

УДК 004.94

Д.А. Гавриш, С.Н. Саранча, М.А. Волк

Харьковский национальный университет радиоэлектроники, Харьков

МЕТОД РАСПРЕДЕЛЕНИЯ ЗАДАЧ С УЧЕТОМ ЗАТРАТ СИНХРОНИЗАЦИИ ПРИ ПАРАЛЛЕЛЬНОМ МОДЕЛИРОВАНИИ СЛОЖНЫХ ЦИФРОВЫХ СИСТЕМ В ГЕТЕРОГЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Для оптимизации процесса моделирования необходимо учитывать затраты среды моделирования на проведение синхронизации, которое основывается на передаче сообщений между рабочими станциями. Для этого целесообразно составить алгоритм распределения задач по рабочим станциям, который учитывает характеристики канала передачи данных каждой отдельно взятой рабочей станции и среднее количество передаваемой информации между параллельными процессами для обеспечения синхронизации.

Ключевые слова: процесс, синхронизация, пропускная способность, симметричный и несимметричный канал передачи данных.

Введение

При автоматизированном проектировании цифровых систем немаловажную роль играет среда моделирования как основной инструмент при функциональном тестировании разрабатываемого электронного компонента. В последнее время можно заметить, насколько быстро увеличивается сложность разрабатываемых систем на кристалле. Рост сложности проектируемых цифровых систем делает невозможным проведение процесса моделирования с использованием одной рабочей станции. Для решения такой проблемы предлагается использовать набор рабочих станций, соединенных коммуникационной сетью, для выполнения распределенного моделирования. Коммуникационная сеть при распределенном моделировании будет использоваться для передачи данных при инициализации среды моделирования, а также для обеспечения синхронизации между параллельными процессами путем передачи сообщений. В общем случае характеристики пропускной способности адаптеров локальных сетей отдельных рабочих станций могут отличаться. Это объясняется использованием различных аппаратных средств коммуникационной системы на различных рабочих станциях, а также различной степенью загруженности отдельных частей коммуникационной системы. На момент запуска имитационного моделирования рабочие станции, а также коммуникационная система могут использоваться другими пользовательскими процессами либо другими процессами моделирования, которые были запущены ранее. Также необходимо отметить различные характеристики сообщений, которые используются параллельными процессами для обеспечения синхронизации.

Возможность проведения распределенного моделирования электронного компонента основывает-

ся на его высокой степени параллелизма. Эта особенность предоставляет возможность моделировать отдельные параллельные блоки описания компонента на отдельных рабочих станциях с выполнением синхронизации данных между ними. Как правило, характеристики передаваемых сообщений при синхронизации различных блоков цифровой системы разные, что делает возможным использование гетерогенной вычислительной системы.

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

Анализ литературных данных

Как известно, среды моделирования электронных компонентов были спроектированы для проведения имитационного моделирования в пределах одной рабочей станции, и они не предусматривали дальнейшую оптимизацию за счет использования распределенных или кластерных вычислений. Статьи [1, 2] касаются возможности проведения распределенного моделирования дискретно-событийных систем. В статье [1] рассматривается реализация окружения для среды моделирования, которая позволяет выполнять распределенные вычисления, используя клиент-серверную архитектуру. В том же источнике [1] рассматриваются интерфейсы, которые должны реализовываться на стороне клиентского и серверного программного обеспечения. Для осуществления коммуникации между различными рабочими станциями был предложен протокол Simple Object Access Protocol (SOAP), который основывается на стандартах XML или HTML. Выбор данного стандарта был обусловлен использованием гетерогенной вычислительной системы.

В статье [2] рассматривается реализация консервативного способа организации распределенного

моделирования с использованием библиотеки SystemC. Программная реализация коммуникационного протокола позволяет использовать стандарты MPI и TCP/IP, что обеспечивает большую гибкость при организации распределенного моделирования. Недостатком данной реализации расширения библиотеки SystemC является ручное распределение параллельных блоков по различным рабочим станциям. Также необходимо вручную назначить набор сигналов для синхронизации.

