

УДК 004.942

АЛГОРИТМИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА РАСПРЕДЕЛЕННОЙ ИМИТАЦИИ ДЛЯ ТЕХНОЛОГИИ АНАЛИЗА РАСПРЕДЕЛЕННЫХ ИМИТАЦИОННЫХ МОДЕЛЕЙ

У статті розглядається алгоритмічна модель процесу розподіленої імітації, яка заснована на програмному поданні розподілених імітаційних моделей. Модель враховує потоки даних, що виникають як всередині окремих моделей, так і при обміні інформацією між моделями, а також часові характеристики процесу моделювання. Модель призначена для використання в інформаційній технології аналізу розподілених імітаційних моделей

Ключові слова: розподілене імітаційне моделювання, алгоритмічна модель, технологія аналізу

В статье рассматривается алгоритмическая модель процесса распределенной имитации, основанная на программном представлении распределенных имитационных моделей. Модель учитывает потоки данных, возникающие как внутри частных моделей, так и при обмене информацией между моделями, а также временные характеристики процесса моделирования. Модель предназначена для использования в информационной технологии анализа распределенных имитационных моделей

Ключевые слова: распределенное имитационное моделирование, алгоритмическая модель, технология анализа

М. А. Волк

Кандидат технических наук, доцент*

E-mail: volk@kture.kharkov.ua

Р. Н. Гридель

Аспирант*

E-mail: 7z@ukr.net

С. Н. Саранча

Кандидат технических наук, доцент*

E-mail: softpro@kture.kharkov.ua

Д. А. Гавриш

Аспирант*

E-mail: gavden@gmail.com

*Кафедра электронных вычислительных машин
Харьковский национальный университет радиоэлектроники
пр. Ленина, 14, г. Харьков, Украина, 61166

1. Введение

Использование технологий распределенного имитационного моделирования в последнее время позволило значительно расширить класс реализуемых имитационных моделей.

Этому процессу способствует интенсивное развитие облачных технологий, предоставляющих мощные вычислительные ресурсы, такие как локальные и глобальные компьютерные сети, суперкомпьютеры, кластера.

Одной из особенностей использования подобных ресурсов является значительная стоимость эксплуатации.

Одной из обеспечивающих технологий распределенного имитационного моделирования является технология анализа моделей, поступающих на исполнение.

Анализ проводится с целью получения характеристик о распределенной модели, которые используются средой моделирования для эффективного статического и/или динамического управления распределенными ресурсами.

2. Анализ литературных данных и постановка проблемы

Сложность задачи анализа распределенных моделей приводит к созданию специальных технологий, обеспечивающих получение априорных и экспериментальных характеристик и их использование в системах распределения ресурсов.

При этом можно выделить два основных направления исследований. 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 – аппарат сетей Петри [2]. В этих случаях разработано множество методических рекомендаций, научных аналитических публикаций, которые помогут разработчику эффективно выбрать нужные средства моделирования, а задачами распределения занимается среда исполнения модели [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.Mitranі [5]. Дальнейшее развитие он получил в работах украинских ученых [9 – 11]. Авторами статьи были разработаны математические модели процессов распределенной имитации. Так, в работе [11] рассматривается обобщенная модель, в работах [12, 13] приведены модификации модели для консервативных и оптимистических методов синхронизации.

В данной статье приводится алгоритмическая модель, учитывающая различные методы синхронизации, а также возможность параллельного независимого расчета определенных параметров.

3. Цель и задачи статьи

Целью работы является рассмотрение алгоритмической модели процесса распределенной имитации. В статье рассматриваются основные параметры модели, непосредственно алгоритм получения основных характеристик процесса имитации, обсуждаются особенности его использования при разработке методов и технологии анализа распределенных имитационных моделей.

4. Алгоритмическая модель процесса распределенной имитации

На основе математических моделей, приведенных в работах [11 – 13] разработана алгоритмическая модель процесса функционирования распределенной имитационной модели, целью реализации которой является вычисление основных параметров критериев оценивания распределенной имитационной модели, к которым относятся:

i – номер шага моделирования;

T_i – глобальное модельное время, достигнутое моделью на i -ом шаге моделирования;

TR_i – реальное (астрономическое) время, затраченное системой моделирования на выполнение i шагов модели;

TR_i^n – реальное (астрономическое) время, затраченное системой моделирования на выполнение i шагов n -ой частной модели;

V_i – объем виртуальной памяти, занимаемый моделью по достижению i -го шага моделирования;

V_i^n – объем виртуальной памяти, занимаемый n -ой частной моделью по достижению i -го шага моделирования.

