ISSN 1512-1232

# დიდი მონაცემების ტექნოლოგიების მიმოხილვა

სოფიო ქათამაძე
უფროსი პროგრამისტი, შპს ეპამ სისტემს (ჯორჯია). დოქტორანტი, ინფორმატიკისა და მართვის სისტემების ფაკულტეტი, საქართველოს ტექნიკური უნივერსიტეტი, თბილისი, კოსტავას 77
skatamadze88@gmail.com

***ანოტაცია:*** *განხილულია Big Data-ს წამყვანი პროგრამული უზრუნველყოფანი, რომელთაც მოწინავე ადგილი უკავიათ 21-ე საუკუნის მიღწევებში. წარმოდგენილია დიდი მონაცემების ტექნოლოგიების აუცილებლობა, როცა ტრადიციული მეთოდებით Big Data-სთან გამკლავება შეუძლებელი ხდება.*
***საკვანძო სიტყვები:*** *დიდი მონაცემები, გამოთვლითი კვანძი, რელაციური მონაცემთა ბაზა, პროგრამირების ინტერფეისი, მანქანური სწავლება*

## 1. შესავალი

XXI-ე საუკუნეში Big Data–მ გაიარა მთელი რიგი ევოლუციური ნაბიჯები. ინფორმაციის ზრდასთან ერთად, Big Data გაფართოვდა არა მხოლოდ მისი მასშტაბით, არამედ ტექნოლოგიითაც. მისი ხუთი ძირითადი მახასიათებლის, მოცულობის, არაერთგვაროვნების, სიჩქარის, სანდოობისა და მნიშვნელობის თვალსაზრისით, თანამედროვე ტექნიკა, ტექნოლოგიები და აღჭურვილობა არის საჭირო. ამრიგად, მონაცემთა მოპოვების, დამუშავების, ანალიზისა და შენახვისთვის აუცილებელია მოწინავე აპარატურა და პროგრამული უზრუნველყოფა.

ამჟამად, Big Data- ს ინფრასტრუქტურა მოიცავს სერვერებს, შენახვის სისტემებს, ღრუბლოვან სერვისებს და ქსელურ აღჭურვილობას. დიდი მონაცემების აპლიკაციები კი მოიცავს პარალელურ და განაწილებულ ფაილურ სისტემებს, აღდგენით და მონაცემთა დამუშავების პროგრამულ უზრუნველყოფას.

## 2. დიდი მონაცემების მოწინავე ტექნოლოგიები

დიდი მონაცემებისთვის ხელმისაწვდომია ზღვა ტექნოლოგიები, რომლებიც დღითიდღე ვითარდება და ტრანსფორმაციას განიცდის. განვიხილოთ Big Data-ს წამყვანი პროგრამული უზრუნველყოფანი. ისინი ძირითადად მსხვილი კომპანიების მიერ დაწყებული პროექტებია, როცა ტრადიციული მეთოდებით დიდი მონაცემების დანიშნულებისამებრ წარმართვა ვერ ხერხდებოდა

Apache Hadoop არის ღია პროგრამული უზრუნველყოფების სისტემა, მრავალი სერვერიდან შემდგარ კლასტერებზე, დიდი მონაცემების დამუშავებისა და შენახვისთვის. იმის ნაცვლად, რომ დაეყრდნოს ინფრასტრუქტურას მაღალი ხელმისაწვდომობის მისაღებად, Hadoop ბიბლიოთეკა პრობლემებს პროგრამულ ფენაშივე უმკლავდება. Hadoop-ის გაშვება შეიძლება როგორც ერთ სერვერზე, ასევე ათასობით სერვერიდან შემდგარ კლასტერზეც. მას აქვს თანდათან მასშტაბების უსაზღვროდ გაზრდის მხარდაჭერა. Hadoop მონაცემების გამოთვლასა და შენახვას ლოკალურად ანხორციელებს.

მომხმარებლებს შეუძლიათ დაამატონ მოდულები Hadoop-ზე, მათი საჭიროებებისა და მოთხოვნების შესაბამისად (მაგალითად, მოცულობა, შესრულება, საიმედოობა, მასშტაბირებადობა, უსაფრთხოება). ჩვენ განვიხილავთ ძირითად მოდულებს, რომლებიც დამუშავების სხვადასხვა ფენაში გამოიყენება.

