ივ. ჯავახიშვილის სახ. თბილისის სახელმწიფო უნივერსიტეტი

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

ნიკა დათუაშვილი

დიდი მონაცემების შენახვის ტექნოლოგიები

BigData Storage Teqnologies

სამაგისტრო პროგრამა: ინფორმაციული ტექნოლოგიები

სამაგისტრო ნაშრომი შესრულებულია კომპიუტერული მეცნიერებების მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

სამაგისტრო ნაშრომის ხელმძღვანელი
ასისტენტ-პროფესორი: პაპუნა ქარჩავა

თბილისი
2017

ანოტაცია

BigData ახალი ინოვაციური პროექტი, რომელიც გამოჩნდა სულ რაღაც ათეული წელია, რომლის განვითარებაც სწორედ იმ ტექნოლოგიების განვითარებამ გამოიწვია რომლებიც გრანდიოზული მასშტაბის მქონე ინფორმაციის მიმოსვლას ახორციელებს.

BigData ეს არის ტექნოლოგია რომელიც უზარმაზარი მასშტაბის ინფორმაციის გადამუშავების საშუალებას გვაძლევს თანამედროვე სამყაროში,ახდენს ამ ინფორმაციის რეალურ დროში აღქმის, დალაგების, ძებნის და მომავლის განჭვრეტის საშუალებას. ამ ტექნოლოგიის გამოყენებით ჩვენ არა მხოლოდ გრანდიოზული მასშტაბის ინფორმაციის გადამუშავება შეგვიძლია არამედ შესაძლებელია რეალურ დროში დამუშავებული ინფორმაციის საფუძველზე სავარაუდო მომავლის ქმედებების გამოთვლაც.

Abstract

BigData is a new innovative project, which has appeared for at least a decade, has been developed by the development of technologies that are carrying out large-scale information.

BigData is a technology that enables a large scale information processing in the modern world, realizing the real time perception, sorting, search and Guessing the future by using this information. Using this technology, we not only can process the grand scale of information, but also calculate Guessing the future actions based on real-time information.

სარჩევი

შესავალი

21-ე საუკუნე წარმოადგენს ე.წ. BigData (დიდი მოცულობის) მონაცემების საუკუნეს. სხვადასხვა წყაროდან მომავალი ინფორმაცია იზრდება შემაშფოთებელი სისწრაფით. 2016 წლის ივლისის მონაცემებით ინტერნეტის მომხმარებელთა რაოდენობამ გადააჭარბა 3,5 მილიარდს, რომელთაგან 2,5 მილიარდი მოდის განვითარებად ქვეყნებზე. დღეისთვის ეს მონაცემი შეიძლება იყოს გაზრდილი. Google-ის შეფასებით web-გვერდების რაოდენობამ გადააჭარბა რამდენიმე ტრილიონს. ყოველდღიურად Facebook-ი აგენერირებს 25 TB-ზე მეტი მოცულობის log მონაცემებს, Twitter-ი აგენერირებს 12 TB-ზე მეტი მოცულობის შეტყობინებას და ა.შ. ეს მაჩვენებელი ყოველწლიურად იზრდება.

ინტერნეტ სერვისების, როგორიცაა email, blogging, social networking, search და e-commerce, სწრაფმა განვითარებამ არსებითად შეცვალა web-მომხმარებლის ყოფაქცევა და ტენდენციები, როდესაც ის ცდილობს შექმნას პროდუქტი, გახადოს ის ხელმისაწვდომი, გააზიაროს და შეიძინოს რაიმე პროდუქტი. მაგალითად, Amazon-დან ყიდულობს სახელმძღვანელოს, eBay-ზე ყიდის ნივთს, მეგობრებთან ურთიერთქმედებს Facebook-ის ან Linkedin-ის მეშვეობით და ა.შ.

მთელი წლის განმავლობაში ორგანიზაციამ შეიძლება დააგენერიროს PB[1]-ზე მეტი მოცულობის ინფორმაცია: web-გვერდების, ბლოგების, ძებნის, email-ის, დოკუმენტების და სხვა მრავალი სერვისის გამოყენებით. მონაცემთა შეფასების არსებული სისტემების მიხედვით მონაცემთა მთლიანი მოცულობის 80% არის ნაწილობრივ სტრუქტურირებული ან საერთოდ არაა სტრუქტურირებული. ყოველი კომპანია ცდილობს (გარკვეული საქმიანობის წარმოების მიზნით) გამოიყენოს მოხერხებული სერვისები და ინოვაციური ტექნოლოგიები მონაცემთა ანალიზისა და მათი დამუშვების ამოცანების გადაწყვეტაში. ასეთ პროცესებში დახარჯულმა დიდმა დრომ შეიძლება ორგანიზაცია მიიყვანოს ბიზნეს-შესაძლებლობების დაკარგვამდე.

ბოლო რამდენიმე ათწლეულის მანძილზე გაზრდილმა გამოთვლითმა სიმძლავრემ უკანა „პლანზე“ გადასწია მონაცემთა გადაცემის სისწრაფის საკითხი, რამაც

[1] Petabait - პეტაბაიტი

მიგვიყვანა გამოთვლითი არქიტექტურის დაბალ დონეზე გადასვლის და მონაცემთა დამუშავების ფართომასშტაბიან მექანიზმის პარადიგმის შეცვლამდე.

Microsoft ფირმის მკვლევარის და მონაცემთა ბაზის პროგრამული უზრუნველყოფის ფუძემდებელის Jim Gray-ის მტკიცებით არსებობდა კიდევ ერთი პარადიგმა, რომელთანაც გამკლავების მიზნით საჭირო იყო ახალი თაობის ისეთი გამოთვლითი ინსტრუმენტის შემუშავება, რომელიც შესაძლებელს გახდიდა მონაცემთა მართვას, ვირტუალიზაციას და გაანალიზებას. გამოთვლითი მანქანების არქიტექტურა უფრო და უფრო დაუბალანსირებელია, დაყოვნება მრავალრიცხოვან პროცესორებსა და ინფორმაციის შენახვის მოწყობილობებს შორის ყოველწლიურად იზრდება, რაც ინფორმაციულად დიდი მოცულობის გამოთვლების წარმოებას ართულებს.

1. რა არის BigData?

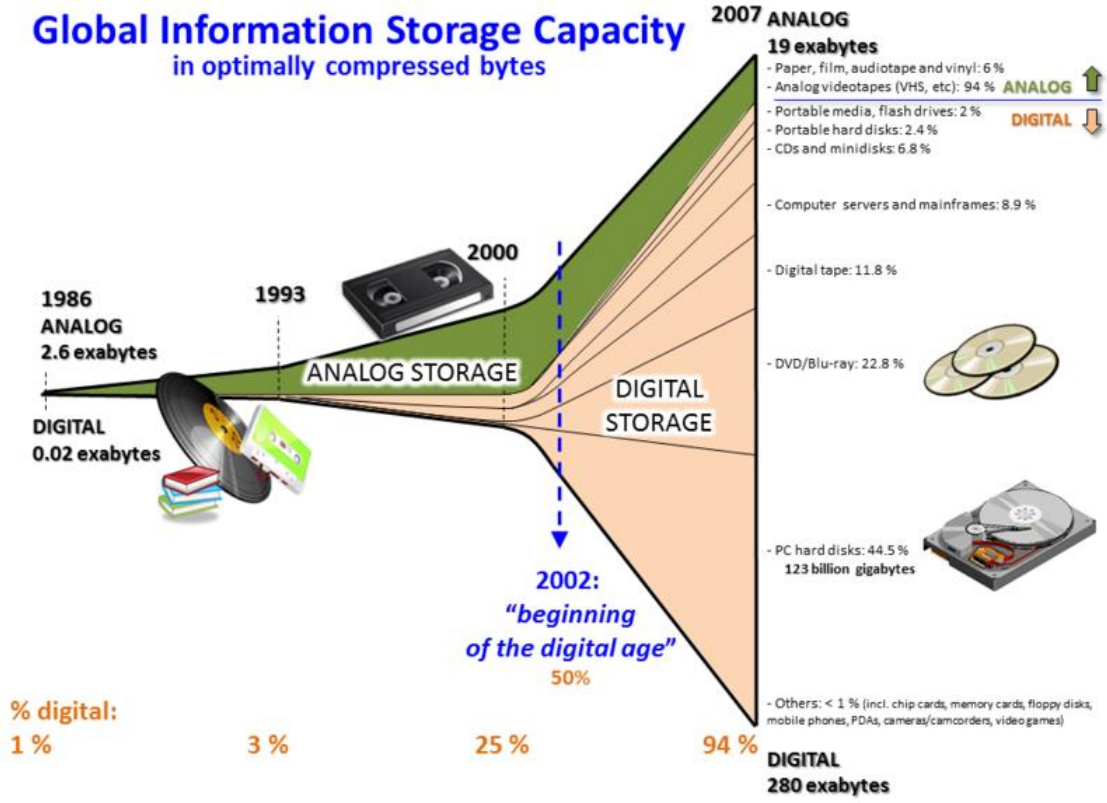
ტერმინი „BigData"-ს გამოჩენას უკავშირებენ კლიფორდ რიჩს (Clifford Reach), რომელმაც 2008 წელს ჟურნალ „Nature"-ისთვის (რომლის რედაქტორიც თვითონ იყო) მოამზადა სპეციალური გამოცემა თემაზე „როგორი ზეგავლენა შეიძლება იქონიოს მეცნიერებაზე ტექნოლოგიებმა, რომლებიც იძლევიან დიდ მონაცემებთან მუშაობის შესაძლებლობას". სტატიაში თავმოყრილი იყო მასალები დასამუშავებელი მონაცემების სწრაფი ზრდის და მათი მრავალსახეობის ფენომენზე და „მოცულობიდან ხარისხზე" ნახტომის პარადიგმის ტექნოლოგიურ პერსპექტივებზე. ავტორის მიერ ტერმინი შემოთავაზებული იყო ინგლისურენოვან გარემოში არსებული მეტაფორების „big oil" (დიდი ნავთობი), „big ore" (დიდი მადნეული) და სხვა, ანალოგიით.

