

УДК 681.324

М.А. Волк, М.А. Филимончук, М. Аль Шиблак, Р.Н. Гридель

*Харьковский национальный университет радиоэлектроники, Харьков***АНАЛИЗ РАСПРЕДЕЛЕННЫХ ИМИТАЦИОННЫХ МОДЕЛЕЙ
С КОНСЕРВАТИВНЫМИ АЛГОРИТМАМИ СИНХРОНИЗАЦИИ**

Рассматривается математическая модель функционирования распределенной имитационной модели под управлением консервативных алгоритмов синхронизации модельного времени. Модель учитывает следующие основные параметры: продвижение модельного времени, объемы памяти, время обслуживания моделей, время простоя ресурсов. Предложенная модель может быть использована при анализе распределенных имитационных моделей, создании эффективных методов распределения ресурсов.

Ключевые слова: *распределенная имитационная модель, консервативные алгоритмы синхронизации.*

Введение

Применение распределенного имитационного моделирования в последнее время позволило значительно расширить класс реализуемых имитационных моделей за счет использования большого количества мощных вычислительных ресурсов, к которым можно отнести локальные и глобальные компьютерные сети, суперкомпьютеры, кластера. Одной из задач в этой области является рациональное управление распределением вычислительных ресурсов при проведении масштабных имитационных экспериментов. Актуальность задачи усиливается значительной стоимостью эксплуатации такого вида ресурсов.

Существует ряд работ в этой области, среди которых можно выделить два основных направления. Первое из них используется в тех случаях, когда в проблемной области, для которой выполняется построение модели, существуют традиционные средства моделирования. Например, в теории массового обслуживания широкое распространение получили программные системы моделирования на базе языка GPSS [1]. Для исследования дискретных динамических систем более широкого класса – ап-

парат сетей Петри [2]. В этих случаях разработано множество методических рекомендаций, научных аналитических публикаций, которые помогут разработчику выбрать нужные средства моделирования [3]. Второе направление ориентировано на природу имитационной модели как представителя того или иного методологического подхода или использует идею о том, что любая имитационная модель представляет собой программный продукт. К таким работам следует отнести [4 – 8].

Основа подхода, который будет использоваться в настоящей работе заложена в трудах Вознесенской Т.В. [6, 7], которые обобщают модель взаимодействия двух имитационных моделей под управлением оптимистических алгоритмов синхронизации, которые были разработаны D. Mitra и I. Mitrani [5]. Дальнейшее развитие он получил в работах украинских ученых [9 – 11].

В работе [11] Приведены результаты исследования существующих методов анализа распределенных имитационных моделей, работающих под управлением различных алгоритмов синхронизации, в глобальных вычислительных системах, объединяющих разнородные ресурсы. Предложена новая математическая модель процесса функцио-

нирования распределенных симуляторов, учитывающая временные характеристики исполнения и передачи сообщений в частных моделях и динамическое изменение объемов памяти распределенной модели.

В данной статье обобщенная модель, полученная в работе [11], рассматривается для распределенных имитационных моделей, работающих под управлением консервативных алгоритмов синхронизации.

Модель функционирования распределенной имитационной модели

В процессе функционирования распределенная имитационная модель затрачивает процессорное время на исполнение своих активностей (подпрограмм, процедур, функций). Активности модели могут вызывать активности по работе с данными (посылка сообщений, сохранение внутреннего состояния), которые, в свою очередь, увеличивают процессорное время по обслуживанию частной модели. При этом, на каждом шаге моделирования может произойти увеличение объема памяти, занимаемого частной моделью. Увеличение памяти обусловлено состояниями внутренних переменных модели, изменениями в объектах операционной системы, обслуживающих программную модель, создаваемыми буферами данных для обмена сообщениями между моделями.

В рассматриваемую модель включим параметры, учитывающие динамику изменения объемов памяти частных имитационных моделей и реальное время, затрачиваемое на выполнение активностей менеджера памяти по выполнению операций с данными [11].

