# Úvod

Distribuované spracovanie údajov a veľké objemov údajov sú dve témy, ktoré sú v posledných rokoch čoraz aktuálnejšie vzhľadom na rastúci objem údajov, ktoré sa generujú každý deň. S technologickým pokrokom a digitalizáciou rôznych priemyselných odvetví exponenciálne narástol objem údajov, ktoré produkujú jednotlivci, organizácie a vlády. To viedlo k potrebe účinných a efektívnych spôsobov spracovania a analýzy týchto údajov, pričom do hry vstupujú technológie distribuovaného spracovania údajov a technológie na spracovanie veľkých objemov údajov, ako je napríklad Hadoop. Distribuované spracovanie údajov sa vzťahuje na proces rozdeľovania veľkých súborov údajov na menšie, lepšie zvládnuteľné časti a ich spracovanie vo viacerých výpočtových uzloch súčasne. To umožňuje rýchlejšie spracovanie a lepšiu škálovateľnosť, pretože na spracovanie rastúcich objemov údajov možno podľa potreby pridať ďalšie výpočtové uzly. Distribuované spracovanie údajov sa stáva čoraz dôležitejším, pretože veľkosť súborov údajov presahuje možnosti tradičných centralizovaných systémov spracovania. Na druhej strane, veľké údaje sa vzťahujú na veľké a komplexné súbory údajov, ktoré je príliš ťažké spracovať pomocou tradičných techník spracovania údajov. Tieto údaje sú často neštruktúrované alebo polo štruktúrované, čo sťažuje ich analýzu pomocou štandardných nástrojov. Technológie na spracovanie veľkých dát, ako je Hadoop, sú navrhnuté na spracovanie týchto veľkých súborov dát ich rozdelením na menšie časti a ich spracovaním v distribuovanej sieti počítačov. Hadoop je open-source rámec, ktorý poskytuje distribuovaný súborový systém a súbor nástrojov na spracovanie a analýzu veľkých dát. Vývoj od klastrových výpočtov ku cloudovým výpočtom a Hadoopu je spôsobený potrebou efektívne a nákladovo efektívne spracúvať obrovské množstvá údajov. V tejto práci si dané technológie lepšie predstavíme popíšeme si aký bol vývoj ich používania až do dnešných dní. (Aggarwal R., 2021)

# 1.0 História hadoopu

2002: Google uverejňuje dva dokumenty o súborovom systéme Google a MapReduce, ktoré opisujú nový spôsob spracovania veľkých súborov údajov v prostredí distribuovaných počítačov. Tieto dokumenty slúžili ako inšpirácia pre Hadoop.

2003: Doug Cutting a Mike Cafarella začali vyvíjať webový vyhľadávač s otvoreným zdrojovým kódom s názvom Nutch, ktorý využíval súborový systém Google a MapReduce na spracovanie veľkých súborov údajov. Rozhodli sa oddeliť distribuované výpočtové komponenty Nutch a vytvoriť samostatný rámec, ktorý nazvali Hadoop podľa hračkárskeho slona, ktorý patril Dougovmu synovi.

2006: Hadoop sa stal projektom Apache Software Foundation a bola vydaná verzia 0.1. Hadoop obsahoval distribuovaný súborový systém Hadoop (HDFS) a programovací model MapReduce, ktorý umožnil vývojárom písať programy, ktoré mohli spracovávať údaje vo veľkom počte uzlov v klastri.

2008: Hadoop 0.18, ktorý obsahoval niekoľko dôležitých vylepšení, ako napríklad Hadoop Streaming, ktorý umožnil vývojárom používať s MapReduce aj iné jazyky ako Java, a Hadoop Fair Scheduler, ktorý umožnil lepšie zdieľanie zdrojov klastra.

2009: Spoločnosť Yahoo! sa stala hlavným prispievateľom do projektu Hadoop a začala ho vo veľkej miere používať pre svoje vlastné potreby spracovania údajov. Popularita Hadoopu rýchlo rástla a stal sa jedným z najpoužívanejších rámcov na spracovanie veľkých dát.

