УДК 004.94

Д.А. Гавриш, С.Н. Саранча

Харьковский национальный университет радиоэлектроники, Харьков

МЕТОД РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ ПРИ ПАРАЛЛЕЛЬНОМ МОДЕЛИРОВАНИИ СЛОЖНЫХ ЦИФРОВЫХ СИСТЕМ В ГОМОГЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Были предложены методы распределения заданий по рабочим станциям при проведении распределенного моделирования сложного электронного компонента. Были сформулированы критерии оценки эффективности методов распределения. Рассмотренные алгоритмы учитывают трудоемкость каждого процесса в описании цифрового компонента и характеристики рабочей станции. Был рассмотрен способ оценки трудоемкости процесса. Была проведена серия экспериментов для сравнения эффективности работы каждого метода. Рассмотренные методы могут использоваться для оптимизации и ускорения процесса моделирования сложных цифровых систем с использованием методов распределенного моделирования.

Ключевые слова: *процесс, вычислительная сложность, ресурсоемкость.*

Введение

При разработке сложных цифровых систем возникает потребность в проведении моделирования этой системы с высокой степенью реалистичности. Имитационное моделирование может использоваться для проведения экспериментов с моделью цифровой системы с целью ее верификации. Данная потребность объясняется высокой стоимостью проведения натурных экспериментов с реальной системой.

Компьютерное моделирование как метод исследований является развитием математического моделирования. В основе компьютерной модели лежит математическая модель, которая строится на основе описания структуры и поведения исследуемой системы на определенном языке программирования.

Рост интереса к возможности проведения распределенного моделирования связан с проблемами ограниченных вычислительных способностей отдельной рабочей станции. При росте сложности проектируемых и верифицируемых цифровых систем, повышении требований к точности и объемам моделирования стало невозможным проведение этого процесса в рамках одной рабочей станции (или этот процесс может занять слишком много времени). Одним из возможных решений данной проблемы является использование распределенного моделирования цифровой системы на множестве рабочих станций, соединенных коммуникационной системой.

Также необходимо учитывать высокую степень параллелизма в описании цифровых систем, что делает возможным проведение моделирования отдельных ее компонентов на разных рабочих станциях с приемлемыми затратами вычислительных и коммуникационных ресурсов для синхронизации.

При организации распределенного моделирования возникает проблема выделения параллельных блоков описания модели и оптимального распределения этих блоков между рабочими станциями.

При выполнении распределенного моделирования сложных цифровых систем необходимо добиться равномерной загрузки всех рабочих станций, что позволяет снизить как время простоя отдельных рабочих станций, так и уменьшить общее время выполнения процесса моделирования. Для этого необходимо рассмотреть производительность отдельной рабочей станции, вычислительную сложность и ресурсоемкость отдельного процесса, а также алгоритмы распределения задач.

Анализ литературных данных. Алгоритм распределения должен быть оптимальным для любой имитационной модели. Однако создание такого алгоритма является достаточно сложной задачей, т.к. алгоритм распределения должен учитывать специфику модели. Имитационные модели могут значительно отличаться друг от друга своими характеристиками: структурой, интенсивностью обмена информацией между отдельными компонентами, алгоритмами поведения этих компонентов.

В настоящее время существуют такие известные технологии распределённого имитационного моделирования, как SPEEDES (Synchronous Parallel Environment for Emulation and Discrete Event Simuation), PARASOL (Parallel Solution) и HLA (High Level Architecture) [1, 2, 3].

Представляется возможным также выделить распределённые версии системы моделирования Triad.Net автоматизированного проектирования Triad [4, 5].

Система имитации Triad.Net является объектно-ориентированной. В последних разработках рассматривалась возможность применения Triad.Net к исследованию телекоммуникационных сетей.

К особенностям архитектуры системы имитации и её реализации [6] следует отнести: возможность выполнения операций над моделью во время процесса имитации, наличие гибкого и эффективного алгоритма исследования модели, описание которого отделено от описания самой модели.

Кроме того, следует отметить масштабируемость алгоритма синхронизации объектов модели (в системе реализованы консервативный и оптимистический алгоритмы синхронизации), возможность повторного использования кода (данные о модели и результатах моделирования хранятся формате XML) и т.д.

