

Рис. 3. Принципиальная схема устройства линейной интерполяции в файле верхнего уровня, созданная в пакете Quartus II

В результате моделирования работы устройства линейной интерполяции в системе автоматического проектирования Quartus II тактовая частота составляет 240 МГц. Один модуль сумматора занимает в микросхеме семейства Stratix EP1S10F484C5 фирмы Altera менее 2%.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Байков В.Д. Сомолов В.Б. Аппаратурная реализация элементарных функций в ЦВМ. – Л.: Изд-во Ленингр. ун-та, 1975. – 96 с.
2. Бандман О.А. Специализированные процессоры для высокопроизводительной обработки данных. – Новосибирск: Наука, 1988. – 204 с.
3. Шекопляс Б.В. Микропроцессорные структуры. Инженерные решения. Справочник / 2-е изд. перераб. и доп. – М.: Радио и связь, 1990. – 512 с.
4. Партала О.Н. Цифровая электроника / Издание 2-е, дополненное – СПб: Наука и Техника, 2001. – 224 с.

УДК 681.518

В.А. Литвиненко, С.А. Ховансков, О.Р. Норкин

АЛГОРИТМ УСКОРЕНИЯ ВЫПОЛНЕНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В КОМПЬЮТЕРНОЙ СЕТИ*

Существует класс задач требующих большого объема вычислений и имеющих жесткое ограничение по времени выполнения. Традиционно такие задачи решаются путем организации распределенных вычислений на нескольких центрах обработки данных вычислительной системы [1, 2].

В качестве вычислительной системы используется либо мультипроцессорная вычислительная машина, либо многомашинная вычислительная система, либо обычная компьютерная сеть, обладающая достаточным или избыточным количеством центров обработки данных (локальная, глобальная сети). При этом вычислительная

* Работа выполнена при поддержке РФФИ (грант № 07-01-00174).

система при организации решения задачи представляется в виде графа вычислительных ресурсов T , имеющего N центров обработки данных.

Для организации распределенных вычислений все множество операций, выполняемых при решении задачи, предварительно делят по функциональным признакам на вычислительные блоки, образуя граф B с M количеством блоков.

Существует множество специальных языков программирования, которые позволяют программистам организовывать распределенные вычисления в сети для решения объемных задач. Недостатком такой организации являются значительные временные затраты. Это накладывает определенные ограничения на быстродействие решения задачи и предъявляет требования к используемым вычислительным ресурсам. В основном таким путем организуется решение задач, имеющих между вычислительными блоками низкую степень связности по данным.

Наиболее доступной вычислительной системой является сеть, в которой при организации распределенных вычислений каждому вычислительному блоку задачи ставится в соответствие компьютер с учетом наличия физических связей в сети, способных реализовать потоки передаваемой информации между вычислительными блоками.

Одним из путей организации и выполнения распределенных вычислений является использование метода принятия коллективного решения мультиагентной системой. Агентом является программный код, выполняемый в фоновом режиме на локальном компьютере. Каждый компьютер, имеющий одного из агентов, работает независимо от других аналогичных компьютеров. Для выполнения распределенных вычислений агенты мультиагентной системы обмениваются между собой данными.

Основной алгоритм программы-агента реализует следующие задачи:

- ◆ получение информации от компьютеров-источников;
- ◆ выполнение на компьютере требуемых операций над полученной информацией;
- ◆ передача результата компьютеру-получателю.

Основная проблема, возникающая при такой организации распределенных вычислений, заключается в том, что компьютеры в сети обладают разными вычислительными мощностями. Это приводит к тому, что при организации распределенных вычислений возможны множество различных вариантов назначения вычислительных блоков задачи компьютерам сети, которые оказывают различное влияние на время выполнения задачи. Это пытаются учитывать, предварительно выбирая из множества наиболее оптимальный вариант, обеспечивающий требуемое время решения задачи.

Существует несколько путей для решения проблемы выбора нужного варианта. Задача выбора может быть упрощена путем подбора для организации распределенных вычислений компьютеров, имеющих одинаковую вычислительную мощность. Недостатком этого пути является ограничение круга решаемых задач только задачами, обрабатываемыми большой массив данных и состоящих из одного небольшого вычислительного блока.

