

ТОМАШЕВСЬКИЙ В.М., д.т.н., професор
 ДІДЕНКО Д.Г., аспірант, асистент кафедри

ПОРІВНЯННЯ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ OPENGPSS ТА GPSS/PC

В статті розглянуті різні аспекти проведення експериментів в дискретно-подійних системах OpenGPSS і GPSS/PC, порівнюються їх якісні та кількісні результати роботи. Предложено підхід розподілу експерименту на незалежні частини для паралельного виконання на вузлах кластера з наступною збіркою результатів, який призводить до підвищення продуктивності системи або зменшенню часу моделювання і при цьому не впливає на правильність кінцевих результатів.

COMPARISON OF JOB OF SIMULATION SYSTEMS OPENGPSS AND GPSS/PC

The report deals with questions of computing experiment in distributed discrete-event simulation systems OpenGPSS and GPSS/PC, their high-quality and quantitative job performances are compared. The problems of experiment distribution by independent part, deployment this part on cluster node, parallel execution and result assembles and here does not influence on the rightness of end-point also were laboured in the report.

Вступ

Імітаційне моделювання є одним з важливих методів аналізу різних складних систем. Разом зі збільшенням продуктивності та складності сучасних комп'ютерних систем постають нові вимоги до систем імітаційного моделювання – прискорення отримання результатів моделювання. Для цього можна використовувати розподілені системи імітаційного моделювання, які розглянуті в роботах [1, 2].

Існує багато реалізацій розподілених систем і систем тиражування експериментів, наприклад, SPEEDES [3], PARASOL [4] або Triad.Net [5]. Нажаль ці системи не охоплюють розробників, які працюють з поширеною мовою імітаційного моделювання GPSS [6]. Тому доцільно підвищення швидкості моделювання вже побудованих GPSS моделей. Останнім часом була розроблена нова розподілена дискретно-подійна система імітаційного моделювання OpenGPSS [7], яка доступна в режимі онлайн і використовує мову імітаційного моделювання GPSS. Але постає проблема довіри до результатів роботи нової системи моделювання.

Постановка задачі

Цілі дослідження: необхідно якісно та кількісно порівняти результати роботи систем імітаційного моделювання GPSS/PC та OpenGPSS. Еталоном при порівнянні може виступати широко відома прикладна програма GPSS/PC версії 2.0 компанії Minuteman Software (<http://www.minitemansoftware.com/>), яку багато років використовують розробники моделей і викладачі вузів.

Проведення обчислювального експерименту в послідовних системах моделювання

Розглянемо типовий обчислювальний експеримент мовою GPSS, який моделює роботу одноканальної системи масового обслуговування (СМО) з чергою (табл. 1).

Табл. 1. Текст тестової GPSS-програми

№	Команди GPSS
100	GENERATE 10,5
110	QUEUE QUE1
120	SEIZE PRIB1
130	DEPART QUE1
140	ADVANCE 15,5
150	RELEASE PRIB1
160	TERMINATE
170	GENERATE 5000
400	TERMINATE 1
500	START 1

Проведемо комп'ютерні прогони тестової GPSS-програми в двох системах моделювання. Для накопичення статистично-значимих результатів в системі GPSS/PC, кожний експеримент проводиться 3 рази, і в звіті відображаються лише середні арифметичні показники в таблицях результатів.

Система OpenGPSS може працювати як в звичайному послідовному режимі

(використовується лише один вузол моделювання), так і в розподіленому режимі, як описано в роботі [8] (використовується два або більше вузла моделювання). Спочатку будемо використовувати систему OpenGPSS всього з одним вузлом моделювання. У зведеній табл. 2 наведена порівняльна інформація про роботу обох систем.

