Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/10BE39Trninic

SISTEM ZA OBUHVAT I OBRADU PODATAKA IZ HETEROGENIH IZVORA PODATAKA I NJIHOVO SKLADIŠTENJE U JEZERU PODATAKA

SYSTEM FOR ACQUISITION AND PROCESSING OF DATA FROM HETEROGENEOUS DATA SOURCES AND ITS PERSISTENCE IN A DATA LAKE

Milorad Trninić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu predstavljen je sistem za obuhvat i obradu podataka iz heterogenih izvora. Projektovanje pomenutog sistema motivisala je potreba za velikim skupom podataka u cilju treniranja modela mašinskog učenja čiji je kvalitet direktno proporcinalan raznolikosti i količini dostupnih podataka. Sistem omogućava proširivost i skalabilnost komponenti za obuhvat i obradu kako bi zadovoljio zahtev rada sa velikim skupom podataka različite strukture. Svi obuhvaćeni podaci se trajno pohranjuju u jezero podataka u neizmenjenom obliku. Procesi obrade podataka transformišu obuhvaćene podatake u skladu sa potrebama klijenta. Implementirani sistem je dokaz koncepta za obuhvat, trajnu pohranu i obradu velikog skupa podataka sa ciljem pripreme podataka za treniranje modela mašinskog učenja.*

Ključne reči: *Veliki skupovi podataka, distribuirani informacioni sistemi, ETL.*

Abstract – *In this paper, a system for acquisition and processing of data from heterogeneous data sources is presented. Design of this system is motivated by the use of the big data sets in training of machine learning models. Quality of the trained models is directly proportional with data volume and variety. System supports extensibility and scalability of the components in order to meet the needs of processing big data sets which have various structures. All of the acquired data is persisted in a data lake with an unaltered structure. Data processing transforms acquired data to suit the client's needs. System implemented in this paper is a proof of concept for acquisition, persistence and processing of big data sets with the goal of preparing the data for training of machine learning models.*

Keywords: *Big Data, distributed information systems, ETL.*

1. UVOD

Sa razvojem platformi koje nude usluge iznajmljivanja servera i padom cena tih usluga kao i razvojem alata otvorenog koda, obrada velike količine podataka postaje dostupna sve većem broju kompanija. Sve više se daje na značaju analizi podataka o poslovanju i razvija se disciplina nazvana nauka o podacima (engl. *data science*). Ova disciplina podrazumeva rad sa velikim skupovima podataka i oslanja se na definisane protočne strukture. Protočne strukture predstavljaju niz akcija primenjenih nad obuhvaćenim podacima, gde je izlaz jedne akcije ulaz u narednu akciju i imaju za cilj pripremu podataka za analizu podataka. Priprema podataka i postavljanje protočnih struktura naziva se inženjering nad podacima (engl. *data engineering*). Količina podataka koja se generiše povećava se svakim danom, a tradicionalni sistemi za obradu podataka ne mogu da prate taj trend. Glavni razlog za to jeste što tradicionalni sistemi nisu projektovani tako da inherentno podržavaju horizontalno skaliranje. Horizontalna skalabilnost podrazumeva sposobnost komponenti da uvećaju svoju efikasnost sa povećanjem opterećenja sistema dodavanjem novih računara u klaster.

Osnovni cilj ovog rada jeste formiranje horizontalno skalabilne i proširive platforme za obuhvat i obradu velike količine podataka koja će omogućiti dostupnost podataka na jednom mestu. Takođe, važno je obezbediti slabu spregnutost između komponenti (engl. *loosely coupled components*) za obuhvat i obradu podataka. U slučaju jakih veza između komponenti obuhvata i obrade podataka javlja se problem implementacione zavisnosti komponenti.

