

# Big Data-nın Proqram Təminatının Testetmə Problemləri

Məkrufə Hacırahimova<sup>1</sup>, Aybəniz Əliyeva<sup>2</sup>

<sup>1,2</sup>İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan

<sup>1</sup>makrufa@science.az, <sup>2</sup>aybeniz63@rambler.ru

**Xülasə—** Məqalədə böyük həcmli verilənlərin emalında istifadə olunan proqram təminatının test edilməsi ilə bağlı problemlər tədqiq olunur. Həmçinin bu problemin həlli ilə bağlı təklif olunmuş modellər nəzərə çatdırılır.

**Açar sözlər—** big data, proqram təminatı, tipik test verilənləri, test proqramı, qrammatik təhlil ağıcı

## I. GİRİŞ

XXI əsrin əvvəllərindən etibarən geniş yayılmağa başlamış Big data (BD) konsepsiyasına müxtəlif yanaşmalar mövcuddur [1]. Lakin onun həcm, müxtəliflik, sürət və dəyər kimi dörd xüsusiyyətinə daha çox önəm verirlər [2]. Verilənlər yığımının həcmi çox yüksək olub, terabayt, petabayt və hətta eksabaytlarla qiymətləndirilir. Müxtəlif növ strukturlaşdırılmış və strukturlaşdırılmamış verilənlərin emalına ehtiyac yaranır. Yüksək sürətlə generasiya olunan verilənlərin emal vaxtı analiz olunması tələb olunur. İnsanlar müxtəlif formatlarda olan verilənlər yığımını intellektual analiz etməklə dəyərli informasiya əldə edə bilirlər. Klassik emal proqramları isə verilənlərin vaxtında və səmərəli analizi üçün yararlı deyildir. BD-nin həcmnin sürətlə artması, müxtəlif struktura malik olması və paylanması verilənlərin yeni emal üsullarının tətbiqini tələb edir.

Hazırda BD texnologiyalarında istifadə olunan və böyük həcmli verilənləri emal etməyə imkan verən bir çox proqramlar hazırlanmışdır [2, 3]. Bəzi hallarda bu proqramların işləmə keyfiyyəti tələblərə cavab vermir. Bu proqramların keyfiyyətinin test edilməsi isə çox çətinidir.

BD-nin proqram təminatı arxitekturasının miqyaslanması son nəticədə onun test edilməsi ilə bağlı problemlərin yaranmasına səbəb olur [4]. Bu problemlərin araşdırılması olduqca vacib məsələlərdəndir.

## II. BİG DATA PROQRAM TƏMİNATININ TESTETMƏ PROBLEMLƏRİ

Böyük həcmli verilənlər müxtəlif veb-servislərdən istifadə olunmaqla ETL (Extract, Transform, and Load) prosesləri vasitəsilə hesablanır, saxlanır və analiz olunur. Xüsusi halda verilənlərin emalı üçün Hadoop platformasından istifadə olunur. Bu isə yüz petabaytlarla verilənləri ənənvi üsullardan daha sürətli emal etməyə imkan verir [5]. Tipik ETL prosesində verilənlər orijinal verilənlər mənbəyindən (məsələn, MySQL verilənlər bazası) əldə edilir, sorğu və analizin tələblərinə cavab verməsi üçün strukturlaşdırılmış formata çevrilir. Sonra isə müştərilərin istifadəsi üçün verilənlər bir

hədəf mənbəyə (məsələn, PostgreSQL verilənlər bazası) yüklənir.

Bununla belə, BD-nin emalı üçün nəzərdə tutulmuş proqram təminatının keyfiyyətinin test edilməsi mürəkkəb və çətin məsələdir. Belə ki, BD-nin emalında istifadə olunan ETL proqram təminatının test edilməsi zamanı üç texniki problem meydana çıxır [6].

Birincisi, müxtəlif tədqiqatlardan, sahələrdən və qurğulardan daxil olan böyük həcmdə verilənlərin emalına çox vaxt sərf olunur, digər texnologiyalardan istifadə olunduqda belə, petabaytlarla verilənlərin emalı günlər və hətta, həftələr tələb edir. Müxtəlif mənbələr və müştərilər üçün kiçik və tipik (representative) verilənlərin yaradılması çətin məsələdir. Müştəri verilənlərinin bütövlükdə və ya qismən istifadəsi (istismarı) bütün emal prosesinin sürətinə təsir edir. Dövlət, biznes, sorğu, statistik paylanma və s. üçün xarakterik olan məhdudiyyətlərdən (constraints) istifadə etməklə böyük populyasiyanı təmsil edən kiçik verilənlər yığımının istifadəsi problemin həllində mühüm rol oynaya bilər.

İkincisi, ETL prosesi zamanı verilənlər bir çox serverlər arasında ötürülür və müxtəlif keçid nöqtələrində kökündən dəyişilir. Hər bir keçid nöqtəsində verilənləri yoxlamaq lazımdır?

Üçüncüsü, ötürülən və dəyişilən verilənləri necə yoxlamalı? Böyük həcmdə verilənlərin əl ilə yoxlanması olduqca çətin, ona görə də onu avtomatlaşdırmaq lazım gəlir.

