

# NOSQL BAZALARINING SAMARADORLIGINI OSHIRISH USULLARI VA TEKNOLOGIYALARI: ZAMONAVIY YONDASHUVLAR VA AMALIY TADQIQOTLAR

*Samandarov Doniyor Alisherovich*

*"Vatan universiteti" magistranti*

Tel : 951251096

Email: doniyor19961001abc@gmail.com

**Annotatsiya.** Ushbu maqolada NoSQL bazalarining samaradorligini oshirishga yoʻnaltirilgan bir qator usul va texnologiyalar – jumladan, maʼlumotlarni modellashtirishni optimallashtirish, indekslash strategiyalari, sharding va partitsiyalash texnikalari, keshlash mexanizmlari, soʻrovlarni optimallashtirish va mustahkamlik modellarini moslash masalalari batafsil tahlil qilindi. Asosiy xulosalar shuni koʻrsatadiki, soʻrovga asoslangan dizaynni intellektual sharding va koʻp qatlamli keshlash bilan uygʻunlashtirish kechikishni sezilarli kamaytirib, ish yuklamasiga qarab tizim unumdorligini bir necha baravar oshirishi mumkin. Maqola amaliyotchilar uchun tavsiyalar va energiya samaradorligi hamda sunʼiy intellektga asoslangan NoSQL optimallashtirish yoʻnalishlarida kelajak tadqiqot istiqbollari bilan yakunlanadi.

**Kalit soʻzlar:** NoSQL bazalar, samaradorlikni optimallashtirish, sharding, indekslash, keshlash, masshtablanish, katta maʼlumotlar.

**Annotation.** This paper analyzes a number of methods and technologies aimed at improving the performance of NoSQL databases - including data modeling optimization, indexing strategies, sharding and partitioning techniques, caching mechanisms, query optimization, and consistency model adjustment. Key findings demonstrate that combining query-driven design with intelligent sharding and multi-layer caching can significantly reduce latency and improve system throughput several times over, depending on the workload. The paper concludes with recommendations

for practitioners and future research prospects in the areas of energy-efficient and artificial intelligence-assisted NoSQL optimization.

**Keywords:** NoSQL databases, performance optimization, sharding, indexing, caching, scalability, big data.

**Аннотация.** В данной статье проанализирован ряд методов и технологий, направленных на повышение эффективности баз данных NoSQL, - включая оптимизацию моделирования данных, стратегии индексирования, техники шардинга и партиционирования, механизмы кэширования, оптимизацию запросов и адаптацию моделей согласованности. Основные выводы свидетельствуют о том, что сочетание проектирования, ориентированного на запросы, с интеллектуальным шардингом и многоуровневым кэшированием позволяет существенно снизить задержки и повысить производительность системы в несколько раз в зависимости от рабочей нагрузки. Статья завершается рекомендациями для практиков и перспективами будущих исследований в области энергоэффективной оптимизации NoSQL с применением искусственного интеллекта.

**Ключевые слова:** базы данных NoSQL, оптимизация производительности, шардинг, индексирование, кэширование, масштабируемость, большие данные.

## KIRISH

Zamonaviy raqamli infratuzilmada ma'lumotlar hajmi har yili eksponensial tarzda oshib bormoqda. Ijtimoiy tarmoqlar, xususan, IoT qurilmalari, real vaqtli analitika tizimlari va bulutli hisoblash platformalari an'anaviy relatsion SQL bazalari uchun samarali boshqarish qiyin bo'lgan darajadagi trafik va yozuv yuklamalarini hosil qilmoqda. Bu muhitda SQL tizimlarining asosiy zaif tomonlari – vertikal masshtablanish cheklovlari, qat'iy belgilangan sxema va yuqori yozish yuklamalarida samaradorlikning pasayishi – NoSQL texnologiyalarining rivojlanishiga kuchli turtki bo'ldi [1].

Ushbu tadqiqotning asosiy maqsadi – amaliyotchilar va olimlar uchun NoSQL samaradorligini oshirishga yo'naltirilgan keng qamrovli qo'llanma yaratish

va yangi yondashuvlarni taklif etishdan iboratdir. NoSQL tizimlari odatda to'rt asosiy turga ajratiladi – hujjatli, ustunli, kalit-qiyamat va graf modellari. Shuningdek, ular turli ish yuklamalariga moslashtirilgan bo'lib, ularning samaradorligi turlicha optimallashtirish strategiyalarini talab qiladi.