В статье [3] были рассмотрены способы оптимизации процесса моделирования, основанного на использовании распределенных вычислительных ресурсов. При этом описание моделируемой системы должно быть реализовано на основе библиотеки SystemC с использованием концепций TLM (Transaction Level Modeling). В данном случае для проведения синхронизации используется передача транзакций между компонентами системы. Преимущество концепции TLM заключается в том, что информация группируется в транзакции, что уменьшает количество передаваемых сообщений, но, с другой стороны, размер каждой транзакции больше, чем значение отдельного дискретного значения.

Как известно, оптимизация вычислительного процесса с использованием распределенной вычислительной системы требует оптимального распределения задач среди множества рабочих станций. При неправильном распределении задач можно получить простой более мощных вычислительных станций и чрезмерную нагрузку на более слабые вычислительные станции. Как правило, это приводит к уменьшению коэффициента использования распределенной вычислительной системы. В статье [4] рассмотрены алгоритмы распределения задач в распределенной вычислительной системе без учета потерь времени при синхронизации параллельных блоков.

Постановка задачи

Для выполнения распределенного моделирования используется набор рабочих станций, коммуникационная система, а также выделенный сервер. Каждая рабочая станция выполняет моделирование набора параллельных блоков описания цифрового компонента. Коммуникационная система исполняет роль среды передачи сообщений между рабочими станциями, а выделенный сервер - синхронизации и формирования результатов моделирования.

Для проведения синхронизации между параллельными процессами используется набор сообщений. С учетом того, что моделируемая система имеет статическую структуру, набор передаваемых сообщений между параллельными процессами можно считать фиксированным. Для решения задачи оптимизации необходимо учитывать следующий набор характеристик сообщений в процессе моделирова-

ния: процесс-отправитель; процесс-получатель; длина сообщения; частота синхронизации (передачи) сообщения.

Данные характеристики сообщений определяют топологическую связность параллельных процессов.

Пусть

$$C_{ij} = F_{ij} \cdot W_{ij},$$

где C_{ij} определяет характеристики сообщений, которые используются для передачи сообщений от процесса i к процессу j ; F_{ij} определяет среднюю частоту передачи сообщения; W_{ij} определяет длину передаваемого сообщения.

Первые три характеристики сообщений являются постоянными в процессе моделирования. Частота передачи сообщения может меняться в процессе выполнения. В дальнейших вычислениях используется худший случай (максимально возможная частота).

Для решения задачи оптимизации необходимо учитывать следующий набор характеристик коммуникационной системы относительно каждой рабочей станции:

- So_i – скорость отправки данных;
- Si_i – скорость получения данных;
- Uo_i – коэффициент использования канала для отправки данных до процесса моделирования;
- Ui_i – коэффициент использования канала для получения данных до процесса моделирования.

Характеристики подключения рабочей станции i к коммуникационной системе можно описать следующим выражением: $WS_i = \{So_i, Si_i, Uo_i, Ui_i\}$.

Исходя из введенных параметров, среда моделирования может использовать следующую пропускную способность канала передачи данных $Soa_i = So_i \cdot (1.0 - Uo_i)$ для передачи сообщений и $Sia_i = Si_i \cdot (1.0 - Ui_i)$ для получения сообщений.

С целью упрощения из дальнейшего рассмотрения исключены вопросы определения физической топологии сети. Допустим, что канал передачи данных позволяет передавать сообщения между двумя произвольными рабочими станциями.

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

- статически или динамически изменяется трафик в коммуникационной системе, который порождается другими запущенными процессами моделирования или пользовательскими приложениями;
- статически или динамически изменяется количество сообщений для обеспечения синхронизации.

Таким образом, необходимо решить задачу оптимального распределения параллельных блоков описания электронного компонента со статической характеристикой топологической связности между ними C_{ij} среди рабочих станций, характеристики подключения к коммуникационной системе которых WS_i . В качестве критерия оптимальности распределения примем превышение полученного в результате распределения входного/выходного трафика. Превышение количества выходного трафика определяется выражением

$$To_i = \frac{\sum_j C_{ij}}{Soa_i} - 1.0$$

при условии, когда процессы i и j находятся на разных рабочих станциях. Превышение количества входного трафика определяется выражением

$$Ti_i = \frac{\sum_j C_{ji}}{Sia_i} - 1.0$$

при условии, когда процессы i и j находятся на разных рабочих станциях. Если в результате распределения 2 и больше параллельных блока находятся на одной рабочей станции, затраты на передачу данных между ними не учитываются.

