Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.6
DOI: https://doi.org/10.24867/26BE30Roncevic

OBRADA VELIKIH KOLIČINA METEOROLOŠKIH PODATAKA

PROCESSING LARGE AMOUNTS OF WEATHER DATA

Nikola Rončević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *U ovom radu prezentujemo moderan pristup za obradu velikih količina podataka koristeći tehnologije računarstva u oblaku. U ovom radu obradu radimo nad meteorološkim podacima, ali se principi mogu primeniti na različite domene.*

Ključne reči: *Velike količine podataka, Obrada podataka, Klaud*

Abstract – *In this article we present modern approach to processing large amounts of data utilizing cloud techologies. In this article the processing is done on weather data but the principles stand for various domains*

Keywords: *Big data, Data processing, Cloud*

1. UVOD

U današnjem, digitalnom vremenu količina podataka koji su nam dostupni je sve veća. Sa porastom podataka raste i potreba za analizom istih radi izvođenja različitih zaključaka koji mogu doneti benefite. Činjenica je da je količina podataka u stalnom porastu i da će podataka koje treba obraditi biti sve više. Samim tim i rastu potrebe za unapređenjem sistema koji se bave obradom istih.

U ovom radu fokus je na jednom problemu obrade velikih količina podataka. Konkretno, u radu je prikazana analiza meteoroloških podataka koji se sakupljaju iz različitih izvora, transformišu se u odgovarajući format, vrši se analiza i na sonovu analize izvode određeni zaključci. Takođe, pripremićemo podatke na takav način da olakšamo buduće analize nad istima.

Tehnologije koje će biti korišćene u svrhu postizanja ovih rezultata su sledeće:

Azure functions [1] - koristićemo ih za prikupljanje podataka sa Interneta.

Azure data lake gen 2 [2] - koristićemo ga za čuvanje kako sirovih podataka tako i određenih podataka koji su pretvoreni u format prihvatljiviji za dalju obradu

Azure databricks [3] - će biti korišćen za pokretanje grupa kompjutera na kojima će se obrađivati podaci upotrebom spark-a.

SQL database [4] - u ovoj bazi podataka će se čuvati podaci u formatu adekvatnom za dalje analize u takozvanoj star/snowflake šemi podataka

PowerBI [5] - za prezentaciju podataka

2. PREGLED SLIČNIH SISTEMA

Sistemi za obradu velikih količina podataka su sve zastupljeniji. Podataka je sve više, obrada istih je veoma korisna i može doneti mnoge benefite. To su

- Veći profiti
- Unapređena sigurnost unutar sistema
- Identifikovanje šablona ponašanja kod korisnika
- Optimizacija troškova
- Unapređene performanse organizacije uviđanjem problematičnih tačaka
- Identifikovanje anomalija u realnom vremenu.

Sistemi koji obrađuju velike količine podataka su mnogobrojni i mnogi od njih nisu zasebni, već su deo već postojećih sistema koji se bave različitim problemima. Ovi sistemi se dele na:

- Sistemi koji obrađuju podatke u realnom vremenu (stream processing)
- Sistemi koji obrađuju skupove podataka (batch processing)

2.1. Sistemi koji obrađuju podatke u realnom vremenu

Sistemi koji obrađuju podatke u realnom vremenu dizajnirani su tako da prihvataju, obrađuju i isporučuju rezultate odmah nakon što podaci stignu. Ovi sistemi su ključni u scenarijima gde je potrebna trenutna obrada i reakcija na dolazne podatke. Karakteristika ovih sistema su:

Trenutna obrada - kako podaci stižu oni se odmah obrađuju, ove obrade su dosta prostije od obrada koje se rade u sistemima za obradu skupova podataka iz razloga što postoji vremensko ograničenje.

Minimalno kašnjenje - vremenski raspon između prijema i obrada podataka je obično od par milisekundi do najviše nekoliko sekundi.

Kontinualnost - sistem radi bez prekida.

2.2. Sistemi koji obrađuju podatke u skupovima

Kako su količine podataka kroz proteklih par decenija eksponencijalno porasle, potrebe za obradom istih podataka su sve veće. Moderni kompjuteri nisu u stanju da samostalno obrade ovakve količine podataka. Iz tog razloga su se javili novi sistemi koji obrađuju podatke u skupovima (batch processing systems). Ono što je karakteristično za ove sisteme je pre svega:

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Zarić, red. prof.

