MA’LUMOTLARNI TAHLIL QILISHDA MAPREDUCE VA APACHE SPARK TEXNOLOGIYALAR TAHLILI

Abdullayeva Umida Farhadovna
E-mail: umidafarhadovna11_07g@mail.ru

Received: 19.09.2025
Revised: 20.09.2025
Accepted: 21.09.2025

Kalit so‘zlar:
ma’lumotlarni tahlil qilish, MapReduce, Apache Spark, katta ma’lumotlar, taqsimlangan hisoblash, ma’lumotlarni qayta ishlash, real vaqt tahlili.

Annotatsiya. Ushbu maqolada katta hajmdagi ma’lumotlarni tahlil qilishda qo‘llaniladigan MapReduce va Apache Spark texnologiyalarining taqqoslash tahlili olib borilgan. Har ikki texnologiya katta ma’lumotlarni taqsimlangan muhitda qayta ishlash imkoniyatini beradi, ammo ular orasida ishlash tezligi, samaradorlik va qo‘llanilish sohalari bo‘yicha sezilarli farqlar mavjud. Tadqiqotda MapReduce va Apache Sparkning arxitekturasi, ishlash prinsiplari, afzalliklari va kamchiliklari batafsil ko‘rib chiqildi. Shuningdek, amaliy misollar orqali ularning katta ma’lumotlarni qayta ishlashdagi samaradorligi taqqoslandi. Natijalar shuni ko‘rsatadiki, Apache Spark real vaqt rejimida ma’lumotlarni tahlil qilish va mashinali o‘qitish jarayonlari uchun ko‘proq qulayliklar yaratadi, shu bilan birga MapReduce murakkab hisoblash vazifalarida barqaror ishlashni ta’minlaydi

Ключевые слова:
анализ данных, MapReduce, Apache Spark, большие данные, распределённые вычисления, обработка данных, анализ в реальном времени.

Аннотация. В данной статье проведён сравнительный анализ технологий MapReduce и Apache Spark, используемых для анализа больших объёмов данных. Обе технологии позволяют обрабатывать большие данные в распределённой среде, однако между ними существуют значительные различия по скорости обработки, эффективности и областям применения. В исследовании подробно рассмотрены архитектура, принципы работы, преимущества и недостатки MapReduce и Apache Spark. Также, с помощью практических примеров, сравнивалась их эффективность в обработке больших данных. Результаты показывают, что Apache Spark

2025
WWW.INSIGHTPUBLISHING.UZ

обеспечивает больше удобств для анализа данных в режиме реального времени и в процессах машинного обучения, в то время как MapReduce демонстрирует стабильную работу при выполнении сложных вычислительных задач

Keywords:
Data analysis, MapReduce, Apache Spark, big data, distributed computing, data processing, real-time analytics.

Annotation. This article presents a comparative analysis of the technologies MapReduce and Apache Spark, which are used for analyzing large volumes of data. Both technologies enable the processing of big data in a distributed environment; however, there are significant differences between them in terms of processing speed, efficiency, and application areas. The study thoroughly examines the architecture, operational principles, advantages, and disadvantages of MapReduce and Apache Spark. Additionally, their efficiency in processing large datasets is compared using practical examples. The results show that Apache Spark offers greater convenience for real-time data analysis and machine learning processes, while MapReduce provides stable performance for complex computational tasks

Kirish. Hozirgi vaqtda an'anaviy dasturiy usullar va apparat vositalaridan foydalangan holda katta ma'lumotlarni qayta ishlash mutlaqo mantiqsiz va ko'pincha imkonsizdir, chunki mavjud bo'lgan juda ko'p ma'lumotlar bunga yo'l qo'ymaydi. Biroq, bunday operatsiyalarni amalga oshirish uchun hali ham bir qancha samarali usullar mavjud. Ushbu maqolada kata hajmli ma’lumotlarni qayta ishlashdagi asosiy muammolar va yechimlar ko’rib chiqilgan. Hozirgi kunda “big data” larni qayta ishlovchi, tahlil qiluvchi bir qancha texnologiyalar va algoritmlar mavjud, bu yerda asosan MapReduce arxitekturasi va Apache Spark texnologiyasi avzalliklari va kamchiliklari ko’rib chiqilgan, tahlil qilingan va xulosalar berilgan, natijalar esa jadval ko’rinishida ko’rsatilgan.