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

$$P = \{P_i\}, P_j \cap P_k = \emptyset, j \neq k, \\ \sum_i (Ti_i + To_i) = \min, Ti_i > 0, To_i > 0$$

1. Метод оценки количества передаваемой информации между параллельными блоками при их синхронизации

Описание цифрового компонента представляет собой фиксированную структуру, вследствие чего в начале моделирования возможно определить формат и длину сообщений, которые будут использоваться в процессе синхронизации. При этом частоту, с которой будут использоваться данные сообщения при синхронизации, предугадать сложно. Для проведения расчетов необходимо оценить частоту использования сообщений в лучшем, среднестатистическом и худшем случае. Для оценки данной частоты необходимо проанализировать область декларации каждого параллельного процесса и выделить набор сигналов, который служит связью с другими параллельными процессами.

Рассмотрим пример простого описания электронного компонента на рис. 1. Модуль Top содержит в себе два компонента "Component1" и "Component2", которые соединены между собой

набором сигналов A, B, C, D, E. Сигнал A является выходным сигналом компонента "Component1" и в то же время входным сигналом для "Component2". В результате семантического анализа можно определить набор параллельных процессов в "Component2", который выступает в качестве драйвера данного сигнала, а также представляется возможным определить набор параллельных процессов в "Component1", в список чувствительности которых входит данный сигнал. Аналогичные операции можно провести со всеми остальными сигналами в описании электронного компонента.

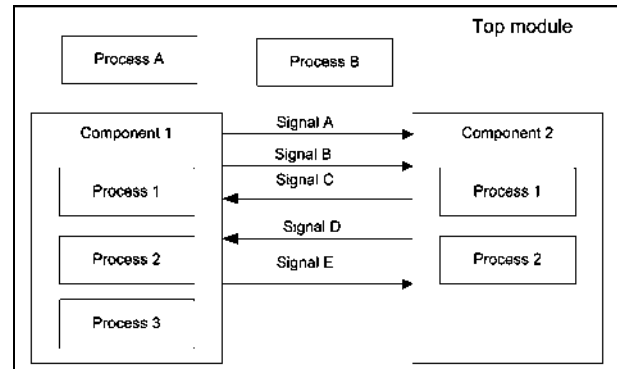


Рис. 1. Пример описания электронного компонента

В частном случае моделируемая система может представлять собой TLM модель. Следовательно, описание модели включает в себя фиксированный набор транзакций, компоненты которого выступают в роли источников или получателей транзакций. За распространение транзакций в TLM модели отвечает программно реализованный коммуникационный интерфейс.

Рассмотрим пример TLM модели на рис. 2. В данном примере компоненты 0-3 являются отправителями (источниками) транзакций. Компонент 4 представляет собой коммуникационную систему, которая оперирует определенным набором транзакций. Как правило, она хранит очередь сообщений, таблицу, в которой хранится информация о связях между компонентами системы. В общем случае коммуникационная система должна реализовывать блокирующие и не блокирующие функции по отправке/получению транзакции. За выполнение процесса синхронизации при использовании блокирующих методов отвечает компонент Response Interconnect.

При оценке количества передаваемой информации между отдельными компонентами системы в данном случае необходимо учитывать размер транзакций, количество компонентов отправителей и получателей, а также оценку частоты выполнения синхронизации. Как и в предыдущем случае, прогнозировать интенсивность обмена транзакциями не представляется возможным, поэтому при поведении вычислений используется худший сценарий.

