

საქართველოს ტექნიკური უნივერსიტეტი
ხელნაწერის უფლებით

გიორგი კვიციანი

მაღალი წვდომადობის მონაცემთა საცავის დაპროექტება გადაუდებელი
დახმარების ოპერატიული მართვის ცენტრისთვის („112“)

სადოქტორო პროგრამა „ინფორმატიკა“
შიფრი 0401

დოქტორის აკადემიური ხარისხის მოსაპოვებლად
წარდგენილი დისერტაციის

ა ვ ტ ო რ ე ფ ე რ ა ტ ი

თბილისი
2018 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკურ უნივერსიტეტში
ინფორმატიკისა და მართვის სისტემების ფაკულტეტი
მართვის ავტომატიზებული სისტემების
(პროგრამული ინჟინერია) დეპარტამენტი

ხელმძღვანელი: პროფ. გია სურგულაძე

რეცენზენტები: -----

დაცვა შედგება ----- წლის "-----" -----, ----- საათზე

საქართველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მართვის
სისტემების ფაკულტეტის სადისერტაციო საბჭოს კოლეგიის სხდომაზე,
კორპუსი -----, აუდიტორია -----

მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ს ბიბლიოთეკაში,
ხოლო ავტორეფერატისა - ფაკულტეტის ვებგვერდზე

სადისერტაციო საბჭოს მდივანი პროფ. თინათინ კაიშაური

ნაშრომის ზოგადი დახასიათება

თემის აქტუალურობა. სადოქტორო ნაშრომის “მაღალი წვდომადობის მონაცემთა საცავის დაპროექტება გადაუდებელი დახმარების ოპერატიული მართვის ცენტრისთვის („112“)“ ფარგლებში ჩატარებულ იქნა სიღრმისეული კვლევა, მოძიებულ და გაანალიზებულ იქნა საკვლევ პრობლემატიკასთან დაკავშირებული სამეცნიერო ლიტერატურა და არსებული გადაწყვეტილებები. დაგროვილი გამოცდილება გამოყენებულ იქნა გადაუდებელი დახმარების ოპერატიული მართვის ცენტრისთვის მონაცემთა საცავის დაპროექტებასა და რეალური პრობლემების გადაწყვეტის შეთავაზებაში. „112“-ში უკვე წარმატებით ვიყენებთ MongoDB მონაცემთა ბაზას დავალებათა მართვის სისტემისთვის. ამასთან ერთად, მიმდინარე პროექტად გვაქვს მონაცემების სხვადასხვა სერვერებიდან MongoDB-ში თავმოყრა, რაც ბევრად უფრო ააჩქარებს და გაამარტივებს მონაცემთა ანალიზსა და სტატისტიკურ დამუშავებას.

სერვერული უზრუნველყოფის არქიტექტურის შემუშავება მნიშვნელოვანი ეტაპია, მაგრამ, ცხადია, საინტერესოა, რა სახით და რა მონაცემებს შევინახავთ მასში.

სადისერტაციო ნაშრომის ექსპერიმენტული ნაწილი ეძღვნება გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ „112“-ისთვის ისეთი პრობლემური საკითხის გაუმჯობესებას, როგორც არის სასწრაფო სამედიცინო მომსახურების ბრიგადების დისპეტჩერიზაცია.

მეცნიერული სიახლე. ნაშრომში წარმოდგენილია შემდეგი სიახლეები:

- მონაცემთა რელაციური და არარელაციური ბაზების მახასიათებლების შედარების საფუძველზე შემუშავებულია მაღალი წვდომადობის მქონე მონაცემთა საცავის დაპროექტების მეთოდოლოგია.
- პროექტის განსახორციელებლად გამოყენებულია ჰორიზონტალური ზრდის მოდელი, რაც ითვლება თანამედროვე და ინოვაციურ მიდგომად. აღნიშნული მოდელი გულისხმობს ინფორმაციის

შენახვა-დამუშავებისთვის სერვერების გაერთიანებით მიღებული ერთიანი რესურსის გამოყენებას. აღნიშნული მოდელის უპირატესობა ის არის, რომ საჭიროებისამებრ, მარტივად არის შესაძლებელი რესურსის დამატება. ასევე, აღნიშნული მოდელის გამოყენებისას, ტექნიკის დაზიანებისგან გამოწვეული ხარვეზები არ არის კრიტიკული და ადვილად გამოსწორებადია.

- პროექტში ერთ-ერთი ყველაზე ინოვაციური მიდგომა არის შარდინგისა და რეპლიკაციის ტექნოლოგიების გამოყენება მაქსიმალური სტაბილურობისა და წარმადობის მისაღწევად.
- გადაუდებელი სამედიცინო სამსახურის გამოწვევების გაანალიზების შედეგად შემუშავდა ინოვაციური მიდგომები არსებული პრობლემების გადასაჭრელად: ხელოვნური ნეირონული ქსელის გამოყენების შესაძლებლობის გამოვლენა სასწრაფო დახმარების ბრიგადის შერჩევის ამოცანისთვის და ალგორითმების სისტემის შექმნა სასწრაფო დახმარების გამოძახებების პრიორიტეტების განსაზღვრის ამოცანისთვის.

სამუშაოს მიზანი. თანამედროვე ტექნოლოგიების სამყაროში ინფორმაციის უდიდეს ნაწილს ადამიანების ნაცვლად კომპიუტერული ტექნიკა ქმნის. ინფორმაციის ნაკადი იმდენად სწრაფად იზრდება, რომ საჭირო ხდება ინფორმაციის დამუშავების თანამედროვე მეთოდების მოძიება. დისერტაციის მიზანია ინფორმაციის შენახვა-დამუშავების ახალი ტექნოლოგიების გამოკვლევა და შესაბამისი სერვერული და პროგრამული უზრუნველყოფის მეთოდოლოგიის ჩამოყალიბება.

სერვერული უზრუნველყოფის გამართვასთან ერთად სადისერტაციო ნაშრომის მიზნად დაისახა „112“-ის დისპეტჩერებისთვის მუშაობის გამარტივება და დისპეტჩერიზაციის პროცესის გაუმჯობესება, რისთვისაც გამოიკვეთა ორი რეალური პრობლემა:

- **სასწრაფო დახმარების ბრიგადის შერჩევის ამოცანა** - სასწრაფოს ბრიგადის გათავისუფლებისა და ახალ მისამართზე

მისვლის სავარაუდო დროის გამოთვლა. მაგალითად, ზარის ინიციატორთან ტერიტორიულად უახლოესი თავისუფალი ბრიგადა საცობის გათვალისწინებით 1 საათის სავალზეა, სხვა შემთხვევაზე გასული ბრიგადა კი შეიძლება ინიციატორის მეზობლად იყოს და 5 წუთში გათავისუფლდეს. ცხადია, რომ დისპეტჩერიზაციის ოპტიმალური მართვით დროც დაიზოგება და ფინანსური თუ ადამიანური რესურსიც.

• **სასწრაფო დახმარების გამომახებების პრიორიტეტების განსაზღვრის ამოცანა** - ამჟამად „112“-ში პრიორიტეტები ხისტად არის განსაზღვრული ინციდენტის ტიპების კატეგორიების და არა თითოეული შემთხვევის კონკრეტული მახასიათებლების სიმძიმის მიხედვით. პიკის საათებში „112“-ში შემოდის ერთი და იმავე პრიორიტეტის მქონე ათობით სამედიცინო საქმე, რომელთა დისპეტჩერიზაციის რიგითობის განსასაზღვრად დისპეტჩერს უწევს მათი სათითაოდ შემოწმება და შედარება, რაც დიდ ძალისხმევას მოითხოვს მაშინ, როდესაც წამები გადაძვწვეტია. მაგალითად, გრიპის სეზონური ეპიდემიის დროს გამომახებები არსებულ ბრიგადებთან შედარებით ბევრად მეტია და ხშირად ერთი და იმავე პრიორიტეტის („დაბალი“, „საშუალო“ ან „მაღალი“) მქონე ათობით „ჩამოკიდებული“ საქმე ელოდება თავისუფალ ბრიგადას.

დასახული მიზნის მისაღწევად აუცილებელია შემდეგი ძირითადი ამოცანების გადაწყვეტა:

- მონაცემთა რელაციური და არარელაციური ბაზების მახასიათებლების შედარების საფუძველზე უნდა განისაზღვროს იმ ამოცანათა კლასები, რომლებისთვისაც ეფექტიანია თითოეული მათგანის გამოყენება.
- ანალიზის საფუძველზე უნდა შეირჩეს მონაცემთა ბაზა გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ „112“-ისთვის.