Табл. 2. Порівняння результатів послідовного моделювання

Показник	Значення	GPSS/PC 2.0	OpenGPSS (один вузол)
Моделний час		5000	5000
Інформація про пристрій PRIB1			
ENTRIES	Кількість входів в пристрій	338	334
UTIL (%)	Коефіцієнт використання пристрою	0,997	0,997
AVE.TIME	Середній час зайняття пристрою	14,75	15,0
Інформація про чергу QUE1			
MAX.	Максимальна довжина черги	169	161
CONTENT	Поточна довжина черги	168	161
ENTRIES	Кількість входів в чергу	506	495
ENTR (0)	Кількість входів з нульовим часом чекання	2	3
AVE.CON	Середня довжина черги	80,85	81,0
AVE.TIME	Середній час перебування всіх транзактів в черзі	798,89	816,0
AVE.(0)	Середній час знаходження транзактів в черзі крім нульових входів	802,06	821,0

Результати моделювання дещо відрізняються, тому що в кожній системі моделювання використовуються різні датчики псевдо-випадкових чисел (ДПЧ).

Розподілені обчислювальні експерименти

Система OpenGPSS використовує автоматичне розподілення GPSS-програм. Далі розглянемо чи впливає це на результати моделювання.

Для того, щоб система OpenGPSS почала працювати в розподіленому режимі, модифікуємо тестову GPSS-програму – додаємо команди керування імітаційним експериментом та отримуємо модифіковану GPSS-програму (табл. 3). Тобто замість того щоб запускати програму моделювання три рази як раніше, будемо виконувати комп'ютерні прогони за допомогою команд керування START, CLEAR та RMULT. Проміжні результати статистики для пристрою PRIB1 та черги QUE1 будемо зберігати у файлі RES.TXT командою RESULT.

Табл. 3. Текст модифікованої GPSS-програми

№	Команди GPSS
100	GENERATE 10,5
110	QUEUE QUE1
120	SEIZE PRIB1

130	DEPART QUE1
140	ADVANCE 15,5
150	RELEASE PRIB1
160	TERMINATE
170	GENERATE 5000
180	SAVEVALUE MSV01,FR\$PRIB1
190	SAVEVALUE MSV02,FC\$PRIB1
200	SAVEVALUE MSV03,FT\$PRIB1
210	SAVEVALUE MSV04,Q\$QUE1
220	SAVEVALUE MSV05,QA\$QUE1
230	SAVEVALUE MSV06,QM\$QUE1
240	SAVEVALUE MSV07,QC\$QUE1
250	SAVEVALUE MSV08,QZ\$QUE1
260	SAVEVALUE MSV09,QT\$QUE1
270	SAVEVALUE MSV10,QX\$QUE1
400	TERMINATE 1
410	CLEAR
420	RMULT 1,2,8,5,8,2,8
500	START 1
510	RESULT RES.TXT,MSV01,01;FR\$PRIB1
520	RESULT RES.TXT,MSV02,02;FC\$PRIB1
530	RESULT RES.TXT,MSV03,03;FT\$PRIB1
540	RESULT RES.TXT,MSV04,04;Q\$QUE1
550	RESULT RES.TXT,MSV05,05;QA\$QUE1
560	RESULT RES.TXT,MSV06,06;QM\$QUE1
570	RESULT RES.TXT,MSV07,07;QC\$QUE1
580	RESULT RES.TXT,MSV08,08;QZ\$QUE1
590	RESULT RES.TXT,MSV09,09;QT\$QUE1

600	RESULT RES.TXT,MSV10,10;QX\$QUE1
610	CLEAR
620	RMULT 5,1,6,6,7,1,1
630	START 1
640	RESULT RES.TXT,MSV01,01;FR\$PRIB1
650	RESULT RES.TXT,MSV02,02;FC\$PRIB1
660	RESULT RES.TXT,MSV03,03;FT\$PRIB1
670	RESULT RES.TXT,MSV04,04;Q\$QUE1
680	RESULT RES.TXT,MSV05,05;QA\$QUE1
690	RESULT RES.TXT,MSV06,06;QM\$QUE1
700	RESULT RES.TXT,MSV07,07;QC\$QUE1
710	RESULT RES.TXT,MSV08,08;QZ\$QUE1
720	RESULT RES.TXT,MSV09,09;QT\$QUE1
730	RESULT RES.TXT,MSV10,10;QX\$QUE1
740	CLEAR
750	RMULT 2,2,5,2,8,9,3
760	START 1
770	RESULT RES.TXT,MSV01,01;FR\$PRIB1
780	RESULT RES.TXT,MSV02,02;FC\$PRIB1
790	RESULT RES.TXT,MSV03,03;FT\$PRIB1
800	RESULT RES.TXT,MSV04,04;Q\$QUE1
810	RESULT RES.TXT,MSV05,05;QA\$QUE1
820	RESULT RES.TXT,MSV06,06;QM\$QUE1
830	RESULT RES.TXT,MSV07,07;QC\$QUE1
840	RESULT RES.TXT,MSV08,08;QZ\$QUE1
850	RESULT RES.TXT,MSV09,09;QT\$QUE1
860	RESULT RES.TXT,MSV10,10;QX\$QUE1

