

ПАРАЛЛЕЛЬНОЕ РАЗБИЕНИЕ ГРАФОВ ДЛЯ ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ

Ладыженский Ю.В. Чикунов П.А.

Донецкий национальный технический университет, кафедра ПМиИ

E-mail: ly@cs.dgtu.donetsk.ua

Abstract

Ladyzhensky Y.V., Chikunov P.A. Parallel graph partitioning for logic simulation of digital devices. Parallel graph partition problems are considered for logic simulations. The Formulations of problems are given. Descriptions of some parallel graph partitioning algorithms are considered. A hierarchical scheme to combine results of algorithms is considered. A software system for parallel graph partitions is described.

Введение

Современные параллельные и распределенные вычислительные системы представляют новые возможности для эффективного исследования сложных научных проблем. Задачи теории графов имеют многочисленные применения в теории и приложениях. Одной из наиболее сложных является задача разбиения графов большой размерности.

Задачи разбиения больших графов схем цифровых устройств часто возникают в имитационном моделировании устройств при функциональной верификации на вентиляльном уровне и уровне регистровых передач. Появление параллельных алгоритмов разбиения графов вызвано стремлением сократить затраты времени на вычисления. Известные последовательные алгоритмы разбиения графов плохо поддаются распараллеливанию [1]. Существуют алгоритмы временной сложности $O(\log^2 n)$, ориентированные на различные модели параллельных вычислений [2,3]. Они, в основном, обеспечивают балансировку загрузки процессоров и уменьшение межпроцессорных обменов данными, но слабо учитывают необходимость максимизации параллелизма [4]. В данной работе рассмотрены некоторые алгоритмы разбиения, ориентирующиеся на выявление и использование всего потенциального параллелизма при моделировании.

Постановка задачи параллельного разбиения графов

Задача разбиения графа часто формулируется следующим образом. Дан конечный граф $G_{nm} = (V_n, E_m)$, где $V_n = \{v_1, \dots, v_n\}$ – множество вершин, а $E_m = \{e_1, \dots, e_m \subseteq V_n \times V_n\}$ – множество ребер, соединяющих вершины. Каждая вершина и ребро могут иметь свой вес. Необходимо разделить вершины из множества V на k подмножеств, имеющих примерно равную сумму весов вершин, причем сумма весов ребер, инцидентных вершинам из различных подмножеств (обрезанных ребер), должна быть минимальна. Часто k равно числу процессоров в вычислительной системе. Веса вершин соответствуют объемам вычислений, веса ребер представляют объемы передаваемых данных. Поэтому выполнение первого условия ведет к сбалансированной загрузке процессоров, а второе условие ограничивает межпроцессорный обмен [4,1].

Согласно [5] семейство \mathcal{R} подграфов графа G покрывает вершины G , если любая вершина G принадлежит, по меньшей мере, одному подграфу из \mathcal{R} . Покрытие, состоящее из непересекающихся по вершинам подграфов, называется разбиением ребер G .

Важнейшим критическим фактором параллельного моделирования являются затраты времени на межпроцессорные обмены данными, возникающие при интенсивной параллельной обработке. При обработке графов с однородными степенями вершин существует взаимосвязь между обрезанными ребрами и межпроцессорными обменами. Поэтому проблема минимизации межпроцессорного обмена, возникающая в ходе моделирования, сводится к задаче минимизации веса разрезанных ребер.

