Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/23BE31Vucaj

SISTEM ZA OBRADU I VIZUALIZACIJU VELIKOG SKUPA TELEMETRIJSKIH PODATAKA IZ TRKA FORMULE 1

SYSTEM FOR BATCH PROCESSING AND VISUALIZATION OF FORMULA 1 BIG DATA TELEMETRY

Aleksa Vučaj, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - *U ovom radu predstavljen je sistem za paketnu obradu i vizualizaciju podataka telemetrije Formule 1. Cilj ovog rada je omogućiti lakšu analizu podataka telemetrije Formule 1 kao i njihovu vizuelizaciju. Da bi se ovo postiglo, projektovan je i implementiran sistem namenjen za paketnu obradu podataka. Glavne karakteristike ovog sistema su da je sistem samostalan zahvaljujući orkestratoru poslova koji dobavlja i procesira podatke nakon svake trke, kao i vizuelizacija obrađenih podataka.*

Ključne reči: *Paketna obrada velikih skupova podataka, Vizualizacija rezultata obrade podataka, Formula 1, Apache Spark, Veliki skupovi podataka*

Abstract - *In this paper, we present a system for batch processing and visualization of Formula 1 telemetry data. The goal of this system is to facilitate easier processing and visualization of Formula 1 telemetry data. To achieve this, a system designed for batch processing was implemented. Main features of the system are that the system can work independently with the help of a job orchestrator, which fetches and processes data after every race, as well as visualization of processed data.*

Keywords: *Big data batch processing, Data visualization, Formula 1, Apache Spark, Big data.*

1. UVOD

Sport, kao jedna od najznačanijih razonoda u svetu, sve više zavisi od prikupljanja podataka i analize istih. Odluke u sportovima, poput američkog fudbala, se već neko vreme najviše zasnivaju na obrađivanju podataka.

Ukoliko se u sport doda da pored samog sportiste na rezultate utiču i karakteristike mašine kojom upravlja takmičar, jasno je da je zavisnost od podataka postaje obavezna i neraskidiva. Formula 1 sigurno jeste jedan od najrazvijenijih i najzavisnijih sportova od različitih parametara koji utiču na performanse bolida i vozača. Osim što podešavanje bolida vozaču omogućava trenutne performanse na stazi, podaci su od još većeg značaja timovima inženjera koji imaju za zadatak da bolide pripreme na optimalan način za sve trke u toku sezone.

Inženjeri prikupljene podatka koriste kako bi pomoću svojih alata mogli da simuliraju milione trka u specijalizovanim softverskim rešenjima, kako bi utvrdili koje postavke su najbolje za njihov tim. S tim u vezi, može se primetiti da je u F1 sve veća prisutnost kompanija kojima se deo delatnosti nalazi i u domenima skladištenja i obrade velike količine podataka. Kompanije poput *Oracle*-a, *Google*-a, *Microsoft*-a i *Amazon*-a sastavni su deo ovog tehničko-tehnološkog sporta, te imaju obostranu korist da u zamenu za pružanje pomoći timovima u analizi podataka, dodatno promovišu svoj brend u jednom od najprestižnijih takmičenja.

Cilj ovog rada jeste omogućiti lakšu analizu podataka telemetrije Formule 1 kao i njihovu vizuelizaciju.

Pod terminom telemetrije se, u ovom radu i u daljem tekstu, smatraju podaci o bolidu koji se sakupljaju u toku njihove vožnje na stazi. Informacije koje su dostupne u okviru ovog rada su informacije o brzini bolida, broju obrtaja motora, trenutnom stepenu prenosa, iks, ipsilon i cet koordinatama bolida na stazi, udaljenosti od starta, udaljenosti od bolida koji je ispred i slično [1].

2. PREGLED TRENUTNOG STANJA U OBLASTI

Interesovanje za telemetriju i Formulu 1 ubrzano raste, broj alata za analizu ne prati taj tempo. Timovi Formule 1 su zapravo jedini u Formuli 1 kojima su ovi podaci neophodni kako bi se sa što više uspeha takmičili u ovom sportu. Telemetrijski podaci i njihovo tumačenje i razumevanje stvara razliku između najboljeg i najgoreg tima. Ovi podaci se koriste u razne svrhe poput simulacija koje koriste timovi kako bi pripremili bolide za predstojeće trke. Svi javno dostupni servisi za pružanje informacija o telemetriji većinski se oslanjaju na *Python* biblioteku *FastF1* kako bi dobavili svoje podatke. Sa druge strane, sami grafikoni širem auditorijumu gledaoca ne znače ništa bez dodatnog pojašnjenja i skretanja pažnje na određene detalje, pa je samim tim broj ljudi zainteresovanih da se bave ovom temom veoma mali.

