



KESHLASH

Yusupov Mirsaid Abdulaziz o'g'li

Farg'ona davlat universiteti amaliy matematikavainformatika

kafedrası katta o'qituvchisi

mirsaidbeky@gmail.com

Nurmatova Hushnozabonu To'chiboy qizi

Farg'ona davlat universiteti 2-bosqich talabasi

nurmatovaxushnozabonu@gmail.com

Anotatsiya: Ushbu maqolada axborotni tezkor va samarali qayta ishlashni ta'minlaydigan muhim texnologiyalardan biri — keshlash (caching) tizimi tahlil qilinadi. Keshlash texnologiyasining asosiy tushunchalari, ishlash prinsipi, afzalliklari, turlari (masalan, brauzer keshi, server keshi, tarmoq keshi) hamda zamonaviy veb-ilovalar va operatsion tizimlardagi qo'llanilishi batafsil yoritilgan.

Kalit so'zlar: keshlash, cache, xotira, veb-ilova, ma'lumotlar bazasi, Redis, Memcached, ASP.NET Core, tezkorlik, samaradorlik, server keshi, brauzer keshi, CDN, TTL, cache invalidation

Abstract : This article analyzes one of the key technologies that ensures fast and efficient data processing — caching systems. The main concepts of caching technology, its working principles, advantages, types (such as browser cache, server cache, and network cache), as well as its application in modern web applications and operating systems are thoroughly discussed.

Keywords: caching, cache, memory, web application, database, Redis, Memcached, ASP.NET Core, performance, efficiency, server cache, browser cache, CDN, TTL, cache invalidation

Аннотация: В данной статье рассматривается одна из ключевых технологий, обеспечивающих быструю и эффективную обработку данных —



система кэширования. Подробно изложены основные понятия технологии кэширования, принципы её работы, преимущества, типы (такие как кэш браузера, серверный кэш и сетевой кэш), а также применение в современных веб-приложениях и операционных системах.

Ключевые слова: кэширование, кэш, память, веб-приложение, база данных, Redis, Memcached, ASP.NET Core, производительность, эффективность, серверный кэш, кэш браузера, CDN, TTL, аннулирование кэша.

Axborot texnologiyalari jadal rivojlanayotgan hozirgi davrda ma'lumotlarga tezkor va ishonchli tarzda murojaat qilish har qachongidan ham muhim hisoblanadi. Zamonaviy dasturiy ta'minotlar va veb-illovalar katta hajmdagi ma'lumotlar bilan ishlashga majbur bo'lib, foydalanuvchilarning talablariga javob bera oladigan darajada tez ishlashi zarur. Ana shunday sharoitda tizimlarning samaradorligini oshirish, tarmoqli yuklamani kamaytirish va foydalanuvchi tajribasini yaxshilash maqsadida qo'llaniladigan asosiy mexanizmlardan biri bu — keshlash (inglizcha “caching”) texnologiyasidir.

Keshlash — bu ma'lumotlarni vaqtincha xotirada saqlab turish orqali keyingi chaqiruvlarda ularga tezkor kirishni ta'minlovchi usuldir. U ma'lumotlar bazasi, server yoki tarmoq manbasidan qayta-qayta ma'lumot olish o'rniga, avvalgi chaqiruv natijasini saqlab, qayta ishlatish imkonini beradi. Bu esa tizimning ishlash tezligini oshirish, javob vaqtini qisqartirish va server resurslaridan oqilona foydalanishga olib keladi. Ayniqsa, ko'p foydalanuvchili tizimlarda keshlash samarali ishlashi uchun muhim bo'lgan texnologiyalardan biridir. Ushbu maqolada keshlash texnologiyasining mohiyati, ishlash prinsipi, asosiy turlari va amaliyotda qanday ishlatilishi haqida batafsil ma'lumot beriladi. Shuningdek, keshlashning zamonaviy dasturiy arxitekturalarda tutgan o'ri, uning afzalliklari va to'g'ri



konfiguratsiya qilinmaganda yuzaga kelishi mumkin bo'lgan muammolar ham tahlil qilinadi.