2. STANJE U OBLASTI

Vodeći klaud (engl. *cloud*) provajderi poput Amazon Veb Servisa (engl. *Amazon Web Service, AWS)*, Majkrosoft Azur (engl. *Microsoft Azure)* i Gugl Klaud Platforme (engl. *Google Cloud Platform, GCP)* poseduju svoje predloge rešenja za obuhvat i obradu podataka uz trajno skladištenje na njihovim implementacijima jezera podataka. Jezera podataka predstavljaju trajna skladišta sirovih podataka. Svako od preloženih rešenja za obuhvat i obradu podataka se sastoji od komponenti za obuhvat, obradu i skladištenje podataka. Razlike predloženih rešenja su posledica razlika između korišćenih komponenti, ali se konceptualno ne razlikuju značajno. Prednosti upotrebe navedenih predloga rešenja za obuhvat i obradu podataka su brojne, polazeći od plaćanja po upotrebi resursa, automatskog skaliranja pa do održavanja klastera od strane provajdera. Problem sa upotrebom rešenja predloženih od klaud provajdera nastaje već kad nekoj od komponenti treba izmeniti implementaciju. Takođe postoji problem i privatnosti podataka, jer mnoge kompanije ne žele da iznose svoje podatke na klaud

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Dimitrieski.

okruženje. Pored navedenih problema, rešenja klaud provajdera ograničavaju portabilnost sistema. Naime, proces obuhvata i obrade podataka postaje implementaciono zavisan od softverskih rešenja odabranog provajdera.

Pored vodećih klaud provajdera postoji i nekoliko kompanija koje nude softverska rešenja za obuhvat i obradu podataka kao servis (engl. *Software as-a-Service, SaaS*), jedna od njih je Apsolver (engl. *Upsolver)*. Konceptualna razlika Apsolvera i rešenja klaud provajdera potiče od činjenice da je Apsolver gotov servis bez mogućnosti izmene implementacije obuhvata i obrade podataka [1]. Rešenja klaud provajdera nude platformu sa komponentama za obradu i obuhvat podataka, a korisnik je zadužen za implementaciju procesa obuhvata i obrade. Ograničenje dostupnih komponenti zaduženih za proces obrade može negativno da utiče na ceo proces obuhvata i obrade podataka iz razloga što postoje specijalizovana softverska rešenja za određene slučajeve upotrebe koja znatno efikasnije rade od univerzalnog softverskog rešenja.

Arhitektura opisana u ovom radu je projektovana sa ciljem da podrži proširivost što omogućava izmenu, dodavanje ili uklanjanje tehnologija koje implementiraju određene komponente arhitekture. Projektovani sistem takođe omogućava potpunu kontrolu nad konfiguracijom komponenti što omogućava optimizaciju za platformu na kojoj se sistem nalazi. Migracija na neki drugi klaud provajder sa prethodno navedenih platformi i servisa nije moguća ili zahteva značajne izmene u implementaciji, dok se u projektovan sistemu migracija svodi na transfer podataka na novo okruženje. Iz navedenih razloga projektovana je opisana arhitektura za potrebe kompanije u okviru čijeg projekta je i implementiran praktični deo ovog rada.

3. ARHITEKTURA SISTEMA ZA OBUHVAT I OBRADU PODATAKA

Projektovana arhitektura sistema za obuhvat i obradu podataka iz heterogenih izvora podataka može se podeliti na četiri jasno definisana modula. To su: (1) modul za obuhvat podataka, (2) modul za obradu podataka, (3) modul za nadzor logova sistema i (4) modul za komunikaciju s klijentom.

3.1. Modul za obuhvat podataka

Modul za obuhvat podataka preuzima podatke sa izvora podataka i prosleđuju ih modulu za obradu podataka. Ovaj modul sastoji se od dve komponente: komponenta zadužena za obuhvat podataka iz izvora i komponenta zadužena za razmenu poruka.

