

УДК 004.621.3

І.С. Іванченко,

В.О. Хорошко,

доктор технічних наук, професор

УПРАВЛІННЯ ПАРАЛЕЛЬНОЮ ОБРОБКОЮ ІНФОРМАЦІЇ

У статті досліджено методи збереження цілісності ІР при паралельній обробці інформації. Розглянуто системи управління інформаційними ресурсами (СУІР), їхні переваги та недоліки під час використання для обробки інформації.

Ключові слова: інформаційні ресурси, методи захисту, система управління інформаційними ресурсами, моделі захисту.

В статье исследованы методы сохранения целостности ИР при параллельной обработке информации. Рассмотрены системы управления информационными ресурсами (СУИР), их достоинства и недостатки в использовании для обработки данных.

Ключевые слова: информационные ресурсы, методы защиты, система управления информационными ресурсами, модели защиты.

The methods for the preservation of the information system integrity while parallel information processing are analyzed. Data resource management control systems, their advantages and lacks in using for information processing are considered.

Keywords: information resources, protection methods, data resource management control system, models of protection.

У зв'язку з чітко вираженою тенденцією до розширення обробки інформації у режимі колективного маніпулювання інформацією, а також інтенсивним розподіленням інформаційних ресурсів (ІР), особливої актуальності набуває одна із задач підтримки цілісності – задача управління процесами паралельної обробки.

Методи збереження цілісності ІР при паралельній обробці інформації розглянуті дуже докладно в [1–3]. Найбільш поширений і надійний метод управління паралельною обробкою пов'язаний з поняттями транзакції та розкладу (графік запуску).

Транзакція – це блок програми інформації, виконання якого зберігає несуперечливість інформаційних ресурсів. Якщо ІР несуперечливі до виконання транзакції, то вони повинні захищатися несуперечливо і після її виконання. Для того, щоб забезпечити виконання цієї умови, транзакція повинна сприйматися як неподільний блок операцій, в якому повністю виконуються всі складні операції або не виконуються жодна з них.

У більшості методів контролю паралельної обробки використовується певна форма перевірки того, чи результат виконання транзакцій такий, яким він був би тоді, коли б усі транзакції, що обробляються в системі у даний інтервал часу, виконувалися послідовно. Ці методи ґрунтуються на теорії зведення процесу паралельного виконання до послідовної форми [4]. У випадку, коли транзакції

виконуються послідовно, тобто виконуються усі сукупності складових операцій однієї транзакції, а потім – наступної, паралельна обробка відсутня. Послідовне виконання зумовлює неможливість взаємовпливу транзакції, коли результат виконання операції однієї транзакції спричиняв би зміну результатів виконання операцій іншої транзакції у процесі її виконання. Однак це не відповідає дійсності у випадку одночасної обробки системою декількох транзакцій. Загальноприйнятим критерієм коректності управління паралельною обробкою є ключове поняття “здатність до впорядкування”. Точніше кажучи, переміжне виконання заданої множини транзакцій буде коректним, якщо вона впорядкована, тобто при її виконанні буде отриманий такий само результат, як і при послідовному виконанні тих же транзакцій [5].

Для гарантованого зведення до послідовної форми необхідно, щоб планувальник (рис. 1) забезпечував контроль паралельної обробки за допомогою механізмів блокувань, або міток часу.