Katta hajmli ma'lumotlar yoki katta ma'lumotlarning ta'rifi. Katta ma'lumotlar hajmi yuz terabayt va petabaytdan ortiq bo'lishi mumkin bo'lgan ma'lumotlarni anglatadi. Bundan tashqari, ushbu ma'lumotlar muntazam ravishda yangilanadi. Masalan, aloqa markazlaridan, ijtimoiy tarmoqlardan, birja savdosi ma'lumotlaridan va hokazolardan olingan ma'lumotlar kiradi. Shuningdek, "katta hajmli ma'lumotlar" tushunchasi ba'zan ularni qayta ishlash usullarini o'z ichiga oladi. Agar terminologiya haqida gapiradigan bo'lsak, unda "Katta ma'lumotlar" nafaqat ma'lumotlarni, balki katta hajmli ma'lumotlarni qayta ishlash

tamoyillarini, ulardan keyingi foydalanish imkoniyatini, katta massivlarda ma'lum bir ma'lumot blokini aniqlash tartibini anglatadi.[2] Bunday jarayonlar bilan bog'liq savollar o'z dolzarbligini yo'qotmaydi. Ularning yechimi ko'p yillar davomida turli xil ma'lumotlarni ishlab chiqaradigan va to'playdigan tizimlar uchun muhimdir. Katta hajmli ma’lumotlar asosan quyidagi asosiy xususiyatlarga ega bo’ladi.[5]

• Volume (hajmi) - taxminan 1 Petabayt va undan yuqori.

• Velocity (tezlik) - yuqori tezlikda ma'lumotlarni yaratish, qabul qilish va qayta ishlash.

• Variety (turli) - ma'lumotlarning geterogenligi, turli formatlar va mumkin bo'lgan strukturaning yetishmasligi.

Ko'pincha ushbu parametrlarga yana ikkita omil qo'shiladi:

• Variability (o'zgaruvchanlik) - qayta ishlash usullarini tanlashga ta'sir qiluvchi daromadning turli intensivligi.

• Value (ahamiyat) - olingan ma'lumotlarning murakkablik darajasidagi farq.[1]

MapReduce - bu klasterni tashkil etuvchi ko'p sonli kompyuterlar ("tugunlar" deb ataladi) yordamida taqsimlangan vazifalar to'plamini hisoblash uchun freymvork.

MapReduce ishi ikki bosqichdan iborat: Map va Reduce, xuddi shu nomdagi yuqori darajadagi funksiyalar nomi bilan atalgan.

Map bosqichi kiritilgan ma'lumotlarni oldindan qayta ishlaydi. Buning uchun kompyuterlardan biri (asosiy tugun - asosiy tugun deb ataladi) topshiriqning kirish ma'lumotlarini oladi, uni qismlarga bo'ladi va oldindan ishlov berish uchun boshqa kompyuterlarga (ishchi tugunlar - ishchi tugun) uzatadi.

Reduce bosqichida oldindan ishlangan ma'lumotlar kamayadi. Asosiy tugun ishchi tugunlardan javoblarni oladi va ular asosida natijani yaratadi - dastlab tuzilgan muammoning yechimi.