Постановка проблемы. При использовании современных языков описания цифровых и смешанных цифро-аналоговых систем, система моделирования оперирует с понятием «процесс», который является основным элементом для представления параллельных взаимодействующих подсистем. Процесс может представлять собой как отдельный оператор (например, параллельный оператор назначения сигнала), так и достаточно большой фрагмент последовательных операторов.

Для выполнения распределенного моделирования используется некое множество рабочих станций под управлением сервера, который осуществляет раздачу заданий на моделирование, где задание представляет собой отдельный процесс. Каждый процесс характеризуется таким набором параметров:

– трудоемкость процесса;

– требуемый минимальный и максимальный объем памяти.

Каждая рабочая станция моделирования обладает аналогичными характеристиками:

– производительность процессора и подсистемы памяти;

– объем установленной физической памяти и объем доступной памяти файла подкачки.

Задача равномерного распределения заданий на моделирование решается по-разному в зависимости от следующих условий:

– гомогенная или гетерогенная вычислительная среда;

– статическая или динамически изменяющаяся трудоемкость отдельных заданий.

С целью упрощения из дальнейшего рассмотрения исключены вопросы определения производительности подсистемы передачи данных (компьютерной сети), топологической связности процессов, времени, требуемого сервером для выполнения синхронизации процессов, а также производительность каждой рабочей станции и так далее. Это является предметом дальнейших исследований.

Таким образом, необходимо решить задачу равномерного распределения заданий со статическими характеристиками трудоемкости TN_i, минимального и максимального объема памяти ($RMIN_i$ и $RMAX_i$ соответственно) на однородные рабочие станции, обладающие идентичными значениями производительности процессоров PR и объемов установленной физической памяти RAM и файла подкачки SW.

Таким образом, все множество процессов требуется разбить на отдельные подмножества с соблюдением указанных условий:

$$P = \{P_i\};$$

$$P_j \cap P_k = \varnothing \quad \text{для всех } j \neq k;$$

$$\sum_i TN_{ij} \approx \text{const};$$

$$\sum_i RMIN_{ij} \leq RAM\ ; \qquad (1)$$

$$\sum_i RMAX_{ij} \leq RAM+SW\ . \qquad (2)$$

Изложение основного материала

1. Метод определения трудоемкости процесса

Трудоемкость алгоритма – количество вычислительной работы, требуемой для реализации алгоритма.

Так как используемые алгоритмы в параллельных блоках, как правило, имеют много параметров для фиксированного размера входных данных, необходимо оценить его трудоемкость в худшем, среднестатистическом и лучшем случае. В дальнейшем при расчетах будет использоваться среднестатистическое значение трудоемкости.

Для оценки трудоемкости процесса необходимо определить «элементарные операции». В качестве таких операций предлагается следующий набор:

– простое присваивание: a←b;

– одномерная индексация a[i]: (адрес (a)+i×длина элемента);

– арифметические операции: (×, /, -, +);

– операции сравнения: a<b;

– логические операции (l_1) {or, and, xor} (l_2).

После введения элементарных операций анализ трудоемкости основных алгоритмических конструкций в общем виде сводится к следующим положениям:

Трудоемкость конструкции «следование» есть сумма трудоемкостей блоков, следующих друг за другом.

$$F_{\text{следование}} = f_1 + \ldots + f_k,$$

где k – количество блоков.

Конструкция «Ветвление» синтаксически представляется с помощью блока if или switch, при этом каждая ветвь имеет некую определенную вероятность выполнения.

if (l) then
  f_{then} с вероятностью *p*
else
  f_{else} с вероятностью *(1–p)*

Общая трудоемкость конструкции «Ветвление» требует анализа вероятности выполнения переходов на блоки «Then» и «Else» и определяется как:

$$F_{"ветвление"}=f_{then}\times p+f_{else}\times(1-p).$$

Конструкция «Цикл» синтаксически представляется с помощью блоков for, while, do..while, foreach.

fori← 1 toN
  // операторы тела цикла
  // трудоемкостью $f_{тела\ цикла}$
end

Выполнению цикла предшествует одна дополнительная инструкция, связанная с инициализацией параметра цикла. В каждой итерации помимо операторов тела цикла также выполняются 3 дополнительные инструкции: проверка условия окончания цикла, модификация параметра цикла, переход в начало цикла. После сведения конструкции к элементарным операциям ее трудоемкость определяется как:

$$F_{"цикл"} = 1+N\times(3+f_{"тела\ цикла"})$$

Использование разделения описания процесса на простые конструкции позволяет провести оценку трудоемкости процесса.