За допомогою команди RMULT встановлюються початкові значення для всіх датчиків псевдо випадкових чисел. Команда START запускає обчислювальний експеримент. У GPSS-програмі можливо декілька разів використовувати команду START, для отримання проміжних результатів моделювання. Команда CLEAR очищає всю статистику системи і всі видаляє транзакти з моделі. Після виконання команди CLEAR уся система, окрім ДПЧ, знаходиться у початковому стані. Команда RESULT заносить результати моделювання у тимчасовий буфер результатів, для збереження значень і наступного аналізу.

Введемо визначення, які необхідні для подальшого розгляду матеріалу статті.

Нехай *сегмент обчислювального експерименту* (COE) – це послідовність команд керування, яка починається командою CLEAR або RMULT, містить не більше однієї команди CLEAR, RMULT, START і закінчується командою START або RESULT. Сегменти обчислювального експерименту розділяються на два типи: залежні і незалежні.

Будемо називати *залежним сегментом обчислювального експерименту* (ЗСОЕ) такі COE, які не містять команду RMULT або не містять команду CLEAR. Дані в таких

сегментах залежать від передісторії станів модельованої системи.

Незалежні сегменти обчислювального експерименту (НСОЕ) – це такі COE, які обов'язково складаються з послідовності команд CLEAR, RMULT, START. Саме послідовність команд CLEAR, RMULT заставляє систему “забути” свої попередні стани, тобто сегмент не залежить від передісторії станів системи.

Будемо називати *кадром сегмента обчислювального експерименту* (КСОЕ) – послідовність COE, яка починається з одного НСОЕ, та включає в себе всі ЗСОЕ, які залежать від цього початкового сегмента.

Припустимо, що S_{ij} – j -а імітаційна модель, яка виконується на i -му вузлі, $i = \overline{1, n}$, $j = \overline{1, n_i^j}$, де n_i^j – кількість моделей на i -му вузлі. У кожній моделі i -го вузла є свій модельний час, який позначимо $T_i = \{t_{i1}, t_{i2}, \dots, t_{in_i^i}\}$.

Наявність послідовності команд CLEAR, RMULT, START, дозволяє виділити незалежні COE, множину яких позначимо $U_{ij} = \{u_{ij1}, u_{ij2}, \dots, u_{ijn_i^j}\}$. Усі інші COE відносимо до залежних, множина – $C_{ij} = \{c_{ij1}, c_{ij2}, \dots, c_{ijn_i^j}\}$.

Зауважимо, що перший COE в моделі завжди незалежний, тому він належить множині $U_{ij} = \{u_{ij1}, u_{ij2}, \dots, u_{ijn_i^j}\}$.

Побудуємо кадр сегмента обчислювального експерименту як послідовність COE, що починається з одного НСОЕ, і включає в себе всі ЗСОЕ, які залежать від цього початкового сегмента:

$$K_{ijm} = \{u_{ijm} \mid \forall t: u_{ijm} \in U_{ij}\} \cup \{c_{ij} \mid c_{ij} \in C_{ij}, c_{ij} \text{ залежить від } u_{ijm}\}, i = \overline{1, n}, j = \overline{1, n_i^j}, m = \overline{1, n_i^m}$$

З побудови КСОЕ впливає незалежність кадрів один від одного.

