

ПАРАЛЕЛЬНИЙ АЛГОРИТМ ІМІТАЦІЇ ПЕТРІ-ОБ'ЄКТНОЇ МОДЕЛІ

І.В. Стеценко

Київський національний університет будівництва та архітектури

Алгоритми імітації дискретно-подійних систем характеризуються значним зростанням часу виконання при збільшенні складності моделі (кількості подій, що описують її функціонування), що може призводити до суттєвих затримок в управлінських процесах та процесах прийняття рішень, де моделі процесів використовуються як складові компоненти. Тому для дослідження великих систем (транспортних, виробничих, фінансово-економічних, еколого-економічних) задача підвищення швидкості обчислень імітаційної моделі є актуальною. Побудова моделей процесів складних виробничих систем та розбудова на їх основі process-aware інформаційних систем розглядається в роботі [1].

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 [1] і є універсальним для класу дискретно-подійних систем. У роботі [2] отримана оцінка складності цього алгоритму аналітичним способом та доведена її справедливості експериментально. Задача розпаралелювання цього алгоритму поставлена в даному дослідженні.

Технології паралельного програмування потребують спеціальної розробки алгоритму, який, по-перше, поділяє усі операції на ті, що можуть біти виконані послідовно, та ті, що можуть бути виконані паралельно, по-друге, визначає окремо послідовність дій кожного потоку. У багатьох прикладних задачах для досягнення значного ефекту паралельні алгоритми будуються з урахуванням специфіки вирішуваної задачі.

Сучасні бібліотеки паралельного програмування містять певні засоби автоматизації побудови паралельних обчислень такі, як розпаралелювання дій циклу або паралельне сортування елементів масиву. Проте, як показало дослідження, їх застосування до алгоритму імітації не дають значного ефекту і, навіть, сповільнюють роботу алгоритму. Це пояснюється тим, що ці дії в алгоритмі повторюються багато разів і кожного разу заново створюються нові потоки для кожного розпаралеленого циклу чи розпаралеленого сортування.

Час, який витрачається на створення потоків, не покривається пришвидшенням, здобутим за рахунок розпаралелювання.

У роботі [3] наведена класифікація існуючих способів розпаралелювання алгоритмів імітації в залежності від рівня, на якому застосовується паралелізм: програмний застосунок, функціонування імітаційної моделі, її алгоритмізація. Усі три способи використовують на різних етапах побудови моделі: розробки моделі засобами графічного інтерфейсу, дослідження моделі експериментальними методами, розробки власне алгоритму імітації. 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.

Паралельний алгоритм Петрі-об'єктної моделі, який запропонований, побудований за такими правилами:

1. Кожний Петрі-об'єкт здійснює імітацію в локальному часі в окремому потоці.

2. 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.

3. Перехід об'єкта є вхідним, якщо він має спільну позицію з іншим об'єктом або у його позицію здійснюється вихід з переходу іншого об'єкта.

4. Якщо здійснюється вихід об'єкта А в об'єкт В, то об'єкт А називатимемо previous-об'єктом для об'єкта В, а об'єкт В називатимемо next-об'єктом для об'єкта А.

5. Припустимо, що модель побудована таким чином, що об'єкт А не може бути одночасно і previous-об'єктом і next-об'єктом для об'єкта В.

6. 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.

7. 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.

8. Об'єкт, який має previous-об'єкт, при досягненні моменту часу входу маркерів в об'єкт, відновлює (запуском `reinststate`-методу)

вихід маркерів у спільну позицію з переходу previous-об'єкта та продовжує імітацію.

9. Об'єкт, який має next-об'єкт, при досягненні моменту виходу маркерів з об'єкта, відміняє (запуском zrollback-метод) вихід маркерів у позицію next-об'єкта. Цей вихід відновить next-об'єкт у відповідний момент часу свого функціонування.

10. 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).

11. 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 реалізації алгоритму використовується багатопоточна технологія java та бібліотека java.util.concurrent, що надає можливість гнучкого управління потоками. Експериментальне дослідження алгоритму свідчить про його коректність та ефективність. На рисунку 1 наведена динаміка зростання часу виконання паралельного та послідовного алгоритмів імітації тестової моделі при збільшенні її складності, що визначається кількістю подій, на двох-ядерному комп'ютері.

Таким чином, побудований паралельний алгоритм імітації Петрі-об'єктої моделі,

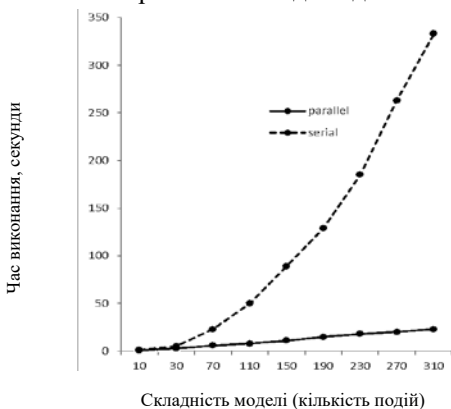


Рис. 1 – Результати експериментального дослідження ефективності паралельного алгоритму імітації

доведена його коректність з використанням тестової моделі, досліджені умови, при яких він забезпечує ефективність. Петрі-об'єктна модель має зручну для організації паралельних обчислень структуру. Одночасне відтворення функціонування об'єктів моделі в окремих потоках з урахуванням динаміки тільки об'єктів, з якими вони безпосередньо зв'язані, надає можливість досягти лінійного залежності часу виконання алгоритму від складності алгоритму. 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 [Text] // 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. Стеценко, И.В. Алгоритм имитации Петри-объектной модели [Текст] // Математичні машини і системи. – Київ, 2012. - №1. – С.154-165.
3. Stetsenko Inna V., Dorosh Vitaliy I., Dyfuchyn Anton Petri-object simulation: sofware package and complexity [Text] // Intelligent Data Acquisition and Advanced Computing Sysytems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference. – IEEE, 2015. – Vol.1. – P.381-385.
4. Walter, J.C. Parallel Simulation of queueing Petri Net models // Karlsruhe Institute of technology. Diploma thesis. – 2013. – [Електронний ресурс] – Режим доступу : <https://sdqweb.ipd.kit.edu/publications/pdfs/walter2013-parallel.pdf>

УДК 004.75

МОДЕЛИРОВАНИЕ ОБЛАЧНОЙ СИСТЕМЫ ВИДЕОНАБЛЮДЕНИЯ

Е.И. Дятлов

*Институт проблем математических машин и систем НАН Украины,
Украина*

Облачные вычисления становятся все более доступными и популярными. Как следствие, все ресурсоёмкие программные решения будут пытаться задействовать облачные вычисления в рамках собственных решений. Видеонаблюдение одна из важных задач в современном мире. Видеонаблюдение может предупредить преступление,