Другой путь заключается в предварительном поиске оптимального варианта назначения из множества имеющихся вариантов [3]. Основным недостатком этого пути является то, что выбор оптимального варианта назначения вычислительных блоков задачи компьютерам сети требует дополнительных временных затрат, которые могут значительно превышать время решения самой задачи.

В [4] был предложен алгоритм выбора оптимального варианта обрабатываемый агентами. Реализация распределенных вычислений начинается с наложения графа B на граф T . Для этой реализации должно быть выполнено условие $N > M$. В

каждую вершину t_j записывается и активируется программный код агента a_j . Каждый агент содержит вычислительный блок b_i информацию о направлении потоков данных между соединенных с ним вершинами графа B . Наложение выполняется случайным образом, как только будут обнаружены в количестве M центры обработки информации.

После запуска вычислительных процессов данные поступают входным агентам частями. По мере их обработки данные передаются в направлении выходных агентов в соответствии со связями графа B . Обработка агентом одной части данных является шагом.

После каждого шага агенты передают соседним агентам обработанные данные и обмениваются с ними служебной информацией. Получая служебную информацию, агенты, на фоне решения задачи, выполняют поиск методом парных перестановок оптимального варианта наложения вершин графа B на вершины графа T .

Алгоритм поиска основан на том, что среди всех маршрутов передачи данных от входных агентов к конечным всегда есть критический маршрут, по которому передача данных занимает наибольшее время. Агенты, делая парные перестановки, перебирают варианты наложения и находят вариант, уменьшающий время передачи данных по критическому маршруту, а, в конечном итоге, уменьшающий общее время обработки данных.

Во время решения задачи под управлением агентов выполняется оптимизация вычислительной системы для ускорения процесса решения. Каждому вычислительному блоку графа задачи наиболее оптимально ставится в соответствие вычислительный узел графа вычислительной системы.

Алгоритм поиска продолжает обрабатываться агентами и после того, как оптимальный вариант наложения вершин графа B на вершины графа T будет найден. Но при этом время решения задачи на данной конфигурации вычислительной системы уже невозможно уменьшать из-за постоянства имеющихся вычислительных ресурсов графа T .

Для примера, на рис.1, а изображен исходный граф B с количеством блоков $M=11$. Для организации распределенных вычислений выбрана локальная сеть T , имеющая $N=14$, при этом выполняется условие $N > M$. Граф B был наложен случайным образом на граф T и после нескольких шагов работы вычислительной системы был найден оптимальный вариант наложения (рис. 1, б) графа B на граф T .

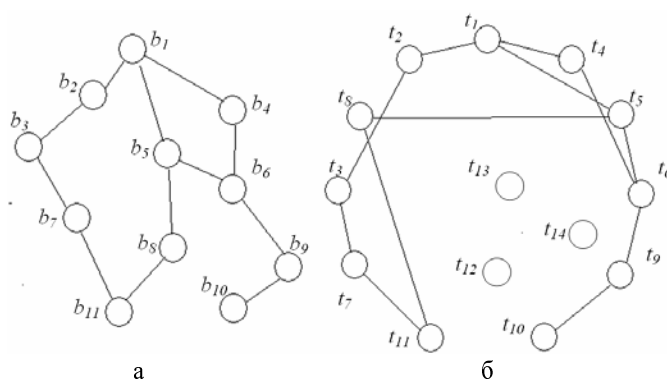


Рис. 1. Исходный граф B (а) и оптимальный вариант наложения графов B и T (б)

С целью дальнейшего сокращения общего времени обработки данных разработан алгоритм работы агентов, позволяющий продолжить уменьшать общее время решения задачи. Он основан на привлечении дополнительных вычислительных устройств в процесс решения некоторых вычислительных блоков графа B задачи. Поскольку вычислительная система представляет собой совокупность компьютеров объединенных в сеть или являющихся частью компьютерной сети. Это дает возможность вовлекать (вербовать) в процесс решения задачи «свободных» компьютеров, т.е. не вошедших в граф вычислительной структуры, что и позволяет продолжить сокращение времени решения задачи.

Свободные компьютеры могут быть завербованы программой-агентом находящейся на одном из компьютеров графа T . Вербовать может любой агент. Вербовка заключается в том, что программа-агент передает свою копию на свободный компьютер и активизирует ее в случае, если на нем нет программы-агента.