2011: Hadoop 1.0, ktorý obsahoval niekoľko dôležitých vylepšení, ako napríklad podporu vysokej dostupnosti (HA) v HDFS, vďaka ktorej sa stal spoľahlivejším, a pridanie HBase, NoSQL databázy postavenej nad Hadoopom.

2012: Hadoop 2.0, ktorý obsahoval niekoľko dôležitých vylepšení, ako napríklad podporu pre YARN (Yet Another Resource Negotiator), ktorá umožnila Hadoopu spustiť viacero spracovateľských strojov na jednom klastri, a podporu pre Hadoop NameNode Federation, ktorá zlepšila škálovateľnosť a spoľahlivosť.

V súčasnosti je Hadoop naďalej populárnym rámcom na spracovanie veľkých objemov údajov a vytvoril veľký ekosystém nástrojov a technológií, ktoré na ňom stavajú. Tento ekosystém zahŕňa nástroje na integráciu, spracovanie, analýzu, vizualizáciu a ďalšie nástroje. (geeksforgeeks, 2019)

# 2.0 Predstavenie Hadoopu

Hadoop je open-source framework určený na ukladanie a spracovanie veľkého množstva údajov v prostredí distribuovaných výpočtových systémov. Rámec Hadoop sa skladá z niekoľkých rôznych komponentov, z ktorých každý vykonáva špecifickú úlohu v celkovom pracovnom procese spracovania údajov. Hlavné komponenty Hadoopu sú:

**Distribuovaný súborový systém Hadoop (HDFS)** je distribuovaný súborový systém, ktorý ukladá údaje vo viacerých uzloch v klastri. HDFS je navrhnutý na ukladanie veľkých súborov údajov na viacerých počítačoch, pričom každý súbor je rozdelený do blokov a uložený na rôznych uzloch v klastri. To poskytuje odolnosť voči chybám, pretože údaje možno replikovať vo viacerých uzloch, čím sa zabezpečí, že strata jedného uzla nebude mať za následok stratu údajov.

Programovací model **MapReduce** je ďalšou kľúčovou súčasťou rámca Hadoop. Ide o paradigmu distribuovaných výpočtov, ktorá umožňuje paralelné spracovanie veľkých súborov údajov v rámci klastra strojov. MapReduce rozdeľuje úlohu na menšie čiastkové úlohy a rozdeľuje ich medzi viaceré uzly v klastri. Keď každý uzol dokončí svoju úlohu, výsledky sa skombinujú a vytvoria konečný výstup.

**YARN** - Apache Hadoop YARN je kľúčovou súčasťou ekosystému Hadoop, ktorá poskytuje možnosti správy zdrojov a plánovania úloh pre aplikácie distribuovaného spracovania bežiace na klastri Hadoop. Oddeľuje funkcie správy zdrojov od spracovateľských rámcov, čím uľahčuje integráciu iných spracovateľských strojov s Hadoopom.

**Hadoop Common**: Hadoop Common je súbor nástrojov a knižníc, ktoré sú spoločné pre ostatné komponenty Hadoop. Poskytuje spoločné funkcie potrebné pre ostatné komponenty, ako sú konfiguračné súbory, protokolovanie a autentifikácia.

**Nástroje Hadoop**: Hadoop obsahuje niekoľko nástrojov príkazového riadka na interakciu s klastrom Hadoop vrátane nástrojov na operácie so súborovým systémom, správu klastra a monitorovanie úloh.

(Kalia K., 2020)

# 3.0 HDFS

Hadoop Distributed File System ( HDFS ) je distribuovaný súborový systém navrhnutý tak, aby fungoval na komoditnom hardvéri. Má veľa podobností s existujúcimi distribuovanými súborovými systémami. Rozdiely od iných distribuovaných súborových systémov sú však značné. HDFS je vysoko odolný voči chybám a je navrhnutý na nasadenie na lacnom hardvéri. HDFS poskytuje vysoko priepustný prístup k údajom aplikácií a je vhodný pre aplikácie, ktoré majú veľké súbory údajov. HDFS zmierňuje niekoľko požiadaviek POSIX, aby umožnil streamingový prístup k údajom systému súborov. HDFS bol pôvodne vybudovaný ako infraštruktúra pre projekt webového vyhľadávača Apache Nutch. HDFS je teraz podprojektom Apache Hadoop. Adresa URL projektu je <https://hadoop.apache.org/hdfs/> . (Hadoop, 2022)