- Skalabilno procesiranje podataka - ovi sistemi mogu da rastu horizontalno (horisontal scaling) što znači da umesto da uvećavamo procesorsku moć jednog kompjutera mi umesto toga dodajemo veći broj mašina koji zajedno radi procesiranje podataka.
- Integracija podataka (Data integration) - Ovakvi sistemi obično obrađuju podatke iz različitih sistema, u različitim formatima i pre nekih ozbiljnijih obrada je uvek potrebno prvo očistiti i transformisati ove podatke
- Tolerancija na greške (Fault tolerance) - veoma je bitno da ovi sistemi budu tolerantni na greške iz razloga što često procesi znaju da traju i danima, te bi fatalna greška mogla da bude kobna i veoma skupa.
- Optimizovani - zbog mogućnosti horizontalnog rasta ovi sistemi su veoma dobro optimizovani da isti iskoriste najefikasnije, ukratko sve aktivnosti koje je moguće raditi paralelno ovi sistemi izvršavaju na taj način, ovo će biti detaljnije objašnjeno u nastavku.

3. KORIŠĆENE TEHNOLOGIJE

U ovoj sekciji dat je detačljniji pregledati tehnologije koje su korišćene za implementaciju našeg sistema.

3.1. Azure funkcije

Serverless računarstvo je model računarstva koji omogućava da se izvršavanje aplikacija radi bez brige o arhitekturi u pozadini uključujući servere, održavanje istih kao i skaliranje. Osnovne karakteristike Azure funkcija su:

- Reaguju na događaje
- Skalabilne
- Integrisana sigurnost
- Mogućnost pamćenja stanja

3.2. Azure data lake storage gen2

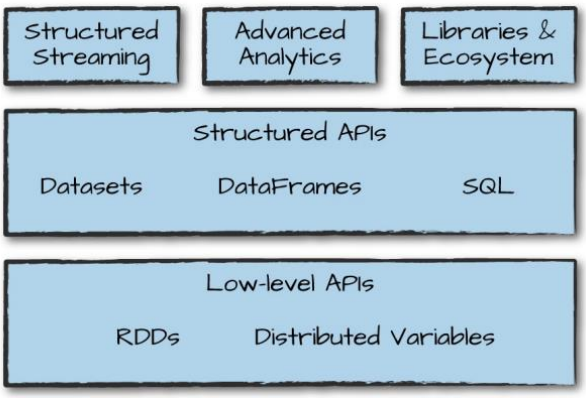
Azure data lake storage gen2 je unapređena verzija *Azure data lake storage gen1* sistema kao i *Azure Blob Storage* sistema. Kao rezultat ovoga dobijamo globalno skalabilan, siguran i visoko dostupan sistem za skladištenje podataka, koji nam omogućava da izvršavamo masivno paralelne analitike. Osnovne karakteristike Azure dana lake gen2 su:

- Hijerarhijska struktura direktorijuma
- Skalabilno i performantno
- Bezbednost na visokom nivou
- Redundantnost podataka (Lokalna, Zonska, Regijska)
- Adekvatni nivoi skladišta srazmerno

3.3. Apache spark

Spark je okruženje i kolekcija biblioteka za paralelno procesiranje podataka na grupama kompjutera, slika 1. U ovom momentu Spark je najpopularniji javno dostupan i razvijan sistem za paralelno procesiranje velikih količina podataka.

Apache Spark započeo je sa razvojem 2009. godine kao istraživački projekat na univerzitetu UC Berkley. U to vreme dominantna tehnologija za procesiranje velikih količina podataka je bio Hadoop Map Reduce. Tim koji je razvija Apache Spark je napravio jedan od najvećih pomaka u domenu obrade velikih količina podataka konceptom upotrebe RAM memorije zarad privremenog čuvanja podatka.



Slika 1. *Pregled kompletnog sistema spark-a*

3.3.1. Arhitektura Apache spark aplikacija

Apache Spark aplikacije sastoje se iz **driver** i **executor** procesa, slika 2.

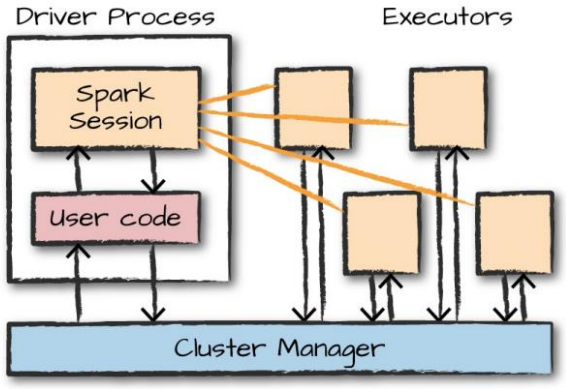
Driver proces je zadužen za pokretanje programa, nalazi se na jednom od kompjutera u grupi kompjutera, i zadužen je za tri stvari:

- Čuvanju informacija o Spark aplikaciji
- Komunikaciju sa korisnikom ukoliko u okviru procesa postoji potreba za time
- Analiziranju, raspoređivanju i zakazivanju procesa na ostalim kompjuterima u grupi kompjutera

Executor proces je zadužen za izvršavanje zadataka koje mu driver proces zada. Ovo znači da executor proces ima samo dva zadatka:

- da izvrši ono što mu se zada
- kada to uradi da javi rezultate driver procesu.