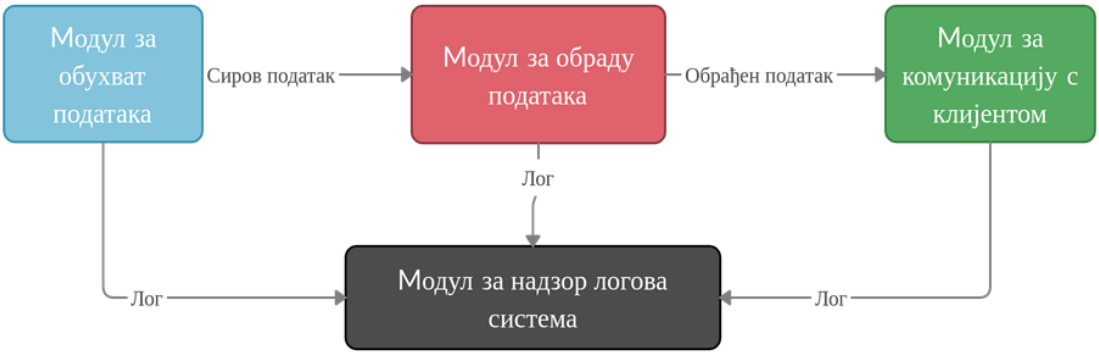
Komponenta zadužena za obuhvat podataka iz izvora ima za zadatak da sakupi podatke iz izvora definisanih od strane klijenta koji koriste obrađene podatke iz ovog sistema. Način obuhvata podataka zavisi od samog izvora podataka, to mogu biti programi koji preuzimaju podatke sa programskog interfejsa aplikacija (engl. *application programming interface, API*), podaci sa internet svega (engl. *internet of things, IoT*) uređaja itd. Najvažniji zahtev koji ova komponenta treba da ispuni je mogućnost ostvarivanja komunikacije sa komponentom za razmenu poruka.

Komponenta zadužena za razmenu poruka ima za cilj prenošenje podataka sa sistema za razmenu poruka u modul za obradu podataka. Postoje dva konceptualno različita načina da se prenos podataka izvrši: direktno i posrednikom tj. brokerom. Direktna komunikacija povećava složenost komponenti koje ostvaruju komunikaciju, jer svaka od komponenti mora biti svesna interfejsa druge komponente sa kojom razmenjuje poruke. Pored dodatne složenosti javlja se i jaka sprega odnosno implementaciona zavisnost između komponenti. Komunikacija putem brokera uvodi pojam teme (engl. *topic*). Tema u ovom vidu komunikacije predstavlja skladište poruka. Na jednu temu mogu se objavljivati poruke ili se na temu može pretplatiti, odnosno čitati poruke sa nje. Navedeni koncept poznat je kao objavi-pretplati (engl. *publish-subscribe*). Upotrebom brokera omogućava slabu spregnutost između komponenti.

3.2. Modul zadužen za obradu podataka

Modul zadužen za obradu podataka sastoje se iz tri komponente: komponenta zadužena za preuzimanje poruka, trajno skaladište podataka - jezero podataka i komponenta zadužena za obradu.

Komponenta zadužena za preuzimanje poruka predstavlja pretplatnika u sistemu razmene poruka. Za svaki od izvora podataka postoji jedna instanca ove komponente koja čita poruke sa komponente za razmenu poruka odnosno odgovarajuće teme. Cilj ove komponente jeste upis sirovih podataka u jezero podataka. Obrada u ovoj komponenti je svedena na dodavanje metapodataka.
Najvažnija osobina ove komponente jeste efikasno čitanje velikog skupa podataka.



Slika 1. *Moduli arhitekute*

Trajno skaladište podataka - jezero podataka u ovoj arhitekturi čini objedinjeno skladište svih podataka obuhvaćenih sa izvora podataka, na koje se naslanjaju odgovarajuće komponente za obradu podataka. U jezero podataka se smeštaju sirovi podaci u njihovom originalnom obliku uz dodatak metapodataka kao što su vreme obuhvata podataka, ime izvora i verzija izvora. Podaci se u trajno skladište upisuju inkrementalno, što znači da se tokom upisivanja novih podataka stari podaci nikada ne brišu niti menjaju.