**Predpoklady a ciele HDFS**

**Zlyhanie hardvéru**

Zlyhanie hardvéru je skôr normou ako výnimkou. Inštancia HDFS môže pozostávať zo stoviek alebo tisícok serverových počítačov, z ktorých každý ukladá časť údajov systému súborov. Skutočnosť, že existuje obrovské množstvo komponentov a že každý komponent má netriviálnu pravdepodobnosť zlyhania, znamená, že niektorý komponent HDFS je vždy nefunkčný. Preto je detekcia porúch a rýchle, automatické obnovenie z nich hlavným architektonickým cieľom HDFS. (Hadoop, 2022)

**Streaming Data Access**

Aplikácie, ktoré bežia na HDFS, potrebujú streamingový prístup k svojim dátovým súborom. Nie sú to aplikácie na všeobecné použitie, ktoré zvyčajne bežia na súborových systémoch na všeobecné použitie. HDFS je navrhnutý skôr na dávkové spracovanie než na interaktívne použitie používateľmi. Dôraz sa kladie skôr na vysokú priepustnosť prístupu k údajom než na nízku latenciu prístupu k údajom. POSIX ukladá veľa tvrdých požiadaviek, ktoré nie sú potrebné pre aplikácie, ktoré sú zamerané na HDFS. Sémantika POSIX v niekoľkých kľúčových oblastiach bola obchodovaná s cieľom zvýšiť rýchlosť prenosu údajov. (Hadoop, 2022)

**Veľké množiny údajov**

Aplikácie, ktoré bežia na HDFS, majú veľké množiny údajov. Typický súbor v HDFS má veľkosť gigabajtov až terabajtov. HDFS je teda vyladený na podporu veľkých súborov. Mal by poskytovať vysokú agregovanú dátovú šírku pásma a škálovať na stovky uzlov v jednom klastri. Mal by podporovať desiatky miliónov súborov v jednej inštancii. (Hadoop, 2022)

**Jednoduchý koherentný model**

Aplikácie HDFS potrebujú pre súbory prístupový model typu zápis-raz-čítanie-viac. Po vytvorení, zapísaní a zatvorení súboru nie je potrebné meniť. Tento predpoklad zjednodušuje problémy s koherenciou údajov a umožňuje prístup k údajom s vysokou priepustnosťou. K tomuto modelu sa perfektne hodí aplikácia MapReduce alebo webový prehľadávač. V budúcnosti sa plánuje podpora pripájania-zápisov do súborov. (Hadoop, 2022)

**„Výpočet s presunom je lacnejší ako presun údajov“**

Výpočet požadovaný aplikáciou je oveľa efektívnejší, ak sa vykonáva v blízkosti údajov, s ktorými pracuje. To platí najmä vtedy, keď je veľkosť súboru údajov obrovská. To minimalizuje preťaženie siete a zvyšuje celkovú priepustnosť systému. Predpokladom je, že často je lepšie migrovať výpočet bližšie k miestu, kde sa nachádzajú údaje, než presúvať údaje tam, kde je spustená aplikácia. HDFS poskytuje aplikáciám rozhrania, aby sa mohli priblížiť k miestu, kde sa nachádzajú údaje. (Hadoop, 2022)

**Prenosnosť naprieč heterogénnymi hardvérovými a softvérovými platformami**

HDFS bol navrhnutý tak, aby bol ľahko prenosný z jednej platformy na druhú. To uľahčuje široké prijatie HDFS ako platformy voľby pre veľkú skupinu aplikácií. (Hadoop, 2022)

HDFS má architektúru master/slave.