Для упрощения представления алгоритма введем вектор α , бинарный элемент α_m которого определяет наличие (отсутствие) определенного параметра распределенной имитационной модели или является

признаком модификации алгоритма синхронизации, который реализован в данной распределенной имитационной модели:

$$\bar{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_M\}, \quad \alpha_m \in \{0, 1\}. \quad (1)$$

Введем следующие значения элементов вектора:

$\alpha_1 = 1$, если требуется вычислять временные параметры функционирования распределенной имитационной модели (TR_i^n, TR_i), и $\alpha_1 = 0$, если временные параметры вычислять не нужно;

$\alpha_2 = 1$, если требуется вычислять изменения объемов памяти при функционировании распределенной имитационной модели (V_i, V_i^n), и $\alpha_2 = 0$, если параметры объемов памяти вычислять не нужно;

$\alpha_3 = 1$, если используется оптимистический алгоритм синхронизации частных моделей, и $\alpha_3 = 0$, если консервативный алгоритм.

Обобщенная математическая модель, без учета методов синхронизации распределенных имитационных моделей, представляет собой следующую систему выражений:

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

где t_i^s - время сохранения состояния модели на i -м шаге моделирования; t_i^m - время передачи данных; $\text{time}(\phi_i^n)$ - 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 - время работы служб системы моделирования на i -ом шаге моделирования; TR_i^n - локальное модельное время n -ой частной модели на i -ом шаге моделирования; T_i - глобальное модельное время на i -ом шаге моделирования; ξ_i^n - значение времени, характеризующее внутреннюю работу процессов между двумя соседними шагами моделирования;

V_i^{ns} – объем дампа памяти для n -ой частной модели на i -ом шаге моделирования; ΔV_i^n - приращение объема памяти n -ой частной модели на i -ом шаге моделирования; $sizeof$ – функция определения объема занимаемой памяти; Λ - множество активностей управления данными модели; Δ - множество активностей взаимодействия моделей; $TR_0, TR_0^n, t_0^n, V_0^n, V_0$ - начальные значения соответствующих параметров.

Приведем алгоритм процесса функционирования распределенной имитационной модели, построенный на базе математической модели (2) с учетом существования вектора (1).

1. Инициализация и загрузка параметров модели. На данном этапе могут быть использованы априорные параметры модели, а также параметры, полученные из предыдущих имитационных экспериментов, которые хранятся в базе данных имитационной среды моделирования.

2. Установление начальных нулевых значений следующим параметрам модели, вычисляемым в процессе выполнения алгоритма:

$$TR_0 = 0, T_0 = 0, TR_0^n = 0, TP_0^n = 0, t_0^n = 0, (n = \overline{1, N}).$$

3. Если $\alpha_2 = 1$ вычисляем начальные объемы памяти, занимаемые моделью на вычислительном ресурсе:

$$\begin{cases} V_0^n = sizeof(\Lambda^n \cup \Delta^n); \\ V_0 = \sum_{n=1}^N V_0^n; \end{cases} \quad (3)$$

Следует обратить внимание на тот факт, что некоторые частные модели могут размещаться на одном вычислительном ресурсе. В этом случае можно добавить вычисление объемов памяти для ресурса, суммируя объемы памяти частных моделей, расположенных на нем.

4. Начало цикла шага моделирования, $i=0$.

5. Если $\alpha_1 = 0$, переход к п. 13 алгоритма.

6. Инициализируем переменные расчета реального времени выполнения модели для i -го шага моделирования.

7. Инициализируем цикл текущего номера частной модели $n=1$;

8. Вычисляем время, затрачиваемое на выполнение n -ой частной модели на i -ом шаге моделирования, которое, в общем случае, является функцией, зависящей от поведения активности передачи данных δ_i^{nk} и размера передаваемых данных ($buff_i^{nk}$):

$$\Delta TR_i^n = \begin{cases} t_i^n, & \text{если } t_i^n = T_i \\ 0, & \text{если } t_i^n \neq T_i \end{cases} + \sum_{k=1}^N a^{nk} \cdot t_i^m(\delta_i^{nk}, sizeof(buff_i^{nk})). \quad (4)$$

9. Вычисляем реальное время, затраченное на выполнение n -ой частной модели за все шаги моделирования, начиная с 0-го до i -го включительно:

$$TR_{i+1}^n = TR_i^n + \Delta TR_i^n. \quad (5)$$

10. Переходим к следующей частной модели $n=n+1$.

11. Если все модели не перебраны ($n \leq N$), переходим к п. 8.

12. Вычисляем реальное время, затраченное на выполнение всей модели за все шаги моделирования, начиная с 0-го до i -го включительно:

$$TR_{i+1} = TR_i + \max_{n=1, N} (time(\phi_i^n) + \Delta TR_i^n) + TR_i. \quad (6)$$

Функция $time(A)$ определяет реальное время выполнения активности ϕ_i^n .