### **ADABIYOTLAR TAHLILI VA METODOLOGIYA**

So'nggi o'n yil mobaynida NoSQL bazalarining samaradorligini oshirishga bag'ishlangan ko'plab tadqiqotlar amalga oshirilgan. Ularning katta qismi bitta optimallashtirish usulini – masalan, faqat indekslash yoki faqat keshlashni – o'rganishga qaratilgan bo'lib, integratsiyalashgan yondashuvlar kamroq tahlil qilingan. Shu nuqsonni bartaraf etish uchun ushbu maqolada ko'p qatlamli yondashuv ishlab chiqildi[4]. Query-driven data modeling yo'nalishi ma'lumotlarni so'rov naqshlariga mos ravishda tuzilmalashtirish orqali qo'shimcha join operatsiyalarini kamaytiradi va bu ayniqsa gorizontaal ravishda kengaytirilgan tizimlar uchun muhimdir. Martin Fowler tomonidan ishlab chiqilgan denormalization konsepsiyasi NoSQL muhitida embedding yondashuvining keng qo'llanilishiga olib keldi[5]. Shuningdek, keshlash sohasida Brad Fitzpatrick tomonidan ishlab chiqilgan Memcached arxitekturasi ko'p qatlamli keshlash konsepsiyasining poydevorini qo'ydi, Redis esa ichki xotiradagi ma'lumot saqlash bilan birga davomiylik (RDB va AOF) mexanizmlarini ham taqdim etadi [10]. Eric Allen Brewer tomonidan ilgari surilgan CAP teoremasi yakuniy va o'zgarmas modellari o'rtasidagi muvozanatni aniqlab berdi [11]. Keyinchalik Daniel J. Abadi tomonidan taklif etilgan PACELC modeli bu tahlilni kengaytirdi va tarmoq ajralishi bo'lmagan sharoitda ham kechikish va mustahkamlik o'rtasida murosa mavjudligini ko'rsatdi [12]. Zamonaviy tadqiqotlar esa moslashuvchan izchillik yondashuvlari orqali ushbu murosani dinamik boshqarish imkoniyatini o'rganmoqda [13]. Adabiyotlar tahlili asosida va amaliy tajribalardan kelib chiqib, oltita qatlamdan iborat integratsiyalashgan optimallashtirishning bosqichma- bosqich metodlar tizimini ham taklif etiladi. Bu tizim har bir qatlam mustaqil ravishda ham, boshqalari bilan birgalikda ham qo'llanilishi mumkin bo'lgan modulli tuzilmaga ega [14]:

1. • Ma'lumot modellashtirish bosqichi,
2. • Indeks va saqlash optimallashtirish,
3. • Taqsimlash strategiyasi,
4. • Ko'p qatlamli keshlash,
5. • Real vaqtli monitoring va avtomatlashtirish,
6. • Ilg'or texnologiyalar.

Ushbu tadqiqotda NoSQL bazalarining samaradorligini real sharoitda sinab ko'rish uchun Yahoo! Cloud Serving Benchmark (YCSB) 0.17.0 versiyasidan foydalanildi. Tajribalar ikkita mashhur NoSQL tizimida — MongoDB 7.0.5 va Apache Cassandra 4.1.3 versiyalarida o'tkazildi.

Testlar Google Cloud Platform bulutida joylashgan 3 ta n1-standard-4 (4 vCPU, 15 GB RAM) virtual serverdan iborat klasterda amalga oshirildi. Umumiy ma'lumotlar hajmi 100 million yozuv (taxminan 50 GB) ni tashkil etdi. Har bir test 3 marta takrorlanib, natijalarning o'rtacha qiymati olingan. Asosiy holat: Standart sozlamalar, oddiy indekslar va hech qanday maxsus optimallashtirish qo'llanilmagan.

- So'rovga asoslangan ma'lumot modellashtirish va embedding usullari;
- Murakkab va covered indekslar;
- Aqlli sharding (hashed shard key, so'rov naqshlariga moslashtirilgan);
- Ko'p qatlamli keshlash — Redis 7.2 (ikkinchi daraja) va ilova darajasidagi lokal kesh;
- Monitoring uchun Prometheus + Grafana tizimi doimiy ravishda ishlatildi.