Keshlash tizimining samarali ishlashi uchun, avvalo, uning asosiy tushunchalarini va qanday ishlashini to'g'ri tushunish zarur. Keshlash — bu tezkor xotirada (RAM, SSD yoki boshqa muvaqqat saqlash qurilmasida) ma'lumotlarni vaqtincha saqlash orqali keyingi murojaatlarda ularni tezroq olish imkonini beruvchi jarayon hisoblanadi. Ma'lumot ilk bor server yoki asosiy manbadan olinganida, u keshlanadi — ya'ni vaqtinchalik saqlanadi. Keyingi murojaatda, agar ma'lumot o'zgarmagan bo'lsa va keshda mavjud bo'lsa, tizim uni to'g'ridan-to'g'ri keshdan olib beradi. Bu nafaqat ishlash tezligini oshiradi, balki resurslardan foydalanishni kamaytiradi.

Keshlash jarayoni odatda quyidagicha ishlaydi: foydalanuvchi ma'lum bir so'rov yuboradi, bu so'rov dastlab keshda bor-yo'qligi tekshiriladi. Agar ma'lumot keshda mavjud bo'lsa, u "cache hit" deb ataladi va ma'lumot keshdan olinadi. Agar mavjud bo'lmasa — ya'ni "cache miss" holati bo'lsa — tizim asl manbaga murojaat qiladi, kerakli ma'lumotni oladi va uni keshga yozadi, shundan so'ng foydalanuvchiga yetkazadi. Bu jarayon ma'lumotlar bilan ishlashda ko'p marta takrorlanadigan so'rovlar uchun ayniqsa foydalidir.

Keshlash mexanizmlarida yana bir muhim tushuncha — bu TTL (Time To Live) yoki yashash muddati. Bu parametr keshlangan ma'lumot qancha vaqtgacha yaroqli ekanligini belgilaydi. TTL tugaganidan so'ng, keshlangan ma'lumot eskirgan (stale) hisoblanadi va tizim uni yangilaydi yoki yangisiga almashtiradi. Bundan tashqari, "cache invalidation" — ya'ni eskirgan keshni bekor qilish va yangilash mexanizmi ham mavjud bo'lib, u tizimdagi o'zgarishlar aniqlanganda ishga tushadi. Keshlash konsepsiyasi operatsion tizimlardan tortib to veb-brauzerlar, tarmoq uskunalari, ma'lumotlar bazalari va veb-serverlargacha bo'lgan ko'plab tizimlarda qo'llaniladi. Uning asosiy maqsadi — ma'lumotlarga murojaat qilishda yuqori tezlikni ta'minlash, javob vaqtini qisqartirish



Keshlash texnologiyasi turli tizimlar va darajalarda qo'llanilishi mumkin bo'lgan ko'p qirrali jarayon bo'lib, uning bir nechta asosiy turlari mavjud. Har bir tur o'zining vazifasi, joylashuvi va maqsadiga ko'ra farqlanadi. Eng keng tarqalgan turlardan biri bu brauzer keshi bo'lib, foydalanuvchining qurilmasidagi brauzerda ishlaydi. Brauzer keshi sahifadagi rasm, skript, stil fayllari kabi statik elementlarni vaqtincha saqlab, foydalanuvchi bir sahifaga qayta kirganida ularni yangidan yuklamasdan, to'g'ridan-to'g'ri lokal xotiradan olib beradi. Bu esa sahifa tez yuklanishiga va tarmoqqa yuklamaning kamayishiga olib keladi.

Server keshi esa server tomonida amalga oshiriladi va ko'p foydalanuvchilar uchun bir xil bo'lgan ma'lumotlarni saqlashga xizmat qiladi. Masalan, statistik ma'lumotlar, so'rov natijalari yoki foydalanuvchi profillarining tez-tez ishlatiladigan qismlari server keshida saqlanishi mumkin. Bu foydalanuvchi so'rovlariga tez javob berish imkonini yaratadi va server resurslarining tejab ishlatilishini ta'minlaydi. Server keshlari odatda markazlashtirilgan bo'lib, Redis yoki Memcached kabi maxsus keshlash tizimlari yordamida tashkil etiladi.