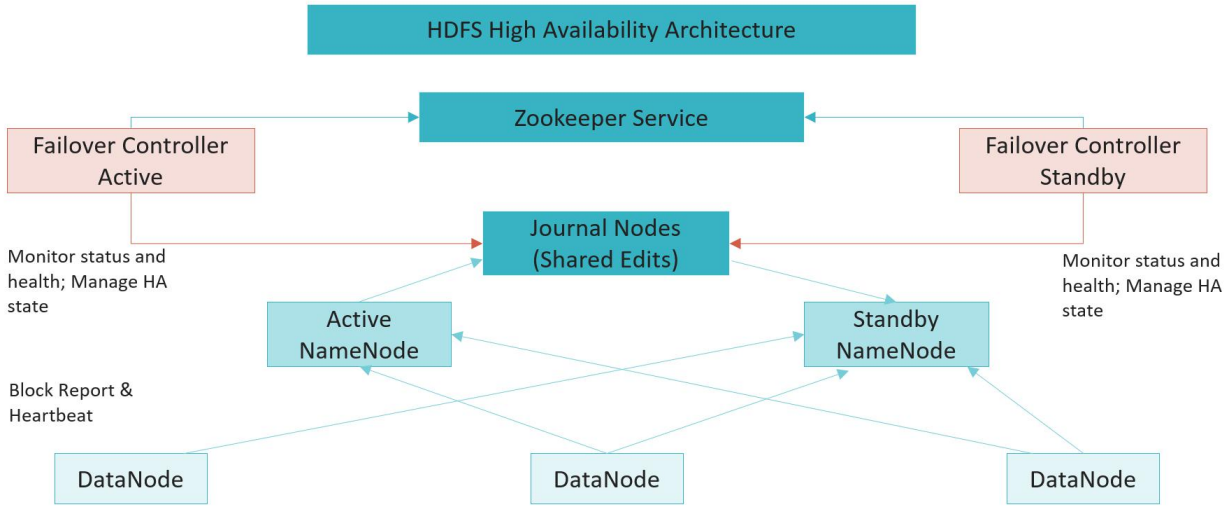
ISSN 1512-1232

### a. მონაცემთა შენახვის ფენა: HDFS, Hbase, Cassandra

Apache Hadoop-ის ბირთვს წარმოადგენს დეცენტრალიზებული, მასშტაბირებადი და პორტატული ფაილური სისტემა - HDFS (Hadoop Distributed File System). HDFS შეიქმნა მაღალი შეყოვნების ოპერაციების პარალელური დამუშავებისთვის. HDFS-ის მთავარი უპირატესობაა მისი პორტაბელურობა ჰეტეროგენულ ტექნიკურ და პროგრამულ პლატფორმებზე. HDFS ინახავს ძალიან დიდ ფაილებს, რომელთა განთავსება შესაძლებელია ეკონომიურ ინფრასტრუქტურაზე. მისი ეფექტურად გამოყენებისთვის სასურველია დიდი ფაილების ნაკლები რაოდენობის შენახვა, ვიდრე მრავლობითი მცირე ფაილების შენახვა. Hadoop ყოფს ფაილებს დიდ ბლოკებად და ანაწილებს მათ კლასტერის შიგნით. შემდეგ იგი გადასცემს კლასტერის კომპონენტებს ამოცანებს მონაცემების პარალელური დამუშავებისთვის. ბლოკი არის HDFS-ის ყველაზე მცირე ერთეული. ის ინახავს მონაცემთა ბლოკებს სხვადასხვა მონაცემთა კვანძებზე (Datanodes).

HDFS-ს აქვს ჰორიზონტალურად მასშტაბების გაზრდის მხარდაჭერა. რესურსების გაზრდა თავისუფლად არის შესაძლებელი Datanode-ების დამატებით კლასტერში. აქედან გამომდინარე, მკვეთრად უმჯობესდება შესრულების ხარისხი.

HDFS-ის საშუალებით ასევე გადაჭრილია მრავალფეროვანი მონაცემების შენახვის პრობლემა. HDFS–ს შეუძლია შეინახოს ყველანაირი მონაცემები (სტრუქტურირებული, ნახევრად სტრუქტურირებული ან არასტრუქტურირებული).




ნახ. 1. HDFS High Availability Architecture

მონაცემთა დამუშავების სიჩქარე - ეს არის დიდი მონაცემების მთავარი პრობლემა. ამ პრობლემის გადასაჭრელად, გამოსავალია მონაცემების გადატანის ნაცვლად გამოთვლების წარმოება Datanode-ზე. ტრადიციული ტექნოლოგიებისგან განსხვავებით, Hadoop არ აკოპირებს მეხსიერებაში მთელ მონაცემებს გამოთვლების შესასრულებლად. ამის ნაცვლად, Hadoop ასრულებს დავალებებს იმ კვანძებზე, სადაც მონაცემები ინახება. ამრიგად, Hadoop ათავისუფლებს ქსელს და სერვერებს მნიშვნელოვანი საკომუნიკაციო დატვირთვისგან. მაგალითად, Hadoop–ზე ტერაბაიტის მონაცემების დამუშავებისთვის საჭიროა მხოლოდ რამდენიმე წამი, როცა კლასიკურ SIEM (Security Information and Event Management) – ზე 20 წთ ან მეტი დრო.

ISSN 1512-1232

Hadoop-ის ერთ-ერთი მთავარი მახასიათებელია მისი მოქნილობა. არ არის აუცილებელი მონაცემთა წინასწარი დამუშავება. შესაძლებელია სასურველი რაოდენობის მონაცემების შენახვა და შემდგომ წაკითხვისას მისი გამოყენება.

Hadoop რომელიმე კვანძის დაზიანების შემთხვევაშიც შეუფერხებლად აგრძელებს ფუნქციონირებას. ის დაზიანებული კვანძის შესასრულებელ პროცესებს ავტომატურად სხვა კვანძებზე ამისამართებს.

HBase არის განაწილებული არა რელაციური მონაცემთა ბაზა. ეს არის ღია პროგრამული უზრუნველყოფის პროექტი, რომელიც აგებულია HDFS-ზე. იგი განკუთვნილია დაბალი შეყოვნების ოპერაციებისთვის. HBase ემყარება სვეტზე ორიენტირებულ key / value მონაცემების მოდელს. მას შეუძლია მაღალი სიჩქარით აწარმოოს ტაბლიცებზე განახლებები და ჰორიზონტალური მასშტაბირებადობით გაანაწილოს ბრძანებები კლასტერებში.

Hbase-ს ბევრი დადებითი მახასიათებელი აქვს, როგორიცაა რეალურ დროში მოთხოვნების დამუშავება, NL (Natural Language) ძიება, დიდ მონაცემთა წყაროებზე მუდმივი წვდომა, ჰორიზონტალური და მოდულური მასშტაბირებადობა, ცხრილების ავტომატური კონფიგურირება. ის გამოიყენება Big Data–ს მრავალ გადაწყვეტილებაში, მაგალითად როგორიცაა Facebook–ის შეტყობინებების პლატფორმა. HDFS–ის მსგავსად, HBase–ს აქვს MasterNode. ის მართავს კლასტერს და მასზე დაქვემდებარებულ კვანძებს, რომლებიც ინახავენ ცხრილების ნაწილებს და ასრულებენ მონაცემებზე ოპერაციებს.

Apache Cassandra განაწილებული ბაზაა ძალიან დიდი მოცულობის სტრუქტურიზებული მონაცემების სამართავად. Facebook-ში შექმნილი ღია პროგრამული უზრუნველყოფის სისტემა არის მასშტაბირებადი (scalable), ხარვეზების მიმართ ტოლერანტული (fault tolerant) და მდგრადი (consistent). იგი მკვეთრად განსხვავდება რელაციურ მონაცემთა ბაზების მართვის სისტემებისაგან. Cassandra-ს იყენებენ ისეთი მსხვილი კომპანიები, როგორიცაა Facebook, Twitter, Cisco, ebay, Netflix და ა. შ.

