

МОДЕЛІ СИНХРОНІЗАЦІЇ БЕЗКОНФЛІКТНИХ РЕПЛІКОВАНИХ ТИПІВ ДАНИХ В МОБІЛЬНИХ ОФЛАЙН ЗАСТОСУНКАХ

Полторацький А.О.

Науковий керівник – д.т.н., проф. Єрохін А.Л.

Харківський національний університет радіоелектроніки

(61166, Харків, просп. Науки, 14, каф. ПІ, ауд. 267, тел. (057) 70-21-446)

There are a lot of ad-hoc and error-prone approaches for eventual consistency. This article reviews the solution of the problem of data synchronization in offline applications. The main goal is to describe main models and approaches of synchronization using existing conflict-free replicated data types (CRDTs). It describes main principles of CmRDT, CvRDT and their modifications. CRDT will be compared to Operational Transformation in scope of text editing tasks. Finally, we will make a conclusion about feasibility of implementation of CRDTs in mobile applications.

Сучасні програмні продукти досить часто потребують підтримки роботи з даними у режимі офлайн. Такими є розподілені сховища даних, редактори тексту, календарі та інші колаборативні застосунки. У якості прикладу такого продукту можна виділити Evernote, що є сервісом для роботи з нотатками. Однією з найважливіших вимог в даному випадку є відсутність конфліктів та блокувань для користувача, або їх мінімізація. Сучасним рішенням даної проблеми є conflict-free replicated data types (CRDT, безконфліктні репліковані типи даних).

Виходячи з CAP-теореми, для будь якої розподіленої системи можливо одночасно забезпечити лише дві з трьох властивостей [1]:

- узгодженість;
- доступність;
- стійкість до розподілення.

CRDT вирішує дану проблему завдяки забезпеченню принципу strong eventual consistency (SEC), допускаючи узгодженість даних у кінцевому рахунку та наявності певних алгоритмів вирішення конфліктів [2].

За моделями синхронізації CRDT розподіляються на два класи [3]:

- комутативні (operation-based, CmRDT);
- засновані на зберіганні стану (state-based, CvRDT).

В комутативних CRDT усім реплікам відправляються лише дані про виконану операцію. Наприклад, при додаванні нового елемента до списку буде відправлений лише доданий елемент. Варто відзначити, що операції мають бути комутативними, щоб не залежати від порядку отримання змін.

Модель, що заснована на зберіганні стану, передбачає відправку цілої структури даних, а не окремих змін. Дані структури даних повинні підтримувати операції:

- query – операція читання без мутації стану;

- update – мутація стану;
- merge – злиття зі станом з іншої репліки.

Операція merge повинна буди асоціативною, комутативною та ідемпотентною [4].

Для успішного використання даного підходу також необхідно, щоб виконувались наступні вимоги [5]:

- дані мають складати напіврешітку;
- репліки мають складати з'язний граф.

Було доведено, що будь-яку структуру CRDT можна представити у вигляді CmRDT та CvRDT. Комутативний підхід потребує налагодженого способу доставки повідомлень до реплік, у той час як CvRDT потребує лише запису та читання стану, але при цьому розмір стану може бути достатньо великим.

Для оптимізації state-based CRDT було створено окремий підклас delta-based CRDT, що об'єднує обидва попередні підходи, розсилаючи дельта-мутатори, що оновлюють стан згідно до попереднього часу синхронізації [5]. Але при першій синхронізації все одно необхідно передати увесь стан. Існує також альтернатива CRDT для роботи з текстом – Operational Transformation. Але потрібно виділити, що підхід Operational Transformation більш вимогливий до ресурсів, складний та за результатами деяких досліджень – менш стабільний [4]. Отже, неможливо обрати однозначно кращий з даних двох інструментів.

Після аналізу існуючих інструментів роботи с CRDT було виявлено, що існує певна кількість готових відкритих програмних рішень для веб-застосунків. У мобільних застосунках, як правило, використовують ad-hoc або проприетарні рішення (наприклад, компанія Apple використовує CRDT у додатку Notes, але не надає публічний інтерфейс для роботи з подібними типами даних), тому є доцільним у майбутньому реалізувати відкриту нативну бібліотеку для роботи з CRDT та протестувати її.

Список використаних джерел

1. Теорема CAP. URL: https://uk.wikipedia.org/wiki/Теорема_CAP (дата звернення: 15.02.2020).
2. Strong eventual consistency. URL: https://en.wikipedia.org/wiki/Eventual_consistency (дата звернення: 15.02.2020).
3. A comprehensive study of Convergent and Commutative Replicated Data Types. URL: <https://hal.inria.fr/inria-00555588/document> (дата звернення: 15.02.2020).
4. Репликация без конфликтов: CRDT в теории и на практике. URL: <https://habr.com/ru/post/272987/> (дата звернення: 15.02.2020).
5. CRDT: Conflict-free Replicated Data Types. URL: <https://habr.com/ru/post/418897/> (дата звернення: 15.02.2020).