Tarmoq keshi, yoki boshqa nomi bilan CDN (Content Delivery Network) keshi esa kontentni foydalanuvchiga yaqin serverlarda saqlab, global miqyosda tarqatish imkonini beradi. Masalan, biror media fayl yoki katta hajmdagi rasm fayli foydalanuvchiga geografik jihatdan eng yaqin CDN tugunidan yetkaziladi, bu esa yuklanish vaqtini sezilarli darajada kamaytiradi. Tarmoq keshi ayniqsa ommaviy axborot saytlarida, video platformalarda va xalqaro miqyosda xizmat ko'rsatuvchi ilovalarda katta ahamiyat kasb etadi. Shuningdek, dasturiy darajadagi mahalliy keshlash (local application cache) ham mavjud bo'lib, bu foydalanuvchining qurilmasi yoki ilova ichida amalga oshiriladi. Bunday keshlash turida ma'lumotlar yoki hisoblash natijalari mahalliy xotirada vaqtincha saqlanib turadi, bu esa foydalanuvchi harakati tezligini oshiradi va ortiqcha hisob-kitoblarni kamaytiradi. Xullas, har bir keshlash turi o'ziga xos texnik yechimga ega bo'lib, uni to'g'ri tanlash va qo'llash tizim samaradorligini oshirishda muhim rol o'ynaydi. Bugungi



kunda keshlash texnologiyalari keng ko‘lamda qo‘llanilib, turli sohalarda ishlatiladigan kuchli vositalardan biriga aylangan. Mashhur keshlash texnologiyalari orasida Redis va Memcached ayniqsa ajralib turadi. Redis — bu ochiq manbali, tezkor, kalit-qiyamat (key-value) shaklidagi ma’lumotlar ombori bo‘lib, nafaqat keshlash, balki ma’lumotlar strukturalarini saqlash, navbatlar, jadvallar, to‘plamlar kabi murakkab operatsiyalarni ham bajarishga mo‘ljallangan. Redis xotirada ishlaydigan tizim bo‘lib, mikrosekundlar darajasida javob qaytaradi, bu esa real vaqt rejimida ishlaydigan tizimlar uchun juda foydalidir. Undan veb-ilovalar, mobil ilovalar va mikroxizmat arxitekturalarida tez-tez foydalaniladi.

Memcached esa Redisga nisbatan soddaroq va engilroq tuzilmaga ega bo‘lib, yuqori darajadagi tezlik talab qilinadigan tizimlar uchun moslashtirilgan. U ham xotirada vaqtincha ma’lumot saqlash uchun mo‘ljallangan bo‘lib, ko‘pincha dinamik veb-saytlardagi tez-tez takrorlanadigan ma’lumotlar, masalan, sessiyalar yoki foydalanuvchi ma’lumotlarini saqlashda ishlatiladi. Memcachedning afzalliklaridan biri uning juda sodda arxitekturasini va juda kam resurs talab etishidir, shuning uchun u juda katta miqyosdagi tizimlarda ham ishonchli ishlaydi.

ASP.NET Core platformasida keshlashni qo‘llash uchun bir nechta variantlar mavjud. Masalan, “In-Memory Cache” ilova darajasida ishlaydi va aynan shu instansiyaga xizmat qiladi, “Distributed Cache” esa Redis yoki SQL Server kabi tashqi manbalar orqali amalga oshiriladi va bir nechta serverlar orasida umumiy keshlashni ta’minlaydi. ASP.NET Core’da keshlash yordamida sahifalar, fragmentlar yoki butun HTTP javoblar keshlanishi mumkin, bu esa sahifa yuklanish tezligini keskin oshiradi. Bundan tashqari, “Response Caching Middleware” orqali foydalanuvchi so‘rovlariga tezkor javob qaytarilishi ta’minlanadi. Amaliy jihatdan olganda, keshlash bugungi veb-ilovalarning ajralmas qismiga aylangan. Masalan, katta miqdordagi foydalanuvchilarga xizmat ko‘rsatadigan e-commerce platformalar, yangilik saytlar, ijtimoiy tarmoqlar yoki onlayn o‘yin tizimlari keshni faol ishlatadi. Chunki bir xil ma’lumotlar ko‘p marta



soʻraladi va har safar server yoki maʼlumotlar bazasiga murojaat qilish tizimni sekinlashtiradi. Keshlash orqali bu muammolar oldi olinadi va foydalanuvchi tajribasi yaxshilanadi. Shu bilan birga, mobil ilovalarda ham foydalanuvchi interfeysi tezroq ishlashi uchun lokal kesh mexanizmlaridan keng foydalaniladi. Xullas, keshlash texnologiyalari turli arxitekturalarda qoʻllanilib, tizim samaradorligini oshirishda muhim rol oʻynaydi.

