Tehnologije Hadoop i Spark

Seminarski rad iz predmeta
Big Data u infrastrukturnim sistemima

Nataša Gavrilović E5 18/2023

Novi Sad, decembar, 2023.

Sadržaj

1. Uvod u Big Data

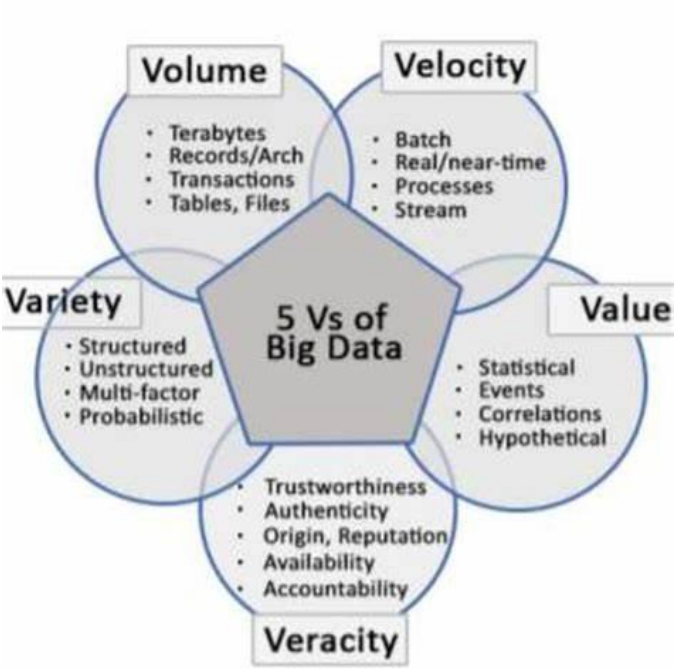
Pojam Big Data predstavlja velike, složene skupove podataka koji su toliko obimni da tradicionalni softver za obradu podataka jednostavno ne može da upravlja njima. Veliki podaci se odnose na izuzetno velike I raznolike kolekcije strukturiranih, nestrukturiranih I polusktrukturiranih podataka koji nastavljaju eksponencijalno da rastu tokom vremena. Te skupove podataka karakterišu raznovrsnost formata, velike brzine obrade i pristupa i veliki obim informacija. Izazovi uključuju projektovanje i realizaciju infrastrukture i servisa za skladištenje velikih količina podataka, njihovu pretragu, analizu, deljenje i vizualizaciju.

Definicije velikih podataka mogu se znatno razlikovati, ali će uvek biti opisane pomoću tri termina čije je početno slovo V na engleskom. Njih je prvo definisao Gartner 2001. godine. To su:

1. Obim (eng. Volume) – najčešća karakteristika povezana sa velikim podacima je njihov veliki obim. Opisuje ogromnu količinu podataka koji su dostupni za prikupljanje i različitih izvora i uređaja na kontinuiranoj osnovi.
2. Brzina (eng. Velocity) – odnosi se na brzinu kojom se podaci generišu. Danas se podaci često proizvode u realnom vremenu i stoga se moraju obrađivati, pristupati i analizirati istom brzinom da bi imali značajan uticaj.
3. Raznolikost (eng. Variety) – podaci su heterogeni, što znači da mogu doći iz mnogo različitih izvora i mogu biti struktuirani, nestruktuirani i polusktruktuirani. Tradicionalniji strukturirani podaci, kao što su podaci u tabelama ili relacionim bazama podataka, sada su dopunjeni nestrukturiranim tekstom, slikama, audio, video datotekama ili polustrkturiranim formatima kao što su podaci senzora koji se ne mogu organizovati u fiksnu šemu podataka.

Pored ova tri originalna V, još tri koja se često spominju u vezi sa iskorištavanjem moći velikih podataka su:

1. Istinitost (eng. Veracity) – podaci mogu biti neuredni, bučni ili skloni greškama što otežava kontrolu kvaliteta i tačnost podataka .Veliki skupovi mogu biti nezgrapni i zbunjujući, dok manji predstavljaju nepotpunu sliku. Što je veća verodostojnost podataka, to su tačniji.
2. Varijabilnost (eng. Variability) – značenje prikupljenih podataka se stalno menja, što može dovesti do nedoslednosti tokom vremena. Ove promene uključuju ne samo promene u kontekstu i tumačenju, već metode prikupljanja podataka na osnovu informacija koje kompanije žele da sakupe i analiziraju.
3. Vrednost (eng. Value) – od suštinskog je značaja da se odredi poslovna vrednost podataka koje prukupljamo. Veliki podaci moraju da sadrže prave podatke, a zatim da budu efikasno analizirani kako bi se došlo do uvida koji mogu pomoći ponošenju odluka.



Slika 1 – 5V model

Podaci mogu biti najvrednija imovina kompanije. Neki od primera velikih podataka koji pomažu u transformaciji organizacija u svakoj industriji:

1. Praćenje ponašanja potrošača i navika kupovine radi isporučivanja hiperpesonalizovanih preporuka maloprodajnih proizvoda prilagođenih pojedičnim kupcima.
2. Praćenje obrazaca plaćanja i njihovo analiziranje u odnosu na istorijske aktivnosti klijenata kako bi se otkrile prevare u realnom vremenu
3. Kombinovanje podataka i informacija iz svake faze putovanja isporuke porudžbine sa hiperlokalnim uvidom u saobraćaj kako bi se pomoglo operaterima da optimizuju isporuku do poslednjeg kilometra.
4. Korišćenje tehnologija zasnovanih na veštačkoj inteligenciji kao što je obrada prirodnog jezika za analizu nestruktuiranih medicinskih podataka kako bismo stekli nove uvide za bolji razvoj lečenja u poboljšanju negu pacijenata.
5. Korišćenje podataka o slikama sa kamera i senzora kao i GPS podataka, za otkrivanje rupa i poboljšavanje održavanja puteva u gradovima
6. Analiziranje javnih skupova podataka satelitskih snimaka i geoprostornih podataka za vizuelizaciju, praćenje, merenje i predviđanje društvenih i ekoloških uticaja operacija lanca snabdevanja.