- უნდა შემუშავდეს კრიტიკული სისტემის ინფრასტრუქტურა, რომელსაც ექნება ამოცანის შესაბამისი საიმედოობა, წვდომადობა და წარმადობა.
- გადაუდებელი სამედიცინო სამსახურის გამოწვევების ანალიზის საფუძველზე გამოვლენილი ამოცანებისთვის ჩამოყალიბდეს კონკრეტული საკვლევი მიმართულებები და განისაზღვროს პრობლემის გადაჭრის ეტაპები.
- **სასწრაფო დახმარების ბრიგადის შერჩევის ამოცანისთვის** შეიქმნას ხელოვნური ნეირონული ქსელის ზოგადი მოდელი, განისაზღვროს ის შემავალი პარამეტრები, რაც შეიძლება, კავშირში აღმოჩნდეს სასწრაფოს ბრიგადის მიერ პაციენტთან დახარჯულ სავარაუდო დროსთან.
- შემუშავდეს მანქანური დასწავლისთვის საჭირო სტრუქტურის მქონე მონაცემთა კრებული.
- გაანალიზდეს მიღებული მონაცემთა კრებული და რედაქტირებულ იქნას ისეთი ანომალიური შემთვევები, რაც შეცდომაში შეიყვანს მანქანური დასწავლის პროცესს. განისაზღვროს კვლევის გაგრძელების შემდგომი ეტაპები.
- **სასწრაფო დახმარების გამოძახებების პრიორიტეტების განსაზღვრის ამოცანისთვის** უნდა შეირჩეს სისტემა პარამეტრებისა, რომლებიც განისაზღვრება გამოძახების მიმღები ოპერატორების მიერ დაფიქსირებული ინფორმაციის ანალიზის შედეგად.
- არჩეულ პარამეტრებს უნდა მიენიჭოს გარკვეული წონები მათი ფარდობითი კლინიკური მნიშვნელოვნების მიხედვით. პარამეტრები და მათი წონები უნდა შეირჩეს მედიცინის სხვადასხვა სფეროში მომუშავე გამოცდილ ექიმებთან კონსულტაციების საფუძველზე.
- წინასწარ განსაზღვრული სამედიცინო ინფორმაციის საფუძველზე უნდა შეიქმნას ალგორითმების სისტემა თითოეული სამედიცინო შემთხვევის გადაუდებლობის ინდექსისა და დონის გამოსავლენად.

- სასწავლო მოდელის შემთხვევებზე შედარდეს ალგორითმებით გამოთვლილი შედეგები ექიმის შეფასებებთან.

კვლევის ობიექტი და მეთოდები. ბოლო ათწლეულში მონაცემთა ბაზების მართვის სისტემები ვითარდებოდა რამდენიმე მიმართულებით. მთავარი იყო რელაციური ბაზების, როგორც უმრავლესი მართვის საინფორმაციო სისტემების ძირითადი კომპონენტის სრულყოფა მწარმოებლურობის თვალსაზრისით. ვითარდებოდა მოთხოვნების დამუშავების ოპტიმიზაციის ალგორითმები. მეორე მიმართულება გახდა NoSQL („არარელაციური“ ან „არა მხოლოდ რელაციური“) ტიპის ბაზების მართვის სისტემების შექმნა და სრულყოფა, განსაკუთრებით „დიდ მონაცემთა“ დასამუშავებლად, სადაც რელაციური ბაზები არაეფექტურია, მათი Join და სხვა რელაციური ოპერაციებით ცხრილების დამუშავების პროცესის დიდი დროის გამო. ასეთია კორპორაციებში დოკუმენტების დასამუშავებელი სისტემები. მესამე მიმართულება ეხება გრაფული ბაზების შექმნას და განვითარებას. ბოლოს, შეიძლება ვახსენოთ მონაცემთა ბაზების მართვის ჰიბრიდული სისტემები, სადაც ინტეგრირდება რელაციური, NoSQL-ის და გრაფული ბაზების კონცეფციები. ყველა მიმართულება აქტუალურია და რომელი ტიპის ბაზას აირჩევს მომხმარებელი, დამოკიდებულია მართვის საინფორმაციო სისტემის კონკრეტულ მიზნებსა და ფუნქციებზე.

ისეთ მუდმივად მზარდ მონაცემთა საცავშიც კი, როგორც აქვს გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ „112“-ს, ფუნქციონირების 6 წელიწადში ჯერ არ დაგროვილა განსაკუთრებით დიდი რაოდენობის მონაცემები, მაგრამ ამის მიუხედავად, მონაცემთა ბაზის დაპროექტებისას დაშვებული ხარვეზები მონაცემების ზრდასთან ერთად უფრო და უფრო იჩენს თავს. რა პრობლემები აღმოჩნდება კიდევ 6 წლის შემდეგ? სადისერტაციო ნაშრომის ფარგლებში გადაუდებელი დახმარების ოპერატიული მართვის ცენტრისთვის შეირჩა არარელაციური მონაცემთა ბაზების ერთ-ერთი ყველაზე გავრცელებული წარმომადგენელი MongoDB.

თეორიული შედეგების ექსპერიმენტული რეალიზაცია ხორციელდება გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ „112“-ის რეალურ ამოცანებზე. ექსპერიმენტულ ნაწილში გამოყენებულია პროგრამირების ენა Python.

კვლევის ძირითადი შედეგები და შედეგების გამოყენების სფერო. დისერტაციის შედეგებს აქვს პრაქტიკული ღირებულება. შექმნილი მონაცემთა საცავის გამოყენება მიზანშეწონილია „112“-ის სხვადასხვა სერვერებიდან ინფორმაციის თავმოყრისთვის, სტატისტიკის, ანალიტიკისა და სხვა ამოცანებისთვის.

ექსპერიმენტულ ნაწილში შექმნილი სისტემის გამოყენებით ობიექტურად დგინდება გრიპის შემთხვევათა მომსახურების რიგი თითოეული მათგანის გადაუდებლობის დონის მიხედვით, - აღნიშნული ალგორითმების საშუალებით გამოვლინდა ყველა შემთხვევა, რომლებიც გამოცდილმა ექიმმა გადაუდებლობის მაღალი დონით შეაფასა.

ექსპერიმენტულ ნაწილში აღწერილი ინოვაციური მიდგომები შეიძლება გამოყენებულ იქნას რეალური პრობლემების გადასაჭრელად სასწრაფო დახმარების დისპეტჩერიზაციის სფეროში. ზემოთ აღნიშნული პრობლემების გადაწყვეტა დაეხმარება „112“-ის დისპეტჩერს გადაწყვეტილების მიღებაში და შეამცირებს დისპეტჩერის რეაგირების დროს.

გადაუდებელი დახმარების ოპერატიული მართვის ცენტრი „112“ მზად არის, გამოიყენოს ხელოვნური ინტელექტი ბიზნეს პროცესების ოპტიმიზაციისთვის და სრულად მოგვაწოდოს არაკონფიდენციალური ტიპის მონაცემები, რაც საკმარისი იქნება კვლევების განსახორციელებლად. კვლევების დასრულების შემდეგ დაიწყება დანერგვის პროცესი, შესასრულებელი სამუშაოს მასშტაბურობიდან გამომდინარე, „112“-თან თანამშრომლობა პროექტის დასრულების შემდეგაც გაგრძელდება. ასევე, გაგრძელდება თანამშრომლობა საქართველოს ტექნიკურ უნივერსიტეტთანაც, რადგან მსგავს ამოცანებზე მუშაობა მნიშვნელოვნად

დაეხმარება Big Data ტექნოლოგიებისა და მანქანური დასწავლის მიმართულებების განვითარებას უნივერსიტეტსა და მთლიანად, ქვეყანაში.

ნაშრომის მოცულობა და სტრუქტურა: დისერტაციის სრული მოცულობა შეადგენს 153 ნაბეჭდ გვერდს; შედგება რეზიუმეს (ორ ენაზე), სარჩევის, შესავლის, ოთხი თავის და დასკვნისგან. ახლავს 36 ნახაზი, 16 ცხრილი და 63 გამოყენებული ლიტერატურის სია.

ნაშრომის აპრობაცია: დისერტაციის ძირითადი შინაარსი მოხსენებული იყო ინფორმატიკისა და მართვის სისტემების ფაკულტეტის „მართვის ავტომატიზებული სისტემების (პროგრამული ინჟინერია)“ კოლეგიის სამეცნიერო სემინარების სხდომებზე, ასევე:

- ბათუმის შოთა რუსთაველის სახელმწიფო უნივერსიტეტის 80 წლის იუბილესადმი მიძღვნილ სტუდენტთა საერთაშორისო სამეცნიერო კონფერენცია, ბათუმი, 25-27 სექტ. 2015 წ., ბათუმის შოთა რუსთაველის სახელმწიფო უნივერსიტეტი;
- IV საერთაშორისო სამეცნიერო კონფერენცია „კომპიუტინგი/ინფორმატიკა, განათლების მეცნიერებები“, თბილისი, 1-3 ოქტ., 2016;
- VIII საერთაშორისო სამეცნიერო-პრაქტიკულ კონფერენცია „ინტერნეტი და საზოგადოება“, INSO-2017, 7-8 ივლისი, აკ.წერეთლის სახ. უნივერსიტეტი, ქუთაისი. 2015;
- ტამპერეს უნივერსიტეტის ინფორმატიკის კვლევითი ცენტრის სემინარებზე, აპრილი-მაისი, 2017 წ., ტამპერე, ფინეთი.
- ერლანგენ-ნიურნბერგის უნივერსიტის მონაცემთა მართვის ფაკულტეტის სემინარებზე, ნოემბერი, 2017 წ., ერლანგენი, გერმანია.
- ევროპის საგანგებო ნომრის ასოციაციის EENA (European Emergency Number Association) კონფერენციაზე, 2018 წლის 25-27 აპრილს, ლუბლიანაში (სლოვენია)

- პროფესორ კონსტანტინე კამკამიძის დაბადების 90 წლის იუბილესადმი მიძღვნილი სამეცნიერო კონფერენცია. თბილისი, 28 მაისი, 2018 წელი.

პუბლიკაციები: დისერტაციის ძირითადი შედეგები გამოქვეყნებულია 7 სამეცნიერო ნაშრომში, რომელთა ჩამონათვალიც მოყვანილია დისერტაციის და ავტორეფერატის ბოლოს.

დისერტაციის მოკლე შინაარსი

ნაშრომის **პირველ თავში** წარმოდგენილია მონაცემთა რელაციური და არარელაციური ბაზების მახასიათებლების შედარება, განსაზღვრულია იმ ობიექტების კლასი, რომლებისთვისაც ეფექტიანია მონაცემთა არარელაციური ბაზების გამოყენება.

