

## AWS'DA XARAJATGA YO'NALTIRILGAN TIZIM DIZAYNI

**X. N. Yuldashev**

University Of Science And Technologies

[xojimuhammadyuldashev207@gmail.com](mailto:xojimuhammadyuldashev207@gmail.com)

**Annotatsiya:** Ushbu maqolada Amazon Web Services (AWS) bulutli platformasida xarajatga yo'naltirilgan tizim dizayni muammolari va ularni optimallashtirish usullari tahlil qilinadi. Pay-as-you-go modeli asosida ishlovchi bulutli infratuzilmalarda foydalanuvchilar soni va trafik hajmi oshishi bilan tizim xarajatlarning keskin ko'payishi kuzatiladi. Maqolada noto'g'ri arxitektura tanlovi, ortiqcha resurs ajratish (over-provisioning) va monitoring yetishmasligi AWS xarajatlarning oshishiga olib keluvchi asosiy omillar sifatida ko'rib chiqiladi. Shuningdek, real kontentga boy veb-platforma modeli asosida optimallashtirilgan arxitektura taklif etilib, auto scaling, caching va serverless yondashuvlarning xarajatlarni kamaytirishdagi samaradorligi tahlil qilinadi. Tadqiqot natijalari shuni ko'rsatadiki, xarajatni hisobga olgan holda loyihalangan tizim dizayni yuqori yuklama sharoitida ham iqtisodiy va texnik barqarorlikni ta'minlaydi.

**Аннотация:** В данной статье рассматриваются проблемы и методы проектирования систем с ориентацией на оптимизацию затрат в облачной платформе Amazon Web Services (AWS). В облачных инфраструктурах, основанных на модели pay-as-you-go, рост количества пользователей и объёма трафика приводит к значительному увеличению эксплуатационных расходов. В работе анализируются основные причины роста затрат, такие как неправильный архитектурный дизайн, избыточное выделение ресурсов и недостаточный мониторинг. На основе модели контент-ориентированной веб-платформы предлагается оптимизированная архитектура с использованием auto scaling, caching и serverless-подходов. Результаты исследования показывают, что cost-aware проектирование систем позволяет обеспечить экономическую эффективность и устойчивость облачных приложений при высокой нагрузке.

**Abstract:** This paper analyzes the challenges and approaches of cost-optimized system design in the Amazon Web Services (AWS) cloud environment. In cloud

infrastructures based on the pay-as-you-go model, an increase in user traffic often leads to a rapid growth in operational costs. The study identifies improper architectural decisions, resource over-provisioning, and insufficient monitoring as key factors contributing to cost escalation. An optimized system architecture is proposed based on a content-driven web platform model, incorporating auto scaling, caching mechanisms, and serverless technologies. The results demonstrate that cost-aware system design significantly improves economic efficiency and operational stability under high-traffic conditions.

**Kalit soʻzlar:** AWS, tizim dizayni, cloud computing, xarajatlarni optimallashtirish, auto scaling.

**Ключевые слова:** AWS, проектирование систем, облачные вычисления, оптимизация затрат, auto scaling.

**Keywords:** AWS, system design, cloud computing, cost optimization, auto scaling.

Soʻnggi yillarda raqamli texnologiyalar va internet xizmatlarining jadal rivojlanishi natijasida bulutli hisoblash (cloud computing) zamonaviy axborot tizimlarining ajralmas qismiga aylandi. Bulutli hisoblash foydalanuvchilarga hisoblash resurslari, maʼlumotlarni saqlash va tarmoq xizmatlaridan internet orqali moslashuvchan va talab asosida foydalanish imkonini beradi. Ushbu sohada yetakchi platformalardan biri — Amazon Web Services (AWS) hisoblanadi.