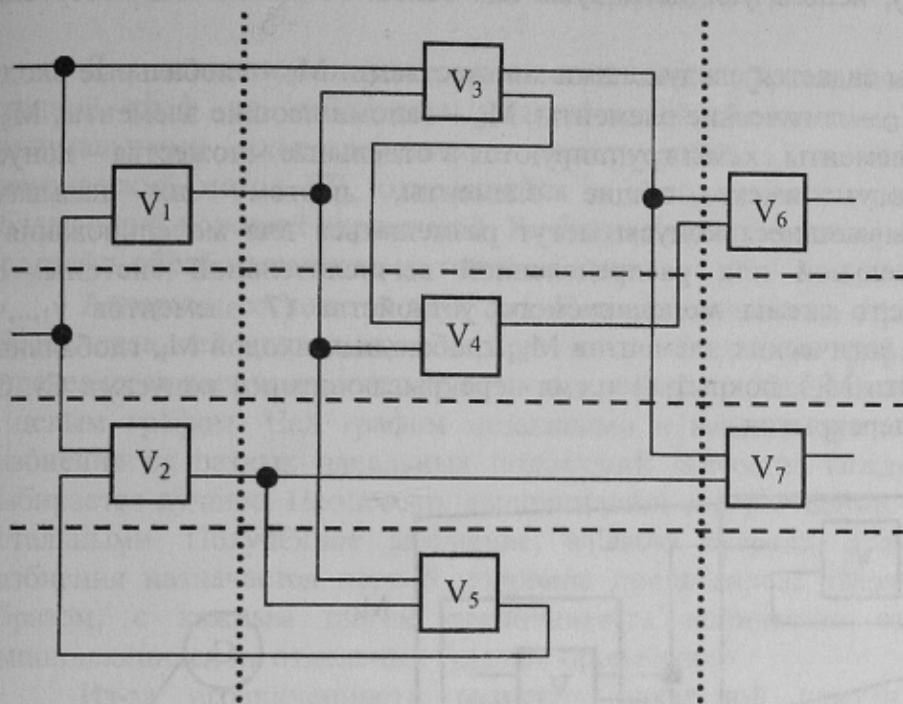


Рисунок 1. Способы разбиения схемы цифрового устройства

На рисунке 1 показаны потенциальные способы разбиения графа моделируемого цифрового устройства, в котором потоки сигналов по большинству ребер идут горизонтально слева направо от входных к выходным вершинам. Согласно конвейерной схеме (пунктирные линии на рисунке 1), процессоры работают в конвейерном режиме, причем любой процессор не может начать работу, пока его предпроцессор, расположенный левее, не закончит выполнение своих задач и не передаст управление. Наоборот, использование параллельной схемы (штриховые линии на рисунке 1) дает максимальный параллелизм с самого начала вычислений.

Максимизацию параллелизма можно понимать как распределение независимых логических элементов моделируемого устройства по наиболее возможному числу процессоров вычислительной системы. Это может быть достигнуто, если в течение всего времени моделирования сбалансирована рабочая нагрузка всех процессоров. Загрузка процессоров вычисляется как среднее число событий, обрабатываемых каждым процессором. Замечено [1], что несбалансированная загрузка приводит к увеличению числа повторных вычислений в оптимистических асинхронных алгоритмах продвижения модельного времени, удлиняя процесс моделирования.

Композиция экспертных подходов с использованием генетических алгоритмов

В работе [4] предлагается многоуровневая иерархическая схема разбиения графов, комбинирующая различные алгоритмы разбиения в рамках состязательной стратегии. Заранее неизвестно, какой из множества используемых алгоритмов даст лучшее разбиение, поэтому на каждом уровне предлагается композиция экспертных подходов, заключающаяся в сравнении результатов работы разных алгоритмов и комбинирующая генерируемые ими разбиения на одном и том же уровне иерархической схемы. В работе представлены алгоритм концентрации обратных конусов (КОК) и алгоритм минимально перекрывающихся конусных кластеров (МПКК), использующие конусы как основные блоки для построения разбиений.

Входная модель схемы задается следующими множествами: M_I – глобальные входы, M_O – глобальные выходы, M_E – логические элементы, M_L – запоминающие элементы, M_S – сигналы (проводники). Все элементы схемы группируются в отдельные множества – конусы $\{C_i\}$. Разные конусы могут иметь общие элементы, поэтому их называют перекрывающимися. Перекрывающиеся конусы могут размещаться для моделирования в разных процессорах параллельной или распределенной вычислительной системы. На рисунке 2 показаны фрагмент схемы моделируемого устройства (7 элементов v_1, \dots, v_7 , соответствующие множества логических элементов M_E , глобальных входов M_I , глобальных выходов M_O , элементов памяти M_L), покрытый тремя перекрывающимися конусами C_1, C_2, C_3 , и полученный гиперграф перекрытия.