**NameNode**: funguje ako Master v klastri Hadoop, ktorý riadi Datanode (podriadené uzly). Nameode sa používa najmä na ukladanie Metadát, t. j. údajov o údajoch. Metaúdajmi môžu byť protokoly transakcií, ktoré sledujú činnosť používateľa v klastri Hadoop. Klaster HDFS pozostáva z jedného NameNode, hlavného servera, ktorý spravuje menný priestor súborového systému a reguluje prístup klientov k súborom. (geeksforgeeks, 2023)

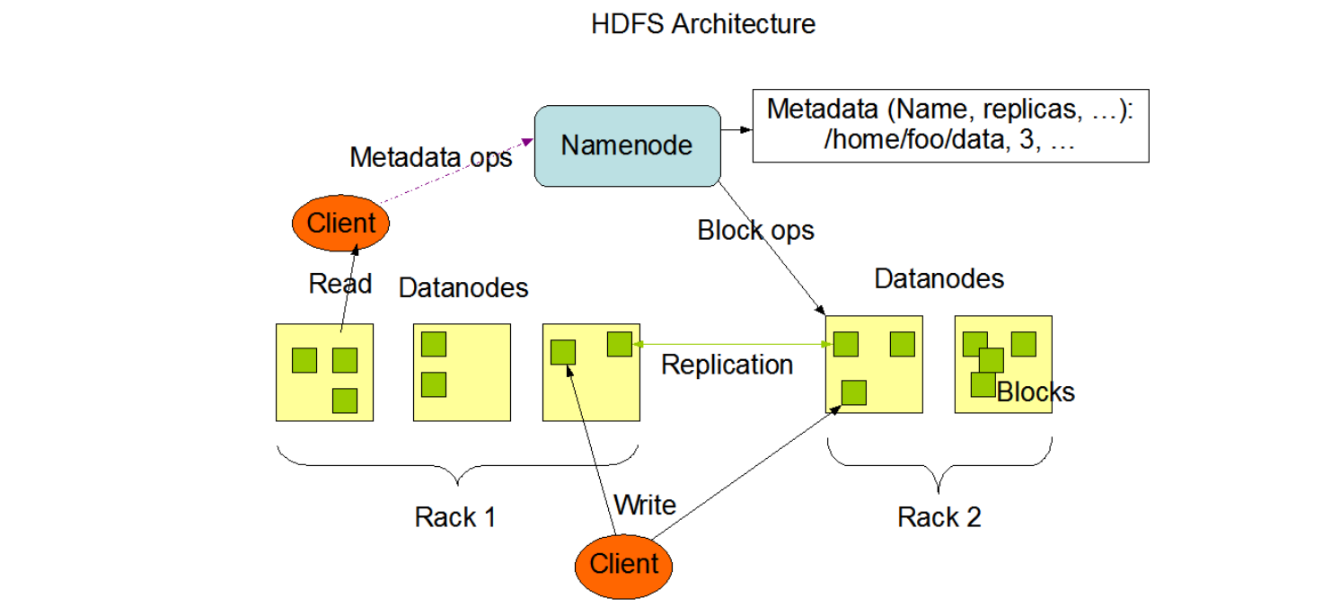
Metaúdajmi môžu byť aj názov súboru, veľkosť a informácie o umiestnení (číslo bloku, id bloku) dátového uzla, ktoré Namenode ukladá na vyhľadanie najbližšieho dátového uzla na rýchlejšiu komunikáciu. Namenode dáva dátovým uzlom pokyny na operácie, ako je vymazanie, vytvorenie, replikácia atď. (geeksforgeeks, 2023)

**DataNode**: DataNode funguje ako Slave. Komunikujú s uzlom NameNode a vykonávajú operácie, ako je replikácia blokov, riešenie zlyhaní a hlásenie stavu údajov, ktoré ukladajú. Ich počet môže byť od 1 do 500 alebo aj viac. Zvyčajne jeden na každý uzol v klastri, ktoré spravujú úložisko pripojené k uzlom, na ktorých bežia Čím väčší je počet DataNode, tým viac údajov dokáže klaster Hadoop uložiť. Preto sa odporúča, aby mal DataNode vysokú kapacitu na ukladanie veľkého počtu blokov súborov. (geeksforgeeks, 2023)

## **Názvový priestor súborového systému**

HDFS podporuje tradičnú hierarchickú organizáciu súborov. Používateľ alebo aplikácia môže vytvárať adresáre a ukladať súbory do týchto adresárov. Hierarchia názvového priestoru súborového systému je podobná väčšine existujúcich súborových systémov; je možné vytvárať a odstraňovať súbory, presúvať súbor z jedného adresára do druhého alebo premenovať súbor. (Hadoop, 2022)