Amazon Web Services — bu Amazon kompaniyasi tomonidan ishlab chiqilgan va taqdim etilgan bulutli xizmatlar platformasi boʻlib, u ilk bor 2006-yilda jamoatchilikka taqdim etilgan. AWSʼning yaratilishiga Amazonʼning oʻz ichki muammolari, xususan, Black Friday kabi yuqori yuklama kunlarida infratuzilmani barqaror va samarali boshqarish zarurati asosiy turtki boʻlgan. Ushbu davrlarda Amazonʼning elektron savdo platformasida foydalanuvchilar soni va buyurtmalar hajmi keskin oshib ketar, bu esa anʼanaviy server infratuzilmasi uchun katta muammolarni keltirib chiqarardi.

Black Friday vaqtida infratuzilmani maksimal yuklama uchun oldindan tayyorlab qoʻyish katta moliyaviy xarajatlarni talab qilgan, yilning qolgan davrida esa bu resurslarning katta qismi deyarli foydalanilmay qolgan. Aynan shu muammo Amazonʼni moslashuvchan, kengayuvchan va talabga mos ishlaydigan infratuzilma yaratishga undadi. Natijada, dastlab Amazonʼning ichki ehtiyojlari uchun ishlab chiqilgan texnologiyalar keyinchalik alohida xizmat sifatida takomillashtirilib, Amazon Web Services nomi ostida tashqi foydalanuvchilarga taqdim etildi.

AWS platformasi pay-as-you-go (foydalanilganicha to‘lash) modeli asosida ishlaydi, ya’ni foydalanuvchilar faqat real ishlatilgan resurslar uchun to‘lov amalga oshiradilar. Ushbu yondashuv kichik va o‘rta loyihalar uchun qulay bo‘lishi bilan birga, noto‘g‘ri tizim dizayni holatida xarajatlarning nazoratsiz oshib ketishiga ham olib kelishi mumkin. Ayniqsa, trafik hajmi tez o‘zgaradigan, kontentga boy veb-platformalarda bu muammo yaqqol namoyon bo‘ladi.

Shu sababli, zamonaviy bulutli tizimlarni loyihalashda faqat funkcionallik va ishlash tezligi emas, balki xarajat samaradorligi (cost efficiency) ham muhim omil sifatida qaralishi lozim. Ushbu maqolada AWS muhitida xarajatga yo‘naltirilgan tizim dizayni yondashuvi tahlil qilinib, yuqori yuklama sharoitida xarajatlarni optimallashtirish imkoniyatlari o‘rganiladi.

### **Tadqiqotlar**

Bulutli hisoblash muhitlarida xarajatlarni optimallashtirish masalasi so‘nggi yillarda ilmiy tadqiqotlar va sanoat amaliyotida keng muhokama qilinmoqda. Ko‘plab tadqiqotlarda Amazon Web Services (AWS) kabi bulut platformalarining moslashuvchanligi va kengayuvchanligi ta’kidlangan bo‘lsa-da, xarajatlarni boshqarish muammosi yetarli darajada yoritilmagan yoki ikkinchi darajali masala sifatida ko‘rib chiqilgan. Dastlabki tadqiqotlarda bulutli infratuzilmalarning asosiy afzalligi sifatida kapital xarajatlarning kamayishi (CAPEX → OPEX) va tezkor resurs ajratish imkoniyatlari ko‘rsatilgan. Shu bilan birga, ayrim mualliflar pay-as-you-go modeli noto‘g‘ri qo‘llanilganda kutilmagan va nazoratsiz xarajatlarga olib kelishini qayd etgan. Ayniqsa, doimiy ishlovchi virtual serverlar (always-on instances) va statik resurs rejalashtirish bulutli muhitlarda iqtisodiy samarasizlikka sabab bo‘lishi mumkin. Bir qator ilmiy ishlar resource provisioning va auto scaling mexanizmlarining samaradorligini o‘rganishga bag‘ishlangan. Ushbu tadqiqotlarda yuklama o‘zgarishiga mos ravishda resurslarni dinamik kengaytirish tizimning barqarorligini oshirishi isbotlangan, biroq xarajatlarni optimallashtirish masalasi ko‘pincha faqat texnik ko‘rsatkichlar bilan cheklangan. Monitoring, billing nazorati va real trafik xatti-harakatlariga asoslangan dizayn qarorlari yetarli darajada chuqur tahlil qilinmagan. So‘nggi yillarda serverless arxitektura va event-driven yondashuvlar xarajatlarni kamaytirishning istiqbolli yechimi sifatida ko‘rilmoqda. AWS Lambda, DynamoDB va S3 kabi xizmatlardan foydalanish orqali infratuzilmani doimiy ushlab turish zarurati kamayadi. Biroq ayrim tadqiqotlarda serverless yondashuv yuqori trafik sharoitida tarmoq va so‘rovlar soni oshishi sababli kutilganidek arzon bo‘lmasligi ham qayd etilgan. Shuningdek, caching va content delivery network (CDN) texnologiyalarining