MapReduce ning afzalligi shundaki, u oldindan qayta ishlash va qisqartirish operatsiyalarini taqsimlangan holda bajarish imkonini beradi. Oldindan ishlov berish operatsiyalari bir-biridan mustaqil ravishda ishlaydi va parallel ravishda amalga oshirilishi mumkin (garchi amalda bu kirish manbai va/yoki ishlatiladigan protsessorlar soni bilan cheklanadi). Xuddi shunday, bir nechta ishchi tugunlar yig'ishni amalga oshirishi mumkin - bu faqat bitta kalit qiymatiga ega bo'lgan barcha dastlabki ishlov berish natijalarini bir vaqtning o'zida bitta ishchi tugun tomonidan qayta ishlanishini talab qiladi. Garchi bu jarayon ketma-ket algoritmlarga qaraganda samarasizroq bo'lsa-da, MapReduce ko'p sonli serverlar tomonidan qayta ishlanishi mumkin bo'lgan katta hajmdagi ma'lumotlarga qo'llanilishi mumkin. Misol uchun, MapReduce bir necha soat

ichida petabayt ma'lumotlarni saralash uchun ishlatilishi mumkin. Parallelizm shuningdek qisman server nosozliklarini tiklashni ta'minlaydi: agar oldindan ishlov berish yoki qisqartirish operatsiyasini bajaruvchi ishchi tugun muvaffaqiyatsizlikka uchrasa, uning ishi boshqa ishchi tugunga o'tkazilishi mumkin (agar bajarilayotgan operatsiya uchun kirish ma'lumotlari mavjud bo'lsa).

Freymvork ko'p jihatdan Map ga asoslangan va funktsional dasturlashda keng qo'llaniladigan funktsiyalarni qisqartiradi[2], garchi freymvorkning haqiqiy semantikasi prototipdan farq qiladi[3].

```
// Map bosqichida ishchi tugunlar tomonidan ishlatiladigan funksiya
// kirish oqimidan kalit-qiymat juftlarini qayta ishlash uchun
void map(String name, String document):
    // kiruvchi ma’lumotlar:
    //   name – hujjat nomi
    //   document – hujjat mazmuni
    for each word in document:
        EmitIntermediate(word, "1");

// Ishchi tugunlari tomonidan Reduce bosqichida foydalaniladigan
funksiya
// Map bosqichida olingan kalit-qiymat juftlarini qayta ishlash
void reduce(Iterator partialCounts):
    // Kiruvchi ma’lumotlar:
    //   partialCounts – guruhlangan oraliq natijalar ro’yxati.
    //     talab qilinadigan qiymat
    int result = 0;
    for each v in partialCounts:
        result += parseInt(v);
    Emit(AsString(result));
```

Ushbu kodda, **Map** bosqichida har bir hujjat so'zlarga bo'linadi va juftliklar qaytariladi, bu yerda kalit so'zning o'zi va qiymat "1" dir. Agar hujjatda bir xil so'z bir necha marta uchrasa, u holda ushbu hujjatni dastlabki qayta ishlash natijasida ushbu so'z qancha marta sodir bo'lsa, shuncha juftlik bo'ladi. Yaratilgan juftliklar keyingi ishlov berish uchun yuboriladi, tizim ularni kalit bo'yicha guruhlaydi (bu holda kalit so'zning o'zi) va ularni bir nechta protsessorlar orasida taqsimlaydi. Guruhdagi bir xil kalitga ega bo'lgan ob'ektlar to'plami ma'lumotlar oqimini qayta ishlovchi, uning hajmini kamaytiradigan **Reduce** funksiyasining kirishiga kiradi. Ushbu misolda kamaytirish funksiyasi berilgan so'zning butun oqim bo'ylab takrorlanishini qo'shib beradi va natija - faqat bitta summa - chiqish sifatida yuboriladi.