13. Если $\alpha_2 = 0$, переход к п. 25 алгоритма.

14. Инициализируем цикл текущего номера частной модели $n=1$;

15. Если $\alpha_3 = 1$, переход к п. 19 алгоритма.

16. Вычисляем приращение объема памяти n -ой частной модели на i -ом шаге моделирования:

$$\Delta V_i^n = \begin{cases} V_i^{ns}, & \text{если } t_i^n = T_i \\ 0, & \text{если } t_i^n \neq T_i \end{cases}. \quad (7)$$

17. Вычисляем общий объем памяти n -ой частной модели и всей модели после i -го шаге моделирования:

$$V_{i+1}^n = V_i^n + \Delta V_i^n. \quad (8)$$

18. Переход к п.23 алгоритма.

19. Вычисляем приращение объема памяти n -ой частной модели на i -ом шаге моделирования:

$$\Delta V_i^n = \begin{cases} V_i^{ns} + V_i^{nm} - \sum_{j=1}^M \tau_j V_{ij}^{nm}, & \text{если } t_i^n = T_i \\ 0, & \text{если } t_i^n \neq T_i \end{cases}. \quad (9)$$

20. Вычисляем уменьшение объема памяти частной модели за счет удаления дампов памяти при откатах и продвижении глобального модельного времени:

$$\Delta V_i^{rn} = \sum_{j=1}^i V_j^n, (\forall j, j < i_{GVT} \vee j > i'). \quad (10)$$

21. Вычисляем общий объем памяти n -ой частной модели после i -го шаге моделирования:

$$V_{i+1}^n = V_i^n + \Delta V_i^n - \sum_k \Delta V_k^n - \Delta V_i^{rn}, \forall k \in [q, i]. \quad (11)$$

22. Переходим к следующей частной модели $n = n+1$.

23. Если все модели не перебраны ($n \leq N$), переходим к п.15.

24. Вычисляем общий объем памяти модели после i -го шаге моделирования:

$$V_{i+1} = V_i + \sum_{n=1}^N \Delta V_i^n - \begin{cases} \Delta V_i^{rn}, & \text{если } \alpha_3 = 1 \\ 0, & \text{если } \alpha_3 = 0 \end{cases}. \quad (12)$$

25. Определяем для каждой модели время будущего события:

$$t_{i+1}^n = \begin{cases} t_i^n + \xi_i^n, & \text{если } t_i^n = T_i \\ t_i^n, & \text{если } t_i^n \neq T_i \\ t_i^n, & \text{если } \exists i'_n \end{cases} \quad (13)$$

26. Продвигаем модельное время:

$$T_{i+1} = \min_{n=1, N} (t_{i+1}^n). \quad (14)$$

27. Увеличиваем значение шага моделирования $i=i+1$.

28. Если не достигнуто условие достижения конца моделирования, переходим к п.5 алгоритма.

30. Фиксация номера шага окончания моделирования $I = i$.

31. Формирование отчетов по результатам моделирования.

32. Конец алгоритма.

Выходной информацией, полученной в результате выполнения алгоритма, является кортеж параметров оценивания распределенной имитационной модели:

$$O = \langle I, \overline{T}_i, \overline{TR}_i, \overline{TR}_i^n, \overline{TP}_i^n, \overline{V}_i, \overline{V}_i^n \rangle, i = \overline{1, N}. \quad (15)$$

Отметим, что кортеж содержит вектора параметров, отражающих их изменение во времени. Элементы векторов с индексом $i = I$ являются значениями соответствующих параметров на момент окончания процесса моделирования.

Основные параметры модели, находящиеся в кортеже (15), вычисляются независимо друг от друга. Данный факт определяет возможность распараллеливания алгоритма.

Предпочтительным в данном случае является применение многопоточного программирования, например, с использованием интерфейса параллельного программирования OpenMP, что позволит повысить эффективность алгоритма на многоядерных процессорах и многопроцессорных компьютерах. При значительных размерах модели возможно применение распределенного программирования, например, на основе интерфейса MPI (Message Passing Interface), так как вычисление параметров модели возможно без обмена данными.

Данный алгоритм лег в основу ряда методов анализа распределенных имитационных моделей.

В частности, был разработан метод сравнения эффективности распределенных имитационных моделей по максимальному продвижению модельного времени. Его применение позволяет оценить значение модельного времени при фиксированном количестве шагов

моделирования для разных методов синхронизации моделей.

