

ДОСЛІДЖЕННЯ РЕАЛІЗАЦІЇ АЛГОРИТМІВ ВЗАЄМНОГО ВИКЛЮЧЕННЯ В РОЗПОДІЛЕНИХ СИСТЕМАХ НА ОСНОВІ ПЕРЕДАЧІ МАРКЕРА

Вінницький національний технічний університет

Анотація

У роботі розглянуто дослідження та програмну реалізацію алгоритмів взаємного виключення в розподілених системах, зокрема методів, що базуються на передачі маркера. Проведено аналіз принципів роботи кільцевого алгоритму Token Ring та алгоритму Suzuki–Kasami для забезпечення коректного доступу до спільних ресурсів. Обґрунтовано вибір мови Python та технології багатопотокової обробки для моделювання процесів взаємного виключення. Створено програмну модель та проведено тестування ефективності алгоритмів за критеріями кількості повідомлень, затримки доступу та стійкості до втрати маркера. Результати дослідження підтверджують високу масштабованість та низькі комунікаційні витрати обраних методів.

Ключові слова: розподілені системи, взаємне виключення, Token Ring, Suzuki-Kasami.

Abstract

The paper presents a study and software implementation of mutual exclusion algorithms in distributed systems, specifically focusing on token-based methods. The operational principles of the Token Ring and Suzuki–Kasami algorithms are analyzed to ensure correct access to shared resources. The choice of the Python programming language and multi-threading technology for modeling mutual exclusion processes is justified. A software model was developed, and the efficiency of the algorithms was tested based on criteria such as message complexity, access latency, and resilience to token loss. The research results confirm high scalability and low communication overhead of the selected methods.

Keywords: distributed systems, mutual exclusion, Token Ring, Suzuki-Kasami.

Вступ

Із розвитком комп'ютерних мереж та хмарних платформ особливого значення набуває проблема організації синхронізованого доступу до спільних ресурсів у розподілених системах [1]. У багатовузлових середовищах відсутність спільної пам'яті та глобального годинника робить класичні методи синхронізації непридатними, що вимагає використання спеціалізованих алгоритмів. Алгоритми на основі передачі маркера дозволяють мінімізувати мережеве навантаження, гарантують відсутність конфліктів у критичних секціях та забезпечують справедливий порядок обслуговування запитів [2]. Метою роботи є дослідження ефективності алгоритмів взаємного виключення на основі передачі маркера в розподілених системах та розробка програмної моделі для аналізу їх функціонування.

Постановка задачі дослідження

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати вимоги до алгоритмів взаємного виключення;
- дослідити механізми роботи алгоритмів Token Ring та Suzuki–Kasami [2, 3];
- розробити програмну реалізацію алгоритмів у середовищі Microsoft Visual Studio з використанням бібліотеки multiprocessing;
- провести експериментальний аналіз впливу кількості вузлів та мережевих затримок на продуктивність системи.

Виклад основного матеріалу

У розподілених системах без спільної пам'яті взаємне виключення реалізується шляхом обміну повідомленнями. Ключовою вимогою є забезпечення того, щоб у будь-який момент часу лише один процес мав доступ до критичної секції [4, 5]. Алгоритми на основі передачі маркера використовують спеціальне повідомлення (маркер), володіння яким дає право на вхід до критичної секції.

Алгоритм Token Ring передбачає логічну організацію вузлів у кільце [2]. Маркер постійно циркулює по колу. Процес, який потребує доступу до ресурсу, чекає на прихід маркера, захоплює його, виконує роботу в критичній секції і передає далі. Перевагою методу є гарантована відсутність взаємоблокувань (deadlocks), але недоліком є висока затримка при низькій інтенсивності запитів.

Алгоритм Suzuki-Kasami є більш досконалим методом, де процес, що бажає увійти до критичної секції, надсилає запит усім іншим учасникам системи [3]. Маркер надсилається лише тоді, коли вузол-власник звільняє ресурс. Алгоритм використовує вектори послідовних номерів запитів для ідентифікації актуальності повідомлень, що дозволяє уникнути зайвих передач маркера та підвищити ефективність системи.

Розроблено програмний комплекс мовою Python, що моделює функціонування розподіленої системи з використанням алгоритмів взаємного виключення [6]. Основними компонентами архітектури є такі.

1. Бібліотеки multiprocessing для створення незалежних процесів, кожен з яких імітує окремих вузол у мережі.
2. Черги (multiprocessing.Queue), що забезпечують асинхронну передачу повідомлень та маркера між вузлами.
3. Структури даних для зберігання номерів останніх запитів (RN) та черги очікування всередині самого маркера.
4. Тестування, що проводить автоматизований збір метрик (час доступу, кількість службових повідомлень) для подальшої статистичної обробки..

