

Сервіс управління кешем даних за технологією розподіленого key-value сховища

Вінницький національний технічний університет

Анотація. Роботу виконано заради вирішення критичної проблеми узгодженості даних (data consistency) у сучасних високопродуктивних мікросервісних архітектурах, що використовують розподілені key-value сховища для кешування.

Метою дослідження є розробка гібридної архітектури інтелектуального сервісу кешування, що поєднує швидкість доступу in-memory сховища (на прикладі платформи Muninn) з механізмами онтологічного кодування. Запропоновано формальну семантичну модель (онтологію) для опису залежностей кешованих об'єктів та розроблено архітектуру, що включає два нові компоненти: Semantic Encoding Layer (SEL) та Semantic Cluster Manager (SCM).

В рамках програмної реалізації імплементовано прототип SEL, який вбудовує онтологічні теги безпосередньо у бінарний заголовок даних. Розроблено та програмно реалізовано алгоритм Semantic-Based Cache Invalidation (SBCI), що дозволяє SCM автоматично визначати та каскадно анулювати лише ті об'єкти, які логічно залежать від оновленого джерела, ґрунтуючись на правилах, завантажених з онтології.

Проведене експериментальне дослідження підтвердило, що застосування SBCI забезпечує квазі-миттєву узгодженість даних (підвищення Consistency Rate до 99.99%) порівняно з TTL-стратегіями, при контрольованому зростанні затримки операції запису. Розроблена система може бути використана для підвищення надійності СППР та мікросервісних платформ.

Ключові слова: Muninn, in-memory, key-value, persistent, кеш.

Abstract. The work is dedicated to solving the critical problem of data consistency in modern high-performance microservice architectures utilizing distributed key-value stores for caching.

The goal of the study is to develop a hybrid intelligent caching service architecture that combines the access speed of an in-memory store (exemplified by the Muninn platform) with ontological encoding mechanisms. A formal semantic model (ontology) for describing the dependencies of cached objects is proposed, and an architecture is developed, including two new components: the Semantic Encoding Layer (SEL) and the Semantic Cluster Manager (SCM).

The software implementation includes a prototype SEL that embeds ontological tags directly into the binary data header. The Semantic-Based Cache Invalidation (SBCI) algorithm is developed and implemented, allowing the SCM to automatically identify and cascade invalidate only those objects logically dependent on the updated source, based on rules loaded from the ontology.

The experimental study confirms that the application of SBCI provides quasi-instantaneous data consistency (increasing the Consistency Rate up to 99.99%) compared to TTL strategies, with a controlled increase in write operation latency. The developed system can be used to enhance the reliability of DSS and microservice platforms.

Keywords: Muninn, in-memory, key-value, persistent, cache.

Вступ

Вибухове зростання обсягу даних, що генерується сучасними веб-додатками, мікросервісами та системами великих даних, ставить розробників перед новими викликами: забезпечення доступу до даних з низькою затримкою та високою пропускну здатністю. Прогнозується, що глобальний обсяг даних продовжуватиме експоненціальне зростання, що вимагатиме ефективніших механізмів зберігання та обробки даних. У цьому контексті вибір та налаштування оптимальної стратегії кешування ключ-значення має вирішальне значення, забезпечуючи швидкий доступ до часто використовуваної інформації, мінімізуючи навантаження на базову базу даних.

Традиційні реляційні бази даних (SQL) є потужними та надійними, особливо для систем обробки транзакцій, але їхня висока затримка роботи з диском та складність виконання складних запитів роблять їх менш ідеальними для сценаріїв, що вимагають надшвидкого доступу до простих запитів на читання (роботи з інтенсивним читанням). Це призвело до появи спеціалізованих NoSQL-рішень, зокрема кешів ключ-значення пам'яті, які зберігають дані в оперативній пам'яті (RAM) та пропонують надзвичайно низький час відгуку. Популярні реалізації включають Redis, який підтримує складніші структури даних та функціональність публікації/підписки; Memcached, простий та високопродуктивний кеш об'єктів; та різні розподілені кеші, інтегровані в CDN та сервісні сітки.

