pripreme objekata agregator ih prosljeđuje sustavu filtriranja čija je jedina zadaća obavljati efikasno filtriranje. Agregator se tako mora pobrinuti da je zaprimljeni zahtjev valjan te da obavi sve potrebne konverzije u standardne mjerne jedinice kao i dodavanje indeksa

mjernog tipa i univerzalnog identifikatora, kako bi se sustav filtriranja rasteretio od tog posla i mogao optimizirati filtriranje.

Prilikom pokretanja agregator učitava iz konfiguracijske datoteke IP adresu i port na kojem sluša zahtjeve, kao i adresu i port sustava filtriranja na koji će prosljeđivati obrađene objekte. Također u konfiguracijskoj datoteci se može podesiti željeni broj dretvi proizvođača kao i potrošača te je za optimalan rad preporučljivo da je suma svih dretva potrošača svih agregatora jednaka broju dretva proizvođača sustava filtriranja jer te dretve međusobno otvaraju TCP konekciju preko koje agregatori šalju sustavu filtriranja pripremljene objekte. Na kraju iz konfiguracijske datoteke učitava se i veličina paketa koji agregator prima od korisnika a ta veličina mora biti jednaka maksimalnoj veličini serijaliziranog objekta koji korisnik može poslati sustavu. Navedeno ovisi o zahtjevima sustava i maksimalnoj veličini vrijednosti koje senzori isporučuju pa ovisi o primjeni sustava.

Glavna dretva servera, nakon inicijalizacije reda čekanja i bazena dretvi proizvođača i potrošača, osluškuje korisničke zahtjeve na definiranoj adresi i portu te preko UDP protokola prima pakete. Nakon primanja paketa predaje ga slobodnoj dretvi proizvođaču te osluškuje daljnje zahtjeve korisnika. Dretva proizvođač iščita primljenu poruku te ako nije u GeoJSON formatu odbacuje ju. Ako je format valjan provjerava definirani tip objekta koji definira da li je riječ o pretplati ili objavi. Kako bi se objekt instancirao primljena poruka se predaje konstruktoru objave odnosno pretplate, a unutar samog konstruktora provjerava se prisutnost i valjanost svih obaveznih atributa. Također u konstruktorima se provjerava da li je definirano ime mjernog tipa i pripadne jedinice valjano i ako nije odustaje se od daljnje obrade jer zahtjev nije valjan. Nakon deserijalizacije objektu se generira i postavlja novi univerzalni jedinstveni identifikator te indeks za navedeni mjerni tip kako bi filter brzo mogao preko indeksa dohvatiti indeksnu strukturu za definirani mjerni tip. Na kraju dretva proizvođač, ako je riječ o objavi, mijenja vrijednost objave iz mjerne jedinice u kojoj je zaprimljena u standardnu mjernu jedinicu preko metode za konverziju koju mjerna jedinica definira. Ako se pak radi o pretplati tada se također radi konverzija uvjeta pretplate u standardnu jedinicu preko metode koju uvjet pretplate definira. Nakon navedenog procesa objekt je spreman za sustav filtriranja te ga agregatorova dretva proizvođač postavlja u red čekanja unutar agregatora. Dretve potrošači unutar agregatora

čitaju pripremljene objekte iz reda čekanja te ih šalju sustavu filtriranja preko TCP veze koja se održava otvorenom dokle god su agregator i sustav filtriranja aktivni kako bi se smanjio utrošak vremena prilikom uspostavljanja veze.



Slika 10 Sekvencijski dijagram koji prikazuje generalni proces rada agregatora

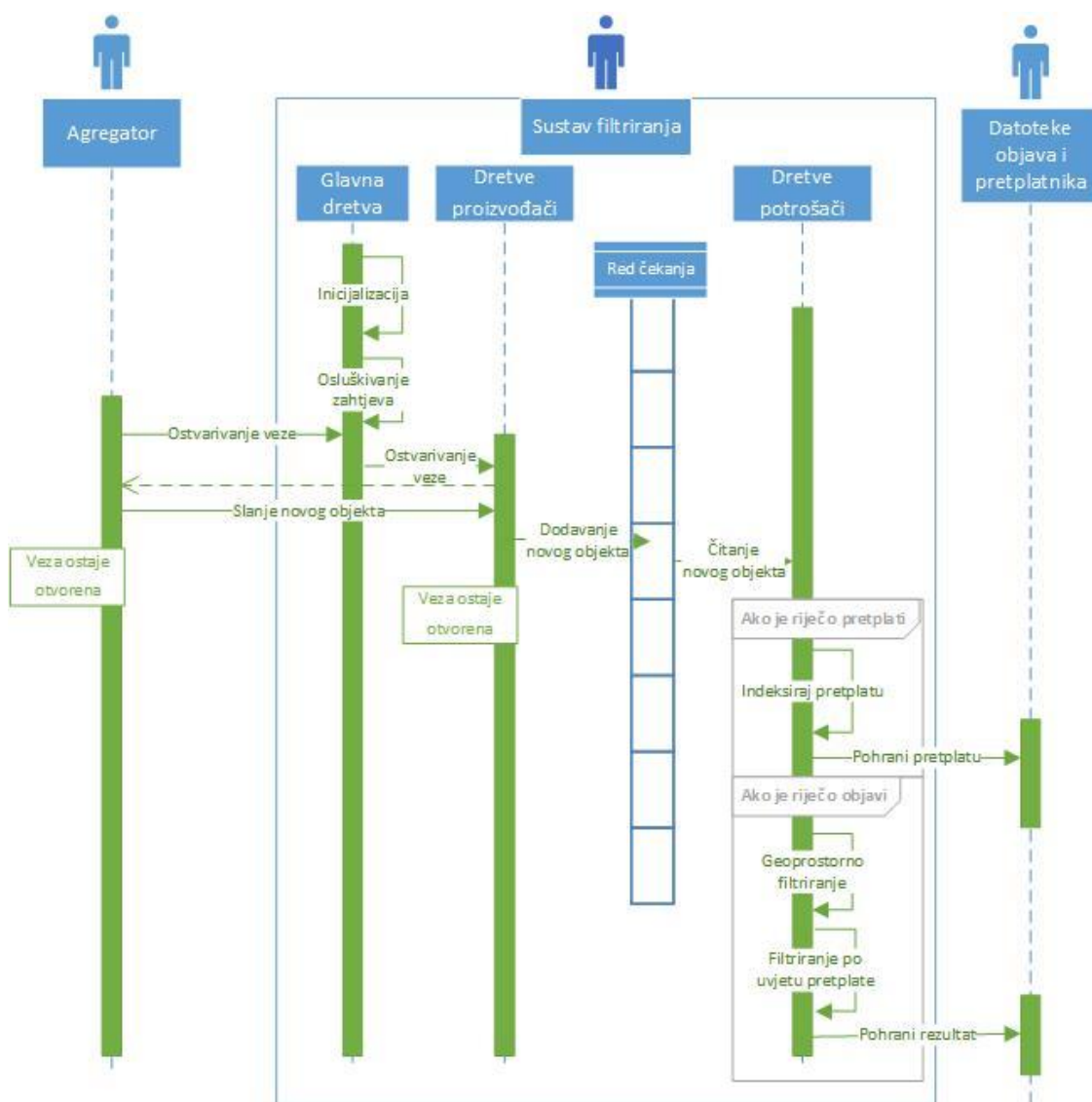
3.3 Sustav filtriranja

Sustav filtriranja od agregatora dobiva pripremljene objekte objava i pretplata te je njegova zadaća kod dolaska objave što brže obaviti filtriranje pretplata a kod dolaska pretplate pohraniti ju i indeksirati. Kako bi se ispitalo koji je optimalan način filtriranja za dani sustav, u filteru je omogućena primjena različitih indeksnih struktura kao i jednostavno dodavanje novih s ciljem pronalaska strategije koju bi

bilo najbolje koristiti za definirani oblik obavijesti i pretplata. Iz tog razloga oblik filtriranja je konfigurabilan od strane korisnika sustava te ovisno o njegovim preferencijama može biti promijenjen i prilagođen.

Kod pokretanja sustav filtriranja iz konfiguracijske datoteke pročita IP adresu i port na kojima sluša zahtjeve agregatora za spajanje i slanje pripremljenih objekata. Nadalje, filter iz konfiguracijske datoteke čita putanje do datoteke u koju upisuje rezultate obrade objava i druge datoteke u koju upisuje sve prisutne pretplate u sustavu. U konfiguracijskoj datoteci je kao i za agregator definiran željeni broj dretvi proizvođača i potrošača koji će se koristiti u filteru. Uz to iz konfiguracijske datoteke se čita putanja do klase koja sadrži željenu implementaciju strategije geoprostornog indeksiranja. Strategija geoprostornog indeksiranja je bilo koja klasa koja nasljeđuje apstraktnu klasu `GeospatialIndexingStrategy` i nudi metode za dodavanje nove pretplate u indeks, micanje pretplate iz indeksa te pronalazak svih pretplata koje se potencijalno presijecaju s danom geometrijom. Korištene geoprostorne strategije indeksiranja su detaljnije objašnjene u poglavlju 4.1 a njihova glavna namjena je efikasno indeksirati geometrije svih pretplata kako bi se po dolasku nove objave uz što manje utroška vremena dohvatile sve pretplate koje su u presjeku s novom objavom. Rezultat upita nad indeksom može vraćati lažne pozitivne rezultate te se vraćene pretplate uvijek naknadno slijedno prolaze i ispituje se dodatni uvjet prostornog odnosa definiran u pretplati, no cilj indeksne strategije je smanjiti skup pretplata koje se zatim slijedno prolaze. Posljednja stavka koja se čita iz konfiguracijske datoteke je da li je aktivno indeksiranje po uvjetima pretplate. Ako je indeksiranje aktivno tada se uz geoprostorni indeks grade i odgovarajući indeksi za uvjete pretplate koji pri nailasku objave vraćaju sve pretplate čiji je uvjet pretplate zadovoljen. U tom slučaju konačni rezultat je presjek skupa pretplata koje su zadovoljile uvjet prostornog odnosa i skupa pretplata čiji je uvjet pretplate zadovoljen, jer je samo takvim pretplatama potrebno isporučiti objavu. Ako je pak indeksiranje isključeno uvjeti pretplata se prolaze slijedno nakon određivanja skupa pretplata koje zadovoljavaju geoprostornu komponentu. Nakon učitavanja konfiguracijske datoteke, iz datoteke u koju se radi pohrana pretplata potrebno je učitati sve pretplate te ovisno o postavkama indeksiranja i indeksirati sve pretplate. Kada se proces indeksiranja obavi sustav filtriranja je spreman za rad.