## III. TEST PLATFORMASI

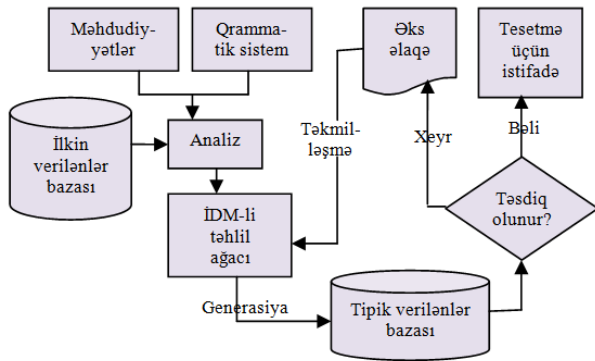
Yuxarıda qeyd olunan hər üç texniki problemin həlli üçün tipik verilənlər yığımının yaradılması, verilənlərin ötürülməsi və dəyişikliyi yoxlamaq üçün BD-nin miqyaslanan test proqramı təklif olunmuşdur [6]. Test proqramının məqsədi böyük həcmli ilkin verilənlərdən tipik verilənlər yığımını yaratmaqdır. Hər iki verilənlər yığımı eyni və ya müxtəlif servislərdə saxlanıla bilər. Tipik verilənlər yığımı verilənlərin ötürülməsini və dəyişikliyi yoxlamaq üçün istifadə oluna bilər. Testetmə proqramı ötürmə öncəsi və ötürmədən sonrakı verilənləri müqayisə etməklə verilənlərin ötürüldüyünü təsdiqləyir. Analoji olaraq proqram həmçinin verilənlərin dəyişikliyi müəyyən edən tələblərə uyğun olaraq onların necə dəyişdiyini yoxlayır. Test platforması hazırda inkişaf etdirilir və prototip ümidverici nəticələr göstərmişdir.

#### A. Test verilənlərinin generasiyası

Test verilənlərinin yaradılması sahəsində müəyyən ilkin metodlar irəli sürülmüşdür. Məsələn, [7]-də təklif edilmiş model verilənlər bazasından tipik verilənlər yığımı yaradır. Lakin burada tipik verilənlər yığımının ölçüsünə necə nəzarət etmək məsələsinə baxılmamışdır. Bundan əlavə, yalnız bazanın verilənləri üçün nəzərdə tutmuşdur.

BD-nin miqyaslanan test proqramı test verilənlərinin generasiyası üçün fərqli model təklif edir (şəkil 1.). Burada test verilənləri yığımının ölçüsünə nəzarət etmək üçün giriş fəzasının bölünməsindən istifadə edilir [8]. Giriş fəzasının bölmələrinin test edilməsi “giriş-sahə modeli” (*input-domain model* - IDM) ilə başlayır. Tester IDM-i ayırır, bölünmüş bloklardan test qiymətlərini seçir və testlərin generasiyası üçün lazımı meyarları tətbiq edir. Burada ilk dəfə olaraq BD kontekstində tipik verilənlər yığımının yaradılması üçün fəza bölmələrinin test edilməsindən istifadə olunur.

Verilənlərdə baş verən dəyişiklikləri əks etdirmək üçün verilənlər bazasından fərqli digər struktur verilənlərindən (auditlərdən) istifadə olunur. IDM-dən istifadə etməklə auditlər çoxluğundan tipik verilənlər yığımını yaratmaq üçün auditin hər bir atributuna (əlaməti) uyğun test qiymətləri çıxarılır. Bu auditlərin strukturunu təsvir etmək üçün qrammatik təhlil sistemi (*grammar*) yazılmışdır. Sonra atributlar da daxil olmaqla bütün test qiymətlərini toplamaq üçün test proqramı auditlərin əksinə (qarşı) olan qrammatik sistemləri təhlil edir (qrammatik), test qiymətlərinin statistik paylanmasını hesablayır və nəticədə qrammatik təhlil ağacı (a parse tree) yaradılır. Test qiymətləri və məhdudiyətlər analiz olunduqdan sonra, hər bir atribut üçün IDM yaradılır və qrammatik təhlil ağacına birləşdirilir. Son tipik verilənlər audit atributları, onların münasibətləri, məhdudiyətləri və IDM-ə malik qrammatik təhlil ağacından generasiya olunur. Tipik verilənlər yığımı istifadə olunur və təkrar istifadə vasitəsilə qiymətləndirilir [5].



Şəkil 1. Tipik test verilənlərin generasiyası

Beləliklə test proqramı giriş fəzasının bölmələrinin test edilməsindən istifadə edərək orijinal böyük verilənlər yığımından kiçik və tipik verilənlər yığımı yaradır. Testetmə üçün bu verilənlər yığımından istifadə edilməsi sürətli proseslər zamanı onların fasiləsiz inteqrasiya və ötürülməsinə mane olmur. Həmçinin testetmə proqramı verilənlərin ötürüldüyü və dəyişikliyə uğradığı müxtəlif keçid nöqtələrində onların girişini və yoxlanmasını təmin edir.