Keshlash texnologiyasi zamonaviy axborot tizimlarining samarali va tez ishlashini taʼminlovchi muhim vositalardan biridir. Maʼlumotlarga tezkor kirish, server yuklamasini kamaytirish, tarmoq trafigini optimallashtirish va foydalanuvchi tajribasini yaxshilash kabi koʻplab afzalliklari tufayli keshlash bugungi veb-ilovalar va dasturiy arxitekturalarning ajralmas qismiga aylangan. Tahlillardan koʻrinib turibdiki, Redis va Memcached kabi mashhur keshlash texnologiyalari katta miqdordagi maʼlumotlar bilan ishlovchi tizimlar uchun yuqori samaradorlikni taʼminlab beradi. Ayniqsa, mikroxizmatlar asosida qurilgan arxitekturalar, ommaviy foydalaniladigan veb-platformalar va real vaqt rejimida ishlovchi tizimlar uchun keshlash muhim strategik yechimdir.

Shu bilan birga, keshlashni toʻgʻri tashkil qilish, uning amal qilish muddati (TTL), cache invalidation strategiyalari va turgʻunlik holatlariga eʼtibor berish lozim. Toʻgʻri sozlanmagan yoki notoʻgʻri boshqariladigan keshlash tizimlari aksincha, tizimning ishonchliligini kamaytirishi yoki foydalanuvchiga eskirgan maʼlumotlarni taqdim etishi mumkin. Shuning uchun har bir tizim arxitekturasi uchun mos keshlash turini va texnologiyasini tanlash, ularni ehtiyotkorlik bilan konfiguratsiya qilish muhim hisoblanadi.

Kelajakda sunʼiy intellekt, avtomatlashtirish va maʼlumotlar tahlili sohalarining rivojlanishi bilan birga, keshlash texnologiyalarining ham yanada rivojlanishi va optimallasuvi kutilmoqda. Shuningdek, keshni boshqarish jarayonlarini avtomatlashtirish, oʻzgarishlarga mos ravishda dinamik tarzda maʼlumotlarni yangilash kabi ilgʻor yondashuvlar ishlab chiqilmoqda. Shu nuqtai



nazardan, dasturchilar va tizim administratorlari uchun keshlash haqida chuqur bilimga ega bo'lish, uni real amaliyotda to'g'ri qo'llash muhim zaruratga aylanmoqda. Keshlash texnologiyasini puxta o'zlashtirgan holda tizimlarning tezkorligi, barqarorligi va samaradorligini oshirishga erishish mumkin.

Keshlash texnologiyasi zamonaviy axborot tizimlarining samarali va tez ishlashini ta'minlovchi muhim vositalardan biridir. Ma'lumotlarga tezkor kirish, server yuklamasini kamaytirish, tarmoq trafigini optimallashtirish va foydalanuvchi tajribasini yaxshilash kabi ko'plab afzalliklari tufayli keshlash bugungi veb-ilovalar va dasturiy arxitekturalarning ajralmas qismiga aylangan. Tahlillardan ko'rinib turibdiki, Redis va Memcached kabi mashhur keshlash texnologiyalari katta miqdordagi ma'lumotlar bilan ishlovchi tizimlar uchun yuqori samaradorlikni ta'minlab beradi. Ayniqsa, mikroxizmatlar asosida qurilgan arxitekturalar, ommaviy foydalaniladigan veb-platformalar va real vaqt rejimida ishlovchi tizimlar uchun keshlash muhim strategik yechimdir.

Shu bilan birga, keshlashni to'g'ri tashkil qilish, uning amal qilish muddati (TTL), cache invalidation strategiyalari va turg'unlik holatlariga e'tibor berish lozim. To'g'ri sozlanmagan yoki noto'g'ri boshqariladigan keshlash tizimlari aksincha, tizimning ishonchliligini kamaytirishi yoki foydalanuvchiga eskirgan ma'lumotlarni taqdim etishi mumkin. Shuning uchun har bir tizim arxitekturasini uchun mos keshlash turini va texnologiyasini tanlash, ularni ehtiyotkorlik bilan konfiguratsiya qilish muhim hisoblanadi.