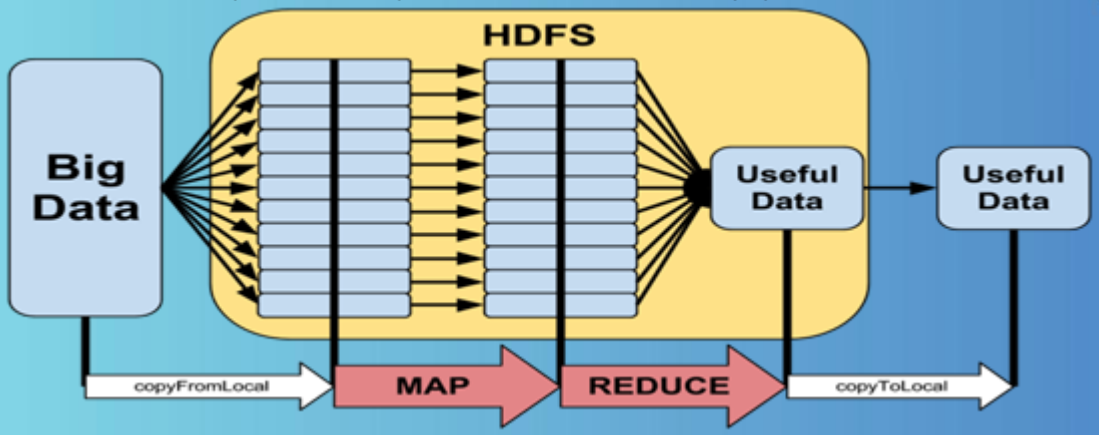
MapReduce va Apache Spark o'rtasidagi farqlar

1. MapReduce - bu kompyuter klasterida parallel, taqsimlangan algoritmga ega bo'lgan katta ma'lumotlar to'plamlarini qayta ishlash va yaratish uchun dasturlash mexanizmi. MapReduce bir nechta komponentlardan iborat, jumladan:

2025
WWW.INGCONF.UZ

- JobTracker - Klasterdagi barcha ishlar va resurslarni boshqaradigan asosiy tugun.
- TaskTrackers - Xaritani ishga tushirish va vazifalarni kamaytirish uchun klasterdagi har bir mashinaga joylashtirilgan agentlar.

JobHistoryServer - Bajarilgan ishlarni kuzatib boruvchi komponent va odatda alohida funksiya sifatida yoki JobTracker bilan joylashtiriladi.



1-rasm. MapReduce arxitekturasi.

Apache Hadoop ochiq kodli dasturiy ta'minot tizimi bo'lib, bitta serverdan minglab mashinalargacha o'lchash va an'anaviy uskunaga ega klasterlarda ilovalarni ishga tushirish uchun mo'ljallangan. Apache Hadoop platformasi ikki darajaga bo'lingan.

• Hadoop taqsimlangan fayl tizimi (HDFS)

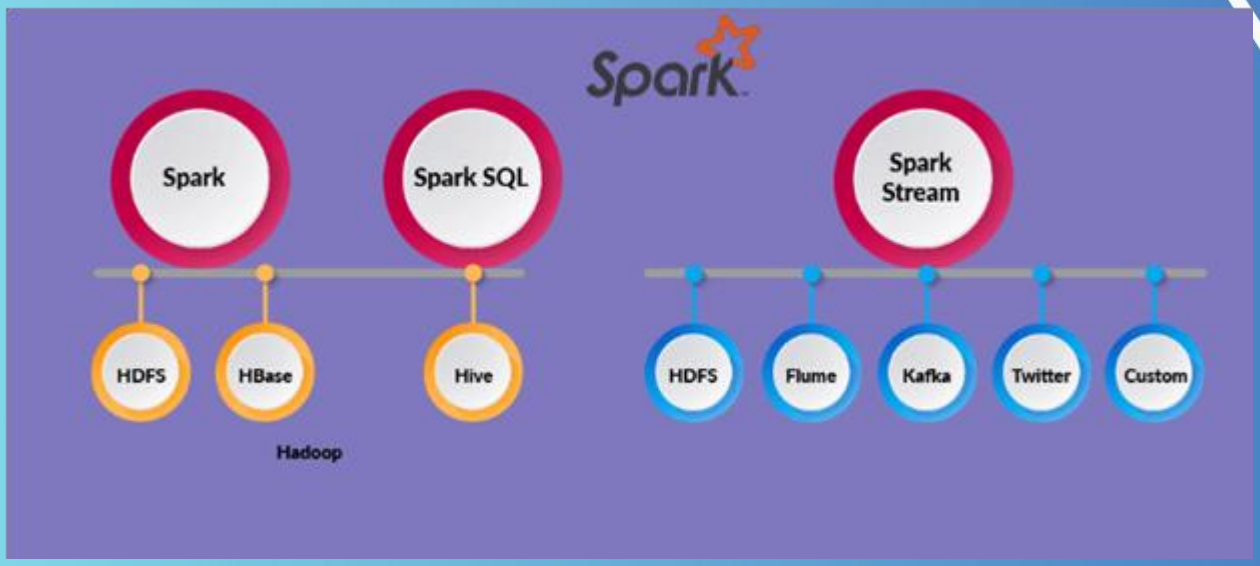
• Qayta ishlash qatlami (MapReduce)