Nakon pokretanja i inicijalizacije sustav filtriranja očekuje zahtjeve od agregatora na definiran port i adresu. Kada zaprimi zahtjev od agregatora sustav filtriranja iz glavne dretve predaje klijentsku priključnicu jednoj od dretva proizvođača koja je dostupna. Dretva proizvođač u filteru uspostavlja vezu s agregatorom preko TCP protokola te čita objekte preko ulaznog toka u kojem agregator šalje pripremljene objekte. Zadaća dretvi proizvođača u filteru je tako da drže otvorenu vezu s agregatorom i nakon čitanja objekta postave ga u red čekanja unutar sustava filtriranja.



Slika 11 Sekvencijski dijagram koji prikazuje generalni proces rada filtera

Dretve potrošači su posljednji dio sustava koji čita objekte iz reda čekanja i obrađuje ih. Proces obrade ovisi o tome da li je u filter došla nova pretplata ili

objava. U slučaju pretplate potrebno ju je serijaliziranu zapisati u datoteku u kojoj se nalaze sve pretplate kako bi se u slučaju zaustavljanja sustava i njegovog ponovnog pokretanja sve pretplate mogle učitati. Uz zapis pretplate potrebno ju je i indeksirati kako bi se kod dolaska objava uzela u obzir. Geoprostorni indeks je uvijek prisutan te je zasebni geoprostorni indeks definiran za svaki mjerni tip u sustavu. Ovime se osigurava brže filtriranje jer mjerni tip predstavlja atribut objave po kojoj se jednostavno klasificiraju i tako nije potrebno razmatrati pretplate drugih mjernih tipova pri dolasku objave određenog tipa. Broj mjernih tipova u sustavu je poznat i ograničen pa veći broj indeksnih struktura ne predstavlja problem sa stajališta memorijskog opterećenja. Uz dodavanje pretplate u geoprostorni indeks, ako je aktivno indeksiranje po uvjetu pretplate, tada je također ovisno o mjernom tipu potrebno dodati pretplatu u odgovarajući indeks po uvjetu pretplate. Kada u filter dođe objava potrebno je obaviti filtriranje pretplata te zatim pohraniti serijaliziranu objavu skupa s listom svih pronađenih pretplata u odgovarajuću datoteku. Početak filtriranja je uvijek jednak i započinje dohvatom svih pretplata koje se potencijalno geoprostorno križaju s novom objavom. To se obavlja preko geoprostornog indeksa koji je definiran za mjerni tipa jednak kao objava. Svaki mjerni tip ima definiran vlastiti indeks čime se osigurava manji broj pretplata u indeksu i njegova manja dubina te veća brzina dohvata kao i manji skup dohvaćenih pretplata. Pronalazak odgovarajuće indeksne strukture ovisno o mjernom tipu brzo se obavlja jer agregator postavlja indeks mjernog tipa koji je ustvari pozicija tog mjernog tipa u polju geoprostornih indeksa za sve mjerne tipove. Nakon određivanja geoprostornih presjeka preko indeksa potrebno je sve dobivene pretplate slijedno proći te usporediti geometriju pretplate, objave i definiran željeni prostorni odnos unutar pretplate. Kada je određen taj skup pretplata potrebno ga je ili slijedno proći ako nije aktivno indeksiranje po uvjetu pretplate ili pak dohvatiti sve pretplate iz indeksa po uvjetu. Ako je indeksiranje po uvjetu pretplate aktivno, nakon dohvata pretplata konačan rezultat je presjek skupova pretplata koje su zadovoljile uvjet i onih koje su zadovoljile geoprostorni uvjet. Ako indeksiranje nije aktivno, nakon slijednog prolaska i provjere uvjeta dobivaju se pretplate koje je potrebno obavijestiti jer su im zadovoljeni svi parametri. Na kraju se objava s filtriranim pretplatama zapisuje u datoteku s rezultatima te je pronađene pretplate potrebno obavijestiti i proslijediti im objavu što je izvan dosega zadatka ovog rada.

4. Strategije indeksiranja

4.1 Geoprostorne strategije indeksiranja

U procesu filtriranja u razvijenom sustavu objavi-pretplati nužno je za geometriju novopridošle obavijesti odrediti sve pretplate čija su geometrija i prostorni uvjet zadovoljeni s obzirom na obavijest. Ovaj proces je moguće obaviti linearnim prolaskom kroz sve pretplate i provjerom geoprostornog uvjeta no takav sustav nije skalabilan i s porastom broja preplatnika dolazi do zasićenja i odziv na novu objavu postaje prevelik. Takav sustav ne nudi učinkovito filtriranje te je pronalazak odgovarajućih preplata potrebno riješiti na drugačiji način.

Kako bi se izbjeglo slijedno provjeravanje svake preplate geometrijska područja preplata se organiziraju u strukture geoprostornih indeksa. Geoprostorni indeks je struktura podataka koja organizira geometrijske podatke tako da nad njima omogućava jednostavno i efikasno postavljanje geoprostornih upita. Geoprostorni upit se sastoji od geometrije za koju se radi pretraga i željenog odnosa geometrija koji se pretražuje. Rezultat takvog upita su sve geometrije unutar geoprostornog indeksa koje zadovoljavaju parametre upita. Geometrijski tipovi koji su podržani u indeksnim strukturama i upitima su standardni tipovi koji uključuju točku, liniju i poligon ili pak skupove točaka, linija i poligona. Odnosi geometrija koji se mogu definirati su različiti i uključuju: udaljenost geometrija, jednakost geometrija, nepresijecanje, presijecanje, dodirivanje, prekrivanje, sadržavanje, duljinu i površinu geometrija. Podržanost upita za pojedine odnose kao i efikasnost takvog upita nad indeksnom strukturom ovise o izvedbi pojedine indeksne strukture i njenim mogućnostima.

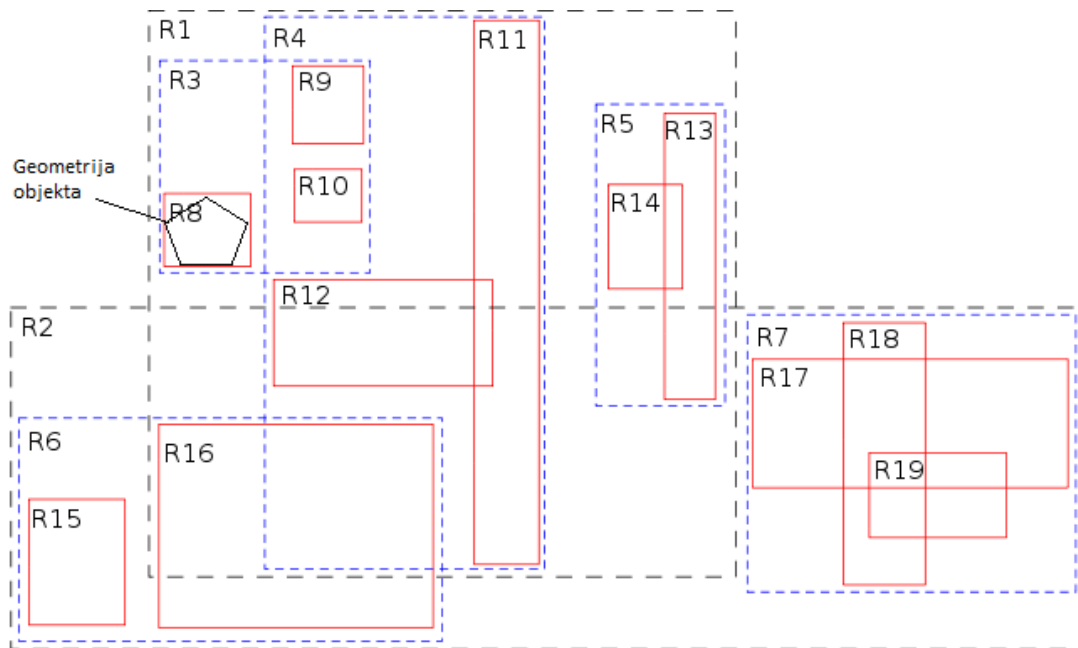
Za potrebe sustava razvijenog u radu zahtjevi za indeksnu strukturu su mogućnost dodavanja nove i micanje stare geometrije iz indeksne strukture te postavljanje upita presijecanja geometrije sa skupom indeksiranih geometrija. Iako su neki od prostornih odnosa koji se koriste unutar filtra poput jednakosti, dodirivanja i prekrivanja podržani u upitima nekih indeksnih struktura, ipak nisu u tolikoj mjeri podržani kao upit presjeka. Pošto su svi preostali prostorni odnosi samo podskup upita presjeka (dodirivanje, prekrivanje itd.) nad indeksnom strukturom se postavlja upit presjeka koji vrati manji podskup preplata koje se

zatim slijedno prolaze i provjerava im se definirani prostorni odnos [4]. Kako se sve pretplate koje su rezultat upita slijedno prolaze u obzir dolaze i indeksne strukture koje za upit vraćaju lažne pozitivne rezultate koji se naknadno moraju provjeriti. Važno je samo da indeks u što kraćem vremenu smanji skup zadovoljenih pretplata koje se presijecaju s objavom kako bi se smanjilo vrijeme slijednog prolaska i povećala skalabilnost cjelokupnog sustava. U okviru rada testirane su i korištene dvije strategije geoprostornog indeksiranja: R-tree i Quadtree koje su opisane u nastavku.