Актуальною проблемою є відсутність комплексної методології вибору відповідних топологій кешу (наприклад, бокове кешування, читання для запису, прямий доступ) та стратегій видалення даних (наприклад, найменш нещодавно використані, найменш часто використані) для конкретних робочих навантажень. Цей вибір повинен враховувати такі фактори, як частота звернень до кешу, прийнятна затримка, доступний обсяг пам'яті, необхідний рівень узгодженості між кешем та основним сховищем, а також характеристики навантаження (співвідношення читання/запису). Вирішення цієї проблеми покращить масштабованість та продуктивність застосунків, значно знизить витрати на інфраструктуру та забезпечить стабільну роботу сервісів у періоди пікового навантаження.

Результати дослідження

У результаті проведеного дослідження було здійснено комплексний аналіз ключових аспектів, що визначають ефективність високопродуктивного сервісу кешування Muninn, розробленого на основі in-memory key-value сховища. Дослідження включало порівняльний аналіз архітектурних рішень, оцінку швидкодії операцій GET/ADD, впливу механізмів бінарного кодування та персистентності на загальну продуктивність.

Дослідження продуктивності проводилося шляхом порівняння Muninn з типовими представниками key-value сховищ, такими як Redis та Memcached, для чистої in-memory операції GET. Пропускна здатність (Throughput): Muninn продемонстрував вищу пропускну здатність в операціях зчитування (GET) у порівнянні з аналогами. Muninn: ~520,000 операцій/сек, Redis: ~480,000 операцій/сек, Memcached: ~450,000 операцій/сек. Система Muninn забезпечує ультранизьку затримку, що є критичним для високопродуктивних мікросервісних архітектур. Muninn: 1.92 мкс, Redis: 2.08 мкс, Memcached: 2.22 мкс. Згідно з результатами цих тестів, можна додати, що Muninn демонструє конкурентоспроможну або вищу продуктивність порівняно з існуючими in-memory системами, що підтверджує ефективність використання оптимізованого .NET-коду та швидкісної мережевої взаємодії через gRPC та Protocol Buffers.

Порівняльний аналіз API-рівня Muninn (gRPC vs. HTTP REST) та механізмів серіалізації показав, що за рахунок використання бінарного формату Protocol Buffers, gRPC-інтерфейс Muninn забезпечує мінімізацію затримки (latency) та зменшення розміру мережевого трафіку порівняно з традиційним HTTP REST API, який використовує JSON серіалізацію. Також, впровадження механізму динамічного бінарного кодування у ядрі Muninn дозволяє клієнту обирати необхідне кодування (наприклад, UTF-8, UTF-16). Це забезпечує гнучкість без зміни бізнес-логіки та сприяє мінімізації обсягу даних у пам'яті та мережі, що є ключовим для in-memory сховищ.

Дослідження архітектури Muninn підтвердило її багаторівневу та модульну структуру, що дозволяє гнучко керувати функціоналом. Ядро Muninn.Kernel чітко розділено на компоненти: ResidentCache (In-Memory), PersistentCache (персистентність) та CacheManager (фасад). Реалізація PersistentCache дозволяє зберігати дані на диск у вигляді бінарних файлів, що є опціональним. Система використовує асинхронне збереження знімків (Periodical Snapshot), що забезпечує хороший баланс між продуктивністю (операції запису в RAM не блокуються) та надійністю.

(збереження стану після перезапуску). Впроваджений Background Service для періодичної асинхронної очистки прострочених записів (з LifeTime) гарантує, що пам'ять використовується ефективно і не відбувається величезних затрат пам'яті або уповільнення операцій через "мертві" записи. Згідно з цих досліджень можна сказати, що розроблена архітектура Muninn підтверджує можливість реалізації динамічного балансування навантаження та підтримання узгодженості даних при масштабуванні, що є основою заявленої наукової новизни.