მიუხედავად იმისა, რომ ტერმინი პირველად გამოჩნდა აკადემიურ სფეროში და გულისხმობდა სამეცნიერო მონაცემების სწრაფი ზრდის და მრავალსახეობის პრობლემას, ის მალევე გახდა ფართოდ გამოყენებადი ინფორმაციული სისტემების მსხვილი მწარმოებლების მიერ როგორებიც არიან IBM, Oracle, Microsoft, Hewlett-Packard, EMC.

BigData არის ტერმინი, რომელიც დაკავშირებულია მონაცემთა იმდენად დიდ ნაკრებთან, რომ მათი დამუშავება მონაცემების დამუშავების ტრადიციული საშუალებების გამოყენებით არის შეუძლებელი. მონაცემთა დამუშავების პრობლემა

გულისხმობს მონაცემთა მოპოვებას, შენახვას, ანალიზს, ძებნას, გაზიარებას, გადაცემას, ვიზუალიზაციას, მოთხოვნას, განახლებას, ინფორმაციის დაცულობას და ა.შ.

„BigData“ ტერმინი მონაცემთა დამუშავებისთვის ხშირად მიმართავს ე.წ. Predictive analytics (წინასწარმეტყველების ანალიტიკა), user behavior analytics (მომხმარებლის ყოფაქცევის ანალიტიკა) ან სხვა რომელიმე მონაცემთა ანალიტიკის გამოყენებას, რომელიც იძლევა დიდი მონაცემებიდან ანალიზისთვის გარკვეული მოცულობის მონაცემების ამოღების შესაძლებლობას. მონაცემთა ანალიზიდან შეიძლება გამოიკვეთოს ბიზნესის წარმატების, დაავადების პრევენციის, ომის საფრთხის აცილების და სხვა ტენდენციები.



ნახ. 1. მონაცემთა ზრდის ტენდენცია

მონაცემთა ნაკრების სწრაფ ზრდას ხელს უწყობს ის გარემოება, რომ მათი მოპოვება ხდება დიდი რაოდენობის, იაფი ღირებულების და სხვადასხვა სახის ინფორმაციის მოპოვების საშუალებების გამოყენებით, როგორიცაა მობილური აპარატები, ანძები, პროგრამების ე.წ. log ფაილები, ვიდეო კამერები, მიკროფონები, რადიო სიხშირის იდენტიფიკატორების (radio frequency identification - RFID) წამკითხველები, უკაბელო სენსორული ქსელები და სხვა. მონაცემების შენახვის

ტექნოლოგიები 1980 წლამდე იძლეოდნენ მონაცემების შესანახად გაორმაგებულ მოცულობას დაახლოებით ყოველ 40 თვეში. 2012 წლის მონაცემებით ყოველდღიურად გენერირდება 2.5 ეგზაბაიტი (2.5×10^{18}) მონაცემი (ნახ. 1).

2012 წელს დ. ბოიდისა და კ.კლაუფორდის სტატიაში მოყვანილი იქნა BigData -ს განმარტება, როგორც კულტურული, ტექნოლოგიური და სამეცნიერო ფენომენი, რომელიც თავის შიგნით აერთიანებდა:

1. ტექნოლოგია: გამოთვლითი სიმძლავრის მაქსიმიზირებას და დიდი მოცულობის მონაცემების შეგროვების, ანალიზის, დაკავშირებულობის და შედარების ალგორითმების სირთულეს;
2. ანალიზი: დიდი მოცულობის მონაცემების ნაკრების გამოსახვა იმ ფორმით, რომ შესაძლებელი იყოს ეკონომიკური, სოციალური, ტექნიკური და იურიდიული მტკიცებულებების მისაღები სტრუქტურის იდენტიფიცირება;
3. მიფოლოგია: საყოველთაო მტკიცებულება, რომ BigData წარმოადგენს ცოდნის უფრო მაღალ ფორმას.

თუ გადავხედავთ Wikipedia-ს შეიძლება მოვძებნოთ BigData-ს განმარტება, რომელიც ეყრდნობა გასაღებ პუბლიკაციებს და ახდენენ ზემოხსენებულ ჟურნალში მოყვანილი ცნების გაფართოებას.

2. BigData მონაცემების მახასიათებლები

BigData მონაცემების გამოჩენის დღიდან BigData მონაცემების მახასიათებლების როლში გამოყოფენ 5 'V'-ს (ნახ. 2), ესენია:

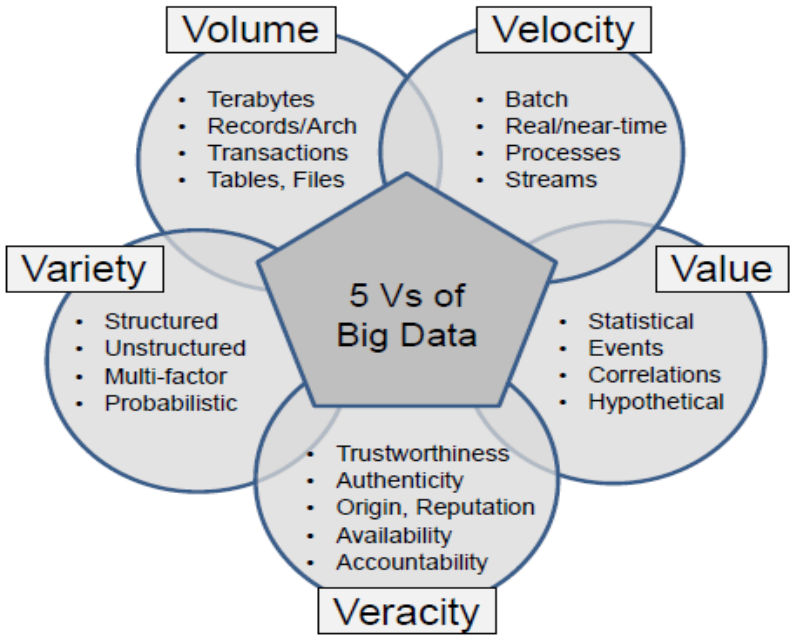
Volume (მოცულობა) - გენერირებული და შენახული მონაცემების მოცულობა. მონაცემების მოცულობა განსაზღვრავს ღირებულებას და პოტენციურ გაგებას წარმოდგენილი მონაცემები მიეკუთვნება თუ არა BigData -ს;

Variety (მრავალფეროვნება) - მონაცემების ტიპი და მახასიათებელი. ეს იძლევა მონაცემთა ანალიზის საჭიროებისას მიღებული შედეგების ეფექტური გამოყენების შესაძლებლობას;

Velocity (სწრაფქმედება) - გულისხმობს მონაცემების სწრაფ და ხარისხიან დამუშავებას;

Variability (ცვალებადობა) - მონაცემების ცვალებადობა შეიძლება იყოს ხელისშემშლელი ფაქტორი მონაცემებზე მიმართვის და მათი მართვის პროცესში;

Veracity (სიზუსტე) - მოპოვებული მონაცემების სიზუსტემ შეიძლება მნიშვნელოვანი ზეგავლენა იქონიოს მიღებულ შედეგზე.



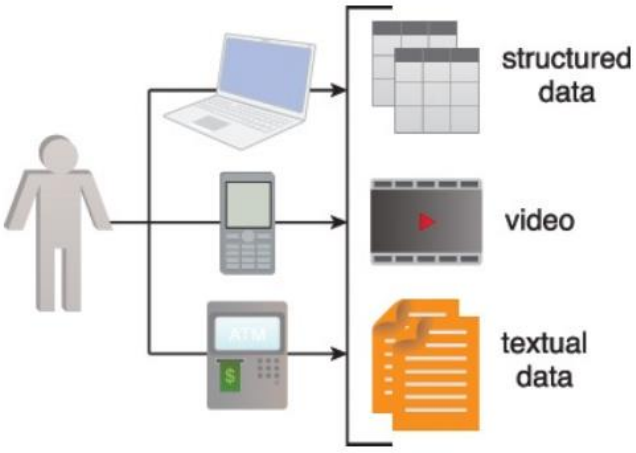
ნახ. 2. BigData-ს მახასიათებლები

3. BigData მონაცემები

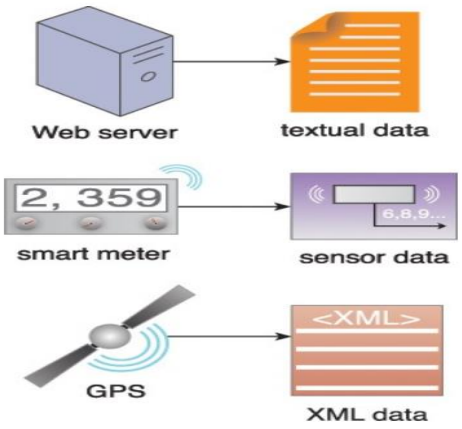
მონაცემები, რომლებსაც მიაკუთვნებენ BigData-ს შეიძლება წარმოიქმნას ორი გზით:

1. ადამიანის მიერ;
2. მანქანის (კომპიუტერი) მიერ.

ადამიანის მიერ გენერირებულ მონაცემებს განეკუთვნება ადამიანის მიერ სისტემასთან ურთიერთქმედების შედეგად მიღებული მონაცემები. ამ ტიპის მონაცემებს განეკუთვნება online სერვისებით ან ციფრული მოწყობილობების მიერ გენერირებული მონაცემები (ნახ. 1.16).



ნახ. 1.16. ადამიანის მიერ გენერირებული მონაცემები



ნახ. 1.17. მანქანის მიერ გენერირებული მონაცემები

მანქანის მიერ გენერირებულ მონაცემებს მიეკუთვნება პროგრამული ან აპარატული უზრუნველყოფით დაგენერირებული მონაცემები, რომლებიც შეესაბამებიან რეალური სიტუაციის აღწერას. მაგალითად, ე.წ. log ფაილები შეიცავენ უსაფრთხოების სერვისების მიერ დაგენერირებულ მონაცემებს ავტორიზირებული საქმიანობის შესახებ, ხოლო ორგანიზაციის მიერ წარმოებული პროდუქტის გაყიდვის სისტემა აგენერირებს მომხმარებლის მიერ შესაბამისი პროდუქტის შეძენისას განხორციელებულ ტრანზაქციის მონაცემებს ან სხვადასხვა პროდუქტის დათვალიერების შესაბამის მონაცემებს. აპარატურული თვალსაზრისით მანქანის მიერ დაგენერირებულ მონაცემს წარმოადგენს მონაცემები, რომელიც გადაიცემა მაგალითად, მობილურ მოწყობილობაში ჩაშენებული მრავალრიცხოვანი მიკროკონტროლერების მეშვეობით. ასეთი მიკროკონტროლერე-ბიდან შესაძლებელია მიღებული იქნას ისეთი მონაცემი, როგორიცაა მაგალითად, სიგნალის გადამცემი ანძის ადგილმდებარეობა, გადაცემული სიგნალის სიმძლავრე (ნახ. 1.17).