Centralni koncept je da što je veća vidljivost, to se efikasnije može steći uvid u donošenje boljih odluka, otkrivanje mogućnosti za rast i poboljšanje poslovnog modela. Da bi big data podaci bili funkcionalni, oni moraju zahtevati tri glavne radnje, a to su integracija, upravljanje i analiza. Veliki podaci prikupljaju terabajte, a ponekad čak i petabajte, neobrađenih podataka iz mnogih izvora koji se moraju primiti, obraditi i transformisati u format koji je potreban poslovnim korisnicima i analitičarima da počnu da ih analiziraju. Za velika podatke je

potrebno veliko skladište, bilo u cloudu, lokalno ili i jedno i drugo. Podaci se takođe moraju čuvati u bilo kom obliku. Takođe ga treba obraditi i učiniti dostupnim u realnom vremenu. Poslednji korak je analiza i delovanje na osnovu velikih podataka, u suprotnom investicija se neće isplatiti. Osim istraživanja samih podataka, takođe je ključno komunicirati i deliti uvide u poslovanje na način koji svako može da razume. Ovo uključuje korišćenje alata za kreiranje podataka kao što su grafikoni i kontrolne table.

Neki od benefita Big Data su poboljšano donošenje odluka. Kada je moguće upravljati velikim podacima i analizirati ih, može se otkriti obrazac i otključati uvidi koji poboljšavaju i podstiču bolje operativne i strateške odluke. Što se tiče povećane agilnosti i inovativnosti, Big data omogućava i prikupljanje i obrađivanje podataka u realnom vremenu i analiziranje kako bi se brzo prilagodili i stekli konkurentsku prednost. Ovi uvidi mogu da ubrzaju planiranje, proizvodnju i lansiranje novih proizvoda, funkcija i ažuriranja. Bolje korisničko iskustvo, kombinovanje i analiza strukturiranih izvora podataka zajedno sa nestrukturiranim pruža korisnije uvide za razumevanje potrošača, personalizaciju i načine za optimizaciju iskustva radi zadovoljavanja potrebe i očekivanja potrošača. Omogućavaju da se integriše automatizovani prenos podataka u realnom vremenu sa naprednom analitikom podataka kako bi se kontinuirano prikupljali podaci, pronašli novi uvidi i otkrili nove mogućnosti za rast i vrednost. Korišćenje alata za analizu velikih podataka omogućava nam da brže obrađujemo podatke i generišemo uvide koji mogu pomoći da odredimo oblasti u kojima možemo smanjiti troškove, uštedimo vreme i povećamo ukupno efikasnost. Poboljšano upravljanje rizikom, analiza ogromnih količina podataka pomaže kompanijama da bolje procene rizik, što olakšava identifikaciju i praćenje svih potencijalnih pretnji i izveštavanje o uvidima koji vode ka snažnoj kontroli strategija i ublažavanja.

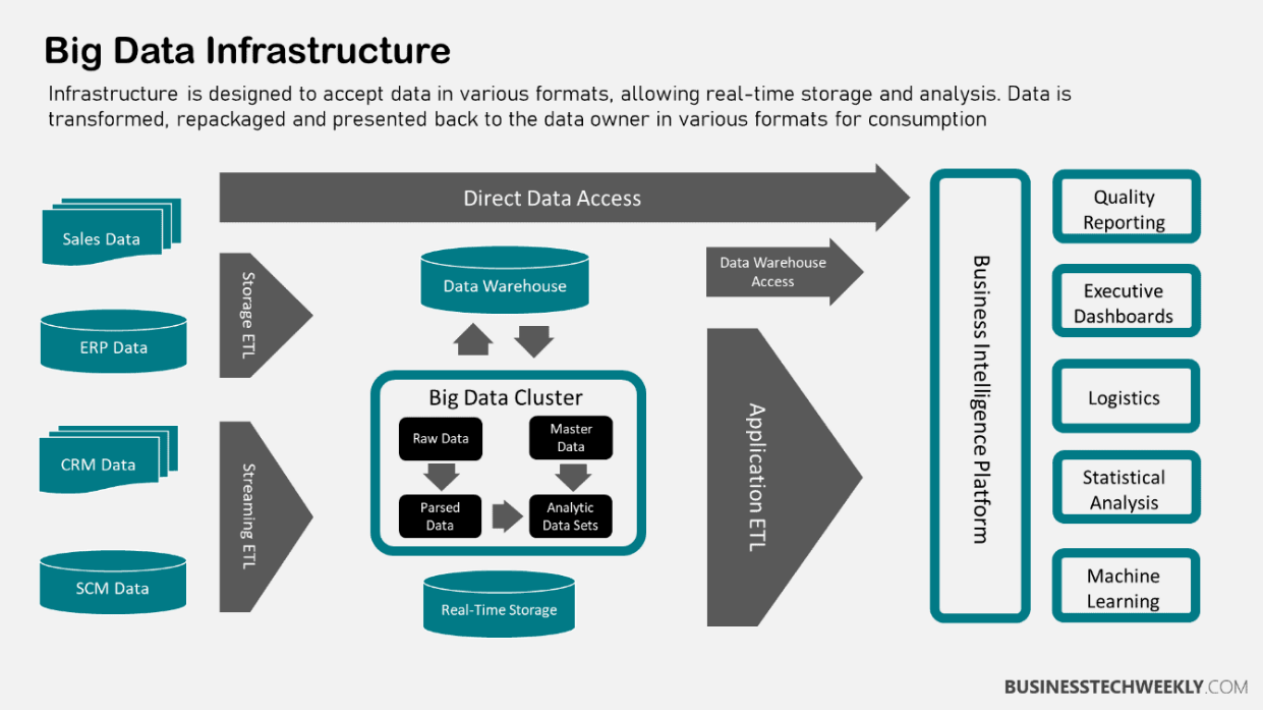
Iako Big data imaju mnoge prednosti, oni predstavljaju neke izazove sa Kojima organizacije moraju biti spremne da se pozabave kada prikupljaju, upravljaju I preduzimaju akcije na tako ogromnoj količini podataka. Najčešće prijavljeni izazovi velikih podataka uključuju brzinu rasta podataka. Po prirodi, uvek se brzo menjanju I povećavaju. Bez čvrste infrastructure koja može da se nosi sa potrebama obrade, skladištenja I mreže bezbednosti, može postati izuzetno teško upravljati. Kvalitet podataka direktno utiče na kvalitet donošenja odluka, analize podataka I strategija planiranje. Posedovanje velikih podataka ne garantuje rezultate osim ako podaci nisu tačni, relevantni I pravilno organizovani za analizu. Ovo može da uspori izveštavanje. Veliki podaci sadrže mnogo osetljivih podataka I informacija, zbog čega je težak zadatak da se kontinuirano obezbedi da obrada I skladištenje podataka ispunjavaju privatnost podataka I regulatorne zahteve, kao što su lokalizacija podataka I zakoni o rezidentnosti podataka. Većina kompanija rad isa podacima raspoređenim u različitim sistemima I aplikacijama širom organizacije. Integrisanje različitih izvora podataka I omogućavanje pristupa podacima za poslovne korisnike je složeno, ali je od vitalnog značaja. Oni sadrže vredne poslovne informacije I informacije o klijentima, čineći visoko vrednim metama za napadače. Pošto su skupovi podataka raznovrsni, može biti teže primeniti sveobuhvatne strategije za njihovu zaštitu.

