

УДОСКОНАЛЕННЯ АЛГОРИТМУ ПОВНОГО ГРУПОВОГО РОЗСИЛАННЯ У СИСТЕМАХ ОБРОБЛЕННЯ ДАНИХ З РОЗПОДІЛЕНОЮ АРХІТЕКТУРОЮ

© Кришталь Р. П., Лаврів О. А., Хархаліс З. М., 2016

Поступове зміщення від індивідуальних до спільних платформ надання сервісів формує нові завдання, пов'язані з їх належним функціонуванням. У роботі здійснено моделювання та проведено дослідження роботи методу повного впорядкованого групового розсилання. Наведено основи побудови логічних годинників. У результаті аналізу виявлено, що наявна реалізація методу не придатна для процесів з non-FIFO чергами подій, тобто не здатна адекватно функціонувати в сучасних умовах високих інтенсивностей надходження запитів на обслуговування. За результатами проведеного аналізу запропоновано удосконалений метод повного впорядкованого групового розсилання, здійснено моделювання його роботи та показано, що запропоновані зміни дають змогу знизити імовірність відмови в сучасних розподілених системах надання сервісів.

Ключові слова: розподілена система, платформа надання сервісів, повна групова розсилка.

R. Kryshstal, O. Lavriv, Z. Kharkhalis
Lviv Polytechnic National University

IMPROVING OF TOTALLY ORDERED MULTICAST ALGORITHM FOR DISTRIBUTED ARCHITECTURE DATA PROCESSING SYSTEMS

© Kryshstal R., Lavriv O., Kharkhalis Z., 2016

The gradual shift from individual to shared service delivery platforms brings distributed service delivery systems to the forefront and the problems associated with their proper functioning. Performance of totally ordered multicasting method was modeled and studied in this paper. The analysis revealed that current realization of the method is not suitable for processing non-FIFO-queue events, thus is not able to function adequately in modern high-intensity flow of service requests. Accordingly, improved method is proposed. The block diagram of this method is depicted in the paper. The simulation process is specified. Performance of modified totally ordered multicasting method was modeled according to this process and it is shown that proposed changes allow decreasing of failure rate in modern distributed service delivery systems. The paper shows the use of totally ordered multicasting method in real systems is complicated by the fact that it only works well in the channel FIFO. Because of this disadvantage, we propose improved method of totally ordered multicasting, which enables the synchronization process to survive, even if the message of sender process is lost. The improvement is to limit the waiting time for a response from all the processes addressed. This is to reduce the probability of failure of the process. However, the signaling information amount increases that circulates in the data network.

Key words: distributed system, service delivery platform, totally ordered multicast.

Вступ

Сфера інформаційних технологій характеризується швидким розвитком використовуваних засобів і методів оброблення даних. Бурхливий розвиток обчислювальних систем і високі вимоги до

технічних та інформаційних ресурсів привели до необхідності використання розподілених систем [1]. Одним з елементів таких систем є розподілені бази даних, основна особливість функціонування яких полягає в тому, що для користувача не має значення, де просторово розташовані необхідні йому дані [2]. Це досягається за рахунок автоматичної реплікації даних, яка дає змогу підтримувати дані в актуальному стані. Проте використання розподілених баз даних породжує проблеми розподіленої обробки даних, оскільки використання розподілених обчислень передбачає щонайменше територіальну віддаленість процесів один від одного. При цьому завдання, пов'язані з розподіленою обробкою даних, можуть виконуватися кількома послідовними етапами в різні періоди часу. Тому виникають проблеми синхронізації процесів між собою та узгодженості даних на розподілених сховищах даних.

Варто зауважити, що для спільної роботи процесів їм необхідний деякий механізм взаємодії. У середовищі паралельних систем зазвичай вважається, що взаємодія процесів забезпечується за допомогою спільних змінних [3]. У цьому випадку один процес здійснює запис в одну з таких змінних, а інший – зчитує записане значення. Своєю чергою, у розподілених системах процеси взаємодіють, пересилаючи повідомлення. Тому для обміну даними або координації спільних дій процеси повинні використовувати механізми відправлення і приймання повідомлень.