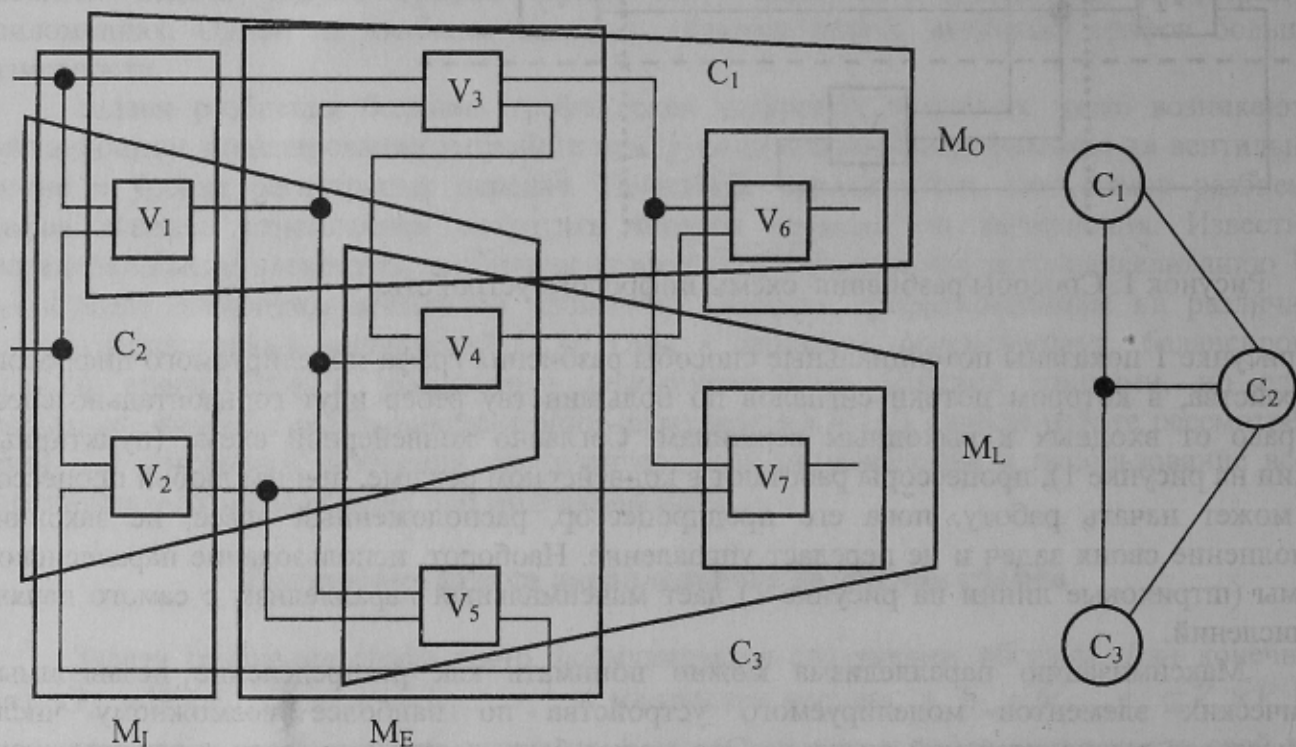


Рисунок 2. Модель схемы с конусами и гиперграф перекрытия.

На каждом этапе многоуровневой схемы разбиения применяются несколько эвристических алгоритмов, выполняющихся параллельным образом. Результаты сгенерированных разбиений сравниваются с учетом критерия качества, и лучшие из них формируют базис для алгоритмов следующего уровня. Таким образом, все экспертные знания алгоритмов в конечном разбиении смешиваются. Критерием качества данного подхода является балансировка загрузки процессоров и снижение коммуникационного обмена.