Нехай K_{ij} – множина КСОЕ на i -му вузлі для j -ої моделі:

$$K_i = \bigcup_{n_i^j}^{j=1} K_{ij},$$

а результати проведення всіх COE з КСОЕ позначимо R_{ij} :

$$R_{ij} = \bigcup_{n_i^j}^{m=1} R_{ijm}.$$

Об'єднання результатів по всім КСОЕ j -ої моделі дає загальний результат:

$$R_i = \bigcup_{n_i^j}^{j=1} R_{ij}.$$

Тобто проведення одного КСОЕ дає один і тільки один результат з множини R_i .

Нехай $A_i = \{A_i^{AgSim}, A_i^{AgRep}, A_i^{AgSpl}, A_i^{AgSnc}, A_i^{AgTrf}, A_i^{AgPwr}, A_i^{AgUstr}, A_i^{AgGbr}\}$ – множина активних агентів i -го сервера, $i = \overline{1, n}$, де n – кількість серверів імітаційного моделювання в кластері [9].

Введемо матрицю \mathbf{B}_{iz} розмірністю $n \times n$, яка вказує чи є зв'язок між i -м та z -м сервером:

$$\mathbf{B}_{iz} = \begin{cases} 1, & \text{якщо існує зв'язок} \\ & \text{між } i\text{-м та } z\text{-м сервером,} \\ 0, & \text{в іншому випадку,} \end{cases} \\ i = \overline{1, n}, z = \overline{1, n}.$$

Стан i -го вузла опишемо як $P_i(A_i, S_i, T_i, K_i, R_i)$, а стан кластера моделювання – $P = \bigcup_{i=1}^n P_i(A_i, S_i, T_i, K_i, R_i)$.

Незалежність КСОЕ дозволяє проводити тиражування – одночасне виконання КСОЕ на різних вузлах кластера імітаційного моделювання, з подальшим об'єднанням результатів.

Система OpenGPSS автоматично виділяє множину КСОЕ, які можуть одночасно виконуватись на різних вузлах кластера. Після запуску j -ої GPSS-моделі агент користувача A_i^{AgUstr} , який виконує функцію компілятора [10] і переводить GPSS-текст у внутрішній формат та виділяє кадри COE:

$$A_i^{AgUstr} : P_i(S_i, K_i) \rightarrow P_i(S_i \cup \{s_{ij}\}, K_i \cup \{k_{ij}\}).$$

Агент реплікації AgRep [11] копіює GPSS-модель та всі КСОЕ на інші вузли:

$$A_i^{AgRep} : P_i(S_i, K_i) \rightarrow P_z(S_z \cup S_i, K_z \cup K_i), \forall i, z : \mathbf{B}_{iz} = 1,$$

тому кожний вузол повністю зберігає інформацію про модель та експеримент для запобігання втрат даних у разі вимкнення одного з вузлів.

У кластері відсутнє централізоване керування запуском КСОЕ – кожний вузол випадковим чином визначає КСОЕ із множини ще необроблених сегментів. Це дозволяє уникати конфліктів блокування: жоден вузол не чекає на результати обробки ОЕ іншими вузлами – вони діють за оптимістичним планом – продовжують виконувати моделювання над сегментами. Цей підхід відрізняється від підходу з динамічним розподілом на підмножини вузлів кластера і визначення локального вузла-координатора за допомогою одного з алгоритмів голосування, які наведені в [12] і мають меншу масштабованість (канал зв'язку координатора може бути перевантаженим) та надійність (при відмові координатора необхідно чекати переголосування).

Агент імітаційного моделювання AgSim виконує КСОЕ на i -му вузлі:

$$A_i^{AgSim} : P_i(\{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{in}\}) \rightarrow P_i(\{t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{in}\}), \\ \text{де } t_{ij} \leq t_{ij}^i.$$

Агент реплікації AgRep, реалізований мовою PL/SQL [13, 14], періодично копіює таблиці результатів REPORT и REPORT_DETAIL на інші вузли:

$$A_i^{AgRep} : P_z(R_z) \rightarrow P_i(R_i \cup R_z), \forall i, z : \mathbf{B}_{iz} = 1.$$

Потім AgRep виконує збирання результатів – об'єднання результатів всіх COE для подальшого виконання та аналізу загальних результатів і завершує ОЕ.

