

И.А. Назарова, канд. техн. наук, доцент,
Л.П. Фельдман, д-р техн. наук, профессор,
Донецкий национальный технический университет, г. Покровск, Украина
ianazard@gmail.com

Масштабируемость параллельных алгоритмов и архитектур при численном решении задачи Коши на основе явных разностных схем

Приведена методика оценки масштабируемости параллельных алгоритмов и архитектур на базе теории изоэффективного анализа. Определены динамические характеристики параллельных одношаговых многостадийных явных методов решения задачи Коши. Проведен изоэффективный анализ параллельных алгоритмов интегрирования СОДУ с альтернативными способами оценки локальной апостериорной погрешности: дублирование шага, вложенные формы и экстраполяция Ричардсона.

Ключевые слова: масштабируемость, функция изоэффективности, общие накладные расходы, задача Коши.

Введение

В связи с интенсивным развитием компьютерной индустрии метрики масштабируемости становятся одними из наиболее важных характеристик при проектировании параллельных алгоритмов и архитектур [1-6]. Однако до сих пор не существует общепринятого определения этого термина. Поэтому введение формального определения и теоретического обоснования метрики масштабируемости для высокопроизводительных вычислений является актуальной и перспективной научно-исследовательской задачей. Не менее важными являются задачи разработки практического инструментария для оценки масштабируемости и сравнения метрики с существующими динамическими характеристиками качества параллельных алгоритмов такими, как коэффициенты ускорения и эффективности.

Целью исследования является повышение эффективности параллельного численного решения задачи Коши для систем обыкновенных дифференциальных уравнений (СОДУ).

Задачей исследования является разработка методов оценки масштабируемости параллельных вычислений при решении задачи Коши для выбора наиболее оптимальной комбинации алгоритма и параллельной архитектуры.

Масштабируемость параллельного решения задачи Коши на основе явных одношаговых разностных схем

Рассматривается численное решение задачи Коши, ассоциированное с решением СОДУ первого порядка с известными начальными условиями:

$$\begin{cases} \frac{d\bar{x}(t)}{dt} = \bar{f}(t, \bar{x}(t)), \\ \bar{x}(t_0) = \bar{x}_0, \end{cases} \quad (1)$$

где правая часть является в общем случае нелинейной функцией и задает отображение $F = \bar{f} : R \times R^m \rightarrow R^m$. Общепринятыми методами оценки локальной апостериорной погрешности решения СОДУ есть [3-4]: дублирование шага по правилу Рунге, технология локальной экстраполяции Ричардсона (ЛЕР) и вложенные методы.

Динамические характеристики параллельных вложенных методов решения задачи Коши на основе явных одношаговых схем

Наиболее эффективным способом определения локальной погрешности при решении задачи Коши являются методы вложенных форм, которые дополнительно требуют вычисления одного или нескольких стадийных коэффициентов для определения более точной аппроксимации решения на шаге. Вложенные методы в случае одношаговых численных схем определяют две формулы Рунге-Кутты смежных порядков точности $r(r')$ и для интегрирования СОДУ имеют вид (ВМРК):

$$\begin{cases} \bar{x}_{n+1} = \bar{x}_n + h \cdot \sum_{i=1}^s b_i \cdot \bar{k}_i, \\ \hat{x}_{n+1} = \bar{x}_n + h \cdot \sum_{i=1}^{s'} \hat{b}_i \cdot \bar{k}_i. \end{cases} \quad (2)$$

Если основу вложенных форм составляют явные разностные схемы (ВЯМРК), то стадийные коэффициенты определяются как:

$$\begin{cases} \bar{k}_i = F(t_n + c_i h; \bar{g}_i), \\ \bar{g}_i = \bar{x}_n + h \sum_{j=1}^{i-1} a_{ij} \cdot \bar{k}_j, i = \overline{1, s'} \end{cases} \quad (3)$$

Время на реализацию последовательного ВЯМРК порядка $r(\hat{r})$ включает вычисление стадийных коэффициентов обеих аппроксимаций решения:

$$T_I = (s+1)mT_F + [s^2m + 6sm + 2s + 4m]t_{op}, \quad (4)$$

где $T_F = \sum_{i=1}^m T_{fi}$ – время вычисления правой части

системы (1), t_{op} – время выполнения одной операции с плавающей точкой. Параллельный алгоритм для ВЯМРК учитывает параллелизм системы и метода, и подробно описан в работах [3-4]. Время реализации параллельного алгоритма ВЯМРК составляет время на выполнение арифметических операций и время на операции передачи данных (операция "all-to-all"):

$$T_{p,comp} = m_I(s+1)T_F + (s^2m_I + 6sm_I + 2s + 4m_I)t_{op} \quad (5)$$

$$T_{p,comm} = s'T_{all-to-all}(m_I, p), \quad m_I = \lceil m/p \rceil, \quad s' = s+1. \quad (6)$$