Для экспериментальных исследований использовалась модель процессорной структуры с $|M_E|=16398$ блоков. Результаты экспериментов показывают, что композиция подходов, представленных различными экземплярами КОК-алгоритма, приводит к лучшим разбиениям. Достоинствами данного метода являются возможность выбора лучшего результата из всех полученных и возможность обработки сложных микропроцессорных структур. Однако в критерий качества данного алгоритма не входит минимизация числа обрезанных ребер.

Параллельное разбиение графов на многопроцессорной системе с распределенной памятью

Для использования всего потенциального параллелизма в [6] предложено разбить входной граф на небольшие подграфы, обрабатываемые на разных процессорах с минимальными взаимодействиями между ними. Для разбиения графа используется классический метод [7], относящийся к методам перемещения групп вершин графа и обладающий хорошей эвристикой. Разбиение производится рекуррентно делением пополам подграфа, обрабатываемого на текущем этапе.

Алгоритм состоит из двух фаз. Первая основана на стратегии «разделяй и властвуй», обеспечивающей нисходящую иерархическую бисекцию графа. Количество этапов деления определяется количеством доступных процессоров. Каждый процессор изначально работает с целым графом. Над графом независимо в каждом процессоре выполняется алгоритм разбиения из разных начальных положений. Качество каждого разбиения оценивается и выбирается лучшее. Процессор, выполнивший это разбиение, обменивается результатами с остальными. Полученное разбиение, в свою очередь, делится пополам. Первая часть разбиения назначается первой половине процессоров, вторая – второй половине. Таким образом, с каждым шагом увеличивается количество отдельных частей разбиения, выполняющихся на отдельных группах процессоров.

Из-за ограниченности размеров локальной памяти не всегда представляется возможным выполнить разбиение сложных подграфов на каждом процессоре. В таком случае, каждый процессор может обрабатывать только определенную часть входной задачи.

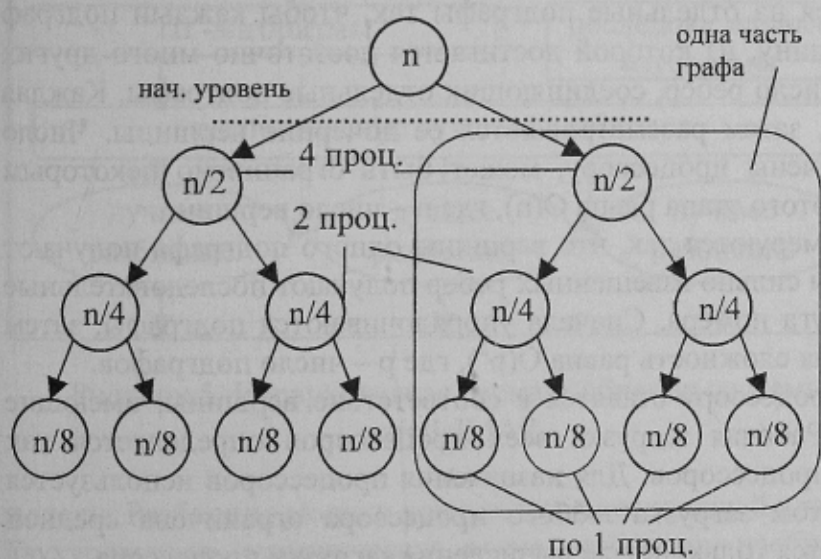


Рисунок 3. Схема деления графа на подграфы для 8 процессоров [6]

На каждом уровне i граф делится на 2^i частей. Вместо всех процессоров, обрабатывающих исходную задачу, имеется 2^i групп процессоров, обрабатывающих каждую часть входной задачи (см. рис. 3). Начальное разбиение графа на 2^i подмножеств выполняется произвольно. Это влияет на качество конечного разбиения, так как последовательные рекуррентные разбиения не учитывают связи между отдельными подмножествами. Во второй фазе рассматриваются ребра, связывающие разные подграфы. Выполняется свертывание разбиений, полученных на парах

