



МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

С.И. Олзоева

МЕТОД АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ В РАСПРЕДЕЛЕННОМ МОДЕЛИРОВАНИИ СИСТЕМ

Аннотация. Рассматривается задача декомпозиции имитационных моделей систем для организации распределенного моделирования сложных систем на многопроцессорных вычислительных системах. Для ускорения процесса моделирования необходимо разбить моделирующую программу на части, выполняющиеся параллельно на разных процессорах. Предлагается подход к использованию метода автоматической классификации, позволяющий автоматизировать распараллеливание моделирующих программ.

Ключевые слова: имитационное моделирование, многопроцессорные вычислительные системы, декомпозиция имитационной модели, автоматическая классификация, распределенное моделирование, сложные системы, ускорение вычислительного процесса, моделирующие программы, параллельные вычисления.

Введение

Большие вычислительные затраты в имитационном моделировании сложных систем требуют разработки методов ускорения процесса моделирования, что можно осуществить на многопроцессорных вычислительных системах. Для этого нужно разбить моделирующую программу на части, выполняющиеся параллельно на разных процессорах. Простое распределение модулей моделирующей программы по процессорам не способствует эффективному ускорению процесса имитации из-за многочисленных передач данных между модулями имитационной модели (ИМ), сильно связанных по данным и управлению. Пересылка данных является наиболее “времяемким” процессом, поэтому необходимо решить задачу автоматизированной декомпозиции программного обеспечения имитационной модели на блоки, по критерию минимизации обмена данными между блоками ИМ.

Задача декомпозиции имитационной модели

Имитационная модель сложной системы строится как множество процессов, исполняемых квази-параллельно в условном времени и взаимодействующих между собой через общие наборы данных. ИМ имеет частично упорядоченное в модельном времени множество событий, изменений состояний системы. При имитации событий выполняются обрабатывающие модули (ОМ), соответствующие наступлению событий.

Обозначим через $E = \{e_1, e_2, \dots, e_n\}$ — множество всех возможных событий в модели; через $E^* = \{s_1, s_2, \dots, s_k, \dots, s_m\}$ — набор всех возможных конечных последовательностей событий $s_k = e_1 e_2 \dots e^k$ в

E. Временная последовательность учитывает времена наступления событий и описывает логическое поведение системы во времени:

$$S = (e_1, t_1)(e_2, t_2)(e_3, t_3) \dots, \quad t_1 < t_2 < t_3 \dots,$$

Имитационная модель выполняет последовательности вида:

$$s(w) = (e_1(w), t_1(w))(e_2(w), t_2(w))(e_3(w), t_3(w)) \dots, \quad (1)$$

где e_i, t_i — случайные переменные, w — выборочная точка, так что $s(w)$ — выборочный путь развития моделирующего процесса. Выполнение программы имитационной модели на обыкновенном компьютере есть воспроизведение последовательности дискретных событий $s(w)$ и обработка этих событий. В случае выполнения имитационной программы на многопроцессорной системе на каждом шаге имитации одновременно может выполняться целый ансамбль обрабатываемых модулей, расположенных на разных процессорах и независимых по данным в рассматриваемый момент времени. Как на обыкновенном, так и на параллельном компьютере искомые оценки характеристик моделируемой системы получают в результате выполнения последовательности ОМ имитационной модели, которая, в свою очередь определяется последовательностью наступления событий (1).

Все ОМ имеют некоторое число аргументов. Конкретными значениями аргументов являются либо входные данные, либо результаты выполнения других ОМ. Любой обрабатывающий модуль — потребитель аргументов не может начать выполняться раньше, чем закончится выполнение всех ОМ — поставщиков для него аргументов. То, какие модули являются поставщиками аргументов для конкретного ОМ, зависит от событий, свершившихся на предыдущем шаге. У одного и того же ОМ на разных шагах имитации могут быть разные поставщики. Следовательно, взаимная связность событий на множестве $E = \{e_1, e_2, \dots, e_n\}$ определяет конкретную последовательность реализаций ОМ. А взаимосвязь событий, в свою очередь, зависит от структуры конкретной моделируемой системы, правил функционирования, распределений вероятностей и частоты взаимодействия элементов системы — все это вместе составляет входные данные имитационной модели.

Задача декомпозиции ИМ состоит в том, чтобы на множестве всех возможных событий $E = \{e_1, e_2, \dots, e_n\}$ выделить группы событий, которые наиболее связаны по входным данным. Это возможно вследствие того, что в некотором событии участвуют не все элементы моделируемой системы, и его наступление приводит лишь к локальному изменению состояния системы, то есть при свершении события следует анализировать изменения состояний лишь части элементов системы.

Например, пусть имеется множество возможных событий $E = \{e_1, e_2, \dots, e_r\}$. Между которыми существует такая связь: свершение события \hat{a}_1 обязательно влечет за собой событие \hat{a}_3 , а свершение события \hat{a}_2 влечет за собой событие \hat{a}_r , то есть \hat{a}_1 и \hat{a}_3 более сильно связаны, чем \hat{a}_1 и \hat{a}_2 . При этом моментами времени наступления событий $\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_r$ являются, соответственно t_1, t_2, t_3 и t_r , при чем $t_1 \leq t_2 < t_3 \leq t_r$. Тогда, при выполнении имитационной программы на обыкновенном компьютере реализуется последовательность действий:

$$s(w) = (e_1(w), t_1(w))(e_2(w), t_2(w))(e_3(w), t_3(w)) \dots (e_r(w), t_r(w)).$$