Для оценки использования памяти необходимо проанализировать область декларации каждого блока процесса.

2. Описание алгоритмов распределения вычислительных заданий при распределенном моделировании

Входными данными для алгоритма распределения вычислительных заданий являются:

– характеристики динамической памяти (объем физической памяти и файла подкачки RAM и SW) – эта информация позволяет учесть текущую степень загруженности рабочей станции;

– статическая оценка трудоемкости выполнения каждого параллельного процесса TN_i;

– характеристики использования памяти параллельным процессом RMIN и $RMAX_i$.

Так как вычислительная среда является гомогенной, то характеристики динамической памяти можно применить ко всем рабочим станциям в системе.

Требования к алгоритмам распределения вычислительных заданий:

– объема физической памяти должно быть достаточно для минимальных требований памяти параллельного процесса (1);

– объема динамической памяти должно быть достаточно для работоспособности системы в самом худшем случае (2);

– $\sum_i TN_{ij} \approx \text{const}$ – суммарная вычислительная трудоемкость параллельных процессов на каждой рабочей станции должна быть одинаковой в идеальном случае.

Для упрощения вычислений предлагается отсортировать параллельные процессы $P=\{P_i\}$ в порядке увеличения их сложности, и определить коэффициент сложности каждого процесса $C_i \in [0:1]$, разделив его вычислительную сложность на значение максимальной трудоемкости $P_{MAX}=MAX(P_i)$.

Для оценки использования памяти необходимо проанализировать область декларации каждого блока процесса.

Для реализации данного метода предложено несколько алгоритмов, которые имеют различные характеристики времени работы и полученных коэффициентов эффективности. В дальнейшем будут рассмотрены следующие методы:

– метод полного перебора;
– модифицированный метод полного перебора;
– ускоренный метод распределения.

3. Критерии оценивания методов распределения

Так как вычислительные задания имеют различные требования к динамической памяти и различную вычислительную сложность, равномерно распределить эти задания в большинстве случаев невозможно. В качестве основного критерия оценки метода распределения выбирается превышение суммарного значения коэффициентов заданий C значения 1:

$$K = \sum_{i,\ WU_i>AVG} WU_i\ ,$$

где $WU = \sum_j C_j$ – степень нагрузки рабочей станции; K – коэффициент использования системы, $AVG = \frac{\sum C_i}{N}$.

Дополнительным критерием оценивания эффективности метода распределения является количество итераций алгоритма.

4. Метод полного перебора

Этот алгоритм имеет самое большое время выполнения, но он гарантировано дает минимальное значение коэффициента K. При этом количество итераций алгоритма фиксированное, и описывается следующим выражением: $O=M^N$, где M – количество заданий, N – количество рабочих станций. Исходя из этого выражения, количество итераций имеет степенную сложность и при большом количестве

заданий обладает малой эффективностью. Алгоритм работы данного метода следующий:

1. Вычислить коэффициент использования вычислительной системы K для текущего распределения работ.

2. Сравнить этот коэффициент с полученным ранее более оптимальным коэффициентом, если таковой имеется.

3. Вычислить следующую комбинацию распределения работ по рабочим станциям.

4. Если все комбинации проанализированы – вернуть результат, иначе вернуться к пункту 1.

Полученный в результате работы этого алгоритма коэффициент эффективности будет использоваться для оценки эффективности работы других алгоритмов.

5. Модифицированный метод полного перебора

Данный метод позволяет получить такой же результат, как и предыдущий, но с меньшими временными затратами. В этом алгоритме из рассмотрения отбрасывается большое количество комбинаций, которые заведомо являются неэффективными. Распределение работ считается заведомо неэффективным, если $K \geq \underset{i}{MIN}(C_i)$. При вычислении K алгоритм анализирует задания от четвертого к первому и пошагово вычисляет нагрузку на каждую рабочую станцию, суммируя значения вычислительной сложности процессов, которые присвоены этой рабочей станции. Если эта нагрузка превышает значение $\underset{i}{MIN}(C_i)$, итерация прекращается, все последующие задания присваиваются WS_0, а текущее анализируемое задание отправляется следующей рабочей станции.