Iako je sam concept Big Date relative nov, poreklo velikih skupova podataka seže u šezdesete I sedamdesete godine proškog veka kada je svet podataka tek počeo sa prvim centrima podataka I razvojem relacione baze podataka. Oko 2005. godine ljudi su počeli da shvataju koliko podataka su korisnici generisali preko Fejsbuka, Jutjuba I drugih onlajn servisa. Hadoop je razvijen iste godine. NoSQL je takođe počeo da dobija na popularnosti tokom ovog vremena. Razvoj Hadoopa, kao I od skoro Sparka, bio je od suštinskog značaja za rast velikih podataka jer čine rad sa velikim podacima lakšim I jeftinijim za skladištenje. U godinama od tada, obim velikih podataka je naglo porastao. Korisnici I dalje generišu ogromne količine podataka. Iakos u veliki podaci daleko dostigli, njihova korisnost tek počinje. Računarstvo u cloudu je još više proširilo mogućnosti velikih podataka. Cloud nudi zaista elastičnu skalabilnost, gde programeri mogu jednostavno da pokrenu klastere da testiraju podskup podataka. I baze podataka grafikona takođe postaju sve važnije, sa njihovom sposobnošću da prikažu ogromne količine podataka na način koji analitiku čini brzom I sveobuhvatnom.

Big Data funkcioniše tako što uključuje tri ključne radnje:

1. Integrisanje – Veliki podaci objedinjuju podatke iz mnogih različitih izvora I aplikacija. Tradicionalni mehanizmi integracije podataka, kao što su estrahovanje, transformacija I učitavanje generalno nisu dorasli zadatku. Potrebne su nove strategije I tehnologije za analitu velikih skupova podataka u terabajtnoj ili čak petabajtnoj skali. Tokom integracije potrebno je da se unesu podaci, da se obrade I uveriti se da su formatirani I dostupni obliku kojim analitičari mogu da počnu.
2. Upravljanje – zahtevaju skladištenje. To rešenje može biti u cloudu, lokalno ili oboje. Može se skladištiti u bilo kojem obliku. Mnogi ljudi biraju rešenje za skladištenje prema tome gde se trenutno nalaze. Cloud postebeno dobija na popularnosti jer podržava trenutne računarske zahteve I omogućava povećavanje resursa po potrebi.
3. Analiziranje – Ulaganje u big data se isplati kada se analizira. Dobije se nova jasnoća pomoću vizuelne analize različitih skupova podataka.

Infrastruktura big data je IT infrastruktura koja sadrži velike podatke. To je kritičan deo ekosistema podataka koji objedinjuje različite alate i tehnologije koje se koriste za rukovanje podacima tokom njihovog životnog ciklusa, od prikupljanja i skladištenja do analiza i rezervnih kopija.



Slika 2. ***Infrastruktura Big Data***

Na slici 2 prikazana je infrastruktura velikih podataka koja je dizajnirana da prihvata podatke u različitim formatima, omogućavajući skladištenje i analizu u realnom vremenu. Podaci se transformišu, ponovo pakuju i vraćaju vlasniku podataka u različitim formatima za upotrebu.

2. Hadoop

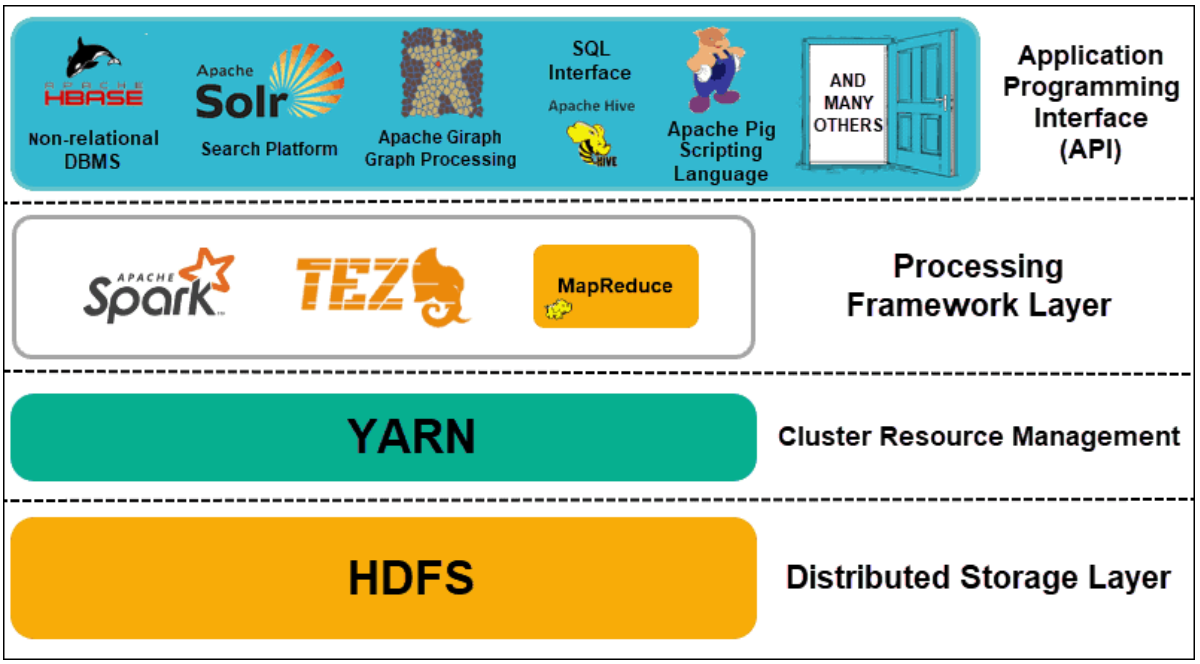
Jedno od rešenja za infrastrukturu Big data je Hadoop. Big Data pruža mogućnost obrade podataka u realnom vremenu, a pretraga se vrši korišćenjem Map Reduce algoritma koji je u sklopu Hadoopa. Na primer, rezultati pretrage u Gugl pretraživaču se dobijaju u milisekundama upravo zahvaljujući ovim tehnologijama. Hadoop je open source biblioteka koja se koristi za efektivno skladištenje i obradu velikih skupova podataka veličine od gigabajta do petabajta podataka. Umesto da koristi jedan veliki računar za skladištenje i obradu podataka, Hadoop omogućava grupisanje više računara da bi se brže analizirali masivni skupovi podataka paralelno. Hadoop se sastoji od 4 glavna modula:

1. Hadoop distribuirani system datoteka (**Hadoop Distributed File System - HDFS**) – distribuirani system datoteka koji radi na sandardnom ili jeftinom hardveru. Pruža bolju propusnost podataka od tradicionalnih sistema datoteka, pored visoke tolerancije grešaka za velike skupove podataka. Podaci se čuvaju u pojedinačnim blokovima podataka u tri odvojene kopije na više čvorova i softverskih rekova. Ako jedan čvor ili rek otkaže, uticaj na šiti sistem je zanemarljiv. Data Nodes obrađuje i čuva blokove podataka, dok Name Nodes upravljaju mnogih Data Nodes, održavaju metapodatke blokova i kontrolišu pristup klijenta.

2. Pregovarač resursa (Yet Another Resource Negotiator - **YARN**) – upravlja I nadgleda čvorove klastera I korišćenje resursa. Planira poslove I zadatke. U prethodnim Hadoop verzijama, Map Reduce se koristio i za obradu podataka i dodelu resursa. Vremenom je neophodnost razdvajanja obrade i upravljanja resursima dovela do razvoja YARNa.
3. **Map Reduce** – framework koji pomaže programima da rade paralelno računanje podataka. Zadatak je da uzima ulazne podatke I pretvara ih u skup podataka koji se može izračunati u parovima vrednosti ključeva. Izlaz funkcije se troši zadacima redukcije da bi se agregirali izlaz I pružio željeni rezultat.
4. **Hadoop Common** – pruža zajedničke Java biblioteke koje se mogu koristiti u svim modulima.

Osnovni princip rada ove tehnologije je da se podaci razbijaju i distribuiraju u delove I analiziraju I obrađuju različite delove istrovremeno, umesto da se obrađuje jedan l blok podataka. Hadup pruža ogromnu memoriju podataka. Primarni benefiti ove tehnologije su:

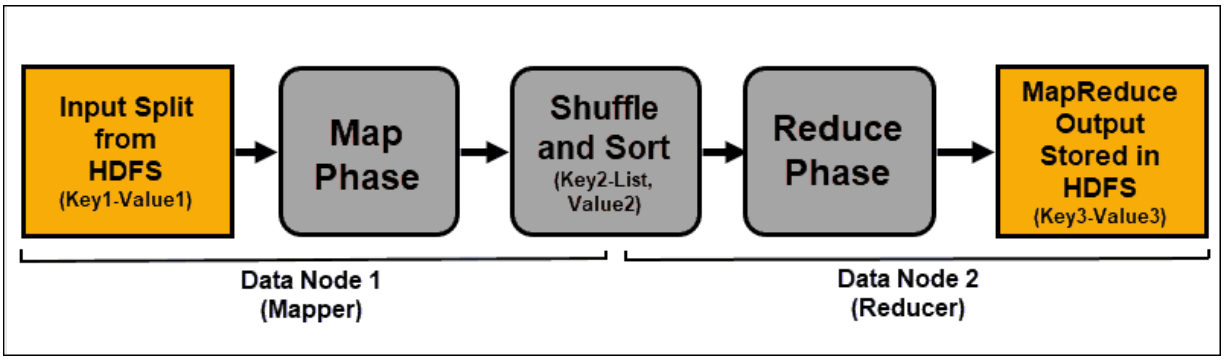
1. Fleksibilnost – za razliku od relacionih baza podataka, nema potrebe za obradom podataka pre skladištenja. Hadup omogućava skladištenje onoliko podataka Koliko želimo. Ovo takođe ukjlučuje nestrukturirane podatke, kao što su slike, video zapisi I tekst.
2. Brzina – pošto istovremeno obrađuje više delova skupa podataka, to je značajno brz alat za dubinsku analizu podataka
3. Skalabilnost – radi u distribuiranom okruženju I skalabilan je, lako je razviti sistem da obrađuje više podataka jednostavnim dodavanjem više čvorova
4. Niska cena – Hadup je besplatan otvoreni okvir
5. Elastičnost – otporan je, podaci uskladišteni na čvoru se repliciraju na druge čvorove klastera da bi se nosili sa hardverskim kvarovima. Takav dizajn obezbeđuje toleranciju grešaka, uvek je dostupna rezervna kopija ako je čvor u kvaru.