Розглянемо приклад. Розділимо модифіковану GPSS-програму з табл. 3 на сегменти, отримаємо три COE, які наведені в таблиці 4. COE1 – незалежний, тому що це перший COE та його виконання ні від чого не залежить; COE2 та COE3 – також незалежні, тому що виконуються команди CLEAR, RMULT, які встановлюють “початковий” стан системи.

Табл. 4. Уривок тексту GPSS-програми з сегментами

№	Команди GPSS	Сегменти	Кадри сегментів
	...	COE1 (НСОЕ)	КСОЕ1
410	CLEAR		
420	RMULT 1,2,8,5,8,2,8		
500	START 1		
510	RESULT RES.TXT,MSV01,01;FR\$PRIB1		
520	RESULT RES.TXT,MSV02,02;FC\$PRIB1		

530	RESULT RES.TXT,MSV03,03;FT\$PRIB1		
540	RESULT RES.TXT,MSV04,04;Q\$QUE1		
550	RESULT RES.TXT,MSV05,05;QA\$QUE1		
560	RESULT RES.TXT,MSV06,06;QM\$QUE1		
570	RESULT RES.TXT,MSV07,07;QC\$QUE1		
580	RESULT RES.TXT,MSV08,08;QZ\$QUE1		
590	RESULT RES.TXT,MSV09,09;QT\$QUE1		
600	RESULT RES.TXT,MSV10,10;QX\$QUE1		
610	CLEAR	COE2 (HCOE)	KCOE2
620	RMULT 5,1,6,6,7,1,1		
630	START 1		
640	RESULT RES.TXT,MSV01,01;FR\$PRIB1		
650	RESULT RES.TXT,MSV02,02;FC\$PRIB1		
660	RESULT RES.TXT,MSV03,03;FT\$PRIB1		
670	RESULT RES.TXT,MSV04,04;Q\$QUE1		
680	RESULT RES.TXT,MSV05,05;QA\$QUE1		
690	RESULT RES.TXT,MSV06,06;QM\$QUE1		
700	RESULT RES.TXT,MSV07,07;QC\$QUE1		
710	RESULT RES.TXT,MSV08,08;QZ\$QUE1		
720	RESULT RES.TXT,MSV09,09;QT\$QUE1		
730	RESULT RES.TXT,MSV10,10;QX\$QUE1		
740	CLEAR	COE3 (HCOE)	
750	RMULT 2,2,5,2,8,9,3		
760	START 1		
770	RESULT RES.TXT,MSV01,01;FR\$PRIB1		
780	RESULT RES.TXT,MSV02,02;FC\$PRIB1		
790	RESULT RES.TXT,MSV03,03;FT\$PRIB1		
800	RESULT RES.TXT,MSV04,04;Q\$QUE1		
810	RESULT RES.TXT,MSV05,05;QA\$QUE1		
820	RESULT RES.TXT,MSV06,06;QM\$QUE1		
830	RESULT RES.TXT,MSV07,07;QC\$QUE1		
840	RESULT RES.TXT,MSV08,08;QZ\$QUE1		
850	RESULT RES.TXT,MSV09,09;QT\$QUE1		
860	RESULT RES.TXT,MSV10,10;QX\$QUE1		
		Неявне автоматичне збирання результатів	

З прикладу видно, що початкові значення всіх ДПЧ, які використовувались раніше обов'язково повинні бути перевизначені командою RMULT.

Проведемо з GPSS-програмою, яка наведена в табл. 4, обчислювальний експеримент для кількості вузлів кластера один, два та три вузла. Незалежні COE можуть безпечно виконуватись на різних вузлах кластера імітаційного моделювання з подальшим збиранням результатів, що приводить до зменшення часу на проведення імітаційного експерименту.