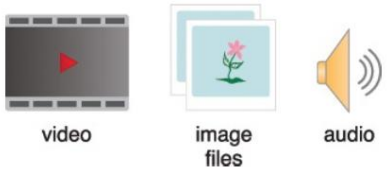
3.1. მონაცემების ტიპები

მიუხედავად იმისა თუ რა ფორმით მოხდა მონაცემების დაგენერირება მონაცემების მთლიანი ნაკრები შეიძლება დაიყოს შემდეგ ტიპებად:

1. <u>სტრუქტურირებული მონაცემები</u>, რომელიც შეესაბამება, მონაცემების გარკვეულ მოდელს ან სქემას და ხშირ შემთხვევაში ინახება ცხრილის სახით. ამ ფორმით

შენახვა იძლევა სხვადასხვა ობიექტებს შორის კავშირების დადგენის შესაძლებლობას, ამიტომ უმეტეს შემთხვევაში ასეთი მონაცემები ინახება სტანდარტულ რელაციურ მონაცემთა ბაზებში. ასეთი მონაცემების დაგენერირება შეიძლება მოხდეს კორპორატიული პროგრამული უზრუნველყოფით ან ინფორმაციული სისტემით, როგორიცაა ERM ან CRM;

2. <u>არასტრუქტურირებული მონაცემები</u>, რომელიც არ შეესაბამება მონაცემთა არანაირ სქემას ან მოდელს. ითვლება, რომ ასეთი მონაცემების მოცულობა ნებისმიერ ორგანიზაციაში არის დაახლოებით 80%. არასტრუქურირებული მონაცემების ზრდის ტემპი სტრუქტურირებულ მონაცემთა მოცულობის ზრდის ტემპთან მიმართებაში საკმარისად სწრაფია. (ნახ. 1.19-ზე ნაჩვენებია არასტრუქტურირებული მონაცემების უმეტესი ტიპები.) ამ ტიპის მონაცემები ხშირად წარმოდგენილია ტექსტური ან ორობითი ფორმით და გადაიცემა, როგორც ფაილი, რომელიც არის ავტონომიური და არარელაციური. ტექსტური ფაილი შეიძლება შეიცავდეს მონაცემებს შეტყობინებებიდან ან ბლოგიდან. ორობითი ფაილი ხშირად არის მედია ფაილი, რომელიც შეიცავს გამოსახულებას, აუდიო- ან ვიდეო-მონაცემებს. ტექნიკურად ტექსტური და ორობითი ფაილის სტრუქტურას განსაზღვრავს ფაილის სტრუქტურა, მაგრამ ეს მომენტი იგნორირებულია და ასევე ისიც, რომ იყოს არასტრუქტურირებული მონაცემთა ფორმატის მიმართ. არასტრუქტურირებულ მონაცემებთან სამუშაოდ სტანდარტული რელაციური ბაზების ნაცვლად გამოიყენება NoSQL ბაზები;



ნახ. 1.19. არასტრუქტურირებულ მონაცემთა ზოგადი ტიპი

3. <u>ნაწილობრივ-სტრუქურირებულ</u> მონაცემებს გააჩნიათ სტრუქტურის და არაწინააღმდეგობრიობის გარკვეული დონე, მაგრამ საკუთარი არსით არაა რელაციური. ნაცვლად ამისა ნაწილობრივ-სტრუქტურირებული მონაცემები არის იერარქიული და ეყრდნობა გრაფებს. ამ ტიპის მონაცემები ჩვეულებრივ შენახულია ტექსტის შემცველ ფაილებში. ნაწილობრივ-სტრუქტურირებული

მონაცემების მაგალითს წარმოადგენს XML და JSON გაფართოების მქონე ფაილები (ნახ. 1.20). ასეთი მონაცემების დამუშავება უფრო ადვილია ვიდრე არასტრუქტურირებული მონაცემებისა.

ნახ. 1. 20. XML, JSON და სენსორის მონაცემები

ნაწილობრივ სტრუქტურირებული

ნაწილობრივ-სტრუქტურირებული მონაცემების საერთო წყაროს მოიცავს ელექტრონული მონაცემების გაცვლა (EDI), ელექტრონული ცხრილები, RSS არხები და მიკროკონტროლერის მიერ გენერირებული მონაცემები. ასეთ მონაცემებს დამუშავების და შენახვის მოქმედებების მიმართ ხშირად გააჩნიათ სპეციალური მოთხოვნები, განსაკუთრებით თუ საბაზისო ფორმატი არ ეყრდნობა ტექსტს.

3.2. მეტამონაცემები

მეტამონაცემები არის ფაილთან დაკავშირებული მონაცემები (ატრიბუტები), რომლებიც იძლევიან ინფორმაციას მონაცემთა ნაკრების და სტრუქტურის შესახებ. ამ ტიპის მონაცემები ძირითადად გენერირდება მანქანის მიერ და შეიძლება დამატებული იქნას მონაცემებზე. მეტამონაცემების კონტროლი განსაკუთრებით მნიშვნელოვანია BigData მონაცემების დამუშავების, შენახვის და ანალიზის ოპერაციებისთვის, ვინაიდან ისინი იძლევიან ინფორმაციას მონაცემების კავშირზე და წარმომავლობაზე მონაცემების დამუშავების მომენტში. მეტამონაცემების მაგალითს შეიცავს:

- XML-ტეგი, რომელიც შეიცავს ინფორმაციას დოკუმენტის ავტორზე და მისი შექმნის დროზე;
- ატრიბუტები, რომლებიც შეიცავენ ინფორმაციას დოკუმენტის ზომაზე, მასზე დაშვების უფლებებზე და ა.შ.

BigData მონაცემების დამუშავების პროგრამა ეყრდნობა მეტამონაცემებს განსაკუთრებით ნაწილობრივ-სტრუქტურირებული და არასტრუქტურირებული მონაცემების დამუშავებისას.

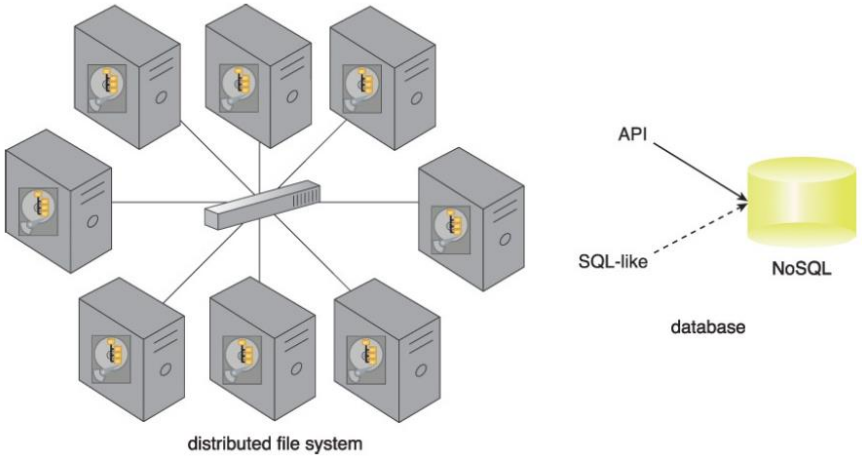
4. დიდი მონაცემების შენახვის ტექნოლოგიები

მონაცემების შენახვის ტექნოლოგიები სწრაფად ვითარდება და სერვერიდან ის გადაიზარდა ქსელში მონაცემების შესანახ ტექნოლოგიად. თანამედროვე გამანაწილებელი (შერწყმული, ჰიბრიდული) არქიტექტურა ერთ კომპონენტად აერთიანებს გამოთვლებს, შესანახ მოწყობილობას, მეხსიერებას და ქსელს, სადაც არქიტექტურა შეიძლება იმართებოდეს მხოლოდ ერთი ადგილიდან. ასეთი მიდგომა განსაკუთრებით მნიშნელოვანია დიდი მონაცემების დამუშავებისას.

დიდმა მონაცემებმა გარკვეულწილად შემოსაზღვრა მეხსიერების და შესანახი მოწყობილობის მოცულობა კლასტერებზე. დიდი მეხსიერების საჭიროების შემთხვევაში „ჰორიზონტალური მასშტაბირებადობა“ იძლევა მეხსიერების მოცულობის გაზრდის (გაფართოების) შესაძლებლობას კლასტერზე ახალი კვანძების (მოწყობილობის) დამატების გზით. დიდი მონაცემების რეალურ დროში ანალიზისთვის მნიშვნელოვანია როგორც მეხსიერების ისე, შესანახი მოწყობილობების ინოვაციური ტექნოლოგიები.

4.1. დისკური შესანახი მოწყობილობა

დისკური შესანახი მოწყობილობები იძლევიან BigData მონაცემების დიდი ხნით და იაფი ღირებულებით შენახვის შესაძლებლობას. ასეთ მოწყობილობაზე შესაძლებელია იმპლემენტირებული იქნას გამანაწილებელი ფაილური სისტემა ან მონაცემთა ბაზის სისტემა (ნახ. 7.1).