Hadoop saqlash qatlami, ya'ni HDFS ma'lumotlarni saqlash uchun javobgardir va MapReduce Hadoop klasterida ma'lumotlarni qayta ishlash uchun javobgardir. MapReduce - bu Hadoop klasteridagi yuzlab yoki minglab serverlar uchun kengayishni ta'minlovchi dasturlash paradigmasi. MapReduce Java dasturlash tiliga asoslangan taqsimlangan hisoblash uchun ishlov berish texnikasi va dasturlash modelidir.[8] MapReduce - bu Hadoop taqsimlangan fayl tizimida (HDFS) saqlangan Hadoop klasterida tuzilgan yoki tuzilmagan ma'lumotlarning katta taqsimlangan to'plamlarini qayta ishlash uchun kuchli ramka. MapReduce-ning kuchli xususiyatlari uning kengaytirilishidir.

2. Apache Spark - bu katta hajmdagi ma'lumotlarni qayta ishlashda tezkor hisoblash uchun mo'ljallangan klaster texnologiyasi. Apache Spark - bu taqsimlangan ishlov berish mexanizmi, lekin u o'rnatilgan klaster resurslari menejeri va taqsimlangan saqlash tizimi bilan birga kelmaydi. Siz tanlagan klaster menejeri va saqlash tizimini ulashingiz kerak. Apache Spark Spark yadrosi va Hadoop uchun mavjud kutubxonalarga o'xshash kutubxonalar to'plamidan iborat. Yadro taqsimlangan ijro mexanizmi va tillar to'plamidir. Apache Spark tarqatilgan ilovalarni ishlab chiqish uchun Java, Scala, Python va

2025
WWW.INSIGHTPUBLISHING.UZ

R kabi tillarni qo'llab-quvvatlaydi. Oqim, SQL, grafika va mashinani o'rganishdan foydalanadigan ish yuklarini qo'llab-quvvatlash uchun Spark yadrosi ustiga qo'shimcha kutubxonalar qurilgan. Apache Spark - bu SQL so'rovlari, grafik va mashinalarni o'rganishni o'z ichiga olgan ommaviy va oqim uchun ma'lumotlarni qayta ishlash mexanizmi. Apache Spark mustaqil ravishda, shuningdek, Hadoop YARN klaster menejerida ishlashi mumkin va shu bilan u mavjud Hadoop ma'lumotlarini o'qiy oladi.



2-rasm. Apache Spark texnologiyasi

MapReduce va Apache Spark o'rtasidagi asosiy farq

- MapReduce qat'iy diskka asoslangan, Apache Spark esa tezkor xotiradan foydalanadi va ishlov berish uchun diskdan foydalanishi mumkin.
- MapReduce va Apache Spark ma'lumotlar turlari va ma'lumotlar manbalari nuqtai nazaridan bir xil muvofiqlikka ega.
- MapReduce va Spark o'rtasidagi asosiy farq shundaki, MapReduce doimiy xotiradan foydalanadi, Spark esa doimiy taqsimlangan ma'lumotlar to'plamidan foydalanadi.
- Hadoop MapReduce xotiraga sig'maydigan ma'lumotlar uchun mo'ljallangan, Apache Spark esa xotiraga sig'maydigan ma'lumotlar uchun, ayniqsa, ajratilgan klasterlarda yaxshiroq ishlashga ega.
- Hadoop MapReduce tejamkor variant bo'lishi mumkin, chunki Hadoop xizmatdir va Apache Spark yuqori xotira mavjudligi tufayli tejamkor hisoblanadi.

