

УДК 004.451

## ОЦІНКА ЕФЕКТИВНОСТІ МЕТОДУ ПРОГНОЗУВАННЯ ВЗАЄМОБЛОКУВАНЬ ПРОЦЕСІВ В КОМП'ЮТЕРНІЙ СИСТЕМІ

К. т. н. Ю. П. Кльоц, С. В. Мостовий

Хмельницький національний університет  
Україна, м. Хмельницький  
sprmostovuy@gmail.com

*В роботі проведено оцінку часової складності та ефективності методу прогнозування взаємоблокування процесів в комп'ютерній системі на прикладі вирішення взаємоблокування процесів у системі керування базами даних MySQL.*

*Ключові слова: взаємоблокування процесів, метод прогнозування взаємоблокувань, ефективність методу.*

При експлуатації комп'ютерних систем (КС) досить часто виникає ситуація блокування процесів, що виконуються у них. Частковим випадком блокування задач є можливість їх взаємного блокування [1]. Взаємне блокування – ситуація в багатозадачному середовищі або системах керування базами даних (СКБД), при якій кілька процесів перебувають у стані нескінченного очікування ресурсів, зайнятих самими цими процесами. Виникнення взаємних блокувань задач призводить до збільшення часу їхнього виконання (може зростати до нескінченності), до неефективного використання ресурсів КС (порожні цикли очікування).

Однією із проблем виконання процесів в КС є уникнення входження їх в стан взаємоблокування. Дані питання досліджувались Дейкстрою, Хоаром, Брінч-Хансеном, Деккером, Петерсенем, Коффманом, Хольтом [2]. На сьогодні розроблена велика кількість методів та алгоритмів для уникнення взаємоблокувань процесів. Проте частина з них носить лише теоретичний характер, оскільки не може бути реалізованою в сучасних КС. Інша частина при реалізації стає достатньо громіздкою і ресурсоемною. Тому розробники операційних систем (ОС) сімейств Windows та Linux, а також сучасних СКБД не включають відомі алгоритми уникнення взаємоблокувань процесів.

За умови завантаження системи процесами до 20% відсутність таких засобів була допустима. Проте стрімкий розвиток апаратних засобів, зростання об'єму та складності програмного забезпечення, яке займається вирішенням масштабних та відповідальних задач, не повинно дозволяти виникнення взаємоблокування, що в свою чергу вимагає розробки нових підходів до вирішення цієї задачі.

Для усунення суттєвих недоліків відомих методів та алгоритмів вирішення проблеми взаємоблокування було розроблено метод прогнозування взаємоблокування процесів, що враховує життєвий шлях процесу [3]. З метою визначення доцільності використання та порівняння запропонованого методу з існуючими засобами вирішення задачі взаємоблокування необхідно провести оцінку його часової складності та ефективності.

Для оцінки часової складності методу прогнозування взаємоблокувань процесів було використано web-сервер Apache2 з PHP 5 та sql-сервер MySQL. Для перевірки роботи методу на сервері запущався на виконання PHP-скрипт, представлений в лістинзі 1, який при паралельному виконанні призводить до виникнення взаємоблокувань.

Лістинг 1

```
$sql = "SELECT COUNT(*) FROM t "; $x = mysql_query($sql);  
$r = mysql_fetch_row($x); $max_id = $r[0];  
$id1 = rand(0, $max_id); do { $id2 = rand(0, $max_id); } while ($id1 == $id2);  
mysql_query("START TRANSACTION;");  
$sql1 = "select a from t where id = $id1 for update"; mysql_query($sql1);
```

```
//Вибір кортежу для обрахунку
usleep(100); //моделювання обробки даних
$sql2 = "select b from t where id = $id2 for update"; mysql_query($sql2);
mysql_query("COMMIT;")
```

В представленому коді видалено бізнес-логіку, зате повністю збережено послідовність запитів, що при паралельному виконанні призводять до появи взаємних блокувань.