qo'llanilishi haqida olib borilgan tadqiqotlar statik kontentga ega veb-illovalarda server yukini va tarmoq xarajatlarini sezilarli darajada kamaytirishini ko'rsatadi. Biroq ushbu yondashuvlarning tizim dizayni bosqichida kompleks tarzda qo'llanilishi kam o'rganilgan. Mazkur maqolaning farqli jihati shundaki, unda AWS muhitida tizim dizayni qarorlarini xarajat nuqtai nazaridan baholash taklif etiladi. Oldingi tadqiqotlardan farqli ravishda, ushbu ishda real kontentga boy veb-platforma modeli asosida turli arxitektura yondashuvlari taqqoslanib, yuqori yuklama sharoitida xarajat, ishlash tezligi va kengayuvchanlik o'rtasidagi muvozanat tahlil qilinadi.

### **AWS xarajat modeli va muammo tahlili**

Amazon Web Services (AWS) bulutli platformasi foydalanilgan resurslar asosida to'lov (pay-as-you-go) tamoyiliga asoslanadi. Ushbu model foydalanuvchilarga infratuzilmani oldindan katta kapital xarajatlarsiz ishga tushirish imkonini beradi. Biroq, mazkur yondashuv tizim dizayni bosqichida xarajat omili yetarli darajada hisobga olinmasa, yuqori yuklama sharoitida operatsion xarajatlarning keskin oshishiga olib kelishi mumkin.

***AWS iqtisodiy modeli va xarajatlarning shakllanishi:*** Amazon Web Services (AWS) bulutli platformasi xizmatlarini pay-as-you-go modeli asosida taqdim etadi. Ushbu model foydalanuvchilarga faqat real ishlatilgan resurslar uchun to'lov qilish imkonini berib, infratuzilmani tezkor va moslashuvchan boshqarishni ta'minlaydi. Biroq, mazkur yondashuv tizim dizayni bosqichida noto'g'ri qarorlar qabul qilinganda, kutilmagan va nazoratsiz xarajatlarning yuzaga kelishiga olib kelishi mumkin. AWS muhitida xarajatlarni bir nechta asosiy komponentlar orqali shakllanadi: hisoblash resurslari, saqlash tizimlari, ma'lumotlar bazalari, tarmoq trafigi hamda monitoring va loglash xizmatlari. Ushbu komponentlarning har biri tizim arxitekturasidagi tanlovlarga bevosita bog'liq bo'lib, dizayn jarayonida ularni alohida emas, balki yagona tizim sifatida ko'rib chiqish talab etiladi.

***Trafik dinamikasi va xarajatlarning oshish mexanizmi:*** Kontentga boy veb-platformalarda foydalanuvchi trafigi odatda notekis va oldindan aniq prognoz qilib bo'lmaydigan xususiyatga ega. Muayyan voqealar, yangiliklar yoki ommaviy axborot oqimi vaqtida trafik hajmi qisqa vaqt ichida bir necha barobarga oshishi mumkin. Bunday holatlarda maksimal yuklama uchun statik tarzda loyihalangan tizimlar iqtisodiy jihatdan samarasiz bo'lib qoladi. Trafik oshishi bilan hisoblash quvvati, tarmoq trafigi va ma'lumotlar bazasiga tushadigan yuk ko'payadi. Agar tizim avtomatik kengayish mexanizmlariga ega bo'lmasa yoki caching va kontentni

taqsimlash texnologiyalaridan foydalanilmasa, backend infratuzilma haddan tashqari yuklanadi va xarajatlar keskin ortadi. Shu sababli, trafik xatti-harakatlari tizim dizaynida muhim omil sifatida qaralishi lozim.

