

АЛГОРИТМИ БЕЗ ОЧІКУВАННЯ

Ботуз В. В.

Науковий керівник – старший викладач, Сокоорчук І. П.
Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

тел.: +38(068) 910-51-68, e-mail: vitalii.botuz@nure.ua

This work explores the concept of wait-free algorithms as a subset of non-blocking synchronization and their advantages and disadvantages in parallel programming. The Kogan-Petrank queue is used as a specific example to illustrate the features of wait-free algorithms. Also this work describes their general development in different periods of time. This work discusses the scalability and reliability of non-blocking algorithms and the potential for further optimization. Overall, this work suggests that wait-free algorithms have the potential to outperform synchronization-based algorithms in productivity without waiting.

Поняття «Алгоритми без очікування» (wait-free) є підрозділом стратегії Неблокуючої синхронізації. Неблокуюча синхронізація – це підхід в паралельному програмуванні, в якому відходять від традиційних примітивів блокування, таких як семафори, м'ютекси або події. Розділ доступу між потоками відбувається за рахунок атомарних операцій (такі як зчитування, модифікація, запис) а також спеціальних механізмів блокування, розроблених під конкретну задачу. Кожна операція виконується за конкретну кількість кроків, що не залежить від інших потоків.

Алгоритми без блокувань (lock-free) – це алгоритми, для яких гарантується системний процес принаймні одного потоку. Наприклад, потік, що виконує операцію “порівняння з обміном” в циклі, теоретично може виконуватися нескінченно, але кожна його ітерація означає, що якийсь інший потік здійснив прогрес, тобто і вся система продовжує прогресувати. Головною перевагою неблокуючих алгоритмів є набагато краща, ніж у випадку блокуючих алгоритмів, масштабованість по кількості процесорів. Також варто зазначити, що якщо ОС зупинить один з потоків фоновою задачею, то інші продовжать виконувати свою роботу у звичному режимі, або навіть візьмуть невиконану роботу на себе. Наглядними прикладами алгоритмів, що побудовані за таким принципом є стек, черга, хеш-таблиця та множина, що нині є загальновідомими та загальноживаними.

Причиною створення таких алгоритмів є прагнення організувати загальний доступ до даних в багатопоточних додатках та повністю позбутися від взаємних блокувань. При їх роботі існує строга гарантія прогресу, тобто кожна операція в такому алгоритмі виконується за певну

визначену кількість кроків, що не залежить від інших потоків. Така властивість має вирішальну роль для систем реального часу, і у цілому є досить доречною для будь-яких програм, якщо затрати на продуктивність не надто високі. Варто зазначити, що головними недоліками таких алгоритмів є зациклювання (livelock) та «стан гонки» (race condition).

У 1980-му році професором Морісом П. Герлігі з Пенсильванського університету Карнегі Мелон було встановлено та доведено, що будь-який алгоритм може бути переведений у wait-free версію.

Алгоритми без очікування були рідкістю як у дослідженнях, так і на практиці до 2011 року, коли Ізраїльські вчені Алекс Коган та Ерез Петранк представили статтю, в якій було продемонстровано реалізацію wait-free алгоритму черги (Kogan-Petrank queue) за допомогою примітиву CAS (Compare And Swap). Ця реалізація була максимальною приближеною за швидкістю до звичайного аналогу, що розвінчувало міф про те що алгоритми без очікування є досить повільними. В наступній їх статті було також запропоновано метод прискорення роботи будь-яких алгоритмів без очікування, що дозволяв оптимізувати wait-free алгоритми до рівня його звичайного аналогу при цьому маючи власні переваги методології неблокуючої синхронізації. Таким чином, реалізація алгоритмів без очікувань нині доступна та використовується для багатьох структур даних.

Переваги алгоритмів без очікування – більша надійність при роботі багатьох потоків, гарантія того, що у випадку припинення роботи одного з потоків додаток продовжить працювати стабільно, краща масштабованість.

Проаналізувавши переваги, недоліки та ключові особливості використання алгоритмів без очікування, можна зробити висновок, що раніше вони не користувалися популярністю через свою ресурсозатратність. Тим не менш, з розвитком алгоритмічної теорії та різних методологій було знайдено рішення, що дозволило мінімізувати наявний недолік низької продуктивності. Саме за останні 11 років їх почали використовувати значно частіше. Доказом цього є алгоритми стеку, черги, хеш-таблиці та багато інших, без яких сучасне професійне програмування неможливе.

Розвитком таких алгоритмів може стати подальша оптимізація, що дозволить їм перевершити алгоритми на основі блокуючої синхронізації за продуктивністю, і зайняти домінуюче місце серед усіх алгоритмів.

Список використаних джерел:

1. Maurice, H., & Shavit, N. (2008). *The Art of Multiprocessor Programming*. Morgan Kaufmann Publishers.
2. ResearchGate. (2011, 12 лютого). *Wait-Free Queues With Multiple Enqueuers and Dequeuers*.
3. Raynal, M. (2012). *Concurrent Programming: Algorithms, Principles, and Foundations*. Springer.