U trenutku pisanja rada moguće je pronaći svega dve internet stranice na kojoj se mogu interaktivno birati željena telemetrija i više sajtova koji se bave analizom unapred pripremljenih grafikona i telemetrije. Sajtovi sa interktivnim odabirom telemetrije mogu se naći na f1-telemetry.net i f1-tempo.com. Oba sajta svoje podatke dobijaju putem *FastF1 Pyhton* biblioteke. Važno je napomenuti da niti jedan od ova dva sajta nema mogućnost uvida u telemetriju za više od jednog kruga.

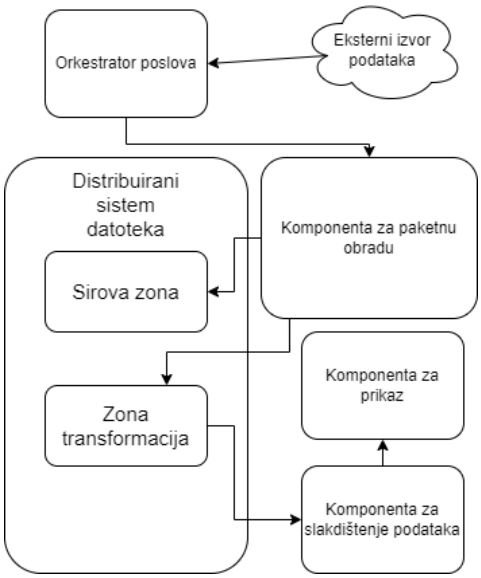
NAPOMENA:
Ovaj rad proistekao je iz master rada čiji je mentor bio dr Vladimir Dimitrieski, docent.

Dakle, analiza na nivou jedne cele trke, više trka ili pak cele sezone nije moguća.

Analiza postojećih alata koji poseduju timovi koji su učesnici F1 nije moguća zbog toga što ne postoji pristup niti uvid u njihove mogućnosti ili arhitekturu.

3. ARHITEKTURA SISTEMA

Zadatak ovog sistema je da podatke o telemetriji Formule 1 iz spoljnih izvora prikupi, izvrši različita izračunavanja i agregacije, skladišti u bazi podataka i na kraju prikaže pomoću alata za pravljenje grafikona i vizualizaciju podataka. Na Slici 1 je prikazana arhitektura sistema, dok će u nastavku teksta biti opisani prikazani moduli.



Slika 1. *Arhitektura sistema*

Podaci se preuzimaju sa jednog od eksternih izvora podataka. Nakon toga, orkestrator poslova izdaje zadatak komponenti za paketnu obradu zadataka da skladišti ove pridošle podatke u *CSV* formatu u sirovu zonu podataka na distribuiranom sistemu datoteka. Iz sirove zone podataka se podaci učitavaju u komponentu za paketnu obradu koja ih obrađuje i transformiše i tako transformisane ih čuva u *CSV* formatu u direktorijumima zone transformacija. Poslednji korak u ovom sistemu je upisivanje transformisanih podataka u trajno skladište podataka.

3.1 Orkestrator poslova

Zadatak orkestratora tokova poslova je da zadaci koji su definisani budu izvršeni u zakazano vreme. Ove zadatke definiše korisnik i mogu biti *Python* ili beš skripte, kao i beš komande.

Apache Airflow kao tehnologija namenjena za orkestraciju tokova poslova ima mnogobrojne alternative. Neke od najpopularnijih tehnologija pored *Airflow*-a jesu *Luigi*, *Apache NiFi*, *AWS Step Functions* i *Prefect*. U uži odabir za korišćenje tehnologije za izvršavanje tokova poslova ušli su *Airflow* i *Luigi*, te će u narednom delu teksta biti dato poređenje ove dve tehnologije.