Slika 3. ***Prikaz Hadoop Arhitekture***

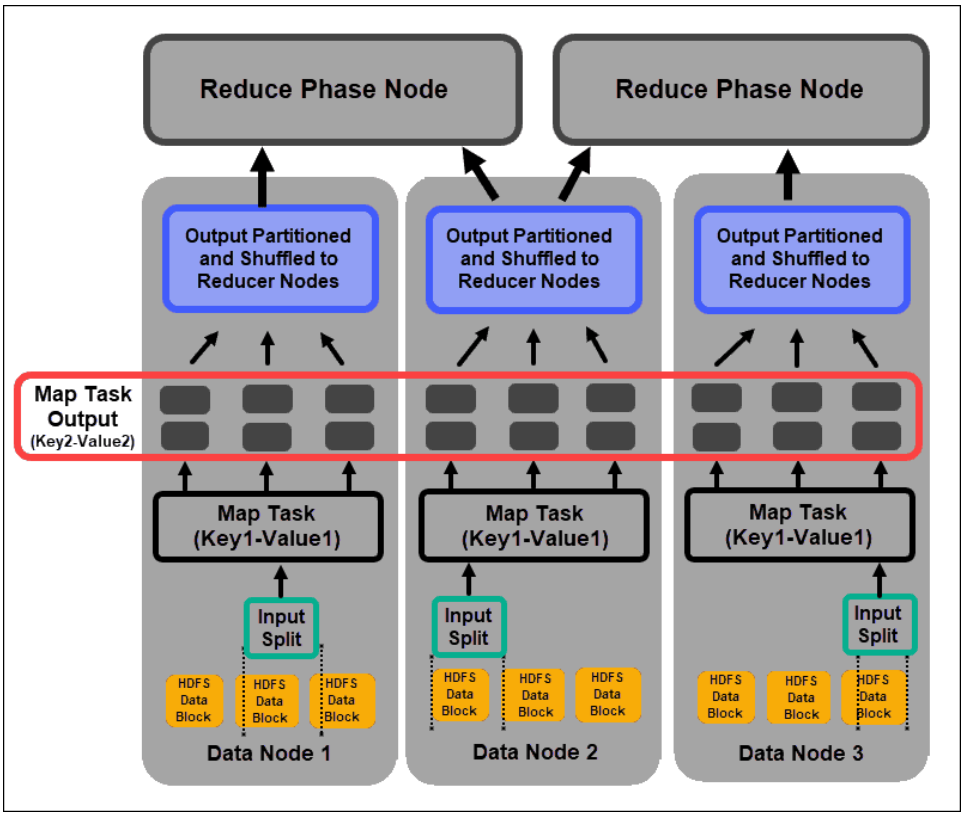
Na slici 3 prikazano je razdvajanje elemenata distribuiranih sistema u funkcionalne slojeve koji pomažu da se pojednostavi upravljanje podacima i razvoj. Hadoop se može podeliti na četiri karakteristična sloja:

1. **Distribute Storage Layer** – svaki čvor u klasteru ima sopstveni prostor na disku, memoriju, propusni opseg i obradu. Dolazni podaci se dele na pojedinačne blokove podataka, koji se zatim čuvaju unutar HDFSa distribuiranog skladišnog sloja. HDFS pretpostavlja da su svaki drajv disk nepouzdani. Kao mera predostrožnosti, on skladišti tri kopije svakog skupa podataka u celom klasteru. Glavni čvor (eng. NameNode) čuva metapodatke za pojedinačni blok podataka i sve njegove replike.
2. **Cluster Resource Management** – Hadup treba savršeno da koordinira čvorove tako da bezbroj aplikacija i korisnika efikasno deli svoje resurse. U početku MapReduce se bavio upravljanjem resursima i obradom podataka. YARN razdvaja ove dve funkcije. Kao de-fakto alat za upravljanje resursima za Hadoop, YARN je sada u mogućnosti da dodeli resurse različitim okvirima napisanim za Hadoop.
3. **Processing Framework Layer** – sloj za obradu se sastoji iz okvira koji analiziraju i obrađuju skupove podataka koji dolaze u klaster. Strukturirani i nestrukturirani skupovi podataka se mapiraju, mešaju, sortiraju, spajaju i redukuju u manje blokove podataka kojima se može upravljati. Ove operacije su raspoređene na više čvorova što je bliže moguće serverima na kojima se nalaze podaci.
4. **Application Programming Interface** – uvođenjem YARNa dovelo je do stvaranja novih okvira za obradu APIja. Projekti koji se fokusiraju na platforme za pretragu, strimovanje podataka, interfejse prilagođene korisniku, programske jezike, razmenu poruka su složeni deo sveobuhvatnog ekosistema.



Slika 4. MapReduce algoritam

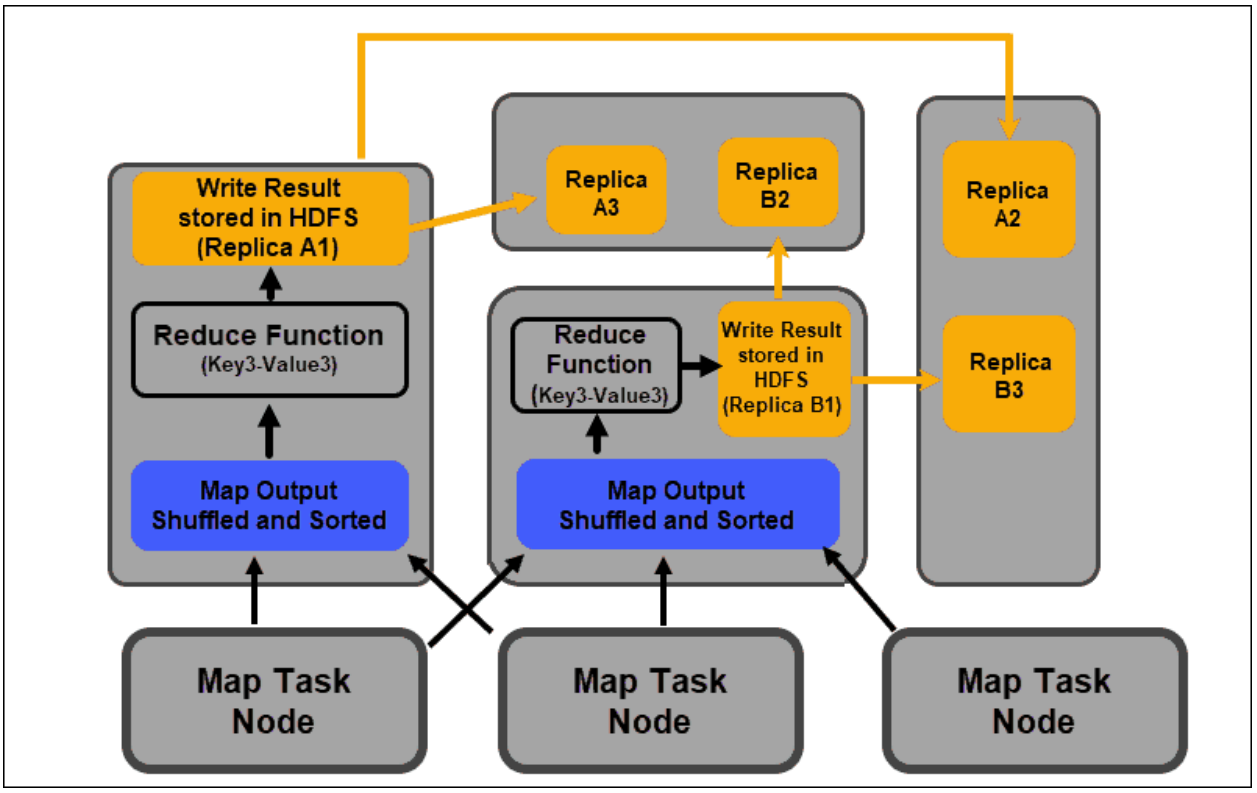
Na slici 4 prikazan je MapReduce algoritam za programiranje koji obrađuje podatke raspoređene po Hadoop klasteru. Kao i kod svakog procesa u Hadoopu, kada se MapReduce pokrene, resourceManager zahteva od Master Application da upravlja i prati životni ciklus MapReduce posla. Resource Manager kontroliše sve resurse obrade u klasteru i njegova primarna svrha je da odredi resurse pojedinačnim aplikacijama koje se nalaze na određenim čvorovima. Održava globalni pregled tekućih i planiranih procesa, rukuje zahtevima za resurse i u skladu sa tim raspoređuje i dodeljuje resurse. Od vitalnog je značaja za Hadoop i trebalo bi da radi na namenskom glavnom čvoru. Hadoop serveri koji obavljaju zadatke mapiranja i redukcije često se nazivaju maperi i reduseri. Resource Manager odlučuje koliko će mapera koristiti. Ova odluka zavisi od veličine obrađivanih podataka i memorijskog bloga koji je dostupan na svakom serveru za mapiranje.