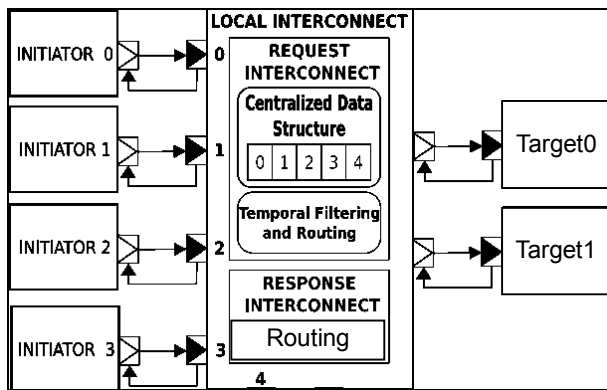


Рис. 2. Пример описания TLM модели

2. Метод оценки скорости подключения и коэффициент использования коммуникационной системы рабочей станцией

Оценка пропускной способности и коэффициентов использования подключения рабочей станции к коммуникационной системе используется для определения ее сравнительных характеристик в гетерогенной вычислительной среде.

Под входной/выходной пропускной способностью подключения к каналу связи подразумевается количество информации, которую может передать/получить рабочая станция по каналу связи за единицу времени. В общем случае данные параметры можно найти в спецификации используемого аппаратного обеспечения, но в большинстве конкретных случаев реальная пропускная способность меньше той, что указана в спецификации. Уменьшение пропускной способности возникает вследствие ряда причин, среди которых:

- разрешение доменов коллизий в среде передачи данных;
- затухание радиосигнала вследствие использования беспроводного сетевого подключения;
- помехи в коммуникационной среде;
- при передаче пакета данных необходимо придерживаться определенного протокола передачи данных, что ведет к увеличению реального количества переданной информации за счет его заголовка, контрольных сумм и т. д.;
- в общем случае протокол передачи данных имеет ограничение на размер передаваемого пакета. Если он меньше минимального размера по протоколу – недостающие биты информации будут дополнены нулями. Если же размер пакета превышает максимальный размер по протоколу, то он будет разделен на 2 и больше пакетов.

Для получения численных характеристик пропускной способности подключения к коммуникационной среде предлагается использовать бенчмарк, который будет отправлять пакеты различной длины от текущей рабочей станции к серверу моделирова-

ния и при этом замерять время, которое необходимо для передачи данных. Так как в процессе моделирования будут использоваться сообщения различной длины, предлагается для упрощения вычислений принять среднее значение пропускной способности.

Так как, помимо обеспечения процесса моделирования, отдельная рабочая станция может быть задействована в ранее запущенном процессе моделирования, или выполнять пользовательское приложение, использующее коммуникационные ресурсы, в алгоритме распределения имеются коэффициенты использования коммуникационного интерфейса. В случае ОС Windows – это счетчики производительности (другими словами, Performance Counters), которые позволяют измерить количество переданной/полученной информации по сетевому интерфейсу за определенный промежуток времени. Таким образом, можно определить следующие параметры:

$$\dot{S}_o = \frac{I_o}{T}, \dot{S}_i = \frac{I_i}{T}, U_o = \frac{\dot{S}_o}{S_o}, U_i = \frac{\dot{S}_i}{S_i}.$$

Для осуществления алгоритма распределения заданий по рабочим станциям необходимо единожды выполнить оценку производительности рабочей станции (далее эти характеристики останутся неизменными), и при каждом запуске этого алгоритма вычислить степень использования коммуникационного интерфейса. Это позволяет увеличить скорость работы алгоритма распределения заданий, так как при каждом его запуске нет необходимости запускать бенчмарк.

3. Методы распределения вычислительных заданий при распределенном моделировании

Входными данными для алгоритма распределения вычислительных заданий являются:

- оценка производительности каждой рабочей станции PR, полученная в результате проведения тестов (бенчмарк);
- характеристики динамической памяти (объем физической памяти и файла подкачки RAM и SW) – эта информация позволяет учесть текущую степень загрузки рабочей станции;
- текущее состояние нагрузки на центральный процессор;
- статическая оценка трудоемкости выполнения каждого параллельного процесса TN_i ;
- характеристики использования памяти параллельным процессом $RMIN_i$ и $RMAX_i$;
- скорость отправления и получения данных по коммуникационному интерфейсу S_o_i , S_i_i ;
- коэффициент использования канала для отправления и получения данных U_o_i , U_i_i ;
- топологическая связь между параллельными процессами i и j C_{ij} ;

Требования к алгоритмам распределения вычислительных заданий:

$$- \sum_i RMIN_{ij} \leq RAM_j - \text{объема физической па-}$$

мяти должно быть достаточно для минимальных требований памяти параллельного процесса.