Apache Spark va Hadoop MapReduce ikkalasi ham nosozliklarga chidamli, ammo nisbatan Hadoop MapReduce Spark dan ko‘ra ko‘proq nosozliklarga chidamli.

- Hadoop MapReduce asosiy Java dasturlash ko'nikmalarini talab qiladi, Apache Spark dasturlash esa interaktiv rejimga ega bo'lgani uchun osonroq.

Spark ommaviy ishlov berish ishlarini MapReduce-ga qaraganda 10-100 marta tezroq bajarishi mumkin, ammo ikkala vosita ham kata hajmli ma'lumotlarni qayta ishlash uchun ishlatiladi.

MapReduce va Apache Spark taqqoslash jadvali

	MapReduce	**Apache Spark**
Ma'lumotlarni qayta ishlash	Faqat ommaviy qayta ishlash uchun	Ommaviy ishlov berish, shuningdek, real vaqtda ma'lumotlarni qayta ishlash
Qayta ishlash tezligi	Apache Spark'dan sekinroq, sababi agar diskda kirish/chiqish kechikishi mavjud bo'lsa	Xotirada 100 baravar tezroq va diskda 10 baravar tezroq amal bajaradi
kategoriya	Ma'lumotlarni qayta ishlash mexanizmi	Data Analytics Engine
Xarajatlar	Apache Spark bilan solishtirganda arzonroq	Katta hajmdagi RAM tufayli nisbatan qimmatroq
Masshtablilik	Ikkala yondashuvda ham bitta klasterda 1000 ta tugun bilan cheklangan	Ikkala yondashuvda ham bitta klasterda 1000 ta tugun bilan cheklangan
Mashinali o’qitish	Mashinali o'qitish bilan integratsiyalashganda MapReduce Apache Mahout bilan ko'proq mos keladi	Apache Spark o'rnatilgan mashinali o'qitish API-lariga ega
Moslik	Barcha ma'lumotlar manbalari va fayl formatlari bilan mos keladi	Apache Spark barcha ma'lumotlar manbalari va Hadoop klasteri tomonidan qo'llab-quvvatlanadigan fayl formatlari bilan birlashishi mumkin.
Xavfsizlik	MapReduce ramkasi Apache Spark bilan solishtirganda xavfsizroq	Apache Spark’dagi xavfsizlik xususiyati rivojlanmoqda va yanada yetuk bo‘lib bormoqda.
rejalashtiruvchi	Tashqi rejalashtiruvchiga bog'liq	Apache Spark o'zining rejalashtiruvchisiga ega
xatolarga chidamlilik	Xatolarga chidamlilik uchun replikatsiyadan foydalanish	Apache Spark nosozliklarga chidamlilik uchun RDD va

2025
WWW.INSIGHTPUBLISHING.UZ

		boshqa saqlash modellaridan foydalanadi
Foydalanish qulayligi	MapReduce JAVA API tufayli Apache Spark bilan solishtirganda biroz murakkab	API-lar tufayli Apache Sparkdan foydalanish osonroq
Dublikatsiyani bartaraf etish	MapReduce bu xususiyatni qo'llab-quvvatlamaydi	Apache Spark har bir kirishni aynan bir marta qayta ishlaydi, shuning uchun takrorlanishni yo'q qiladi.
Tilni qo'llab-quvvatlash	Asosiy til Java, lekin C, C++, Ruby, Python, Perl, Groovy kabi tillar ham qo'llab-quvvatlanadi.	Apache Spark Java, Scala, Python va R-ni qo'llab-quvvatlaydi
Kechikish	Juda yuqori kechikish	MapReduce Framework bilan solishtirganda tezroq
murakkablik	Kodlarni yozish va disk raskadrovka qilish qiyin	Yozish va disk raskadrovka qilish oson
Apache	Ma'lumotlarni qayta ishlash uchun ochiq kodli ramka	Yuqori tezlikda ma'lumotlarni qayta ishlash uchun ochiq kodli ramka
kodlash	kod qatorlari nisbatan ko'proq	kod qatorlari nisbatan kamroq
interaktiv rejim	Interaktiv emas	Interaktiv
infratuzilma	O’rta darajadagi apparat	O'rta va yuqori darajadagi apparat
SQL	Hive so'rovlar tilini qo'llab-quvvatlaydi	Spark SQL ni qo'llab-quvvatlaydi