При продвижении модельного времени t_i^n n -й модели на i -м шаге моделирования происходит сохранение состояния модели, что приводит к росту памяти, занимаемой данными программной частной моделью V_i^n . На операции с памятью затрачивается процессорное время TR_i^n , которое состоит из времени сохранения состояния модели t_i^s , времени передачи сообщения от частной модели n частной модели k (t_i^m). Время t_i^m в общем случае является функцией, зависящей от поведения активности передачи данных δ_i^{nk} и размера передаваемых данных ($buff_i^{nk}$). В свою очередь время, затраченное на выполнение всей имитационной модели на i -м шаге моделирования ΔTR_i определяется как максимум среди суммы времен исполнения поведенческой активности φ_i^n и времени вы-

полнения операций с данными TR_i^n . В результате, получаем следующую математическую модель, описывающую развитие распределенной имитационной модели:

$$\left\{ \begin{array}{l} TR_0 = 0; \\ TR_0^n = 0; \\ \Delta TR_i^n = \begin{cases} t_i^s, & \text{если } t_i^n = T_i \\ 0, & \text{если } t_i^n \neq T_i \end{cases} + \\ \quad + \sum_{k=1}^N a^{nk} \cdot t_i^m(\delta_i^{nk}, \text{sizeof}(buff_i^{nk})); \\ TR_{i+1}^n = TR_i^n + \Delta TR_i^n; \\ TR_{i+1} = TR_i + \max_{n=1, N}(\text{time}(\varphi_i^n) + \Delta TR_i^n) + TR_i^M; \\ t_0^n = 0; \\ t_{i+1}^n = \begin{cases} t_i^n + \xi_i^n, & \text{если } t_i^n = T_i; \\ t_i^n, & \text{если } t_i^n \neq T_i; \end{cases} \\ V_0^n = \text{sizeof}(\Lambda^n \cup \Delta^n); \\ V_0 = \sum_{n=1}^N V_0^n; \\ \Delta V_i^n = \begin{cases} V_i^{ns}, & \text{если } t_i^n = T_i; \\ 0, & \text{если } t_i^n \neq T_i; \end{cases} \\ V_{i+1}^n = V_i^n + \Delta V_i^n; \\ V_{i+1} = V_i + \sum_{n=1}^N \Delta V_i^n; \\ T_0 = 0; \\ T_{i+1} = \min_{n=1, N}(t_{i+1}^n); \\ n = \overline{1, N}. \end{array} \right. \quad (1)$$

где TR_i – реальное время, затраченное на выполнение i шагов процесса моделирования всей имитационной модели; TR_i^n – реальное время, затраченное n -ой частной моделью на i -м шаге моделирования на работу с данными; t_i^s – время сохранения состояния модели на i -м шаге моделирования; t_i^m – время передачи данных; $\text{time}(\varphi_i^n)$ – время исполнения поведенческой активности φ_i^n ; a^{nk} – вероятность посылки сообщения от частной модели n к частной модели k ; N – количество частных моделей; TR_i^M – время работы служб системы моделирования на i -ом шаге моделирования; t_i^n – локальное модельное время n -ой частной модели на i -м шаге моделирования; T_i – глобальное модельное время на i -м шаге моделирования; ξ_i^n – значение времени, характеризующее внутреннюю работу

процессов между двумя соседними шагами моделирования; V_i^n – объем памяти, занимаемый данными n -ой частной моделью на i -м шаге моделирования; V_i^{ns} – объем дампа памяти для n -й частной модели на i -м шаге моделирования; ΔV_i^n – приращение объема памяти n -ой частной модели на i -м шаге моделирования; V_i – объем памяти, занимаемой всей распределенной имитационной моделью на i -м шаге моделирования; TR_0 ; TR_0^n ; t_0^n ; V_0^n ; V_0 – начальные значения соответствующих параметров.

Параметры T_i ; V_i ; TR_i вычисляются независимо друг от друга отдельными процедурами.

Следовательно, возможна реализация параллельного вычисления этих параметров как на системах с общей (SMP) так и раздельной памятью (ММР).

Кроме того, отсутствие количественных данных о каком-либо члене выражения (1) (например, времени, которое характеризует внутреннюю работу процессов между двумя соседними шагами моделирования), повлечет за собой невозможность вычисления только одного из них.