ნახ. 7.1. გამანაწილებელი ფაილური სისტემა ან მონაცემთა ბაზის სისტემა

განვიხილოთ ორივე ტექნოლოგია ცალცალკე:

4.1.1. გამანაწილებელი ფაილური სისტემა

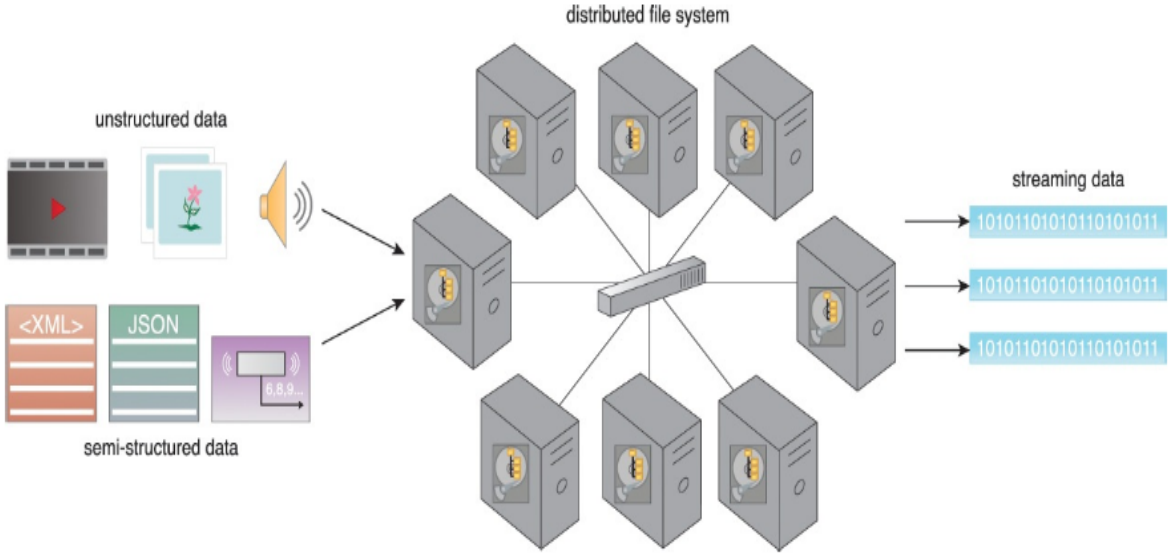
ჩვეულებრივი ფაილური სისტემის მსგავსად გამანაწილებელი ფაილური სისტემა (distributed file system) იძლევა დიდი მონაცემების შენახვის შესაძლებლობას. მას გააჩნია ე.წ. less data storage (ნაკლები მონაცემთა შესანახი მოწყობილობა) სქემის მხარდაჭერა. სხვაგვარად რომ ვთქვათ, გამანაწილებელი ფაილური სისტემა მასში განთავსებული მონაცემების მრავალ ადგილზე კოპირებისა და მონაცემების დუბლირების გზით იძლევა მონაცემების სიჭარბეს და მაღალი სისწრაფით ხელმისაწვდომობის შესაძლებლობას.

მოწყობილობა, რომელზეც იმპლემენტირებულია გამანაწილებელი ფაილური სისტემა იძლევა მონაცემების მარტივი, სწრაფი წვდომის შესაძლებლობას და ინახავს დიდი მოცულობის მონაცემებს, რომელიც თავისი არსით არაა რელაციური. მაგალითად, როგორიცაა ნაწილობრივ-სტრუქტურირებული ან არასტრუქტურირებული მონაცემები. მონაცემების სიჭარბის გამო პარალელიზმის მართვის მექანიზმი იძლევა მონაცემებზე კითხვა/ჩაწერის ოპერაციების სწრაფი შესრულების შესაძლებლობას, რომელიც წარმოადგენს დიდი მონაცემების velocity მახასიათებელს.

გამანაწილებელი ფაილური სისტემა არაა იდეალური გადაწყვეტა დიდი რაოდენობის და მცირე მოცულობის ფაილების ნაკრებისთვის. ვინაიდან ამ შემთხვევაში წარმოიშობა დიდი რაოდენობით დისკური ძებნის ოპერაცია, რაც ამცირებს მონაცემებზე წვდომის სისწრაფეს. ასევე, მრავალი პატარა მოცულობის ფაილის დამუშავება ახდენს

შესაბამისი დამუშავების მექანიზმის გამოყოფას იმ დროით სანამ შედეგი არ იქნება სინქრონიზირებული კლასტერზე, რაც თავის მხრივ იწვევს დიდ დაყოვნებას.

მიუხედავად ასეთი შეზღუდვისა გამანაწილებელი ფაილური სისტემა ბრწყინვალედ მუშაობს მცირე რაოდენობის, მაგრამ დიდი მოცულობის მქონე ფაილებთან, მიმდევრობითი წვდომის გზით. ამ შემთხვევაში, მრავალი პატარა ფაილი გარკვეული ფორმით ჯგუფდება ერთ ფაილად, რათა მოხდეს შესანახი მოწყობილობის და დამუშავების მექანიზმის ოპტიმიზირება. ეს მიდგომა გამანაწილებელ ფაილურ სისტემას აძლევს შესაძლებლობას აამაღლოს სწრაფქმედება, როდესაც მონაცემები არიან ხელმისაწვდომი ნაკადურ რეჟიმში კითხვა/ჩაწერის ოპერაციების მიმდევრობით შესრულების გზით (ნახ. 7.2).



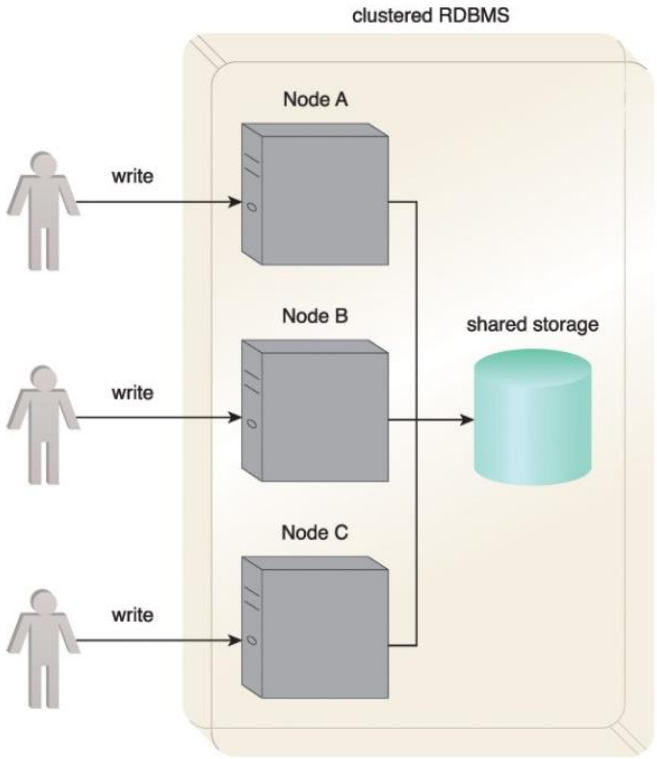
ნახ. 7.2. გამანაწილებელი ფაილურ სისტემა მონაცემების წვდომით ნაკადურ რეჟიმში კითხვა/ჩაწერის ოპერაციის მიმდევრობითი განხორციელებით

გამანაწილებელი ფაილური სისტემა შესაფერისია იმ შემთხვევაშიც, როცა მოითხოვება მონაცემების დიდი ნაკრების შენახვა ან არქივირება. დამატებით ის იძლევა BigData მონაცემების დიდი ხნით იაფი შესანახვის შესაძლებლობას. ამის გაკეთება შესაძლებელია იქიდან გამომდინარე, რომ გამარტივებულია კლასტერზე დიკური მოწყობილობის დამატების შესაძლებლობა მისი გათიშვის გარეშე. უნდა აღინიშნოს, რომ გამანაწილებელი ფაილური სისტემას არ გააჩნია ფაილების ძებნის შესაძლებლობა მათი კონტექსტის მიხედვით.

4.1.2. RDBMS მონაცემთა ბაზები

მცირე ზომის მონაცემების შენახვისთვის საუკეთესო გადაწყვეტას წარმოადგენს რელაციური მონაცემთა ბაზის სისტემა (RDBMS - Relational database management systems), რომელიც იძლევა მონაცემებზე კითხვა/ჩაწერის ოპერაციის განხორციელების შესაძლებლობას შემთხვევითობის პრინციპის დაცვით. RDBMS ბაზები არის ACID-თავსებადი, რაც ნიშნავს, რომ ის შემოიფარგლება მხოლოდ მარტივი კვანძით. ამის გამო, არ ხდება მონაცემების ჭარბი რაოდენობით წარმოდგენა.

მიღებული BigData მონაცემების სწრაფი დამუშავებისვის საჭიროა რელაციური ბაზა იყოს მასშტაბირებადი. RDBMS ბაზებს გააჩნიათ ვერტიკალური მასშტაბირებადობის თვისება, მაგრამ არა ჰორიზონტალური, რომელიც არის საკმარისად ძვირადღირებული და ხელისშემშლელი მასშტაბირებადობისთვის. რაც ამ ტიპის ბაზების გამოყენებას არ აყენებს უპირატესობაში.



ნახ. 7.3. კლასტერილიზირებული რელაციური მონაცემთა ბაზა
გამანაწილებელი მეხსიერების შესანახი მოწყობილობით

უნდა აღინიშნოს, რომ არსებობს მრავალი რელაციური ბაზის სისტემა, რომლის ამუშავებაც შესაძლებელია ცკლასტერზე, როგორიცაა IBM DB2 pureScale, Sybase ASE

Cluster Edition, Oracle Real Application Clusters (RAC) და Microsoft Parallel Data Warehouse (PDW) (Figure 7.3).

4.2. NoSQL მონაცემთა ბაზები

Not-only SQL (NoSQL) წარმოადგენს ტექნოლოგიას, რომელიც გამოიყენება არარელაციურ მონაცემთა ბაზებთან სამუშაოდ. ისინი გამოიჩევიან მაღალი ხარისხით მასშტაბირებადობითა და მტყუნებამედეგობის მიმართ მდგრადობით.

