Параллельные символьные вычисления в Тамбовском государственном университете

Г.И.Малашонок, О.Н.Переславцева, Д.С.Ивашов

Символьные вычисления с большим объемом данных невозможны без суперкомпьютеров. Сегодня стоит задача создания системы символьных вычислений с использованием суперкомпьютеров. Это приведет к качественному скачку вычислительных технологий во многих приложениях.

Для иллюстрации мы рассматриваем два примера:

– вычисление характеристического полинома линейного оператора и

– факторизацию полиномов многих переменных.

Сложность символьных вычислений существенно выше, чем у численных. Поэтому с появлением суперкомпьютеров символьные вычисления переживают новый этап развития.

Пример А.

Решение системы n линейных уравнений с n неизвестными требует ~ n^3 операций.

Если все числа в системе являются целыми и для их записи нужно m машинных слов, то для точного решения такой системы нужно ~ $n^5 m^2$ операций.

Если решать эту задачу, переходя к многим конечным полям, то потребуется ~ $n^4 m^2$ операций,

p-адический метод позволяет решить задачу за ~ $n^3 m^2$ операций.

Пример B.

Теперь рассмотрим символьную матрицу.
Пусть, например, требуется решить систему линейных уравнений, где каждый коэффициент — полином степени ***d*** с целыми коэффициентами.

Прямые вычисления потребуют ~ $n^3(nd)^2(nm)^2$ операций. Мы не выписываем логарифмические множители.

Если решать эту задачу, переходя к многим конечным полям, то потребуется ~ $n^5 dm$ операций.

Даже если все полиномы первой степени и коэффициенты занимают одно машинное слово, то точное решение такой системы в n^2 раз сложнее, чем приближенное решение числовой системы.

И для решения системы размера $1000x1000$ требуется в миллион раз больше операций.

Без суперкомпьютера не обойтись.

Задача, которая стоит сегодня, – это ***создание системы символьных вычислений для проведения вычислений на суперкомпьютере***.

Создание такой системы станет качественным скачком современных вычислительных технологий. Такая система найдет применение при решении задач во многих разделах физики, химии, биологии, в задачах математического моделирования в электронике, радиотехнике, управлении и других областях. Появится возможность решать трудные задачи с большим числом переменных за реально допустимое время. Это требует создания новых подходов и алгоритмов.

Необходимо разработать:

- быстрые ***методы вычислений*** с полиномами многих переменных, композициями трансцендентных функций, функциональными матрицами и др. объектами.
- специальные ***структуры данных*** для представления алгебраических объектов в программной среде, которые предназначены для эффективного распараллеливания вычислительного процесса.
- динамическую децентрализованную ***схему управления*** параллельным вычислительным процессом, пригодную для разреженных и неоднородных данных.

В качестве иллюстрации мы приводим два примера.

Первый – это задача вычисления характеристического полинома линейного оператора над модулем целочисленных многочленов многих переменных.

Второй – это задача факторизации полиномов многих переменных с целыми коэффициентами.

Для этих задач были разработаны параллельные программы и проведены эксперименты на кластере MVS100k МСЦ РАН.

Вычисление характеристических полиномов линейных операторов

Характеристическим полиномом матрицы A линейного оператора называется полином $F(x)=det(A-xE)$, где E – единичная матрица.

Непосредственное вычисление коэффициентов характеристического полинома для целочисленной или полиномиальной матрицы требует большого числа операций. Поэтому применяется алгоритм, основанный на методе гомоморфных образов.

Общая схема метода гомоморфных образов следующая:

1) выберем необходимое количество числовых и полиномиальных модулей.
2) найдем образы исходной матрицы полиномов в каждом факторкольце;
3) решим задачу с помощью алгоритма Данилевского в каждом факторкольце,
4) восстановим решение в исходной области по китайской теореме об остатках.

Алгоритм естественно распараллеливается по отдельным модулям, т.е. вычисление характеристического полинома по каждому простому модулю происходит независимо, затем восстанавливаются искомые коэффициенты характеристического полинома. Граф алгоритма – двухуровневое дерево: корень — листья.

Все вычисления происходят на листовых вершинах. За каждой листовой вершиной закрепляется:
(а) некоторое число простых модулей и
(б) множество коэффициентов характеристического полинома, которые будет восстанавливаться на этой вершине.

Перед восстановлением происходит обмен данными между листовыми вершинами. Восстановленные коэффициенты передаются сразу с листовых вершин в корневую вершину.

Оценка времени вычислений.

Пусть A – полиномиальная матрица размера nxn, элементы которой являются полиномами от t переменных, у этих полиномов

a – наибольший по абсолютной величине числовой коэффициент,

s – наибольшее количество мономов,

deg_i – наибольшая степень по переменной x_i, ($i=1,...,$t),

Пусть k – количество процессоров.

Время, необходимое для вычисления характеристического полинома без учета времени пересылки данных между процессорами, равно

$$T_{n,k} = \Theta\left(\frac{1}{k} n^{t+4} (\log_2 n + \log_2 a + \log_2 s) \prod_{i=1}^{t} deg_i\right).$$

А время, необходимое для пересылки данных между процессорами в n^2 раз меньше, чем $T_{n,k}$.

Это позволяет прогнозировать время вычисления характеристического полинома.

Используя результаты экспериментов, проведенных на кластере MVS100k МСЦ РАН, построены графики зависимости прогнозируемого времени вычислений характеристических полиномов числовых и полиномиальных матриц от размера матриц.

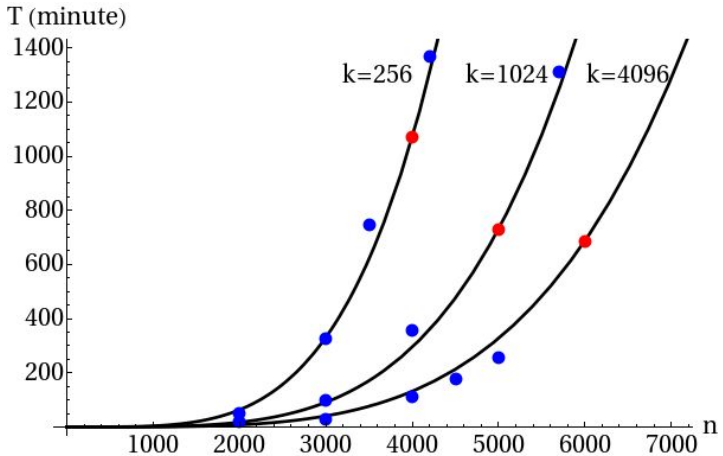

Рис. 1. Время вычисления характеристического полинома для числовой матрицы размера *nxn,* элементами которой являются 7-ми разрядные двоичные числа, при использовании *k* процессоров.

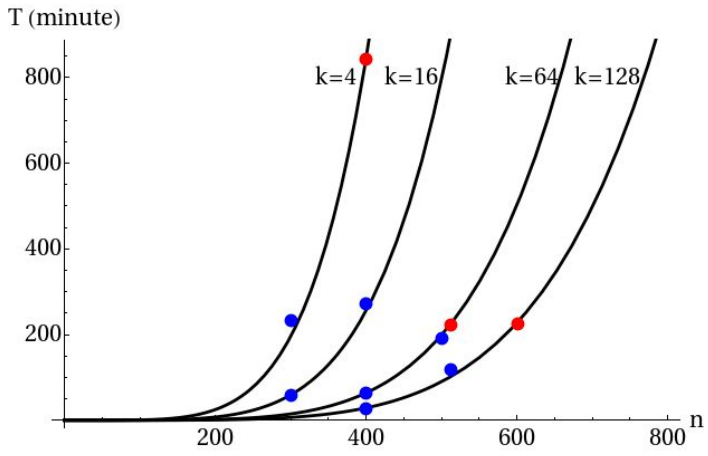

Рис. 2. Время вычисления характеристического полинома для полиномиальной матрицы *nxn*, элементами которой являются линейные полиномы одной переменной с 7-ми разрядными двоичными коэффициентами, при использовании *k* процессоров.

Алгоритм разложения многочленов многих переменных на множители

Пусть $F(x_1,\dots,x_n)\in\mathbb{Z}(x_1,\dots,x_n)$ многочлен от n переменных $x_1,\dots,x_n$ с целочисленными коэффициентами. Схема факторизации состоит из следующих этапов.

Этап 1. Вычисление сомножителей $f_1,\dots,f_{2^n-1}$, имеющих *различные наборы переменных*: $F(x_1,\dots,x_n)=f_1(x_1)f_2(x_2)\dots f_{2^n-1}(x_1,\dots,x_n).$

Этап 2. Вычисление *кратных сомножителей* h_{ij} многочленов f_i : $f_i=\prod_{j=1}^{r_i}h_{ij}^{s_{ij}}$, где $i=1,\dots,2^n-1$, s_{ij} - кратность j-го сомножителя и $s_{ij}\neq s_{ik}$ при $j\neq k$.

Этап 3. Вычисление *неприводимых сомножителей* g_{ijk} : $h_{ij}=\prod_{k=1}^{r_k}g_{ijk}$.
В результате многочлен $F(x_1,\dots,x_n)$ можно записать в виде $F(x_1,\dots,x_n)=\prod_{i=1}^{2^n-1}f_i=\prod_{i=1}^{2^n-1}\prod_{j=1}^{r_i}h_{ij}^{s_{ij}}=\prod_{i=1}^{2^n-1}\prod_{j=1}^{r_i}\prod_{k=1}^{r_k}g_{ijk}^{s_{ij}}$.

Схема алгоритма вычисления сомножителей, имеющих различные наборы переменных:

У многочлена F от *n* переменных может быть не более чем 2^n-1 сомножителей f_i, которые имеют различные наборы переменных: $F(x_1,\dots,x_n)=\prod_{i=1}^{2^n-1} f_i$.

Представим многочлен F в виде многочлена от переменной x_1 с коэффициентами из $\mathbb{Z}(x_2,\dots,x_n)$: $F(x_1,\dots,x_n)=\prod g_i(x_2,\dots,x_n)x_1^i$, где $k=deg_{x_1}F(x_1,\dots,x_n)$. Вычислим НОД полиномиальных коэффициентов g_i: $f_1(x_2,\dots,x_n)=GCD(g_0(x_2,\dots,x_n),\dots,g_k(x_2,\dots,x_n))$. Многочлен $f_1(x_2,\dots,x_n)$ не содержит переменную x_1 и является сомножителем многочлена F. После деления исходного многочлена F на $f_1(x_2,\dots,x_n)$, получим второй сомножитель $f_2(x_1,\dots,x_n)$ многочлена F.

Для каждого из полученных многочленов будем продолжать этот процесс для переменных $x_2,x_3,\dots,x_n$. В результате можем получить не более, чем 2^n-1 сомножителей.

Уменьшение числа операций и получение эффективной схемы распараллеливания, как и в прошлом примере, достигается за счет метода гомоморфных образов. Но теперь используются только числовые модули:

1) выберем необходимое количество числовых модулей.

2) решим задачу с помощью выше описанного алгоритма в каждом факторкольце.

3) восстановим решение в исходной области по китайской теореме об остатках, используя алгоритм Ньютона.

Таблица 1

Время, ускорение и эффективность параллельной программы факторизации многочленов.

Количество процессоров n	1	2	4	8	16	32
Время, с	2819	1429	752	379	202	113
Ускорение S_h		1.97	3.75	7.5	14	25
Эффективность E_h, %		98.6	94	93	87.2	78

Для многочлена $F(x,y,z)=f_1^5(x)f_2^7(y)f_3^7(z)f_4^5(x,y)f_5^5(x,z)f_6^7(y,z)f_7^7(x,y,z)$, где f_i – неприводимые плотные многочлены со 100-битными коэффициентами, у которых старшая степень по каждой переменной равна 2, приведем время вычисления сомножителей для различного количества процессоров в кластере. В таблице ускорение S_n показывает, во сколько раз быстрее происходят вычисления на n процессорах по сравнению с вычислением на одном процессоре: $S_n=t_1/t_n$, где t_n - время вычисления на n процессорах. Эффективностью E_n параллельного алгоритма мы называем величину: $E_n=(S_n/n)*100\%$.

Работа частично поддержана грантом РФФИ 12-07-00755а

Список литературы:

1. Малашонок Г.И. Руководство по языку "Mathpar". Тамбов: Издательский дом ТГУ, 2013, 133с.
2. Переславцева О.Н. О вычислении коэффициентов характеристического полинома. Вычислительные методы и программирование: новые вычислительные технологии. 2008. Т. 9. № 2. С. 180-184.
3. Ивашов Д.С. Об алгоритме факторизации полиномов многих переменных //Вестник Тамбовского Университета. Серия: Естественные и технические науки. 2012. 17, вып.2. 591-597

СПАСИБО ЗА ВНИМАНИЕ