

Ігор Цимбалюк, Юрій Грицюк*

¹ студент Національного лісотехнічного університету України, Львів, Україна. 23tsymbaliuk.i@nltu.lviv.ua

² професор кафедри ІПЗ, Національного лісотехнічного університету України, Львів, Україна. <https://orcid.org/0000-0001-8183-3466>.
yurii.hrytsiuk@nltu.edu.ua

Застосування XOR фільтрів для агрегації та валідації транзакцій у направлених ациклічних графах

Анотація. Описано умови поділу направленого ациклічного графа на хронологічні сектори з урахуванням класів транзакцій. Для їх оброблення удосконалено алгоритм валідації блоків в паралельних гілках направленого ациклічного графа і їх поширення вузлами мережі. Проведено оцінку ефективності порівняно з стандартними підходами до вирішення подібних завдань. Результати дослідження можна використати в неоднорідних розподілених мережах для встановлення консенсусу.

Ключові слова – валідація, консенсус, паралельні обчислення, пріоритизація процесів, хронологічні блоки, порядкові обмеження.

Направлені ациклічні графи завдяки нелінійній структурі адаптовані до структурування даних при проведенні паралельних обчислювальних процесів та гарантування порядковості при подальших перевірках. Як у будь-яких паралельних процесів труднощі становлять синхронізація їх стану, адже в неоднорідних розподілених мережах, орієнтованих на швидкість роботи, не існує гарантії уникнення колізій між альтернативними гілками, тобто такими, в яких записані дії (транзакції) над одними і тими ж даними. Для вирішення конфліктів визначається пріоритетність кожної, яка залежить від кількості підтверджень (кожне посилення на вузол графа від наступних – є його підтвердженням), часом створення (старіші в пріоритеті), чи тривалістю ланцюжка (більшу вагу мають довші), що шукається шляхом обходу гілки графа. Часова складність виконання для даного підходу:

$$O(V + E), \quad (1)$$

де: V – кількість вершин ($V > 1$), E – кількість ребер ($E > 1$), зазвичай, до останнього спільного блока. Їхня кількість зростає при зниженні частоти синхроніза-

ції майже експоненціально, бо кількість ребер відповідає підтвердженням транзакції, що заохочується мережею. Натомість, постійна синхронізація знижує переваги паралельного виконання.

Пропонується умовно поділити граф в хронологічному порядку на сегменти (блоки) між синхронізаціями, тривалості прийнятної для проведення однієї транзакції та ввести обмеження на одну вихідну транзакцію в блоці для кожного запису. При перевірці тільки вихідних адрес транзакцій, а не усіх, зменшується обсяг пам'яті, необхідний для побудови фільтру. За такої глибини блоку простіше виконувати злиття графів. Умови можна змінювати, наприклад, передбачаються глобальні транзакції, одночасні до великої кількості записів. Формально, це буде одна транзакція, але іншого типу. Транзакцій без адресата (порожні) може бути довільна кількість, адже вони не впливають на баланс. Як окремий випадок, розглядаємо двосторонню транзакцію, тобто обмін, який повинен містити дві транзакції, і які не логічно записувати в різні блоки. Незалежно від класу транзакції – вона має бути проведена тільки на одному вузлі кластера, інакше буде оброблена як дублююча. Для синхронізації виконується об'єднання їх в XOR фільтрі [1]. У довільному порядку, який дає змогу здійснити вибір як одного, так і декількох лідерів мережі, передати на перевірку блок вихідних адрес обраному валідатору. Вони перевіряються на входження у фільтр та за ідеальних умов виконуються в межах лінійного часу, адже за аналогією з фільтром Блума гарантується точність при відсутності елемента в множині. Вищу продуктивність можна досягнути при застосуванні паралельних обчислень, до яких XOR фільтр повністю адаптований. Кожен вузол мережі містить весь адресний простір, що дає можливість виконати оброблення будь-яких транзакцій незалежно від запиту [2]. Перевірені елементи блоку вносимо у фільтр, який використовуємо для перевірки блоків від інших вузлів, поки всі не будуть синхронізовані. Ефективність цієї перевірки для елемента може бути виражена як:

$$O(1 + p + C/n), \quad (2)$$

де: p – похибка фільтру ($0 \leq p \leq 1$, регулюється заданою кількістю біт), C – кількість колізій ($0 \leq C \leq n$), n – кількість елементів для перевірки.

Елементи, що не пройшли перевірку, можна обробляти довільним способом, що залежить від правил консенсусу мережі. Оскільки за нормальних умов таких елементів бути не повинно – найпростіше просто їх відхилити. Синхронізація мережі проводиться на кожному кроці, тому можна вважати кожну попередню транзакцію підтвердженою, що означає:

- відсутність потреби глибокого пошуку в графі;
- зменшення кількості ребер графу;
- зниження використання пам'яті для перевірки;
- простіше злиття блоків.

Висновки. Застосування хронологічного поділу направленого ациклічного графа на сегменти знижує кількість необхідних ребер графу для підтвердження транзакцій, підвищується швидкість його оброблення та злиття. Використання XOR фільтру для агрегації сегментів оптимізує об'єм пам'яті та вирішує вразливості, пов'язані з дублюючими транзакціями без додаткових операцій над даними. Оброблення транзакцій за класом надає винятки для дій, продуктивність яких постраждає від обмежень блоків.

Список використаних літературних джерел

- [1] Thomas Mueller Graf, Daniel Lemire, (December 19, 2019). Xor Filters: Faster and Smaller Than Bloom Filters. <https://doi.org/10.48550/arXiv.1912.08258>
- [2] Tianyu Wang, Zizhan Chen, Wenbin Zhu, Qian Wei, Zhaoyan Shen, Zili Shao. (January 2023). IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (99), 1-1. <https://doi.org/10.1109/TCAD.2023.3301393>

Application of XOR filters for aggregation and validation of transactions in directed acyclic graphs

Ihor Tsybalyuk, Yuriy Hrytsyuk

The conditions for dividing a directed acyclic graph into chronological sectors are described, taking into account transaction classes. To process them, the algorithm

for validating blocks in parallel branches of a directed acyclic graph and their propagation by network nodes has been improved. An assessment of efficiency in comparison with standard approaches to solving similar problems has been carried out. The results of the study can be used in heterogeneous distributed networks to establish consensus.