ასევე, განხილულია Hadoop პლათფორმა, რისი მეშვეობითაც შეგვიძლია მონაცემთა საცავის წარმადობის მნიშვნელოვნად გაუმჯობესება.

მიმოხილულია მანქანური დასწავლისა და მისი ქვე-მიმართულებების ტექნოლოგიები. განხილულია Python პროგრამირების ენის მახასიათებლები.

NoSQL მიმდინარეობა პოპულარული გახდა Google Big Table და Amazon DynamoDB (Key-Value storage) კომპანიის პროდუქტების გამოსვლის შემდეგ. ორივე კომპანია NoSQL-ის საჭიროებამდე რელაციური ბაზების პრობლემებმა მიიყვანა, დღეს კი უკვე 200-ზე მეტი სხვადასხვა არარელაციური მონაცემთა ბაზაა ცნობილი.

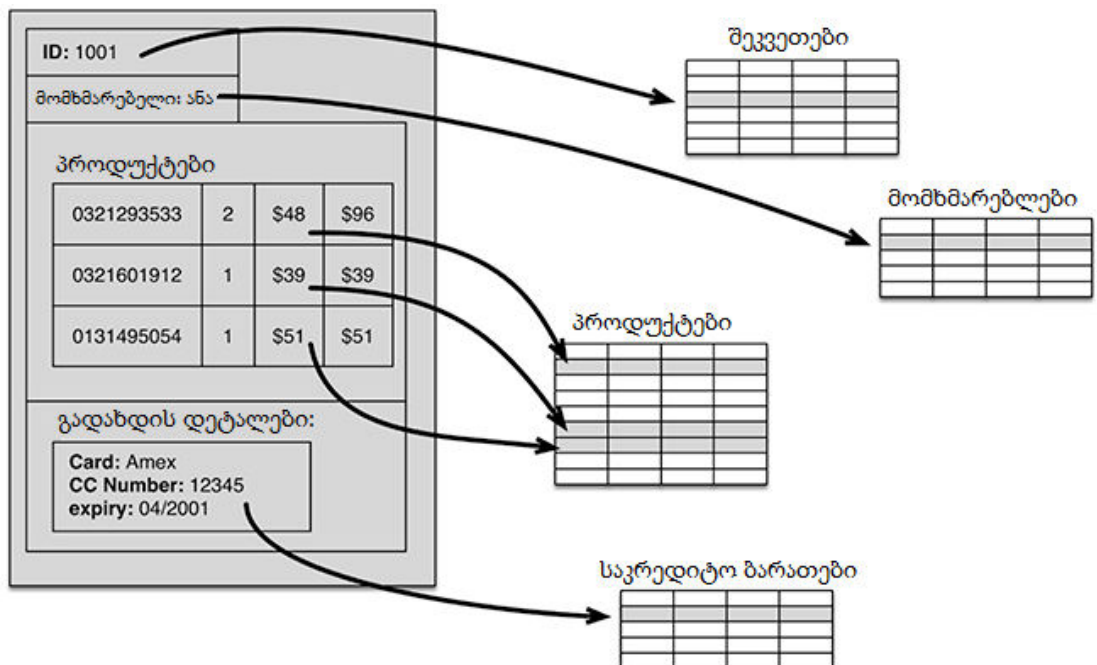
მონაცემთა არარელაციურ ბაზებში ინახება უსქემო მონაცემები, ადვილია მათი დაჯგუფება (clustering), გამოირჩევა დიდი სისწრაფით და ამარტივებს ამოცანის გადაჭრას.

მონაცემთა რელაციურ ბაზაში ინფორმაციას ვერ შევიტანთ წინასწარ ჩამოყალიბებული სტრუქტურის გარეშე. არარელაციურ ბაზებში მსგავსი პრობლემა არ გვაქვს, – ნებისმიერი სახით შეგვიძლია ინფორმაციის შეტანა ცხრილში, ამისთვის ცხრილის წინასწარ შექმნაც კი არაა საჭირო.

მიუხედავად ამგვარი თავისუფლებისა, გარკვეული სქემის დაცვა მაინც სასურველია, შემდგომ ინფორმაციის გაფილტვრა რომ არ გართულდეს.

რელაციურ მოდელში მონაცემები აისახება ერთი სტრუქტურით, აპლიკაციაში კი გამოიყენება სრულიად განსხვავებული სახით. ამგვარად, მონაცემთა სტრუქტურა, რომელიც წარმოდგენილია მონაცემთა ბაზაში, აბსოლუტურად განსხვავდება ოპერატიულ მეხსიერებაში მისი შესაბამისი სტრუქტურისაგან, რომელსაც შემდგომ იყენებს აპლიკაცია.

მონაცემთა დამუშავების პროცესში საკმაოდ დიდი დრო და რესურსი ეთმობა სხვადასხვა ცხრილებიდან მონაცემების წაკითხვას, მათ გაერთიანებას აპლიკაციისთვის საჭირო ობიექტში (Join), მიღებული ობიექტის დამუშავებას, შემდეგ კი შედეგად მიღებული დამუშავებული ობიექტის დაშლას და ცვლილებების განთავსებას შესაბამის ცხრილებში. მონაცემთა რელაციურ ბაზებში ზოგიერთი სტრუქტურის წარმოსადგენად გვიწევს მონაცემებითა და პროგრამით არასასურველი მანიპულირება (ნახ.1).



ნახ. 1 არარელაციური და რელაციური მონაცემთა ბაზების სტრუქტურების შედარება

მონაცემთა არარელაციურ ბაზებს, რელაციურთან შედარებით, აქვს სრულიად განსხვავებული მოდელი. დოკუმენტზე ორიენტირებული NoSQL ბაზები მონაცემებს ინახავს JSON (JavaScript Object Notation) ფორმატში, ძირითადად იმ სახით, რომელსაც მიიღებს აპლიკაცია. ის შეიძლება შეიცავდეს რელაციური მოდელის რამდენიმე ცხრილის გადაბმით მიღებულ ინფორმაციას ერთ დოკუმენტში/ობიექტში, რაც უზრუნველყოფს ჩაწერა/წაკითხვის ოპერაციების წარმადობის გაუმჯობესებას (ნახ.1).

დინამიური სქემების მოდელი საშუალებას გვაძლევს, იერარქიული კავშირები და რთული სტრუქტურები შედარებით მარტივად წარმოვადგინოთ.

სასურველია, გვესმოდეს როგორც რელაციური, ასევე არარელაციური ბაზების პრინციპები და კონკრეტული ამოცანისთვის შევარჩიოთ სასურველი სისტემა. ცხრილ 1-ში მოცემულია რელაციურ და არარელაციურ მონაცემთა ბაზებისთვის შესაბამის ამოცანათა მახასიათებლების განსხვავების აღწერა.

ცხრ. 1. რელაციური და არარელაციური მონაცემთა ბაზების ამოცანათა კლასების შედარება

NoSQL	<ul style="list-style-type: none"> • მონაცემთა საცავს უნდა შეეძლოს განსაკუთრებით დიდი დატვირთვის გამკლავება • ტრანზაქციები ძირითადად OLAP სახისაა • გჭირდება საცავი, რომლის განვითარებაც ჰორიზონტალურაა საჭირო (horizontal scale) • თუ სასურველია სიმარტივე, რთული SQL მოთხოვნების გარეშე
RDBMS	<ul style="list-style-type: none"> • მონაცემთა საცავს უნდა შეეძლოს ასევე დიდი დატვირთვის გამკლავება, მაგრამ ტრანზაქციები ძირითადად OLTP სახისაა • გჭირდება წარმადობა უფრო დახვეწილ და რთულ სტრუქტურაზე • თუ გჭირდება SQL ენის გამოყენება

NoSQL ბაზებში მონაცემების შენახვის 4 ძირითადი ტიპი არსებობს:

1. **Key-Value Store** – აქვს დიდი ჰემ ცხრილი გასაღებებისა და მათი მნიშვნელობებისათვის. მონაცემებზე წვდომა გვაქვს პირველადი გასაღების (Primary Key) გამოყენებით. (მაგალითად, Riak, Amazon S3 [Dynamo], Redis...);

2. **Document-based Store** - ინახავს იარლიყიანი ელემენტებისგან (tagged elements) შემდგარ დოკუმენტებს. ამ ტიპის მონაცემთა ბაზებს JOIN ოპერატორის მხარდაჭერა არ აქვთ. ცხრილების გადაბმის/გაერთიანების ლოგიკა აპლიკაციის მხარეს არის დასაწერი. სამაგიეროდ, ობიექტზე ორიენტირებული აპლიკაციისთვის ძალიან მარტივია მთელი საჭირო ინფორმაციის ერთი დოკუმენტიდან ამოღება. (მაგალითად, MongoDB, CouchDB...);

3. **Column-based Store** - მეხსიერების თითოეული ბლოკი ინახავს მხოლოდ ერთი სვეტის ინფორმაციას, რაც ბევრად ამარტივებს აგრეგატული ფუნქციების (MIN, SUM, AVG, COUNT...) შესრულებას. ამ ტიპის მონაცემთა ბაზები კვლავ ცხრილურ სტრუქტურას იყენებენ, მაგრამ JOIN ოპერატორის მხარდაჭერა არ აქვთ. ცხრილების გადაბმის/გაერთიანების ლოგიკა აპლიკაციის მხარეს არის დასაწერი. (მაგალითად, HBase, Cassandra);

4. **Graph-based**-ქსელური ტიპის მონაცემთა ბაზა, რომელიც იყენებს წიბოებსა და კვანძებს მონაცემების შესანახად და წარმოსადგენად. მაგალითად, Neo4.