ასეთი მონაცემთა ბაზების მნიშნელოვან მახასიათებლებს წარმოადგენენ:

- Schema-less data model - მონაცემები შეიძლება წამოდგენილი იქნას საწყისი ფორმით;
- Scale out rather than scale up - მეტი მოცულობის შესანახი სივრცის მისაღებად შესაძლებელია ბაზისთვის მეტი რაოდენობის კვანძების დამატება. ამისათვის, აუცილებლობას არ წარმოადგენს არსებული კვანძის ჩანაცვლება ახალი, მაღალი წარმადობის მქონე კვანძით ან მიმდინარე კვანძის გაუმჯობესება;
- Highly available - ეს არის კლასტერზე დაფუძნებული ტექნოლოგია, რომელიც მდგრადია მტყუნებამედეგობის მიმართ
- Lower operational cost - NoSQL მონაცემთა ბაზების უმეტესი შექმნილია ღია კოდით და მისი ლიცენზირება არ მოითხოვს ფინანსურ დანახარჯს. ბაზა შეიძლება იმპლემენტირებული იყოს გამანაწილებელ სისტემაზე.
- Eventual consistency - მონაცემები იკითხება მრავალი კვანძიდან, რომლებიც ჩაწერის ოპერაციის განხორციელების შემდეგ შეიძლება არ იყოს ხელმისაწვდომი. თუმცა საბოლოო ჯამში ყოველი კვანძი რჩება თანმიმდევრულ მდგომარეობაში.
- BASE (არა ACID) თავსებადობა გულისხმობს მაღალი ხარისხით მონაცემთა ხელმისაწვდომობა არ იყოს დარღვეული იმ შემთხვევშიც, როცა შეიძლება დაფიქსირდეს ქსელთან/კვანძთან დაკავშირების პრობლემა.
- API driven data access - მონცემებზე წვდომა მხარდაჭირილია API-ზე დაფუძნებული მოთხოვნებით, მათ შორის RESTful API-ის ჩათვლით, იმის

გათვალისწინებით, რომ ზოგიერთ იმპლემენტაციას შეიძლება გააჩნდეს თავსებადობა SQL-მსგავს მოთხოვნასთან.

- Auto sharding-ი და replication-ი - ჰორიზონტალური მაშტაბულობის მხარდაჭერისთვის და მაღალი ხელმისაწვდომობისთვის NoSQL მონაცემთა ბაზები ავტომატურად იყენებს „sharding“-ს და განმეორების ტექნოლოგიას, სადაც მონაცემთა ნაკრები დაყოფილია ჰორიზონტალურად და შემდეგ კოპირდება მრავალ კვანძზე.
- Distributed query support – NoSQL მონაცემთა ბაზებს გააჩნიათ მხარდაჭერა მრავალ shard-ზე ერთიანი მოთხოვნის გაკეთებისა.
- Polyglot persistence – NoSQL მოცემთა ბაზები არ ზღუდავენ ტრადიციული RDBMS სისტემების გამოყენებას. ორივე მათგანის თანაარსებობა არის შესაძლებელი გვერდიდგვერდ. ეს რათქმაუნდა არის ძალიან კარგი იმ თვალსაზრისით, რომ სისტემაში შეიძლება საჭირო გახდეს დამუშავება როგორც სტრუქტურირებული ისე ნაწილობრივ-სტრუქრურირებული ან არასტრუქტურირებულ მონაცემებისა.
- Aggregate-focused - რელაციური მონაცემთა ბაზებისგან განსხვავებით, რომლებიც არიან ეფექტურები სრულად ნორმალიზირების მონაცემების შემთხვევაში, NoSQL მონაცემთა ბაზა ინახავს არანორმალიზირებულ გაერთიანებულ მონაცემებს, რომელიც გვარიდებს join ოპერაციებს და ღრმა ასახვებს აპლიკაციების ობიექტებსა და მონაცემთა ბაზის ჩანაწერებს შორის.

4.3. NoSQL მონაცემთა ბაზის ტიპები

NoSQL მონაცემთა ბაზები მონაცემების შენახვის პრინციპიდან გამომდინარე შეიძლება დაიყოს შემდეგ 4 ტიპად:

1. key-value (გასაღები-მნიშვნელობა)
2. document (დოკუმენტი)
3. column-family (სვეტების-ჯგუფი)
4. graph (გრაფი)

4.3.1. Key-Value (გასაღები-მნიშვნელობა)

Key-value პრინციპის მიხედვით ბაზაში მონაცემების შენახვა ხორციელდება ჰეშ-ცხრილის მსგავსად key-value წყვილის გამოყენებით (ნახ. 7.11). ცხრილში განთავსებული მნიშვნელობები (value) ხელმისაწვომია გასაღების (key) მეშვეობით. მონაცემთა ბაზაში მნიშვნელობები ინახება გაუმჭვირვალე ფორმით და ამიტომ ინახება როგორც ე.წ. BLOB[2]-ი. შენახული მონაცემები შეიძლება იყოს მიღებული ნებისმიერი გაერთიანებით დაწყებული სენსორული მონაცემებიდან და დამთავრებული ვიდეო გამოსახულებით.

key	value	
631	John Smith, 10.0.30.25, Good customer service	← text
365	10101101010110101011101011010101101010111010110	← image
198	<CustomerId>32195</CustomerId><Total>43.25</Total>	← XML

ნახ. 7.11. key-value პრინციპით მონაცემების შენახვის მაგალითი

მნიშვნელობების ძებნა შეიძლება განხორციელდეს მხოლოდ გასაღების მეშვეობით, ვინაიდან ბაზა არ აქცევს ყურადღებას შენახული აგრეგირებული მონაცემების დეტალებს. ნაწილობრივი განახლება დაუშვებელია. განახლება მოითხოვება ჩასმის ან წაშლის ოპერაციის განხორციელების შემდეგ. ასეთ ბაზებში არაა მხარდაჭერილი ინდექსები და ამის გამო მასში ჩაწერის ოპერაციების სრულდება საკმარისად სწრაფად. რადგანაც ის ეყრდნობა შენახვის მარტივ მოდელს, ამიტომ არის მაღალი მასშტაბირებადობის.

ამ ტიპის მონაცემთა ბაზის გამოყენება მიზანშეწონილია იმ შემთხვევებში, როცა:

- მოითხოვება არასტრუქტურირებული მონაცემების შენახვა;
- მოითხოვება მაღალი სწრაფქმედებით კითხვა/ჩაწერის ოპერაციების შესრულება;
- მნიშვნელობა წარმოადგენს ავტონომიურ არსს და არაა დამოკიდებული სხვა მნიშვნელობაზე;
- მნიშვნელობას გააჩნია შედარებით მარტივი სტრუქტურა ან არის ბინარული;
- მოთხოვნის სტრუქტურა მარტივია, მხოლოდ ჩასმის, ამოღების და წაშლის ოპერაციებისთვის;

[2] **BLOB** (*Binary Large Object* - დიდი ოროობითი ობიექტი) ოროობითი მნიშვნელობების მასივი.

- შენახული მნიშვნელობების მანიპულირება ხდება გამოყენებით დონეზე.

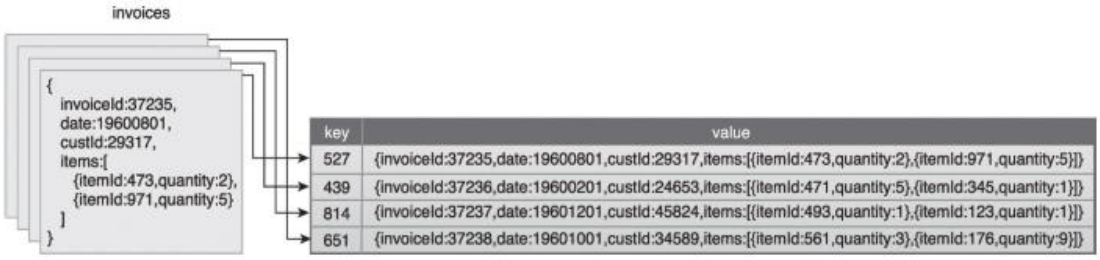
ხოლო მათი გამოყენება არაა მიზანშეწონილი იმ შემთხვევებში, როცა:

- აპლიკაცია მნიშვნელობის ჩანაწერის ატრიბუტების მიხედვით ითხოვს ძებნის ან გაფილტვრის ოპერაციის განხორციელებას;
- სხვადასხვა key-value არსებს შორის არსებობს კავშირი;
- Key-ების მნიშვნელობების ჯგუფის განახლება საჭიროა მარტივ ტრანზაქციაში;
- მრავალი key მოითხოვება მარტივი ოპერაციით მანიპულირებისას;
- მოითხოვება სხვადასხვა მნიშვნელობებისთვის მიმდევრობითი სქემა
- მოითხოვება მნიშვნელობის ინდივიდუალური ატრიბუტების განახლება

Key-value ტიპის მონაცემთა ბაზის ელემენტებს შეიცავს შემდეგი მონაცემთა ბაზები: Riak, Redis და Amazon Dynamo DB.

4.3.2. Document

დოკუმენტების შენახვის მონაცემთა ბაზა მონაცემებს ინახავს როგორც წყვილი „key-value“ (გასაღები - მნიშვნელობა), თუმცა მისგან განსხვავებით value არის დოკუმენტი, რომელიც შეიძლება მოთხოვნილი იქნას მონაცემთა ბაზით. ასეთ დოკუმენტებს შეიძლება გააჩნდეთ მასშტაბური, დალაგებული სტრუქტურა, როგორც ინვოისს (ნახ. 7.12). ტექსტი შეიძლება კოდირებული იყოს ტექსტზე ორიენტირებული კოდირების სქემით (როგორიცაა XML ან JSON), ან იყენებდეს ბინარული კოდირების სქემას (როგორიცაა BSON - Binary JSON)



ნახ. 7.12. JSON მონაცემების აღწერა დოკუმენტების შენახვის მონაცემთა ბაზაში

Key-value ბაზის მსგავსად, მრავალი ეს ბაზა ინახავს მონაცემების კოლექციას ან ბლოკს, რომელიც შეიძლება ორგანიზებული იქნას key-value წყვილის სახით. დოკუმენტების ბაზასა და key-value ბაზას შორის განსხვავება მდგომარეობს შემდეგში:

- დოკუმენტის ბაზა ინფორმირებულია მონაცემის მნიშვნელობაზე;
- შენახული მნიშვნელობა აღწერს თავის თავს. სქემა შეიძლება მიღებული იქნას მნიშვნელობის სტრუქტურიდან ან დოკუმენტისთვის სქემაზე მიმთითებელი განთავსებულია მნიშვნელობაში;
- გამოყენებული ოპერაცია შეიძლება უთითებდეს აგრეგირებული მონაცემის შიგნით გარკვეულ ველზე;
- გამოყენებული ოპერაციით შეიძლება მიღებული იქნას აგრეგირებული მნიშვნელობის ნაწილი;
- შესაძლებელია როგორც უშუალოდ მონაცემების ისე, აგრეგირებული მონაცემების ნაწილობრივი განახლება;
- მხარდაჭერილია ინდექსები, რომლებიც იძლევა ძებნის ოპერაციების განხორციელების საშუალებას.