## **Replikácia údajov**

HDFS je navrhnutý tak, aby spoľahlivo ukladal veľmi veľké súbory naprieč počítačmi vo veľkom klastri. Ukladá každý súbor ako sekvenciu blokov; všetky bloky v súbore okrem posledného bloku majú rovnakú veľkosť. Bloky súboru sú replikované kvôli odolnosti voči chybám. Veľkosť bloku a faktor replikácie sú konfigurovateľné pre každý súbor. Aplikácia môže určiť počet replík súboru. Faktor replikácie je možné zadať pri vytváraní súboru a neskôr ho možno zmeniť. Súbory v HDFS sú zapisovateľné len raz a majú vždy jeden zapisovač. (Hadoop, 2022)

Obrázok 1-HDFS Architektúra Zdroj: (Hadoop, 2022)

# 3.0 MapReduce

MapReduce je programovací model používaný na efektívne paralelné spracovanie veľkých súborov údajov distribuovaným spôsobom. Údaje sa najprv rozdelia a potom sa skombinujú, aby sa vytvoril konečný výsledok. Knižnice pre MapReduce sú napísané v mnohých programovacích jazykoch s rôznymi optimalizáciami. Účelom MapReduce v Hadoop je zmapovať každú z úloh a potom ju zredukovať na ekvivalentné úlohy na zabezpečenie menšej réžie v sieti klastra a na zníženie výpočtového výkonu. Úloha MapReduce je rozdelená najmä na dve fázy Map Phase (fáza mapovania) a Reduce Phase (fáza redukcie).

Fáza Map a fáza Reduce sú dve základné zložky programovacieho modelu MapReduce. Vo fáze Map sú vstupné údaje rozdelené na menšie časti a spracované funkciou Map, zatiaľ čo vo fáze Reduce sú priebežné výsledky vytvorené funkciou Map agregované pomocou funkcie Reduce. Pozrime sa bližšie na jednotlivé fázy: (Taylor, 2023)

**Fáza Map:**

Vo fáze Map sa vstupné údaje rozdelia na menšie časti a každá časť sa spracuje nezávisle pomocou funkcie Map. Funkcia Map vezme každý vstupný záznam a použije naň transformáciu, pričom ako výstup vytvorí súbor dvojíc kľúč-hodnota. Funkcia Map môže vykonávať rôzne typy operácií, napríklad filtrovanie, triedenie, počítanie alebo agregáciu údajov.

Výstupom funkcie Map je súbor dvojíc kľúč-hodnota, kde kľúč predstavuje jedinečný identifikátor záznamu a hodnota predstavuje transformované údaje. Kľúče vytvorené funkciou Map sa používajú na rozdelenie údajov do rôznych skupín, ktoré sa potom posielajú do fázy Reduce na ďalšie spracovanie.

**Fáza redukcie:** (Taylor, 2023)

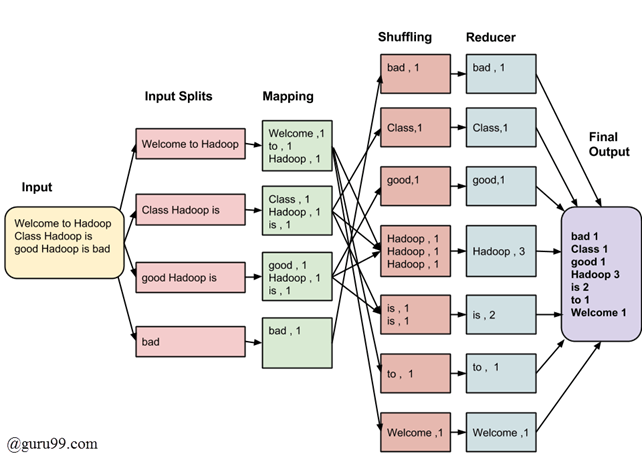
Vo fáze Reduce sa páry kľúč-hodnota vygenerované funkciou Map zoskupia podľa kľúčov a spracujú sa funkciou Reduce. Funkcia Reduce vezme všetky hodnoty priradené ku konkrétnemu kľúču a použije na ne transformáciu, čím sa vytvorí súbor výstupných dvojíc kľúč-hodnota.