ექსპონენციალურად გაზრდილი რაოდენობის მონაცემების დამუშავებას სრულიად განსხვავებული სისტემა სჭირდება არა მხოლოდ რელაციური და არარელაციური ბაზების დონეზე, არამედ იმ სერვერული არქიტექტურის დონეზე, სადაც ვაყენებთ მონაცემთა ბაზებს. დღეისთვის საუკეთესო გამოსავალი დააპროექტა Apache Software Foundation-მა, სახელით Hadoop.

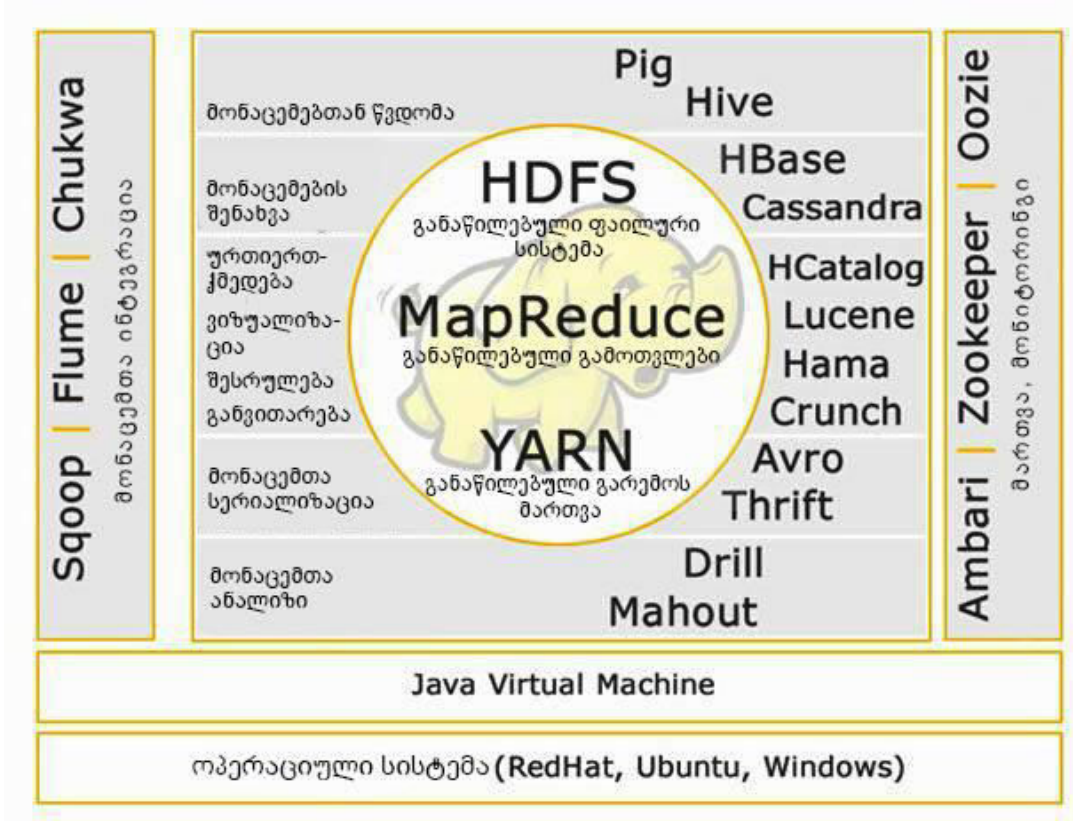
Hadoop ეკოსისტემაში სხვადასხვა პროდუქტია გაერთიანებული, ბირთვად კი სამი ძირითადი კომპონენტი აქვს (ნახ.2):

HDFS (Hadoop Distributed File System) - განაწილებული ფაილური სისტემა მონაცემების შესანახად

Map Reduce - მთავარი კომპონენტი განაწილებული გამოთვლების ჩასატარებლად

YARN (Yet Another Resource Negotiator) - განაწილებული გარემოს მართვა.

კონსისტენტურობისთვის Hadoop ფაილური სისტემა „Shared Nothing“ მოდელს იყენებს, რაც გულისხმობს, არქიტექტურას, სადაც კლასტერში შემავალი კვანძები ერთმანეთისგან სრულიად დამოუკიდებლად მუშაობენ. საერთო რესურსის არქონა ნიშნავს, რომ კვანძების ერთმანეთთან ურთიერთქმედება მინიმუმამდეა დაყვანილი და შესაბამისად, აღარ გვაქვს “Bottleneck” (ბოთლის შევიწროვებული ყელის) პრობლემა. აღნიშნული მოდელი გვაძლევს ჰორიზონტალური ზრდის თითქმის ულიმიტო შესაძლებლობას.



ნახ. 2 Hadoop ეკოსისტემა

მეორე თავში გადმოცემულია დისერტაციაში განხილული თეორიული საკითხების პრაქტიკული რეალიზება. ყურადღება გამახვილებულია განაწილებული ბაზების მონაცემთა მთლიანობის (Data Integrity) და ტრანზაქციათა იზოლირების დონეებზე (Transaction Isolation Levels). განიხილება ACID (Atomicity, Consistency, Isolation, Durability) პრინციპები.

მონაცემთა დოკუმენტ-ორიენტირებული, არარელაციური ბაზის MongoDB საფუძველზე განიხილება ბაზის მწარმოებლობის ძირითადი მახასიათებლები და აღწერილია სისტემის გამართვის ეტაპები.

იქიდან გამომდინარე, რომ ბოლო წლებში ექსპონენციალურად იზრდება არასტრუქტურირებული მონაცემების რაოდენობა, კრიტიკულად მნიშვნელოვანი ხდება ინფორმაციის შენახვის სტრატეგიების შემუშავება. ამოცანის შესაბამისი მონაცემების შენახვისა (Storage) და კონსისტენტობის (Consistency) მოდელები არის გასაღები იმისა, თუ როგორ მოვარგოთ მონაცემთა საცავის ინფრასტრუქტურა დღევანდელ მოთხოვნებს.

მიუხედავად იმისა, რომ NoSQL მიმართულების აქტიურ განვითარებაზე მეტყველებს ბოლო წლებში შექმნილი 150-ზე მეტი სხვადასხვა ბაზა, მაინც არ იქნება სწორი იმის თქმა, რომ რელაციური ბაზები წარსულს ჩაბარდა. მრავლად არის ამოცანები, რომელთა გადაწყვეტისთვის ისევ და ისევ რელაციური ბაზები არის საიმედო, უფრო მეტიც, ბევრია ისეთი ამოცანა, რომლებიც რელაციური ბაზების გარეშე ვერ გადაწყდება.

მართალია, შეგვიძლია არარელაციურ ბაზებშიც ჩავდოთ ისეთი ლოგიკა, რომელიც სრულად დააკმაყოფილებს ACID მოთხოვნებს, მაგრამ გარკვეული ზღვრის შემდეგ ძალიან რთულდება მონაცემების მოვლა და პროგრამული უზრუნველყოფა. ასე რომ, დიდი სისტემების დამპროექტებლებს უნდა ესმოდეთ როგორც რელაციური, ასევე არარელაციური ბაზების პრინციპები და კონკრეტული ამოცანის მიხედვით

აირჩიონ შესაბამისი მონაცემთა ბაზა. ასევე, გასათვალისწინებელია ის ფაქტიც, რომ, როგორც წესი, მონაცემთა საცავებში NoSQL ბაზები ოპტიმიზირებულია OLAP და არა OLTP მოთხოვნებზე. შესაბამისად, თუ ცნობილია, რომ ამოცანის მიზნებიდან გამომდინარე, უფრო ხშირად მოკლე და მსუბუქი ტრანზაქციების გამოყენება იქნება საჭირო, გადაწყვეტილების მისაღებად უნდა განისაზღვროს, რამდენად ეფექტური იქნება ტრადიციული რელაციური ბაზების ნაცვლად არარელაციური ბაზების გამოყენება.

ამგვარად, რელაციური და არარელაციური მონაცემთა ბაზების გამოყენება თანამედროვე საინფორმაციო სისტემების ასაგებად, კი არ გამორიცხავენ, არამედ ავსებენ ერთმანეთს.

მონაცემთა ბაზასთან ერთდროულად რამდენიმე მომხმარებლის მიმართვისას არსებობს იმის შესაძლებლობა, რომ ერთი მომხმარებლის მიერ შესრულებული ცვლილებები ზემოქმედებას მოახდენს სხვა მომხმარებლის მუშაობის პროცესზე. ნაშრომში გამოვლენილია რამდენიმე გავრცელებული პრობლემა და შედარებული სხვადასხვა სისტემებში მათი გადაჭრის მეთოდები:

დაკარგულ განახლებათა პრობლემა (the lost update problem). დამთხვევათა პრობლემების თვალსაჩინო სახეა ონლაინ დაჯავშნის სისტემა, როდესაც 2 მომხმარებელი ერთდროულად ჯავშნის თავისუფალ ნომერს და ორივე მათგანი დარწმუნებულია, რომ ნომერი სწორედ მან დაჯავშნა.

შეცდომითი წაკითხვა (dirty reads) ხდება მაშინ, როდესაც ერთი ტრანზაქცია კითხულობს იმ მონაცემებს, რაც ზუსტად იმავე დროს იცვლება მეორე ტრანზაქციის მიერ. ცხადია, თუ მეორე ტრანზაქცია უკან დაბრუნდება, პირველ მომხმარებელს უკვე წაკითხული, არასწორი ინფორმაცია დარჩება.

