

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ПРИ РЕШЕНИИ ЗАДАЧИ КОШИ НЕЯВНЫМИ МЕТОДАМИ РУНГЕ-КУТТЫ

Назарова И.А.

Кафедра ПМИИ, ДонНТУ
nazarova@r5.dgtu.donetsk.ua

Abstract

Nazarova I.A. Rise efficiency of the parallel computational systems for solution of Cauchy's problem by implicit Runge-Kutta methods. Parallel algorithms for systems of ordinary differential equations on base implicit Runge-Kutta methods are considered. The potential system and algorithm parallelism is exploited. Obtained algorithms are realized on DMM multicomputers with ring, matrix and hypercube topology. The theoretical performance analysis and a runtime simulation allow us to value the presented algorithms.

Проблема повышения эффективности функционирования высокопроизводительных параллельных вычислительных систем, несмотря на имеющиеся успехи, далека до полного разрешения, как в теоретическом, так и в практическом отношении[1]. Наиболее сложной является задача определения оптимальной структурной и алгоритмической организации параллельных вычислений с целью достижения достаточных характеристик эффективности[2]. В данной статье предлагается решение поставленной задачи в применении к одному конкретному классу методов интегрирования систем обыкновенных дифференциальных уравнений. Параллельные алгоритмы решения СОДУ на базе явных схем Рунге-Кутты, а также многошаговые алгоритмы Адамса-Башфорта достаточно изучены[3-5]. неявные методы Рунге-Кутты в силу большой вычислительной сложности и слабого внутреннего параллелизма широкого распространения не получили. В статье предлагается параллельный алгоритм полностью неявного метода Рунге-Кутты и исследуется эффективность его отображения на структуры многопроцессорных ВС различных топологий: кольцо, тор, гиперкуб.

1. Разработка параллельного алгоритма полностью неявного метода Рунге-Кутты

Численно решается задача Коши для обыкновенного дифференциального уравнения первого порядка с известными начальными условиями:

$$\begin{cases} y' = f(x, y); \\ y(x_0) = y_0. \end{cases} \quad (1)$$

Методы Рунге-Кутты с числом стадий, равным s в общем виде определяются формулами (2) и (3)[6]:

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^s a_{ij} k_j), i = \overline{1, s} \quad (2)$$

$$y_{n+1} = y_n + h \cdot \sum_{i=1}^s b_i \cdot k_i. \quad (3)$$

Коэффициенты a_{ij}, c_i, b_i определяют уникальный вариант метода Рунге-Кутты и выбираются из соображений точности. Если $a_{ij} = 0$ при $i \leq j$, то метод Рунге-Кутты является явным. В том случае, если $a_{ij} = 0$ при $i < j$ (нижняя треугольная матрица), и хотя бы одно значение $a_{ii} \neq 0$, то k_i определяются неявно из уравнения:

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^i a_{ij} \cdot k_j), i = \overline{1, s}, \quad (4)$$

такой метод называют диагонально неявным (ДНРК-метод).

Если в диагонально-неявном методе Рунге-Кутты все диагональные элементы одинаковы: $a_{ii} = \gamma$, то такой метод называют однократно диагонально неявным (ОДНРК-метод). Во всех остальных случаях имеем полностью неявный метод Рунге-Кутты, его представление в виде таблицы Батчера изображено на рисунке 1.

c_1	a_{11}	a_{12}	...	a_{1s}
c_2	a_{21}	a_{22}	...	a_{2s}
...
c_s	a_{s1}	a_{s2}	...	a_{ss}
	b_1	b_2	...	b_s

Рисунок 1 - Таблица Батчера для s - стадийного полностью неявного метода Рунге-Кутты

Рассмотрим параллельный алгоритм численного решения одного дифференциального уравнения полностью неявным методом Рунге-Кутты на основе квадратурных формул Радо и Лобатто[5]. Именно эти неявные методы имеют хорошие характеристики устойчивости и точности. Так например, s - стадийный метод РадоIA имеет порядок практически в 2 раза больше, чем число стадий и обладает А- устойчивостью. Однако характеристики параллелизма для всех полностью неявных методов не зависят от того, какой конкретно метод используется, поэтому далее процесс распараллеливания показан в общем виде.

$$\begin{cases}
 y_{n+1} = y_n + h \cdot \sum_{i=1}^s b_i \cdot k_i, \\
 k_1 = f[x_n + c_1 h, y_n + h(a_{11}k_1 + a_{12}k_2 + \dots + a_{1s}k_s)]; \\
 k_2 = f[x_n + c_2 h, y_n + h(a_{21}k_1 + a_{22}k_2 + \dots + a_{2s}k_s)]; \\
 \dots \\
 k_s = f[x_n + c_s h, y_n + h(a_{s1}k_1 + a_{s2}k_2 + \dots + a_{ss}k_s)].
 \end{cases} \quad (5)$$

Применение полностью неявного метода Рунге-Куты для решения ОДУ требует определения шаговых $k_i, i=1, \dots, s$ коэффициентов, которые связаны системой s нелинейных уравнений. Для решения такой системы используем метод последовательных итераций:

$$\begin{cases}
 g_i^{(0)} = 0; \\
 g_i^{(l)} = a_{i1}k_1^{(l)} + a_{i2}k_2^{(l)} + \dots + a_{is}k_s^{(l)}; \\
 i = 1, \dots, s.
 \end{cases}$$

$$\begin{cases}
 k_1^{(0)} = f(x_n + c_1 h, y_n + h \cdot g_1^{(0)}); \\
 k_2^{(0)} = f(x_n + c_2 h, y_n + h \cdot g_2^{(0)}); \\
 \dots \\
 k_s^{(0)} = f(x_n + c_s h, y_n + h \cdot g_s^{(0)}).
 \end{cases}$$

$$\dots$$

$$\begin{cases}
 k_1^{(N+1)} = f[x_n + c_1 h, y_n + h \cdot g_1^{(N)}]; \\
 k_2^{(N+1)} = f[x_n + c_2 h, y_n + h \cdot g_2^{(N)}]; \\
 \dots \\
 k_s^{(N+1)} = f[x_n + c_s h, y_n + h \cdot g_s^{(N)}].
 \end{cases}$$

В качестве модели высокопроизводительной параллельной вычислительной системы используется идеализированная модель - мультикомпьютер с распределенной памятью и коммутационной схемой произвольной топологии. Особенной чертой мультикомпьютера является тот факт, что доступ к локальной памяти значительно более быстрый, чем доступ к отдаленной памяти. По этой причине операции чтения/записи выполняются быстрее, чем операции приема/передачи. Поэтому желательно, чтобы доступ к локальным данным был более частым, чем обращение к отдаленным данным. Эта особенность называется локальностью и является одним из наиболее фундаментальных требований к параллельным программам.

Параллельный алгоритм решения ОДУ полностью неявным методом Рунге-Кутты разработан с использованием аппарата графов влияния[7] и представлен на рисунке 2.

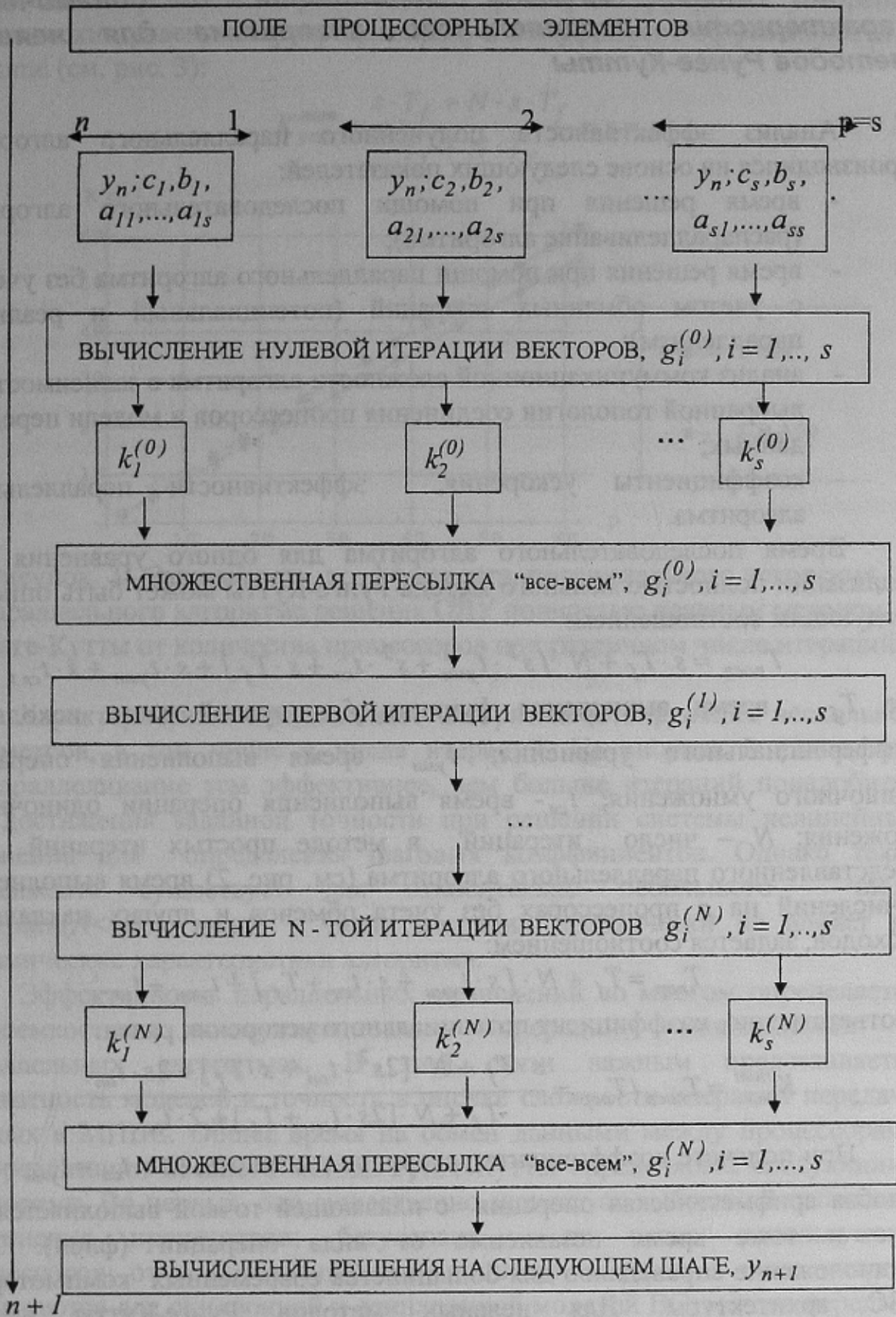


Рисунок 2- Алгоритм параллельного полностью неявного метода Рунге-Кутты для ОДУ

2. Анализ времени выполнения и динамических характеристик параллельного алгоритма для неявных методов Рунге-Кутты

Анализ эффективности полученного параллельного алгоритма производился на основе следующих показателей:

- время решения при помощи последовательного алгоритма (распараллеливание алгоритма);
- время решения при помощи параллельного алгоритма без учета и с учетом обменных операций (потенциальный и реальный параллелизм);
- анализ коммуникационной сложности алгоритма в зависимости от выбранной топологии соединения процессоров и модели передачи данных;
- коэффициенты ускорения, эффективности параллельного алгоритма.

Время последовательного алгоритма для одного уравнения при реализации полностью неявного метода Рунге-Кутты может быть описано следующим соотношением:

$$T_{\text{посл.}} = s \cdot T_f + N \cdot [s^2 \cdot t_{\text{умн}} + s^2 \cdot t_{\text{сл}} + s \cdot T_f] + s \cdot t_{\text{умн}} + s \cdot t_{\text{сл}},$$

где T_f - время вычисления функции f - правой части исходного дифференциального уравнения; $t_{\text{умн}}$ - время выполнения операции одиночного умножения; $t_{\text{сл}}$ - время выполнения операции одиночного сложения; N - число итераций в методе простых итераций. Для представленного параллельного алгоритма (см. рис. 2) время выполнения вычислений на p процессорах без учета обменов и других накладных расходов, задается соотношением:

$$T_{\text{пар}} = T_f + N \cdot [s \cdot t_{\text{умн}} + s \cdot t_{\text{сл}} + T_f] + t_{\text{умн}} + t_{\text{сл}}.$$

Соответственно, коэффициент потенциального ускорения равен:

$$K_{\text{уск}}^{\text{пот}} = T_{\text{посл.}} / T_{\text{пар}} = \frac{s \cdot T_f + N \cdot [2s^2 \cdot t_{\text{он}} + s \cdot T_f] + 2s \cdot t_{\text{он}}}{T_f + N \cdot [2s \cdot t_{\text{он}} + T_f] + 2 \cdot t_{\text{он}}}$$

При подсчете коэффициента ускорения считалось, что $t_{\text{сл}} = t_{\text{умн}} = t_{\text{он}}$ - любая арифметическая операция с плавающей точкой выполняется за одно и тоже время независимо от вида операции (флоп). Это предположение справедливо для большинства современных компьютеров RISC архитектуры. Для неявных методов Рунге-Кутты время интегрирования в основном определяется числом обращений к функции правой части уравнения (1) и соотношением между временем вычисления функции f и флопом. Проведенные эксперименты показали, что для сложных функций правой части (1) (время вычисления функции f

значительно превышает время одного флопа) коэффициент ускорения практически равен числу процессоров, а коэффициент эффективности – единице (см. рис. 3):

$$K_{\text{уск}}^{\text{ном}} \approx \frac{s \cdot T_f + N \cdot s \cdot T_f}{T_f + N \cdot T_f} = s = p$$

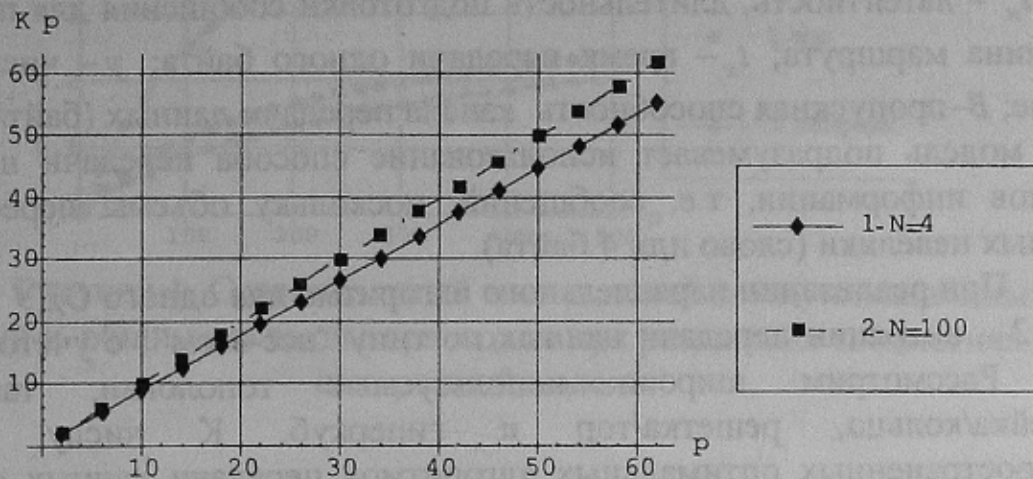


Рисунок 3 - Зависимость коэффициента потенциального ускорения параллельного алгоритма решения ОДУ полностью неявным методом Рунге-Кутты от количества процессоров при различном числе итераций

Характеристики параллельного алгоритма есть функции нескольких параметров, в том числе и числа итераций. Из рисунка 3 видно, что распараллеливание тем эффективнее, чем больше итераций понадобится для достижения заданной точности при решении системы нелинейных уравнений для определения шаговых коэффициентов. Однако такая зависимость существует для относительно небольшого числа итераций ($N < 100$), иначе – число итераций практически не влияет на динамические характеристики алгоритма.

Эффективность параллельных вычислений во многом определяется трудоемкостью коммуникационных операций, выполняемых в параллельных алгоритмах. В этой связи важным представляется адекватность моделей и точность в оценке сложности операций передачи данных в МПВС. Общее время на обмен данными между процессорами при реализации неявного метода Рунге-Кутты определяется следующими факторами. Во-первых, оно существенно зависит от выбранной топологии соединения процессоров; безусловно, от числа используемых процессоров; от коммуникационных констант, которые в свою очередь различаются для синхронной и асинхронной моделей ВС; метода передачи данных: сообщений или пакетов; и конечно, от характеристик самой задачи и алгоритма.

Для оценки времени выполнения операции передачи одного сообщения объемом n байт между двумя задачами локализованными на

различных процессорах (“точка-точка”) при распределенной памяти использовалась следующая модель[8]:

$$T_{m-m} = t_n + t_k \cdot n \cdot l, \quad t_k = \frac{y}{B}, \quad (6)$$

где t_n – латентность, длительность подготовки сообщения для передачи; l – длина маршрута; t_k – время передачи одного байта; y – число байт в слове; B – пропускная способность канала передачи данных (байт/секунда). Эта модель подразумевает использование способа передачи неделимых блоков информации, т.е. сообщений, поскольку объемы пересылаемых данных невелики (слово или 4 байта).

При реализации параллельного алгоритма для одного ОДУ требуется $N + 2$ операции передачи данных по типу “все-всем” с учетом модели (6). Рассмотрим широко используемые топологии, такие как линейка/кольцо, решетка/тор и гиперкуб. К числу наиболее распространенных оптимальных алгоритмов передачи данных относится класс методов покоординатной маршрутизации [8-10]. Идея этих методов заключается в том, что поиск путей передачи данных осуществляется последовательно для каждой размерности рассматриваемой топологии.

Для кольцевой топологии каждый процессор может инициировать рассылку своего сообщения в каком-либо выбранном направлении по кольцу. Без существенных ограничений предполагаем, что любой процессор имеет возможность осуществлять одновременно прием и передачу данных (двунаправленные линки). Таким образом, для топологии кольцо коммуникационная составляющая первого алгоритма равна:

$$T_{обм} = (N + 2) \cdot (p - 1) \cdot T_{m-m}, \text{ причем } l=1, n=1.$$

Для топологии решетка/тор множественная рассылка по типу “все-всем” может быть осуществлена обобщением способа передачи для кольцевой структуры, вначале организуется передача данных по горизонтали, а затем – по вертикали решетки:

$$T_{обм} = 2(N + 2) \cdot (\sqrt{p} - 1) \cdot t_n + (N + 2) \cdot (p - 1)t_k.$$

Для гиперкуба алгоритм множественной рассылки может быть получен обобщением способа передачи “все – всем ” для решетки на размерность куба: $lp = \log_2 p$.

$$T_{обм} = (N + 2) \cdot \sum_{i=1}^{lp} (t_n + 2^{i-1} t_k) = t_n \cdot (N + 2) \cdot \log_2 p + t_k \cdot (p - 1)(N + 2).$$

Более информативной величиной является отношение коммуникационных затрат к общим накладным расходам параллельного алгоритма:

$$Z = \frac{T_{обмена}}{T_{общ}} = \frac{T_{обмена}}{T_{обмена} + T_{выч}}.$$

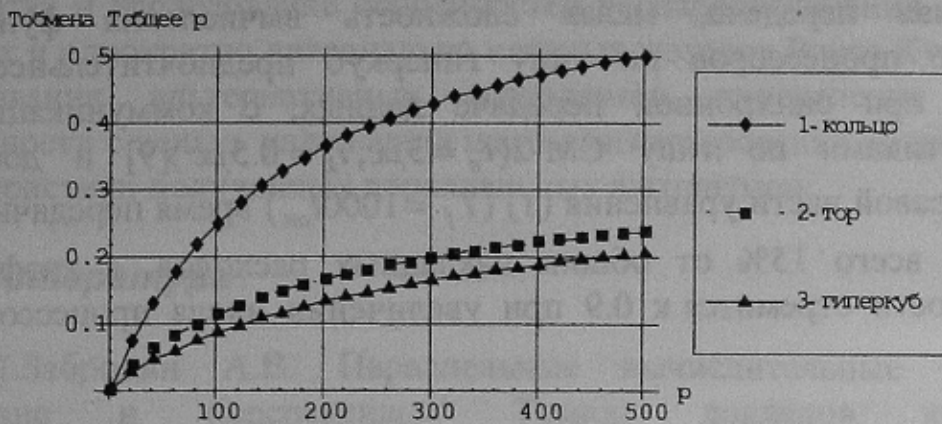


Рисунок 4- Сравнительный анализ долей коммуникационных операций к общим накладным расходам при различных топологиях для синхронной модели

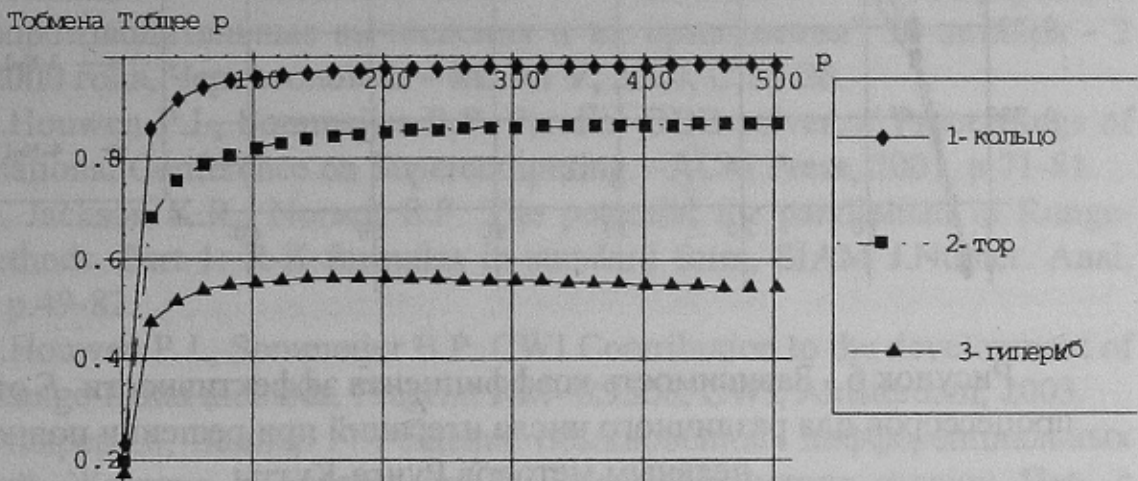


Рисунок 5 - Сравнительный анализ долей коммуникационных операций к общим накладным расходам при различных топологиях для асинхронной модели

Выберем оптимальную топологию соединения процессоров, если таковая существует при варьировании всех остальных характеристик. Выбор топологии производился при анализе множества следующих характеристик $T_{обм}$, Z , $K_{уск}$, E , как некоторых функций $\varphi(p, t_n, t_k, t_{оп}, T_f)$. Анализ предложенных топологий показал, что безусловно наихудшим вариантом для рассматриваемого алгоритма является соединение линейка/кольцо (см. рис. 4,5), так при использовании характеристик многопроцессорной системы iPSC/860 ($t_n = 175 \mu s$, $t_k = 0.39 \mu s$) [8] имеем 70% операций обменов и эффективность параллельных вычислений равную 0.2. Топологии гиперкуб и тор часто дают достаточно близкие

характеристики, но даже при худшей комбинации всех аргументов (асинхронная передача, малая сложность вычисления функции f) соединение процессоров по типу гиперкуб предпочтительнее. Так, например, при синхронной передаче данных, с коммуникационными характеристиками по типу CM-2 ($t_n = 3 \mu\text{с}$, $t_k = 0.5 \mu\text{с}$) [9] и достаточно сложной правой части уравнения (1) ($T_f = 1000t_{on}$) время передачи данных составляет всего 15% от общих накладных расходов, а коэффициент эффективности стремится к 0.9 при увеличении числа процессоров.

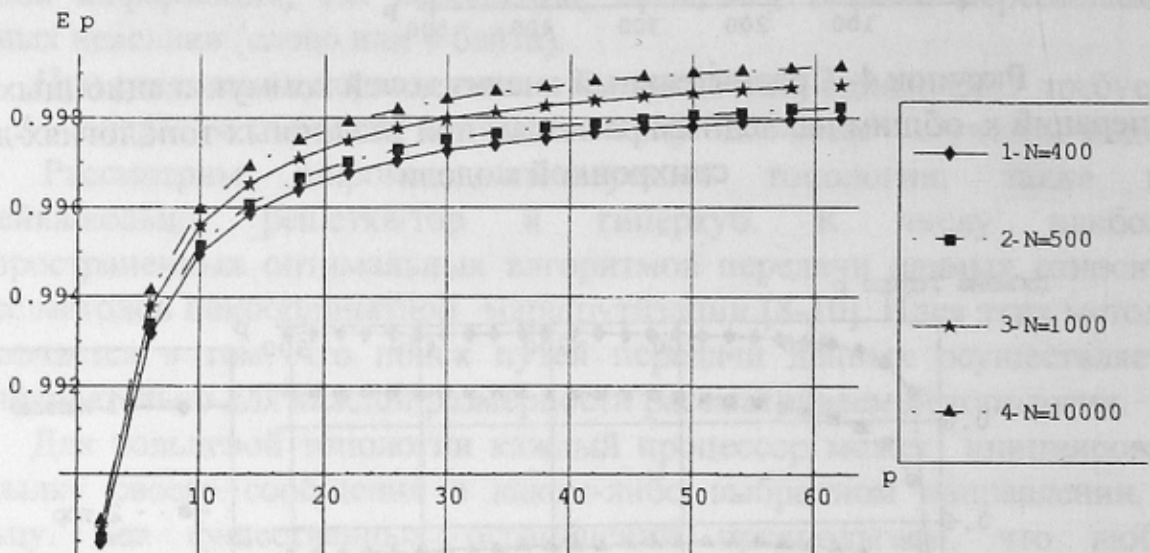


Рисунок 6 - Зависимость коэффициента эффективности E от числа процессоров для различного числа итераций при решении полностью неявным методом Рунге-Кутты

Определение характеристик параллелизма осуществлялось с помощью пакета *Mathematica*® (Wolfram Research Inc.).

Заключение

Численный эксперимент и проведенный сравнительный анализ динамических характеристик параллельного алгоритма на основе неявных методов Рунге-Кутты показал, что достаточно большая по сравнению с явными методами вычислительная сложность этих методов не является препятствием для использования их в высокопроизводительных мультипроцессорных системах. Однако требуется тщательный учет всех составляющих параллельной системы для того, чтобы такое решение было масштабируемым и эффективным. В частности практически линейное ускорение и близкая к единичной эффективность достигаются при использовании синхронной передачи данных в топологии гиперкуб.

Перспективным направлением дальнейших исследований является разработка и исследование особенностей распараллеливания диагонально неявных и однократно диагонально неявных методов Рунге-Кутты, а также исследование альтернативных механизмов определения локальной погрешности с целью управления шагом интегрирования на динамические характеристики полученных параллельных алгоритмов.

Литература

- 1.Забродин А.В. Параллельные вычислительные технологии. Состояние и перспективы// Тезисы докладов конференции "Высокопроизводительные вычисления и их приложения" 30 октября - 2 ноября 2000 года, Черногловка. – М.:МГУ, 2000, С.8-9.
- 2.Андреев А.Н., Антонов А.С., Воеводин В.В. и др. Комплексный подход к анализу эффективности программ для параллельных вычислительных систем.//Тезисы докладов конференции "Высокопроизводительные вычисления и их приложения" 30 октября - 2 ноября 2000 года, Черногловка. – М.:МГУ, 2000, С.18-20.
- 3.Houwen P.J., Sommeijer B.P. Parallel ODE solver. // Proceedings of the International Conference on Supercomputing. –ACM Press, 2001, p.71-81.
4. Jackson K.R., Norsett S.P. The potential for parallelism in Runge-Kutta methods. Part 1: R-K formulas in standard form, SIAM J.Numer. Anal. 32,2001, p.49-82.
- 5.Houwen P.J., Sommeijer B.P. CWI Contribution to the development of parallel Runge-Kutta methods, Preprint NM- R9506, CWI, Amsterdam, 2003.
- 6.Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи: Пер. с англ. – М.: Мир, 1999. – 685с.
- 7.Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608с.
- 8.Foster I. Designing and Building Parallel Programs. – Addison-Wesley, 1999. – 302 с.
9. Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных ВС. – СПб.: БХВ-Петербург, 2002. – 400с.
10. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных машин. – Нижний Новгород: Изд-во ННГУ им. Н.И.Лобачевского, 2000. – 176с.

Дата надходження до редакції 10.05.2005 р.