$$- \sum_i RMAX_{ij} \leq RAM_j + SW_j - \text{объема динами-}$$

ческой памяти должно быть достаточно для работоспособности системы в самом худшем случае.

$$- \sum_i TN_{ij} \approx a \cdot PR_j \cdot U_j - \text{суммарная вычисли-}$$

тельная трудоемкость параллельных процессов на каждой рабочей станции пропорционально зависит от производительности центрального процессора, умноженного на коэффициент его простоя.

$$- \sum_i (Ti_i + To_i) = \min, Ti_i > 0, To_i > 0 - \text{превы-}$$

шение затрат времени на передачу и получение сообщение при синхронизации должно быть минимальным.

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

4. Критерии оценивания методов распределения

Так как вычислительные задания имеют различные требования к динамической памяти и различную вычислительную сложность, несимметричную топологию связей между логическими процессами, различную степень использования коммуникационных интерфейсов, равномерно распределить эти задания в большинстве случаев невозможно. В качестве основного критерия оценки метода распределения выбирается превышение суммарного значения коэффициентов заданий TC значения 1: $Kws = \sum_i WU_i > AVGWU_i$, где $WU = \sum_j TC_j$ – степень нагрузки рабочей станции, Kws – коэффициент использования системы, $AVG = \sum TC_i / N$. Прогнозируемые затраты на синхронизацию между параллельными процессами

$$Kc = \sum_i (Ti_i + To_i), Ti_i > 0, To_i > 0.$$

Таким образом, критерием эффективности будет минимальное значение коэффициента $K = Kc + Kws$. Также критерием оценивания эффективности метода распределения является количество итераций алгоритма.

Рассмотрим методы распределения заданий с учетом использования гетерогенной вычислительной системы.

5. Метод полного перебора

Этот алгоритм имеет самое большое время выполнения, но он гарантировано дает минимальное значение коэффициента K .

При этом количество итераций алгоритма фиксировано 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 наиболее оптимальным коэффициентом, вычисленным на предыдущих итерациях, заполнить наиболее оптимальный результат.

3. Вычислить следующую комбинацию распределения работ по рабочим станциям.

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

6. 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 $K \geq M(TC_i)$. При вычислении K алгоритм анализирует задания от последнего 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 $M(TC_i)$, 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.

7. Ускоренный метод распределения

Этот алгоритм предоставляет результат за одну итерацию, при этом значение K не является самым оптимальным, но приемлемым. Этот метод пошаго-

во заполняет рабочие станции, в первую очередь, распределяя самые сложные задачи, а потом, по мере возможности, – более простые.

Алгоритм работы этого метода следующий:

1. Отсортировать рабочие станции по мере уменьшения производительности.
2. Отсортировать задания по мере уменьшения трудоемкости.
3. Выбрать рабочую станцию. Пошагово, в порядке убывания трудоемкости выбирать задания до тех пор, пока нагрузка на рабочую станцию не превысит вычисленное ранее среднее значение нагрузки.
4. Перейти к следующей рабочей станции и повторить п. 3.
5. Если все задачи распределены по рабочим станциям – выход из метода.
6. На текущей стадии распределение любого задания на произвольную рабочую станцию приведет к превышению среднего значения нагрузки на нее. Поэтому необходимо отсортировать рабочие станции в порядке уменьшения текущего значения нагрузки.
7. Распределить оставшиеся задачи по рабочим станциям: на менее загруженную станцию – более сложную задачу.

8. Результаты сравнения методов распределения задач по рабочим станциям

Для проведения экспериментов была создана программа, которая автоматизировано генерирует набор рабочих станций с различными характеристиками производительности центрального процессора, динамической памяти и коммуникационного интерфейса, и набор задач с различными требованиями к динамической памяти, вычислительной сложности, топологии связей между ними. После генерации входных данных последовательно запускается 3 различных метода по распределению задач по рабочим станциям. Так как в экспериментах использовался метод полного перебора, для уменьшения времени работы алгоритма количество рабочих станций и задач ограничивалось. В эксперименте было проанализировано 64 комбинации. Максимальное количество рабочих станций 8, максимальное количество заданий 8. В худшем случае метод полного перебора проанализировал $8^8=16\ 777\ 216$ комбинаций. В ходе эксперимента были проанализированы 3 различных соотношения между количеством рабочих станций N и количеством задач M :

– $M < N$ – количество рабочих станций больше количества задач. В этом случае, как правило, на каждую рабочую станцию спланировано по одному заданию, некоторая часть вычислительной системы не задействована в процессе моделирования.