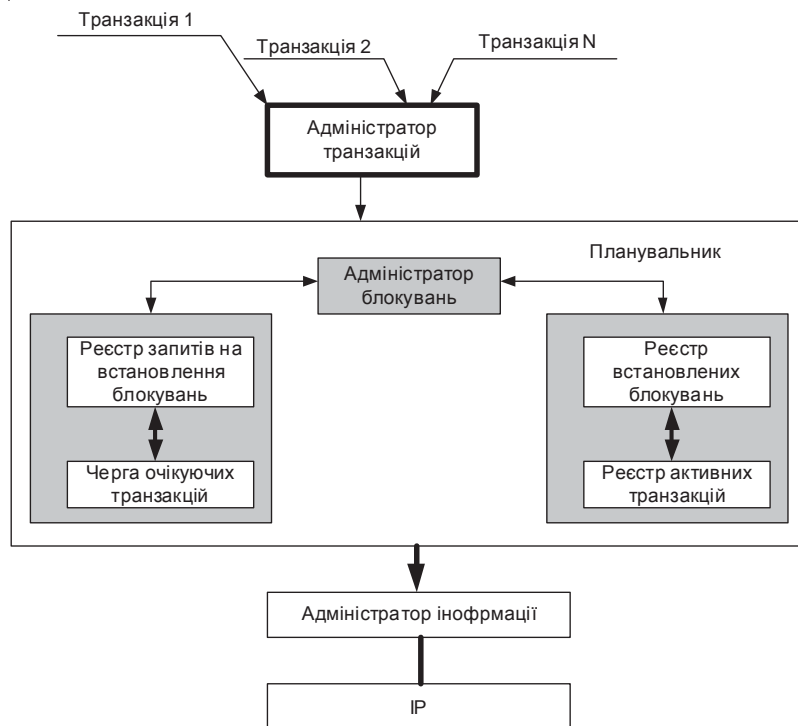


Рис. 1. Узагальнена модель централізованої СУІР

Управління цілісністю є невід’ємною функцією системи управління інформаційними ресурсами (СУІР) і повинне забезпечувати підтримку IP в узгодженому стані при колективному режимі роботи з паралельним маніпулюванням інформацією. Підтримка цілісності є дуже важливою навіть у випадку роботи системи з одним користувачем, оскільки кожне оновлення інформації в IP повинне задовольняти її структурі та семантичним обмеженням на елементи інформації.

У системах колективного користування виконання транзакції може починатися, перериватися та поновлюватися у довільні моменти часу. З погляду користувача це виглядає так, ніби множина транзакцій виконується паралельно, але істинний паралелізм можливий лише тоді, коли обчислювальна система об’єднує декілька центральних процесорів. Однак, у будь-якому випадку, чи присутній

реальний паралелізм виконання системних операцій, чи ні, неприпустимим є порушення роботи однієї транзакції іншою.

Паралельне виконання декількох транзакцій є коректним, якщо і тільки якщо їх спільний результат буде таким самим, як і при виконанні цих транзакцій послідовно в певному порядку.

Узгоджені стани інформаційних ресурсів забезпечують послідовні, або такі, що можуть бути зведені до послідовних розкладів.

Розглянемо можливі розклади, наведені в табл. 1 [6]. У табл. 1 зчитування й запис поля А позначено відповіддю Z_r -А і Z_n -А.

Таблиця 1

Можливі розклади транзакцій

Розклад 1	Розклад 2	Розклад 3	Розклад 4	Розклад 5
$T_1: Z_r$ -А	$T_1: Z_r$ -А	$T_1: Z_n$ -А	$T_1: Z_r$ -А	$T_2: Z_r$ -А
$T_1: Z_n$ -А	$T_2: Z_r$ -А	$T_1: Z_n$ -А	$T_2: Z_r$ -А	$T_1: Z_n$ -А
$T_2: Z_r$ -А	$T_1: Z_n$ -А	T_1 : скасувати	T_1 : перервати	$T_2: Z_r$ -А
$T_2: Z_n$ -А	$T_2: Z_n$ -А			

Розкладом сукупності транзакції називається порядок, у якому виконуються елементарні кроки (операції) цих транзакцій (читання, запис, блокування і т.д.). Природно, кроки будь-якої заданої транзакції повинні з'являтися у розкладі в порядку їх входження у загальну програму виконання. Розклад вважається послідовним, якщо усі кроки кожної транзакції виконуються послідовно і таким, що зводиться до послідовного, якщо його результат є еквівалентним результату деякого послідовного розкладу.