Модификация модели для консервативных алгоритмов синхронизации

Особенностью консервативных алгоритмов синхронизации модельного времени распределенных имитационных моделей является недопущение продвижения модельного времени частных имитационных моделей при вероятности появления события с меньшим значением временной метки.

Основным механизмом обеспечения такой синхронизации является посылка/ожидания сообщений, разрешающих или приостанавливающих продвижение локального модельного времени частных моделей. При этом рост памяти частной модели незначителен и обусловлен главным образом буфером обмена данными с другими моделями.

Следовательно, при стандартной реализации этих алгоритмов, отсутствует необходимость в выполнении дампа памяти частных моделей с целью сохранения внутреннего состояния для обеспечения функций синхронизации.

В таком случае в выражение (1) можно не учитывать параметры V_i^{ns} (объем памяти для сохранения состояния модели) и t_i^s (время сохранения состояния модели), и выражение (1) примет вид:

$$\left\{ \begin{array}{l} TR_0 = 0; \\ TR_0^n = 0; \\ \Delta TR_i^n = \sum_{k=1}^N a^{nk} \cdot t_i^m(\delta_i^{nk}, \text{sizeof}(\text{buff}_i^{nk})); \\ TR_{i+1}^n = TR_i^n + \Delta TR_i^n; \\ TR_{i+1} = TR_i + \max_{n=1, N}(\text{time}(\varphi_i^n) + \Delta TR_i^n) + TR_i^M; \\ t_0^n = 0; \\ t_{i+1}^n = \begin{cases} t_i^n + \xi_i^n, & \text{если } t_i^n = T_i; \\ t_i^n, & \text{если } t_i^n \neq T_i; \end{cases} \\ V_0^n = \text{sizeof}(\Lambda^n \cup \Delta^n); \\ V_0 = \sum_{n=1}^N V_0^n; \\ V_{\text{const}} = V_0; \\ T_0 = 0; \\ T_{i+1} = \min_{n=1, N}(t_{i+1}^n); \\ n = \overline{1, N}. \end{array} \right. \quad (2)$$

Реализация некоторых целей эксперимента, например, многовариантный анализ или обеспечение ветвления алгоритмов анализа в прошлом, могут потребовать сохранения состояния. В этом случае, указанные параметры могут быть оставлены, но условие их включения на определенных шагах моделирования должно определяться логикой эксперимента априорно, до начала моделирования.

В консервативных алгоритмах частные модели простаивают, ожидая разрешения на продвижение модельного времени. Вследствие этого, ресурс, выделенный для обслуживания этой частной модели, простаивает. Некоторые алгоритмы распределения ресурсов учитывают данный факт при балансировке нагрузки.

Добавим к математической модели процесса имитации выражение, учитывающее время простоя частной модели:

$$\left\{ \begin{array}{l} TR_0^n = 0; \\ TR_{i+1}^n = \begin{cases} TR_i^n, & \text{если } t_i^n = T_i; \\ TR_i^n + \max_{n'}(\text{time}(\varphi_i^{n'}) + \Delta TR_i^{n'}); \\ n' = \overline{1, N} / n' = n, & \text{если } t_i^n \neq T_i, \end{cases} \end{array} \right. \quad (3)$$

где TR_i^n – время простоя n -ой частной модели на момент достижения i -го шага моделирования; n' – индекс множество активностей модели, за исключением частной модели с индексом n .

Выражение (3) может эффективно использоваться в статических и динамических алгоритмах распределения ресурсов. При реализации алгоритма

на основе выражения (3), поиск максимума можно вести на множестве частных моделей, активированных на i -м шаге моделирования.

Выводы

Получена математическая модель функционирования распределенной имитационной модели с консервативным алгоритмом синхронизации частных моделей (выражение 2). Для данного класса моделей предложено математическое выражение, учитывающее простой вычислительных ресурсов при обслуживании локальной частной модели, на которых произведено распределение имитационной модели (выражение 3).

Данная модель может быть использована в методах и средствах анализа распределенных имитационных моделей, при построении автоматизированных систем анализа распределенных имитационных моделей, планировщиков распределенных вычислительных систем, при создании и исполнении распределенных имитационных моделей, параллельном и распределенном программировании.