Slika 5. Map faza

Na slici 5. prikazan je proces mapiranja, koji uzima pojedinačne logičke izraze podataka uskladištenih u blokovima podataka HDFS. Ovi izrazi mogu obuhvatati nekoliko blokova podataka i nazivaju se podelama ulaza. Ulazne podele se uvode u proces mapiranja kao parovi ključ - vrednost. Zadatak mapiranja je da prolazi kroz svaki par ključ - vrednosti i da kreira novi skup parova ključ - vrednost, različit od originalnih ulaznih podataka. Kompletan asortiman svih parova predstvlja rezultat zadatka mapiranja. Na osnovu ključa iz svakog para, podaci se grupišu, particioniraju i mešaju u čvorove reduktora.

Sledeća faza je Shuffle proces u kome se rezultati svih zadatka mape kopiraju u čvorove reduktora. Kopiranje izlaza zadatka maps je jedina razmena podataka između čvorova tokom celog Map Reduce posla. Izlaz zadatka mape treba da bude uređen da bi se poboljšala efikasnost faze smanjenja. Preslikani parovi ključ - vrednost, koji se mešaju iz čvorova mapera, poređani su po ključu sa odgovarajućim vrednostima. Faza smanjenja počinje nakon što se unos sortira po ključu u jednoj ulaznoj datoteci. Faze nasumice i sortiranja se odvijaju paralelno. Čak i kada se izlazi iz maps, preuzimaju iz čvorova mapera, oni se grupišu i sortiraju na čvorovima reduktora.



Slika 6. Reduce faza

Izlazi maps se mešaju i sortiraju u jednu ulaznu datoteku za smanjenje koja se nalazi u čvoru reduktora. Funkcija redukcije koristi ulaznu datoteku da agregira vrednosti na osnovu odgovarajućih mapiranih ključeva. Izlaz iz procesa redukcije je novi par ključ - vrednost. Ovaj rezultat predstavlja izlaz celog Map Reduce posla i podrazumevano se čuva u HDFS. Svi redukcioni zadaci se odvijaju istovremeno i rade nezavisno jedan od drugog. Zadatak smanjenja

je takođe opcioni. Mogu postojati slučajevi u kojima je rezultat zadatka mape željeni željeni rezultat i nema potrebe da se proizvede jedna izlazna vrednost.

3. Spark

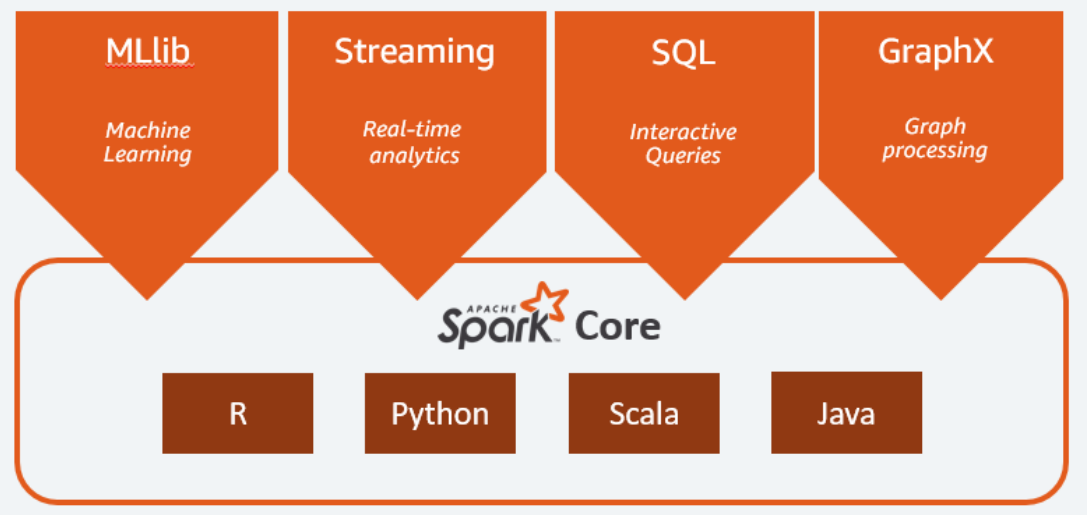
Brz, fleksibilan i pogodan za programere. Apache Spark je vodeća platforma za SQL, grupnu obradu, stream obradu i mašinsko učenje. To je okvir za obradu podataka koji može brzo da obavlja zadatke obrade na veoma velikim skupovima podataka, a takođe može da distribuira zadatke obrade podataka na više računara, bilo samostalno ili u tandemu sa drugim distribuiranim računarskim alatima. Spark takođe preuzima deo programskog tereta ovih zadataka sa ramenima programera pomoću APIja koji je jednostavan za korišćenje koji apstrahuje veći deo napornog rada distribuiranog računarstva i obrade velikih podataka. Spark je 2009. godine postao jedan od ključnih okvira za distribuiranu obradu velikih podataka u svetu, Može se primeniti na različite načine, obezbeđuje izvorne veze za programske jezike kao što su Java , Scala, Python i R, koji podržavaju SQL, strimovanje podataka i mašinsko učenje i obradu grafova. Cilj Sparka je bio da stvori novi okvir, optimizovan za brzu iterativnu obradu kao što je mašinsko učenje i interaktivna analiza podataka, uz zadržavanje skalabilnosti i tolerancije grešaka Hadoop Map Reduce.

