

DOI 10.36074/logos-18.08.2023.41

СУМІСНЕ РЕДАГУВАННЯ ТЕКСТІВ В РОЗПОДІЛЕНИХ СИСТЕМАХ

ORCID ID: 0000-0001-9239-4913

Г.Є. Безугластарший викладач кафедри системотехніки
*Харківський національний університет радіоелектроніки***В. В. Клімов**здобувач вищої освіти
факультету «Автоматизації та комп'ютерно-інтегрованих технологій»
Харківський національний університет радіоелектроніки

УКРАЇНА

Однією з поширених міжгалузевих задач є створення та редагування текстових документів, що також потребує розв'язання в контексті розподілених систем. Аналіз існуючих рішень для сумісного редагування тексту (Google Docs, Notion, Etherpad) свідчить про обмежену здатність цих систем до розв'язання проблем конфліктів під час одночасних операцій редагування даних, контролю версій та нестабільного мережевого з'єднання. Реплікація дозволяє забезпечити миттєвість редагування тексту та високий рівень доступності системи шляхом забезпечення надлишковості [1]. Реплікація з одним лідером – у системі виділено один спеціальний вузол – лідер, що опрацьовує усі запити на зміну даних. Реплікація з багатьма лідерами – на відміну від попередньої моделі, система дозволяє багатьом вузлам незалежно опрацьовувати запити на запис та зміну даних. Кожен вузол системи, що реалізує такий тип алгоритму, виступає одночасно як лідер і підписник, здатний застосовувати зміни та реплікувати їх на інші вузли. Ця модель забезпечує кращу доступність та пришвидшити продуктивність виконання запитів на запис. Відсутність єдиної точки відмови в такій моделі також підвищує надійність. Недоліком реплікації з багатьма лідерами є необхідність розв'язувати конфлікти – розбіжності між репліками, що виникають унаслідок відмов чи затримок в мережі. Для вирішення проблеми виникнення конфліктів, спричинених одночасними змінами даних у розподілених системах існують два основних підходи: Operational transformation та Conflict-free replicated data types.

Operational transformation (OT) – це підхід для забезпечення узгодженості, створений для одночасного редагування впорядкованого списку елементів, як то список символів, з яких складаються текстові документи [2]. OT гарантує, що послідовність операцій, застосованих кожним користувачем, призводить до узгодженого та дійсного стану документа для всіх учасників, незалежно від порядку отримання та застосування операцій. Це робить можливим спільне редагування тексту в режимі реального часу з мінімальною кількістю конфліктів і проблем із синхронізацією даних [3].

Коректні алгоритми управління OT потребують централізованого серверу для роботи, що унеможлиблює їх використання у peer-to-peer системах, унеможлиблює використання наскрізного шифрування, зменшує надійність системи, створюючи єдину точку відмови [4]. OT – технологія забезпечення сильної узгодженості, що також спричиняє виникнення накладних витрат на синхронізацію для забезпечення консистентності реплік документа.

Conflict-free replicated data type (CRDT) – це абстрактний тип даних з чітко визначеним інтерфейсом, який створений бути реплікованим в декількох конкурентних процесах та здатний автоматично передбачувати розв'язувати конфлікти. CRDT були створені для різноманітних типів даних: множини, списки, колекції «ключ-значення», графи, лічильники та JSON дерева. CRDT забезпечують збіжність за допомогою комутативності – якщо дві репліки отримали однакові набори оновлень, вони опиняться в однаковому стані, навіть якщо оновлення надійшли в різному порядку. Системи на основі CRDT застосовують евентуальну модель консистентності (eventual consistency) та оптимістичну реплікацію [5].

Система спільного редагування з реплікацією має синхронізувати розподілені репліки документів та застосовувати зміни до кожної з них. Є два основних підходи для розповсюдження змін: на основі синхронізації стану (state-based); на основі синхронізації операцій (operation-based). У системі з використанням state-based CRDT, репліки синхронізуються обмінюючись усім своїм станом через мережу. Ця функція має бути комутативною та ідемпотентною, щоб виконати умови евентуальної збіжності. У системах, побудованих на основі operation-based CRDT, кожна модифікація розподілених даних кодується як операція та надсилається мережею. CRDT на основі операцій гарантує, що будь-які дві паралельні операції є комутативними; тобто репліки можуть застосовувати ці операції в будь-якому порядку, і результат буде однаковим.

Для вирішення проблеми розв'язання конфліктів під час операцій одночасного редагування, а також для забезпечення цілісності даних в умовах розподіленої архітектури доцільне використання CRDT. Такий вибір є оптимальним, зважаючи на переваги таких типів даних у вигляді комутативності та ідемпотентності операцій, толерантності до тимчасових розділень у мережі.

Список використаних джерел:

- [1] Dallabetta P. (2022) Data replication explained: examples, types, and use cases. Вилучено із URL: <https://redis.com/blog/what-is-data-replication>.
- [2] Kleppmann M. (2017) Designing data-intensive applications: the big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media, 624 p.
- [3] Sun C., Ellis C. (1998) Operational transformation in real-time group editors. the 1998 ACM conference, м. Seattle, Washington, United States, New York, New York, USA, Вилучено із URL <https://doi.org/10.1145/289444.289469>
- [4] Geoffrey Litt, Sarah Lim, Martin Kleppmann, Peter van Hardenberg (2022) Peritext: A CRDT for Collaborative Rich Text Editing / G. Litt та ін. Proceedings of the ACM on Human-Computer Interaction. (InteractionVolume 6, Issue CSCW2: p. 1–36.) URL: <https://doi.org/10.1145/3555644>.
- [5] Preguiça, N., Baquero, C., Shapiro, M. (2018). Conflict-Free Replicated Data Types CRDTs. In: Sakr, S., Zomaya, A. (eds) Encyclopedia of Big Data Technologies. Springer, Cham. https://doi.org/10.1007/978-3-319-63962-8_185-1