Komponenta zadužena za obradu podataka u ovoj arhitekturi ima za cilj da predstavi obuhvaćene podatke na način koji klijentu odgovara. Obrada podataka u ovoj arhitekturi je paketna. Obrađeni podaci mogu predstavljati pogled nad sirovim podacima u jezeru podataka ili izvedenu informaciju iz podataka. Neke od najvažnijih osobina ove komponente su dobre performanse paketne obrade podataka i dostupnost brojnih konektora za čitanje i pisanje podataka. Pored procesa obrade ova komponenta zadužena je za slanje obrađenih podataka u modul za komunikaciju sa klijentom.

3.3. Modul za komunikaciju s klijentom

Klijent u ovom sistemu predstavlja svaku eksternu aplikaciju ili korisnika koji je u komunikaciji sa sistemom. Ovaj modul postoji da ne bi dolazilo do remećenja procesa izvršavanja, odnosno da klijent ne može da menja podatke u bilo kojoj komponenti sistema i time potencijalno promeni ishod drugih procesa obrade podataka. Postoji i aspekt skrivanja implementacionih detalja procesa obrade od klijenta iz razloga što klijent ne bi trebalo da vodi računa o načinu implementacije već samo o rezultatima obrade podataka. Kako su klijentski podaci izlazni podaci paketne obrade, komunikacija se obavlja skladištenjem podataka u bazu podataka kojima klijenti mogu da pristupe. Odabrana baza podataka treba da ima sledeće osobine: mogućnost skladištenja velike količine podataka i mogućnost skladištenja polustrukturiranih i nestrukturiranih podataka.

3.4. Modul za nadzor logova sistema

Kako je u pitanju platforma za obradu velikog skupa podataka može se zaključiti da skoro sve komponente imaju više od jednog procesa, odnosno funkcionišu u distribuiranom režimu. Kako zahtevi za resursima rastu, raste i broj procesa pa i čvorova klastera. Ovaj skup komponenti ima za cilj da na jednom mestu okupi sve logove sistema, omogući njihovu pretragu i pregled u korisničkom interfejsu. Najvažnije osobine ovih komponenti su brza pretraga logova i pregledan korisnički interfejs.

4. IZBOR TEHNOLOGIJA ZA IMPLEMENTACIJU KOMPONENTI SISTEMA

Tehnologija koja će predstavljati komponentu za razmenu podataka u ovom sistemu je Apači Kafka (engl. *Apache Kafka*). Kafka je distribuirani sistem za razmenu poruka koji se zasniva na objavi-pretplati načinu razmene poruka uz posredstvo najmanje tri brokera [2]. Brokeri sadrže teme koje se skladište na disku. Teme su sačinjene od skupa poruka. Skup poruka je podeljen u particije koje se repliciraju i time omogućavaju otpornost na otkaze [2]. Kafka može da skalira dodavanjem brokera.

Tehnologija odabrana za komponentu za preuzimanje poruka je Apači Spark (engl. *Apache Spark*). Spark je radni okvir koji omogućava distribuiranu obradu podataka. Ono što je prednost Spark-a u odnosu na druge radne okvire za obradu podataka jeste brzina obrade [3]. Veća brzina obrade velike količine podataka u Spark radnom okviru je omogućena izvršavanjem u memoriji, odloženom evaluacijom (engl. *lazy evaluation*) i optimizacijama plana izvršavanja (engl. *execution plan*). Glavni razlog za odabir Spark radnog okvira za preuzimanje poruka sa sistema za razmenu poruka jeste to što ima brojne konektore za čitanje i pisanje podatka.

Jezero podataka u ovom sistemu predstavlja trajno skladište svih podataka koji su obuhvaćeni sa izvora podataka. Tehnologija odabrana za ovu komponentu je *HDFS*. *HDFS* je distribuirani sistem datoteka koji je deo Hadup radnog okvira za distribuiranu obradu podataka. Blok podataka je najmanja jedinica manipulacije u *HDFS* sistemu datoteka [4]. Replikaciom blokova postiže se tolerancija otkaza [4]. *HDFS* skalira dodavanjem čvorova sa podacima. Razlog za upotrebu *HDFS*-a za jezero podatak jeste mogućnost skaliranja i činjenica da je *HDFS* softver otvorenog koda [4].