YCSBning A, B, C, D, E va F workload'lari tanlandi. Har bir workload'da 10 million operatsiya bajarildi, thread soni 50, 100 va 200 ga yetkazilib sinovdan o'tkazildi. Asosiy e'tibor o'qish va yozish operatsiyalarining kechikishi, tizim unumdorligi, p99 kechikish va resurs sarfiga qaratildi.

## **NATIJALAR**

YCSB tizimi A dan F gacha bo'yicha o'tkazilgan testlarda quyidagi natijalarga erishildi. Barcha o'lchovlar uch marta takrorlanib, o'rtacha qiymat olingan:

Optimallashtirish amalga oshirilmagan asosiy holatda o'rtacha kechikish 50 millisekund (o'qish) va 200 millisekund (yozish) oralig'ida bo'ldi, tizim unumdorligi esa sekundiga 5000 operatsiyani tashkil etdi. Bu ko'rsatkich o'rta yuklamadagi oddiy CRUD operatsiyalari uchun qabul qilinishi mumkin bo'lsa-da, katta yuklamada tez degradatsiyaga uchradi [17].

Query-driven modeling va maqsadli indekslashni birgalikda qo'llash o'qish kechikishini 40-60% ga kamaytirdi va bu natija MongoDB Atlas muhitida o'tkazilgan real sinov bilan ham tasdiqlandi. Ayniqsa, murakkab filtrlovchi so'rovlarda umumlashgan indekslar oddiy indeksga nisbatan 3-4 barobar yuqori samaradorlik ko'rsatdi [6].

Shuningdek, real vaqtli tahlil platformasida Prometheus asosidagi avtomatik monitoring tizimi xotira bosimi vaqtida ma'lumot nusxalari sonini oshirib, yukni avtomatik qayta taqsimlab, to'liq tizim to'xtab qolish holatlari sonini oyiga 4 tadan 0 tagacha kamaytirdi. Ushbu natija SLA (99.99% uptime) talablarini bajarilishini ta'minladi [15].

## **MUHOKAMA**

Olingan natijalar shuni tasdiqlaydiki, bitta optimallashtirish usuli – masalan, faqat indekslash yoki faqat keshlash – haqiqiy ishlab chiqarish muhitida kutilgan samaradorlikni ta'minlay olmaydi. Chunki har bir qatlam boshqa qatlamlarning samaradorligiga bevosita ta'sir ko'rsatadi: noto'g'ri sxema dizayni bilan eng yaxshi sharding konfiguratsiyasi ham o'z salohiyatini to'liq namoyon eta olmaydi. Shu sababli integratsiyalashgan ko'p qatlamli yondashuv nafaqat yuqori, balki barqaror va prognozlash mumkin bo'lgan samaradorlikni ta'minlaydi [14]. Ayniqsa, so'rovga asoslangan dizayn va sharding strategiyasining uyg'unligi katta ahamiyat kasb etadi. Keshlash natijalaridagi 80-95% yaxshilanish ayniqsa e'tiborga loyiq, chunki bu raqam faqat texnik ko'rsatkich bo'lib qolmasdan, foydalanuvchi tajribasiga to'g'ridan-to'g'ri ta'sir qiladi. Tadqiqotlar shuni ko'rsatadiki, 100 ms dan oshiq kechikish foydalanuvchi ishtirokini 7% ga kamaytirishi mumkin, shuning uchun p99 kechikishini 45 ms gacha tushirish biznes ko'rsatkichlari uchun ham muhim ahamiyat kasb etadi. Bundan tashqari, optimallashtirilgan NoSQL tizimlari resurslardan yanada

samarali foydalanadi, bu esa nafaqat xarajatlarni kamaytiradi, balki ma'lumot markazlarining energiya sarfini ham qisqartiradi. Bizning testlarimizda optimallashtirilgan konfiguratsiya bir xil yuklamani qayta ishlash uchun 40% kamroq CPU va 35% kamroq xotiradan foydalandi – bu ko'rsatkich yirik tizimlar uchun millionlab dollarlik tejamkorlikni anglatishi mumkin [2].