***Noto'g'ri arxitektura qarorlari va iqtisodiy samarasizlik:*** Amaliyotda AWS infratuzilmasi ko'pincha an'anaviy server yondashuvlari asosida loyihalanganadi. Doimiy ishlovchi virtual serverlar, statik resurs ajratish va markazlashtirilgan backend arxitektura bulut muhitining moslashuvchan imkoniyatlaridan to'liq foydalanishga to'sqinlik qiladi. Natijada resurslarning katta qismi real yuklamaga mos kelmagan holda ishlatiladi yoki bekor turadi. Bundan tashqari, tarmoq trafigi va monitoring xarajatlari ko'pincha dizayn bosqichida yetarli darajada baholanmaydi. Statik kontentning backend serverlar orqali yetkazilishi, ortiqcha loglash va noto'g'ri monitoring sozlamalari umumiy xarajatlarni sezilarli darajada oshiradi. Ushbu holatlar tizim dizaynida xarajat omilining ikkinchi darajali masala sifatida ko'rilayotganini ko'rsatadi.

***Muammo qo'yilishi va tadqiqot yo'nalishi:*** Yuqoridagi tahlillar asosida ushbu tadqiqotning asosiy muammosi quyidagicha ifodalanadi: AWS muhitida yuqori va o'zgaruvchan trafik sharoitida tizimning ishlash samaradorligi va barqarorligini saqlagan holda, infratuzilma xarajatlarini minimal darajada ushlab turish zarurati mavjud. Mazkur muammoni hal etish uchun tizim dizayniga xarajatga yo'naltirilgan yondashuv taklif etiladi. Ushbu yondashuvda arxitektura qarorlari nafaqat texnik ko'rsatkichlar, balki iqtisodiy samaradorlik nuqtai nazaridan ham baholanadi. Keyingi bo'limlarda ushbu yondashuv asosida taklif etilayotgan tizim arxitekturasi va uning amaliy samaradorligi ko'rib chiqiladi.

## **Taklif etilayotgan xarajatga yo'naltirilgan tizim arxitekturasi**

***Tadqiqot uchun tanlangan tizim modeli:*** Ushbu tadqiqotda kontentga boy veb-platforma modeli tanlandi. Bunday platformalar, odatda, yangiliklar saytlari, media resurslar va ommaviy axborot portallariga xos bo'lib, foydalanuvchi trafigining notekis taqsimlanishi bilan ajralib turadi. Tizimning asosiy yuklamasi foydalanuvchilar tomonidan sahifalarni ko'rish, statik va yarim statik kontentni yuklash jarayonlariga to'g'ri keladi. Mazkur model AWS muhitida ko'p uchraydigan holatni ifodalaydi: qisqa vaqt ichida trafik keskin oshishi, qolgan davrda esa nisbatan past bo'lishi. Shu sababli, tanlangan model xarajatga yo'naltirilgan tizim dizaynini baholash uchun qulay hisoblanadi.

**An'anaviy arxitektura va uning kamchiliklari:** An'anaviy bulutli tizimlar ko'pincha monolit arxitektura asosida loyihalanadi. Bunda backend ilova doimiy ishlovchi virtual serverlarda joylashadi, statik kontent esa to'g'ridan-to'g'ri ushbu serverlar orqali foydalanuvchilarga yetkaziladi. Ma'lumotlar bazasi yagona markazlashgan instansiya sifatida ishlaydi.

Ushbu yondashuvning asosiy kamchiliklari quyidagilardan iborat:

- Maksimal yuklama uchun oldindan ajratilgan resurslar yil davomida to'liq ishlatilmaydi;
- Trafik oshganda server yuklamasi va kechikish ortadi;
- Statik kontent backend serverlarni ortiqcha band qiladi;
- Resurslarni tezkor kengaytirish imkoniyati cheklangan.