У першому розкладі (табл. 1) розузгодження інформації ІР не може виникнути, оскільки транзакції T_1 і T_2 виконуються послідовно одна за одною. У другому розкладі модифікація транзакції T_1 втрачається. У третьому розкладі також відбувається втрата модифікації T_2 , викликана тим, що після запису T_2 настає відміна транзакції T_1 . Ситуація розкладу 4 відома під назвою неправильного зчитування, оскільки вибране транзакцією T_2 значення після цього видаляється з ІР. Дві вибірки у розкладі 5 дають різні значення, оскільки транзакція T_2 переривається записом. Для обмеження порядку виконання транзакції необхідні протоколи, складність яких залежить від того, на якому рівні вимагається підтримувати цілісність при роботі з інформаційними ресурсами.

Для виявлення розкладів, що можуть бути зведені до послідовних, і завдяки цьому зберігають ІР в узгодженому стані, можна застосувати метод, що ґрунтується на побудові графа очікувань [6]. У цьому графі транзакціям відповідають вершини, а дуга, яка зв'язує вершини T_i та T_j , означає, що вхід T_j залежить від входу T_i , тобто T_j використовує значення поля, вироблене транзакцією T_i . Якщо граф очікувань містить контури, то це означає, що відповідний розклад не зводиться до послідовного отже, дає суперечливі результати. На рис. 2 подані графи очікувань, що відповідають розкладам, наведеним у табл. 2.

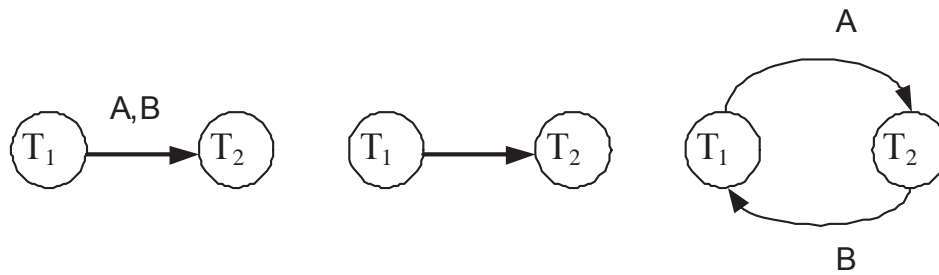


Рис. 2. Графи очікувань розкладів транзакцій, наведених в табл. 2

У табл. 2 Z_n інтерпретується як модифікація вхідними змінними. Розклад S_1 – послідовний (несуперечливий), S_2 – такий, що може бути зведений до послідовного (несуперечливий), а S_3 – містить контур (суперечливий).

Таблиця 2

Розклад транзакцій

Розклад S_1	Розклад S_2	Розклад S_3
$T_1: Z_n-A$	$T_1: Z_n-A$	$T_1: Z_n-A$
$T_1: Z_n-B$	$T_2: Z_n-A$	$T_2: Z_n-A$
$T_2: Z_n-A$	$T_1: Z_n-B$	$T_2: Z_n-B$
$T_2: Z_n-B$	$T_2: Z_n-B$	$T_1: Z_n-B$

Монопольне використання поля запису в процесі роботи СУІР можна забезпечити за допомогою блокування (захоплення). Для цього ІР розбивається на частини, тобто елементи, які можна блокувати в процесі здійснення операції обробки інформації. Блокуючи деякий елемент, транзакція може таким чином перешкоджати доступу до нього з боку інших транзакцій до моменту його розблокування.

Вид та розмір елементів повинен вибиратися в залежності від моделі та середовища функціонування ІР, так, щоб досягався певний компроміс між ступенем паралелізму виконання операцій і затратами системи на підтримку блокувань. Зокрема, у реляційній моделі ІР можна вибирати великі елементи, такі, як відношення, або малі, такі, як окремі кортежі чи навіть їх окремі поля; у мережній моделі, як елемент, може розглядатися сукупність усіх записів одного типу, або екземпляри наборів.