Метод получения динамических характеристик виртуальной памяти позволяет эффективно управлять распределением ресурсов на основе полученных оценок использования частными моделями оперативной и внешней памяти. При этом учитываются не только статические характеристики занимаемой памяти частной модели в момент распределения, но и динамические, возникающие в процессе функционирования модели. В большинстве случаев, в процессе имитации объем памяти, занимаемый частной моделью, растет за счет дампов памяти, создания новых сообщений, внутренних объемов динамической памяти. В последнем случае может возникнуть ситуация переполнения виртуальной памяти ресурса, что приведет к отказу ресурса для системы моделирования. Предложенный метод может быть использован для оценки качества полученного множества возможных распределений или до начала поиска оптимального распределения для исключения некоторых подмножеств вариантов распределений. Последнее применение метода позволит не только исключить тупиковые варианты распределения, но и сократить время работы алгоритмов распределения.

Также был разработан метод оценки простоя ресурсов при различных вариантах распределения ресурсов и различных алгоритмах синхронизации. Метод может быть использован при построении динамических алгоритмов планирования, а также эффективно использовать вычислительные ресурсы с учетом локальных задач и служебных программ среды моделирования.

5. Выводы

В статье рассмотрена алгоритмическая модель процесса функционирования распределенной имитационной модели под управлением консервативных и оптимистических алгоритмов синхронизации. Рассмотрены возможности распараллеливания алгоритма. Описаны примеры разработанных методов анализа распределенных имитационных моделей на основе предложенной алгоритмической модели. Полученные результаты могут быть использованы в ряде других методов анализа распределенных имитационных моделей для разных алгоритмов синхронизации, рассмотрение которых выходит за рамки данной работы. Кроме этого, они могут использоваться в методах статического и динамического распределения ресурсов в задачах распределенного имитационного моделирования.

Литература

1. Томашевский, В. Имитационное моделирование в среде GPSS [Текст] / В. Томашевский, Е. Жданова. – М.: Бестселлер, 2003. – 416с.
2. Юдицкий, С. А., Метод анализа конфигураций организационных систем на сетях Петри [Текст] / С. А. Юдицкий, И. А. Мурадян. // УБС, №16, 2007. – С.163–170.
3. Крэйн, М. Введение в регенеративный метод анализа моделей [Текст] / М. Крэйн, О. Лемуан. – М.: Наука, 1982.–104с.
4. Окольнішников, В. В. Разработка средств распределенного имитационного моделирования для многопроцессорных вычислительных систем [Текст]: диссертация д-ра техн. наук: 05.13.18/ В. В. Окольнішников. – Новосибирск, 2006. – 227с.

5. Mitra, D. Analysis and optimum performance of two message-passing parallel processors synchronized by rollback [Текст] / D. Mitra, I. Mitrani //Performance'84, pp.35-50, 1984.
6. Вознесенская, Т. В. Математическая модель алгоритмов синхронизации времени для распределённого имитационного моделирования [Текст] / Т. В. Вознесенская // Программные системы и инструменты. Тематический сборник факультета ВМиК МГУ им. Ломоносова №1. 2002. – С.56-66.
7. Вознесенская, Т. В. Математическая модель для анализа производительности распределенных систем имитационного моделирования [Текст] / Т.В. Вознесенская //Искусственный интеллект (Донецк), 2002 No 2, с.74-78.
8. Миков, А. И. Программные средства оптимизации распределенного имитационного эксперимента [Текст]:сб. науч. тр./ А. И. Миков, Е. Б. Замятина, А. А. Козлов // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции (21-26 сентября 2009 г., г. Новороссийск).- М.: Изд-во МГУ, 2009. - 524 с.
9. Ladyzhensky, Y.V. Software system for event-driven logic simulation [Текст]/ 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 [Текст]/ Ю. В. Ладыженский, Г. А. Тесленко // Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. – Донецк, 2008, №9. – с. 55-62.
11. Волк, М. А. Анализ распределенных имитационных моделей в гетерогенных вычислительных системах [Текст]/ Волк М. А. // Науковий вісник Чернівецького національного університету імені Юрія Федьковича. Серія: Комп'ютерні системи та компоненти. – Том 1, випуск 2. – Чернівці: ЧНУ, 2010. – С.35-39.
12. Волк, М. А. Анализ распределенных имитационных моделей с консервативными алгоритмами синхронизации [Текст] / Волк М. А., М. А. Филимончук, М. Ал Шиблак, Р. Н. Гридель.// Збірник наук. Праць ХУПС.- 2012, Вип. 1(30), сс. 95-98.
13. Волк, М. А. Анализ распределенных имитационных моделей с оптимистическими алгоритмами синхронизации [Текст] / М. А. Волк, Р. Н. Гридель, М. Ал Шиблак // Системи обробки інформації. – 2013. –Вип. 1(108). - С.35-40.