Так как \hat{a}_1 и \hat{a}_2 обрабатываются модулями, независимыми по входным аргументам, то реализацию последовательности $s(w)$ на параллельном компьютере можно осуществить, объединив события \hat{a}_1 и \hat{a}_3 в группу $E^1 = \{e_1, e_3\}$, а события \hat{a}_2 и \hat{a}_r в группу $E^2 = \{e_2, e_r\}$, $E = E^1 \cup E^2$. Соответственно, последовательность действий $s(w)$ разбивается на последовательности $s^1(w) = (e_1^1(w), t_1(w))(e_3^1(w), t_3(w))$ и $s^2(w) = (e_2^2(w), t_2(w))(e_r^2(w), t_r(w))$, которые можно выполнить на разных процессорах. Взаимосвязь между процессами $s^1(w)$ и $s^2(w)$ сохраняется в виде обмена сообщениями, необходимого для синхронизации модельного времени, но обмен будет минимальным ввиду отсутствия обращения к общим наборам данных.

Таким образом, необходимо выделить во множестве всех возможных событий $E = \{e_1, e_2, \dots, e_n\}$ такие группы событий E^1, E^2, \dots, E^m , чтобы взаимосвязь событий внутри группы была максимальной, а между группами минимальной. Это можно осуществить, применив методы кластерного анализа.

Метод решения задачи декомпозиции

Для формализации описания ИМ сложной системы используется РДО — метод [1]. В ИМ элементы моделируемой системы являются ресурсами и образуют множество R . В некотором событии e участвуют не все ресурсы системы. Ресурсы, состояния которых изменяются, называются релевантными событию e . Множество релевантных ресурсов событию e обозначим: $R^e, R^e \in R$.

Ресурсы в процессе взаимодействия между собой выполняют определенные действия, связанные с определенными событиями. Если в описании действия исключить привязку к временной оси, оставив лишь длительность его выполнения $\Delta t = t_e - t_p$, то получим виртуальное (возможное) действие. В имитационной модели множество V представляет собой множество виртуальных действий в исследуемой системе. Взаимосвязь действий, происходящих в ИМ, определяется множеством релевантных ресурсов каждого действия. Степень совпадения этих множеств у различных действий характеризует объем взаимосвязи между действиями. Одинаковые действия, отличающиеся множествами релевантных ресурсов, можно выполнить параллельно на разных процессорах вычислительной системы, осуществляя при этом координацию разнесенных действий в едином модельном времени. Следовательно, задача декомпозиции ИМ состоит в классификации множества виртуальных действий модели по степени сходства их множеств релевантных ресурсов. Это можно осуществить, применив методы кластерного анализа или автоматической классификации.

Для отражения динамических характеристик системы формируется матрица признаков X , содержащая качественную информацию о структуре связей моделируемой системы и количественную информацию об интенсивности связи действий имитационной модели. Эта информация извлекается из входных данных модели, которые зависят от конкретной моделируемой сложной системы [2]. Итак, задано конечное множество объектов $V = (v_1, v_2, \dots, v_n)$, в котором каждый объект v_j имеет набор признаков $X_j = (x_{j1}, x_{j2}, \dots, x_{jp})$. Необходимо произвести разбиение множества V на классы наиболее связанных объектов по критерию минимизации связей между классами. Для решения задачи вводится метрика количественной характеристики различия объектов:

$$d^2(X_i, X_j) = \sum_{k=1}^p (x_{ki} - x_{kj})^2$$

Разбиение множества виртуальных действий ИМ производится с использованием алгоритма, заключающегося в последовательном объединении исходных элементов и построении дендрограммы. Критерием объединения кластеров является значение целевой функции

$$W: d_{i,k} = \min_{k=L,m} \{d_{ik}\}, \{k \neq i\}$$

где i и k — номера кластеров, m — количество кластеров. В последовательности возрастающих величин целевой функции W выявляется скачок. Разрезание дендрограммы в месте «скачка» дает число кластеров, состоящих из наиболее связанных между собой объектов. В результате кластеризации множества виртуальных действий имитационной модели получаем M непересекающихся кластеров, где $V_i = (v_1^i, v_2^i, \dots, v_{n_i}^i)$ — есть i -й кластер, состоящий из n_i действий

$$n = \sum_{i=1}^M n_i$$

Число полученных кластеров определяет количество блоков декомпозированной ИМ. Виртуальные действия, входящие в один кластер, формируют единый блок имитационной модели. Событиям, определяющим действие v_p , соответствуют конкретные обрабатывающие модули имитационной мо-

дели. По результатам произведенной классификации виртуальных действий формируются: матрица сопряжения полученных блоков ИМ; матрица интенсивностей взаимодействий между блоками; распределенная база данных моделирования.

Заключение

Предлагаемый подход декомпозиции имитационной модели с использованием метода автоматической классификации позволяет автоматизировать распараллеливание моделирующих программ по критерию минимизации межпроцессорных обменов данными для их выполнения на многопроцессорной вычислительной платформе.

Список литературы:

1. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Языки РДО. — М.: АНВИК, 1998, — 427 с.
2. Олзоева С.И. Распределенное моделирование в задачах разработки АСУ. — Улан-Удэ: Изд-во ВСГТУ, 2005, — 219 с.

References (transliteration):

1. Emel'yanov V.V., Yasinovskii S.I. Vvedenie v intellektual'noe imitacionnoe modelirovanie slozhnyh diskretnykh sistem i processov. Yazyki RDO. — M.: ANVIK, 1998, — 427 s.
2. Olzoeva S.I. Raspredelennoe modelirovanie v zadachah razrabotki ASU. — Ulan-Ude: Izd-vo VSGTU, 2005, — 219 s.