процессоров, обрабатывающих различные подмножества графа. Таким образом, свертка и переразбиение подмножеств уменьшают общее число обрезанных ребер. Этот процесс выполняется до тех пор, пока наблюдается улучшение качества разбиения. Экспериментальное исследование проводилось на 32-процессорной машине СМ-5, представляющей собой мультимьюльтикомпьютерную систему передачи сообщений с синхронной МММД-архитектурой. Эта система состоит из компьютерных узлов, соединенных сетью передачи сообщений. Каждый узел содержит процессор, локальную память и устройства ввода/вывода. Для сравнения исследовался последовательный аналог данного метода на машине DEC-5000, процессор которой превосходит по вычислительной мощности процессорные узлы СМ-5. При логическом моделировании цифровых устройств не наблюдалось линейное увеличение скорости моделирования для 32-процессорной системы. Это объясняется существенной зависимостью процесса моделирования от начального разбиения. Достоинствами данного метода являются хорошее соотношение качество/время и возможность обработки задач большой размерности.

Разбиение схем для максимизации параллелизма вычислений при логическом моделировании

В [8] рассматривается алгоритм, ориентированный на максимизацию параллелизма при вычислениях в логическом моделировании (МПЛМ). МПЛМ-алгоритм обеспечивает параллельную обработку делением всей схемы на подграфы так, чтобы уменьшить различия в значениях локального модельного времени каждого процессора. Следовательно, уменьшается число ложных вычислений и откатов во времени. Достигается балансировка загрузки с уменьшением межпроцессорных обменов и увеличением параллелизма. МПЛМ-алгоритм является неитерационным и обладает линейной временной сложностью.

Моделируемое устройство представляется ориентированным графом, вершины соответствуют логическим вентилям, а представляют логические связи. Дополнительные подмножества содержат наборы глобальных входных и выходных вершин. Рассматриваемый алгоритм назначает вершины графа процессорам асинхронной многопроцессорной системы. Процесс выполняется в три этапа так, чтобы критерии качества разбиения могли рассматриваться отдельно, и снижалась вычислительная сложность алгоритма.

На первом этапе граф делится на отдельные подграфы так, чтобы каждый подграф содержал глобальную входную вершину, из которой достигается достаточно много других вершин, при этом минимизируется число ребер, соединяющих отдельные подграфы. Каждая вершина просматривается один раз, затем рассматриваются ее дочерние вершины. Число вершин, которые могут быть назначены процессору, может быть ограничено некоторым числом. Вычислительная сложность этого этапа равна $O(n)$, где n – число вершин.

На втором этапе вершины нумеруются так, что вершины одного подграфа получают последовательные номера, а вершины сильно взвешенных ребер получают последовательные или недалеко отстоящие друг от друга номера. Сначала упорядочиваются подграфы, затем вершины внутри них. Вычислительная сложность равна $O(p^2)$, где p – число подграфов.

На третьем этапе каждому процессору ставятся в соответствие вершины, имеющие только последовательные номера. Рабочая загрузка всех процессоров определяется как общая нагрузка, деленная на число процессоров. Для назначения процессоров используется последовательный алгоритм, при этом загрузка любого процессора ограничена средней загрузкой. Обработка задачи начинается только после вычисления загрузки процессора.

Так как последний этап отвечает только за балансировку нагрузки, а после первых двух этапов все вершины упорядочены, алгоритм показывает хорошие характеристики параллелизма и межпроцессорного обмена.

Экспериментальные исследования проводились на системе Intel Paragon XP/S. Машина содержит 64 вычислительных узла, соединенных высокопроизводительной сетью. Моделирование сложной схемы (20717 вентилях, 38 входных вентилях, 304 выходных вентилях, 1426 триггеров) при использовании 64 узлов алгоритмом МПЛМ заняло 5 секунд, а последовательными алгоритмами – 140 секунд.

Достоинствами данного подхода являются минимизация межпроцессорного обмена и небольшая временная сложность.