Funkcia Reduce môže vykonávať rôzne typy operácií, napríklad sčítanie, spriemerovanie alebo počítanie údajov. Výstupom funkcie Reduce je súbor dvojíc kľúč-hodnota, ktoré predstavujú konečný výstup úlohy MapReduce. Výstup sa zvyčajne zapisuje do súboru alebo databázy a môže sa použiť ako vstup pre ďalšie úlohy MapReduce alebo iné aplikácie. (Taylor, 2023)

**Premiešanie a triedenie:**

Medzi fázou Map a fázou Reduce sa nachádza fáza Shuffle and Sort, ktorá je zodpovedná za triedenie výstupu funkcie Map a jeho zoskupenie podľa kľúčov pred odoslaním do fázy Reduce. Fáza Shuffle and Sort zabezpečuje, že všetky dvojice kľúč-hodnota spojené s konkrétnym kľúčom sú spracované rovnakou funkciou Reduce bez ohľadu na to, v ktorom uzle boli vytvorené. (Taylor, 2023)

Stručne povedané, programovací model MapReduce poskytuje škálovateľný a voči chybám odolný rámec na spracovanie veľkých súborov údajov distribuovaným a paralelným spôsobom. Fáza Map aplikuje transformáciu na vstupné údaje a generuje súbor dvojíc kľúč-hodnota, zatiaľ čo fáza Reduce agreguje priebežné výsledky vygenerované fázou Map a vytvára konečný súbor výstupných dvojíc kľúč-hodnota. Fáza Shuffle and Sort zabezpečuje zoskupenie údajov a ich spracovanie príslušnou funkciou Reduce. (Taylor, 2023)



Obrázok 2-architektúra mapreduce zdroj : (Taylor, 2023)

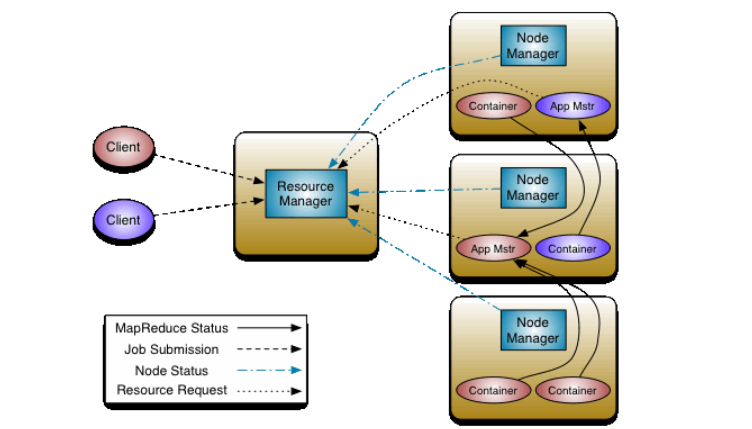
# 4.0 YARN

bol predstavený vo verzii Hadoop 2.0 a oddelil možnosti správy zdrojov a plánovania od rámca spracovania MapReduce, ktorý bol vo verzii Hadoop 1.0 úzko prepojený s distribuovaným súborovým systémom Hadoop (HDFS). Toto oddelenie umožnilo, aby sa do systému Hadoop integrovali ďalšie rámce pre distribuované spracovanie, ako napríklad Apache Spark a Apache Flink, a aby na vykonávanie svojich pracovných úloh využívali rovnaký súbor zdrojov. (Apache Hadoop, 2023)

**YARN** je navrhnutý tak, aby podporoval rôzne typy aplikácií distribuovaného spracovania okrem, ako napríklad streamovanie dát v reálnom čase, spracovanie grafov a dávkové spracovanie. Poskytuje centrálneho správcu zdrojov, ktorý riadi prideľovanie zdrojov, ako je procesor a pamäť, rôznym aplikáciám bežiacim na klastri. YARN obsahuje aj správcu uzlov, ktorý beží na každom uzle v klastri a spravuje zdroje pridelené aplikáciám bežiacim na danom uzle. (Apache Hadoop, 2023)

Yarn sa skladá z dvoch hlavných komponentov: ResourceManager a NodeManager. ResourceManager je zodpovedný za správu prideľovania zdrojov v rámci klastra vrátane CPU, pamäte a šírky pásma siete. Prijíma požiadavky na zdroje od klientskych aplikácií, prideľuje im zdroje a monitoruje ich využitie. Správca uzlov beží na každom jednotlivom uzle v klastri a je zodpovedný za správu zdrojov na danom uzle vrátane monitorovania aplikácií, ktoré na ňom bežia, a podávania správ správcovi zdrojov.

Yarn poskytuje aj podporu pre rôzne typy aplikácií vrátane MapReduce, Spark a HBase. To umožňuje, aby na tom istom klastri bežali rôzne aplikácie, pričom Yarn poskytuje potrebné zdroje a spravuje ich využívanie. (Apache Hadoop, 2023)



Obrázok 3 architektúra yarn zdroj: (Apache Hadoop, 2023)

# 5.0 Apache hive

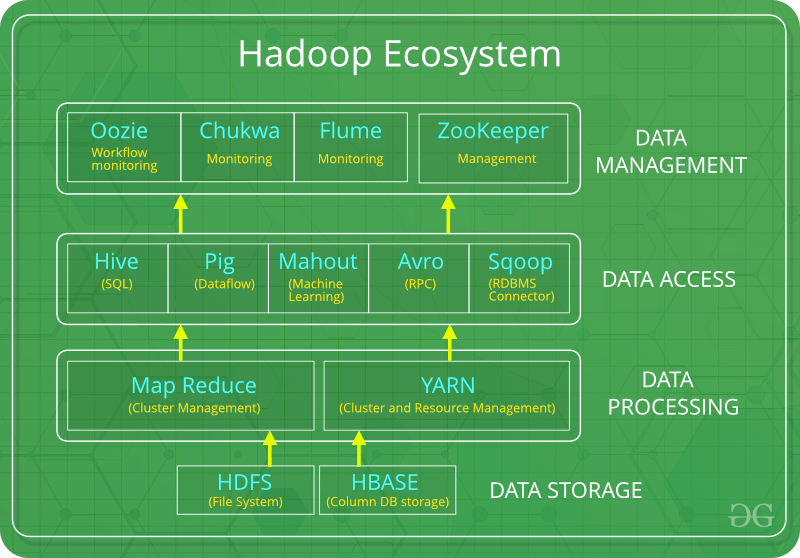
Apache Hive bol vyvinutý spoločnosťou Facebook pre vývojárov so skúsenosťami s SQL. Pomocou tohto komponentu môžu vývojári SQL písať príkazy Hive Query Language ako štandardné príkazy SQL .Apache Hive je v podstate otvorený systém dátového skladu založený na Hadoop, ktorý uľahčuje jednoduché ad-hoc dotazy a sumarizáciu údajov. Umožňuje tiež rýchlu analýzu veľkých súborov údajov uložených v rôznych súborových systémoch a databázach integrovaných s Apache Hadoop. Hive poskytuje vývojárom SQL jednoduchý spôsob písania príkazov pomocou Hive Query Language (HQL), ktoré možno použiť na veľké množstvo neštruktúrovaných údajov. Cieľom Hive je uľahčiť programovanie MapReduce, pretože nemusíte písať zdĺhavý kód Java. Dopyty píšete jednoducho v HQL a ten automaticky prekladá dotazy podobné SQL do dávkových úloh MapReduce.Apache (Bizety Staff, 2020)

# 6.0 Apache Pig

Apache Pig je vysokoúrovňová platforma na vytváranie programov, ktoré bežia na Apache Hadoop. Jazyk pre túto platformu sa nazýva Pig Latin Ide o jednoduchý skriptovací jazyk podobný SQL je a jeho hlavným cieľom je vykonať požadované operácie a usporiadať konečný výstup v požadovanom formáte. Pig môže vykonávať svoje úlohy Hadoop v MapReduce, Apache Tez alebo Apache Spark. (Bizety Staff, 2020)