Список литературы

1. Томашевский В. Имитационное моделирование в среде GPSS / В. Томашевский, Е. Жданова. – М.: Бестселлер, 2003. – 416 с.
2. Юдицкий С.А. Метод анализа конфигураций организационных систем на сетях Петри / С.А. Юдицкий, И.А. Мурадян // УБС. – 2007. – № 16. – С. 163-170.
3. Крэйн М. Введение в регенеративный метод анализа моделей / М. Крэйн, О. Лемуан. – М.: Наука, 1982. – 104 с.
4. Окольников В.В. Разработка средств распределенного имитационного моделирования для многопроцессорных вычислительных систем / В.В. Окольников: дис. ... д-ра техн. наук: 05.13.18. – Новосибирск, 2006. – 227 с. – РГБ ОД71:07-5/433.
5. Mitra D. Analysis and optimum performance of two message-passing parallel processors synchronized by rollback / D. Mitra, I. Mitrani // Performance'84, pp. 35-50, 1984.

6. Вознесенская Т.В. Математическая модель алгоритмов синхронизации времени для распределенного имитационного моделирования / Т.В. Вознесенская // Программные системы и инструменты. Тематический сборник факультета ВМиК МГУ им. Ломоносова. – № 1 – С. 56-66.

7. Вознесенская Т.В. Математическая модель для анализа производительности распределенных систем имитационного моделирования / Т.В. Вознесенская // Искусственный интеллект (Донецк), 2002. – № 2. – С. 74-78.

8. Миков А.И. Программные средства оптимизации распределенного имитационного эксперимента / А.И. Миков, Е.Б. Замятина, А.А. Козлов // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: тр. Всероссийской суперкомпьютерной конф. (21 – 26 сентября 2009, г. Новороссийск). – М.: Изд-во МГУ, 2009. – 524 с.

9. Ladyzhensky Y.V. Software system for event-driven logic simulation / Y.V. Ladyzhensky, Y.V. Popoff // IEEE EWDWT, Odessa, September 15 – 19, 2005. – P. 119-122.

10. Ладыженский Ю.В. Математическая модель динамического алгоритма продвижения времени для распределенного логического моделирования цифровых систем / Ю.В. Ладыженский, Г.А. Тесленко // Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. – Донецьк, 2008. – № 9. – С. 55-62.

11. Волк М.А. Анализ распределенных имитационных моделей в гетерогенных вычислительных системах / М.А. Волк // Науковий вісник Чернівецького національного університету імені Юрія Федьковича. Серія: Комп'ютерні системи та компоненти. – Т. 1, вип. 2. – Чернівці: ЧНУ, 2010. – С. 35-39.

Поступила в редколлегию 24.11.2011

Рецензент: д-р техн. наук, проф. С.Г. Удовенко, Харьковский национальный университет радиоэлектроники, Харьков.

АНАЛІЗ РОЗПОДІЛЕНИХ ІМІТАЦІЙНИХ МОДЕЛЕЙ С КОНСЕРВАТИВНИМИ АЛГОРИТМАМИ СИНХРОНІЗАЦІЇ

М.О. Волк, М. Аль Шиблак, Р.М. Грідель

Розглядається математична модель функціонування розподіленої імітаційної моделі під управлінням консервативних алгоритмів синхронізації модельного часу. Модель враховує наступні основні параметри: просування модельного часу, обсяги пам'яті, час обслуговування моделей, час простою ресурсів. Запропонована модель може бути використана при аналізі розподілених імітаційних моделей, створення ефективних методів розподілу ресурсів.

Ключові слова: розподілена імітаційна модель, консервативні алгоритми синхронізації.

ANALYSIS OF SIMULATION MODEL WITH CONSERVATIVE SYNCHRONIZATION ALGORITHMS

M.O. Volk, M. Al Shiblak, R.M. Gridel

A mathematical model of the distributed simulation model for the running of conservative synchronization algorithms of model time is considered. The model takes the following key parameters: the promotion of model time, memory, service time models, idle resources. The proposed model can be used in the analysis of distributed simulation models, developing effective methods of allocating resources.

Keywords: distributed simulation model, conservative synchronization algorithms.