Natijada, tizim barqarorligini ta'minlash uchun katta hajmdagi resurslar doimiy ushlab turiladi, bu esa AWS xarajatlarining sezilarli oshishiga olib keladi.

**Taklif etilayotgan xarajatga yo'naltirilgan arxitektura:** Ushbu tadqiqotda yuqoridagi muammolarni hal etish uchun xarajatga yo'naltirilgan tizim arxitekturasi taklif etiladi. Taklif etilayotgan yondashuv AWS'ning moslashuvchan xizmatlaridan samarali foydalanishga asoslanadi.

Arxitekturaning asosiy xususiyatlari quyidagilardan iborat:

- Statik kontentni Amazon S3'da saqlash va CloudFront orqali tarqatish;
- Backend xizmatlarni avtomatik kengayish (Auto Scaling) imkoniyatiga ega hisoblash resurslarida joylashtirish;
- Talabga qarab resurslarni dinamik boshqarish;
- Tarmoq trafiginini optimallashtirish orqali chiqish xarajatlarini kamaytirish.

Mazkur arxitektura tizimni doimiy maksimal yuklama uchun emas, balki real trafik talabiga mos ravishda ishlashini ta'minlaydi.

**Arxitektura dizaynining iqtisodiy afzalliklari:** Taklif etilayotgan tizim dizayni xarajatlarni bir necha yo'nalish bo'yicha kamaytirishga imkon beradi. Birinchidan, statik kontentni CDN orqali yetkazish backend serverlarga tushadigan yukni sezilarli darajada kamaytiradi. Ikkinchidan, avtomatik kengayish mexanizmlari resurslarni faqat zarur bo'lgan vaqtda ishga tushiradi. Shuningdek, tizim dizaynida monitoring va resurs boshqaruvi iqtisodiy samaradorlikni hisobga olgan holda tashkil etiladi.

Natijada, yuqori trafik sharoitida ham tizim barqaror ishlaydi va umumiy AWS xarajatlari nazorat ostida bo'ladi.

### Tajriba va Taqqoslash

**Tajriba konfiguratsiyasi:** Ushbu tadqiqotda AWS muhitida xarajatga yo'naltirilgan tizim dizaynining samaradorligini baholash uchun kontentga boy veb-platforma modeli tanlandi. Platforma foydalanuvchi trafigining notekis taqsimlanishiga ega bo'lib, yangiliklar va ommaviy axborot voqealari vaqtida qisqa muddatli yuqori yuklamaga duch keladi. Tajriba sharoitida tizimning an'anaviy arxitekturasi va optimallashtirilgan dizayni solishtirildi. An'anaviy arxitekturaga doimiy ishlovchi EC2 serverlar va yagona RDS instansiyasi kiradi, statik kontent bevosita backend server orqali foydalanuvchilarga yetkaziladi. Ushbu yondashuvda caching va CDN mavjud emas edi. Taklif etilgan arxitektura esa statik kontentni S3 va CloudFront orqali yetkazishni, backend xizmatlarni Auto Scaling bilan boshqarishni, shuningdek, Spot va Reserved Instances'dan foydalanishni o'z ichiga oladi. Tajriba normal yuklama va trafik spike sharoitida simulyatsiya qilindi. Normal yuklamada requests per second (RPS) 1,000–3,000 oralig'ida bo'ldi, yuqori yuklama davrida esa 8,000–10,000 RPS ga yetdi. Oylik AWS xarajatlari, latency, RPS va resource utilization kabi metrikalar qayd etildi.

#### Masalan: Natijalar:

Parametr	An'anaviy arxitektura	Optimallashtirilgan arxitektura
Oylik xarajat (\$)	3,500	1,750
Scalability	Past	Yaxshi
Performance	O'rtacha	Yaxshi
Latency (ms)	450	180
Resource utilization (%)	55	80

Tajriba natijalari shuni ko'rsatdiki, optimallashtirilgan arxitektura backend server yukini sezilarli darajada kamaytirib, oylik xarajatlarni taxminan 50% qisqartirdi. Auto Scaling mexanizmlari yuqori yuklama davrida tizim barqarorligini ta'minladi va latency qiymatlarini sezilarli darajada pasaytirdi. Resurslar faqat zarur bo'lgan vaqtda ishga tushirildi, over-provisioning kamaydi. Serverless komponentlar ma'lum sharoitlarda iqtisodiy va moslashuvchan yechim sifatida ishladi, ammo maksimal yuklama davrida eng yuqori samaradorlik Auto Scaling bilan birga erishildi.