Якщо кластер складається з одного вузла, то всі COE виконуються на ньому. При наявності двох вузлів: на одному вузлі виконуються два сегмента, а третій HCOE виконується на другому вузлі. Для трьох вузлів – кожен сегмент виконується на окремому вузлі. Середнє арифметичне результатів наведено в табл. 5. Система GPSS/PC не може працювати в розподіленому режимі, тому для неї наведена інформація при послідовному проведенні експерименту. Як видно із табл. 5 результати моделювання знову приблизно однакові.

Табл. 5. Порівняння результатів розподіленого моделювання

Показник	Розшифровка	GPSS/PC 2.0	OpenGPSS (один вузол)	OpenGPSS (два вузла ¹)	OpenGPSS (три вузла ²)
Астрономічний час, с	Астрономічний час моделювання, секунди	4	8	9	6
Модельний час	Модельний час	5000	5000	5000	5000
FR\$PRIB1	Коефіцієнт використання пристрою (в тисячних долях)	997	997	998	998
FC\$PRIB1	Число входів в пристрій	338	331	341	338
FT\$PRIB1	Середній час використання пристрою	14	15	14	14
Q\$QUE1	Довжина черги	168	161	162	158
QA\$QUE1	Середня довжина черги	80	79	82	77
QM\$QUE1	Максимальна довжина черги	169	162	162	159
QC\$QUE1	Загальне число входів в чергу	506	492	503	596
QZ\$QUE1	Число нульових входів в чергу	2	1	1	1
QT\$QUE1	Середній час знаходження транзакта в черзі (нульові входи включно)	798	811	819	776
QX\$QUE1	Середній час знаходження транзакта в черзі (без нульових входів)	802	813	820	777

Порівняння швидкості моделювання

Розглянемо, як кількість транзактів в моделі впливає на швидкість моделювання, для чого в нашій останній GPSS-програмі у рядку 170 замість блоку «GENERATE 5000» будемо використовувати «GENERATE 5000», «GENERATE 10000», «GENERATE 20000»... «START 90000». Усі експерименти будемо проводити для різної кількості вузлів кластера. Результати проведення експериментів відображені на рис. 1. По осі абсцис відкладені

значення модельного часу, а по осі ординат астрономічний час в секундах.

На рис. 1 астрономічний час моделювання для системи з двома вузлами «приблизно» такий самий як і для одного вузла, тому що GPSS-програма містить три (непарне число) сегмента, із-за чого один з вузлів «чекає» другий вузол поки той послідовно моделює два сегменти. Швидкість збільшується але не в два рази. На тренд, що відображає час для трьох вузлів, на початку моделювання впливають накладні витрати, але для модельного часу 90 тис. од. система моделювання OpenGPSS випереджає GPSS/PC.

¹ Використовується середнє арифметичне по двом вузлам.

² Використовується середнє арифметичне по трьом вузлам.