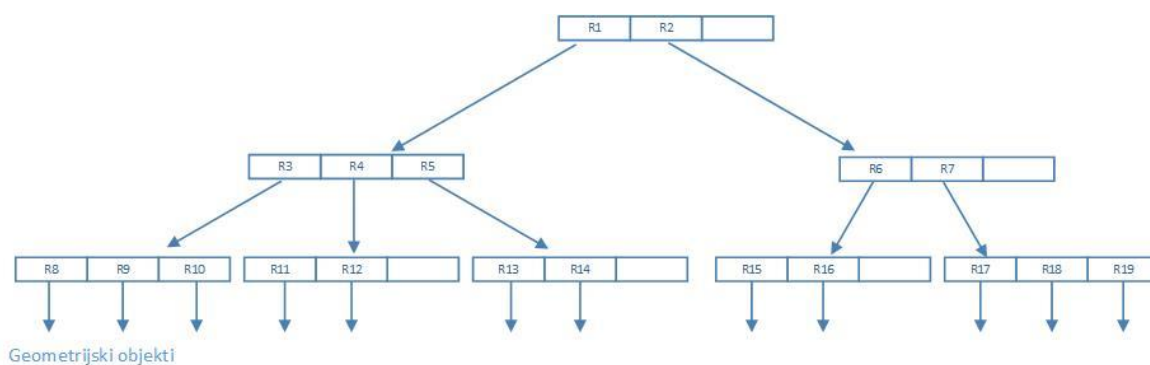
4.1.1 R-tree

R-tree je struktura podataka koja omogućava indeksiranje geoprostornih podataka (točke, linije, poligoni) i postavljanje prostornih upita nad kreiranom strukturom. Prostorni upiti uključuju geometrijski podatak i dodatni uvjet pretrage poput presjeka, najbližih geometrija, geometrija koje su udaljene za neku maksimalnu udaljenost s obzirom na predanu geometriju itd.. R-tree navedeno omogućava organiziranjem indeksiranih podataka s obzirom na njihovu blizinu u skupine koje se zatim povezuju s drugim bliskim skupinama u višim razinama stablaste strukture [6].

Unutar R-tree strukture svaka geometrija se opisuje minimalnim pravokutnikom kojim geometrija može biti u potpunosti obuhvaćena (engl. minimum bounding rectangle). Zatim ovisno o parametru strukture M koji označava broj unosa u pojedinom čvoru stabla, M pravokutnika koji opisuju geometrije se grupiraju zajedno i ponovno se na višoj razini stabla opisuju pravokutnikom koji obuhvaća svih M pravokutnika niže razine [6]. Određivanje minimalnih pravokutnika i grupiranje je prikazano na slici 12. Navedeni proces se ponavlja sve dok se ne dođe do najviše razine u kojoj su sve geometrije nižih razina obuhvaćene s M pravokutnika najviše razine. Opisanim procesom se dobiva struktura prikazana na slici 13. Dobivena struktura je balansirano stablo u kojem su svi listovi na jednakoj razini. R-tree tako omogućava postavljanje upita pretrage u trajanju $O(\log_M n)$ gdje je n ukupni broj geometrija u stablu, a M broj članova pojedinog čvora, odnosno broj geometrija koje se zajedno grupiraju [6].



Slika 12 Prikaz određivanja minimalnih pravokutnika i grupiranja u R stablu (Guttman, 1984) [6]



Slika 13 R-tree struktura definirana grupiranjem na slici 12 (Guttman, 1984) [6]

Pretraga po presjeku s predanom geometrijom vrši se tako da se predanoj geometriji odredi minimalni pravokutnik koji ju obuhvaća. Zatim se krenuvši od korijena stabla gleda presjek pravokutnika iz upita i svih pravokutnika u korijenskom čvoru. Ako presjek postoji potrebno je dohvatiti sve pravokutnike niže razine koje pravokutnik više razine obuhvaća i tako za svaki pravokutnik za koji se nađe presjek. Postupak se rekurzivno ponavlja sve do najniže razine u kojoj se dohvaćaju svi zadovoljeni geometrijski objekti najniže razine [6]. Kako su

geometrije opisane minimalnim pravokutnikom te se ne gleda njihov stvarni presjek u rezultatu može postojati određeni broj lažnih pozitivnih rezultata pa je potrebna dodatna provjera kako bi se utvrdila valjanost dobivenih rezultata.

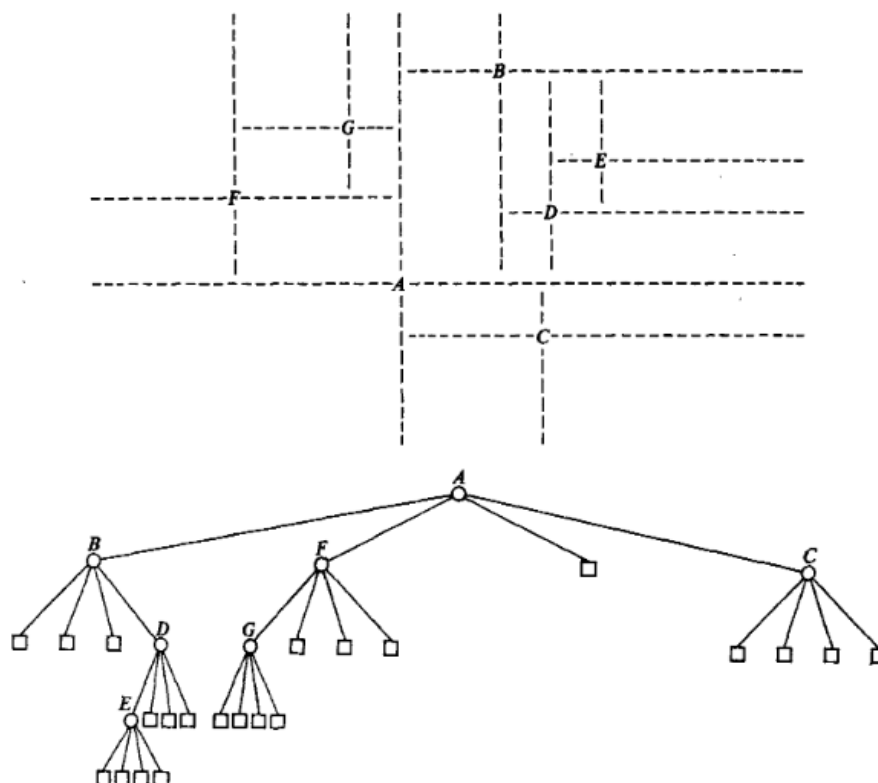
Najzahtjevniji dio je izgradnja R-tree strukture. Zahtjevnost algoritma izgradnje je u tome da se konstruira balansirano stablo u kojem će pravokutnici koji opisuju geometrije obuhvaćati što je manje moguće praznog prostora te se minimalno međusobno preklapati kako bi upiti nad strukturom bili što efikasniji. JTS knjižnica u svojoj izvedbi R-tree strukture koristi STR (engl. Sort-Tile-Recursive) algoritam prilikom izgradnje indeksa kako bi rekurzivno sortirala indeksirane geometrije. Pozitivna strana takvog indeksa je vidljiva u brzini upita koji se može postaviti nad njime, no negativna strana je što zbog izgradnje efikasne strukture, indeks jednom kada je formiran ne omogućava dodavanje novih geometrijskih struktura u njega. Konkretno, u razvijenom sustavu ovo predstavlja problem kod dolaska nove pretplate u sustav koja ne može biti dodana u kreiranu indeksnu strukturu. U implementaciji se problem rješava tako da se novo dodana pretplata dodaje u zasebni set i vraća na svaki upit postavljen nad indeksnom strukturom (ovo nije problem jer upit može vraćati lažne pozitivne rezultate) a kada skup novih pretplata naraste na veličinu pola ukupne indeksne strukture, tada se gradi nova indeksna struktura.

4.1.2 Quadtree

Quadtree je struktura podataka koja definira podjelu i organizaciju dvodimenzionalnog prostora te tako između ostalih primjena omogućava i indeksiranje geoprostornih podataka (točke, linije, poligoni) u strukturu nad kojom se zatim mogu postavljati prostorni upiti [7]. Uz geoprostorno indeksiranje Quadtree struktura pronalazi primjenu u obradi slika, detekciji sudara, aproksimaciji geometrije a u okviru ovog rada najzanimljivija je njena primjena u indeksiranju geometrijskih podataka.

Indeksiranje geometrije u strukturi Quadtree započinje određivanjem minimalnog pravokutnika koji obuhvaća geometriju (engl. minimum bounding rectangle). Pripremljena geometrija dodaje se u strukturu te nakon umetanja geometrija dijeli ukupni prostor po svim osima na 4 dijela – sjeveroistočni, jugoistočni, jugozapadni i sjeverozapadni. Element unutar strukture zbog toga ima

točno 4 djece. Pri umetanju svake sljedeće geometrije određuje se u koji dio prostora koji je određen prethodnim dijeljenjem nova geometrija pripada, pod uvjetom da pripada u cijelosti u neki od prostora te se sukladno tome dodaje u strukturu. U slučaju da pravokutnik pripada u više prostora na koje je trenutna razina podijeljena tada se dodaje u istu razinu. Nakon dodavanja potprostor u koji je geometrija dodana se ponovno dijeli po istom principu na 4 nova potprostora te se proces kod dodavanja rekursivno ponavlja [7]. Elementi u listovima sadrže samo podatke te nemaju djece sve dok se struktura ne proširi novim elementima niže razine. Ovakva struktura omogućava brzo pretraživanje u $O(\log n)$ složenosti zbog stablaste strukture a za razliku od R-tree strukture omogućuje jednostavnije dodavanje novih elemenata i nakon kreiranja indeksa. Pronalazak presjeka korištenjem Quadtree strukture započinje usporedbom na korijenskoj razini te se ako postoji presjek geometrije upita i prostora na pojedinoj razini uzimaju sve geometrije na toj razini a za presjek ponovno provjeravaju 4 područja niže razine [7].