Вибір більших елементів скорочує накладні витрати системи на підтримку блокувань, тоді як вибір малих елементів створює можливість паралельного виконання багатьох транзакцій.

Функціональний компонент СУІР, який називається планувальником, призначає і реєструє блокування, а також виконує роль арбітра між кількома запитами, що вимагають доступу до одного й того ж елемента інформації (рис. 1).

У роботах [1–3] вказується, що при паралельній обробці інформації можливим є виникнення особливого стану, який називається нескінченним очікуванням, однак, це питання не є специфічним для систем інформаційних ресурсів.

Більш серйозна проблема – це так звані тупики або взаємоблокування, коли кожна із транзакцій, перебуваючи на стадії виконання, блокує елементи інформації, що їх потребує інша для його завершення. У принципі можливі тупикові ситуації з участю трьох і більше транзакцій, але, як показано в [7], на практиці дуже рідко трапляються тупики з участю більш ніж дві транзакції.

На практиці, не всі доступні сьогодні СУІР здатні достовірно виявляти тупикові ситуації. Деякі СУІР виконують детектування взаємоблокувань, регулярно перевіряючи час очікування тією чи іншою транзакцією доступу до запитуваного ресурсу. Детектування виникнення тупикової ситуації у таких системах відбувається, якщо транзакція не завершується тим чи іншим чином (тобто, із фіксацією або скасуванням результатів) протягом деякого наперед заданого проміжку часу. Реакцією на виникнення тупикової ситуації у деяких системах є автоматичний перезапуск транзакції спочатку за умови, що обставини, які призвели до тупикової ситуації, не повторяться знову в майбутньому. У прикладну програму, що ініціювала цю транзакцію, системою посилається повідомлення про те, що виконання такої транзакції завершилося помилкою, після чого прикладна програма повинна сама генерувати відповідну реакцію. Поза тим, незважаючи на наявність різних можливих підходів у вирішенні цієї проблеми, рекомендується завжди вирішувати її, враховуючи інтереси кінцевого користувача.

Інший метод детектування тупиків ґрунтується на побудові графа, у якому проводиться дуга від транзакції до запису, який відшукується, а потім – дуга від запису до транзакції, яка в цей момент його використовує. Якщо граф має контури, то це означає наявність взаємоблокувань (рис. 2).

Процедури детектування взаємоблокування завершують свою роботу, відмінюючи одну із взаємоблокованих транзакцій і проштовхуючи в черзі очікування іншу (рис. 3).

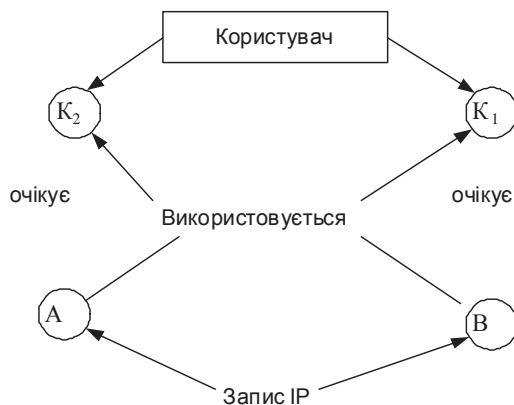


Рис. 3. Взаємоблокування транзакцій, виявлене за допомогою детектуючих контурів

Іншим варіантом контролю паралельної обробки є двофазне блокування. Вважають, що транзакція проводиться у відповідності із протоколом двофазного блокування, якщо усі операції блокування (блокування на читання, блокування на запис) передують першій операції розблокування в транзакції. Опція “блокування на читання” дозволяє зчитувати елемент інформації, тоді як “блокування на запис” дозволяє й зчитувати й оновлювати його. Іншими словами, двофазний протокол являє собою просто вимогу, відповідно до якої у кожній транзакції всі операції блокування повинні передувати усім операціям розблокування. Транз-

акції, що підпорядковуються цьому протоколу, мають назву двофазних. Перша фаза називається фазою блокування, а друга – фазою розблокування.