# 7.0 Apache Spark

Apache Spark, pôvodne vyvinutý v UC Berkeley, je ultrarýchly jednotný analytický nástroj pre strojové učenie a veľké dáta. Internetoví giganti ako Yahoo, Netflix a eBay nasadili Spark vo veľkom meradle na spracovanie petabajtov údajov v klastroch s viac ako 8 000 uzlami.Spark sa primárne používa na spracovanie údajov o dávkach v pamäti. Podporuje tiež spracovanie tokov kombinovaním tokov údajov do menších dávok a ich spustením. (Apache Hadoop, 2023)



Obrázok 4 hadoop ecosystem zdroj (Bizety Staff, 2020)

Záver

Súhrnne povedané, Apache Hadoop YARN je kľúčovou súčasťou ekosystému Hadoop, ktorá poskytuje možnosti správy zdrojov a plánovania úloh pre aplikácie distribuovaného spracovania bežiace na klastri Hadoop. Oddeľuje funkcie správy zdrojov od spracovateľských rámcov, čím uľahčuje integráciu iných spracovateľských strojov s Hadoopom.

Jednou z hlavných výhod používania distribuovaného spracovania údajov a technológií na spracovanie veľkých objemov údajov, ako je Hadoop, je ich schopnosť spracovať obrovské objemy údajov škálovateľným a nákladovo efektívnym spôsobom. To viedlo k ich väčšiemu rozšíreniu v rôznych odvetviach vrátane finančníctva, zdravotníctva, maloobchodu a výroby. Napríklad v oblasti financií sa technológie na distribuované spracovanie údajov a veľké objemy údajov môžu používať na analýzu veľkých objemov finančných údajov v reálnom čase, čo poskytuje prehľad o trendoch na trhu a pomáha identifikovať potenciálne investičné príležitosti.

S distribuovaným spracovaním údajov a veľkými dátami sú však spojené aj výzvy. Jednou z hlavných výziev je zložitosť technológie a zručnosti potrebné na prácu s ňou. Na plné využitie výhod týchto technológií potrebujú organizácie kvalifikovaných dátových vedcov a inžinierov, ktorí dokážu vyvinúť a udržiavať potrebnú infraštruktúru a procesy v nej.

Na záver možno konštatovať, že distribuované spracovanie údajov a technológie na spracovanie veľkých objemov údajov, ako je Hadoop, sú v dnešnom svete založenom na údajoch čoraz dôležitejšie. Tým, že tieto technológie umožňujú spracovanie a analýzu veľkých a komplexných súborov údajov, umožňujú organizáciám získať cenné poznatky o ich činnosti a prijímať informovanejšie rozhodnutia. Zložitosť technológie a zručnosti potrebné na prácu s ňou však znamenajú, že organizácie musia investovať do budovania potrebných kapacít, aby mohli plne využívať výhody týchto technológií.

# Bibliografia

Aggarwal R., V. J. (2021). Small files’ problem in Hadoop: A systematic literature review. *Journal of King Saud University –Computer and Information Sciences*.

Apache Hadoop. (29. 7 2023). *Apache Hadoop YARN*. Dostupné na Internete: https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html

Bizety Staff. (20. 6 2020). *Hadoop Ecosystem: MapReduce, YARN, Hive, Pig, Spark, Oozie, Zookeeper, Mahout, and Kube2Hadoop*. Dostupné na Internete: https://www.bizety.com/2020/06/20/hadoop-ecosystem-mapreduce-yarn-hive-pig-spark-oozie-zookeeper-mahout-and-kube2hadoop/

geeksforgeeks. (18. 1 2019). *Hadoop | History or Evolution.* Dostupné na Internete: https://www.geeksforgeeks.org/hadoop-history-or-evolution/

geeksforgeeks. (3. 1 2023). *Hadoop – Architecture*. Dostupné na Internete: https://www.geeksforgeeks.org/hadoop-architecture/

Hadoop, A. (18. 5 2022). *HDFS Architecture Guide*. Dostupné na Internete: https://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html

Kalia K., G. N. (2020). Analysis of hadoop MapReduce scheduling in heterogeneous. *Ain Shams Engineering Journal*.

Taylor. (6. 1 2023). *What is MapReduce in Hadoop? Big Data Architecture*. Dostupné na Internete: https://www.guru99.com/introduction-to-mapreduce.html