

ДИНАМИЧЕСКОЕ ДЕЦЕНТРАЛИЗОВАННОЕ УПРАВЛЕНИЕ ПАРАЛЛЕЛЬНЫМ ВЫЧИСЛИТЕЛЬНЫМ ПРОЦЕССОМ

© Г.И. Малашонок, © А.М. Добычин

Ключевые слова: управление параллельным вычислительным процессом, децентрализованное управление параллельными вычислениями.

Аннотация

Обсуждается схема децентрализованного управления параллельным вычислительным процессом для рекурсивных древовидных алгоритмов. Оцениваются преимущества и недостатки такой схемы по сравнению со схемами управления, которые включают диспетчер.

Задача управления параллельным вычислительным процессом – основная задача программирования с использованием параллельных вычислительных кластеров.

Для задач компьютерной алгебры известна параллельная LLP схема [1]. Это схема централизованного управления. Она обеспечивает распределение поддеревьев по процессорам, отдавая предпочтение тем поддеревьям, у которых корневая вершина находится на самом низком уровне (Low Level Parallelization). Для этого используется диспетчер – отдельный процессор, играющий роль координатора.

В случае больших кластерных систем динамическое централизованное управление распределением процессоров кластера по подзадачам древовидных алгоритмов может терять эффективность из-за большого числа обращений к диспетчеру. В этом случае преимущество будет у такой схемы распределения, в которой не требуется единый координирующий диспетчер. Однако при этом потеряется возможность отбирать низкоуровневые деревья.

Будем использовать основные определения и понятия из работы [1].

Имеется множество процессоров и темпорированное дерево рекурсивного алгоритма. У каждой вершины дерева имеется одно входное ребро и некоторое количество выходных ребер, которым приписаны натуральные веса. Вес обозначает очередность вычислений поддеревьев, которые начинаются с этих ребер. Совпадение весов обозначает параллельность вычислений.

Задача управления вычислительным процессом состоит в «назначении» поддеревьев определенным процессорам кластера. Это должно происходить в процессе вычислений, и не должен использоваться единый координирующий центр – диспетчер.

Сформулируем основные черты предлагаемого варианта схемы децентрализованного управления.

При каждой вершине дерева алгоритма создается изначально пустой список дополнительных процессоров (ДП). Пусть ребра некоторой вершины V имеют h различных весов. При этом ребер веса i будет $v(i)$. При вычислении поддеревьев, которые начинаются с ребер веса i , создадим регистр из $v(i)$ флагов ребер и одного флага вершины V .

Каждый флаг ребра будет принимать значения 0,1,2, которые имеют следующий смысл. Флаг ребра 2 означает, что закончено вычисление поддерева, начинающегося с этого ребра. Флаг ребра 1 означает, что начаты вычисления всех листовых вершин поддерева, начинающегося с этого ребра, но вычисление всего поддерева не закончено. Флаг ребра имеет значение 0 в остальных случаях.

Флаг вершины V также принимает значения 0,1,2. При этом значение 2 принимается флагом вершины V тогда, когда $i=h$, все флаги ребер веса h равны 2 и вычислен подблок h . Если значение флага вершины V не равно 2 и нет 0-флагов ребер веса i , то значение флага вершины V принимает значение 1. Значение 0 флаг вершины V принимает в остальных случаях. Таким образом, значение флага вершины V , равное 0, означает, что найдется листовая вершина в поддереве с корневой вершиной V , вычисление которой еще не начиналось.

При переходе к вычислению ребер очередного веса в вершине V флаги обнуляются.

При изменении состояния одного из флагов ребер пересчитывается флаг вершины V . Для этого используется счетчик $N0$ числа флагов 0 и счетчик $Nh2$ числа флагов 2. Если закончено вычисление всех поддеревьев старшего веса h вершины V , то флаг вершины V принимает значение 2. Если счетчик $N0$ равен нулю, то флаг вершины V принимает значение 1, а иначе он равен 0.

Все вычисления закончены, если в корневой вершине дерева флаг принял значение 2.

Развитие вычислительного процесса происходит следующим образом.

Главный процессор получает корневую вершину дерева, входные данные и список дополнительных процессоров ДП, содержащий все процессоры кластера: $1, 2, \dots, n$.

Пусть поддерево, полученное процессором s , ($s=1..n$), начинается с вершины V с флагом 0 и списком ДП, содержащим g процессоров, включая s . Пусть $1..h$ - веса ребер, исходящих из вершины V . Производится параллельное вычисление поддеревьев, начинающихся со всех ребер, исходящих из вершины V и имеющих вес i ($i=1..h$). Сначала вычисляются все поддеревья с весом 1, затем - с весом 2, и так далее - до веса h . При этом каждый раз вычисляются только те поддеревья, у которых флаги равны нулю.

Если число процессоров g в списке ДП не превосходит числа ребер $u(i)$ веса i , то каждому процессору z из списка ДП передается по одной вершине и список ДП, состоящий из одного процессора z . Если $g > u(i)$, то список ДП делится на $u(i)$ частей так, чтобы s был первым процессором в первой части. Первому процессору в j -той ($j=1..u(i)$) части списка ДП передается j -тая вершина и вся j -тая часть списка ДП, если передаваемая вершина не листовая. Если передаваемая вершина является листовой, то передается только сам первый процессор.

Установка флагов ребер и перераспределение в списке ДП происходит следующим образом.

Когда процессор s получил листовую вершину, которая связана входным ребром R с вершиной V , то флаг ребра R устанавливается равным 1. Когда он заканчивает вычисление этой вершины, тогда флаг ребра R устанавливается равным 2 и номер процессора s дописывается в список ДП вершины V . Когда изменяется флаг вершины, то его новое значение присваивается флагу ребра, входящему в эту вершину. Если вершина V не корневая вершина дерева алгоритма, флаг вершины V не равен 0 и ее список ДП не пуст, то ее список ДП дописывается в список ДП вершины, с которой она связана входным ребром.

Предполагается реализовать такую схему организации вычислительного процесса, как альтернативную для схемы LLP. Обе схемы предполагается сравнить на реальных вычислительных задачах на кластере ТГУ им.ГРДержавина и на кластере МСЦ АН.

Работа выполнена при частичной поддержке грантов РФФИ (проект 08-07-97507).

Список литературы

1. Малашинок Г.И., Валеев Ю.Д. Организация параллельных вычислений в рекурсивных символично-численных алгоритмах. Труды конференции ПаВТ'2008 (Санкт-Петербург). Челябинск: Изд-во ЮУрГУ, 2008. С. 153-165.

Поступила в редакцию 17 ноября 2008 г.

ПРИЛОЖЕНИЕ БАЗИСОВ ГРЕБНЕРА В РОБОТОТЕХНИКЕ

© А.Г. Поздникин

Ключевые слова: стандартные базисы полиномиальных идеалов, базисы Гребнера, робототехника.

Аннотация

Статья посвящена обзору современных работ, в которых рассматривается проблема управления роботами с использованием стандартных базисов полиномиальных идеалов или базисов Гребнера.

Управление такими сложными механизмами, как многозвенные манипуляторы, может оказаться довольно трудоемким. Даже при выполнении простейшей операции - перемещение вершины многозвенного манипулятора из точки A в точку B - для расчета управляющей информации требуется выполнить большой объем вычислений [1]. В настоящее время для анализа сложных задач механики манипуляторов используется векторный метод, метод матриц, метод винтов [2]. Состав, параметры и классификация роботов по различным признакам описываются в работах Юревича Е. И. [3] и [4].