Система анализа и параллельного разбиения графов parallelMAP

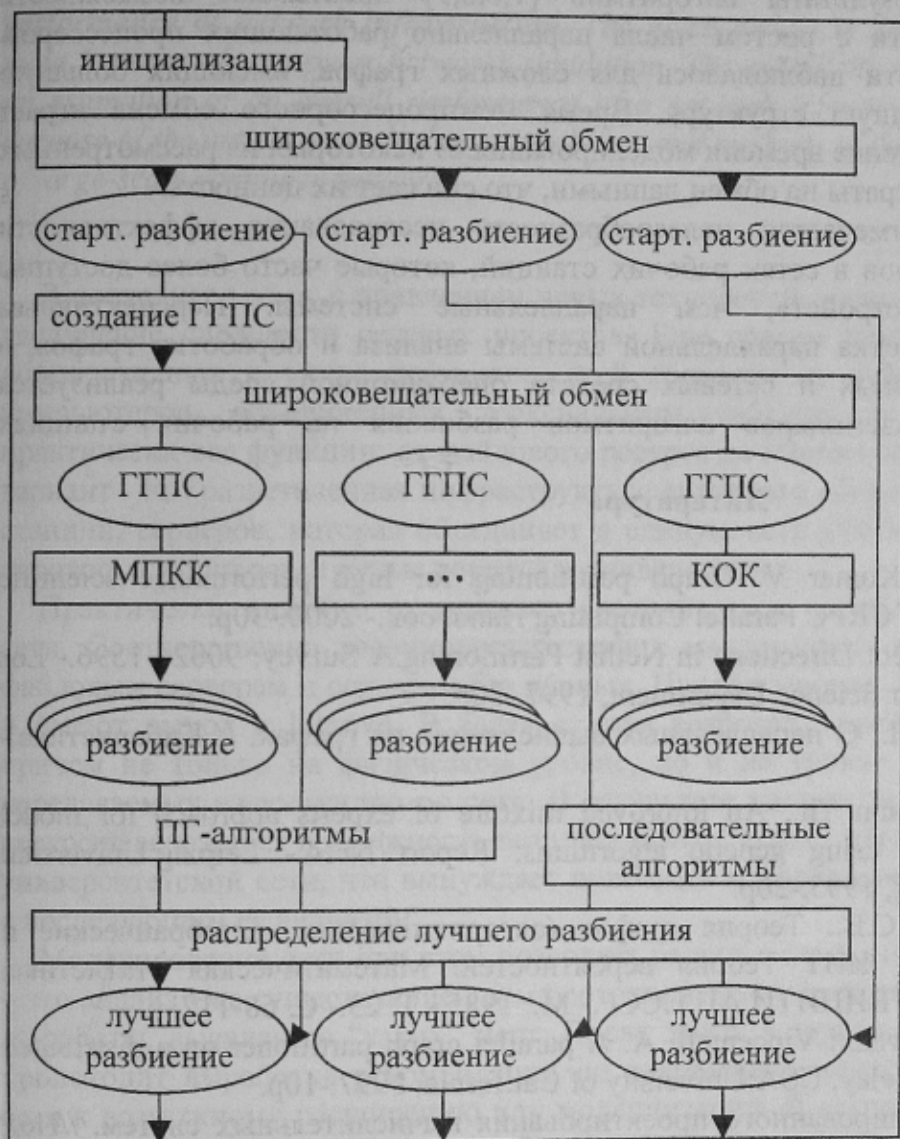


Рисунок 5. Иерархическая схема разбиения системы parallelMAP

Разбиение модели выполняется восходящей кластеризацией конусами. Применяется двухуровневая иерархическая схема разбиения, изображенная на рис. 5. Инициализация выполняется быстрым предварительным разбиением (например, последовательным алгоритмом [9]), в котором определяется базовое множество конусов; на втором уровне иерархии это множество разбивается на набор блоков с помощью множественной системы алгоритмов. На втором уровне применяются параллельные генетические (ПГ) алгоритмы