Рівні взаємоблокувань, представлені на рис. 1, показують взаємоблокування в КС. Суцільною лінією показано рівень взаємоблокувань при використанні стандартних засобів MySQL для виявлення заблокованих транзакцій. Пунктирною лінією показано рівень взаємоблокувань, що виникали при використанні запропонованого методу прогнозування взаємоблокувань.

Час виконання процесів при використанні різних механізмів вирішення взаємоблокувань показано на рис. 2. Суцільною лінією показано середній час виконання процесів при використанні стандартного механізму вирішення взаємоблокувань, пунктирною – при використанні запропонованого методу прогнозування взаємоблокувань, штрих-пунктирною – середній час виконання процесів за умови ненастання взаємоблокувань.

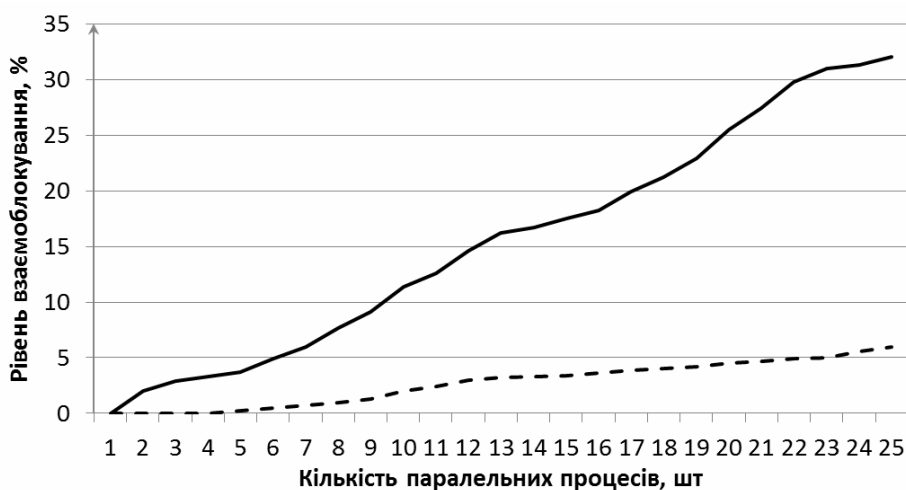


Рис. 1. Залежність рівня взаємоблокувань від кількості паралельних процесів

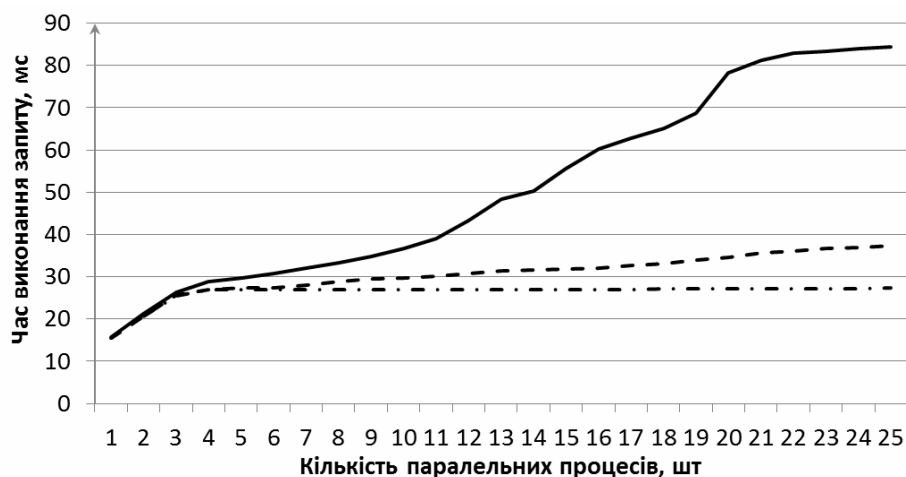


Рис. 2. Залежність часу обробки запитів від кількості паралельних процесів

У випадку використання стандартного механізму виявлення взаємоблокувань спостерігається значне зростання часу виконання навіть при незначній кількості паралельних процесів. В цей самий період не спостерігається значного завантаження процесора (не більше 20%), оскільки процеси більшу частину часу очікують доступу до заблокованих ресурсів.