MapReduce va Apache Spark

MapReduce va Apache Spark katta ma'lumotlarni qayta ishlash uchun eng muhim vositadir. MapReducening asosiy afzalligi shundaki, uni bir nechta hisoblash tugunlari bo'ylab ma'lumotlarni qayta ishlash uchun osongina masshtablash mumkin, Apache Spark esa MapReduce-ni mukammal ravishda to'ldiradigan yuqori tezlikdagi hisoblash, moslashuvchanlik va nisbatan qulay foydalanishni taklif qiladi. MapReduce va Apache Spark bir-biri bilan simbiotik munosabatlarga ega. Hadoop tarqatilgan fayl tizimi kabi Spark taqdim etmaydigan xususiyatlarni taqdim etadi, Spark esa unga muhtoj bo'lgan

2025
WWW.INSIGHTPUBLISHING.UZ

ma'lumotlar to'plamlari uchun real vaqt rejimida ishlov berishni ta'minlaydi. MapReduce diskda, Apache Spark esa xotirada ishlaydi. MapReduce va Apache Spark birgalikda katta hajmli ma'lumotlarni qayta ishlash, tahlil qilish va Hadoop klasterini yanada ishonchli qilish uchun kuchli vositalar hisoblanadi.

Xulosa. Ushbu maqolada MapReduce va Apache Spark texnologiyalarining katta ma’lumotlarni tahlil qilishdagi afzalliklari va kamchiliklari tahlil qilindi. MapReduce an’anaviy taqsimlangan hisoblash modeli sifatida katta hajmdagi ma’lumotlarni ishonchli qayta ishlashni ta’minlasa-da, u real vaqt rejimida ishlashda cheklovlarga ega. Apache Spark esa yuqori tezlikda ma’lumotlarni qayta ishlash, xususan, real vaqt tahlili va mashinali o‘qitish jarayonlarida samaradorlikni oshirish imkonini beradi. Tadqiqot natijalari shuni ko‘rsatdiki, zamonaviy katta ma’lumotlar muhitida Apache Spark texnologiyasi ko‘proq mos keladi, ammo murakkab va barqaror hisoblash talab qilinadigan vazifalarda MapReduce hali ham samarali vosita sifatida qolmoqda. Kelajakda ushbu texnologiyalarni birgalikda qo‘llash va ularni yanada optimallashtirish istiqbollari mavjud.

FOYDALANILGAN ADABIYOTLAR

1. Abdullayev A.A., Tursunov B.T. (2021). Katta hajmdagi ma’lumotlarni qayta ishlashda bulutli texnologiyalarning roli. – TATU ilmiy axborotnomasi, №2, 45–50-betlar.
2. Usmonov I.I. (2020). Spark texnologiyasidan foydalangan holda katta ma’lumotlarni parallel tahlil qilish. – Axborot texnologiyalari va kompyuter muhandisligi jurnali, Toshkent axborot texnologiyalari universiteti, №4, 31–35-betlar.
3. Ergashev M.M., Rasulov D.R. (2022). MapReduce asosida taqsimlangan hisoblash modellari va ularning amaliy qo‘llanilishi. – Informatika va raqamli texnologiyalar, 1(1), 55–62-betlar.
4. Normurodov Sh.N. (2023). Apache Spark muhitida mashinali o‘qitish algoritmlarining optimallashtirilgan ishlatilishi. – TATU Termiz filiali ilmiy maqolalar to‘plami, №3, 17–23-betlar.
5. Rasulova Z.X., Karimov A.B. (2021). Katta ma’lumotlar bilan ishlashda samarali platformalarni taqqoslash. – Yosh olimlar ilmiy jurnali, 2(5), 40–46-betlar.
6. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1), 107-113.
7. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... & Zaharia, M. (2016). MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, 17(34), 1-7.