Дуже часто має значення не точний час настання певної події (наприклад, надсилання та отримання повідомлення), а послідовність, в якій ці події відбувалися, тобто потрібно визначити, відбулася певна подія в цьому процесі до або після іншої події в іншому процесі [4, 5]. У цих випадках виконання процесів можна описувати в термінах послідовності подій за умов відсутності знання точного часу їх настання.

Зусилля провідних учених у сфері інформаційно-телекомунікаційних технологій зосереджені на удосконаленні взаємодії сервісних процесів на рівні застосунків. Це зумовлено тим, що якість сприйняття сервісу визначається не лише параметрами мережі передавання даних, а й особливостями взаємодії компонентів композитного сервісу. Часто для цього застосовується метод повного впорядкованого групового розсилання повідомлень між процесами елементарних сервісів. Проте, у його роботі не враховано того, що повідомлення у процесі їх передавання можуть втрачатись. В реальних системах це призводить до збою у процесі обміну повідомлення, і як наслідок, до відмови у наданні сервісу. Тому вдосконалення цього методу є актуальним завданням. Це дасть змогу знизити імовірність відмови у наданні композитних сервісів у сучасних розподілених системах.

Дослідження методу повного впорядкованого групового розсилання

Для практичного дослідження методу повного впорядкованого групового розсилання [6] розглянемо два канали зв'язку між процесами, щодо яких здійснювалася комунікація: канал FIFO та non-FIFO.

Канал, який зберігає порядок переданих по ньому повідомлень, називається чергою, або каналом FIFO (англ. First In First Out). Це означає, що якщо повідомлення m_1 надіслано раніше, ніж повідомлення m_2 по тому самому каналу, то отримано повідомлення m_1 також буде раніше, ніж повідомлення m_2 .

Non-FIFO канал можна подати у вигляді множини, в яку процес-відправник додає повідомлення, а процес-адресат вилучає їх у випадковій послідовності.

На рис. 1 показано процес оновлення баз даних для каналів FIFO та non-FIFO. Згідно з наведеними результатами можна стверджувати, що у разі використання каналу non-FIFO алгоритм повного впорядкованого групового розсилання втрачає свою ефективність, оскільки значення часових позначок повідомлень, які надійшли у систему і перебувають в черзі, невпорядковане, що призведе до неузгодженості бази даних. Проте у разі використання каналу FIFO спостерігається впорядкована черга повідомлень, у якій на кожному наступному повідомленні значення часової позначки більше від попередньої.