ყოველ დოკუმენტს შეიძლება გააჩნდეს განსხვავებული სქემა. შესაბამისად, შესაძლებელია სხვადასხვა სახის დოკუმენტის შენახვა ერთიდაიმავე კოლექციაში. მონაცემთა ბაზაში დოკუმენტის განთავსების შემდეგ შესაძლებელია მისთვის დამატებითი ველის მიერთება. ამრიგად, იძლევა უფრო მოქნილ სქემის მხარდაჭერას.

უნდა აღინიშნოს, რომ დოკუმენტის შენახვის მონაცემთა ბაზა არაა შეზღუდული ისეთი დოკუმენტების შენახვაზე როგორიცაა XML ფაილი, უფრო მეტიც ის იძლევა ნებისმიერი გაერთიანების შენახვის შესაძლებლობას, რომელიც შეიძლება შედგება ნებისმიერი რაოდენობის ველების კოლექციისგან ბრტყელი სტრუქტურით ან ჩალაგებული სქემისგან. ნახ. 7.12-ზე ნაჩვენებია NoSQL მონაცემთა ბაზაში შენახული JSON დოკუმენტი.

დოკუმენტის შენახვის მონაცემთა ბაზის გამოყენება მიზანშეწონილია იმ შემთხვევებში, როცა:

- მოითხოვება დოკუმენტზე-ორიენტირებული ნაწილობრივ სტრუქტურირებული ბრტყელი სტრუქტურის მქონე ან ჩადგმული სქემის მქონე დოკუმენტის შენახვა;
- მოითხოვება სქემის ევოლუცია ვინაიდან დოკუმენტის სტრუქტურა უცნობია ან შეიძლება შეიცვალოს;

- აპლიკაცია ითხოვს აგრეგირებული მონაცემების ნაწილობრივ განახლებას დოკუმენტის სახით;
- მოითხოვება ძებნის ოპერაციების განხორციელება დოკუმენტის სხვადასხვა ველებზე;
- სერიალიზირებული ობიექტის სახით უნდა იქნას შენახული დომეინ ობიექტი;
- მოთხოვნის სტრუქტურა იწვევს ჩამატების, ამოღების, განახლების და წაშლის ოპერაციების განხორციელებას.

ხოლო მისი გამოყენება არაა მიზანშეწონილი იმ შემთხვევებში, როცა:

- მოითხოვება მრავალი დოკუმენტის განახლება, როგორც მარტივი ტრანზაქციის ნაწილი;
- მოითხოვება ისეთი ოპერაციების შესრულება, რომელიც გულისხმობს მრავალი დოკუმენტის გაერთიანებას ან ინახება ნორმალიზებული მონაცემების სახით;
- შესანახი მონაცემები არაა საკუთარი თავის აღმწერი ან არ გააჩნია სქემაზე მიმთითებელი
- ბინარული მონაცემების შენახვა მოითხოვება

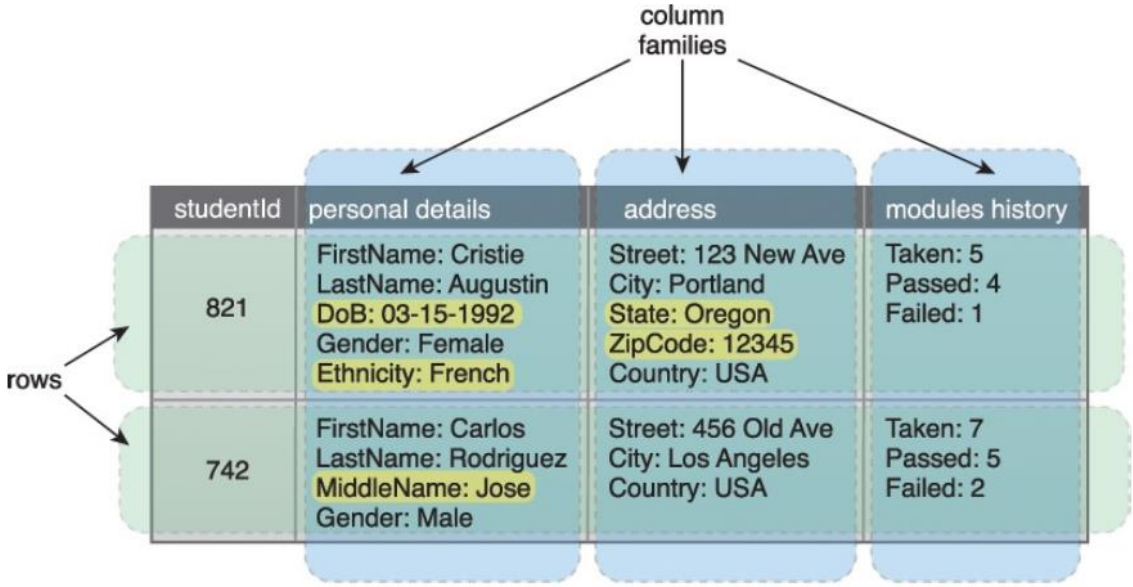
Document ტიპის მონაცემთა ბაზის ელემენტებს შეიცავს შემდეგი მონაცემთა ბაზები: MongoDB, CouchDB და Terrastore.

4.3.3. Column-Family

Column-family ტიპის მონაცემთა ბაზა წააგავს ტრადიციულ მონაცემთა ბაზას, მაგრამ სტრიქონზე აკავშირებენ ჩანაწერის სვეტებს კოლექციებად (ნახ. 7.13). ყოველი სვეტი ჩანაწერისთვის შეიძლება შეიცავდეს განსხვავებული რაოდენობის სვეტების ერთობლიობას, რომელთა განახლება შესაძლებელია მარტივად. ყოველისტრიქონი შედგება მრავალი სვეტების კოლექციიდან (column-families) და განსხვავებული სვეტებისგან. ყოველი სტრიქონის იდენტიფიცირება ხდება გასაღების მნიშვნელობით.

ამ ტიპის მონაცემთა ბაზა იძლევა მონაცემებზე სწრაფი წვდომის შესაძლებლობას კითხვა/ჩაწერის ოპერაციების შემთხვევითი პრინციპით განხორციელების წესის გამოყენებით. ყოველი ფიზიკური ჩანაწერისთვის ინახავს განსხვავებული სვეტების

კოლექციას, რომელიც ბაზაში ძებნის ოპერაციის განხორციელებისას იძლევა სწრაფი გამოხმაურების შესაძლებლობას.



ნახ. 7.13. Column-family მონაცემთა ბაზის მაგალითი

Column-family მონაცემთა ბაზის გამოყენება მიზანშეწოლილია იმ შემთხვევებში, როცა:

- რეალურ დროში მოითხოვება კითხვა/ჩაწერის ოპერაციების განხორციელება შემთხვევითობის პრინციპის დაცვით და შესანახ მონაცემებს გააჩნია გარკვეული სტრუქტურა;
- მონაცემებს გააჩნია ცხრილური სტრუქტურა, ყოველი სტრიქონი შეიცავს დიდი რაოდენობით ველებს და არსებობს ჩადგმული ურთიერთდაკავშირებული მონაცემების ჯგუფები;
- მოითხოვება სქემის ევოლუციის თავსებადობა, ვინაიდან column families შეუძლია დამატების ან წაშლის ოპერაციების განხორციელება სისტემის წარმადობაზე ზემოქმედების გარეშე;
- გარკვეული ველები ხშირად არიან წვდომადი ერთდროულად და ძებნის განხორციელება საჭიროა ველების მნიშვნელობის გამოყენებით;
- მოთხოვნის სტრუქტურა იწვევს ჩასმის, ამოღების, განახლების და წაშლის ოპერაციის შესრულებას.

ხოლო მისი გამოყენება არაა მიზანშეწოლილი იმ შემთხვევებში, როცა:

- მოითხოვება რელაციურ მონაცემებზე წვდომა;
- მოითხოვება ACID ტრანზაქციისთან თავსებადობა;
- მოითხოვება ბინარული მონაცემების შენახვა
- მოითხოვება SQL-თავსებადი მოთხოვნის შესრულება

Column-family ტიპის მონაცემთა ბაზის ელემენტებს შეიცავს შემდეგი მონაცემთა ბაზები: Cassandra, HBase და Amazon SimpleDB.