Suštinski najbitnija karakteristika kod ove komponente je način i mogućnost za zakazivanje zadataka i tokova poslova. Distribuirana priroda *Airflow* omogućava i distribuirano izvršavanje zadataka koje je potrebno podesiti na izvršiocu nakon čega je dovoljno prepustiti zakazivaču koji je sastavni deo *Airflow* da sam pokreće zadatke. *Luigi* ne podržava mogućnost da sam pokreće zadatke, već je moguće namestiti kron posao (engl. *Cron Job*) putem komandne linije kako bi on pokrenuo izvršavanje poslova. Iako distribuirano izvršavanje u ovom slučaju ne igra ključnu ulogu, sposobnost *Airflow*-a da se zakaže i izvrši je ključna u odabiru ove tehnologije. Osim toga, *Airflow* ima mogućnost da pokrene izvršavanje sledećeg zadatka iako se njegov prethodnik i dalje nije potpuno završio. Ovo preklapanje ubrzava izvršavanje tokova, ali je kompleksno za implementirati

3.2 Distribuirani sistem datoteka

Distribuirani sistem datoteka u ovom sistemu služi za skladištenje podataka u sirovom i transformirsanom obliku. Podaci se čuvaju u *CSV* formatu i u zoni transformacije i u sirovoj zoni. U sirovu zonu se upisuju tek preuzeti, neobrađeni podaci. Komponenta za paketnu obradu učitava sirove podatke i nad njima vrši transformacije. Nakon trasformisanja, podaci se čuvaju u zoni transformacije, odakle se prepisuju u bazu podataka.

Kao rezultat pretrage alternativa za distribuirani sistem datoteka *Hadoop* najčešći odgovor na koji se nailazi je *HBase*. *HBase* je nerelacioni sistem za upravljanje bazom podadataka usmerenom na kolone (engl. *Column-oriented*). *HBase* kao svoju osnovu ima *HDFS* i namenjen je za skladištenje i upravljanje velikom količinom strukturiranih i polu strukturiranih podataka, a takođe svoju otpornost na greške ekspoatiše iz *HDFS*-a.

HDFS je sistem namenjen za skladištenje velikih skupova podataka na klasteru izgrađenom od kućnih uređaja i namenjen je za paketnu obradu podataka, dok je sa druge strane *HBase NoSQL* baza podataka. *HBase* je optimizovan za nasumičan pristup pisanju i čitanju velikih skupova podataka i može da podrži veći broj korisnika istovremeno.

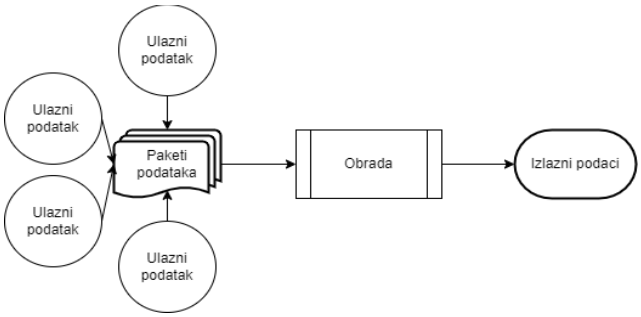
HDFS skladišti podatake na osnovu hijerarhijske strukture datoteka pri čemu su te datoteke izdeljene u blokove podataka i distribuirane u klasteru. Za razliku od *HDFS*-a, *HBase* podatke organizuje po redovima i kolonama unutar sebe.

3.3. Paketna obrada podataka

Paketna obrada podataka istovremeno obrađuje veliku količinu podataka koji se prvo skladište, a zatim se čitaju u paketima [2]. Specifikacija veličine paketa zavisi od implementacije sistema, te veličina može biti ograničena na različite načine kao što su broj podataka, vremenski okvir u kom su proizvedeni, ograničenja sistema i slično. Paketi podataka se zajedno šalju obrađivaču podataka. Za dobijanje krajnjih rezultata obrade potrebno je da se svaki zasebno obrađeni zapis izvrši uređivanje i sumiranje rezultata obrade. Na Slici 2 konceptualno je prikazana paketna obrada.

Tehnologija izabrana za implementaciju modula koji se bavi paketnom obradom podataka jeste *Apache Spark*. Kao jedna od mogućih alternativa u kojoj je moguće implementirati paketnu obradu podataka je *Apache Flink*. *Spark* i *Flink* su radni okviri otvorenog koda namenjeni za procesiranje velikih skupova podataka i jedni su od vodećih kada je u pitanju njihova upotreba. Obe ove tehnologije namenjene su za obradu velikih skupova podataka u paralelnom režimu pritom pružajući podršku putem prog-

ramskog interfejsa tokom analize i manipulacije podacima.