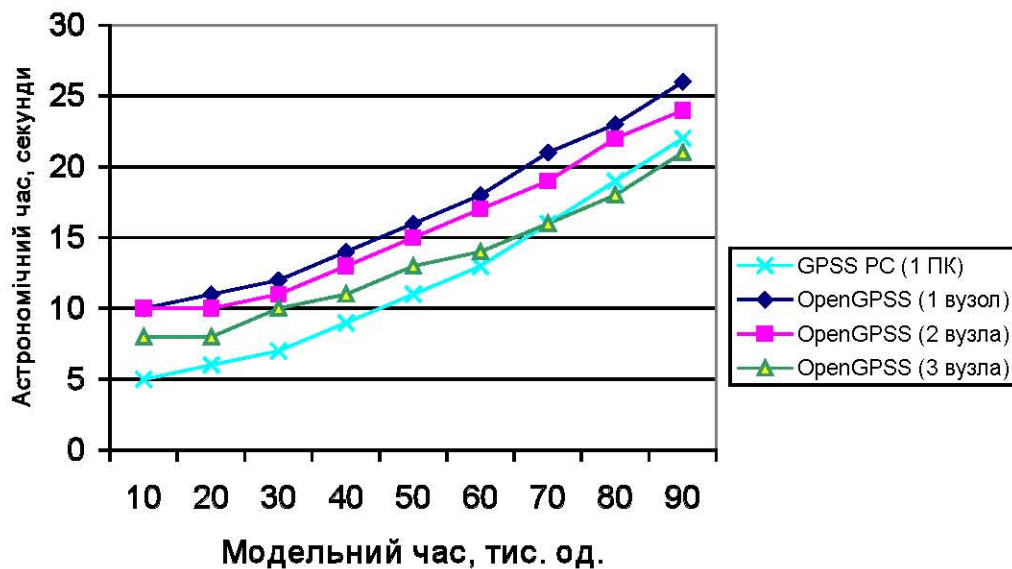


Рис.1. Залежність астрономічного часу моделювання від кількості вузлів кластера та модельного часу

Висновки

1. Системи GPSS та OpenGPSS на однакових GPSS-програмах видають приблизно однакові результати при послідовному і розподіленому режимі проведення експерименту. Для отримання більш точних результатів необхідно збільшити кількість прогонів моделі.

2. Запропонований підхід розділення експерименту на незалежні частини для подальшого паралельного виконання на вузлах кластера і збирання результатів призводить до підвищення продуктивності системи

(зменшення часу моделювання), що помітно на кластері з трьома або більше вузлами.

3. Отримані результати підтверджують можливість прозорого автоматичного розділення експерименту на незалежні частини для прискорення швидкості моделювання.

Перспективи досліджень

Оптимізація режимів роботи кластера для підвищення швидкості моделювання.

Побудова та дослідження кластера з великою кількістю вузлів (16-32).

Побудова розподіленої дискретно-подійної системи імітаційної моделювання з оптимістичною синхронізацією модельного часу у вузлах кластера.

Список літератури

1. Richard M. Fujimoto. Parallel and Distributed Simulation Systems. Wiley, 2000.
2. Замятина Е.Б. Современные теории имитационного моделирования: Специальный курс. – Пермь: ПГУ, 2007. – 119 с.
3. SPEEDES. <http://www.speedes.com>.
4. Mascarenhas E., Knop F., Vernon R. ParaSol: A multithreaded system for parallel simulation based on mobile threads. Winter Simulation Conference, 1995.
5. А.И. Миков, Е.Б. Замятина, А.Н. Фирсов. Инструментальные средства удалённого параллельного моделирования. В книге Proceedings of XXII International Conference “Knowledge-Dialogue-Solution”.– FOI-COMMERCE, Sofia, 2006, pp. 280-287.
6. Шрайбер Томас Дж. Моделирование с использованием GPSS. – М.: Машиностроение, 1980. – 593 с.
7. Киевский центр имитационного моделирования. <http://www.simulation.kiev.ua>.
8. Діденко Д.Г. Реалізація тиражування обчислювального експерименту в розподіленій системі моделювання OpenGPSS. – Наукові вісті – К. 2007. – № 5. С. 49-53.

9. Томашевский В.Н., Диденко Д.Г. Агентная архитектура распределенной дискретно-событийной системы имитационного моделирования OpenGPSS. Системні дослідження та інформаційні технології. № 4, 2006. – К.: ВПК “Політехніка”, 2006. С.123–133.
10. Ахо Альфред, Сети Рави, Ульман Джеффри. Компиляторы: принципы, технологии, инструменты. Пер. с англ. – М.: Издательский дом "Вильямс", 2001. – 768 с.
11. Діденко Д.Г. Агент реплікації в розподіленій дискретно-подійній системі імітаційного моделювання OPENGPSS. Матеріали міжнародної наукової конференції “Інтелектуальні системи прийняття рішень та прикладні аспекти інформаційних технологій”, 2006. С. 264–266.
12. Таненбаум Э., Стеен ван М. Распределенные системы. Принципы и парадигмы. – СПб.: Питер, 2003. – 877 с.
13. Кайт Том. Oracle для профессионалов. Книга 2. Расширение возможностей и защита: Пер. с англ. – М.: ДиасофтЮП, 2003. – 848 с.
14. Кайт Том. Oracle для профессионалов. Книга 1. Архитектура и основные особенности: Пер. с англ. – М.: ДиасофтЮП, 2003. – 672 с.