#### B. Verilənlərin yoxlanması

ETL prosesləri zamanı verilənlərin tamlığına əmin olmaq üçün onların ötürülməsini və dəyişikliyi yoxlamaq lazım gəlir.

Verilənlərin ötürülməsini yoxlamaq nisbətən sadədir. Çünki məlum məsələdir ki, gözlənilən qiymət ilkin qiymətə bərabər olacaqdır. Əgər verilənlər bir verilənlər bazasından digərinə ötürülsə, sütunların, sətirlərin sayını, sütunların adını və verilənlərin tipini yoxlamaqla ilkin və hədəf (*target*) verilənlərini az bir zamanda yoxlamaq olar. İlkin və hədəf verilənləri təqdim edildikdə yoxlama avtomatlaşdırıla bilər [6].

Verilənlərin ötürülməsinin yoxlanılmasından fərqli olaraq, onların dəyişikliyinə yoxlanılması daha mürəkkəbdir. Məsələn, on ilkin cədvəldəki verilənləri bir hədəf cədvəlində cəmləmək lazım gəlir. Bu halda hədəf cədvəlinin bəzi sütunlarının istifadə etdiyi verilənlərin tipi ilkin mənbə cədvəldəki verilənlərlə eyni ola bilər, digər sütunlar isə müxtəlif tipli verilənlərdən istifadə edə bilər.

Verilənlərin ötürülməsinin yoxlanılması üçün iki layihə təklif edilir [6]. Birinci layihə hədəf verilənlərinin düzgün verilənlər tipinə və yüksək qiymət aralıqlarına sahib olub-olmadığını yoxlayır. Test proqramı tələblərə uyğun olaraq verilənlərin tipini və qiymət aralıqlarını aşkar edir, sonra isə hədəf verilənlərinin yoxlanması üçün test yaradır.

İkinci layihə hər bir transformasiya qaydasının dəqiqliyinin yoxlanılması üçün ətraflı spesifikasiyanı verir. Test platforması hədəf verilənlərinin transformasiyasının dəqiqliyini qiymətləndirmək üçün ilkin mənbə verilənlərini hədəf verilənləri ilə müqayisə edir.

Hər iki layihə transformasiya spesifikasiyanın test proqramının oxuya biləcəyi bir formatda yazılmasını tələb edir. Sonra platforma avtomatik spesifikasiyanı təhlil edir və transformasiyanı yoxlamaq üçün testlər yaradır.

Beləliklə, ilkin və hədəf verilənlərinin yoxlanılması həyata keçirilir.

#### NƏTİCƏ

Yuxarıda qeyd olunan problemlər bütün BD istifadəçilərinin qarşılaşdığı mühüm problemlərdir. Gələcəkdə BD-nin proqram təminatının test edilməsinin keyfiyyətinin yaxşılaşdırılması layihəçilərin, proqramçıların qarşısında duran əsas məsələlərdəndir. Verilənlərin generasiyasını sürətləndirmək üçün paralel hesablamalardan və Hadoop analitik proqram platformalarından, maşın təlimi üsullarından istifadə edilməsi planlaşdırılır [6].

BD-nin müxtəlif texnologiyalarından, üsullarından və proqram vasitələrindən istifadə etməklə bu problemin həllində müəyyən uğurlar əldə etmək olar.

ƏDƏBİYYAT

- [1] R.M. Əliquliyev, M. Ş. Hacırahimova, "Big Data" fenomeni: problemlər və imkanlar, *İnformasiya texnologiyaları problemləri*, 2014, №2, səh. 3-16.
- [2] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *Access, IEEE*, 2014, vol. 2, pp. 652–687.
- [3] R.M. Əliquliyev, M. Ş. Hacırahimova, A.S. Əliyeva, "Big data-nin aktual elmi-nəzəri problemləri", *İnformasiya cəmiyyəti problemləri*, 2016, №2, səh. 37–49.
- [4] NESSI – Software Engineering White Paper: Software engineering for and with Big Data, 2014, [http://www.nessi-europe.eu/Files/Private/NESSI\\_SE\\_WhitePaper-FINAL.pdf](http://www.nessi-europe.eu/Files/Private/NESSI_SE_WhitePaper-FINAL.pdf)
- [5] R.M.Əliquliyev, R.M.Əliquliyev, M.Sh.Hajirahimova, "Big Data integration architectural concepts for oil and gas industry", *IEEE 10th International Conference on Application of Information and Communication Technologies*, 2016, pp. 20-24.
- [6] N. Li, A. Escalona, Y. Guo; J. Offutt. "A Scalable Big Data Test Framework", 2015, *IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, <http://ieeexplore.ieee.org/abstract/document/7102619/>
- [7] A. Alexandrov, C. Brucke, V. Markl, "Issues in big data testing and benchmarking," *Sixth International Workshop on Testing Database Systems*, ser. DBTEST '13, New York, NY, USA, 2013.
- [8] P. Ammann, Offutt, *Introduction to Software Testing*. Cambridge, UK: Cambridge University Press, 2008, <http://www.cse.hcmut.edu.vn/~hiep/KiemthuPhanmem/Tailieuthamkhao/Introduction%20to%20Software%20Testing.pdf>