Hadoop Map Reduce je programski model za obradu podataka sa paralelnim, distribuiranim algoritmom. Međutim, izazov je sekvencijalni proces u više koraka koji je potreban za pokretanje posla. Sa svakim korakom, Map Reduce čita podatke iz klastera iz klastera, izvodi operacije i zapisuje nazad rezultat u HDFS. Pošto svaki korak zahteva čitanje i pisanje sa diska, Map Reduce poslovi su sporiji zbog kašnjenja diska. Spark je kreiran da odgovori na ograničenja za Map Reduce tako što obrađuje u memoriji, smanjuje broj koraka u poslu i ponovo koristi podatke u više paralelnih operacija. Sa Sparkom, potreban je samo jedan korak gde se podaci čitaju u memoriju, izvršavaju operacije i vraćaju rezultati, što rezultira mnogo bržim izvršavanjem. Spark takođe ponovo koristi podatke koristeći keš memoriju da bi u velikoj meri ubzao agoritme mašinskog učenja koji više puta pozivaju funkciju na istom skupu podataka. Ponovna upotreba podataka se postiže kreiranje Data Framea, apstrakcije preko otpornog distribuiranog skupa podataka (RDD). Ovo dramatično smanjuje kašnjenje čineći Spark višesturko bržim od Map Reduce, posebno kada se radi o mašinskom učenju i interaktivnoj analitici. Osim razlika u dizajnu Spark i Hadoop Map Reduce mnoge organizacije su otkrile da su ovi okviri velikih podataka besplatni, koristeći ih zajedno za rešavanje šireg poslobnog izazova. Spark je fokusiran na interaktivne upite i radna opterećenja u realnom vremenu. Nema sopstveni sistem za skladištenje , ali radi analitiku na drugim sistemima za skladištenje kao što je HDFS ili drugi popularnim prodavnicama kao što su Amazon, Kasandra itd. Spark na Hadoopu koristi Yarn da deli zajednički klaster i skup podataka kao i drugi Hadoop motori, obezbeđujući dosledne nivoe usluge i odgovora.

U srcu samog sparka je concept otpornog distribuiranog skupa podataka (eng. Resilient distributed Dataset, RDD), programske apstrakcije koja predstavlja nepromenljivu kolekciju objekata koji se mogu podeliti na računarski klaster. Operacije na RDD-ovima se takođe mogu

podeliti na klaster I izvršiti u paralelnom batch procesu, što dovodi do brzog I skalabilnog paralelnog procesa. On pretvara korisničke komande za obradu podataka u usmereni aciklični grad (eng. Directed Acyclic Graph, DAG). DAG je sloj za planiranje Sparka, on određuje koji zadaci će se izvršavati na kojim čvorovima I kojim redosledom.

RDD- ovi se mogu kreirati od jednostavnijih tekstualnih datoteka, SQL baza podataka, noSQL storova, kao što su Kasandra I MongoDB, Amazon S3 buckets I slično. Veliki API Spark Core je izgrađen na ovom RDD konceptu, omogućavajući tradicionalnu mapu I smanjenje funkcionalnost, ali takođe pružajući ugrađenu podršku za spajanje skupova podataka, filtriranje, uzrokovanje I agregaciju. Spark radi na distribuiran način, kombinovanjem procesa jezgra drajvera koji deli Spark aplikaciju na zadatke I distribuira ih među mnoge izvršne procese koji obavljaju posao. Ovi izvršioci se mogu povećavati I smanjivati prema potrebi za potrebe aplikacije.



Slika 7. Spark Workloads

Spark uključuje Spark Core kao osnova za platformu, Spark SQL za interaktivne upite, Spark Streaming za analiziku u realnom vremenu, Spark MLlib za mašinsko učenje i Spark GraphX za obradu grafova.

1. Spark Core je osnova platforme. Odgovoran je za upravljanje memorijom, otklanjanje grešaka, planiranje, distribuciju i nadgledanje poslova i interakciju sa sistemima za skladištenje podataka. Izložen je kroz interfejs za programiranje aplikacija (API) izgrađen za Javu, Skalu, Pajton i R. Ovi Apiji skrivaju složenost distribuirane obrade za jednostavnih operatora visokog nivoa.
2. MLlib je biblioteka algoritama za mašinsko učenje podataka u velikom obimu. Modele mašinskog učenja mogu da obuče sa R ili Pajtonom na bilo kom Hadoop izvoru podataka, sačuvani pomoću MLliba i uvezani u cevovod zasnovan na Javi i Skali. Dizajniran za brzo i interaktivno računanje koje se pokreće u memoriji, omogućavaju brzo pokretanje mašinskog učenja. Algoritmi iključuju mogućnost klasifikacije, regresije, grupisanja, kolaborativnog filtriranja i rudarenja šablona.
3. Spark Striming je rešenje u realnom vremenu koji koristi mogućnosti brzog zakazivanja Spark Core za analizu striminga. Unosi podatke u mini serije i omogućava analizu tih

podataka sa istim kodom aplikacije napisanim za grupnu analitiku. Ovo poboljšava produktivnost programera, jer oni mogu da koriste isti kod za grupnu obradu i za apliakcije za striming u realnom vremenu. Spark Striming podržava podatke sa Tvitera, Kafke, Flumea, HDFDs itd.

4. Spark GraphX je distribuirani okvir za obradu grafova izgrađen na vrhu Sparka. Obezbeđuje ETL, istraživačku analizu i iterativno izračunavanje grafa kako bi omogučio korisnicima da interaktivno grade i transformišu strukturu podataka grafikonima u razmeri. Dolazi sa veoma fleksibilnim APIjem i izborom distribuiranih grraf algoritama.
5. SparkSQL je interfejs koji današnji programeri najčešće koriste prilikom kreiranja aplikacija. On je fokusiran na obradu strukturiranih podataka, koristeći pristup okvira podataka koji je pozajmljen od R I Pythona (pandas biblioteka). Sugeriše da takođe obezbeđuje interfejs kompatibilan SQL2003 za upite podataka. Pored standardne SQL podrške, Spark SQL obezbešuje standardni interfejs za čitanje I pisanje u druga skladišta podataka uključujući JSOn, JDBS itd. Ostala popularna skladišta podataka su Apache Kasandra, MongoDB I mogu se koristiti uvlačenjem zasebnih konektora iz ekosistema Spark Packages. Omogućava transparentno korišćenje korisnički definisanih funkcija (User-Defines Functions, UDFs) u SQL upitima. Primer izbora kolona iz okvira podataka prikazan je u listingu 1.

.

```
citiesDF.select("name", "pop")
Cities.createOrReplaceTempView(„cities")
Spark.sql(„SELECT name, pop FROM cities")
```

Listing 1 Primer SPARKQL Koda

Koristeći SQL interfejs, registruje se okvir podataka kao privremena tabela, nakon čega se izdaju SQL upiti. Apache Spark koristi optimizator upita pod nazivamo Katalist (eng. Catalyst), koji ispituje podatke i upite kako bi proizveo efikasan plan upita za lokalizaciju podataka i izračunavanje koji će izvršiti potrebne proračune širom klastera. Od Apache Spark 2.x , Spark SQL interfejs okvira podataka i skupova podataka, u suštini otkucani okvir podataka koji se može proveriti u vreme kompajliranja da li je ispravan , bio je preporučen pristup za razvoj. RDD interfejs je i dalje dostupan, ali se preporučuje ako se nešto ne može rešiti u okviru Spark SQL paradigme.

Industrijski standard za manipulaciju podacima i analizu u Pajton programsku jeziku je Pandas biblioteka. Kod Apache Spark 3.2 verzije, obezbeđen je novi API koji omogućava da se veliki deo pandas APIja koristi transaprentno sa Sparkom.

Na fundamentalnom nivou, Spark aplikacija se sastoji od dve glavne komponente: drajvera, koji konvertuje korisnički kod u više zadataka koji se mogu distribuirati po radničkim čvorovima, i izvršilaca koji se pokreću na tim radničkim čvorovima i izvršavaju zadatke koji su im dodeljeni. Može da radi u samostalnom režimu klastera, koji jednostavno zahteva Apache Spark okvir i Javu virtualnu mašinu na svakom čvoru u klasteru. Ovo je istorijski značilo da se radi na Hadoop YARNu, ali kako je Hadoop postao manje ukorenjen, sve više i više kompanija se

okreće primeni Apache Sparka na kubernetesu. Ovo se odrazilo u izdanjima Apache Spark 3.x koja poboljšavaju integraciju sa Kubernetesom, uključujući definisanja pod šablon za drajvere i izvršioce i korišćenje prilagođenih planera kao što su Volcano. Spark se može naći na sva tri velika clouda, a to su Amazon ERM, Azure HDInsight i Google Cloud Dataproc.

6. Zaključak

U današnjem dobu digitalne transformacije, količina podataka koja se generiše svakodnevno eksponencijalno raste. Upravljnje ovim ogromnim volumenom podataka postalo je ključni izazov za ogranizacije širom sveta. Tehnologije poput Apache Hadoop, Apache Spark i uopšteno Big Data igraju ključnu ulogu u rešavanju ovog izazova. Hadoop, kao distribuirani sistem za skladištenje i obradu velikih podataka, omogućava organizacijama da skaliraju svoje kapacitete za skladištenje i analizu podataka u skladu sa njihovim potrebama. Kroz Map Reduce paradigmu, Hadoop pruža efikasno rešenje za obradu podataka na klasteru računara, što omogućava bržu analizu i donošenje odluka zasnovanim na podacima. Sa druge strane, Spark predstavlja evoluciju u obradi podataka u velikim merama. Njegova sposobnost obrade podataka u memoriji i podrška za različite analitičke zadatke čine ga moćnim alatom za naprednu analizu podataka. Spark takođe olakšava intragraciju sa drugim tehnologijama i podržava rad u stvarnom vremenu, čime organizacijama pruža fleksibilnost i brzinu odgovora na dinamička poslovna pitanja. U kontekstu infrastrukturnih sistema, implementacija Hadoopa i Sparka omogučava organizacijama da izgrade snažnu i skalabilnu arhitekturu za obradu i analizu Big data. Integracija obih tehnologija olakšava plaćenje, analizu i izvlačenje vrednih informacija iz podataka, stvarajući osnovu za donošenje informisanih poslovnih odluka. Tehnologije Big data, kao što su Hadoop i Spark, nisu samo rešenje za trenutne izazove u vezi sa obimom podataka, već i osnova za budući napredak u analizi podataka .Kroz kontinualno usavršavanje i prilagođavanje, organizacije mogu iskoristiti prednosti ovih tehnologija kako bi ostvarile konkuretske prednosti u dinamičkom poslovnom okruženju.

7. Literatura

1 Oracle, What is big data
2.Wikipedia, Big Data
3. Amazon, What is Hadoop
4. Cloud Google, What is Big data
5. Infoworld, What is Apache Spark, the Big Data platform that crushed Hadoop
6. Business tech Weekly, Big Data Infrastructure
7. Phonixnap, Apache Hadoop architecture explained
8. Aws Amazon, What is Spark?