კასანდრას მთავარი გამოწვევა, არქიტექტურული თვალსაზრისით, არის გაუმკლავდეს დიდი მოცულობის მონაცემებით დატვირთვას მრავალი კვანძის (Node) მეშვეობით ისე, რომ არც ერთი შეფერხების წერტილი (Single Point of Failure) არ ჰქონდეს. კასანდრა თავის node-ებს ე. წ. Peer-to-peer განაწილებულ სისტემაში აერთიანებს და მონაცემებს კლასტერის ყველა node-ზე ანაწილებს. კასანდრას კლასტერში ყველა კვანძი ერთსა და იმავე როლს ასრულებს. თითოეული მათგანი დამოუკიდებელია, თუმცა, ამავე დროს, დაკავშირებულია სხვა node-ებთან. თითოეულ მათგანს შეუძლია წაკითხვის და ჩაწერის მოთხოვნების მიღება, მიუხედავად იმისა, თუ მონაცემები რეალურად კლასტერის რა ნაწილშია ლოკალიზებული. როცა რომელიმე კვანძი ფერხდება და ავარიულად ითიშება, ეს მოთხოვნები შესასრულებლად ქსელში სხვა node-ებს ეგზავნებათ. ამასთანავე, კლასტერში ერთი ან რამდენიმე კვანძი მონაცემების კონკრეტული ნაწილისათვის რეპლიკების ფუნქციას ასრულებს.

### b. მონაცემთა დამუშავების ფენა: MapReduce, Yarn

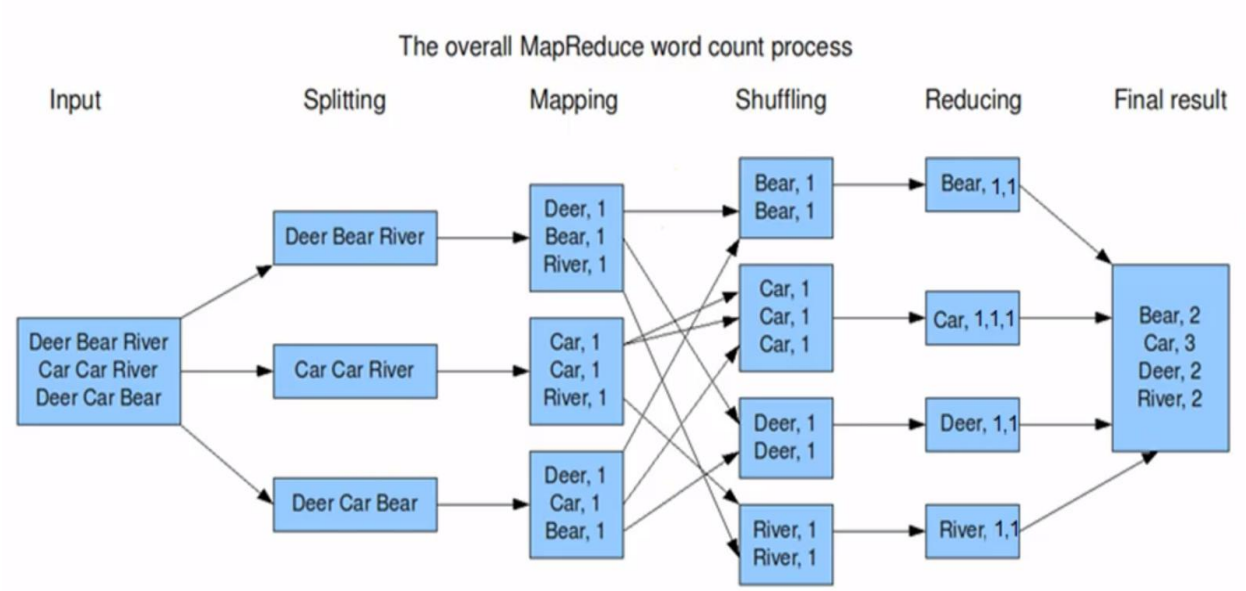
MapReduce და YARN წარმოადგენს Hadoop–ზე მონაცემთა დამუშავების ორ ვარიანტს. ისინი შექმნილია სამუშაოს დაგეგმვის, რესურსებისა და კლასტერის მართვისთვის. MapReduce არის პროგრამირების მოდელი, ხოლო YARN არის განაწილებული კლასტერის არქიტექტურა.

ISSN 1512-1232

MapReduce არის პროგრამების შემუშავების გარემო, რომელიც შედგება პროგრამირების მოდელისა და მისი იმპლემენტაციისგან. MapReduce–ს დიდი სარგებელი აქვს Big Data-ს პროგრამებისთვის. ის ამარტივებს მასიური მოცულობის მონაცემების დამუშავებას ეფექტური მექანიზმების საშუალებით.

MapReduce პროგრამირების მოდელი იყენებს შემდეგ ორ ფუნქციას, რომლებიც ანხორციელებენ მონაცემთა გამოთვლებს: Map ფუნქცია და Reduce ფუნქცია.

Map ფუნქცია ყოფს შეყვანილ მონაცემებს (მაგალითად, გრძელი ტექსტური ფაილი) მონაცემთა დამოუკიდებელ დანაყოფებად, რომლებიც წარმოადგენს key-value წყვილებს. შემდეგ, MapReduce აგზავნის ყველა key-value წყვილს Mapper-ში, რომლებიც ამუშავებენ თითოეულ მათგანს ინდივიდუალურად რამდენიმე პარალელური Map დავალების გამოყენებით. მონაცემთა თითოეული დანაყოფი ენიჭება გამოთვლის უნიკალურ კვანძს. Mapper-ს რეზულტატად გამოაქვს ერთი ან მეტი შუალედური key-value წყვილი. ამ ეტაპზე MapReduce-ს ევალება ყველა შუალედური key-value წყვილის შეგროვება, მათი დახარისხება და დაჯგუფება გასაღების (key) მიხედვით. ამრიგად, შედეგად მიიღება სორტირებული მრავალი გასაღები ასოცირებულ მნიშვნელობებთან ერთად.