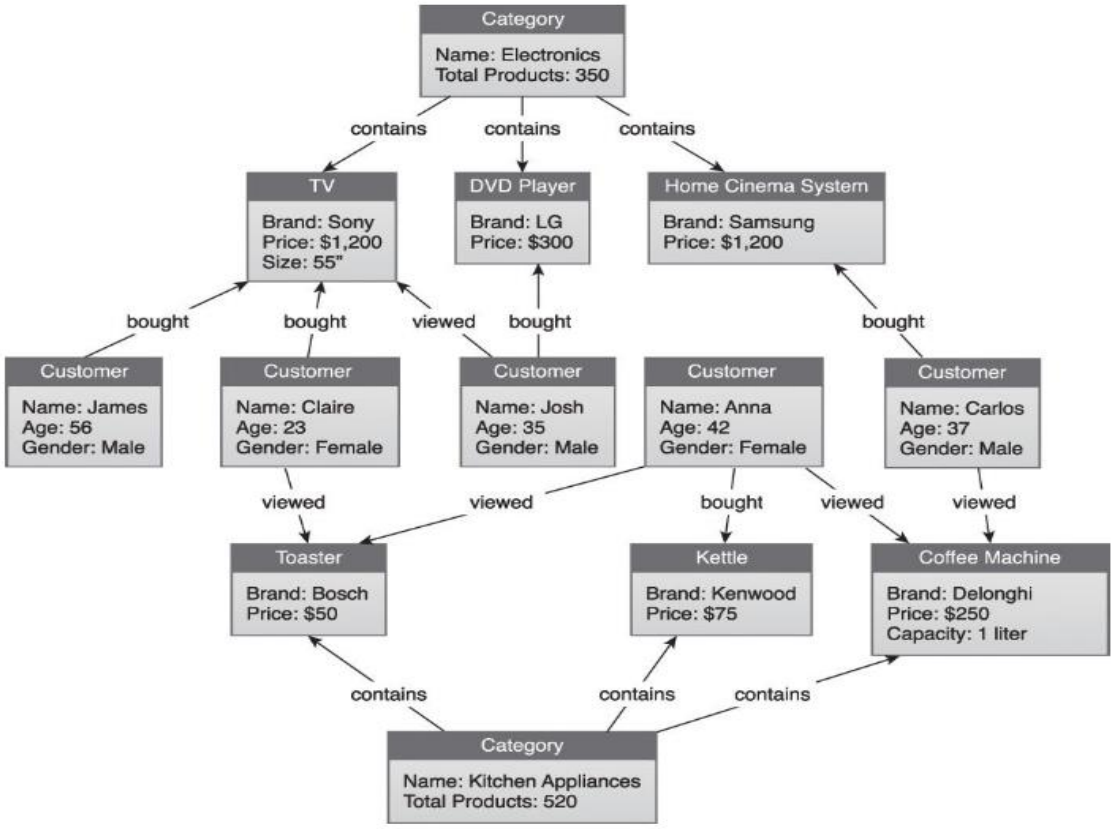
4.3.4. Graph

გრაფის მონაცემთა ბაზა გამოიყენება დაკავშირებული არსის შესანახად. სხვა NoSQL მონაცემთა ბაზისგან განსხვავებით, სადაც ყურადღება გადატანილია არსის სტრუქტურაზე, გრაფის მონაცემთა ბაზაში ყურადღება გადატანილია არსებს შორის კავშირებზე (7.14). არსები ინახება როგორც კვანძები[3] (მათ ასევე უწოდებენ წვეროებს), მაშინ როცა არსებს შორის კავშირები შენახულია როგორც წიბოები. RDBMS ენაში ყოველი კვანძი შეიძლება წარმოადგენდეს მარტივ სტრიქონს, მაშინ როცა წიბო აღნიშნავს join ოპერაციას.

კვანძები შეიძლება იყენებდნენ ერთზე მეტი ტიპის კავშირებს მათ შორის მრავალი წიბოს გამოყენებით. ყოველ კვანძს შეიძლება გააჩნდეს ატრიბუტის მონაცემები წყვილი key-value, როგორც მყიდველს: ID, name და age ატრიბუტები.

ყოველს წიბოს შეიძლება გააჩნდეს საკუთარი წყვილი key-value სახის ატრიბუტის მონაცემები, რომელიც შეიძლება გამოყენებული იქნას მოთხოვნის შედეგის გაფილტვრისას. მრავალი წიბოს ქონა წააგავს მრავალ ე.წ. „foreign keys“ RDBMS-ში. მიუხედავად ამისა ყველა კვანძს არ მოეთხოვება იმავე წიბოების ქონა. მოთხოვნები კვანძის ან/და წიბოს ატრიბუტების გამოყენებით იძლევიან დაკავშირებული კვანძების მოძებნის შესაძლებლობას (რასაც კვანძების გადაკვეთას უწოდებენ). წიბო შეიძლება იყოს ერთ- ან ორ-მიმართულებიანი კვანძების გადაკვეთის მიღების შემთხვევაში. ჩვეულებრივ, გრაფის ტიპის ბაზები უზრუნველყოფენ თანმიმდევრულობას ACID თავსებადობით.

[3] ტერმინი არ მოდის წინააღმდეგობაში კლასტერის კვანძთან



ნახ. 7.14. გრაფის მონაცემთა ბაზა ინახავს არსებს და მათ კავშირებს

გრაფის მონაცემთა ბაზიდან მონაცემთა გამოყენებისას წარმატების ხარისხი დამოკიდებულია კვანძების რაოდენობაზე და მათ შორის წიბოების ტიპებზე. რაც უფრო დიდი რაოდენობის კვანძი და მეტი განსხვავებული ტიპის წიბო იქნება გამოყენებული მით უფრო მეტი განსხვავებული მოთხოვნის დამუშავება იქნება შესაძლებელი. როგორც შედეგი მნიშვნელოვანია აღინიშნოს, რომ ეს არის მნიშვნელოვანი კვანძებს შორის ყველა შესაძლო კავშირების გამოსავლენად. ეს არამხოლოდ ჭეშმარიტია აღწერილი სიტუაციისთვის, არამედ მონაცემთა ანალიზის შემთხვევაშიც.

გრაფის მონაცემთა ბაზა ილევა ახალი ტიპის კვანძის დამატების შესაძლებლობას ბაზაში ცვლილების გაკეთების აუცილებლობის გარეშე, ეს რათქმაუნდა განსაზღვრავს ახალ კავშირს კვანძებს შორის, როგორც კავშირის ახალი ტიპი ან კვანძების წარმოდგენა მონაცემთა ბაზაში.

გრაფის შესანახი მოწყობილობის გამოყენება მიზანშეწონილია იმ შემთხვევებში როცა

- საჭიროა დაკავშირებული არსების შენახვა;
- მოთხოვნილი არსი ეყრდნობა ყველასთან კავშირის ხასიათზე ვიდრე მის ატრიბუტებთან კავშირს;
- საჭიროა დაკავშირებული არსების მოძებნა;
- საჭიროა კვანძების გადაკვეთის ტერმინებში არსებს შორის მანძილის მოძებნა;

ხოლო მისი გამოყენება არაა მიზანშეწონილია იმ შემთხვევებში, როცა

- საჭიროა დიდი რაოდენობით კვანძების ატრიბუტების ან წიბოს ატრიბუტების განახლებას, ვინაიდან ეს საჭიროებს კვანძის ან წიბოს მოძებნას, რომელის არის მუდმივი ოპერაცია კვანძების გადაკვეთასთან მიმართებაში.
- კვანძებს გააჩნიათ დიდი რაოდენობის ატრიბუტები ან ჩალაგებული მონაცემები;
- ბინარული შესანახი მოწყობილობა მოითხოვება.

გრაფის შესანახი მოწყობილობის მაგალითს შეიცავს Neo4J, Infinite Graph და OrientDB ბაზები.

4.4. NewSQL Databases

NoSQL მონაცემთა ბაზებს გააჩნიათ მაღალი მასშტაბურობა, ხელმისაწვდომობა, მტყუნებამედეგობის მიმართ მდგრადპბა და კითხვა/ჩაწერის ოპერაციების სრაფად განხორციელების შესაძლებლობა. თუმცა მათ არ გააჩნიათ მხარდაჭერა ისეთივე ტრანზაქციების შესრულებისა და თავსებადობისა როგორიც გააჩნია ACID თავსებად RDBMS ბაზებს.

NewSQL მონაცემთა ბაზები აერთიანებენ RDBMS-ის ACID თვისებას NoSQL მონაცემთა ბაზის მასშტაბურობისა და მტყუნებამედეგობის თვისებას. ჩვეულებრივ, NewSQL მონაცემთა ბაზებს გააჩნიათ SQL-თავსებადი სინტაქსი მონაცემების განსაზღვრისა და მათზე მანიპულირების ოპერაციებისთვის განსახორციელებლად და ყოველთვის იყენებენ ლოგიკურ რელაციურ მონაცემთა მოდელს მონაცემთა შესანახად.

NewSQL მონაცემთა ბაზები შესაძლებელია გამოყენებული იქნას დიდი მოცულობის ტრანზაქციების საჭიროების მქონე OLTP[4] სისტემების შესაქმნელად,

[4] Online Transaction Processing

მაგალითად, საბანკო სისტემის. ისინი შეიძლება ასევე გამოყენებული იქნან რეალური დროში ანალიზისთვის, მაგალითად ოპერაციული ანალიზისთვის, როგორც ზოგიერთი იმპლემენტაცია იძლევა მეხსიერების შესანახ მოწყობილობის ეფექტური გამოყენების საშუალებას.

როგორც NoSQL შესანახ მოწყობილობასთან შედარებამ გვიჩვენა NewSQL შესანახ მოწყობილობა წარმოგვიდგენს მარტივ გადასვლას ტრადიციული RDBMS სისტემიდან მაღალი მასშტაბურობის მონაცემთა ბაზებში რომელსაც გააჩნია SQL-ის მხარდაჭერა.

NewSQL მონაცემთა ბაზების ელემენტებს შეიცავს VoltDB, NuoDB და InnoDB.

4.5. მეხსიერებაში შესანახი მოწყობილობა

რადგანაც დღეს უკვე პრობლემას არ წარმოადგენს დიდი მოცულობის ფიზიკური მეხსიერების ქონა სისტემაში და ის თავისი სწრაფქმედებისთ აღემატება როგორც HDD ისე, SSD (solid state hard drives) მოწყობილობის სწრაფქმედებას კითხვა/ჩაწერის ოპერაციის განხორციელების საჭიროებისას, ამიტომ მკვლევრებამ დაიწყეს ფიქრი დასამუშავებელი მონაცემების ფიზიკურ მეხსიერებაში (შემდეგში უბრალოდ გამოვიყენებთ ტერმინს მეხსიერება) განთავსების შესაბამისი ტექნოლოგიის შემუშავებაზე.

მეხსიერებაში მონაცემების შენახვის შესალებლობა გვერდს უვლის დისკური I/O ოპერაციების განხორციელებისას დაყოვნებებს, რომელსაც იწვევს ფიზიკურ მეხსიერებასა და შესანახ მოწყობილობას შორის მონაცემთა გადაადგილების შესაბამისი ოპერაციები. შესაბამისად, მცირდება მონაცემების კითხვა/ჩაწერის ოპერაციების რაოდენობა, რაც თავის მხრივ აჩქარებს მონაცემების დამუშავების პროცესს.

მეხსიერების მონაცემთა შესანახად გამოყენების შემთხვევაში მისი მოცულობის გაზრდა შესაძლებელია ჰორიზონტალურად განთავსებულ კლასტერულ სისტემებზე.

კლასტერზე დაფუძნებული მეხსიერება იძლევა დიდი მოცულობის მონაცემების შენახვის შესაძლებლობას, BigData datasets-ის ჩათვლით, რომელზეც წვდომა დისკურ მოწყობილობასთან მიმართებაში შეიძლება განხორციელდეს წარმოუდგენლად სწრაფად. რაც მნიშვნელოვნად ამცირებს BigData analytics-ის შესრულების დროს, შესაბამისი იძლევა რეალურ დროში BigData analytics-ის საშუალებას.