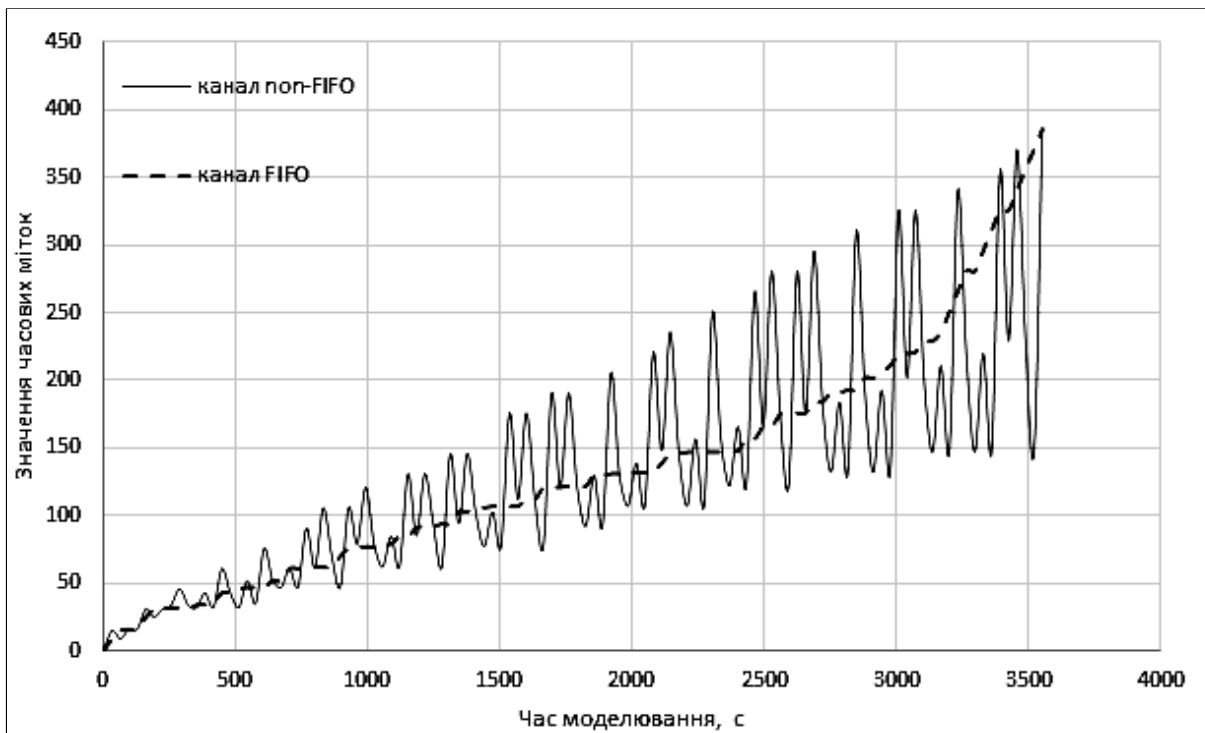


Рис. 1. Процес оновлення баз даних для каналу FIFO та non-FIFO з використанням алгоритму повного групового розсилання

На рис. 2 подано середню тривалість синхронізації процесів у системі протягом часу моделювання для каналу зв'язку між процесами FIFO і non-FIFO. Отримані результатами показують, що алгоритм повного впорядкованого групового розсилання втрачає ефективність з використанням каналу non-FIFO, оскільки діапазон часу синхронізації для такого каналу тримається у межах 1700–3200 мс, а з використанням каналу FIFO – в межах 700 мс. Також варто зазначити, що під час моделювання системи з використанням каналу non-FIFO не враховано втрату повідомлень.

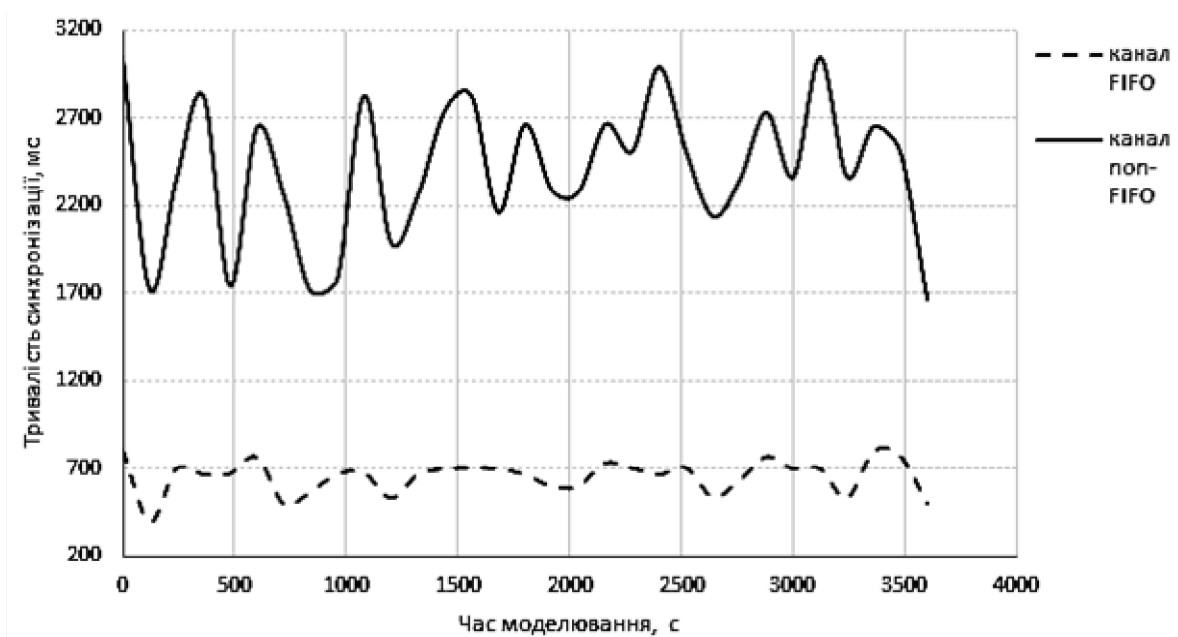


Рис. 2. Середня тривалість синхронізації протягом періоду моделювання для каналу FIFO та non-FIFO (без урахування можливості втрати повідомлення)