არაგანმეორებადი წაკითხვა (nonrepeatable read). ამ პრობლემას ვაწყდებით იმ შემთხვევაში, როდესაც მიმდინარე ტრანზაქცია ერთსა და

იმავე მონაცემებს შესრულების სხვადასხვა ეტაპებზე რამდენჯერმე კითხულობს. წაკითხვებს შორის შუალედში თუ სხვა ტრანზაქცია მონაცემებში ცვლილებას შეიტანს, პირველი ტრანზაქციის მიერ სხვადასხვა დროს ამოღებული მონაცემები ერთმანეთის იდენტური აღარ იქნება.

მოჩვენებითი წაკითხვის (phantom read) პრობლემა გვხვდება, თუ ერთი ტრანზაქციის განმავლობაში ერთი და იგივე მოთხოვნა რამდენჯერმე ეშვება და მეორე მოთხოვნის შედეგში გვხვდება ისეთი ჩანაწერები, რაც არ გვექონდა პირველი მოთხოვნის დროს. ეს პრობლემა არაგანმეორებადი კითხვის პრობლემის მსგავსია, იმ განსხვავებით, რომ ცვლილება შედეგის ჩანაწერების რაოდენობებშია.

ამ ტიპის ყველა პრობლემის გადაჭრა ხორციელდება ჩაკეტვის მექანიზმებით (locking mechanisms) რომლებიც უზრუნველყოფს მომხმარებლებისა და ტრანზაქციების ერთმანეთისაგან იზოლირებას. ადრე პროგრამისტებს საჭირო დაცვის განსახორციელებლად ჩაკეტვების მართვის საკუთარი სისტემების შემოტანა სჭირდებოდათ, მაგრამ დღეს ჩაკეტვების მართვას მთლიანად მონაცემთა ბაზების მართვის სიტემა ახორციელებს. დაკარგულ განახლებათა პრობლემას თანამედროვე მონაცემთა ბაზების მართვის სიტემები წაკითხვისა და ჩაწერის ჩაკეტვების მართვით ახორციელებს. სხვა შესაძლო პრობლემების გადასაწყვეტად ჩვენ უბრალოდ განვსაზღვრავთ საჭირო იზოლაციის დონეს იმ ოთხი სტანდარტული დონიდან, რომლებსაც ადგენს 1992 წლის SQL სტანდარტი. დაცვის ოთხი სხვადასხვა სტანდარტის არსებობა დაკავშირებულია იმ ფაქტთან, რომ რაც უფრო ძლიერია დაცვა (მაღალია იზოლაციის დონე) მით უფრო დაბალია მოქმედების სისწრაფე:

- Read uncommitted დონე არ უზრუნველყოფს დაცვას იმ დამთხვევათა პრობლემებისაგან, რომლებიც განვიხილეთ, მაგრამ უზრუნველყოფს მოქმედების მაქსიმალურ სისწრაფეს, რადგან იზოლაციის ამ დონეში არ ხდება მოთხოვნილი მონაცემების ბლოკირება (lock). ცხადია, ამ დონის გამოყენებისას მზად უნდა ვიყოთ, რომ თუ ერთი პროგრამიდან შეტანილი

ცვლილებები უკან დაბრუნდება, update-სა და rollback-ს შორის შუალედში ყველა სხვა მოთხოვნას dirty data-ს ანუ დამახინჯებულ მონაცემებს გავატანთ. ამ დონეს გამოვიყენებთ ისეთი მოთხოვნებისთვის, რომელთა სისწორეც არ არის სასიცოცხლოდ აუცილებელი, ან თუ ვიცით, რომ სანამ ჩვენ ვკითხულობთ, მესამე მხარე ცვლილებას არ შეიტანს;

- Read committed დონე გარანტირებულად არიდებს თავს შეცდომითი წაკითხვების პრობლემას, რადგანაც იზოლაციის ამ დონეზე ნებადართულია მხოლოდ დაფიქსირებული ტრანზაქციებით შეცვლილი მონაცემების წაკითხვა;

- Repeatable read დონე იცილებს ასევე არაგანმეორებადი წაკითხვების პრობლემასაც;

- Serializable დონე უზრუნველყოფს ერთდროული ტრანზაქციების სრულ განცალკევებას, რაც ხორციელდება ტრანზაქციაში მონაწილე ყველა სტრიქონის ჩაკეტვით. ფანტომების წარმოქმნის პრობლემა წყდება გასაღების დიაპაზონის დაბლოკვით. იზოლირების ამ დონის დაყენების შემთხვევაში მცირდება პარალელიზმის ხარისხი და იზრდება სისტემაზე დატვირთვა.

ჩვეულებრივ ყველაზე უფრო ხშირად გამოიყენებენ read committed იზოლაციის დონეს.

სხვადასხვა მონაცემთა ბაზებში ტრანზაქციის იზოლირების დონეებს გარკვეული განსხვავებები აქვს. მაგალითად, oracle მონაცემების წაკითხვისას lock-ს საერთოდ არ იყენებს და dirty data-ს პრობლემა latch-ებით აქვს გადაწყვეტილი.

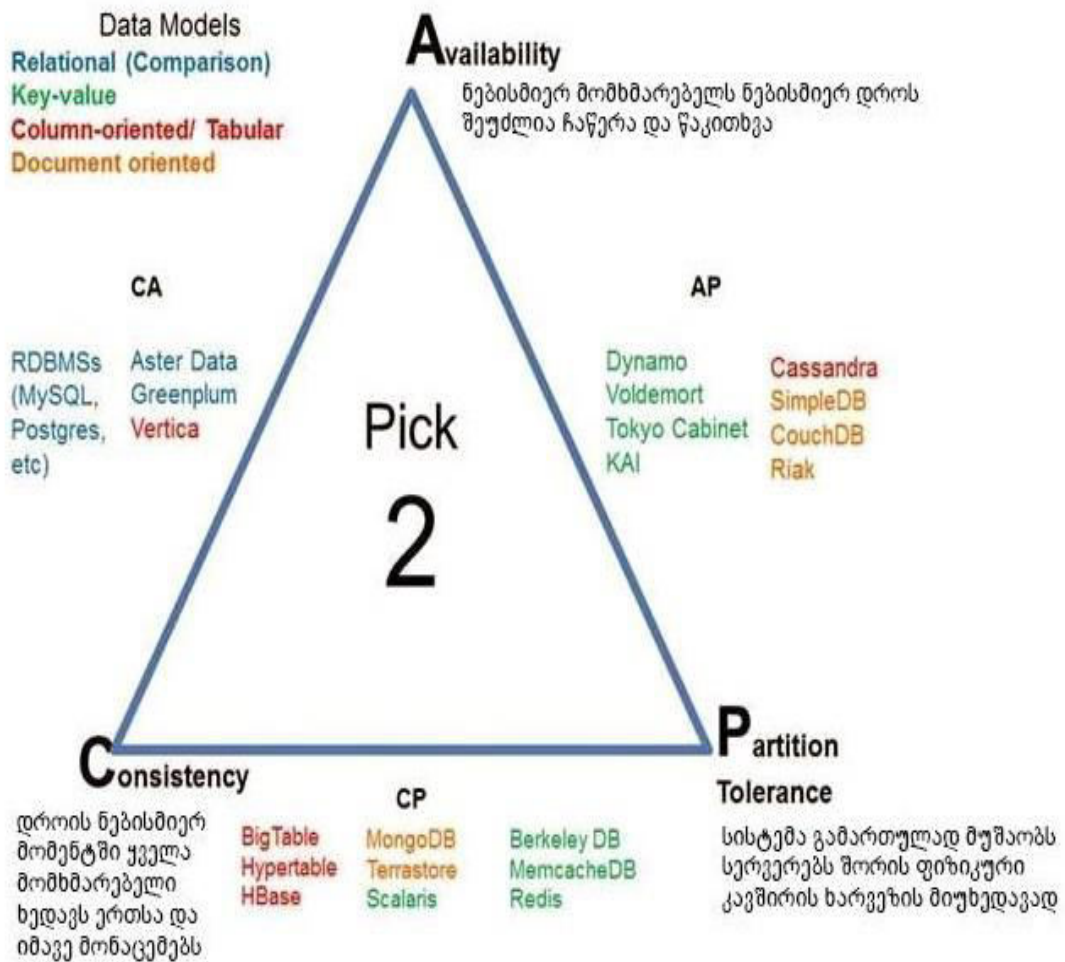
Lock-ის შემთხვევაში რიგში ვდგებით, რომ მივიღოთ მონაცემებზე წვდომა, latch-ის დროს კი პერიოდულად ვაკითხავთ მონაცემებს და ვამოწმებთ, თუ არის თავისუფალი.

Oracle, MsSQL Server და DB2-ში იზოლაციის დონე შეიძლება მიეთითოს კონკრეტული სესიის ან ტრანზაქციისთვის (DB2-ს დამატებით აქვს აპლიკაციის დონეც) [3,4].

NoSQL სისტემებისთვის, კონკრეტულად კი MongoDB-სთვის ტრანზაქციების იზოლირების მსგავსი მეთოდები არ გვაქვს [5]. ამ პრობლემას MongoDB წყვეტს თითოეული დოკუმენტის ატომარობის პრინციპის დაცვით. ასე რომ, საინფორმაციო ბაზის დიზაინის შემუშავებისას, მთელი ის მონაცემები, სადაც გვჭირდება, რომ ატომარობის პრინციპი დაცული იყოს, უნდა შევინახოთ ერთ დოკუმენტში. ძირითად შემთხვევებში დოკუმენტის დონეზე არსებული ატომარობა საკმარისია იმ საიმედოობის მისაღწევად, რასაც რელაციური მონაცემთა ბაზები ACID პრინციპების დაცვით გადაწყვეტდა.

მაგალითად, MongoDB-ს შემთხვევაში ერთმანეთთან დაკავშირებული მონაცემები შეგვიძლია მოვათავსოთ ერთ დოკუმენტში, ჩადგმული მასივის (nested arrays) ან ჩადგმული დოკუმენტის (nested document) გამოყენებით. ამ შემთხვევაში მხოლოდ ერთი დოკუმენტის განახლება მოგვიწევს, რაც, შესაბამისად, ატომარული პროცესი იქნება. რელაციურ მონაცემთა ბაზებში იმავე სახის დაკავშირებული მონაცემების წარმოსადგენად დაგვჭირდებოდა სხვადასხვა ცხრილები და ჩანაწერები, რაც აუცილებელს გახდიდა ტრანზაქციის იზოლირების დონეების გამოყენებას მონაცემების განახლებისთვის.

CAP თეორემა (Consistency, Availability, Partition_tolerance) ან ბრიუერის (*Brewer Eric*) თეორემა არის ევრისტიკული მტკიცება იმის შესახებ, რომ განაწილებული გამოთვლების ნებისმიერ რეალიზაციაში შესაძლებელია სამი (C,A,P) თვისებიდან მხოლოდ ორის უზრუნველყოფა (ნახ.3).



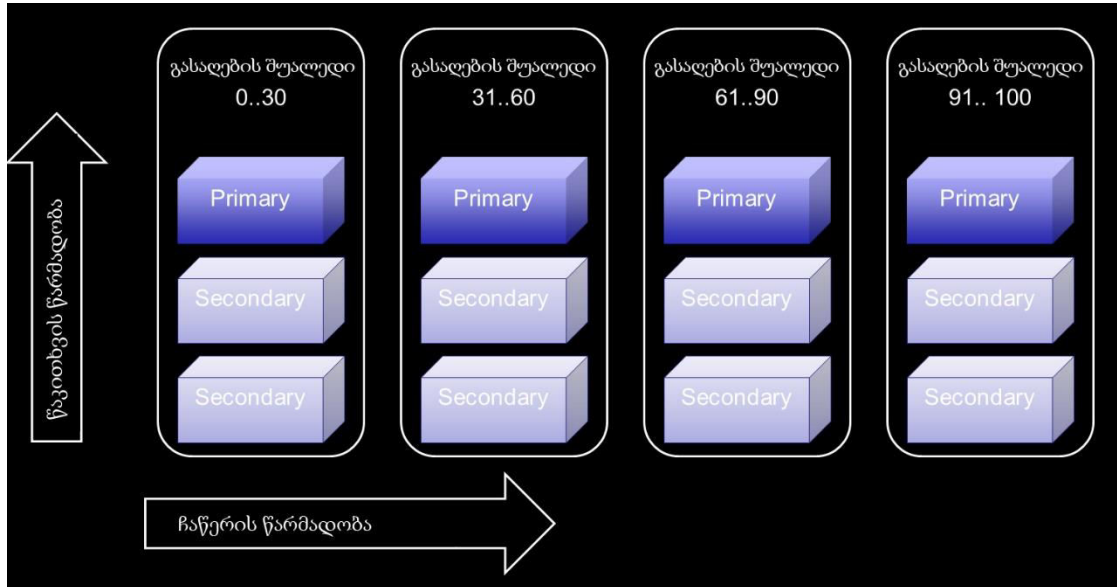
ნახ. 3 CAP თეორემა

Consistency (მთლიანობა) – განაწილებული კომპიუტერული სისტემის ყველა კვანძში (node – კლასტერში) მონაცემები შეთანხმებულია (არაწინააღმდეგობრივია დროის მოცემულ მომენტში);

Availability (წვდომადობა) – ნებისმიერი მოთხოვნა განაწილებულ სისტემასთან სრულდება კორექტული პასუხით (თუ რომელიმე კვანძი გათიშულია, მას ცვლის სხვა);

Partition Tolerance (განაწილების მდგრადობა) – კლასტერი განაგრძობს ფუნქციონალობას მაშინაც კი როდესაც კვანძებს შორის არის კომუნიკაციის პრობლემა.

შემუშავებულია კრიტიკული სისტემის არქიტექტურა, ინფრასტრუქტურა და დაპროექტებულია მაღალი წვდომადობის მქონე მონაცემთა საცავი (ნახ.4).



ნახ. 4 სერვერული უზრუნველყოფა replication და sharding ტექნოლოგიებით

მონაცემთა ბაზის სამივე ძირითადი მახასიათებლის (პროცესორი, ოპერატიული მეხსიერება და ხისტი დისკი), ისევე როგორც საიმედოობისა და წვდომადობის გასაუმჯობესებლად ვიყენებთ Sharding-ისა და რეპლიკაციის ტექნოლოგიებს. სისტემა აკმაყოფილებს წამოყენებულ ყველა ტექნიკურ და ფუნქციონალურ მოთხოვნილებას, ამასთან ერთად მიღწეულია მაღალი ხარისხობრივი მახასიათებლები, როგორცაა უსაფრთხოება, მართვადობა და სისწრაფე.

რეპლიკაცია არის პროცესი, რომლის დროსაც მონაცემები სინქრონიზირდება ერთი, მთავარი სერვერიდან რამდენიმე სათადარიგო სერვერზე. რეპლიკაციის დროს მთავარ (primary) სერვერზე ჩაწერისა და კითხვის ოპერაციების განხორციელება შეგვიძლია, სათადარიგო სერვერებიდან კი მხოლოდ ვკითხულობთ.

რეპლიკაციას რამდენიმე ძირითადი დადებითი მხარე გააჩნია:

- მონაცემების კითხვის გაუმჯობესება – სერვერის რამდენ ასლსაც გავმართავთ, იმდენი დამოუკიდებელი წყაროდან შეგვიძლია ინფორმაციის პარალელურ რეჟიმში წაკითხვა;

- Disaster Recovery – პრობლემის ან გეგმიური სამუშაოების დროს შეგვიძლია რომელიმე სათადარიგო (standby) სერვერი ვაქციოთ ძირითად (primary) სერვერად;

- აღარ არის საჭირო backup-ებისა და ინდექსების რეორგანიზაციის გამო მონაცემთა ბაზის გაჩერება ან შენელება;

- რეპლიკა სეტები აპლიკაციის დონეზე არ ჩანან.

Sharding ტექნოლოგია მონაცემთა ბაზის დანაწევრების (database partitioning) ერთერთი სახეა. Shard ტერმინი ერთი მთლიანის პატარა ნაწილს ნიშნავს, – შარდინგის მეშვეობით დიდი ზომის უმართავ ბაზას ვყოფთ პატარა, სწრაფ და ადვილად მართვად ნაწილებად.

საჩვენებლად აწყობილია 4 რეპლიკა სეტი, რომლებიც შარდინგ ტექნოლოგიით ინაწილებენ რესურსს (ნახ.4). სასწავლო რეჟიმში ყველა სერვისი ერთ კომპიუტერზე იყო გაშვებული და განსხვავებულ პორტებზე მუშაობდა. რეალურ შემთხვევაში მონგოს სერვისები გაშვებული გვაქვს სხვადასხვა სერვერზე.

მთავარ სერვერთან კავშირის დაკარგვის ან გათიშვის შემთხვევაში სათადარიგო სერვერები აწყობს არჩევნებს და ირჩევს ახალ primary სერვერს.

რეპლიკა სეტების არჩევნებში მონაწილეობს წინასწარ განსაზღვრული სერვერები – არბიტრები. არბიტრი არ შეიცავს მომხმარებელთა ინფორმაციას, მისი ერთადერთი დანიშნულებაა არჩევნებში ხმის მიცემა. თუ Replica set-ში გვაქვს ლუწი რაოდენობის კვანძი, მაშინ არბიტრის დამატებით ვიღებთ კენტი რაოდენობის წევრებს, რითაც ვაღწევთ ხმათა უმრავლესობას არჩევნებში.

მესამე თავში გაანალიზდა გადაუდებელი სამედიცინო სამსახურის გამოწვევები და გამოვლინდა ამოცანები კომპიუტერული მეთოდების გამოყენებისათვის.

„112“-ის დისპეტჩერის ამოცანაა, გამოავლინოს ყველაზე მაღალი კატეგორიის გადაუდებლობის შემთხვევა აქტიური გამოძახებების სიიდან და გადასცეს ის იმ სასწრაფო დახმარების ბრიგადას, რომელსაც შეუძლია ყველაზე სწრაფი რეაგირება. ასეთი ამოცანა საკმაოდ რთული გადასაწყვეტია, რადგან გამოძახებათა რაოდენობა ხშირად ბევრია, რაც სტრესულ სიტუაციაში მუშაობასთან ერთად იწვევს დისპეტჩერის დაღლილობას, ნერვიულობას და აქედან გამომდინარე, არასწორი გადაწყვეტილებების გაზრდილ რიცხვს.

სასწრაფოს ბრიგადის გათავისუფლებისა და ახალ მისამართზე მისვლის სავარაუდო დროის არცოდნის გამო დისპეტჩერებს ხშირად უწევთ არაოპტიმალური გადაწყვეტილებების მიღება.

დისპეტჩერიზაციის გაუმჯობესების მიზნით, უნდა შეიქმნას კომპიუტერული სისტემის მოდელი რომელიც ობიექტური გამოთვლების საფუძველზე საშუალებას მოგვცემს, ერთის მხრივ, გამოვლინდეს ყველაზე მაღალი კატეგორიის გადაუდებლობის მქონე შემთხვევები (გამოძახებები), მეორეს მხრივ კი შეირჩეს სასწრაფო დახმარების ის ბრიგადები, რომლებიც შეძლებენ არჩეული შემთხვევების ყველაზე სწრაფად მომსახურებას.

ბრიგადის შერჩევის პროცესი იწყება პაციენტის შესაძლო ჰოსპიტალიზაციის ალბათობის გამოთვლით. თუ ჰოსპიტალიზაციის ალბათობა მაღალია ($P \geq 0.5$) მოცემულ შემთხვევაზე გაგზავნილი სასწრაფო დახმარების მანქანა აღარ განიხილება როგორც თავისუფალი დროის უახლოეს ინტერვალში.

თუ შემთხვევის ჰოსპიტალიზაციის ალბათობა დაბალია ($P < 0.5$) ალგორითმი შეაფასებს სასწრაფო დახმარების მანქანის შესაძლო დაკავებულობის დროს და მიაწვდის ამ ინფორმაციას დისპეტჩერს.

ამ დროის შუალედის გავლის შემდეგ, სავარაუდოა, რომ ბრიგადა განთავისუფლდება.

უნდა აღინიშნოს, რომ ალგორითმი იძლევა მხოლოდ ალბათობის სიდიდეებს და არა დროის ზუსტ შუალედებს. მიღებული რეკომენდაცია დისპეტჩერმა უნდა გადაამოწმოს უშუალოდ ბრიგადის ექიმთან.

პაციენტის ჰოსპიტალიზაციის ალბათობა, ისევე, როგორც მიახლოებითი დრო, რასაც ბრიგადა ხარჯავს პაციენტზე, გამოითვლება ისეთი მკაცრად განსაზღვრული მონაცემების საფუძველზე, როგორცაა მაგალითად, ინციდენტის ტიპი, პაციენტის ასაკი და სქესი, ბრიგადის ინფორმირების, რეაგირების, ადგილზე მისვლისა და გათავისუფლების დროები.

პაციენტთან დახარჯული ალბათური დროის გამოთვლის გარდა, მნიშვნელოვანი იქნება ბრიგადის გადაადგილების სავარაუდო დროც, რასაც მარშრუტიზაციის სერვისებით მივიღებთ. არსებული სერვისები მოგვცემს საშუალებას, გავარკვიოთ, გზებზე არსებული „საცობების“ გათვალისწინებით, რა დრო იქნება საჭირო ერთი პუნქტიდან მეორემდე გადასაადგილებლად (მაგალითად Google traffic).

გადაუდებელი სამედიცინო სამსახურის დისპეტჩერიზაციის ეფექტურობის გაზრდის მიზნით შემუშავებულ იქნა ხელოვნური ნეირონული ქსელის ზოგადი მოდელი და დაიგეგმა ამოცანის გადაწყვეტის ეტაპები. პროგრამირების ენა Python-ის საშუალებით შესრულდა extract, transform, load (ETL) და მონაცემთა წინასწარი დამუშავების (data preprocessing) პროცესები, რის შედეგადაც მიღებულ იქნა მანქანური დასწავლისთვის საჭირო სტრუქტურის მქონე მონაცემთა კრებული ბოლო 5 წლის გადაუდებელი სამედიცინო დახმარების შემთხვევების ბაზაზე. შემდგომ ეტაპზე Python პროგრამირებისა და Weka ხელსაწყო გამოყენებით გაანალიზებულ იქნა მიღებული მონაცემთა ნაკრები. გამოვლინდა, წაიშალა და შესწორდა ანომალიური შემთხვევები, რაც გაუთვალისწინებლობის დროს უარყოფითად აისახებოდა მანქანური დასწავლის შედეგებზე.

მეოთხე თავში სასწრაფო სამედიცინო მომსახურების დისპეტჩერების დასახმარებლად შეიქმნა გრიპის სეზონის შემთხვევებზე მორგებული

მოდელი და ალგორითმები თითოეული შემთხვევის სასწრაფო სამედიცინო მომსახურების გადაუდებლობის დასადგენად. პროგრამული რეალიზაციის მიზნით გამოყენებულია მულტიპარადიგმული ენა Python. ალგორითმების გამოყენებით შესაძლებელია ობიექტურად იქნას დადგენილი გრიპის შემთხვევათა მომსახურების რიგი და თითოეული შემთხვევის გადაუდებლობის დონე. დისპეტჩერს წარედგინება ყველაზე მაღალი გადაუდებლობის მქონე შემთხვევები, რის საფუძველზეც მას შეუძლია სწრაფად მიიღოს ადეკვატური გადაწყვეტილება სასწრაფო დახმარების მანქანებისა და ექიმთა ბრიგადების გაგზავნის თაობაზე.

საჭირო მონაცემთა მოპოვება შესაძლებელია ჩვეულებრივი ბარათებიდან, რომლებიც ივსება შსს სსიპ „112“-ის ოპერატორების მიერ თითოეული გამოძახების შემთხვევაზე.

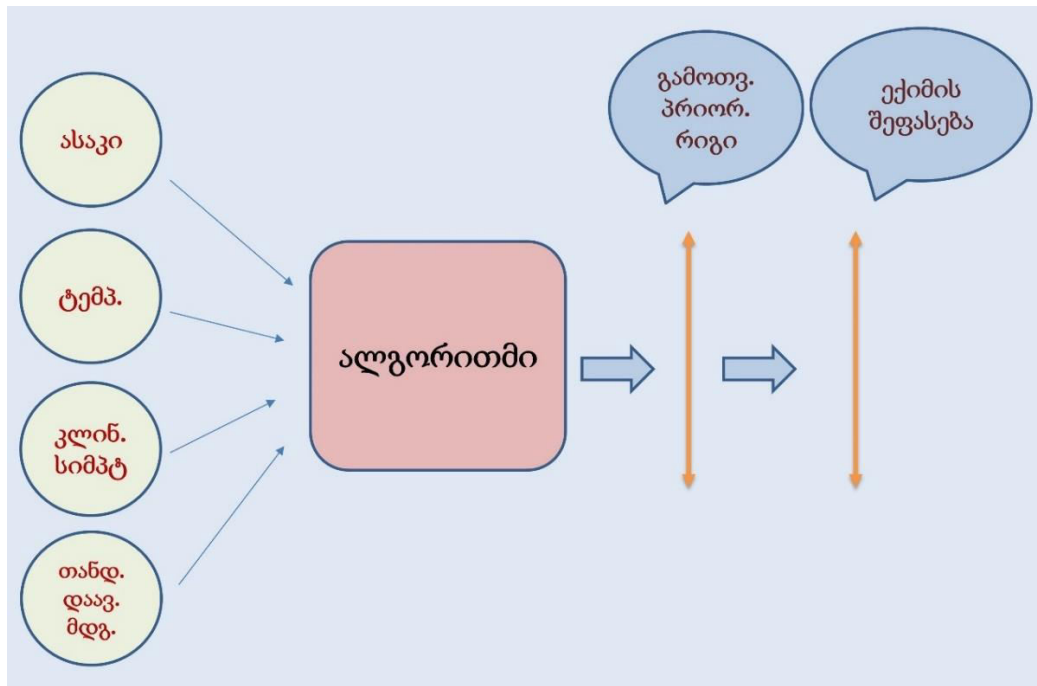
გამოცდილ ექიმებთან კონსულტაციის შემდეგ შეირჩა გრიპის შემთხვევების შემდეგი ობიექტური მახასიათებლები:

- პაციენტის ასაკი,
- ტემპერატურა,
- კლინიკური სიმპტომები,
- თანმხლები ქრონიკული დაავადებები და მდგომარეობები,

რომლებიც ზრდიან გრიპის გართულების რისკ ფაქტორს.

გრიპის შემთხვევებზე შემოსული გამოძახებების ტიპიური ნიმუშების მიხედვით შედგენილია საცდელი მოდელი. ამ მოდელზე გამოიცადა სხვადასხვა ალგორითმი გამოძახების მომსახურების გადაუდებლობის ინდექსის ანუ რისკ-ფაქტორის გამოსათვლელად.

სხვადასხვა ალგორითმის ობიექტურობის გამოსავლენად გამოთვლის შედეგები შედარდა გამოცდილი ექიმის მიერ საცდელი მოდელის შემთხვევების გადაუდებლობის ინდექსის შეფასებებთან (ნახ.5).



ნახ. 5 გრიპის შემთხვევათა პრიორიტეტულობის დადგენის სქემა

საცდელი მოდელის შემთხვევებისთვის შედგა მომსახურების პრიორიტეტულობის ექიმისა და ალგორითმების მიერ დალაგებული რიგები: ექიმის ქულის მიხედვით და ალგორითმების მიხედვით გამოთვლილი მომსახურების გადაუდებლობის ინდექსის მნიშვნელობებით.

ალგორითმის შერჩევასა განსაკუთრებული ყურადღება ექცეოდა ყველაზე უფრო პრიორიტეტული შემთხვევების გამოვლენას. უპირატესობა ენიჭებოდა იმ ალგორითმს, რომელიც გამოავლენდა ექიმის მიერ მითითებულ მაღალი გადაუდებლობის ინდექსის მქონე ყველა შემთხვევას, ან ასეთი შემთხვევების უმეტესობას.

მოდელი აკმაყოფილებს მის მიმართ წაყენებულ ზოგად მოთხოვნებს და წარმოადგენს მოქნილ და დინამიურ სისტემას.

აღნიშნული ალგორითმების საშუალებით შესაძლებელია გამოვლინდეს ყველა შემთხვევა, რომლებსაც გამოცდილი ექიმი პრიორიტეტულად მიიჩნევს.