У випадку, коли не виникає взаємоблокувань процесів, час їх виконання зростає незначно (на 3% при кількості паралельних процесів 25 і рівні завантаження процесора не більше 20%).

Використання запропонованого методу прогнозування взаємоблокувань показує помітно полігше наростання середнього часу виконання процесів за таких самих умов.

Зниження часу виконання процесів при виконанні єдиного потоку пояснюється відсутністю затримок, пов'язаних з очікуванням звільнення ресурсу.

В ході дослідження було проведено порівняння часу виконання фаз транзакцій при максимальному завантаженні системи і різних режимах роботи, що представлено на рис. 3.

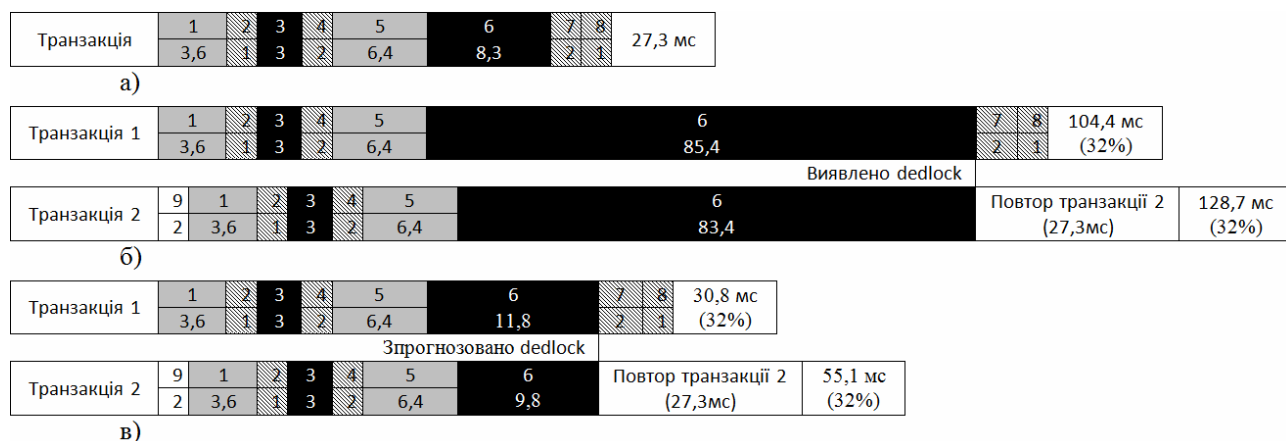


Рис. 3. Часові діаграми виконання фаз транзакції без взаємоблокування (а), із стандартним виявленням взаємоблокування (б) та із прогнозуванням взаємоблокування (в):

1 – підготовчі дії; 2 – запуск транзакції; 3 – очікування звільнення кортежу id1; 4 – виконання 1-го SELECT; 5 – опрацювання даних; 6 – очікування звільнення кортежу id2; 7 – виконання 2-го SELECT; 8 – завершення транзакції; 9 – затримка від початку запуску

На рис. 3, а представлено часову діаграму виконання фаз транзакції за умови, що при її виконанні не відбулось взаємоблокування. Середній час виконання транзакції склав 27,3 мс, з них 11,3 мс (41%) зайняло очікування заблокованих ресурсів. В ході дослідження таких транзакцій виявилось 36%.