– $M \approx N$ – в случае, когда характеристики рабочих станций практически одинаковые, а также при однородных параллельных процессах в описании цифровой системы ожидается получить самый оптимальный случай для проведения распределенного моделирования.

– $M > N$ – наиболее часто встречаемый случай на практике.

В качестве оценки быстродействия метода предлагается рассматривать количество проанализированных комбинаций I в процессе работы алгоритма.

Это поможет различить эффективность работы методов при малой размерности M и N .

Ускоренный метод распределения задач позволяет получить результат за одну итерацию и генерирует только одну выходную комбинацию. В этом случае $I = 1$.

В методе полного перебора количество анализируемых комбинаций определяются следующим выражением: $I = M^N$. Этот метод не является эффективным.

При использовании модифицированного метода полного перебора количество итераций заранее определить невозможно, но оно определяется следующим выражением $I \leq M$.

Количество итераций при использовании этого метода приведено в табл. 1.

Таблица 1

Количество итераций при использовании ускоренного метода полного перебора

| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|----|-----|------|------|------|-------|--------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 3 | 3 | 4 | 3 | 3 | 3 | 3 |
| 3 | 1 | 3 | 13 | 4 | 4 | 11 | 14 | 13 |
| 4 | 1 | 3 | 63 | 73 | 16 | 80 | 22 | 11 |
| 5 | 1 | 4 | 36 | 292 | 513 | 412 | 56 | 4 |
| 6 | 1 | 19 | 1 | 61 | 81 | 112 | 915 | 1225 |
| 7 | 1 | 6 | 288 | 671 | 2969 | 5477 | 21637 | 2523 |
| 8 | 1 | 53 | 589 | 4159 | 3726 | 7792 | 532 | 554955 |

В качестве критерия эффективности работы метода используется коэффициент K . При использовании метода полного перебора и ускоренного метода полного перебора полученные коэффициенты эффективности имеют одинаковое значение. Полученные коэффициенты приведены в табл. 2.

При использовании ускоренного метода распределения задач полученные коэффициенты эффективности можно считать приемлемыми и близкими к оптимальным значениям. Они приведены в табл. 3.

Таблиця 2

Коефіцієнти ефективності
розподілення задач К при використанні
методів повного перебору і ускореного
метода повного перебору

| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|------|-------|------|------|------|------|------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0,1 | 0,09 | 0,28 | 0,02 | 0,02 | 0,06 | 0,13 |
| 3 | 0 | 0,09 | 0,17 | 0,09 | 0,23 | 0,42 | 0,38 | 0,32 |
| 4 | 0 | 0,59 | 0,12 | 0,52 | 0,17 | 0,28 | 0,26 | 0,17 |
| 5 | 0 | 0,1 | 0,29 | 0,24 | 0,17 | 0,1 | 0,07 | 0,03 |
| 6 | 0 | 0,12 | 0,01 | 0,24 | 0,47 | 0,46 | 0,51 | 0,34 |
| 7 | 0 | 0,05 | 0,23 | 0,15 | 0,08 | 0,19 | 0,43 | 0,77 |
| 8 | 0 | 0,06 | 0,012 | 0,23 | 0,05 | 0,2 | 0,3 | 0,55 |

Таблиця 3

Коефіцієнти ефективності
розподілення задач К при використанні
ускореного метода розподілення задач

| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|------|-------|------|------|------|------|------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0,1 | 0,09 | 0,28 | 0,02 | 0,02 | 0,06 | 0,13 |
| 3 | 0 | 0,09 | 0,17 | 0,09 | 0,23 | 0,42 | 0,38 | 0,32 |
| 4 | 0 | 0,59 | 0,22 | 0,52 | 0,48 | 0,3 | 0,26 | 0,17 |
| 5 | 0 | 0,16 | 0,29 | 0,24 | 0,3 | 0,16 | 0,19 | 0,13 |
| 6 | 0 | 0,27 | 0,01 | 0,28 | 0,27 | 0,52 | 0,2 | 0,54 |
| 7 | 0 | 0,08 | 0,24 | 0,19 | 0,38 | 0,31 | 0,37 | 0,4 |
| 8 | 0 | 0,04 | 0,016 | 0,27 | 0,38 | 0,34 | 0,1 | 0,79 |