Система parallelMAP [9] является распределенной средой реального времени, используемой при параллельном моделировании микропроцессорных структур. Система выполняет анализ входной модели и предоставляет средства для разбиения. Она используется корпорацией IBM на параллельных машинах SP и кластерах рабочих станций RS/6000 для разбиения сложных процессорных моделей при системном моделировании. Система parallelMAP является логическим развитием идей, представленных в работе [4]. Система комбинирует и сравнивает алгоритмы разбиения, обеспечивает экспериментальную среду для разработки новых алгоритмов, а также генерирующую среду для входных моделей. Отдельные экземпляры системы моделирования parallelMAP совместно выполняются в слабосвязанной процессорной системе, где каждый экземпляр имитирует часть входной аппаратной

разбиения с использованием вспомогательных структур промежуточных данных, таких как гиперграфы перекрытия и связи (ГПС). Для ПГ-алгоритмов в качестве начальной популяции необходимо множество исходных разбиений. Для создания исходных разбиений применяются алгоритмы из [6,9].

Заключение

Современные системы моделирования должны обрабатывать модели микропроцессорных структур со сложностью более миллиона схемных уровней, поэтому использование параллельных вычислений является необходимым.

Экспериментальные результаты алгоритмов [1,6,8,9] показывают возможности увеличения производительности с ростом числа параллельно работающих процессоров. Увеличение производительности наблюдалось для сложных графов, имеющих большую размерность и сильно связанную структуру. Время межпроцессорного обмена играет существенную роль в общей оценке времени моделирования. В некоторых из рассмотренных алгоритмов не учитываются затраты на обмен данными, что снижает их ценность.

В работах [1,2,10] отмечается целесообразность исследования эффективности решения задач разбиения графов в сетях рабочих станций, которые часто более доступны разработчикам цифровых устройств, чем параллельные системы. Перспективным направлением является разработка параллельной системы анализа и обработки графов, в которой за счет многопоточковых и сетевых средств операционной среды реализуется параллельное выполнение экземпляров алгоритмов разбиения на рабочих станциях локальной сети.

Литература

1. Schloegel K., Karypis G., Kumar V. Graph partitioning for high performance scientific simulations. //Early Draft of CRPC Parallel Computing Handbook.- 2000.-30p.
2. Alpert C.J., Kahng A.B. Recent Directions in Netlist Partitioning.A Survey: 90024-1596.- Los Angeles, CA,UCLA Computer Science Department, 1997.-80p.
3. Иванов Е.А., Шевченко В.П.. О параллельных вычислениях на графах. // Кибернетика.- 1984.- №3.-С. 89-94.
4. Hering K., Haupt R., Villmann Th.. An improved mixture of experts approach for model partitioning in VLSI-design using genetic algorithms: Report Nr14.- Leipzig:Universitat Leipzig, Institut fur Informatik, 1995.-20p.
5. Козырев В.П., Юшманов С.В.. Теория графов (алгоритмические, алгебраические и метрические проблемы). // ИНТ Теория вероятностей. Математическая статистика. Теоретическая кибернетика. ВИНТИ АН СССР.- М.: 1985.- Т. 23.- С. 68-117.
6. Buch P., Sanghavi J., Sangiovanni-Vincentelli A. A parallel graph partitioner on a distributed memory multiprocessor.- Berkeley, USA:University of California, 1997.-10p.
7. Теория и методы автоматизированного проектирования вычислительных систем. //Под ред. Брейера М.- М.: Мир, 1977.-283с.
8. Kim H.K., Jean J. Concurrency preserving partitioning (CPP) for parallel logic simulation. // Proc. of the 10th Workshop on Parallel and Distributed Simulation.-1996.-P. 98-105.
9. Hering K., Hennings H., Haupt R.. DRIVE: A distributed environment supporting combination of sequential and parallel modules. //Proceedings of the IASTED International Conference.- Cambridge, Massachusetts, 1999.-10p.
10. Чикунов П.А. Применение параллельных алгоритмов разбиения графов в схемотехническом и научном имитационном моделировании. // Материалы научно-практической конференции «Наука-жизнь-производство-2001».- Красноармейск: КФ ДонНТУ, 2001.-С.35.