УДК 004.75

ОПТИМИЗАЦИЯ ЗАДЕРЖЕК В CQRS С ИСПОЛЬЗОВАНИЕМ КЭШИРОВАНИЯ НА ОСНОВЕ ГРАФА ЗАВИСИМОСТЕЙ

Федякин Д.С. (Национальный исследовательский университет ИТМО)
Научный руководитель – к.т.н. Королёва Ю.А. (Национальный исследовательский университет ИТМО)

Введение. Архитектура CQRS широко применяется в high-load системах, таких как финансовые платформы и IoT-решения, благодаря разделению операций записи и чтения. Однако ключевой проблемой остается асинхронная синхронизация между write- и read-моделями, приводящая к задержкам в актуализации данных. В системах реального времени, например, биржевых торговых платформах, даже 200-миллисекундная задержка между изменением цены акции и обновлением котировок может вызвать финансовые потери. В IoT-средах задержки обработки аварийных событий угрожают безопасности и надежности систем. Современные подходы к оптимизации CQRS включают:

1) Базовое кэширование (Redis, Memcached) — сокращает нагрузку на БД, но не учитывает семантические связи между событиями и запросами, что приводит к избыточному обновлению кэша.
2) Пакетная обработка событий (Apache Spark, Flink) — снижает нагрузку за счет агрегации данных, но увеличивает задержки из-за накопления пакетов.
3) Event Sourcing — обеспечивает журналирование событий, но требует сложной синхронизации read-моделей.

Основная часть. Для минимизации задержек синхронизации между write- и read-моделями события обрабатываются в режиме реального времени с использованием Apache Kafka Streams. Этот инструмент позволяет группировать события по временным интервалам, что обеспечивает пакетную обработку связанных данных (например, изменение цены акции и её влияние на индексы). Граф зависимостей строится автоматически, используя метаданные событий (например, тэги Kafka) и анализ логов запросов к read-моделям. Для хранения графа применяется Redis Graph, где узлы представляют события и модели, а рёбра — связи между ними.

Выводы. Разработанный метод позволяет сократить задержки синхронизации в CQRS-системах и повысить их отказоустойчивость. Практическое применение включает финансовые платформы, IoT и ритейл.

Список использованных источников:

1) Fowler M. CQRS // martinfowler.com. – 2011. – URL: https://martinfowler.com/bliki/CQRS.html.
2) Kreps J., Narkhede N., Rao J. Kafka: a Distributed Messaging System for Log Processing // Proceedings of NetDB. – 2011. – P. 1–7.
3) Шкрябин Г.Д. Стратегии кэширования для повышения производительности в распределенных системах // Вестник науки №12 (81) том 1. С. 1220 - 1231. 2024 г. ISSN 2712-8849 // Электронный ресурс: https://www.вестник-науки.рф/article/19307