Таким образом, отбрасывается M^J заведомо неэффективных комбинаций, где j – индекс текущего анализируемого задания.

6. Ускоренный метод распределения

Этот алгоритм предоставляет результат за одну итерацию, при этом значение K не является самым оптимальным, но приемлемым. Этот метод пошагово заполняет рабочие станции, в первую очередь, распределяя самые сложные задачи, а потом, по мере возможности, – более простые.

Рассмотрим пример работы алгоритма для 10 задач и 5 идентичных рабочих станций.

Таблица 1

Пример входных распределений сложности вычислительных процессов

	T0	T1	T2	T3	T4
C_i	1.0	0.8	0.8	0.55	0.5

	T5	T6	T7	T8	T9
C_i	0.4	0.35	0.26	0.25	0.1

В этом случае $AVG = \frac{\sum C_i}{N} = \frac{5.01}{5} \approx 1$.

Задачи Task_7 с $C_i = 0.26$ и Task_8 с $C_i = 0.25$ остались незадействованными. В данной ситуации метод на второй итерации назначит WS_2 задачу Task_7 и WS_4 задачу Task_8. В итоге данный метод предложит вариант распределения, который приведен в табл. 3.

Коэффициент эффективности в этом случае равен $K = 0.06 + 0.1 = 0.16$, что очень близко к оптимальному результату.

Результат распределения задачи методом полного перебора приведен в табл. 5.

Таблица 2

Пошаговое выполнение первого этапа распределения

№	WS_0	WS_1	WS_2	WS_3	WS_4
1	T0	–	–	–	–
2	T0	T1 T7	–	–	–
3	T0	T1 T9	T2	–	–
4	T0	T1 T9	T2	T3 T5	–
5	T0	T1 T9	T2	T3 T5	T4 T6
∑	1.0	0.9	0.8	0.95	0.85

Таблица 3

Результаты работы алгоритма распределения

	WS_0	WS_1	WS_2	WS_3	WS_4
Tasks:	T0	T1 T9	T2 T7	T3 T5	T4 T6 T8
∑	1.0	0.9	1.06	0.95	1.1

7. Сравнение эффективности рассматриваемых методов

Самым быстрым алгоритмом является «ускоренный метод распределения». Этот метод дает приемлемый результат за минимальное время. Самый оптимальный результат можно получить, применив «Модифицированный метод полного перебора». Он работает быстрее метода полного перебора.

Таблица 4

Сравнение рассмотренных алгоритмов распределения задач

	Полный перебор	Модифицированный полный перебор	Ускоренный метод
Коэффициент эффективности	0.06	0.06	0.16
Количество итераций	9 765 625	3 841	1
Время выполнения	0.014 с	125.23 с	0.07 с

Таблица 5

Распределение задач методом полного перебора

	WS_0	WS_1	WS_2	WS_3	WS_4
Tasks:	T0	T1 T9	T2 T8	T3 T5	T4 T6 T7
∑	1.0	0.9	1.05	0.95	1.01

Коэффициент эффективности в оптимальном случае равен $K = 0.05 + 0.01 = 0.06 < 0.1$.

Выводы

Для оптимизации процесса распределенного моделирования сложных цифровых систем необходимо оптимально распределить параллельные логические процессы по рабочим станциям. Для этого необходимо учитывать параметры каждого отдельно взятого процесса: вычислительная сложность и затраты памяти, которые необходимы для моделирования. Также учитываются параметры производительности рабочей станции и характеристики динамической памяти. Было рассмотрено 3 алгоритма распределения процессов. Самым быстрым алгоритмом, который дает приемлемый результат, является ускоренный метод распределения, самым быстрым алгоритмом, который дает оптимальный результат, является ускоренный метод полного перебора. В качестве дальнейшего развития предлагается модифицировать данные методы для распределения задач в гетерогенной вычислительной среде. Для этого необходимо учитывать разные показатели производительности различных рабочих станций.