Slika 2. *Konceptualni prikaz paketne obrade podataka*

Spark je više optimizovan za paketnu obradu podataka i koristi se u slučajevima gde je za poslove koji vrše paketnu obradu prihvatljivo da imaju nešto višu latenciju u odnosu na obrađivače u realnom vremenu [3]. Upravljanje memorijom kod *Spark*-a oslanja se na model upravljanja memorijom poznat kao *RDD Caching*. *RDD* kešing omogućava skladištenje rezultata međukoraka u toku obrađivanja podataka što ovom sistemu veoma pogoduje jer je potrebno imati više međurezultata tokom agregiranja telemetrijskih podataka. Model obrade podataka (engl. *Data Processing Model*) koji koristi *Spark* je distribuirani skup podataka otporan na otkaze (engl. *Resilient Distributed Dataset, RDD*). *RDD* model predstavlja kolekciju elementa otpornih na otkaze koje je moguće obrađivati u paralelnom režimu.

Glavna prednost *Spark*-a je to da su performanse u poređenju sa *Hadoop*-om i do 100 puta brže za skladištenje pomoćnih rezultata u memoriji, kao i 10 puta kada se podaci čuvaju na disku [4] Prethodno napisana osobina je u ovom slučaju od velikog značaja jer u toku svake iteracije kreiraju privremene rezultate koje koristimo u toku transformisanja telemetrijskih podataka.

3.4 Komponenta za vizuelizaciju

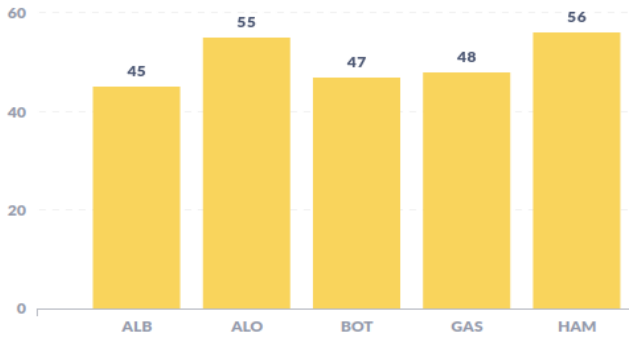
Postoji mnogo alternativa koje je moguće koristiti za vizuelizaciju podataka poput *Tableau*, *Looker*, *Power BI*, *Superset* i mnogih drugih.

Jedan od kriterijuma koji je bio bitan pri odabiru alata za vizuelizaciju svakako mora biti izgled samog korisničkog interfejsa. *Metabase* je alat koji je namenjen za korišćenje „obe" vrste korisnika. Prilagođen je i ljudima koji ne dolaze iz sveta informacionih tehnologija, te nemaju nikakvo predzanje o upitima nad bazama podataka. Koristeći alate za pravljenje upita i već gotovih grafikona, ovaj alat mogu da koriste svi koji imaju minimalno potrebno predznanje u korišćenju kompjutera. S druge strane, korisnici koji imaju iskustvo u pravljenju upita nisu uskraćeni za mogućnost da svoje znanje iskoriste za dobijanje grafikona.
Metabase interfejs sastoji se iz 3 glavne celine kada su u pitanju podaci. Podaci mogu biti predstavljeni u klasičnom tabelarnom prikazu, pomoću različitih grafikona ili u vidu skupa grafikona na jednom mestu, to jest kontrolnih tabli. Osim toga, *Metabase* pruža sigurnost podataka koristeći pristup podacima na osnovu korisnikove uloge u sistemu (engl. *Role-Based Access Control, RBAC*).

4. PRIKAZ IZGLEDA SISTEMA

Implementiran sistem omogućava interaktivnu analizu prikupljenih telemetrijskih podataka. U okviru ovog poglavlja cilj je prikazati moguće izglede aplikacije i njene mogućnosti.

Na Slici 3 prikazan je stubičasti grafikon koji prikazuje koji je najveći broj promena stepena prenosa na Velikoj nagradi Monaka za 5 vozača. Ovaj grafikon je moguće dobiti u svega 4 koraka koristeći *Metabase*-ov editor za pravljenje upita nad bazom podataka.



Slika 3. *Broj promena stepena prenosa na VN Monaka*

Pomoću ovog sistema moguće je veoma lako generisati različite grafikone koristeći telemetriju ne samo iz jedne sesije, već i iz cele sezone, što je glavno ograničenje svih trenutno dostupnih alata. Koristeći ugrađeni vizuelni editor za upite moguće je brzo i lako vizuelizovati unapred pripremljene podatke.

U sklopu *Metabase*-a moguće je i sačuvati generisane grafikone za kasnije korišćenje i grupisati ih u kontrolne table. Kontrolne table su veoma pogodne za korišćenje u slučajevima kada je potrebno brzo pristupiti različitim tipovima grafikona. Dodatna funkcionalnost koju *Metabase* pruža svojim korisnicima je da se pretplate na neku od kontrolnih tabli i da pretplaćenim korisnicima pošalje nove obaveštenje kada se novi grupa grafika formira.