Kelajakda sun'iy intellekt, avtomatlashtirish va ma'lumotlar tahlili sohalarining rivojlanishi bilan birga, keshlash texnologiyalarining ham yanada rivojlanishi va optimallasuvi kutilmoqda. Shuningdek, keshni boshqarish jarayonlarini avtomatlashtirish, o'zgarishlarga mos ravishda dinamik tarzda ma'lumotlarni yangilash kabi ilg'or yondashuvlar ishlab chiqilmoqda. Shu nuqtai nazardan, dasturchilar va tizim administratorlari uchun keshlash haqida chuqur bilimga ega bo'lish, uni real amaliyotda to'g'ri qo'llash muhim zaruratga



aylanmoqda. Keshlash texnologiyasini puxta o'zlashtirgan holda tizimlarning tezkorligi, barqarorligi va samaradorligini oshirishga erishish mumkin.

Foydalanilgan adabiyotlar

1. Raximov, Q. O., Tojimamatov, I. N., & Xo, H. R. O. G. L. (2023). SUNIY NEYRON TARMOQLARNI UMUMIY TASNIFI. Scientific progress, 4(5), 99-107.
2. Ahmed, S. (2021). *Caching Techniques for Web Applications*. Springer.
3. Microsoft Docs. (2024). *Caching in ASP.NET Core*. <https://learn.microsoft.com>
4. Joshi, R. (2020). *System Design Interview: Caching Strategies*. GitBooks.
5. Ulanov, D. (2019). *Redis Essentials*. Packt Publishing.
6. NGINX, Inc. (2022). *NGINX Caching Guide*. [<https://www.nginx.com>]
7. Онаркулов, М. К. (2023). ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ В ЗАДАЧАХ РАСПОЗНАВАНИЯ РЕЧИ. INNOVATIVE DEVELOPMENTS AND RESEARCH IN EDUCATION, 2(18), 248-250.
8. Onarqulov, M., Yaqubjonov, A., & Yusupov, M. (2022). Computer networks and learning from them opportunities to use. Models and methods in modern science, 1(13), 59-62.
9. Nginx Docs – Serving static content
10. Raximov, Q. O., Tojimamatov, I. N., & Xo, H. R. O. G. L. (2023). SUNIY NEYRON TARMOQLARNI UMUMIY TASNIFI. Scientific progress, 4(5), 99-107.
11. Ibragimov, S. M. (2020). IMPROVING THE EFFECTIVENESS OF TEACHING INFORMATION TECHNOLOGY IN UNIVERSITIES USING THE METHOD OF INDIVIDUALIZATION. *Экономика и социум*, (11), 127-130.
12. Mamirovich, I. S., Revkatovich, I. E., Rustamjon o'g, H. O. K., & Yigitali o'g'li, R. J. (2023). IJTIMOIY TARMOQLARDA BIG DATA TEXNOLOGIYASIDAN



FOYDALANISH TAHLILI. "RUSSIAN" ИННОВАЦИОННЫЕ ПОДХОДЫ В СОВРЕМЕННОЙ НАУКЕ, 9(1).

13. Tojimamatov, I. N., Mamalatipov, O. M., & Karimova, N. A. (2022). SUN'IY NEYRON TARMOQLARINI O 'QITISH USULLARI.
14. Tojimamatov, I., Mirkomil, M. M., & Saidmurod, S. (2023). BIG DATANING TURLI SOHALARDA QO 'LLANILISHI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 18(6), 61-65.
15. Nurmatovich, T. I. (2024, April). SUN'IY NEYRONNING MATEMATIK MODELI HAMDA FAOLLASHTIRISH FUNKTSIYALARI. In "USA" INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE TOPICAL ISSUES OF SCIENCE (Vol. 17, No. 1).
16. Memcached Developers. (2023). *Memcached Official Documentation*. [<https://memcached.org>]
17. Zhang, W., & Liu, X. (2020). *Performance Analysis of Cache Systems in Cloud Computing*. IEEE Access.
18. Apache Software Foundation. (2023). *Caching in Apache HTTP Server*. [<https://httpd.apache.org>]
19. Smith, J. (2016). *Efficient Memory Management with Cache Algorithms*. ACM SIGPLAN.
20. Amazon Web Services. (2023). *ElastiCache for Redis and Memcached*. [<https://aws.amazon.com/elasticache/>]