Список литературы

1. Миков А.И. Информационные процессы и нормативные процессы в IT [Текст] / А.И. Миков. – Книжный дом ЛИБРОКОМ, 2013. – 256 с.

2. Аврамчук Е.Ф. Технология системного моделирования [Текст] / Е.Ф. Аврамчук, А.А. Вавилов, С.В. Емельянов и др.; под общ. ред. С.В. Емельянова и др. – М.: Машиностроение; Берлин: Техник, 1998. – 520 с.

3. Бигдан В.В. Становление и развитие имитационного моделирования в Украине [Текст] /В.В. Бигдан, В.В. Гусев, Т.П. Марьянович, М.А. Сахнюк // Пр. міжнар. симп. «Комп'ютери у Європі. Минуле, сучасне та майбутнє». – К., 1998. – С. 182-193.

4. В.А. Горбачев. Технология моделирования систем [Текст]: учеб. пособие для студ. вузов / В.А. Горбачев. – Х.: Компания СМИТ, 2005. – 162 с.

5. Mikov A.I. Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems [Текст] / A.I. Mikov // Gero J.S. and F .Sudweeks F. (eds), Advances in Formal Design Methods for CAD, Preprints of the IFIPWG 5.2Workshop on Formal Design Methods for Computer-Aided Design, Mexico, 1995. – P. 105 -127.

6. Миков А.И. Система оперирования распределёнными имитационными моделями сетей телекоммуникаций [Текст] / А.И. Миков, Е.Б. Замятина, А.Х. Фатыхов // Труды Первой Всероссийской научной конференции «Методы и средства обработки информации». – М.: Изд-во МГУ, 2003. – С. 437-443.

7. Fujimoto R.M. Parallel and Distributed Simulation [Текст] / R.M. Fujimoto // Proc. of the Winter Simulation Conf. – 1999. – P. 122-131.

8. High Level Architecture Object Model Template Specification. Version1.3. [Текст] U.S. Department of Defense. 5 February, 1998 (27 July 1998 DocumentRelease).

9. Колмогоров А. Теория систем. Математические методы и моделирование [Текст] / А. Колмогоров, С. Новиков // Сб. статей под общ. ред. А. Колмогорова. – М: Мир, 1989. – 314 с.

Поступила в редколлегию 11.10.2013

Рецензент: д-р техн. наук, проф. О.Ф. Михаль, Харьковский национальный университет радиоэлектроники, Харьков.

МЕТОД РОЗПОДІЛУ ЗАВДАНЬ ПРИ ПАРАЛЕЛЬНОМУ МОДЕЛЮВАННІ СКЛАДНИХ ЦИФРОВИХ СИСТЕМ В ГОМОГЕННОМУ ОБЧИСЛЮВАЛЬНОМУ СЕРЕДОВИЩІ

Д.О. Гавриш, С.М. Саранча

Було запропоновано методи розподілу завдань по робочих станціях при проведенні розподіленого моделювання складного електронного компоненту. Були сформульовані критерії оцінки ефективності методів розподілу. Розглянуті алгоритми враховують трудомісткість кожного процесу в описі цифрового компоненту і характеристики робочої станції. Був розглянутий спосіб оцінки трудомісткості процесу. Була проведена серія експериментів для порівняння ефективності роботи кожного методу. Розглянуті методи можуть використовуватися для оптимізації і прискорення процесу моделювання складних цифрових систем з використанням методів розподіленого моделювання.

Ключові слова: *процес, обчислювальна складність, ресурсоемність.*

TASK MANAGEMENT METHOD FOR PARALLEL SIMULATION OF COMPLEX DIGITAL SYSTEM IN HOMOGENEOUS COMPUTER ENVIRONMENT

D.A. Havrysh, S.N. Sarancha

The method of task management in computer system was proposed to provide distributed simulation of complex model. The criterion of efficiency for methods was established. The considered algorithms take into account the computation complexity of each parallel block in the model. The set of experiments was provided for these methods to compare efficiency. These methods can be used to accelerate simulation process and increase efficiency of simulation with using of distributed simulation protocol.

Keywords: *task, computational complexity, resource consumption.*