Згідно з отриманими результатами можна зробити висновки, що цей метод погано адаптований до використання в реальних системах, оскільки основною його вимогою є застосування каналу FIFO. Такий канал, здебільшого, неможливо реалізувати на практиці.

Удосконалення методу повного групового розсилання, яке ґрунтується на модельному підході синхронізації, дає можливість використовувати його у каналах зв'язку non-FIFO. Роботу модифікованого алгоритму можна охарактеризувати так. У разі надходження запиту на зміну даних у базі процес посилає широкомовне повідомлення всім процесам, які наявні в системі. У повідомленні передаються такі параметри: значення часової мітки, ідентифікатор процесу, а також порядок цього повідомлення у локальній черзі процесу-відправника. Можливість використання останнього параметра забезпечують алгоритми взаємного виключення, які впорядковують запити до процесу запису/зчитування даних з розподіленої бази. Отримавши запит, решта процесів аналізують вміст отриманого повідомлення, де параметр порядку в локальній черзі процесу-відправника має першочерговий пріоритет.

На основі цього параметра визначається порядок у локальній черзі процесів-отримувачів повідомлення. Проте може настати такий момент, коли два чи більше процеси надсилають повідомлення з однаковими ідентифікаторами порядку, тому в такому випадку конфлікт буде вирішуватися за допомогою часових позначок процесів.

Після того, як процес опрацював запит і помістив його у свою локальну чергу повідомлень, він надсилає повідомлення-підтвердження до відправника запиту. Варто уточнити, що ідентифікатор процесу-відправника міститься в повідомленні. Процес, який ініціював повідомлення-запит, очікує підтвердження від усіх процесів, наявних у системі. Підтвердження дають можливість визначити, чи не втрачено повідомлення під час передавання по каналу зв'язку. Також потрібно зазначити, що тривалість очікування обмежена. Якщо протягом цього встановленого часу всі повідомлення-підтвердження не надходять, то таке повідомлення вважається втраченим. Після того процес-відправник надсилає повідомлення до процесів, від яких уже встигло надійти підтвердження, про видалення цього повідомлення з черги.

Описаний метод групового розсилання має і недоліки:

- у разі втрати повідомлення кількість пакетів зростає, що збільшить навантаження на мережу;
- втрата повідомлення на видалення з локальної черги процесів не розглядається і вважається, що якість каналу зв'язку достатня для передавання повідомлення-підтвердження.

Дослідження удосконаленого методу повного групового розсилання

З метою дослідження проведено імітаційне моделювання процесу обміну повідомлення на прикладному рівні між процесами розподіленої системи.

Коротко процес моделювання можна подати так:

- існує розподілена база даних (наявність декількох реплік бази даних, які територіально рознесені);
- процеси в системі обмінюються повідомленнями. Вони є абстракцією над механізмом опрацювання даних.

Мета моделювання – забезпечення узгодженості реплік розподіленої бази даних за допомогою впорядкування повідомлень, які надходять у систему.

Модель подано у вигляді програмної реалізації на С# з використанням мультипотокості. В цьому випадку йдеться про умовну мультипотокість порівняно з реальною розподіленою системою, оскільки потоки на одному комп'ютері мають певні особливості [6]. У цьому полягає обмеженість загальності результатів моделювання. Також у моделюванні не розглядається втрата повідомлення-підтвердження.

Удосконалений метод повного групового розсилання представлено на рис. 3. Цей метод, на відміну від відомого, передбачає обмеження допустимої тривалості очікування відповіді від

процесів-отримувачів на запит від процесу-відправника. Це дає змогу в процесі координації сервісних процесів уникнути ситуації безмежного очікування, або ж “зависання” процесів. Тобто натомість буде здійснено повторну спробу координації, що дасть змогу користувачеві у допустимий період отримати реакцію на свої дії у розподіленій системі.