ნახ. 7.16. მონაცემებზე წვდომა ფიზიკური მეხსიერებიდან
და შესანახი მოწყობილობიდან

ნახ. 7.16-ზე შედარებულია ფიზიკურ მეხსირერებაში განთავსებულ მონაცემზე და შესანახ მოწყობილობაზე განთავსებულ მონაცემზე წვდომის დრო.

მეხსიერებაში BigData მონაცემების შენახვა შესაძლებელია კლასტერულ სისტემაზე, მაღალი ხელმისაწვდომობისა და სიჭარბის უზრუნველყოფით. შესაბამისად, ჰორიზონტალური მასშტაბურობა შეიძლება მიღწეული იქნას მარტივად კვანძის ან მეხსიერების დამატებით. დისკური შესანახ მოწყობილობასთან მიმართებაში მეხსიერებაში განთავსებული შესანახი მოწყობილობა არის უფრო ძვირადღირებული ვინაიდან ძირითადი მეხსიერება მყარ დისკთან მიმართებაში უფრო ძვირი ღირებულებისაა.

მიუხედავად იმისა, რომ 64-თანრიგა მანქანა იძლევა 16 ეგზაბიტი (2^{64}) მეხსიერების ქონის შესაძლებლობას, არსებობს ფიზიკური ლიმიტი მანქანაზე, როგორიცაა მეხსიერების სლოტების რაოდენობა. რელურად გაცილებით ნაკლები მეხსიერების მეხსიერების გამოყენება ხდება. გამოთვლებისთვის ეს არ არის მხოლოდ მეხსიერების მოცულობის დამატება, მაგრამ ასევე შესაძლებელია საჭირო რაოდენობის კვანძების დამატება, რომელიც საჭიროა მეხსიერების საჭირო ლიმიტის მისაღებად. ასეთი ზრდა მნიშვნელოვნად გაზრდის მოხსიერების მოწყობილობის ღირებულებას.

გარდა იმისა, რომ დიდი მოცულობის ფიზიკური მეხსიერების ქონა „ძვირადღირებული სიამოვნებაა“, მას ასევე გააჩნია სხვა შეზღუდვა, ის არ იძლევა

მონაცემების დიდი ხნით შენახვის შესაძლებლობას. შესაბამისად, მხოლოდ აქტურლურრი ან ახალი მონაცემების განთავსება უნდა ხდებოდეს მეხსიერებაში.

მეხსიერებაში მონაცემების განსათავსებლად შესაძლებელია გამოყენებული იქნას ე.წ. schema-less ან schema-aware სქემა.

Schema-less პრინციპის გამოყენების შემთხვევაში შეიძლებელია მონაცემებზე წვდომისათვის key-value წყვილზ დაფუძნებული სტრუქტურის მიღება.

ასეთი მეხსიერების გამოყენება მიზანშეწონილია იმ შემთხვევებში, როცა:

- მონაცემების მიღება ხორციელდება სწრაფად და მოითხოვება სწრაფი ანალიზი ან ნაკადის პროცესირება;
- მოითხოვება ინტერაქტიულ მოთხოვნათა პროცესირება და რეალურ დროში მონაცემების ვიზუალიზაცია what-if ანალიზისა და drill-down ოპერაციების ჩათვლით;
- ერთიდაიგივე მონაცემთა ნაკრები მოითხოვება მონაცემთა დამუშავების მრავალი ამოცანისთვის;
- მონაცემთა დამუშავება იწვევს იტერაციულ წვდომას ერთიდაიმავე მონაცემთა ნაკრებზე, როგორიცაა გრაფზე ორიენტირებული ალგორითმები;

ხოლო მისი გამოყენება არაა მიზანშეწონილია იმ შემთხვევებში, როცა:

- მონაცემების პროცესირება გულისხმობს პაკეტურ დამუშავებას;
- ძალიან დიდი მოცულობის მონაცემების განთავსება უნდა მოხდეს მეხსიერებაში დიდი დროითი შუალედით მონაცემების სიღრმისეული ანალიზისთვის;
- Datasets არის ექსტრემალურად დიდი და მისი განთავსება ასებულ მეხსიერებაში არის შეუძლებელი;

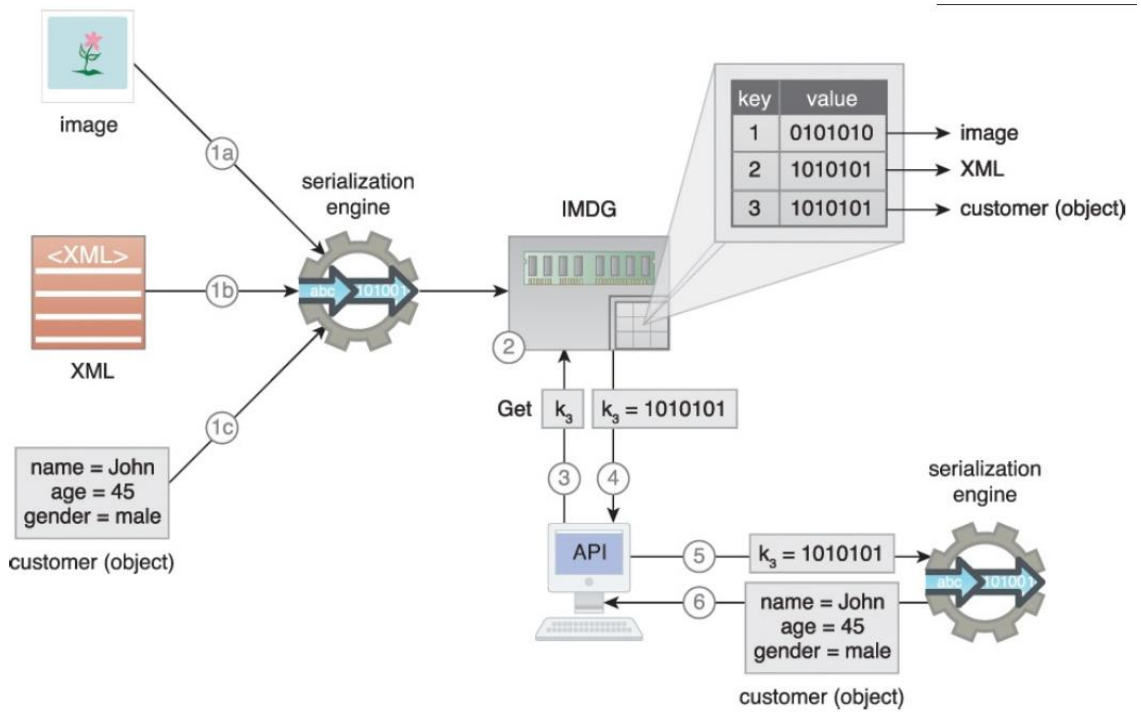
მეხსიერებაში განთავსებული შესანახი მოწყობილობის იმპლემენტირება შესაძლებელია ორი მეთოდით:

- In-Memory Data Grid (IMDG)
- In-Memory Database (IMDB)

მიუხედავად, იმისა რომ ორივე ტექნოლოგია იყენებს მეხსიერებას როგორც მონაცემთა შენახვის გარემო, რითაც განსხვავდება ისინი ერთმანეთისაგან არის მეხსიერებაში მონაცემთა შენახვის წესი.

4.5.1. In-Memory Data Grids

IMDG-ი შესანახი მოწყობილობა მონაცემებს ინახავს როგორც წყვილი key-value მრავალ კვანძზე, სადაც key და value შეიძლება იყოს ნებისმიერი ბიზნეს ობიექტი ან აპლიკაციის მონაცემები სერიალიზირებული ფორმით. ამას გააჩნია schema-less მონაცემთა შესანახი მოწყობილობის თავსებადობა ნაწილობრივ-სტრუქტურირებული/ არასტრუქტურირებული მონაცემებთან. მონაცემებზე წვდომა ხორციელდება API-ს გამოყენებით (ნახ. 7.18).



ნახ. 7.18. IMDG-ი შესანახი მოწყობილობა

4.5.2. In-Memory Database

ეს მეთოდი მეხსიერებაში მონაცემების შესანახად გულისხმობს მონაცემთა ბაზების ტექნოლოგიის გამოყენებას და ზრდის RAM სწრაფქმედებას რათა დაძლეული იქნას დისკური მოწყოპილობის მიერ გამოწვეული დაყოვნება. ნახ. 7.25-ზე ნაჩვენებია

დასკვნა

BigData როგორც ასეთი ტექნოლოგიის გამოყენება როგორც ვხედავთ დღევანდელ მსოფლიოში ყოველ წუთს და წამს უფრო საჭირო და აუცილებელი ხდება. ჩვენ ვხედავთ რომ ყოველ წელს ვირტუალური სამყარო უფრო და უფრო იხვეწება და ისეთ მასშტაბებს აღწევს, რომ შეუძლებელი ხდება ამხელა მასშტაბების ძველი ტექნოლოგიებით კონტროლი. BigData ტექნოლოგია არის ერთადერთი რომელსაც თანამედროვე ინტერნეტ სამყაროს მთელი ინფორმაციის დამუშავება და კონტროლი ძალუძს. ასე რომ მის განვითარებაზე ტყუილად არ იხარჯება მილიონობით თანხები.

შეგვიძლია ამ ყველა ინფორმაციაზე დაყრდნობით ვთქვათ, რომ უახლოეს მომავალში არა მხოლოდ წარსულ ინფორმაციაზე გვექნება წვდომა, არამედ მომავლის წინასწარმეტყველებასაც შევძლებთ, ეს ტენდენცია არის ის რომელსაც უნდა დაეყრდნოს მთელი მსოფლიო და ინტერნეტ ინდუსტრია. ჩვენ ნამდვილად შევძლებთ მომავლის განჭვრეტას, ეს იქნება ეკონომიკური კრახი, პოლიტიკური შეხედულებები, თუ ჯანმრთელობა.

გამოყენებული ლიტერატურა

1. https://en.wikipedia.org/wiki/Big_data;
2. M Hilbert, P López, The world's technological capacity to store, communicate, and compute information, science, 2011 (http://www.martinhilbert.net/WorldInfoCapacity.html/)
3. *J.Hellerstein, "Parallel Programming in the Age of BigData", 2008*
4. *T. Segaran, J.Hammerbacher (). Beautiful Data: The Stories Behind Elegant Data Solutions. O'Reilly Media. p. 257. ISBN 978-0-596-15711-1. 2009.*