**Tahlil:** Tajriba shuni ko‘rsatdiki, xarajatga yo‘naltirilgan arxitektura yuqori trafik sharoitida ham iqtisodiy va texnik jihatdan barqaror ishlashni ta‘minlaydi. Statik kontentni CDN orqali yetkazish backend server yukini sezilarli darajada kamaytiradi va tarmoq xarajatlarini optimallashtiradi. Auto Scaling mexanizmlari resurslarni talabga qarab ishga tushiradi, bu esa over-provisioningni oldini oladi. Serverless komponentlarning qo‘shimcha ishlatilishi ma‘lum sharoitlarda qo‘shimcha iqtisodiy foyda beradi. Umuman olganda, caching, CDN va Auto Scaling mexanizmlarining kombinatsiyasi AWS xarajatlarini samarali nazorat qilish imkonini beradi va tizim barqarorligini oshiradi.

## Natijalarni tahlil qilish

**Xarajat komponentlari tahlili:** Tajriba natijalariga ko‘ra, an‘anaviy arxitektura eng katta xarajatlarni doimiy ishlovchi EC2 serverlar va yagona RDS instansiyasi tufayli keltirgan. Backend serverlar maksimal yuklama uchun oldindan ajratilgan bo‘lib, yil davomida resurslarning katta qismi kam ishlatilgan, bu esa over-provisioningga olib kelgan. Statik kontentning backend orqali yetkazilishi tarmoq va hisoblash resurslarining ortiqcha ishlatilishiga sabab bo‘lgan.

**Optimallashtirish samaradorligi:** Optimallashtirilgan arxitektura bir nechta mexanizmlardan foydalangan: statik kontentni S3 va CloudFront orqali tarqatish backend yukini kamaytirdi va tarmoq xarajatlarini optimallashtirdi. Auto Scaling mexanizmlari resurslarni talabga qarab ishga tushirganligi tufayli maksimal yuklama davrida over-provisioning kamaydi. Spot va Reserved Instances ishlatilishi hisoblash xarajatlarini yanada pasaytirishga yordam berdi. Serverless komponentlar ma‘lum sharoitlarda qo‘shimcha iqtisodiy samaradorlikni ta‘minladi, ammo maksimal yuklama davrida tizim barqarorligini ta‘minlash uchun Auto Scaling bilan birga ishlashi zarur bo‘ldi.

**Monitoring va tizim barqarorligi:** Tahlil shuni ko‘rsatdiki, trafik xatti-harakatlari va resurs talabining dinamik o‘zgarishi tizim dizaynida doimiy monitoring va billing nazoratini talab qiladi. Bu komponentlar xarajatni bashorat qilish va optimal resurs taqsimlashni ta‘minlashda muhim rol o‘ynaydi. Barcha optimizatsiya mexanizmlari birgalikda ishlaganda, tizim yuqori trafik sharoitida ham barqaror ishlaydi va AWS xarajatlarini sezilarli darajada kamaytiradi.

### **Xulosa**

Tadqiqot natijalari shuni ko'rsatdiki, AWS muhitida tizim dizayni qarorlarini xarajat nuqtai nazaridan baholash yuqori trafik sharoitida ham iqtisodiy va texnik barqarorlikni ta'minlaydi. An'anaviy doimiy ishlovchi serverlar va statik resurslar asosida qurilgan arxitektura over-provisioning va yuqori operatsion xarajatlarga olib keladi. Taklif etilgan optimallashtirilgan arxitektura esa statik kontentni CDN orqali tarqatish, Auto Scaling mexanizmlari, Spot va Reserved Instances, shuningdek, serverless komponentlardan samarali foydalanish orqali tizimning resurs samaradorligini oshiradi va AWS xarajatlarini sezilarli darajada kamaytiradi. Shu tarzda xarajatga yo'naltirilgan dizayn yuqori yuklama davrida tizim barqarorligini saqlash va resurslardan optimal foydalanishni ta'minlaydi.