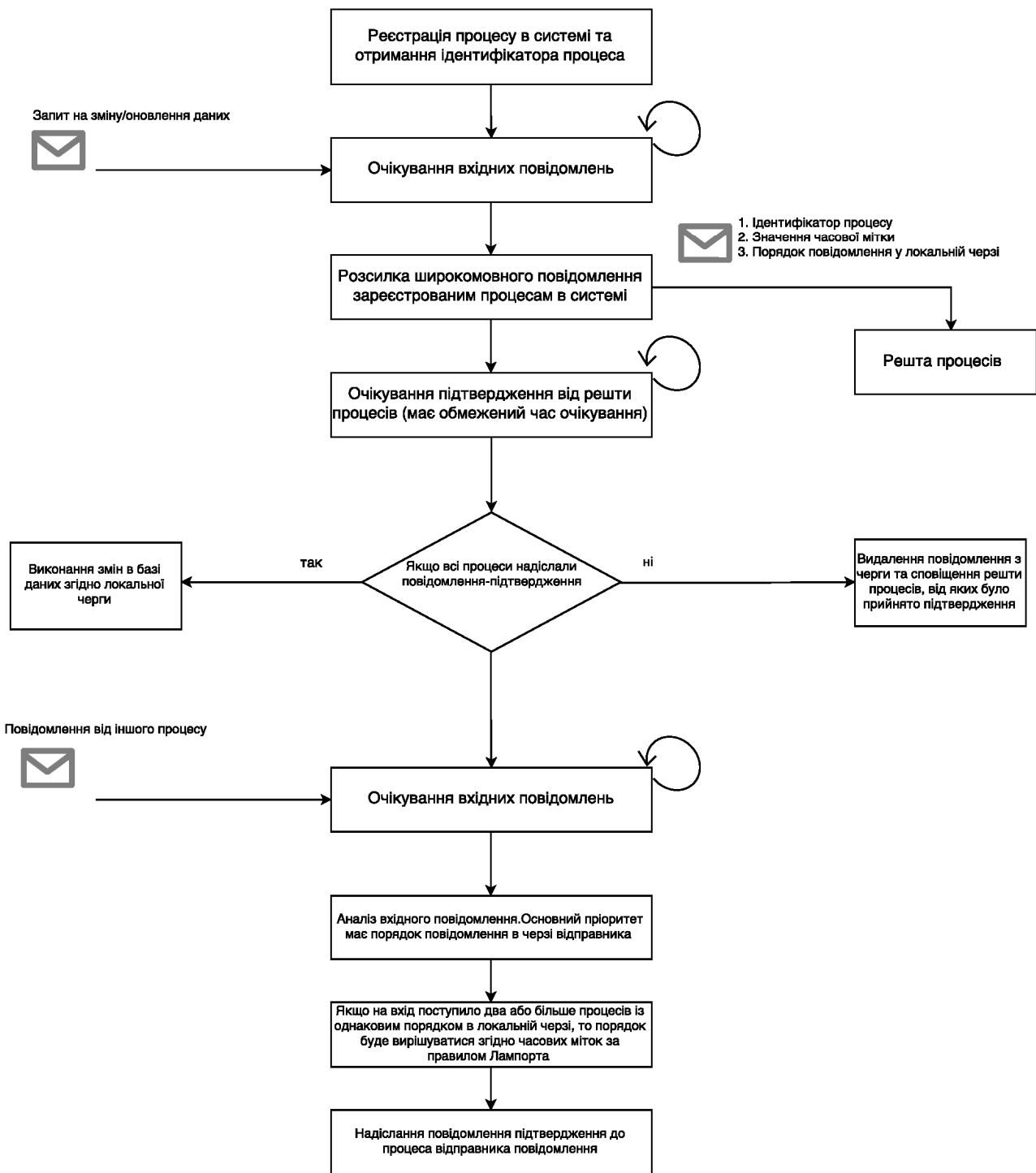


Рис. 3. Удосконалений метод повного групового розсилання

На рис. 4 зображено процес оновлення баз даних для базового та удосконаленого методів повного впорядкованого групового розсилання з використанням каналу non-FIFO. З отриманих результатів можна побачити, що модифікований метод повного групового розсилання ефективний у разі використання каналу зв'язку non-FIFO порівняно з базовим. Для модифікованого алгоритму

значення часових позначок в черзі постійно зростає і однакових значень немає, що задовольняє вимоги синхронізації часу в розподілених системах і приводить до узгодженості реплік баз даних.

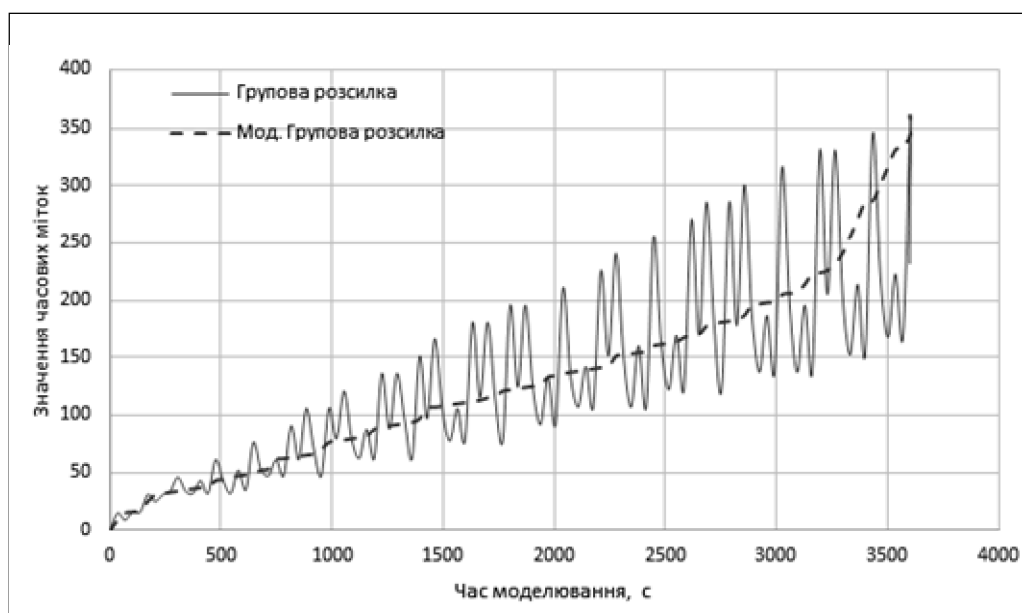


Рис. 4. Процес оновлення баз даних з використанням базового та удосконаленого алгоритму повного групового розсилання для каналу зв'язку pop-FIFO

На рис. 5 представлено середню тривалість синхронізації між процесами протягом часу моделювання з використанням базового та удосконаленого методів повного групового розсилання для каналу зв'язку pop-FIFO. Для моделювання згенеровано втрату одного пакета, щоб наочно показати відмінність між досліджуваними методами. Отримані результати свідчать, що у разі втрати повідомлення базовий метод припиняє роботу, відповідно час синхронізації різко зростає і процес опрацювання повідомлень припиняється. Проте у разі втрати повідомлення системи з використанням удосконаленого методу процес синхронізації реплік баз даних продовжується, а в момент часу втрати повідомлення різко збільшується час синхронізації. Це можна пояснити тим, що для модифікованого алгоритму встановлений час очікування повідомлення-підтвердження від решти процесів.

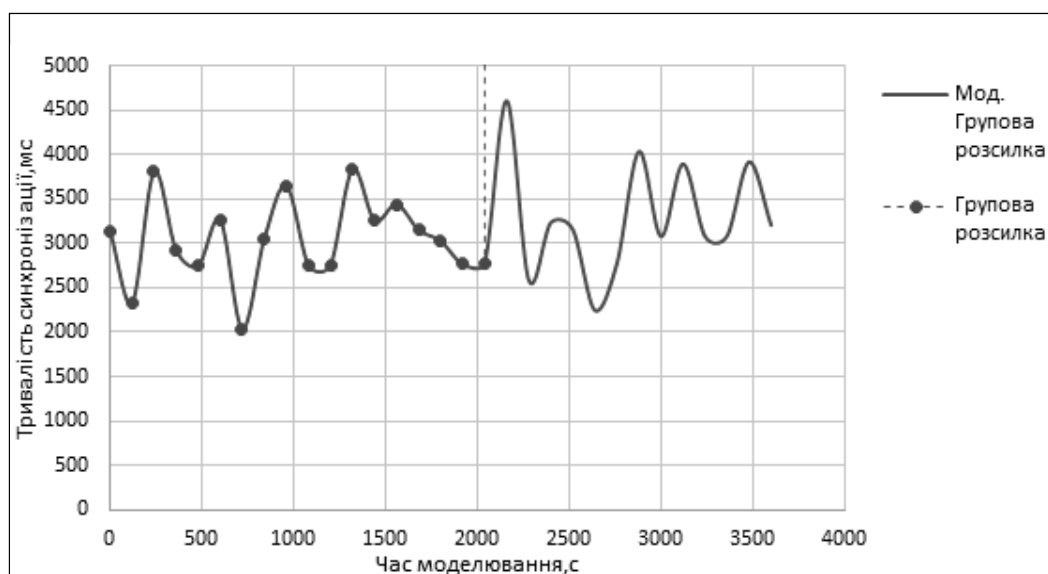


Рис. 5. Середня тривалість синхронізації за час моделювання для двох методів з використанням pop-FIFO каналу (з втратою одного повідомлення)

Висновки

Проведено дослідження методу повного впорядкованого групового розсилання в розподілених базах даних із синхронізацією на основі логічного годинника Лампорта. Проаналізовано його ефективність у каналах зв'язку типу FIFO та non-FIFO. Встановлено, що використання цього методу в реальних системах ускладнене тим, що він задовільно працює лише в каналі FIFO. У зв'язку з цим запропоновано удосконалений метод повного впорядкованого групового розсилання, який дає можливість зберігати працездатність процесу синхронізації навіть за умови, що повідомлення процесу-відправника втрачається. Удосконалення полягає у обмеженні часу очікування на відповідь від усіх процесів-адресатів. Це забезпечує зниження імовірності відмови процесу. Проте зростає кількість службової інформації, яка циркулює у мережі передавання даних.

1. Таненбаум Э., М. ван Стеен. *Распределенные системы. Принципы и парадигмы*. Санкт-Петербург: Питер. 2003. 877 с.: ил. 2. Топорков В. В. *Модели распределенных вычислений*. Москва: Физматлит. 2004. 320 с. 3. Fidge C. *Logical time in distributed computing systems* // *IEEE Computer*. August, 1991. Vol. 24, No. 8. P. 28–33. 4. Fowler J., Zwaenepoel W. *Causal distributed breakpoints* // *Proceedings of the 10th International Conference on Distributed Computing Systems*. 1990. P. 134–141. 5. Lamport L. *Time, clocks and the ordering of events in a distributed system* // *Communications of the ACM*. 1978. No. 21. P. 558–564. 6. Rauber T., Rüniger G. *Parallel Programming For Multicore and Cluster Systems*. Springer. 2013. 516 p. 7. Singhal M., Kshemkalyani A. *An efficient implementation of vector clocks* // *Information Processing Letters*. August. 1992. No. 43. P. 47–52.

References

1. Tanenbaum, E. and van Steen, M. (2003), *Distributed Systems. Principles and Paradigms*, Piter. 2. Toporkov, V. (2004), *Distributed computing models*, Fizmalit. 3. Fidge, C. (1991), "Logical time in distributed computing systems", *IEEE Computer*, Vol. 24, No. 8, pp. 28–33. 4. Fowler, J. and Zwaenepoel, W. (1990), "Causal distributed breakpoints", *Proceedings of the 10th International Conference on Distributed Computing Systems*, pp.134–141. 5. Lamport, L. (1978), "Time, clocks and the ordering of events in a distributed system", *Communications of the ACM*, no. 21, pp. 558–564. 6. Rauber, T. and Rüniger, G. (2013), *Parallel Programming For Multicore and Cluster Systems*, Springer. 7. Singhal, M. and Kshemkalyani, A. (1992), "An efficient implementation of vector clocks", *Information Processing Letters*, no. 43, pp. 47–52.