Висновки

Для оптимізації процесу розподіленого моделювання складної цифрової системи необхідно оптимально розподілити виконання паралельних блоків по робочим станціям. Для цього слід врахувати багато різних факторів, таких як характеристики кожного паралельного процесу (ресурсомісткість, яка визначається вичислительною складністю і потребами пам'яті), а також характеристики кожної робочої станції (продуктивність центрального процесора, характеристики динамічної пам'яті, характеристики підключення до комунікаційної системи, поточна зайнятість). Розглянуті методи дозволяють розподілити завдання по вичислительним системам при різних конфігураціях робочих систем і характеристиках процесів, а також при різному співвідношенні кількості процесів і наявних робочих станцій. В результаті проведених експериментів були підтверджені дані, отримані теоретично.

МЕТОД РОЗПОДІЛУ ЗАДАЧ З ВРАХУВАННЯМ ЗАТРАТ НА СИНХРОНІЗАЦІЮ ПРИ ПАРАЛЕЛЬНОМУ МОДЕЛЮВАННІ СКЛАДНИХ ЦИФРОВИХ СИСТЕМ В ГЕТЕРОГЕННОМУ ОБЧИСЛЮВАЛЬНОМУ СЕРЕДОВИЩІ

Д.О. Гавриш, С.М. Саранча, М.О. Волк

Для оптимізації процесу моделювання необхідно врахувати затрати середовища моделювання на проведення синхронізації, яке базується на передачі повідомлень між робочими станціями. Для цього необхідно розробити алгоритм розподілу задач між робочими станціями, який враховує характеристики каналу передачі даних кожної окремої робочої станції і середні значення кількості інформації, що передається паралельними процесами для забезпечення синхронізації.

Ключові слова: процес, обчислювальна складність, ресурсомісткість.

TASK MANAGEMENT METHOD FOR PARALLEL SIMULATION OF COMPLEX DIGITAL SYSTEM IN HETEROGENEOUS COMPUTING ENVIRONMENT IN RESPECT OF COMMUNICATION SYSTEM RESTRICTIONS

D.O. Havrysh, S.M. Sarancha, M.O. Volk

To optimize simulation process need to analyze communication system usage for synchronization purposes. Synchronization is based on message exchange using local network between 2 workstations. The considered algorithms take into account the communication system characteristic and characteristics of local network adaptor of each workstation. The method is also has ability to analyze amount of information that is need to transfer for synchronization purposes.

Keywords: task, computational complexity, resource consumption.

Поступила в редакцію 22.01.2015

Список литературы

1. Meftali S. SOAP based distributed simulation environment for System-on-Chip (SoC) design / S. Meftali, A. Dzirri, L. Charest, P. Marquet, J.-L. Dekeyser In Forum on Specification and Design Languages, FDL'05, December 2005.
2. Trams M. Conservative distributed discrete event simulation with SystemC using explicit lookahead [Електронний ресурс] / М. Trams. – Режим доступа: <http://digital-force.net/publications/>, 2004.
3. Aline Mello, Isaac Maia, Alain Greiner; Parallel Simulation of SystemC TLM 2.0 Compliant MPSoC on SMP Workstations. lip6 - Laboratoire d'Informatique de Paris 6 Universit'e Pierre et Marie Curie 4, Place Jussieu - Paris - France.
4. Гавриш Д.А. Метод розподілення задач при паралельному моделюванні складних цифрових систем в гетерогенній вичислительній середі / Д.А. Гавриш, С.М. Саранча // Системи обробки інформації. – X. : ХУПС, 2014. – Вип. 2 (118). – С. 147-152.

Рецензент: д-р техн. наук, проф. О.Ф. Михаль, Харківський національний університет радіоелектроніки, Харків.