ნახ. 2. MapReduce sample process

შემცირების (Reduce) ფუნქცია გამოიყენება შუალედური მონაცემების დასამუშავებლად. თითოეული უნიკალური გასაღებისთვის, შემცირების ფუნქცია აერთიანებს გასაღების მნიშვნელობებს წინასწარ განსაზღვრული პროგრამის შესაბამისად (მაგ. ფილტრაცია, შეჯამება, დახარისხება, ჰეშირება, საშუალო არითმეტიკულის ან მაქსიმუმის პოვნა). ამის შემდეგ, ის აწარმოებს ერთ ან რამდენიმე key-value წყვილს. დაბოლოს, MapReduce პროგრამების შემუშავების გარემო ფაილში ინახავს მიღებულ key-value ყველა წყვილს.

YARN MapReduce–სთან შედარებით უზრუნველყოფს უკეთეს მასშტაბირებადობას, გაძლიერებულ პარალელიზმს და რესურსების მენეჯმენტს. ის მოიცავს ოპერაციული სისტემის ფუნქციებს Big Data ანალიტიკური პროგრამებისთვის. Hadoop–ის არქიტექტურა შეიცვალა YARN რესურსების მენეჯერის ინტეგრაციისთვის. ზოგადად, YARN მუშაობს HDFS-ის ზედა ფენაზე, რაც მრავალი პროგრამის პარალელურად შესრულების საშუალებას იძლევა. მისი საშუალებით შესაძლებელია მონაცემების დამუშავება როგორც პაკეტებად, ასევე რეალურ დროში ინტერაქტიულად -

ISSN 1512-1232

ნაკადებების პროცესინგით. YARN თავსებადია MapReduce პროგრამირების ინტერფეისთან (API). YARN–ზე გასაშვებად - მომხმარებლებმა MapReduce ოპერაციების რეკომპილაცია უნდა მოახდინონ.

### c. მონაცემთა გამოკითვის ფენა: Apache Pig, JAQL, Apache Hive, Apache Spark SQL

Apache Pig არის ღია პროგრამული უზრუნველყოფის გარემო, რომელიც ქმნის მაღალი დონის სკრიპტულ ენას, სახელწოდებით Pig Latin. ის ამცირებს MapReduce-ის სირთულეებს, MapReduce სამუშაო პროცესების პარალელურ შესრულებით Hadoop-ზე. თავისი ინტერაქტიული გარემოს საშუალებით, Pig like Hive, ამარტივებს პარალელურად მასიური მონაცემების შესწავლასა და დამუშავებას HDFS-ის გამოყენებით (მაგ., მონაცემთა კომპლექსური ნაკადი ETL (Extract Transform Load)–სთვის, მონაცემთა სხვადასხვა ანალიზი).

Pig Latin-ს ბევრი უპირატესობა აქვს. იგი ეფუძნება ინტუიტიურ სინტაქსს, რომელიც ხელს უწყობს MapReduce-ის სამუშაო პროცესების მარტივ განვითარებას. მომხმარებლებს შეუძლიათ გამოიყენონ Pig Latin-ის ენა მონაცემთა ატვირთვისა და დამუშავების მიზნით. Pig Latin არის Java პროგრამირების ენის ალტერნატივა, რომელსაც აქვს Directed Acyclic Graph (DAG)-ის მსგავსი სკრიპტები. SQL-ისგან განსხვავებით, Pig-ს არ სჭირდება სქემა და შეუძლია ნახევრად სტრუქტურირებული და არასტრუქტურირებული მონაცემების დამუშავება. მას Hive-ზე მეტი მონაცემთა ფორმატების მხარდაჭერა აქვს. Pig-ის გაშვება შესაძლებელია როგორც ლოკალურად ერთ JVM გარემოში, ასევე განაწილებულ Hadoop-ის კლასტერზე.

JAQL არის დეკლარაციული ენა Hadoop-ის ეკოსისტემაში, რომელიც მხარს უჭერს მონაცემთა მასიურ დამუშავებას. ის მაღალი დონის მოთხოვნებს გარდაქმნის MapReduce პროცესებად. ის შეიქმნა ნახევრად სტრუქტურირებულ მონაცემებთან სამუშაოდ JSON (JavaScript Object Notation) ფორმატის საფუძველზე. ამასთან, ის შეიძლება გამოყენებულ იქნას მონაცემთა სხვა ფორმატებთან სამუშაოდ. ასე რომ, Pig-ის მსგავსად JAQL არ საჭიროებს მონაცემთა სქემას. JAQL გთავაზობთ რამდენიმე ჩაშენებულ ფუნქციას, ძირითად ოპერატორებს და I / O გადამყვანებს. ასეთი მახასიათებლები უზრუნველყოფს მონაცემთა დამუშავებას, შენახვას, თარგმნას და გადაყვანას JSON ფორმატში.

Apache Hive არის მონაცემთა სანახი სისტემა, რომელიც შექმნილია Apache Hadoop-ის გამოყენების გამარტივების მიზნით. MapReduce-ისგან განსხვავებით, რომელიც ფაილებით მართავს მონაცემებს HDFS-ში, Hive საშუალებას გაძლევთ წარმოადგინოთ მონაცემები სტრუქტურირებულ მონაცემთა ბაზაში, რომელიც უფრო ნაცნობია მომხმარებლებისთვის. Hive-ის მონაცემთა მოდელი ძირითადად ემყარება ცხრილებს. ასეთი ცხრილები წარმოადგენს HDFS დირექტორიებს და იყოფა სექციებად. თითოეული დანაყოფი კი შედგება ბაკეტებისგან.

Hive წარმოადგენს SQL-ის მსგავს ენას, სახელწოდებით HiveQL, რომელიც მომხმარებლებს საშუალებას აძლევს Hadoop-ზე დაფუძნებულ HDFS-ის ან Hbase-ის მონაცემებთან ქონდეთ წვდომა და სასურველი ოპერაციები განახორციელონ.