Slika 14 Prikaz izgradnje Quadtree strukture u kojoj je element A korijenski element te dijeli prostor na 4 potpodručja u koja se zatim umeću ostali elementi te rekursivno dijele prostor (Finkel, Bentley, 1974) [7]

4.2 Indeksiranje po uvjetima pretplate

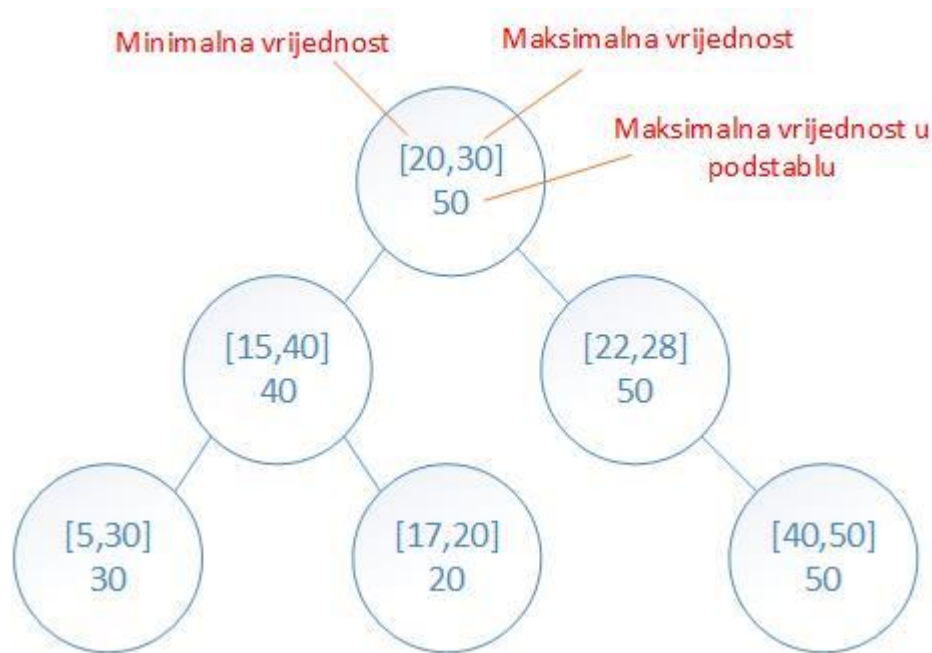
U filteru uz provjeru geometrije i zadovoljenosti prostornog odnosa pretplata koji se učinkovito radi primjenom geoprostornog indeksa, pri dolasku objave potrebno je obaviti i filtriranje po uvjetu pretplate u odnosu na vrijednost objave. Kako bi se obavila provjera po uvjetu postoje dvije mogućnosti koje su implementirane u sustavu. Jedna je nakon određivanja pretplata čiji je definirani geometrijski uvjet zadovoljen slijedno proći sve zadovoljene pretplate i provjeriti definiran uvjet pretplata. Kako skup pretplata koje zadovoljavaju geometrijski presjek u prosječnom slučaju nije velik, slijedni prolazak ne predstavlja veliko kašnjenje u obradi objave. Druga mogućnost je definirati indeksne strukture za svaki od mogućih uvjeta pretplata te kao i kod geometrijskog indeksa, prilikom nailaska objave iz kreirane strukture dohvatiti sve pretplate čiji su uvjeti zadovoljeni. Konačan rezultat pretplata je u slučaju korištenja oba indeksa presjek skupova pretplata kojima su zadovoljeni geometrijski uvjeti odnosno uvjeti pretplate.

Struktura indeksa je proizvoljna te ovisi o uvjetu pretplate i tipu podatka za koji se uvjet definira. Unutar sustava su podržana tri osnovna tipa podatka – numerički, tekstualni i enumeracija. Za ta tri tipa su definirani i odgovarajući uvjeti pretplata te i odgovarajuće indeksne strukture opisane u ovom poglavlju. Zahtjevi svake od indeksnih struktura su da za minimalan utrošak vremena vrati sve pretplate čiji su uvjeti za predanu vrijednost objave zadovoljeni. Također indeksi moraju omogućiti dodavanje novih pretplata i uklanjanje starih. Prilikom definiranja novog tipa podatka potrebno je odrediti odgovarajuću indeksnu strukturu u slučaju da se namjerava koristiti indeksiranje po uvjetu pretplate.

4.2.1 Interval tree

Numerički tip podatka u sustavu podržava decimalne brojeve kao vrijednosti obavijesti. Kako su vrijednosti decimalni brojevi, pretplatniku je u uvjetu pretplate omogućeno definiranje prihvatljivog raspona decimalnih brojeva u vrijednosti pristigle objave. Prihvatljivi raspon se definira s minimalnom i maksimalnom vrijednosti te oznakama da li je minimalna odnosno maksimalna vrijednost uključena u raspon. Sukladno takvoj definiciji uvjeta pretplate potrebno je odabrati

indeksnu strukturu koja će omogućiti brzi dohvat svih pretplata čiji je raspon vrijednosti zadovoljen s obzirom na vrijednost objave. Interval tree je indeksna struktura koja omogućava indeksiranje brojevnih intervala i postavljanje upita presjeka intervala s indeksiranim intervalima. Iako sama vrijednost objave nije interval već isključivo decimalna vrijednost sama decimalna vrijednost se može definirati kao interval s tom minimalnom i maksimalnom vrijednosti i uključenim granicama kako bi se postavio upit nad takvom strukturom.



Slika 15 Prikaz Interval tree strukture

Struktura je jednaka strukturi binarnog stabla pretraživanja u kojoj su vrijednosti čvorova decimalni brojevi te se ovisno o algoritmu broj s većom vrijednosti od trenutnog čvora stavlja u jedno podstablo a onaj s manjom u drugo. Ideja Interval tree strukture je zadržati pozitivno svojstvo binarnog stabla u kojem se operacija pretrage izvodi u trajanju $O(\log n)$ a strukturu iskoristiti za indeksiranje intervala [8]. Kako bi se to osiguralo vrijednost svakog čvora sadrži interval i najveću maksimalnu vrijednost u nekom od intervala desnog podstabla. Prilikom dodavanja intervala u strukturu uspoređuje se minimalna vrijednost intervala koji se dodaje s onim u trenutnom čvoru. Ako je vrijednost manja dodavanje se nastavlja u lijevom podstablu, a ako je vrijednost veća nastavlja se u desnom podstablu. Kada se dođe do najnižeg čvora koji nema vrijednost upisuje se novi

interval te je potrebno osvježiti podatak maksimalne vrijednosti podstabla u svim nadređenim čvorovima u kojima je zbog umetanja moglo doći do promjene [8]. Ovisno o primjeni algoritma stablo i nakon dodavanja odnosno brisanja čvora treba balansirati kako bi se zadržala dobra svojstva pretrage.

Pretraga u kreiranoj strukturi s drugim intervalom (u radu je korišten interval koji predstavlja jedan decimalni broj) za koji se želi provjeriti u koje sve od indeksiranih intervala pripada kreće od korijenskog čvora. Provjerava se postoji li presjek između intervala korijena i uvjeta te ako postoji, element korijenskog čvora je jedan od intervala u presjeku koji se vraća kao rezultat. Ako je maksimalna vrijednost intervala s kojim se radi pretraga manja od minimalne vrijednosti korijenskog čvora tada pretragu nije potrebno obavljati u lijevom podstablu. Ako je pak minimalna vrijednost intervala s kojim se radi pretraga veća od maksimalne vrijednosti svih podstabla koju čvor čuva tada nije potrebno raditi daljnju pretragu niti u jednom od podstabla jer sigurno nema intervala koji će biti u presjeku. U suprotnom, provjera se po istom principu nastavlja u podstablama u kojima ju je potrebno obaviti do listova stabla ili ranije spomenutih uvjeta zaustavljanja [8].

U sklopu rada korištena je implementacija Interval tree strukture u Javinoj knjižnici „Breinify: brein time utilities” koja omogućava i definiranje uključenosti granica intervala što je jedan od zahtjeva uvjeta pretplate. Također podržana je pretraga i povrat više intervala i jednostavno dodavanje pretplata kao elementa čvorova što su također zahtjevi sustava.

4.2.2 Apache Lucene

Za tekstualni tip podataka unutar sustava, uvjet pretplate koji pretplatnik može definirati sastoji se od seta ključnih riječi koje definiraju interesne pojmove pretplatnika. Ako naiđe objava koja u vrijednosti sadrži neku od ključnih riječi koju je pretplatnik definirao tada je uvjet te pretplate zadovoljen. Kako bi se efikasnije obavio dohvat pretplata čije su ključne riječi sadržane u tekstu potrebno je obaviti indeksiranje prema ključnim riječima definiranim u pretplati u strukturu koja će zatim omogućiti brzu pretragu kod dolaska tekstualne objave.

Navedeni problem predstavlja problem pretrage teksta i indeksiranja dokumenata. Indeksiranje i pretraživanje dokumenata je složen postupak koji

nadilazi razmatranja ovog rada, no generalni proces se sastoji od uklanjanja stop riječi iz dokumenta, svođenja riječi na korijenski oblik te indeksiranja dokumenta s obzirom na dobiveni skup riječi. Rezultat je struktura u kojoj pojmovi sadržani u dokumentima pokazuju na dokumente i preko njih je moguće obaviti pretraživanje i rangiranje dokumenata. Postoje brojne implementacije indeksiranja i pretrage teksta a jedna često korištena je i Javina knjižnica Apache Lucene [9].

Apache Lucene omogućava opisano indeksiranje dokumenata kao i postavljanje upita s jednom ili više ključnih riječi te i rangiranje dokumenata s obzirom na relevantnost [9]. Iako je generalna ideja slična, u okviru ovoga rada dokumenti su pretplate čiji tekst su definirane ključne riječi a sama pretraga se obavlja s tekstualnim dokumentom koji nailazi kao vrijednost objave. Također samo rangiranje pretplata po važnosti nije od interesa već je samo potrebno dohvatiti sve pretplate koje sadrže neku od riječi objavljene vrijednosti među ključnim riječima. Proces indeksiranja je dakle sastavljanje ključnih riječi pretplate u tekst koji se indeksira, a proces pretraživanja sastoji se od parsiranja objave na riječi i pretragu indeksiranih pretplata s dobivenim skupom riječi objave.

4.2.3 Indeksiranje enumeracija

Tip podataka enumeracije definiran je tako da se unutar vrijednosti objave može naći samo ograničen skup vrijednosti iz unaprijed definiranog skupa. Skup vrijednosti nije velik a unutar uvjeta pretplate, poput ključnih riječi u tekstualnom tipu, moguće je definirati interesni skup enumeracija za koje korisnik želi primati obavijesti.