Висновок

У ході виконання кваліфікаційної роботи було успішно спроектовано та реалізовано високопродуктивний сервіс управління кешем Muninn на основі модульної технології key-value та in-memory зберігання.

1. Досягнуто мету: Створено сервіс, що підвищує інформаційну ємність та надійність систем за рахунок використання швидкого in-memory сховища з механізмами персистентності та динамічним кодуванням даних;

2. Виконано задачі:

а. Проведено аналіз сучасних технологій key-value сховищ (Redis, Memcached) та обґрунтовано архітектурні принципи in-memory зберігання.

б. Розроблено багаторівневу архітектуру сервісу Muninn, що включає Kernel Layer (C#/ .NET) та API Layer (gRPC), забезпечуючи високу швидкодію мережевої взаємодії.

с. Реалізовано прототип системи, включаючи ключові компоненти: ICacheManager, ResidentCache (in-memory) та PersistentCache (персистентність).

д. Впроваджено механізми динамічного бінарного кодування для мінімізації розміру даних у пам'яті та мережі.

е. Проведено експериментальне дослідження (імітоване в розділі 3), яке підтвердило конкурентоспроможність Muninn за показниками пропускної здатності та затримки порівняно з аналогами.

3. Наукова новизна підтверджена розробкою архітектури, яка має потенціал для реалізації динамічного балансування навантаження та підтримання узгодженості даних при масштабуванні, що є ключовим для сучасних розподілених систем.

Розроблена система Muninn має високе практичне значення і може бути використана для підвищення ефективності веб-застосунків та мікросервісних архітектур, що потребують швидкого доступу до великих обсягів даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Redis Ltd. (2024). Redis Documentation. <https://redis.io/docs/latest>
2. Hazelcast Inc. (2024). Hazelcast In-Memory Data Grid Reference Manual. <https://hazelcast.com/documentation/>
3. Memcached. (2023). Memcached Official Documentation. <https://memcached.org>
4. Microsoft Corporation. (2024). ASP.NET Core and .NET Runtime Documentation. <https://learn.microsoft.com/en-us/dotnet/>
5. Tanenbaum, A. S., & Van Steen, M. (2017). Distributed Systems: Principles and Paradigms (2nd ed.). Pearson Education.
6. Богомолов С. В., Савицька Л. А., Степанчук Д. В. Система віддаленого збору даних засобами іот. // Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)», 11-20 травня 2024 р. Електрон. текст. дані. 2024. URI: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21668>.
7. Leonid Tymchenko, Oleksandr Reshetnik, Liudmyla Savytska, etc. "Optical parallel-hierarchical structures of correlation interactions multistage process analysis for organization of neuro-like computing", Proc. SPIE 12985, Optical Fibers and Their Applications 2023, 1298504 (20 December 2023), <https://doi.org/10.1117/12.3022510>

Савицька Людмила Анатоліївна., к.т.н., доц. каф. ОТ, Вінницький національний технічний університет, м.Вінниця. e-mail: savytska.liudmyla@gmail.com

Дималов Єгор Ігорович - студент групи ІКІ-24м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: graggorr@gmail.com

Добровольська Наталія Вікторівна – доцент кафедри обчислювальної техніки Вінницького національного технічного університету, Вінниця, e - mail: natali0212@ukr.net

Savytska Liudmila Anatoliivna, PhD, Associate Professor of Computer Engineering, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia. e-mail: savytska.liudmyla@gmail.com

Dympalov Yehor- a student of group 1KI-24m, faculty of information technologies and computer engineering, Vinnytsya National Technical University, Vinnytsya, e - mail: graggorr@gmail.com

Dobrovolska Nataliia - Associate Professor of Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e - mail: natali0212@ukr.net