Podaci iz jezera podataka se obrađuju paketnom obradom i smeštaju u klijentska skladišta. Tehnologija odabrana za ovaj zadatak je Spark, konkretno njegov *SQL* modul. Ovaj modul na Spark dodaje mogućnost pisanja naredbi u jeziku za strukturirane upite (engl. *structured query language, SQL*) kao i tipizaciju objekata kojima se manipuliše [3]. Spark je odabran za obradu podataka zbog superiornije brzine u odnosu na njemu slične radne okvire poput Hadup MapRidjus (engl. *Hadoop MapReduce)* [3]. Pored navedenih osobina Spark aplikacije imaju dobru interoperabilnost sa drugim tehnologijama za skladištenje podataka [3].

Nakon obrade podaci se smeštaju u klijentska skladišta. Kako je obrada paketna i kako su u pitanju velike količine podataka mogućnosti koje ova komponente treba da ima su: horizontalno skaliranje, efikasno čitanje velike količine podataka i mogućnost rada se polustrukturiranim i nestrukturiranim podacima. Tehnologija koja zadovoljava ove kriterijume je MongoDB [5]. MongoDB je najkorišćenija distribuirana nerelaciona (engl. *NoSQL)* baza podataka [5]. MongoDB kao i sve do sada navedene tehnologije podržava horizontalno skaliranje izvršavanjem procesa horizontalnog particionisanja (engl. *sharding*) [5].

Pošto svaka od navedenih tehnologija radi u distribuiranom režimu očekuje se i veliki broj čvorova u sistemu. Problem sa ovim jeste detekcija loga nekog od procesa bilo koje komponente sistema. Za komponentu koja vrši nadzor sistema odabran je Elastik Stek (engl. *Elastic Stack).* Sastoji se od tri softverska rešenja kompanije Elastik *(engl. Elastic),* u pitanju su Elastiksrč *(engl. Elasticsearch)*, Logsteš *(engl. Logstash)* i Kibana (engl. *Kibana*). Elastiksrč predstavlja skladište teksta loga sa velikom brzinom pretrage. Logsteš je procesor formata logova. Kibana u ovom sistemu služi kao korisnički interfejs za pretragu logova.

5. IMPLEMENTACIJA

Sistem je projektovan za obuhvat i obradu podataka iz heterogenih izvora podataka. Za dokaz koncepta odabrani izvor je VebKroler (engl. *WebCrawler) aplikacija.* VebKroler ima za cilj skupljanje podatak sa veb sajtova čiji su osnovni sadržaj vesti iz oblasti finansija. Napisana je u programskom jeziku Pajton i koristi Selenijum, radni okvir za otvaranje stranica i navigaciju kroz stablo elemenata veb stranice. Razvijene su dve celine *VebKroler* aplikacije. Prva celina obuhvata hiperlinkove od vesti na određenim putanjama. Druga celina obuhvata podatke poput naslova i tela članka, glavne slike, datum objave itd. sa svakog obuhvaćenog hiperlinka. Ove dve celine komuniciraju putem Kafke.

Nakon obuhvata podataka druga celina šalje podatke u vidu poruke na Kafku, temu koja se zove isto kao i domen sa kojeg su podaci obuhvaćeni. Sa ove teme poruke se preuzimaju od strane aplikacije nazvane KrolerKonzjumer (engl. *CrawlerConsumer)* napisane pomoću Spark radnog okvira. Na preuzete poruke dodaju se metapodaci poput vremena obuhvata i imena teme. Obogaćeni podaci čuvaju se na *HDFS*-u u Parke (engl. *Parquet)* formatu.

Podaci sa *HDFS*-a se čitaju u aplikaciji *KarensiPredikt* (engl. *CurrencyPredict)* koja predstavlja implementaciju komponente za obradu podataka. Aplikacija *Karensi-Predikt* ima za cilj da na osnovu teksta vesti odredi koja se valuta spominje u tekstu i kakav je sentiment. Određivanje valuta i sentimenta je svedeno na problem klasifikacije. Za svaku valutu postoji jedan binarni klasifikator koji određuje da li se ta valuta pominje ili ne pominje u tekstu. Korišćeni binarni klasifikator je logistička regresija [6]. Pošto tekst vesti nema strukturu, korišćene su tehnike obrade slobodnog teksta (engl. *Natural Language Processing NLP)* za pretprocesiranje i ekstrakciju odlika (engl. *features*). Neke od pomenutih tehnika su: ukljanjanje stop reči, tokenizacija, svođenje reči na osnovni oblik (engl. *stemming*) i drugo [7].