Grupom kompjutera koji će izvršavati Spark aplikaciju, upravlja cluster manager.



Slika 2. *Pregled izvršavanja Apache Spark aplikacije*

3.3.2. Apache spark Data frame

DataFrame je osnovna gradivna jedinica Spark API-ja. On predstavlja tabelu podataka sa redovima i kolonama. Ono što je bitno razumeti o DataFrame API-ju je to da je on distribuiran, odnosno DataFrame predstavlja logički naše

podatke, dok se oni u realnosti nalaze na grupi kompjutera razbacano. Podaci mogu biti podeljeni po kompjuterima po različitim parametrima i uslovima i ova podela se zove particionisanje podataka.

3.3.3. Apache spark Data frame

Da bi svaki executor mogao da izvršava svoj posao u paraleli, Spark razbija podatke u grupe koje se nazivaju particije. Particija je kolekcija redova podataka iz jednog DataFrame-a koji se nalaze na jednoj mašini.

Ukoliko imamo samo jednu particiju mogućnost paralelizma ne postoji. Ako imamo 5 particija podataka i 10 mašina, mogućnost paralelizma je 5, a ako pak imamo 100 particija i 10 kompjutera mogućnost paralelizma je 10. Ono što se iz ovoga može izvući je to da uvek treba imati barem onoliko particija koliko kompjutera se nalazi u našem skupu kompjutera, a često i više.

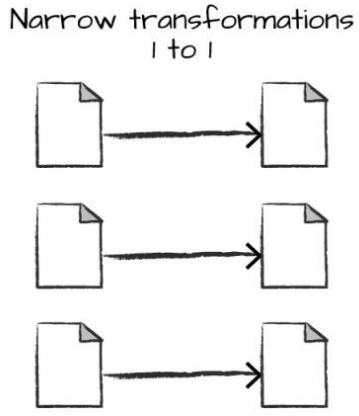
3.3.4. Transformacije nad podacima

Particionisanje je ono što omogućava da podatke obrađujemo paralelno, ali šta tačno znači obrada podataka?

Obrada podataka u okviru Spark-a i generalno u sferi obrade velikih količina podataka znači isto što i inače, to mogu biti transformacije određenih kolona, dodavanje novih kolona kombinacijom postojećih, spajanje više tabela podataka, filtriranje podataka po određenim parametrima i slično.

Ono što je bitno da razumemo u vezi transformacija nad podacima je da se one mogu podeliti u dve grupe: Narrow transformacije i Wide transformacije podataka.

Narrow transformacije podataka možemo izvršiti na jednom kompjuteru dok Wide transformacije najčešće zahtevaju da kombinujemo podatke sa različitih kompjutera u skupu kompjutera, slika 3.

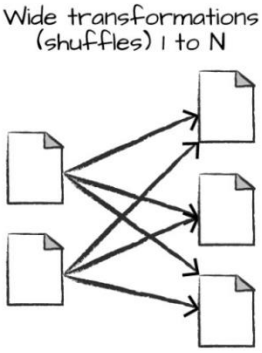


Slika 3. *Narrow transformacije*

Wide transformacije su komplikovanije, zato što one zahtevaju da se podaci sa različitih mašina sada nađu na jednoj, slika 4.

4. Sistem koji je implementiran

Zamisao ovog rada jeste da metereološke informacije objedinimo na jednom mestu i pripremimo ih za dalju obradu uz poštovanje najboljih praksi izrade ovakvih sistema.

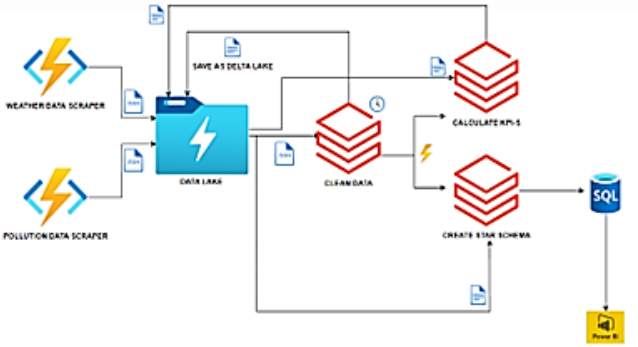


Slika 4. *Wide transformacije*

Takođe će biti demonstrirani neki od načina na koji ovi podaci mogu biti iskorišćeni ali to nije fokus ovog rada. Ove podatke treba da koriste domenski eksperti koji znaju tačno šta i kako žele da uvide, u okviru oblasti obrade velikih količina podataka, naš fokus je da im to omogućimo na što bolji i efikasniji način.