Za razliku od tekstualnog tipa gdje je broj različitih tekstova i ključnih riječi proizvoljno velik, mogući skup objava i uvjeta pretplate je ograničen. Ta činjenica je iskorištena u radu te je indeksna struktura kreirana tako da se indeks definira kroz mapu u kojoj je ključ enumeracija, a vrijednost te enumeracije set svih pretplata koje su definirale tu enumeraciju kao interes. Ovakva struktura je jednostavna, no omogućava brzi dohvat svih pretplata koje zadovoljavaju definiranu enumeraciju u vrijednosti obavijesti, a s druge strane indeksiranje nove pretplate je jednostavno a pošto je skup vrijednosti enumeracija ograničen ne postoji opasnost od pretjeranog memorijskog zauzeća.

5. Proširivanje sustava

Uslijed dodavanja novih mjernih tipova potencijalno se može javiti potreba za nekim novim tipom podatka koji ne spada u osnovna tri: numerički, teksturalni ili enumeraciju. Uz to, zbog bolje optimizacije procesa filtriranja moguća je želja za odabirom drugačije strategije geoprostornog indeksiranja. Razvijeni sustav stoga omogućava jednostavno proširenje dodavanja podrške za novi tip podatka u sustav, kao i novu strategiju geoprostornog indeksiranja.

5.1 Dodavanje novog tipa podatka

Potreba za novim tipom podataka može se javiti zbog dodavanja novog mjernog tipa koji se ne može prikazati niti s jednim od definiranih tipova. Dodavanje novog tipa podatka sastoji se od tri koraka: dodavanje navedenog tipa u enumeraciju, definiranje uvjeta pretplate za novi tip podataka te definiranje indeksne strategije za taj uvjet pretplate.

Dodavanje novog tipa u enumeraciju sastoji se od dodavanja novog tipa u enumeraciju te izmjenu statičke metode za provjeru valjanosti vrijednosti unutar enumeracije. Navedena metoda koristi se za standardnu provjeru valjanosti vrijednosti koja se definira u objavi ako se za pojedini tip ne nadjača metoda provjere. Ova provjera stoga omogućava definiranje prihvatljive forme vrijednost za novo stvoreni tip podatka.

Nakon dodavanja tipa u enumeraciju potrebno je stvoriti odgovarajući uvjet pretplate za novi tip podataka, ako se korisnicima za navedeni tip želi omogućiti postavljanje uvjeta na vrijednost objave. Sam izgled i atributi uvjeta pretplate su proizvoljni, no valjana klasa uvjeta pretplate mora naslijediti apstraktnu klasu `SubscriptionCondition` i implementirati sve njene apstraktne metode. Metode koje se moraju definirati su metode za serijalizaciju i deserijalizaciju uvjeta pretplate u tekstualni tip podataka kako bi se uvjet pretplate mogao serijalizirati i deserijalizirati zajedno s cijelom pretplatom. Uz to, nužne metode su i metode za validaciju primljene vrijednosti objave s obzirom na definirani uvjet pretplate te pretvorba uvjeta pretplate iz predane mjerne jedinice u standardnu.

Na kraju potrebno je definirati indeksnu strategiju za kreirani tip kako bi se omogućilo efikasno indeksiranje pretplata po uvjetu pretplate. Sam indeks nije nužno definirati te će se u protivnom koristiti pristup slijednog prolaska pretplatama, no preporučeno je implementirati ga zbog brzine izvođenja. Sama implementacija sastoji se od nasljeđivanja apstraktne klase `PropertiesIndexingStrategy` i definiranja svih apstraktnih metoda koje su nužne za indeksiranje. Također, kreirana klasa za indeksiranje mora biti smještena u istom paketu kao ostale klase indeksiranja i nazvana u formatu imena tipa podataka i sufiksa „`IndexingStrategy`” kako bi metoda tvornica mogla uspješno učitati kreiranu klasu. Obavezne metode koje indeks mora osigurati su metode indeksiranja nove pretplate, micanje pretplate iz indeksa i povrat svih pretplata koje zadovoljavaju definiranu objavu. Sama izvedba indeksa je proizvoljna i ovisi o tipu podataka (pogledati poglavlje 4.2) no nužno je da se metodom povratka vraća točan skup pretplata koje zadovoljavaju vrijednost obavijesti. Također, metode indeksa moraju biti sinkronizirane kako bi im se moglo pristupiti višedretveno iz sustava za filtriranje.

5.2 Dodavanje nove geoprostorne strategije indeksiranja

Dodavanje nove indeksne strategije može biti odluka koja je rezultat pronalaska optimalne strategije geoprostornog indeksiranja za razvijeni sustav. Odabir strategije geoprostornog indeksiranja i njene učinkovitosti ovisi o objavama i pretplatama koje dolaze u sustav. Ako u sustav dolazi izrazito velik broj novih pretplata u odnosu na nove objave tada je bolje odabrati indeksnu strukturu koja će omogućiti jednostavno i brzo dodavanje novih pretplata u strukturu. Ako je pak naglasak na velikom broju objava u odnosu na pretplate, tada se isplati žrtvovati povećano vrijeme indeksiranja kako bi se dobila maksimalna brzina dohvata pretplata pri nailasku objave. Uz navedeno, odabir geoprostorne strategije može ovisiti i o geometrijama pretplata u sustavu. Ako su geometrije geografski jako raspršene tada će se odabrati indeksna struktura koja bolje radi s takvim podacima u odnosu jako zgusnute geometrije.

Dodavanje nove strategije geoprostornog indeksiranja sastoji se od definiranja klase nove indeksne strategije i postavljanja punog paketa i imena nove klase u konfiguracijsku datoteku pod atribut „`active geospatial indexing strategy`”. Klasa

navedena u tom atributu će biti učitana kao aktivna geoprostorna indeksna strategija prilikom pokretanja servera za filtriranje. Definirana klasa mora naslijediti apstraktnu klasu `GeospatialIndexingStrategy` te implementirati sve apstraktne metode te klase. Metode koje geoprostorna indeksna strategija mora definirati su metoda dodavanja nove pretplate u indeks, metoda dodavanja liste pretplata u indeks, micanje pretplate iz indeksa i metoda koja vraća sve pretplate koje su potencijalno u presjeku s predanim geoprostornim objektom. Sama implementacija navedenih metoda je proizvoljna no nužno je da se poštuje definirani ugovor svake metode koji je vidljiv i u dokumentaciji navedenih metoda. Metoda dodavanja nove pretplate mora pretplatu zabilježiti unutar indeksa te kada dođe odgovarajući upit vratiti ju kao rezultat. Metoda za dodavanje liste pretplata služi za inicijalizaciju prilikom pokretanja sustava te se koristi jer pojedini indeksi moraju biti izgrađeni nakon dodavanja pojedine pretplate pa se na ovaj način proces inicijalizacije može ubrzati. Metoda uklanjanja pretplate mora maknuti pretplatu iz indeksa te ju više nikad ne vratiti niti na jedan upit. Metoda koja vraća sve pretplate koje su potencijalno u presjeku s predanim geoprostornim objektom mora vratiti sve pretplate koje stvarno jesu u presjeku, no može potencijalno vratiti lažne pretplate odnosno one koje stvarno nisu u presjeku s predanim geoprostornim objektom. Razlog tomu je jer se pretplate još jednom slijedno prolaze zbog provjere zadovoljenosti definiranog prostornog odnosa koji može samo biti precizniji od presjeka (dodir, sadržavanje, prekrivanje itd.) te zbog toga i implementacije većine indeksa koji rade na navedenom principu nema potrebe za eliminacijom lažnih pozitivnih rezultata u toj metodi. Također nužno je da sve metode indeksa budu sinkronizirane kako bi se indeksu moglo jednostavno višedretveno pristupiti.

6. Primjena sustava

Razvijeni sustav objavi-pretplati primjenu može naći u bilo kojem sustavu gdje postoji potreba za objavi-pretplati odnosom klijenata koji sa sobom nose geoprostornu informaciju preko koje je potrebno izvesti filtriranje. Vrijednost u objavama, kao i uvjeti u pretplatama mogu biti definirani kroz proizvoljni tip podataka te je time dana fleksibilnost primjene u sustavima s različitim atributima uz geoprostorni. Uz filtriranje po geoprostornoj komponenti objava, moguće je i definirati filtriranje po vrijednosti objave s obzirom na definirani tip podataka. Time se omogućava dodatna fleksibilnost postavljanja uvjeta od strane pretplatnika ako je takva funkcionalnost potrebna. Razvijeni sustav je za cilj imao ispitati najkvalitetniju centraliziranu izvedbu u pogledu brzine izvođenja s obzirom na filtriranje geoprostornih podataka. S tim ostvarenim ciljem i fleksibilnosti podataka u sustavu, glavna ideja primjene razvijenog sustava je prilagodba željenoj namjeni i integracija u veći sustav koji će uz filtriranje nuditi i ostale potrebne komponente za željeno korisničko iskustvo. Većina sustava će imati potrebu za definiranjem isporuke obavijesti pretplatnicima nakon filtriranja, proces registriranja ili neki oblik definiranja prihvatljivih pretplatnika i objava itd.. Neke od ideja primjena geoprostornog sustava objavi-pretplati s učinkovitim filtriranjem su navedene u nastavku.

Osnovna primjena koja je i temeljna namjena sustava razvijanog u okviru rada je sustav u kojem su objave senzorski podatci a pretplatnici korisnici zainteresirani za mjerenja senzora. Logika primjene razvijenog sustava je u tome da su senzori smješteni na određeno geografsko područje te za njega obavljaju mjerenja. Ovisno o vrsti i primjeni senzora to može biti jednostavna točka, linija ili pak poligon. Senzori se potencijalno mogu i pomicati, no pošto u svakoj objavi šalju geografsko područje to ne predstavlja problem. Same mjerene vrijednosti kao i mjerni tipovi i mjerne jedinice mogu biti raznoliki te sustav omogućava definiranje novih tipova i jedinica uslijed dodavanja novih senzora u sustav. Korisnici se kroz pretplate pretplaćuju samo na interesno područje a uz to mogu definirati i preferirani mjerni tip kako bi izbjegli neželjene objave. Ovakvim sustavom omogućava se pružanje mjerenja senzora iz različitih domena skupu korisnika s različitim zahtjevima na jednostavan način uz učinkovito filtriranje.