5. ZAKLJUČAK

Formula 1 je specifičan sport iz više razloga. Prvi razlog jeste da za dobar rezutlat nije dovoljan samo sposoban takmičar, već i performantan bolid kojim takmičar upravlja. Drugi je da za razliku od ostalih sportova u kom se tim ili pojedinac takmiče u izolovanim događajima koji se na kraju odražavaju na finalno stanje u šampionatu, u Formuli 1 se svi takmičari takmiče u isto vreme sa svim ostalim protivnicima tokom svih događaja.

Poboljšavanje bolida dugo je zavisilo isključivo od sposobnosti vozača da prenese svoje utiske mehaničarima. Sada, inženjeri pretežno se oslanjaju na rezultate analiza i obrada podataka koje skupljaju tokom slobodnih treninga i trka kako bi bili u mogućnosti da shvate šta se tačno događa sa bolidom. Time, sakupljanje, obrada i tumačenje rezultata čine neizostavne činioce svakog tima koji ima ambicije za uspesima u ovom sportu.

Ovaj rad usmeren je na razvijanje sistema koji gledaocima omogućava da dublje zađu u inače nedostupnu analizu događaja koje je moguće videti tokom 3 dana programa

trkačkog vikenda. Subjektivni osećaji koji se dobija putem prenosa trka na televiziji ili tokom boravka na stazi mogu biti potpuno različiti od onoga šta pokazuju podaci. Zbog toga je potrebno da se prilikom svake diskusije ili analize onoga što je viđeno na trci uključe i nepobitne činjenice kako bi se dobila potpuna slika.

Realizovani sistem sastoji se od nekoliko ključnih delova. Prvi deo je za dobavljanje podataka sa izvora podataka bez kojih vršiti bilo kakvu analizu. Dalje sledi sposobnost sistema da prikupljene podatke učita, obradi, agregira i prirpemi za vizualizaciju. Poslednji, ali možda najbitniji deo, jeste čuvanje i prikaz tih podataka. Bez dobre vizualizacije nema ni analize koja bi mogla dublje dočarati šta se tačno desilo na stazi.

Smer u kom ide dalji napredak ovog sistema fokusiran je na razvijanje striming aspekta obrade podataka. Obrada podataka u realnom vremenu dala bi potpuno novu dimenziju analize podataka jer bi gledaoci mogli dobiti poređenja za manje od minute od trenutka dešavanja na stazi. Dalji napredak ovog sistema mogao bi se ogledati i u povezivanju postojećih informacija o telemetriji sa podacima o vremenskim uslovima na i oko staze. Ovime bi se mogao dobiti dublji uvid u to koji bolidi i vozači se kako ponašaju pri različitim vremenskim uslovima. Primer ove analize mogao bi dovesti do pravljenja relacija vezane za kakve performanse ima određeni tim na stazama koji su na višoj nadmorskoj visini poput Meksiko Sitija ili kako se koji vozač nosi sa visokom vlažnosti vazduha.

6. LITERATURA

[1] theOehrl. (2022). *FastF1 Documentation* [Onlajn]. Dostupno na: https://theoehrly.github.io/Fast-F1/

[2] Packt. (2022). *Distributed batch processing* [Onlajn]. Dostupno na: https://subscription.packtpub.com/-book/big-data-and-business-intelligence/9781784391409/1-/ch01lvl1sec13/distributed-batch-processing.

[3] ProjectPro. (2023). *Apache Flink vs Spark - Will one overtake the other?* [Onlajn] Dostupno na: https://www.projectpro.io/article/apache-flink-vs-spark-will-one-overtake-the-other/282.

[4] Vaidja, N. (2022). *Apache Spark Architecture - Spark Cluster Architecture Explained* [Onlajn]. Dostupno na: https://www.edureka.co/blog/spark-architecture/

Kratka biografija:

Aleksa Vučaj rođen je 8. septembra 1998. godine u Novom Sadu, Vojvodina. Pohađao je gimnaziju „Isidora Sekulić”. Fakultet tehničkih nauka upisao je 2017. godine na smeru Računarstvo i automatika, da bi 2021. diplomirao. Master studije na studijskom programu Računarstvo i automatika usmerenje Računarstvo visokih performansi upisuje 2021. i polaže sve planom i programom predviđene predmete.