4.1 Weather data scrapper i pollution data scrapper

Uloga skupljača meteoroloških podataka (*weather data scrapper*) i skupljača podataka o zagađenju (*pollution data scrapper*) jeste ta da prikupljaju informacije o trenutnim vremenskim uslovima, kao i o zagađenjima na određenim lokacijama, slika 5. U našem slučaju to su različiti gradovi Srbije.



Slika 5. *Pregled sistema*

4.2 Data lake

Azure data lake gen 2 je rešenje za skladištenje velikih količina podataka, koje pruža Microsoft i mi ga koristimo za naš Data Lake. U njemu ćemo čuvati podatke u raznim formatima. Inicijalno se čuvaju podaci u formatu u kakvom se i dobavljaju. Kasnije, mi izvršavamo čišćenje ovih podataka kao i određene transformacije nad istima. Transformisane podatke, kao i određene kalkulacije koje želimo da budu javno dostupne, ćemo nakon obrade opet čuvati u našem Data Lake.

4.3 Clean dana

U ovom stadijumu našeg sistema vršimo čišćenje podataka kao i određene transformacije. Ovaj proces će se izvršavati na grupi kompjutera (cluster) koji će biti pokrenuti u okviru Azure databricks radnog prostora. Nakon ovih transformacija podaci će opet biti sačuvani nazad u Data Lake.

4.4 Calculate KPI's

U ovom segmentu sistema, koristeći očišćene podatke izvodimo određena standardna izračunavanja nad podacima. Ove podatke na kraju opet čuvamo u Data Lake gde im se može pristupiti javno, a i mogu se koristiti za kasnije analize i obrade.

4.5 Create star scheme

U ovom koraku se fokusiramo na kreiranje star scheme, koja predstavlja očišćene podatke u normalizovanoj formi, ovi podaci se zatim čuvaju u SQL bazi podataka u tabelama. Pored njih u bazi podataka se nalaze i dodatne tabele o kojima će kasnije biti više rečeno.

4.6 Power BI graph

Nakon čuvanja podataka u bazi podataka uz pomoć Power BI ćemo pokazati na koji način možemo iskoristiti podatke iz baze podataka za dalje i detaljnije analize.

5. ZAKLJUČAK

Konstantno povećavanje količine podataka u digitalnom svetu nas tera da nalazimo nove načine da obrađujemo ove podatke. Postoji sve više moćnih rešenja za procesiranje, sakupljanje, čuvanje, analizu i vizualizaciju istih. Microsoft Azure nudi integrisana rešenja koja se mogu iskoristiti efikasno u ove svrhe, kao što je i demonstrirano u dosadašnjem radu.

Uspešna integracija Azure funkcija, Azure Data Lake-a, Azure Databricks-a, Azure SQL-a i PowerBI-a pokazuje da je ovaj ekosistem veoma kvalitetan.

6. LITERATURA

[1] Azure functions - https://learn.microsoft.com/en-us/azure/azure-functions/
[2] Azure data lake gen 2 - https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction
[3] Azure databricks - https://learn.microsoft.com/en-us/azure/databricks/
[4] SQL Database - https://www.programiz.com/sql/database-introduction
[5] Power BI - https://learn.microsoft.com/en-us/power-bi/
[6] Apache Flink - https://nightlies.apache.org/flink/flink-docs-stable/
[7] Apache Spark (Spark streaming) - https://spark.apache.org/docs/latest/streaming-programming-guide.html
[8] Apache Spark - https://spark.apache.org/docs/3.4.1/

Biografija

Nikola Rončević rođen je 2. III 1998. godine u Novom Sadu. Završio je osnovnu školu Jovan Popović u Novom Sadu kao i Gimnaziju Jovan Jovanović Zmaj. Osnovne akademske studije je završio na Fakultetu Tehničkih Nauka u Novom Sadu sa prosekom 9.17.

Radi u firmi Inviggo kao programer poslednje tri godine gde se bavi sistemima za obradu velikih količina podataka kao i razvojem sistema za banke iz sfere otvorenog bankarstva.