Аналіз результатів тестування програми виконано для різної кількості вузлів у системі. (табл.1-3).

Таблиця 1 – Часові характеристики роботи алгоритму Suzuki-Kasami

Кількість вузлів (N)	Загальна кількість запитів	Середня затримка, мс	Пропускна здатність
2	100	1.25	800.45
4	100	2.58	387.60
8	100	5.42	184.50

Таблиця 2 – Комунікаційні витрати системи

Кількість вузлів (N)	Загальна кількість повідомлень	Повідомлень на один вхід у критичну секцію
2	200	2.00
4	400	4.00
8	800	8.00

Таблиця 3 – Коефіцієнт ефективності використання ресурсу

Кількість вузлів (N)	Коефіцієнт корисної дії
2	0.92
4	0.85
8	0.78

Проаналізувавши обчислені показники часової затримки та комунікаційної складності при різній кількості вузлів системи (табл. 1–3), можна зробити такі висновки:

- при малій кількості запитів на вхід до критичної секції значення часової затримки є мінімальними, проте, при збільшенні кількості вузлів витрати на передачу повідомлень-запитів зростають лінійно; це свідчить про те, що для невеликих розподілених систем (2–4 вузли) алгоритм забезпечує практично миттєвий доступ до ресурсу;
- збільшення кількості процесів у системі при інтенсивному потоці запитів призводить до стабілізації пропускної здатності, але супроводжується зростанням кількості службових повідомлень; при цьому коефіцієнт ефективності використання ресурсу дещо знижується через зростання часу очікування маркера в черзі, що є характерною особливістю алгоритмів на основі передачі дозволу.

Отже, основними чинниками, які впливають на зміну розглянутих показників (час очікування, кількість повідомлень та ефективність синхронізації), є кількість активних вузлів у системі, інтенсивність генерації запитів до спільного ресурсу та безпосередньо обраний алгоритм.

Висновки

Досліджено проблему взаємного виключення, як одну із ключових задач синхронізації в розподілених системах, визначено основні сфери її застосування в мережевих архітектурах та хмарних обчисленнях. Розглянуто різні підходи до забезпечення доступу до критичної секції, зокрема методи, що базуються на передачі маркера, такі як кільцевий алгоритм Token Ring та алгоритм Suzuki-Kasami. Описано алгоритми їх функціонування, проведено аналіз їхньої складності та стійкості до відмов вузлів. На основі літературних джерел досліджено принципи побудови розподілених середовищ без спільної пам'яті, де обмін даними відбувається виключно шляхом передачі повідомлень..

Програмно реалізовано алгоритми Token Ring та Suzuki-Kasami мовою Python з використанням бібліотеки multiprocessing для моделювання розподілених вузлів [2, 3, 6]. Описано всі компоненти програмного коду, включаючи механізми черг запитів та вектори послідовних номерів. Також наведено UML-діаграми, що відображають структуру та логіку взаємодії процесів у системі [7]. Проведено тестування розробленої програми, проаналізовано її результати.

Визначено, що збільшення кількості вузлів у системі призводить до лінійного зростання часу очікування маркера та кількості повідомлень, необхідних для входу в критичну секцію. Це підтверджує теоретичну складність алгоритму Suzuki-Kasami (N повідомлень на один вхід) та свідчить про високу ефективність обраного токенного алгоритму в умовах великої кількості запитів порівняно з безмаркерними методами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Rauber, T. , Runger, G. Chemnitz Parallel Programming for Multicore and Cluster Systems (3. edition). Springer. 2023. 554 p.
2. Token ring algorithm. URL: <https://www.ques10.com/p/2211/short-note-on-token-ring-algorithm/>
3. Suzuki-Kasami algorithm. URL: https://en.wikipedia.org/wiki/Suzuki-Kasami_algorithm.
4. У чому різниця між м'ютексом, монітором та семафором. URL: <https://javarush.com/ua/groups/posts/uk.2174.u-chomu-rznicja-mzh-mjuteksom-montorom-ta-semaforom>.
5. Критичні секції та проблема взаємного виключення. URL: <https://vseosvita.ua/library/krytychni-sektsii-ta-problema-vzaiemnoho-vykliuchennia-875683.html>
6. Tkinter — Python Interface to Tcl/Tk. Python.org. URL: <https://docs.python.org/3/library/tkinter.html>
7. The Unified Modeling Language. URL: <https://www.uml-diagrams.org/>.

Москалец Альона Ігорівна – студентка кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: alena.i.moskalets@gmail.com;

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Moskalets Alyona Ihorivna – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: alena.i.moskalets@gmail.com;

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua