Вступ

Інформаційні системи поступово розширюють свою дію від простих процесів накопичення та зберігання даних до процесів управління організацією чи бізнес-системою, від управління потоками даних (workflow management) до управління бізнес-процесами (business process management). Відповідно до цього зростає важливість використання методів аналізу процесів функціонування систем замість методів аналізу даних, а для інформаційних систем, спрямованих на аналіз процесів, з'явився новий термін «процесно-орієнтовані (process-aware) інформаційні системи» [1]. Основним формалізмом, який використовується для аналізу процесів, є мережі Петрі [2]. Для того є такі причини: зручність графічного представлення мережі Петрі, її математичний формалізм, високий рівень деталізації представлення процесів функціонування дискретно-подійних систем, різноманітні розроблені засоби аналізу мережі Петрі.

Алгоритми імітації дискретно-подійних систем використовуються як складові компоненти інформаційних систем для оптимізації управлінських процесів та процесів прийняття рішень. Суттєві затримки обчислень в таких системах призводять до негативних наслідків в роботі технічних систем, які вони обслуговують. Моделі сучасних транспортних, виробничих, фінансово-економічних, еколого-економічних систем містять опис сотень елементів і тисяч подій, тому підвищення швидкодії алгоритмів імітації є важливою задачею.

Петрі-об'єктний підхід зменшує час виконання алгоритму імітації за рахунок переходу від перегляду стану елементарних переходів мережі Петрі до перегляду стану груп таких переходів, об'єднаних у змістові об'єкти моделі. Алгоритм імітації Петрі-об'єктної моделі викладений у публікації [3] і є універсальним для класу дискретно-подійних систем. У роботі [4] отримана оцінка складності цього алгоритму аналітичним способом та доведена її справедливість експериментально. Задача розпаралелювання цього алгоритму поставлена в даному дослідженні.

© Стеценко І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

Безперечними перевагами паралельних обчислень є швидкість, ефективність використання обчислювальних ресурсів, (гіпотетична) можливість досягнення лінійної залежності часу виконання обчислень від складності моделі, можливість реалізації обчислень в розподілених системах. Технології паралельних обчислень потребують спеціальної розробки алгоритму, який, по-перше, поділяє усі операції на ті, що можуть бути виконані послідовно, та ті, що можуть бути виконані паралельно, по-друге, визначає окремо послідовність дій кожного потоку. У прикладних задачах для досягнення значного ефекту паралельні алгоритми будуються з урахуванням специфіки вирішуваної задачі, використовуючи декомпозицію простору даних задачі (domain decomposition) або функціональну декомпозицію на підзадачі (functional decomposition) [5].

Розробка паралельних алгоритмів імітації є складною задачею, оскільки потребує контролю за станом моделі в часі. Імітаційна модель виконує упорядковану в часу послідовність подій і будь-яке порушення послідовності подій є порушенням логіки функціонування моделі. Існуючі підходи до побудови паралельних алгоритмів імітації ґрунтуються на розбитті функціонування моделі на логічні процеси, які взаємодіють між собою [6]. Корегування роботи процесів здійснюється так, щоб не порушити упорядкованість усіх подій моделі в часі. Оскільки Петрі-об'єктна модель конструюється з об'єктів, динаміка яких описується стохастичною мережею Петрі, то динаміка її від самого початку створення розбита на елементарні процеси і не потребує штучного розбиття для розпаралелювання. Іншою відмінністю побудованого алгоритму паралельної імітації є те, що кожен Петрі-об'єкт використовує локальну змінну часу (замість загальної для всієї моделі), узгоджуючи її просування тільки з об'єктами, які безпосередньо з ним взаємодіють.

Паралельні обчислення та алгоритми імітації

Багатопоточна технологія java містить широкі можливості для паралельного програмування: синхронізовані методи та фрагменти програм, засоби взаємодії потоків, блокуючі об'єкти [7]. Бібліотеки паралельного програмування містять певні засоби автоматизації паралельних обчислень такі, як розпаралелювання дій циклу або паралельне сортування елементів масиву. Проте їх безпосереднє застосування до алгоритму імітації не дають значного ефекту і, навіть, сильно сповільнюють його роботу. Це пояснюється тим, що в алгоритмі імітації такі дії часто повторюються (при кожному просуванні модельного часу), і кожного разу заново створюються нові потоки для кожного розпаралеленого циклу чи розпаралеленого сортування. В результаті, час, який витрачається на створення потоків, не покривається прискоренням, здобутим за рахунок розпаралелювання.

© Стеценко І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

У роботі [8] наведена класифікація існуючих способів розпаралелювання алгоритмів імітації в залежності від рівня, на якому застосовується паралелізм: програмний застосунок (application level), функціонування імітаційної моделі (functional level), її алгоритмізація (event level). Усі три способи використовують на різних етапах процесу моделювання: на етапі виведення результатів моделювання при програмуванні візуальних середовищ імітаційного моделювання, на етапі експериментування з імітаційною моделлю при проведенні повторюваних прогонів моделі, на етапі реалізації моделі при програмуванні імітаційного алгоритму. Перші два способи успішно використовуються у прикладних програмах, а третій вимагає спеціальної розробки програми, що враховує особливості конкретної моделі. При паралелізмі імітаційного алгоритму модель розбивають на фрагменти, які можуть певний час виконуватись незалежно один від одного, і запускають їх на одночасне виконання.

Контроль за просуванням часу в імітаційних алгоритмах потребує розробки механізмів синхронізації. В роботі [9] наводяться консервативний та оптимістичний [6] способи. В обох способах фрагменти моделі обмінюються між собою повідомленнями про час наступної найближчої події. При консервативному способі кожен фрагмент знаходиться в очікуванні, доки від інших не надійшла інформація про час наступної події. Тобто синхронізація відбувається за рахунок очікування, що призводить до неефективного використання паралелізму. При оптимістичному - фрагмент повертається у відповідний минулий стан, якщо від іншого фрагменту надійшла інформація про подію, яка виконана до поточного моменту часу. Тобто синхронізація відбувається за рахунок відміни вже виконаної події, що потребує запам'ятовування попереднього стану елементів моделі. Такий спосіб є більш ефективним з точки зору часу виконання, але є менш ефективним з точки зору використання ресурсів пам'яті.

© Стеценко І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

Алгоритм імітації Петрі-об'єктної моделі

Петрі-об'єктна модель складається з об'єктів, динаміка яких описується стохастичною мережею Петрі з багатоканальними та конфліктними переходами. Математичні рівняння такої мережі виведені в [10] і доведено, що вони містять в собі, як окремий випадок, рівняння детермінованої мережі Петрі та рівняння базової мережі Петрі. Поєднання Петрі-об'єктів в моделі відбувається двома можливими способами: з використанням спільної позиції або з використанням ініціалізації подій. Як доведено в [11], таке з'єднання забезпечує, що динаміка всієї моделі описується стохастичною мережею Петрі. Цей факт гарантує обчислюваність моделі, а з математичних рівнянь стохастичної мережі Петрі слідує оцінка обчислювальної складності Петрі-об'єктної моделі у вигляді [4]:

$$O\left(n \cdot q \cdot v \cdot time \cdot \left(v \cdot \left(n \cdot v + k^2 + k + v\right) + q^2 + q\right)\right), \qquad (1)$$

де $time$ - час моделювання, n - середня кількість переходів одного Петрі-об'єкта, v - середня кількість активних каналів переходу мережі Петрі в одиницю часу, q - кількість Петрі-об'єктів, k - середня кількість конфліктних переходів одного Петрі-об'єкта.

У роботі [4] представлено виведення оцінки обчислювальної складності Петрі-об'єктної моделі та результати експериментальних досліджень, що її підтверджують. Для великих систем кількість переходів одного Петрі-об'єкта значно менша за кількість об'єктів моделі, тобто $n << q$. Окрім того, завжди виконано $k \le n$. Тому для великих систем вираз (1) зростає не швидше ніж $O\left(n \cdot q^3 \cdot v^2 \cdot time\right)$. Отже, обчислювальна складність Петрі-об'єктної моделі зростає пропорційно кубу кількості її Петрі-об'єктів.

Послідовний алгоритм імітації Петрі-об'єктної моделі без конфлікту об'єктів складається з таких дій:

1. Визначити Петрі-об'єкт O_{min} з найменшим значенням часу найближчої події t_{min}.
2. Просунути модельний час у момент найближчої події t_{min} і виконати подію: вихід маркерів з переходів Петрі-об'єкта O_{min}, вхід маркерів в усі Петрі-об'єкти моделі.
3. Дії 1,2 повторювати, доки модельний час менший за кінець інтервалу моделювання.

Якщо вихід з різних Петрі-об'єктів здійснюється одночасно у спільну позицію, яка є вхідною для інших Петрі-об'єктів, то виникає конфлікт об'єктів, оскільки від того, який саме Петрі-об'єкт здійснить першим вихід маркерів, залежить подальше функціонування моделі. Конфлікт розв'язується вибором одного об'єкта з рівною ймовірністю серед об'єктів з найвищим пріоритетом. Отже, алгоритм імітації Петрі-об'єктної моделі з конфліктом об'єктів складається з таких дій:

1. Визначити список K Петрі-об'єктів з найменшим значенням часу найближчої події t_{min}.
2. Відсортувати список K за значеннями пріоритету та вибрати з рівною ймовірністю серед об'єктів з найвищим пріоритетом об'єкт O_{min}, для якого виконуватиметься подія.
3. Просунути модельний час у момент найближчої події t_{min} і виконати подію: вихід маркерів з переходів Петрі-об'єкта O_{min}, вхід маркерів в усі Петрі-об'єкти моделі.
4. Дії 1,2,3 повторювати, доки модельний час менший за кінець інтервалу моделювання.

Зауважимо, що складова q^2 у виразі (1) присутня через сортування Петрі-об'єктів за пріоритетом. Тому алгоритм імітації Петрі-об'єктної моделі без конфлікту для великих систем має зростання складності пропорційно квадрату кількості об'єктів.

Таким чином, Петрі-об'єктна модель має алгоритм та структуру, які є зручними для розпаралелювання: динаміка її відтворюється однотипними фрагментами моделі, які діють за загальними для всіх фрагментів правилами. Як слідує з закону Амдала, успішність розпаралелювання залежить від співвідношення частин паралельного та послідовного виконання у

© Стецюк І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

побудованому алгоритмі. Звичайний алгоритм імітації організований таким чином, що при кожному просуванні часу здійснюється одна або декілька подій. Оскільки подій, які виконуються одночасно, як правило, небагато, то спрямовувати зусилля на розпаралелювання дій, що реалізують події в кожний конкретний момент часу, не ефективно. Потрібно організувати імітацію моделі таким чином, щоб здійснювалась власне одночасна імітація фрагментів моделі.

Паралельний алгоритм Петрі-об'єктного моделювання

В реальному світі об'єкти змінюються в часі у відповідності до своєї динаміки одночасно, не зважаючи один на одного аж доки не виникне подія, в якій вони задіяні спільно. Звідси, надамо можливість кожному Петрі-об'єкту функціонувати незалежно в межах, доки на його функціонування не впливають події в інших Петрі-об'єктах. Розглянемо взаємодію двох Петрі-об'єктів A і B (рис. 1). Введемо означення:

Означення 1. Перехід мережі Петрі-об'єкта є вихідним, якщо при його запуску здійснюється вихід у спільну позицію з іншим об'єктом або здійснюється вихід у позицію іншого об'єкта (рис.1 а,б).

Означення 2. Позиція об'єкта є вхідною, якщо вона є спільною з позицією іншого об'єкта або у неї здійснюється вихід з переходу іншого об'єкта (рис.1 в,г).

Означення 3. Якщо об'єкт A має вихідний перехід, з якого здійснюється вихід маркерів у позицію об'єкта B, то об'єкт B є next-об'єктом для об'єкта A.

Означення 4. Якщо об'єкт B має позицію, в яку здійснюється вихід маркерів з вихідного переходу об'єкта A, то об'єкт A є previous-об'єктом для об'єкта B.

Означення 5. Безпечним інтервалом імітації об'єкта є інтервал часу між двома послідовними виходами маркерів з вихідного переходу його previous-об'єкта.

© Стецюк І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

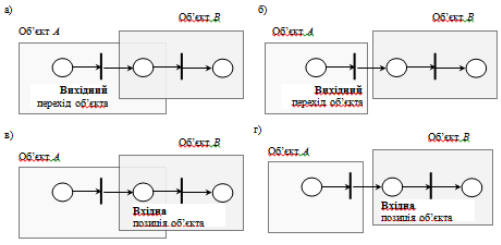

Рис.1 – Взаємодія Петрі-об'єктів: а) вихід маркерів з переходу об'єкта у спільну позицію його next-об'єкта, б) вихід маркерів з переходу об'єкта у позицію його next-об'єкта, в) вхід маркерів у спільну позицію об'єкта з його previous-об'єкта, г) вхід маркерів у позицію об'єкта з переходу його previous-об'єкта.

Таким чином, для кожного об'єкта мають бути визначені його previous-об'єкти та next-об'єкти. Якщо відповідного зв'язку немає, то в об'єкті указується посилання на «нульовий» previous- або next-об'єкт (null object). На інтервалі часу між послідовними входами маркерів ззовні динаміка об'єкта не залежить від функціонування його previous-об'єктів, тому події в межах цього інтервалу можуть відбуватись в об'єктах незалежно один від одного, а значить паралельно. Припустимо, що модель побудована таким чином, що для довільної пари об'єктів A і B об'єкт A не може бути одночасно previous-об'єктом і next-об'єктом для об'єкта B. Побудуємо паралельний алгоритм Петрі-об'єктної моделі за такими правилами:

1. Кожний Петрі-об'єкт здійснює імітацію в локальному часі в окремому потоці.
2. Кожний об'єкт, який має next-об'єкт, передає йому інформацію про моменти виходу маркерів з об'єкта і призупиняє своє функціонування при значному накопиченні такої інформації.

© Стецюк І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

3. Кожен об'єкт, який має previous-об'єкт, здійснює імітацію в межах до наступної події входу в нього маркерів з previous-об'єкта, поступово просуваючи свій локальний час до повного вичерпання накопиченої інформації про вхідні події, або очікує надходження інформації про вхідні події зі свого previous-об'єкта.
4. Об'єкт, який має previous-об'єкт, при досягненні моменту часу входу маркерів в об'єкт, відновлює вихід маркерів у спільну позицію з переходу previous-об'єкта та продовжує імітацію.
5. Об'єкт, який має next-об'єкт, при досягненні моменту виходу маркерів з об'єкта, призупиняє вихід маркерів у позицію next-об'єкта. Цей вихід відновить next-об'єкт у відповідний момент часу свого функціонування.
6. Об'єкт, який вичерпав час моделювання, передає повідомлення про це next-об'єкту, якщо такий є, і завершує свою роботу. Разом з завершенням роботи об'єкта завершує роботу і потік, ним генерований.
7. Об'єкт, який отримав від previous-об'єкта сигнал про завершення його роботи і, водночас, вичерпав усі накопичені події, припиняє свою роботу.

Отже, функціонування кожного об'єкта відбувається у тісній взаємодії з його previous та next-об'єктами, але незалежно від інших об'єктів, що надає можливість кільком об'єктам одночасно здійснювати імітацію, не порушуючи логіку функціонування один одного. Узгодження подій в різних об'єктах відбувається за рахунок завдання безпечних інтервалів часу, в межах яких функціонування об'єкта не залежить від інших об'єктів.

Для кожного об'єкта введемо буфер зовнішніх подій, який зберігає моменти часу надходження маркерів з інших об'єктів, що не опрацьовані на поточний момент часу. В межах від одного моменту зовнішньої події до наступної гарантовано не виникає зовнішніх подій і об'єкт може виконувати імітацію своїх внутрішніх подій. Порожній стан буфера означає, що наступний момент зовнішньої події невідомий (на поточний момент виконання алгоритму імітації). Стан буфера, в якому останній його елемент є нескінченність (максимально допус-

тиме число), означає, що previous-об'єкт завершив свою роботу і надходження зовнішніх подій не очікується.

Локальний час об'єкта просувається за такими правилами:

- якщо час найближчої події менший за межу безпечного інтервалу, то просунути локальний час у момент найближчої події, виконати (внутрішні) події, для яких час збігається з поточним моментом, та продовжити імітацію об'єкта;
- інакше
  - якщо межа безпечного інтервалу менша за межу інтервалу моделювання
    - просунути локальний час на межу безпечного інтервалу та виконати зовнішню подію (відновлення маркерів у вхідних позиціях об'єкта),
  - інакше просунути локальний час на межу безпечного інтервалу та завершити імітацію об'єкта

Межа безпечного інтервалу визначається за наступними правилами:

- якщо немає previous-об'єкта, то межа встановлюється в час моделювання;
- інакше
  - якщо буфер зовнішніх подій не порожній і перша подія менша за час моделювання, то межа встановлюється в момент першої зовнішньої події;
  - інакше - в час моделювання.

Об'єкт знаходиться в очікуванні (призупиняє своє функціонування), якщо і тільки якщо:

- він знаходиться у stop-стані: кількість маркерів недостатня для запуску переходів і момент виходу маркерів з переходу більший за границю безпечного інтервалу і, якщо є previous-об'єкт, перша подія в буфері менша за границю безпечного інтервалу;
- є previous-об'єкт і буфер зовнішніх подій порожній;
- є next-об'єкт і його буфер зовнішніх подій перевищує заданий ліміт.

© Стецюк І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

Об'єкт розблоковується (припиняє очікування) якщо і тільки якщо:

- його previous-об'єкт здійснив вихід у позицію, спільну з даним об'єктом;
- його next-об'єкт подав сигнал про відновлення роботи об'єкта у разі, якщо він вичерпав буфер подій до заданого ліміту.

Об'єкт завершує імітацію, якщо досягнутий кінець інтервалу моделювання.

Синхронізація виходу маркерів у спільну позицію відбувається таким чином:

- у момент здійснення виходу маркерів з вихідного переходу Петрі-об'єкта вихід маркерів у позицію next-об'єкта не здійснюється, замість цього поточний момент часу запам'ятовується в буфері зовнішніх подій next-об'єкта;
- коли локальний час об'єкта досягає першого значення буфера подій, здійснюється вхід маркерів у вхідну позицію об'єкта, а перше значення з буфера зовнішніх подій видаляється.

Якщо в буфері об'єкта перший елемент нескінченність, це означає, що його previous-об'єкт завершив імітацію на інтервалі імітації і зовнішніх подій більше не очікується. Об'єкт продовжує імітацію до завершення інтервалу моделювання без очікування зовнішніх подій.

Якщо об'єкт має декілька previous-об'єктів, то для кожного з них створюється свій буфер подій. Момент найближчої події можна визначити тільки, коли надійшла інформація про наступні події від усіх previous-об'єктів. Тому робота такого об'єкта призупиняється, якщо хоч один буфер подій порожній, і відновлюється, якщо усі буфери подій не порожні. Щоб запобігти взаємному блокуванню об'єктів, потрібно відслідковувати стан, в якому об'єкт і його previous-об'єкти одночасно знаходяться в стані очікування (коли хоч один з їх буферів подій є порожнім). В цьому випадку потрібно визначити найближчу подію для об'єкта і його previous-об'єктів, просунути локальний час в момент найближчої події і продовжити моделювання.

© Стецюк І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

Якщо об'єкт має декілька next-об'єктів, то робота такого об'єкта призупиняється, якщо досягає ліміту буфер подій хоч в одному з його next-об'єктів, і відновлюється, якщо в усіх next-об'єктах кількість подій в буфері подій менша за назначений ліміт.

Алгоритм, описаний вище, виходить з таких припущень: 1) модель має хоча б один об'єкт, для якого не заданий previous-об'єкт, і хоча б один об'єкт, для якого не заданий next-об'єкт; 2) для будь-якого об'єкта A всі об'єкти моделі можуть поділені на дві підмножини, що не перетинаються: об'єкти, які є його previous-об'єктами або previous-об'єктами його previous-об'єктів, та об'єкти, які є його next-об'єктами або next-об'єктами його next-об'єктів. В реальних умовах перше припущення відповідає відкритій системі, а другого можна досягти, конструюючи модель з Петрі-об'єктів, що не мають зворотних зв'язків.

Реалізація паралельного алгоритму імітації

Для реалізації алгоритму використовується багатопоточна технологія java та бібліотека java.util.concurrent, що надає можливість гнучкого управління потоками. Метод main() програми створює Петрі-об'єктну модель і запускає кожний її Петрі-об'єкт в окремому потоці. При запуску Петрі-об'єкта в його методі run() визначається межа безпечного інтервалу та здійснюється запуск імітації на безпечному інтервалі, доки не досягнутий кінець інтервалу моделювання. Імітація Петрі-об'єкта призупиняться, якщо буфер його зовнішніх подій порожній або буфер зовнішніх подій його next-об'єкта перевищив заданий ліміт. При досягненні межі безпечного інтервалу здійснюється відповідна зовнішня подія та її видалення з буфера зовнішніх подій. Позначимо:

timeLocal – локальний час Петрі-об'єкта,

timeMod – кінець інтервалу моделювання,

timeLimit – границя безпечного інтервалу імітації Петрі-об'єкта,

timeMin – момент найближчої події,

firstTimeExternalInput – перший момент в буфері зовнішніх подій,

isEmpty – буфер зовнішніх подій порожній,

© Стецюк І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

input() – вхід маркерів в переходи мережі Петрі-об'єкта,

output() – вихід маркерів з переходів мережі Петрі-об'єкта,

goUntil(t) – імітація Петрі-об'єкта на інтервалі часу від моменту timeLoc до моменту t,

moveTimeLocal() – просування часу об'єкту.

Метод run() Петрі-об'єкта містить такі дії:

- доки timeLocal < timeMod:
  - якщо є previous-об'єкт,
    - очікувати, доки isEmpty;
    - timeLimit = firstTimeExternalInput;
    - якщо timeLimit > timeMod, то timeLimit = timeMod;
  - інакше timeLimit = timeMod;
  - якщо timeLocal < timeLimit, виконати goUntil(timeLimit);
  - інакше завершити метод run().
- кінець циклу доки.

Метод goUntil(t) здійснює імітацію об'єкта в межах безпечного інтервалу, призупиняючи її, якщо буфер зовнішніх подій порожній:

- доки timeLocal < timeLimit:
  - очікувати, доки мережа знаходиться у stop-стані;
  - input();
  - якщо timeMin < timeLimit,
    - timeLocal = timeMin;
    - output() і продовжити цикл методу goUntil;
  - інакше
    - якщо timeLimit >= timeMod,
      - timeLocal = timeMod;

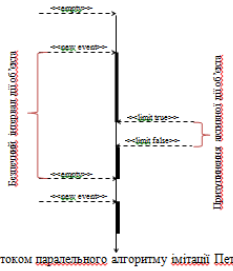

Рис. 3 –Управління потоком паралельного алгоритму імітації Петрі-об'єктної моделі

З опису алгоритму випливає, що на час виконання алгоритму впливатимуть затримки на очікування повідомлень від previous-об'єктів та від next-об'єктів. Останні можна завжди зменшити за рахунок збільшення обмеження на кількість подій в буфері зовнішніх подій, а перші залежать від внутрішньої складності об'єктів і не можуть бути змінені тільки параметрами алгоритму. Отже, моделі з послідовною структурою, в яких затримки на очікування повідомлень від previous-об'єктів більші, матимуть більший час виконання алгоритму. Петрі-об'єкти, які мають однакові previous-об'єкти, працюватимуть в алгоритмі незалежно один від одного та одночасно.

Тестування алгоритму здійснювалось на моделі, яка відтворює імітацію послідовно з'єднаних систем масового обслуговування (СМО). Для такої мережі масового обслуговування можуть бути виконані аналітичні розрахунки і порівняні потім з результатами імітації. Петрі-об'єктна модель складається з об'єктів «Генератор» та «Група СМО» (рис.4). Петрі-об'єкт «Група СМО» відтворює динаміку послідовно з'єднаних кількох СМО. Кількість СМО, об'єднаних в групу, указується в конструкторі Петрі-об'єкта.

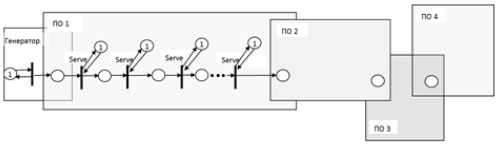

Рис.4 –Тестова Петрі-об'єктна модель мережі масового обслуговування

За результатами тестових прогонів виконувалось порівняння середнього значення черги кожної СМО з розрахованим аналітичним. Спостережувана похибка складає не більше 3%.

Експериментальне дослідження проводилось на двох ядерному комп'ютері (Processor Intel® Core™2 Duo CPU E8400, @3.00 GHz, RAM 4 GB). Порівняння часу виконання послідовного та паралельного алгоритму імітації Петрі-об'єктної моделі, побудованого з використанням глобальної змінною часу та найпростіших засобів управління потоками, показало значне збільшення часу виконання паралельного алгоритму при збільшенні кількості подій в моделі (рис.5). Це пояснюється тим, що глобальна змінна часу та синхронізовані методи запуску подій змушують об'єкти очікувати один одного при кожному просуванні часу. Алгоритм працює правильно, але не ефективно.

Навпаки, паралельний алгоритм імітації Петрі-об'єктної моделі з локальними змінними часу, який розроблений, за результатами експериментального порівняння його часу виконання з часом виконання послідовного алгоритму імітації свідчить про високу його ефективність: при збільшенні кількості подій в моделі: при кількості подій близько 300 спостерігається зменшення часу виконання у 14 разів і лінійна залежність часу виконання паралельного алгоритму від складності моделі (рис.6). При подальшому збільшенні кількості Петрі-об'єктів в моделі до 700 і, відповідно, кількості подій до майже 7000 лінійна залежність часу виконання паралельного алгоритму від складності моделі зберігається. Це означає, що завдя-

- передати сигнал next-об'єкту про завершення своєї роботи;

- інакше
  - якщо є previous-об'єкт, який не завершив свою роботу,
    - очікувати, доки буфер подій порожній;
    - timeLocal = timeLimit;
    - виконати зовнішню подію (вхід маркерів у спільну позицію та видалення з буфера зовнішніх подій);
    - відновити дію previous-об'єкта, якщо кількість зовнішніх подій зменшилась до прийнятної;
  - timeLocal = timeLimit, що означає виконання умови виходу з циклу;
- кінець циклу доки.

Метод output() у разі, якщо здійснюється вихід маркерів у вихідну позицію Петрі-об'єкта, заносить подію у буфер зовнішніх подій next-об'єкта і призупиняє дію об'єкта, якщо буфер зовнішніх подій його next-об'єкта перевищує заданий ліміт. Повідомлення про поповнення буфера зовнішніх подій відновлює дію об'єкта, який знаходився в очікуванні.

Взаємодія потоків представлена на діаграмі (рис.2). Потоки *A*, *B*, *C* ілюструють дію потоків, що запускають імітацію Петрі-об'єктів *A*, *B*, *C* таких, що *A* є previous-об'єктом для *B*, *B* є next-об'єктом для *A* і previous-об'єктом для *C*, *C* є next-об'єктом для *B*. Пара повідомлень <<new event>> та <<empty>> інформує об'єкт про надходження нової події від previous-об'єкта та, навпаки, що таких подій немає. Повідомлення <<no event>> сповіщає next-об'єкт про завершення імітації об'єктом, а значить про припинення надходження подій від previous-об'єкта. Пара повідомлень <<limit true>>, <<limit false>> інформує об'єкт про перевищення ліміту буферу зовнішніх подій його next-об'єкту або, навпаки, про те, що такого перевищення немає.

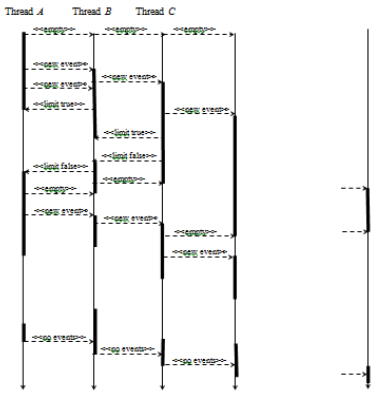

Рис. 2 – Взаємодія потоків паралельного алгоритму імітації Петрі-об'єктної моделі

Управління роботою потоку двома парами повідомлень представлене на рисунку 3. Об'єкт знаходиться в очікуванні, доки його буфер зовнішніх подій порожній, і розпочинає свою роботу при першому надходженні інформації про момент зовнішньої події. Безпечний інтервал дії об'єкта завершується, якщо його буфер зовнішніх подій порожній. Під час активної дії об'єкта може надійти повідомлення про перевищення ліміту буферу зовнішніх подій його next-об'єкту, яке призупиняє активну дію об'єкта доки не надійде альтернативне повідомлення.

ки паралелизму час обчислення Петрі-об'єктної моделі зменшився від кубічної оцінки (1) до лінійної (рис.7).

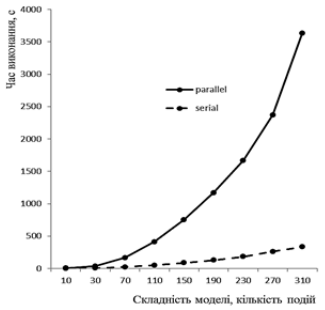

Рис.5 – Ефективність паралельного алгоритму імітації з глобальною зміною часу

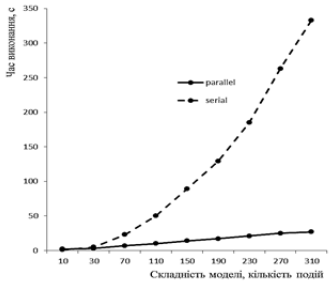

Рис.6 – Ефективність паралельного алгоритму імітації з локальною зміною часу

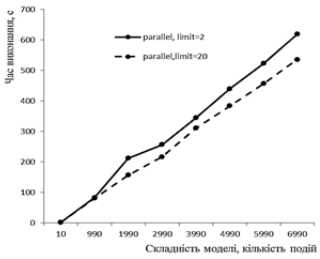

Рис.7 –Зростання часу виконання паралельного алгоритму імітації з локальною зміною часу

Висновки

Ефективний паралельний алгоритм імітації може бути побудований тільки в результаті його повної перебудови, що враховує усі особливості організації паралельних обчислень: створення та запуск потоків, блокування їх дій та синхронізацію з діями інших потоків. Петрі-об'єктна модель має зручну для організації паралельних обчислень структуру, оскільки складається з елементів, динаміка яких описана стохастичною мережею Петрі. Проте використання стандартних засобів розпаралелювання та глобальної зміної часу призводить до коректного, але не швидкого алгоритму імітації. Паралельний алгоритм імітації Петрі-об'єктої моделі, який розроблений, одночасно відтворює функціонування об'єктів моделі в окремих потоках з урахуванням динаміки тільки об'єктів, з якими вони безпосередньо зв'язані. Експериментальне дослідження алгоритму свідчить про його коректність та ефективність. Порівняння з аналітичними моделями масового обслуговування свідчить про достатню високу точність з величиною похибки, що не перевищує 3%. Дослідження ефективності

алгоритму свідчить про близьку до лінійної залежність часу виконання алгоритму від складності моделі.

Отже, застосування паралельних обчислень надало можливість зменшити складність алгоритму імітації від кубічної до лінійної, що є надзвичайно важливим при моделюванні великих систем. Оскільки Петрі-об'єктний підхід є універсальним для дискретно-подійних систем, то розроблений алгоритм може стати універсальним засобом для паралельної імітації таких систем.

Література

1. W.M.P. van der Aalst Process-Aware Information Systems: Lessons to be Learned from Process Mining // Transactions on Petri Nets and Other Models of Concurrency II. Special Issue on Concurrency in Process-Aware Information Systems. - Springer, 2009. - P.1-26.

2. W.M.P. van der Aalst Business Process Management as the "Killer App" for Petri Nets // Software and Systems Modeling. – 2015. – http://wwwis.win.tue.nl/~wvdaalst/publications/z1.pdf

3. Стеценко И.В. Алгоритм имитации Петри-объектной модели // Математичні машини і системи. – Київ, 2012. - №1. – С.154-165.

4. Stetsenko Inna V., Dorosh Vitaliy I., Dyfuchyn Anton Petri-object simulation: software package and complexity // Intelligent Data Acquisition and Advanced Computing Sysytems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference. – IEEE, 2015. – Vol.1. – P.381-385.

5. Foster I. Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engeneering. - Addison-Wesley Longman Publishing Co., 1995 - 370 p.

6. Fujimoto Richard M. Parallel and distributed simulation // Proceedings of the 2015 Winter Simulation Conference. – IEEE, 2015. - P.45-49.

7. Garg Vijay K. Concurrent and distributed computing in java. - John Wiley & sons, Inc., 2004. – 305 p.

8. Walter J.C. Parallel Simulation of queueing Petri Net models // Karlsruhe Institute of technology. Diploma thesis. – 2013. – [Електронний ресурс] – https://sdqweb.ipd.kit.edu/publications/pdfs/walter2013-parallel.pdf

9. Fujimoto R. M. Parallel and distributed simulation system // Proceedings of the 2001 Winter Simulation Conference - P.147-157.

10. Stetsenko Inna V. State equations of stochastic timed petri nets with informational relations // Cybernetics and systems analysis Volume 48, Number 5(2012), 784-797.

11. Степенко И.В. Теоретические основы Петри-объектного моделирования систем // Математичні машини і системи. – Київ, 2011. - №4. – С.136-148.

© Стеценко І.В. Паралельний алгоритм імітації Петрі-об'єктної моделі
Кібернетика і системний аналіз, № 53(4), 2017. DOI 10.1007/s10559-017-9963-1
Київський політехнічний інститут імені Ігоря Сікорського

Степенко И.В.

Параллельный алгоритм имитации Петри-объектной модели

В статье рассматривается разработка эффективного алгоритма имитации дискретно-событийных систем с большим количеством элементов. С применением Петри-объектного моделирования и параллельных вычислений разработан алгоритм, который одновременно воспроизводит функционирование Петри-объектов модели в отдельных потоках. Линейная зависимость времени выполнения разработанного алгоритма от сложности модели подтверждается результатами экспериментальных исследований.

Ключевые слова: стохастическая сеть Петри, алгоритм имитации, параллельные вычисления

Степенко І.В.

Паралельний алгоритм імітації Петрі-об'єктної моделі

В статті розглядається розробка ефективного алгоритму імітації дискретно-подійних систем з великою кількістю елементів. Із застосуванням Петрі-об'єктного моделювання та паралельних обчислень розроблений алгоритм, який одночасно відтворює функціонування Петрі-об'єктів моделі в окремих потоках. Лінійна залежність часу виконання розробленого алгоритму від складності моделі підтверджується результатами експериментальних досліджень.

Ключові слова: стохастична мережа Петрі, алгоритм імітації, паралельні обчислення

Stetsenko I.V.

The parallel Petri-object simulation algorithm

The article considers the development of efficient simulation algorithm of the discrete event systems with a lot of elements. With the use of Petri-object simulation and parallel computing the algorithm is developed that simultaneously reproduces the model's Petri-objects functioning in separate streams. The linear relationship between the runtime of the developed algorithm and the complexity of the model is confirmed by the results of experimental study.

Key words: stochastic Petri met, simulation algorithm, parallel computing