დასკვნა

სადისერტაციო ნაშრომის ფარგლებში ჩატარებული კვლევის შედეგების საფუძველზე შესაძლებელია შემდეგი დასკვნების გაკეთება:

- მონაცემთა რელაციური და არარელაციური ბაზების მახასიათებლების შედარების საფუძველზე განსაზღვრულია იმ ობიექტების კლასი, რომლებისთვისაც ეფექტიანია მონაცემთა არარელაციური ბაზების გამოყენება.

- მონაცემთა ბაზების მწარმოებლურობის ძირითადი მახასიათებლების - მონაცემთა მთლიანობის (Data Integrity), ტრანზაქციათა იზოლირების დონეების (Transaction Isolation Levels), ACID (Atomicity, Consistency, Isolation, Durability) პრინციპების და CAP თეორემის გათვალისწინებით გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ „112“-ის მონაცემთა ბაზად შეირჩა მონაცემთა დოკუმენტ-ორიენტირებული, არარელაციური ბაზა MongoDB.

- შემუშავებულია კრიტიკული სისტემის ინფრასტრუქტურა, აღწერილია სისტემის გამართვის ეტაპები და დაპროექტებულია მაღალი წვდომადობის მქონე მონაცემთა საცავი.

- მონაცემთა ბაზის სამივე ძირითადი მახასიათებლის (პროცესორი, ოპერატიული მეხსიერება და ხისტი დისკი), ისევე როგორც საიმედოობისა და წვდომადობის გასაუმჯობესებლად გადაწყდა Sharding-ისა და რეპლიკაციის ტექნოლოგიების გამოყენება.

- გაანალიზდა გადაუდებელი სამედიცინო სამსახურის გამოწვევები და განისაზღვრა საკვლევი მიმართულებები კომპიუტერული მეთოდების გამოყენებისათვის. პროგრამირების ენა Python-ის საშუალებით შესრულდა extract, transform, load (ETL) და მონაცემთა წინასწარი დამუშავების (data preprocessing) პროცესები, რის შედეგადაც მიღებულ იქნა მანქანური დასწავლისთვის საჭირო სტრუქტურის მქონე მონაცემთა კრებული ბოლო 5 წლის გადაუდებელი სამედიცინო დახმარების შემთხვევების ბაზაზე.

- Python პროგრამირებისა და Weka ხელსაწყოს გამოყენებით გაანალიზებულ იქნა მიღებული მონაცემთა ნაკრები. გამოვლინდა, წაიშალა და შესწორდა ანომალური შემთხვევები, რაც გაუთვალისწინებლობის დროს უარყოფითად აისახებოდა მანქანური დასწავლის შედეგებზე.

- სასწრაფო სამედიცინო მომსახურების დისპეტჩერიზაციის გასაუმჯობესებლად შეიქმნა გრიპის სეზონის შემთხვევებზე მორგებული კომპიუტერული მოდელი და გამოთვლების საფუძველზე დადგინდა კრიტერიუმები გრიპის თითოეული შემთხვევის გადაუდებლობის დონისა და მომსახურების რიგის გამოსავლენად.

- შექმნილი სისტემის გამოყენებით ობიექტურად დგინდება გრიპის შემთხვევათა მომსახურების რიგი თითოეული მათგანის გადაუდებლობის დონის მიხედვით. ამასთან, აღნიშნული ალგორითმების საშუალებით გამოვლინდა ყველა შემთხვევა, რომლებიც გამოცდილმა ექიმმა გადაუდებლობის მაღალი დონით შეაფასა. შესაძლებელია დისპეტჩერს წარედგინოს კომპიუტერულად განსაზღვრული გადაუდებლობის დონის მიხედვით აგებული შემთხვევათა პრიორიტეტული რიგი, რის საფუძველზეც მას შეუძლია სწრაფად მიიღოს ადეკვატური გადაწყვეტილება სასწრაფო დახმარების ბრიგადების გაგზავნის თაობაზე. ეს ქმნის სასწრაფო სამედიცინო დახმარების ბრიგადების დისპეტჩერიზაციის გაუმჯობესების რეალურ შესაძლებლობას გრიპის სეზონის დროს.

გამოქვეყნებული ლიტერატურა

1. სურგულაძე გ., კვიციანი ნ., კვიციანი გ. კორპორაციული აპლიკაციების აგება დაპროგრამების სერვის-ორიენტირებული ტექნოლოგიით. *სტუ-ს შრ.კრ. „მართვის ავტომატიზებული სისტემები“*. 2016, 1, 21, 215 - 220.
2. სურგულაძე გ., კვიციანი გ. ტრანზაქციის იზოლირების დონეები რელაციურ და არარელაციურ მონაცემთა ბაზებში. *“კომპიუტინგი/ინფორმატიკა, განათლების მეცნიერებები, მასწავლებლის განათლება“ IV საერთაშორისო-სამეცნიერო კონფერენციის მოხსენებათა თეზისები*. 2016, 161 - 168.

3. სურგულაძე გ., კვიციანი გ., კახელი ბ. NoSQL მონაცემთა ბაზების განვითარების პერსპექტივები და პრობლემები მართვის საინფორმაციო სისტემებში. *სტუ-ს შრ.კრ. „მართვის ავტომატიზებული სისტემები“*. 2016, 2, 22, 230 - 239.
4. სურგულაძე გ., კვიციანი გ. შესავალი NoSQL მონაცემთა ბაზებში (MongoDB). თბილისი: სტუ. „IT-კონსალტინგის ცენტრი“, თბილისი, 2017, 151 p.
5. სურგულაძე გ., კვიციანი გ. კონსისტენტური მოდელები რელაციურ და არარელაციურ მონაცემთა ბაზებში. *VIII საერთაშორისო სამეცნიერო და პრაქტიკული კონფერენციის “ინტერნეტი და საზოგადოება” (INSO2017) ნაშრომების კრებული*. 2017, 151 - 155.
6. **Kiviladze G.** Semi automated management of defining the case priority in the flu epidemic season for Emergency Medical Service. *GESJ:Computer Sciences and Telecommunications*. 2018, 1, 53, 101 - 105.
7. ჩოგოვაძე გ., ფრანგიშვილი ა., კვიციანი გ., სურგულაძე გ., ნარეშელაშვილი გ. ინფორმაციული საზოგადოება, მონაცემთა მენეჯმენტის ახალი ტექნოლოგიები და ექსტრემალური სიტუაციების მართვის სისტემები. *სტუ-ს შრ.კრ. „მართვის ავტომატიზებული სისტემები“*. 2018, 1, 25, 7 - 16.

ABSTRACT

Planning high availability data warehouse for Emergency Response Center (“112”)

Dissertation thesis represents the problem of planning high availability data center for Emergency Response Center (“112”). The data saving and processing methods and technologies are discussed. As a result of critical analysis non-relational approaches are chosen, particularly a document-oriented database (MongoDB).

Specifications of relational and non-relational databases are compared. The class of objects is defined, for which the application of non-relational databases is effective. Attention is focused on the issue of integrity of the distributed bases, the ACID principles are discussed. On the base of document-oriented non-relational database MongoDB main characteristics of base productivity are discussed.

The architecture of the critical system was worked out. That makes possible to achieve data integrity, horizontal scaling and sufficiently high availability for the problem. The MongoDB database is already successfully used by the Ministry of Internal Affairs of Georgia at LEPL “112”.

Besides projecting of data center, it is important to perform the analyses and processing of the data. Based on the analysis of the data obtained from co-

operation with "112", the study directions have been determined to solve real problems:

- **EMS brigade selecting.** The aim of the task is improvement of brigade distribution that results in selecting the ambulance car that is most proper for the fastest responding. The general model of Artificial Neural Network was worked out and the stages of problem solving were planned. Programming language Python was used to perform extract, transform, load (ETL) and data preprocessing processes. As a result the dataset structured for machine learning was received. On the next stage the received dataset was analyzed by means of Python programming and Weka toolbox. The anomaly cases that reflected negatively on machine learning results were detected and corrected or eliminated. Currently the research is going on: the machine learning methods are discussed and the results are compared. It is noteworthy that this project was presented to the European Emergency Number Association (EENA) conference on April 25-27, 2018 in Ljubljana (Slovenia).

- **EMS case priority defining. Modeling EMS dispatching in flu-seasons.** The aim of the task is defining priority of cases according to their emergency level that improves EMS service. Highly experienced medics were consulted for choosing the factors describing the state of a patient objectively. The emergency level of each case was estimated by an experienced doctor. Having in mind the possible clinical complications of influenza cases, a special system for processing medical information is worked out. The special parameters are chosen and their proper weights are defined. An original system is applied to process case-card information and supply dispatchers with its result – emergency level of each influenza case, as well as the priority line of cases of the current set. The model set was formed using typical EMS registered influenza cases gathered from the usual case-cards filled by operators of Georgian Emergency and Operative Response Center. The special system of converting medical information into the set of quantitative parameters is presented in the thesis as well as the algorithms developed to compute the emergency index of influenza cases. The research presented in this work shows that the accuracy of revealing the most urgent cases by means of the applied algorithms is the same as that of an experienced medic. The applied system makes it possible to reveal all cases estimated as most urgent by the medic.

The automated system of EMS brigade dispatching should be considered as a real help for dispatchers in flu-seasons.

This research was supported by Shota Rustaveli National Science Foundation (SRNSF) [PhDF2016_219]. That gave me the possibility to hold 2 research visits in Finland (University of Tampere) and Germany (Friedrich-Alexander University

Erlangen-Nürnberg) in 2017.

The fact that foreign colleagues are interested in this work can be explained by the acuteness of the selected topic and its popularity among the informatics researchers. The first results of the research show that the project should be continued. I am ready to work at it after finishing the dissertation.