Nakon određivanja pomenutih valuta u tekstu i sentimenta teksta članka rezultati se smeštaju u odgovarajuću kolekciju MongoDB baze podataka.

6. ZAKLJUČAK

Kompanije neprestano pokušavaju da pronađu najbolje rešenje za obuhvat i obradu podataka koje se uklapa u njihove zahteve po pitanju arhitekture, tehnologija i sigurnosti podataka. Postoje skupovi gotovih alata koji se mogu koristiti u cilju obrade i obuhvata podataka, ali pored svojih prednosti uvode i razna ograničenja. Ograničenja gotovih sistema jesu mogućnost odabira tehnologija. migracija sistema na druge platforme i privatnost podataka. Upotreba tehnologija za koju kompanija ima raspoloživ inženjerski kadar je vrlo važna pri planiranju projekata, što znači da bi kompanija trebalo da zaposli nove ljude koji poznaju nametnute tehnologije ili da postojeći kadar uči nove tehnologije.

Sistem opisan u ovom radu je zamišljen da pomogne pri izvlačenju korisnih informacija iz obuhvaćenih podataka. Izvedene informacije postaju nešto što doprinosi vrednosti kompanije, samim tim se javlja i zahtev za zaštitom izvedenih informacija. Iz navedenih razloga javila se potreba za projektovanjem i implementacijom sistema za obuhvat i obradu podataka iz heterogenih izvora podataka kao i njihovo trajno skladištenje. Skladištenjem podataka u ovakvom sistemu vremenom bi se stvorio veliki skup podataka od interesa za kompaniju koji bi doprineli kvalitetu izvedenih informacija iz tih podataka po ceni skladišta i resursa obrade.

Upotrebom ovog sistema, u kojem su jasno razdvojeni procesi za obuhvat i obradu, mogu se lakše i podeliti zadaci na međusobno nezavisne razvojne timove. Iz svega navedenog zaključuje se da je glavni doprinos ovog rada proširiva i skalabilna arhitektua koja omogućava razdvajanje obuhvata i obrade velikog skupa podataka kao i trajno čuvanje sirovih podataka iz heterogenih izvora podataka.

U okviru daljeg razvoja sistema biće omogućen ispravan rad na više od jednog servera, čime bi se u praksi podržala obrada velikog skupa podataka. Zatim je u planu omogućavanje da komponente za obradu podataka skaliraju po zahtevu posla koji se obavlja kao i izbegavanje usporenja procesa obrade porastom skupa podataka implementacijom inkrementalne obrade.

7. LITERATURA

[1] Yoni Iny, "Upsolver - Technical Whitepaper: The Modern Data Lake Architecture", 2019

[2] https://kafka.apache.org/documentation/ (pristupljeno u avgustu 2020.)

[3] https://spark.apache.org/ (pristupljeno u julu 2020.)

[4] Tom White, *Hadoop: The Definitive Guide, Fourth Edition*, O'Reilly Media, Inc., 2009

[5] Kristina Chodorow, Michael Dirolf, *MongoDB: The Definitive Guide*, O'Reilly Media, Inc., 2015

[6] https://spark.apache.org/docs/latest/ml-guide.html (pristupljeno u avgustu 2020.)

[7] https://en.wikipedia.org/wiki/Natural_language_processing#Common_NLP_Tasks (pristupljeno u septembru 2020.)

Kratka biografija:

Milorad Trninić rođen je u Novom Sadu 1995. godine. Dilpomirao je na Fakultetu tehničkih nauka 2018. godine iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika sa prosečnom ocenom 9,12. Od maja 2020. godine radi kao inženjer podataka.