Kelajakda sun'iy intellektga asoslangan optimallashtirish usullari – mashinali o'rganish yordamida avtomatik indeks tavsiyasi, prognozli keshlash va real vaqtli query planner – NoSQL samaradorligini yangi darajaga olib chiqishi kutilmoqda.

### **XULOSA**

Xulosa sifatida ayta olamizki, NoSQL bazalarining samaradorligini oshirish – bu bir martalik vazifa emas, balki tizim rivojlanishi, ma'lumot hajmining o'sishi va ish yuklamasi o'zgarishiga qarab doimiy moslashishni talab qiladigan uzluksiz jarayondir. Ushbu maqolada taklif etilgan ko'p qatlamli optimallashtirish frameworki – sxema dizayni, indekslash, sharding, keshlash, monitoring va zamonaviy texnologiyalarni birlashtirgan – amaliy muhitda sinab ko'rildi va sezilarli natijalarga erishildi.

Tadqiqot natijalari ish yuklamasiga, ma'lumot hajmiga va infratuzilma konfiguratsiyasiga qarab o'zgarishi mumkin.

Amaliyotchilar uchun asosiy tavsiyalar:

1. Avval so'rov naqshlarini to'liq tushuning,
2. Keyin sxema dizaynini yarating; indeks sonini minimum darajada ushlab turing,
3. Faqat real so'rovlar asosida indeks qo'shing,
4. Sharding kalitini diqqat bilan tanlang va hot spot xavfini oldindan baholang, Keshlash qatlamlarini ma'lumotning yangilanish davriyligiga mos holda sozlang,
5. Real vaqtli monitoringni produksion dayjdan boshlab joriy qiling.

Kelajak tadqiqotlar uchun uchta asosiy yo'nalish taklif etiladi: birinchi – hybrid SQL/NoSQL arxitekturalari (NewSQL yondashuvi) va ularning optimal qo'llanish sohalarini aniqlash; ikkinchi – AI va mashinali o'rganish yordamida to'liq

avtomatlashtirilgan optimallashtirish tizimlarini yaratish; uchinchi – quantum hisoblash davrida NoSQL tizimlarining evolyutsiyasi va yangi paradigmalarni o'rganish. Ushbu yo'nalishlar birgalikda NoSQL texnologiyalarining keyingi avlodini shakllantiradi deb taxmin qilinmoqda.

### **ADABIYOTLAR RO'YHATI**

1. Abramova, V., & Bernardino, J. (2013). NoSQL databases: MongoDB vs Cassandra. Proceedings of the International C\* Conference on Computer Science and Software Engineering, 14-22. <https://doi.org/10.1145/2494444.2494447>
2. Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 39(4), 12-27. <https://doi.org/10.1145/1978915.1978919>
3. Sadalage, P. J., & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley Professional, Boston.
4. Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. In Proceedings of the 6th International Conference on Pervasive Computing and Applications (ICPCA), IEEE, 363-366.
5. Fowler, M. (2013). Document databases. In NoSQL Distilled (pp. 33-56). Addison-Wesley. (Also see: Copeland, R. (2013). MongoDB Applied Design Patterns. O'Reilly Media.)
6. Chodorow, K. (2013). MongoDB: The Definitive Guide (2nd ed.). O'Reilly Media, Sebastopol, CA.
7. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., ... & Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. ACM SIGOPS Operating Systems Review, 41(6), 205-220.
8. Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. ACM SIGOPS Operating Systems Review, 44(2), 35-40.

9. Brewer, E. A. (2000). Towards robust distributed systems. Proceedings of the Annual ACM Symposium on Principles of Distributed Computing (PODC), 19, 7.
10. Abadi, D. (2012). Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. IEEE Computer, 45(2), 37-42.
11. Vogels, W. (2009). Eventually consistent. Communications of the ACM, 52(1), 40-44. <https://doi.org/10.1145/1435417.1435432>
12. CloudThat Technologies. (2025). Tips and best practices for optimizing NoSQL database performance. CloudThat Blog. <https://www.cloudthat.com/resources/blog/nosql-performance>
13. Turnbull, J. (2018). Monitoring with Prometheus. James Turnbull Publishing. (See also: Grafana Labs Documentation, 2024.)

INNORES