Sljedeća mogućnost primjene je u sustavima za objavu vijesti, raznih upozorenja, predviđanja, zanimljivosti i kulturnih obavijesti. Ideja je da se korisnici pretplaćuju na interesno područje a drugi skup korisnika koji nudi različite informativne vijesti i ima pravo objavljivanja može jednostavno na željenom geoprostornom području doprijeti do svih zainteresiranih korisnika za to područje. Ovakve obavijesti su generalno tekstualnog oblika a uvjet preplate uz definirano geometrijsko područje i mjerni tip koji može predstavljati kategoriju informacije, najčešće sadrži definirane ključne riječi za koje je pretplatnik zainteresiran. Definicija podržanih obavijesti je proizvoljna te postoji mogućnost definiranja vlastitog tipa podatka (npr. s naslovom i tijelom) kao i uvjeta preplate.

Jedno od mogućih razmatranja je potencijalna primjena funkcionalnosti sustava u navigacijskim uređajima, odnosno dojavama o potencijalnim gužvama, opasnostima, radovima i sl. na prometnicama. Geometrija cestovnih dionica s obavijestima bi u tom slučaju bile razne linije te bi se za njih slale obavijesti, dok bi zainteresirani putnici koji će naići na tu dionicu dobili obavijest za odabranu geometriju puta i kao posljedicu obavijesti mogli razmotriti alternativne rute.

Pošto su društvene mreže u današnje vrijeme sveprisutne i dolaze u raznim formatima moguće je naći primjenu geoprostornog objavi-pretplati sustava i u njima. Razvijeni sustav moguće je primijeniti u društvenoj mreži u kojoj će korisnici primati i slati obavijesti ovisno o svojoj lokaciji. Moguće je tako dizajnirati sustav za primanje obavijesti u blizini korisnika s definiranim radiusom ili pak za neko određeno interesno područje. Vrijednosti obavijesti u slučaju takvog sustava bi bio sadržaj koji je učestao na takvim mrežama poput teksta, slike ili videa a pretplatnici bi se mogli pretplaćivati na pojedinu kategoriju ovisno o željenom sadržaju.

Navedene ideje su samo neki od primjera u kojima učinkovito geoprostorno filtriranje u sklopu objavi-pretplati arhitekture može biti primjenjivo a generalna poveznica je da su to sustavi koji imaju naglasak na događajima na geoprostornom području za koje je potrebno obavijestiti skup nepoznatih korisnika s obzirom na njihove preferencije.

7. Eksperimentalna evaluacija

Filtriranje pretplata kod dolaska objave je kritična operacija. Smanjivanjem trajanja filtriranja smanjuje se vrijeme odziva sustava na dolazak nove objave. Kako se priprema formata i validacija objave odnosno pretplate rješava u agregatorima čiji je broj proizvoljan i može rasti s brojem korisnika, a svi nadolazeći zahtjevi se mogu pomoću reverznog proxy poslužitelja preusmjeravati do svih agregatora, filter predstavlja centralnu komponentu koju je potrebno optimizirati za najbolje rezultate. Uz to u komponenti filtera postoje otvorene opcije korištenja geoprostornih indeksa, višedretvenosti i indeksa po uvjetu pretplate pa je potrebno odabrati najbolju kombinaciju navedenoga. Kako bi se evaluirale mogućnosti filtriranja odabran je jedan testni slučaj na kojem je ispitana efikasnost pojedinih tehnika filtriranja.

7.1 Opis eksperimenta

Kako bi se evaluirao filter potrebno je generirati pretplate koje će se filtrirati te zatim generirati niz objava i izmjeriti koliko je prosječno vrijeme obrade jedne objave. Za generiranje objava potrebno je definirati geometriju područja za koje se objavljuju, te vrijednosti koje će objavljivati. Za pretplate je također potrebno definirati geometriju područja za koje se pretplaćuju i potencijalno i dodatni uvjet pretplate. Za eksperiment je zato potreban velik (stotine tisuća ili milijuni) skup podataka koji sadrži geometrijski podatak i neke dodatne podatke koji se mogu iskoristiti za vrijednosti objava. Odabrani set podataka su podatci o automobilskim nesrećama unutar Ujedinjenog Kraljevstva u razdoblju od 2005. do 2015. godine (preuzet s www.kaggle.com/silicon99/dft-accident-data). Navedeni skup sadrži nešto više od 1,7 milijuna zapisa o nesrećama. Kako je geometrija svake nesreće točka u prostoru što nije dobro za evaluaciju presjeka koja je poželjna u eksperimentu korišten je još jedan dodatni skup podataka. To su geometrijski podatci o geometrijama područja, distrikta i sektora unutar Ujedinjenog Kraljevstva (preuzet s www.opendoorlogistics.com/downloads/). Kako bi se dobile geometrije poligona većih područja, točke nesreća su mapirane u odgovarajuća područja, distrikte i sektore te su takve poligonske geometrije korištene u objavama i

pretplatama (ideja preuzeta iz Kassab, Liang, Gao, 2010 [10]). Za vrijednosti objava korišten je podatak o broju automobila u pojedinoj automobilskoj nesreći. Navedeni podatak se grupirao u enumeracije vrijednosti od 1 do 8 te >8 nesreća.

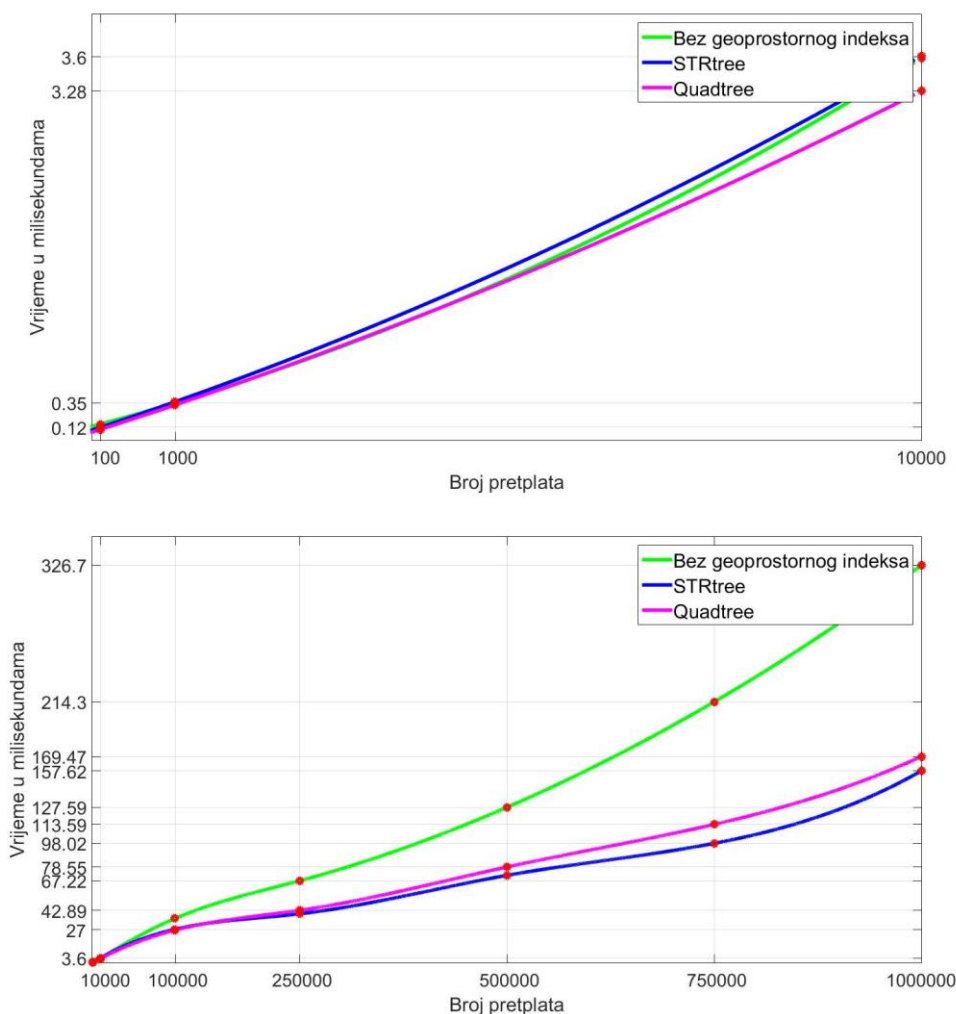
Ukupni broj valjanih geometrijskih točaka nesreća iznosi 1 780 355. U eksperimentima, broj objava je bio fiksni i iznosio 100 000 kako bi se dobila precizna vrijednost trajanja obrade jedne objave, a svi eksperimenti bi bili izvedivi u realnom vremenu. Za geometrije objava, za njih ukupno 70% ostavljena je geometrija točke automobilske nesreće, njih 20% su mapirane u sektor Ujedinjenog Kraljevstva, a 10% u distrikt. Za vrijednost objave uzeta je vrijednost broja automobila u nesreći koja je mapirana u enumeraciju. Svih 1 780 355 zapisa je pretvoreno u objave a zatim se u eksperimentu dohvaćao skup od 100 000 za procesiranje. Broj pretplata je na svakom eksperimentu postepeno povećavan i iznosio je 100, 1000, 10 000, 100 000, 250 000, 500 000, 750 000 i 1 000 000. Broj pretplata se povećavao kako bi se vidio utjecaj porasta broja pretplatnika na odziv sustava prilikom korištenja pojedinih tehnika filtriranja. Same pretplate su generirane iz istog skupa automobilskih nesreća te je geometrija za sve dobivena tako da su se točke nesreća mapirale u sektore Ujedinjenog kraljevstva. Za eksperimente evaluacije geoprostornih indeksa i dretva korištene su pretplate bez dodatnog uvjeta pretplate jer nije relevantan za navedene evaluacije. U eksperimentu gdje se uspoređuje utjecaj indeksa po uvjetu pretplate, pretplatama je nasumično dodijeljen jedan od 5 uvjeta – interes za objave s 1 automobilom, objave s 2 automobila, s 3-4, s 5-6 automobila ili pak 7 i više.

Evaluacija je izvođena na poslužitelju u sklopu Microsoft Azure platforme za koji je dodijeljeno 16 GB radne memorije, procesor Intel Xeon E5-2673 s 2,6 GHz i 2 fizičke jezgre i 4 logička procesora. Operacijski sustav na kojem se eksperiment izvodio je Microsoft Windows Server 2016 Datacenter a verzija Java platforme -10.