После нахождения оптимального варианта назначения общее время прохождения данных по вершинам графа задачи по-прежнему определяется критическим маршрутом.

Каждый агент определяет свое местонахождение относительно критического маршрута. Агент, принадлежащий критическому маршруту и выполняющий b_i за определенное время, вербует вычислительный узел $w \in (N - M)$. Для этого он передает служебную информацию соседним с ним компьютерам и вычислительный модуль агенту завербованного компьютера.

Появляется новая вершина в графах задачи и вычислительной системы. Таким образом, может быть завербован случайным образом любой компьютер сети.

После включения завербованного компьютера в вычислительную систему в постоянно повторяющемся процессе оптимизации варианта назначения будут участвовать и вычислительные ресурсы узла w .

После нескольких шагов будет найдено новое оптимальное соответствие между вершинами графов задачи и вычислительной системы, имеющее меньшее общее время прохождения данных по вершинам графа задачи.

В приведенном на рисунке (см. рис. 1) варианте время прохождения данных через вычислительную систему при оптимальном варианте определяется критическим маршрутом $r_{крит} = (t_1, t_5, t_6, t_9, t_{10})$.

После выполнения агентом узла t_9 алгоритма вербовки свободного узла был завербован узел t_{14} . После обмена между узлами t_9 и t_{14} информацией граф T изменился (рис. 2). Благодаря этому, время прохождения данных через вычислительную систему уменьшилось, поскольку оно определяется новым критическим маршрутом $r_{крит} = (t_1, t_5, t_8, t_{11})$.

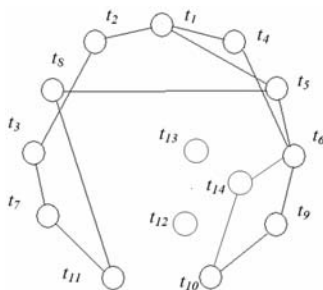


Рис. 2. Вариант наложения измененного графа B и графа T

Алгоритм агента:

1. Получение входных данных.
2. Формирование адреса узла сети.
3. Если этот адрес использовался, то переход к п.1.
4. Передача по сформированному адресу вербуемого узла служебной информации.
5. Если пришел ответ от вербуемого узла, то переход к п.1.
6. Передача служебной информации соседним узлам графа B о завербованном узле.
7. Конец работы алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лорин Г. Распределенные вычислительные системы. – М.: Радио и связь, 1984. – 296 с.
2. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.
3. Ховансков С.А., Мельник Э.В, Блушвили И.В. Метод организации распределенных вычислений в управляющих системах // Мехатроника, автоматизация, управление. «Новые технологии», №4. 2003.
4. Ховансков С.А., Литвиненко В.А. Организация распределенных вычислений на основе мультиагентного подхода // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТРТУ, 2007, №1(73). – С. 246-250.

УДК 681.3

О.И. Овчаренко

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ БЫСТРЫХ ПРЯМЫХ МЕТОДОВ РЕШЕНИЯ СЕТОЧНЫХ ЭЛЛИПТИЧЕСКИХ УРАВНЕНИЙ

Прогресс в численном моделировании важных прикладных задач теории поля, гидродинамики стал возможен благодаря появлению быстрых прямых методов решения сеточных уравнений [1]. К числу этих методов следует, в первую очередь, отнести циклическую редукцию (cyclic reduction (CR)), Фурье-алгоритм (Fourier-algorithm (FA)) и типа FACR(L), предназначенных для решения сеточных эллиптических уравнений, с оценками количества арифметических операций $O(N \log_2 N)$ или даже $O(N \log_2(\log_2 N))$.

Основными этапами быстрых прямых методов являются решение трехдиагональных систем уравнений и выполнение Фурье-преобразований, которые можно реализовать различными алгоритмами. Для каждого алгоритма, в свою очередь, можно распределить исходную информацию по процессорам несколькими способами. В связи с этим основные этапы быстрых прямых методов могут быть реализованы как последовательно (в одном процессоре), так и параллельно (на всем решающем поле МВС).

Поэтому актуальной становится задача не только построения параллельных алгоритмов быстрых прямых методов, но и поиск наиболее эффективных в зависимости от размерности задачи, количества процессоров, значения параметра, характеризующего быстродействие каналов связи при заданной топологии многопроцессорной вычислительной системы (МВС).