Hive არ არის შესაფერისი რეალურ დროში განხორციელებული ოპერაციებისათვის. Hadoop-ის მსგავსად, Hive განკუთვნილია მასშტაბირებადი დამუშავებისთვის, ასე რომ მცირე სამუშაოებსაც შეიძლება რამდენიმე წუთი დასჭირდეს. HiveQL მოთხოვნებს გარდაქმნის სამუშაოებში, რომელთა დამუშავება პაკეტებად ხორციელდება.

ISSN 1512-1232

Apache Spark არის ღია პროგრამული უზრუნველყოფის განაწილებული სისტემა, რომელიც შეიქმნა UC Berkeley AMPLab–ში. Spark Hadoop-ის მსგავსია, მაგრამ მუშაობის გასაუმჯობესებლად ის მონაცემებს სისტემის მეხსიერებაში ინახავს. ეს არის აღიარებული ანალიტიკური პლატფორმა, რომელიც უზრუნველყოფს სწრაფ, ადვილად გამოსაყენებელ და მოქნილ გამოთვლებს. Spark ანხორციელებს კომპლექსურ ანალიზს მონაცემთა დიდ ნაკრებზე. მართლაც, Spark აწარმოებს პროგრამებს 100x – ჯერ უფრო სწრაფად ვიდრე Hive და Apache Hadoop MapReduce სისტემის მეხსიერების საშუალებით. Spark დაფუძნებულია Apache Hive პროგრამის კოდების ბაზაზე. სისტემის მუშაობის გასაუმჯობესებლად, Spark–მა შეცვალა Hive– ის ფიზიკური შემსრულებელი ძრავა. სპარკს რამდენიმე პოპულარული პროგრამირების ენის მხარდაჭერა აქვს - Python, Java, Scala და R. იგი შედგება უამრავი ბიბლიოთეკისაგან, რომლებიც განკუთვნილია მრავალფეროვანი ამოცანების გადასაჭრელად, სტანდარტული SQL-დან დაწყებული მონაცემთა ნაკადებით თუ მანქანური სწავლებით დამთავრებული. Spark- ს შეუძლია იმუშაოს ყველა ფაილურ სისტემასთან, რომელსაც Hadoop-ის მხარდაჭერა აქვს. Spark-ის გაშვება შეიძლება როგორც ლეპტოპზე, ისე ათასობით სერვერიდან შემდგარ კლასტერზეც. ამის წყალობით, Spark არის საკმაოდ მოსახერხებელი სისტემა დიდი მონაცემების გარემოს აწყობის დასაწყებად და თანდათან მასშტაბების უსაზღვროდ გასაზრდელად.

Spark-ის მონაცემთა მოდელი ეფუძნება Resilient Distributed Dataset (RDD) აბსტრაქციას. RDD წარმოადგენს მხოლოდ წაკითხვად ობიექტებს, რომელიც ინახება მრავალი გამოთვლითი კვანძის მეხსიერებაში. მათი აღდგენა შესაძლებელია დანაყოფის დაკარგვის შემთხვევაშიც.

აღსანიშნავია, რომ სპარკში მონაცემებთან ურთიერთობის ფუნქციონალი მხოლოდ DataFrame-ების გამოყენებით არ შემოიფარგლება. Spark SQL ბიბლიოთეკის დამსახურებით, ნებისმიერი DataFrame შეგვიძლია დავარეგისტრიროთ დროებით ცხრილად ან წარმოდგენად (View), ხოლო მათგან შედეგი მივიღოთ კარგად ნაცნობ SQL-ზე დაწერილი ბრძანებების გამოყენებით. ეფექტური მუშაობის თვალსაზრისით, მნიშვნელობა არ აქვს, აპლიკაციის ლოგიკას DataFrame-ებზე დავწერთ თუ Spark SQL ბიბლიოთეკას გამოვიყენებთ – სპარკი ორივე შემთხვევაში იდენტურ სამუშაო გეგმებს აგენერირებს.

Spark SQL–ის ერთ-ერთი მნიშვნელოვანი მახასიათებელია ის, რომ იგი აერთიანებს ორ აბსტრაქციას: რელაციურ ცხრილებს და RDD-ს. ასე რომ, პროგრამისტებს SQL ბრძანებებთან ერთად შეუძლიათ კომპლექსური ანალიტიკის გამოყენებაც. კერძოდ, მომხმარებლებს შეუძლიათ შეასრულონ მოთხოვნები, როგორც გარე წყაროებიდან (მაგ. Parquet Files, Hive Tables) იმპორტირებულ მონაცემებზე, ასევე არსებულ RDD- ებში შენახულ მონაცემებზე. გარდა ამისა, Spark SQL საშუალებას გაძლევთ ჩაწეროთ RDD Hive-ის ცხრილებში ან Parquet ფაილებში. ეს ხელს უწყობს მონაცემთა მოთხოვნების სწრაფ პარალელურ დამუშავებას განაწილებულ დიდ მონაცემთა ნაკრებზე. Spark იყენებდა HiveQL მოთხოვნის ენას. აპლიკაციის სწრაფი განვითარებისათვის, შემდგომ შეიმუშავა Spark SQL. მისი საშუალებით სწრაფად არის ახალი ოპტიმიზაციების დანერგვა შესაძლებელი.

### d. მონაცემთა წვდომის ფენა: მონაცემთა მიღება: Sqoop, Flume, Chukwa და Kafka

Apach Sqoop არის ღია პროგრამული უზრუნველყოფის პროდუქტი. იგი უზრუნველყოფს Apache Hadoop-სა და სტრუქტურირებულ მონაცემთა ბაზას შორის

ISSN 1512-1232

მონაცემების ეფექტურად გადაცემას (როგორიცაა რელაციური მონაცემთა ბაზები, საწარმოს მონაცემთა სანახები და NoSQL მონაცემთა ბაზები). Sqoop გთავაზობთ ბევრ უპირატესობას. მაგალითად, ის უზრუნველყოფს სწრაფ მუშაობას, ხარვეზების მიმართ ტოლერანტულობას და სისტემის ოპტიმალურ გამოყენებას. იმპორტირებული მონაცემების ტრანსფორმაცია ხორციელდება MapReduce ან ნებისმიერი სხვა მაღალი დონის ენის გამოყენებით, როგორიცაა Pig, Hive ან JAQL. ის მარტივად ინტეგრირებადია HBase, Hive და Oozie-სთან. Sqoop HDFS–დან დაიმპორტებულ მონაცემებს მრავლობით ფაილად აგენერირებს. ეს ფაილები შეიძლება იყოს ტექსტური ფაილები, ორობითი Avro ან SequenceFiles, რომლებიც შეიცავს სერიულ მონაცემებს. Sqoop ექსპორტის დროს პარალელურად მუშავდება ტექსტური ფაილების ნაკრები და ახალ მწკრივებად ხდება ჩასმა სამიზნე მონაცემთა ბაზის ცხრილში.