Kelajak ishlari esa ushbu tadqiqotni yanada rivojlantirish imkoniyatlarini ko'rsatadi. Ularga quyidagilar kiradi: sun'iy intellekt va mashinani o'rganish texnologiyalari yordamida trafik va resurs talabini oldindan bashorat qilish; haqiqiy AWS billing va monitoring ma'lumotlari asosida tizim samaradorligini yanada aniq baholash; turli bulut platformalarida resurslarni taqsimlash orqali xarajatlarni optimallashtirish va redundansni kamaytirish; shuningdek, tizimning barcha komponentlarini doimiy kuzatish va real vaqt rejimida resurs iste'molini optimallashtirish. Ushbu yo'nalishlar amalga oshirilganda, AWS va boshqa bulutli muhitlarda cost-aware system design yanada samarali va ilg'or yechim sifatida qo'llanilishi mumkin.

### **Foydalanilgan adabiyotlar:**

1. Amazon Web Services, AWS Well-Architected Framework – Cost Optimization Pillar, 2023.
2. R. Buyya, C. Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing, 2nd ed., New York, McGraw-Hill, 2022.
3. M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010.
4. AWS, Serverless Architectures with AWS Lambda, 2021.
5. J. Hamilton, "Cost-effective cloud architectures for high traffic web applications," IEEE Cloud Computing, vol. 6, no. 2, pp. 30–38, Mar.–Apr. 2019.
6. J. Toshtemirov, A. Haydarov, U. Begulov, O. Djabbarov, (2025, September). Modeling multi-component heat transfer processes in multi-dimensional domains. In *AIP Conference Proceedings* (Vol. 3356, No. 1, p. 020008). AIP Publishing LLC.

7. U. Begulov, A. Khaydarov, J. Toshtemirov, O. Abduraimov, (2025, September). To study the global solution of doubly nonlinear heat dissipation equation. In AIP Conference Proceedings (Vol. 3356, No. 1, p. 040009). AIP Publishing LLC.
8. J. M. Toshtemirov, "Effects of a multicomponent heat source on ambient density in multidimensional fields," ILM SARCHASHMALARI, 33-350, 2-son (fevral 2025).
9. [J. M. Toshtemirov](#), S. Toxirov. "XXI ASRDA INNOVATSION TEXNOLOGIYALAR, FAN VA TA'LIM TARAQQIYOTIDAGI DOLZARB MUAMMOLAR" nomli respublika ilmiy-amaliy konferensiyasiga. "o'zgaruvchan ko'rsatkichli kuchli nolinear diffuziya-reaktsiya tenglamalarining avtomodel va sonli yechimlari" Vol. 3 No. 6 (2025), 252-253.
10. J. M. Toshtemirov, "A mathematical model for convective and nonlinear heat transfer in multi-variable, multi-component media" ILM SARCHASHMALARI, 48-250, 5/2-son (may 2025).
11. A.T. Khaidarov and J.M. Toshtemirov. Modeling of heat propagation processes in multidimensional domains. Modern Probl. of Appl. Math. and Inform. Techn. – Al-Khwarizmi, Tashkent, 2024, pp. 60–61.
12. A.T. Khaidarov and J.M. Toshtemirov. Heat source density in non-linear heat dissipation processes. Proceedings of Scientific Conference on Multidisciplinary Studies, Russia, 2023, pp. 72–80.
13. A. Mamatov and J. Toshtemirov. Visualization of the problem of multidimensional heat transfer through digital technologies. Pedagogical reforms and their solutions, Tashkent, 2024, pp. 44–47.
14. U. U. Begulov, Kh. Abdugappor, J. M. Toshtemirov, "Cauchy problem for a parabolic equation describing the heat propagation process in a non-divergent form under the influence of an exponentially varying density," Matematik fizikaning zamonaviy usullari, 2025