На рис. 3, б представлено часову діаграму взаємоблокування двох транзакцій із стандартним механізмом їх вирішення. Транзакція 1 при першому звертанні (4) до таблиці БД вибирає та блокує кортеж із ключем id1. Транзакція 2 починає виконуватись на 2 мс пізніше, ніж транзакція 1, і при першому звертанні (4) вибирає та блокує запис із ключем id2. Після опрацювання отриманих даних (5) транзакція 1 звертається до кортежу з id2. Оскільки він заблокований, транзакція переходить до очікування звільнення ресурсу (6). В свою чергу, транзакція 2 звертається до кортежу з id1. Він також виявляється заблокованим, і транзакція 2 переходить в режим очікування звільнення ресурсу (6). Процеси потрапляють в ситуацію взаємного блокування і самостійно не можуть вийти з циклу нескінченного очікування. Взаємоблокування усуває КС, яка періодично аналізує граф процесів і ресурсів. Оскільки задача є алгоритмічно складною, вона виконується через інтервали часу в 150–200 мс. Після виявлення взаємоблокування один із процесів, що пізніше надійшов у систему, примусово завершується і повторно запускається на виконання. Інший із заблокованих процесів продовжує своє виконання. Великі часові проміжки між повторними аналізами графа призводять до значних затримок при виявленні взаємоблокувань. Середній час процесу, який був виконаний повторно, складає 128,7 мс, процесу, який продовжив роботу після взаємного блокування – 104,4 мс. В ході дослідження таких транзакцій виявилось по 32%, оскільки процеси у взаємоблокування потрапляють парами.

На рис. 3, в представлено часову діаграму взаємоблокування двох транзакцій із прогнозуванням взаємоблокувань. До моменту запиту в транзакції 2 кортежу з ключем id2 часові діаграми ідентичні.

В момент цього запиту (б) транзакція 2 потрапляє в граничний стан, і для неї проводиться прогнозування. Час прогнозування складає близько 10 мс. В результаті прогнозу визначається ймовірність взаємоблокування процесів, після чого обирається один із них. Цей процес знімається з виконання і запускається повторно. Середній час виконання процесу, що був виконаний повторно, складає 55,1 мс; процесу, що продовжив роботу після взаємного блокування – 30,8 мс. В ході дослідження таких транзакцій виявилось по 32%.

З отриманих досліджень середній час виконання транзакцій із стандартним механізмом виявлення взаємоблокувань склав  $27,3 \cdot 0,36 + 104,4 \cdot 0,32 + 128,7 \cdot 0,32 = 84,4$  мс, із прогнозуванням взаємоблокувань  $27,3 \cdot 0,36 + 30,8 \cdot 0,32 + 55,1 \cdot 0,32 = 37,3$  мс, тобто застосування методу прогнозування взаємоблокувань забезпечило зменшення часу виконання процесів в 2,3 рази. Це дозволяє більш ефективно використовувати ресурси КС.

Таким чином, в роботі запропоновано оцінку часової складності та ефективності методу прогнозування взаємоблокувань процесів в КС. Проведено його експериментальне дослідження на прикладі модифікованої СКБД MySQL. Середній час виконання процесу зменшився з 84,4 мс при використанні стандартних засобів MySQL до 37,3 мс при використанні запропонованого методу. Отримані результати вказують на зменшення часових витрат на виконання процесів, що в свою чергу дозволяє опрацьовувати в 2,3 рази більшу кількість даних в одиницю часу для задач, в яких часто виникають взаємоблокування (відбувається конкурентна боротьба за ресурси).

#### ВИКОРИСТАНІ ДЖЕРЕЛА

1. Kaveh N., Emmerich W. Deadlock detection in distribution object systems // Software Engineering Notes.– 2001.– Vol. 26, N 5.– P. 44–51.
2. Coffman E. G., Elphick M. J., Shoshani A. System deadlocks // Computing Surveys.– 1971.– Vol. 3, N 2.– P. 67–78.
3. Савенко О. С., Мостовий С. В. Прогнозування потрапляння процесів у стан взаємоблокування в комп'ютерних системах // Труды XIII Междунар. науч.-практич. конфер. «Современные информационные и электронные технологии».– Украина, г. Одеса, 2012.– С. 74.

---

Yu. P. Klyots, S. V. Mostovoy

#### **The efficiency estimation of the method for prediction of process deadlocks in a computer system.**

In the paper the estimation is given of time complexity and efficiency of the method for prediction of process deadlocks in a computer system. The estimation is based on the solution of deadlock processes in the database management system MySQL.

Keywords: *deadlock, method for deadlock prediction, method efficiency.*