7.2 Rezultati eksperimenta

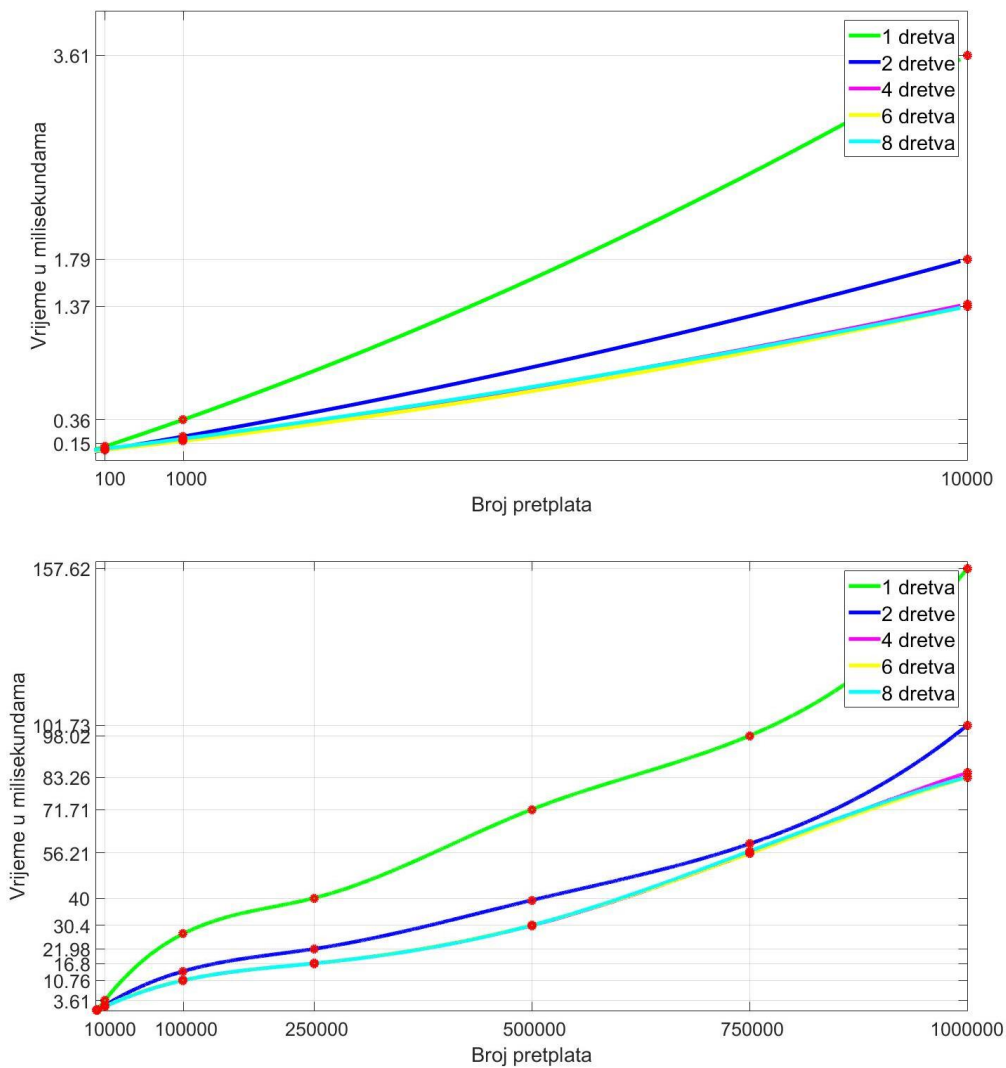
Prvi provedeni eksperiment je evaluacija korištenih geoprostornih strategija indeksiranja. Evaluirano je filtriranje s dvije strategije opisane u poglavlju 4.1 – R-tree i Quadtree te je obavljen test filtriranja pretplata linearnim prolaskom (bez korištenja geoprostornog indeksa) kako bi se vidio smisao korištenja geoprostornih

strategija. Korištene su implementacije geoprostornih strategija indeksiranja knjižnice JTS. Za 3 navedena testa evaluirano je trajanje obrade 100 000 objava za slučaj kada se broj pretplata povećava sa 100 pa sve do 1 000 000. Na temelju izmjerenih točaka izvedena je aproksimacija funkcije koja opisuje vrijeme obrade objave s obzirom na broj pretplata koja je dobivena korištenjem spline funkcije programskog paketa Matlab. Na slici 16 prikazana je usporedba strategija te je zbog preglednosti razdvojena na interval ispod 10 000 pretplata i iznad. Vidljivo je da s nižim brojem pretplata korištene strategije ne daju pomak u odnosu na linearni prolazak jer sam dohvat iz indeksa traje približno jednako. Povećanjem pretplata do izražaja dolazi ubrzanje primjenom indeksa koje u odnosu na linearni prolazak donosi 40 - 50% kraće vrijeme izvođenja. Pri porastu broja pretplata STRtree strategija u odnosu na Quadtree donosi nešto bolje rezultate te je stoga korištena u daljnjim eksperimentima.



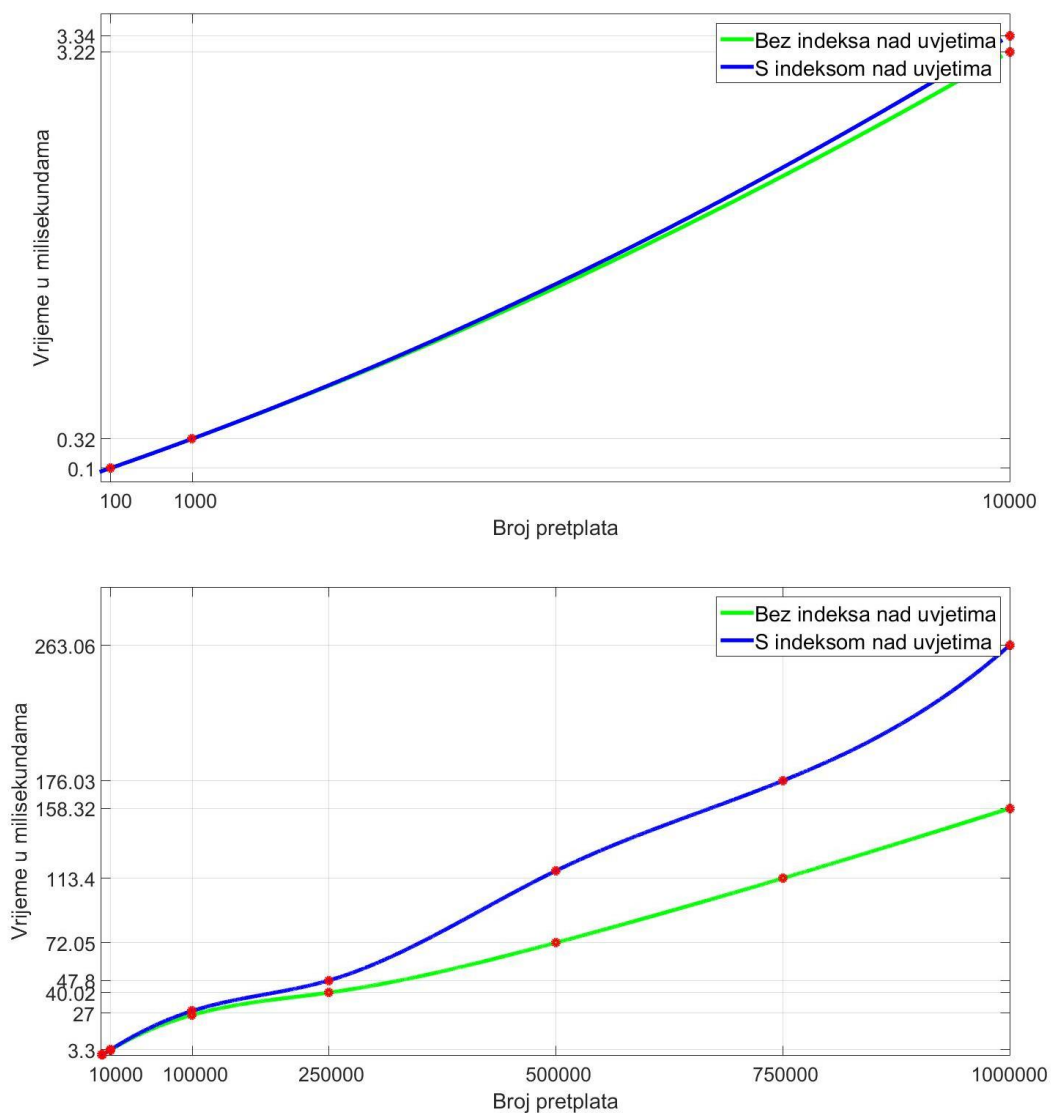
Slika 16 Usporedba geoprostornih strategija indeksiranja

Uz odabir geoprostorne strategije indeksiranja filtriranje se može obavljati s proizvoljnim brojem radnika. Sljedeći eksperiment je evaluirao kako povećanje broja dretva radnika utječe na raniju evaluaciju korištenja STRtree strategije indeksiranja. Rezultati su vidljivi na slici 17. S povećanjem broja dretva na dvije vidljivo je prosječno smanjenje vremena izvođenja do 50%. Kada se broj dretva poveća s 2 na 4 vidljivo je dodatno smanjenje do 20%. Svako povećanje broja dretva iznad toga ne donosi promjenu što je vidljivo i u činjenici da su na grafu funkcije koje prikazuju 4,6 i 8 dretva stopljene zajedno. Razlog tome je u činjenici da se na u testnom okruženju nalaze 4 logička procesora koja se maksimalno iskorištavaju s 4 dretve. Uz to, daljnjim povećavanjem broja dretva usko grlo bi postao pristup indeksnoj strukturi koji je sinkroniziran zbog višedretvenog pristupa.



Slika 17 Usporedba izvođenja s povećanjem broja dretva radnika kod strategije STRtree

Posljednja evaluacija odnosi se na korištenje indeksa za uvjet pretplate. Uz indeksiranje pretplata s obzirom na definiranu geometriju i slijednog prolaska i provjere uvjeta nakon primjene indeksa moguće je pretplate indeksirati i po uvjetu pretplate. Zatim se nakon dohvata pretplata iz oba indeksa gleda presjek dobivenih skupova i rezultat su pretplate koje zadovoljavaju oba kriterija. Vrsta indeksa uvjeta ovisi o korištenom tipu podataka te je u eksperimentu korišten indeks nad enumeracijama jer se vrijednost uvjeta definira za broj automobila u nesrećama koji je dan kao enumeracija. Rezultati evaluacije korištenja indeksa nad uvjetom pretplate dani su na slici 18.



Slika 18 Usporedba korištenja indeksa po uvjetima pretplate

Vidljivo je da korištenje dodatnog indeksa donosi sporije vrijeme izvođenja u odnosu na korištenje geoprostornog indeksa i slijedni prolazak. Dva su glavna razloga takvog rezultata. Jedan je u činjenici da geoprostorni indeks nakon upita vraća relativno mali skup pretplata čiji linearni prolazak i provjera uvjeta pretplate ne predstavlja toliko opterećenje u odnosu na korištenje dodatnog indeksa za dohvrat svih pretplata čiji je uvjet zadovoljen. Nakon dohvata potrebno je pronaći presjek skupova što ako su oba skupa ili jedan veliki može predstavljati značajno kašnjenje u odnosu na linearni prolazak manjim skupom koji vrati geoprostorni indeks. Iz toga proizlazi i drugi problem koji je u činjenici da vrijednosti mogućih enumeracija ima malo – u slučaju eksperimenta samo 9. Zbog toga se iz indeksa uvjeta dohvaća izrazito veliki broj pretplata jer je za puno njih zadovoljen uvjet pretplate. Nakon toga je potrebno pronaći presjek što se predstavlja kao spora operacija. Korištenje dodatnog indeksa po uvjetu pretplate bi stoga imalo smisla kada bi i taj indeks bio jednako efikasan i vraćao mali skup pretplata te u nekim drugim primjenama i tipovima podataka potencijalno ima smisla, no u ovom slučaju i primjeni u enumeracijama nema.

Zaključak

Provedena eksperimentalna evaluacija pokazala je da je za optimalan rad filtera na testiranom sustavu najbolje koristiti STRtree geoprostornu strategiju s 4 dretve radnika i bez dodatnog indeksa nad uvjetima pretplate. Primjena geoprostornog indeksa je preporučiva kada broj pretplatnika dosegne 100 000 i više jer u takvim sustavima geoprostorni indeks najviše dolazi do izražaja. STRtree se pokazao nešto bolji od Quadtree indeksa, no zbog potrebe za ponovnom izgradnjom indeksa kod dodavanja novih pretplata u STRtree strukturu u sustavima u kojima je poznato da će pretplate dolaziti dinamički i u velikom broju bolji izbor je Quadtree struktura koja uz podjednake performanse daje bolje rezultate kod dodavanja novih pretplata. Preporučeni broj dretva generalno prati broj dostupnih logičkih procesora koji su na raspolaganju filteru. Uz to, s porastom broja dretva povećava se konkurentnost pristupa indeksu pa je stoga ubrzanje koje se može postići paralelizmom na ovaj način ograničeno. Primjena dva indeksa se evaluacijom pokazala kao lošija metoda od primjene samo geoprostornog. U sustavima u kojima indeks nad uvjetima pretplate vraća dovoljno mali skup pretplata koji je približan onom koji vraća geoprostorni indeks vrijedno je testirati vrijeme izvođenja s oba indeksa, no kako je evaluacija pokazala dodatni indeks se nije pokazao kao koristan u testiranom slučaju.

Implementirani sustav nudi mogućnost učinkovitog filtriranja tokova geoprostornih objava te se pokazuje kao skalabilno rješenje pri porastu broja pretplata ako se primjenjuje geoprostorno indeksiranje i višedretvenost. Unutar sustava moguće je definirati nove tipove podataka te mjerne tipove i jedinice kako bi bio primjenjiv neovisno o vrsti senzora koji generiraju objave. Sustav rješava probleme obrade i filtriranja objava i pretplata no kako bi našao praktičnu primjenu u budućem razvoju potrebno ga je integrirati u sustav koji nudi ostale komponente potrebne za praktičnu primjenu.

Razvijeni sustav ne rješava probleme registracije i autentifikacije korisnika sustava što je zadaća neke prethodne komponente s kojom bi se sustav povezivao. Registracija i autentifikacija omogućila bi ograničavanje objavitelja na stvarne senzore koji generiraju vrijednosti te pretplaćivanje stvarnih klijenata

sustava. Uz registraciju, nad sloj slanja paketa koji vrše objavitelji i pretplatnici preko UDP protokola u većini primjena bi se trebao razviti sloj sigurne isporuke preko kojeg će korisnici dobiti potvrdu slanja. Također, obrada unutar sustava završava u filteru gdje se filtrirane pretplate samo zapisuju u datoteku s rezultatima. U praktičnoj primjeni potrebno je vršiti isporuku obavijesti filtriranim pretplatnicima. Za to bi se u sustavu brinula komponenta za isporuku sadržaja koja bi pretplatnicima prema njihovom identifikatoru i preferiranoj metodi isporuke mogla dostaviti obavijesti. Na kraju, kako bi sustav bio jednostavan za korištenje potrebno je razviti sučelje prema pretplatnicima za jednostavnije generiranje pretplata i potencijalno i prema objaviteljima koje bi prilikom primjene kod senzorskih uređaja bilo razvijeno u okviru API-ja kojem bi pristupali.

U okviru rada razvijen je i evaluiran geoprostorni sustav objavi-pretplati za učinkovito filtriranje toka senzorskih podataka, a sve navedene komponente su dodatne funkcionalnosti koje bi omogućile bolju i lakšu praktičnu primjenu razvijenog sustava.

Literatura

- [1] Coulouris, G; Dollimore, J; Kindberg, T. Distributed Systems: Concepts and Design. 5th edition. Addison-Wesley Publishing Company, 2012.
- [2] Baldoni, R; Querzoni, L; Virgillito, A. Distributed event routing in publish/subscribe communication systems: A survey. Technical Report 15-05. Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza", 2005.
- [3] Podnar Žarko, I; Pripužić, K; Lovrek, I; Kušek, M. Raspodijeljeni sustavi: Radna inačica udžbenika. v.1.3. FER, 2017.
- [4] Egenhofer, M J. A formal definition of binary topological relationships. Foundations of Data Organization and Algorithms: 3rd International Conference, FODO, Paris, France, 1989, pp. 457-472.
- [5] Butler, H; Daly, M; Doyle, A; Gillies, S; Schaub, T; Schmidt, C. The GeoJSON Format Specification. Internet Engineering Task Force (IETF). Request for Comments: 7946, 2016.
- [6] Guttman, A. R-Trees: A Dynamic Index Structure for Spatial Searching. SIGMOD'84, Proceedings of Annual Meeting, Boston, 1984, pp. 47-57.
- [7] Finkel, R A; Bentley J L. Quad Trees: A Data Structure for Retrieval on Composite Keys. Acta Informatica. Vol. 4 (1), 1974, pp. 1-9.
- [8] Cormen, T H; Leiserson, C E; Rivest, R L; Stein, C. Introduction to Algorithms. Third Edition. London, England: The MIT Press, 2009
- [9] The Apache Software Foundation, Apache Lucene, *Quickstart*, <https://lucene.apache.org/core/quickstart.html>, 23.05.2018.
- [10] Kassab, A; Liang, S; Gao, Y. Real-time notification and improved situational awareness in fire emergencies using geospatial-based publish/subscribe. International Journal of Applied Earth Observation and Geoinformation. Vol. 12 (6), 2010, pp. 431-438.

Sažetak

Geoprostorni sustav objavi-pretplati za učinkovito filtriranje toka senzorskih podataka

Cilj ovog rada je proizvesti učinkoviti geoprostorni sustav objavi-pretplati te provesti evaluaciju najučinkovitijeg načina filtriranja za kreirani sustav. Arhitektura objavi-pretplati sastoji se od skupa objavitelja koji šalju poruke sustavu bez poznavanja primatelja i skupa pretplatnika koji se pretplaćuju na određene kategorije i sadržaj objava bez poznavanja objavitelja. U okviru rada objavitelji su senzori koji izvode mjerenja za određeno geografsko područje. Mjerenja koja senzori izvode mogu biti izražena kroz tipove podataka i mjerne jedinice koji su proširivi u sustavu. Pretplatnici se pretplaćuju na interesno geografsko područje te mogu postavljati dodatni uvjet nad izmjerenom vrijednosti objave. Prilikom dolaska nove objave potrebno je profilirati pretplatnike s obzirom na definirane uvjete preferiranog geografskog područja i dodatne uvjete nad mjernom vrijednosti. Kako bi se filtriranje učinkovito provelo koriste se razne strategije indeksiranja koje se na kraju rada evaluiraju i donosi se odluka o optimalnom procesu filtriranja za dani slučaj.

Ključne riječi: geoprostorni podatci, objavi-pretplati, filtriranje, GeoJSON, senzorski podatci, tokovi podataka, obrada u stvarnom vremenu, indeksne strategije

Summary

Geospatial Publish/Subscribe System for Efficient Sensor Data Stream Filtering

The goal of this paper is to implement efficient geospatial publish/subscribe system and evaluate which is the most effective filtering method in the implemented system. In publish/subscribe architecture there is a set of publishers who send messages to system without knowing who the recipients of the messages are and a set of subscribers who subscribe to categories and certain message content without knowing who the publishers are. In the system which was developed as part of the paper, publishers are sensors which measure data for certain geographic area. Sensors can perform measuring in custom data types and measuring units which can be added to the system. Subscribers subscribe to a certain geographic area and can add additional conditions on measured value inside the publication. When the new publication arrives, it is necessary to filter subscribers depending on their preferred geographic area as well as additional conditions on measured value. In order to perform filtering as efficiently as possible, indexing strategies were used and evaluated to decide which is the optimal filtering process for the implemented publish/subscribe system.

Keywords: geospatial data, publish/subscribe, filtering, GeoJSON, sensor data, data stream, real time processing, indexing strategies