Як правило, множина усіх транзакцій, які можуть коли-небудь виконуватися паралельно з деякою заданою транзакцією, наперед невідома, адже вимушена вимога полягає у тому, щоб усі транзакції були двофазними. Однак при двофазному блокуванні можуть виникати ситуації взаємоблокування операцій обробки інформації, тому необхідно передбачати механізм їх виявлення і ліквідації.

В автоматизованих системах обробки інформації, у тому числі й Системах автоматизації проектування, в якості компонентів інформаційного забезпечення можуть використовуватись СУІР загального призначення. Однак, окрім того, що така СУІР може виявитись досить дорогою, вона може містити багато засобів, що є надлишковими в аспекті функціональних вимог цієї конкретної сфери застосування. Тим часом, засоби ці призводять до невиправданого зростання грошових та додаткових витрат ресурсів цільової інформаційної системи. Оскільки СУІБ загального призначення створюються для розв'язання широкого кола прикладних задач, вони часто не можуть конкурувати за продуктивністю зі спеціалізованими системами, спроектованими згідно вимог конкретної чітко визначеної задачі. Тому у багатьох випадках більш раціональною та виправданою є розробка спеціалізованої СУІР, а це у свою чергу вимагає розробки відповідного механізму керування процесами паралельної обробки інформації.

Процеси паралельної обробки, що стосуються операцій над інформаційними ресурсами, можуть описуватися різними моделями, які відрізняються, головним чином, ступенем деталізації опису режиму доступу до елементів інформації ІР, тобто типом застосовуваних блокувань.

Висновки

Таким чином, хоч проблема управління процесом паралельної обробки не є новою і деякі методи її вирішення реалізовані у комерційних СУІР, вона, як вказує визнаний фахівець із теорії проектування ІР Дейт К. Дж., є досить заплутаною через те, що при її розв'язанні необхідно враховувати багато факторів і не існує щодо цього однозначних рекомендацій. Тому проблема проектування процесів управління паралельною обробкою залишається актуальною. Аналіз показав, що задачі управління паралельною обробкою можуть бути формалізовані й розв'язані з використанням відповідних алгоритмів та програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Дейт К. Дж.* Введение в системы баз данных / Дейт К. Дж. – СПб. : Изд. дом “Вильямс”, 1999. – 848 с.
2. *Хансен Г.* Базы данных : разработка и управление / Г. Хансен, Д. Хансен. – М. : ЗАО “Изд. БИНОМ”, 1999. – 699 с.
3. *Чери Ст.* Логическое программирование и базы данных / Ст. Чери. – М. : Мир, 1992. – 352 с.
4. *Андреев В.І.* Стратегія управління інформаційною безпекою / В.І. Андреев, В.Д. Козюра, Л.М. Скачек, В.О. Хорошко. – К. : Вид. ДУІКТ, 2007. – 277 с.
5. *Cazey M., Stonebraker M.* The Performance of Concurrency Control Algorithms for DBMS in Proc. 10th Antil Conf. on Very Large Date Bases, p. 107–118. – Singapore, August, 1994.
6. *Озкарахан Э.* Машины баз данных и управление базами данных / Ozkaraхан Э. – М. : Мир, 1989. – 965 с.
7. *Капустяк М.В.* Модели передачи информации с учетом обнаружения, недопущения и устранения тупиковых ситуаций / М.В. Капустяк, Т.И. Олешко, В.А. Хорошко // Вісник ДУІКТ. – 2006. – Т. 4. – № 3. – С. 152–162.

Отримано 29.08.2012