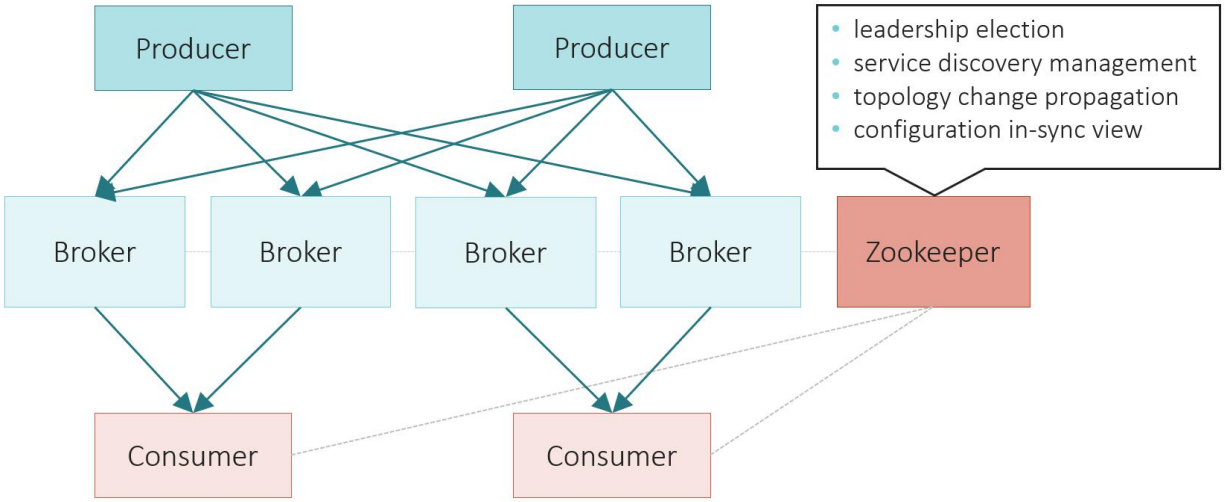
Flume უზრუნველყოფს მონაცემების შეგროვებას, ინტეგრაციასა და გადაცემას გარე მოწყობილობებიდან HDFS–ზე. მას აქვს მარტივი მოქნილი არქიტექტურა და ამუშავებს მონაცემთა მასიურ ნაკადებს განაწილებულ სისტემებში. Flume გთავაზობთ სხვადასხვა მახასიათებლებს, მათ შორის ხარვეზების მიმართ ტოლერანტულობას, საიმედო მექანიზმს მრავალი აღდგენის სერვისით. მიუხედავად იმისა, რომ Flume კარგად ინტეგრირდება Hadoop-თან, ის დამოუკიდებელი კომპონენტია, რომელსაც შეუძლია სხვა პლატფორმებზე მუშაობა. იგი ცნობილია თავისი შესაძლებლობებით, რომ აწარმოოს სხვადასხვა პროცესები ერთ აპარატზე. Flume–ის გამოყენებით, მომხმარებელს შეუძლია რეალურ დროში გაანალიზოს და გადაიტანოს მონაცემები სხვადასხვა წყაროებიან (როგორიცაა Avro RPC წყარო და syslog) მონაცემთა სანახებში (როგორიცაა HDFS და HBase).

Chukwa არის მონაცემთა შეგროვების სისტემა, რომელიც შექმნილია Hadoop-ის ეკოსისტემაში. Chukwa-ს მიზანია დიდი განაწილებული სისტემების მონიტორინგი. იგი იყენებს HDFS-ს მონაცემების შეგროვებისთვის სხვადასხვა პროვაიდერებისგან, ხოლო MapReduce-ს შეგროვებული მონაცემების ანალიზისთვის. ის უზრუნველყოფს შედეგების ვიზუალიზაციას, მონიტორინგს და ანალიზს.

Chukwa გთავაზობთ მოქნილ და ძლიერ პლატფორმას Big Data–სთვის. ეს საშუალებას აძლევს ანალიტიკოსებს შეაგროვონ და გააანალიზონ დიდ მონაცემთა ნაკრებები, აგრეთვე დააკვირდნენ და აჩვენონ შედეგები.

Apache Kafka - დიდ მონაცემთა ნაკადების დამუშავების პროგრამული უზრუნველყოფის პლატფორმა მაღალი გამტარუნარიანობითა და მცირე შეყოვნებით. ე.წ. publish/subscribe messaging სისტემების ერთ-ერთი წარმომადგენელია. ასეთი სისტემებისათვის დამახასიათებელია შემდეგი მოდელი: გამგზავნი (Publisher) აგენერირებს მონაცემებს (მესიჯებს), რომელიც რომელიმე კონკრეტული მიმღებისათვის არაა განკუთვნილი. კერძოდ, იგი აკლასიფიცირებს შეტყობინებებს და, როგორც წესი, თავს უყრის მათ ერთ ცენტრალურ წერტილში (ბროკერი, Broker). მიმღები (გამომწერი, Subscriber) კი ამ ბროკერიდან კითხულობს შეტყობინებებს. Apache Kafka-ს ხშირად ახასიათებენ, როგორც ნაკადების განაწილებულ პლატფორმას. მონაცემები კაფკაში არის შენახული საიმედოდ და თანმიმდევრულად. შეტყობინებების წაკითხვა კაფკადან ხდება დეტერმინისტულად. დამატებით, მონაცემები განაწილებულია კაფკაშიც ისე, რომ, პირველ რიგში, დამატებითი საფრთხეები არიდებულია თავიდან (რეპლიკაცია ხდება) და ასევე, სისტემის მასშტაბირებადობაც იოლია საჭიროების შემთხვევაში (ნახ. 1).

ISSN 1512-1232




ნახ. 3. Kafka Architecture

**e. მონაცემთა ნაკადი: Storm და Spark Streaming**

Storm არის ღია პროგრამული უზრუნველყოფის განაწილებული სისტემა. მას, Hadoop–ისგან განსხვავებით, აქვს უპირატესობა გაუმკლავდეს მონაცემთა რეალურ დროში დამუშავებას რომელიც შექმნილია მონაცემების პაკეტებად პროცესინგისთვის.

Trident API-ზე დაყრდნობით, Flume-თან შედარებით, Storm არის ბევრად ეფექტური კომპლექსური მოთხოვნების დამუშავების განხორციელებაში.

Storm ემყარება ტოპოლოგიას, რომელიც შედგება შემდეგი კომპონენტებისგან: spouts, bolts და streams. Spout არის ნაკადების წყარო. Bolt გამოიყენება შემავალი ნაკადების დასამუშავებლად გამომავალი ნაკადების წარმოების მიზნით. ამრიგად, Storm მიზანშეწონილია ნაკადებზე ტრანსფორმაციების შესრულებისთვის "spouts" და "bolt"-ების გამოყენებით.

Storm არის ადვილად გამოსაყენებელი, სწრაფი, მასშტაბირებადი და ხარვეზებთან ტოლერანტული სისტემა. Storm ავტომატურად ანახლებს ჩავარდნილ პროცესებს. თუ პროცესი განმეორებით დახარვეზდა, Storm მას ამისამართებს სხვა კვანძზე და იქ ახორციელებს მის გადატვირთავს. ის ბევრგან გამოიყენება ეფექტურად, მაგალითად როგორიცაა რეალურ დროში ჩატარებული ანალიტიკა, ონლაინ მანქანური სწავლება, უწყვეტი გამოთვლა და განაწილებული RPC (Remote Procedur Call).

Storm გამოიყენება შედეგების მოსამზადებლად, რომელთა შემდეგი ანალიზი შესაძლებელია Hadoop-ის სხვა საშუალებების გამოყენებით. მას შეუძლია წამში გადაამუშავოს მილიონობით ჩანაწერი.

როგორც ზემოთ ავღნიშნე, Apache Spark არის უნიფიცირებული გამოთვლითი ძრავა და ბიბლიოთეკების ერთობლიობა კლასტერებზე მონაცემების პარალელური პროცესინგისათვის.

Spark Streaming არის კიდევ ერთი კომპონენტი, რომელიც უზრუნველყოფს პროცესების ავტომატურ პარალელურ შესრულებას, ასევე მასშტაბირებად და ხარვეზების მიმართ ტოლერანტული ნაკადების დამუშავებას. ის საშუალებას აძლევს მომხმარებლებს, პაკეტების მსგავსი დავალებები შექმნან Java-სა და Scala-ში. შესაძლებელია მონაცემთა პაკეტებისა და ნაკადების დამუშავების ინტეგრირება. თითოეულ ნაკადზე შესრულებული პროცესი წარმოადგენს მცირე პაკეტების

ISSN 1512-1232

დამუშავების ერთობლიობას. მისი გამოთვლა RDD-ში შენახულ მეხსიერების მონაცემებზე ხორციელდება.

### f. მონაცემთა ანალიზი - Apache Mahout, R, Apache Spark MLlib

Apache Mahout არის ღია მანქანური სწავლების პროგრამული უზრუნველყოფის ბიბლიოთეკა. Mahout შეიძლება ინტეგრირდეს Hadoop-ის ეკოსისტემაში და MapReduce-ის საშუალებით შეასრულოს ალგორითმები. Mahout სხვა პლატფორმებზე მუშაობისთვისაც გამოიყენება.

Mahout ძირითადად ჯავას ბიბლიოთეკების ნაკრებია. მას აქვს უპირატესობა ეფექტურად განახორციელოს მასშტაბირებადი მანქანური სწავლების პროგრამები და ალგორითმები მონაცემთა დიდ ნაკრებზე. Mahout ბიბლიოთეკა გთავაზობთ ანალიტიკურ შესაძლებლობებს და მრავალჯერ ოპტიმიზირებულ ალგორითმებს. მას აქვს სხვადასხვა ბიბლიოთეკების მხარდაჭერა, როგორებიც არის კლასტერიზაცია (მაგ. K-means, fuzzy K-means, Mean Shift), კლასიფიკაცია, კოლაბორაციული ფილტრაცია (პროგნოზებისა და შედარებისთვის), ნიმუშებისა და ტექსტის დამუშავება (ტექსტის დასკანერებისთვის და კონტექსტური მონაცემების მინიჭებისთვის). Mahout-ის დამატებითი ინსტრუმენტები მოიცავს თემის მოდელირებას, განზომილების შემცირებას, ტექსტის ვექტორიზაციას, მათემატიკის ბიბლიოთეკას და ა.შ. დიდი კომპანიები, რომლებმაც დანერგეს მანქანური სწავლების მასშტაბირებადი ალგორითმები, არიან Google, IBM, Amazon, Yahoo, Twitter და Facebook.

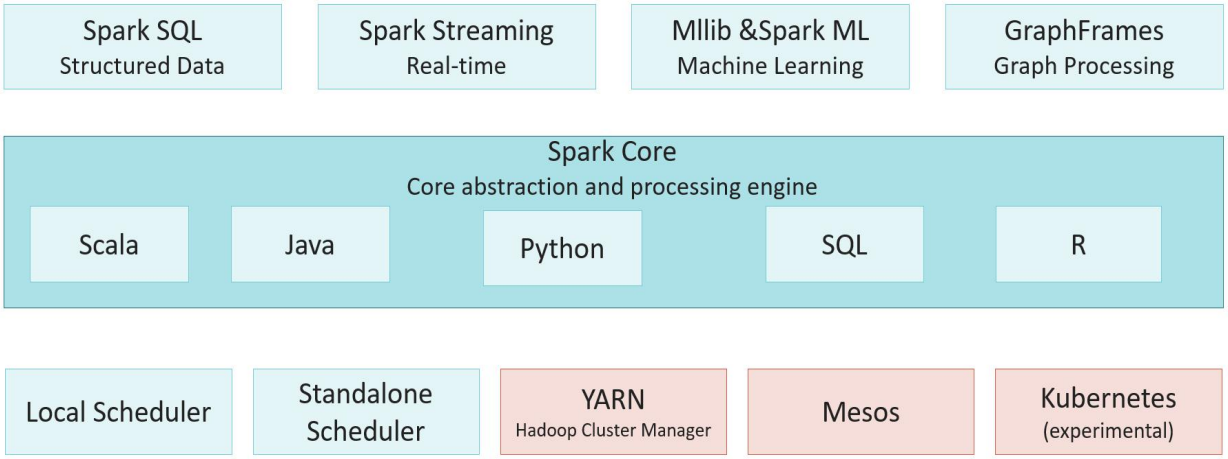
R არის პროგრამირების ენა სტატისტიკური გამოთვლების, მანქანური სწავლებისა და გრაფიკისთვის. R არის უფასო, ღია პროგრამული უზრუნველყოფა. R- პროექტის განვითარება ეყრდნობა მომხმარებლების, დეველოპერებისა და კონტრიბუტორების საზოგადოებას. R პროგრამირების ენა მოიცავს კარგად განვითარებულ, მარტივ და ეფექტურ ფუნქციებს, მათ შორის ციკლებს, მომხმარებლის მიერ განსაზღვრულ რეკურსიულ ფუნქციებს და სხვადასხვა საშუალებებს. მსხვილი კომპანიები (როგორიცაა Cloudera, Hortonworks და Oracle) იყენებენ R-ს დიდ მონაცემთა ანალიტიკისთვის.

R-ის ერთი ნაკლი არის მისი შეზღუდული შესაძლებლობები გაუმკლავდეს უკიდურესად დიდ მონაცემთა ნაკრებებს ერთი გამოთვლითი კვანძის მეხსიერების შეზღუდვების გამო.

R გთავაზობთ კლასიფიკაციის მოდელების უფრო სრულ კომპლექტს (ალგორითმების ტიპებთან დაკავშირებით) Mahout-თან შედარებით. ხოლო მეხსიერების მართვის პრობლემების გამო, შეიძლება უფრო პრაქტიკული იყოს Mahout, Spark, SAS ან სხვა პროგრამული უზრუნველყოფის გამოყენება ფართო გამოთვლების უკეთ შესრულებისთვის.

Spark პროექტი შედგება მრავალი კომპონენტისგან (ნახ. 4). მონაცემთა ანალიტიკისთვის კი გამოიყენება Spark MLlib. MLlib არის განაწილებული მანქანური სწავლების პროგრამული გარემო, რომელიც შექმნილია Spark–ის ბიბლიოთეკების გამოყენებით. MLlib გთავაზობთ სხვადასხვა სახის მანქანური სწავლების ოპტიმიზირებულ ალგორითმებს, როგორიცაა კლასიფიკაცია, რეგრესია, კლასტერიზაცია და კოლაბორაციული ფილტრაცია. Mahout- ის მსგავსად, MLlib გამოიყენება მანქანური სწავლების კატეგორიებისთვის. Mahout-ისგან განსხვავებით, Mlib რეგრესიის მოდელების მხარდაჭერა აქვს. MLlib Mahout–თან შედარებით არის ახალგაზრდა ტექნოლოგია.

ISSN 1512-1232




ნახ. 4: Spark Components

## 3. დასკვნა

მკვლევარები და პროფესიონალები დიდ მონაცემთა შესწავლისას და სხვადასხვა წყაროებიდან ინფორმაციის მოპოვებისას დიდი გამოწვევების წინაშე დგანან. სირთულეები არსებობს სხვადასხვა ოპერაციების დროს, როგორიც არის: მონაცემთა მოძიება, შენახვა, გაზიარება, ანალიზი, მართვა და ვიზუალიზაცია. გარდა ამისა, არსებობს უსაფრთხოებისა და კონფიდენციალურობის საკითხები.

მონაცემთა მზარდმა რაოდენობამ გარდაუვალი გახადა Big Data ტექნოლოგიების დანერგვა და გამოყენება. დიდი სისტემებისთვის მონაცემთა მართვის სწორი არქიტექტურის შერჩევა დიდ სირთულეებთან არის დაკავშირებული. ბევრი გამოწვევა ახლავს თან დიდი მონაცემების ტექნოლოგიებით მიღებულ გადაწყვეტილებებს. დღეს მრავლობითი პროგრამული უზრუნველყოფა არის შექმნილი ზღვა ინფორმაციის მართვისა და დამუშავებისთვის. ამ თემის აქტიალურობიდან გამომდინარე, სტატიაში განვიხილე Big Data-ს წამყვანი პროგრამული უზრუნველყოფანი, რომელთაც მოწინავე ადგილი უკავიათ 21-ე საუკუნის მიღწევებში.

## ლიტერატურა

1. Hong L., Luo M., Wang R., Lu P., Lu. W., Lu L. Big Data in Health Care: Applications and Challenges. Data and Information Management. 2018, 2(3), 175-197
2. Aboudi E.N., Benhlima L. Big Data Management for Healthcare Systems: Architecture, Requirements, and Implementation. Adv Bioinformatics. 2018: 4059018.
3. Oussous A., Benjelloun F.Z., Lahcen A.A, Belfkih S. Big Data technologies: A survey. Journal of King Saud University - Computer and Information Sciences. 2018, Vol. 30 (4), pp.431-448
4. ჯაფარიძე ა. დიდი მონაცემები - ნაკადების დამუშავება რეალურ დროში. სამაგისტრო ნაშრომი. 2019, თბილისი

## OVERVIEW OF BIG DATA TECHNOLOGIES

Discusses the leading Big Data software that is at the forefront of 21st century achievement.
Is presented the importance of Big Data technologies when it becomes impossible to deal with Big Data by traditional methods.

Article received: 2021-05-22