Определим время передачи данных алгоритма ВЯМРК для различных топологий, объем передаваемых данных равен $\lceil m/p \rceil$, t_s – латентность, t_w – время передачи одного слова:

$$\begin{aligned} \text{– кольцо: } T_{all-to-all}^R &= s' \cdot \left[(t_s + \lceil m/p \rceil \cdot t_w) \cdot (p-1) \right]; \\ \text{– тор: } T_{all-to-all}^M &= s' \cdot \left[2t_s(\sqrt{p}-1) + \lceil m/p \rceil \cdot t_w \cdot (p-1) \right]; \\ \text{– гиперкуб: } T_{all-to-all}^H &= s' \cdot \left[t_s \cdot \log_2 p + t_w \cdot \lceil m/p \rceil \cdot (p-1) \right]. \end{aligned}$$

Анализ масштабируемости параллельных алгоритмов ВЯМРК требует определить их динамические характеристики, как функции:

а) параметров задачи и метода:

1) размера входа – размерность СОДУ, m ;
2) порядка примененного метода интегрирования, $r(r')$, $r' = r+1$ и числа стадий $s(s')$, $s' = s+1$;

3) сложности правой части (1), соотношение времени ее вычисления и времени на реализацию одной операции с плавающей точкой, флопом: $T_F \gg t_{op}$ или $T_F \sim t_{op}$;

б) временных характеристик параллельной архитектуры: t_{op}, t_s, t_w и числа процессоров.

Исследование вариации общего времени выполнения для параллельных явных вложенных одношаговых методов

Сначала проводится анализ поведения общего времени реализации параллельного алгоритма на параллельной архитектуре, $T_p = T_{p,comp} + T_{p,comm}$. Известно, что при

увеличении числа процессоров в ВС общее время выполнения параллельного алгоритма уменьшается. Но существуют ситуации, при которых величина T_p при росте p достигает минимального значения, а затем начинает расти (см. рис. 1).

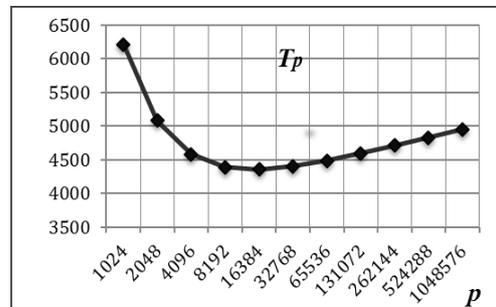


Рисунок 1 – График зависимости общего времени реализации параллельного алгоритма вложенного метода Рунге-Кутты от числа процессоров для ВС топологии гиперкуб

Определим средствами изоэффективного анализа [1-2] за счет каких причин возникает такой эффект, связано ли это с параметрами ВС или алгоритма, или по другим причинам. Разработаем алгоритм нахождения такого значения числа процессоров p , при котором время выполнения параллельного алгоритма будет минимальным, а ускорение и эффективность максимальными: p_{max} . При использовании числа процессоров более чем p_{max} , или достигается насыщение, или время выполнения начинает расти. На практике каждый параллельный алгоритм имеет ненулевую последовательную часть (закон Амдала), которая не может быть распараллелена. С этой точки зрения, время общего выполнения параллельного алгоритма на параллельной архитектуре может быть представлено через функцию общих накладных расходов на параллелизм:

$$T_p = \frac{T_I + T_o(T_I, p)}{p}. \quad (7)$$

Поскольку T_I не зависит от p , поведение T_p определяется соотношением темпа увеличения/уменьшения накладных расходов и количества процессоров. Пусть $T_o \leq \Theta(p)$: величина накладных расходов растет медленнее, чем размерность процессорного поля. Тогда общая степень p в (7) будет отрицательной, поэтому следует ожидать, что с ростом p величина T_p будет продолжать уменьшаться до бесконечности. В противном случае, $T_o > \Theta(p)$, скорость роста накладных расходов на параллелизм выше скорости роста числа

процессоров, существует нижняя граница для T_p . Однако, не следует забывать, что максимальное количество процессоров имеет в качестве верхней границы максимальную степень параллелизма, величину Dop , которая в свою очередь зависит от параметров последовательного алгоритма [5]. Применим приведенные рассуждения к анализу общего времени выполнения параллельного алгоритма ВЯМРК с обходами на топологии гиперкуб:

$$T_p = \frac{T_1}{p} + \frac{T_o(T_1, p)}{p}. \quad (8)$$

Терм T_1/p в (8) уменьшается с ростом p , в то время, как терм T_o/p увеличивается при тех же условиях (рис. 2).

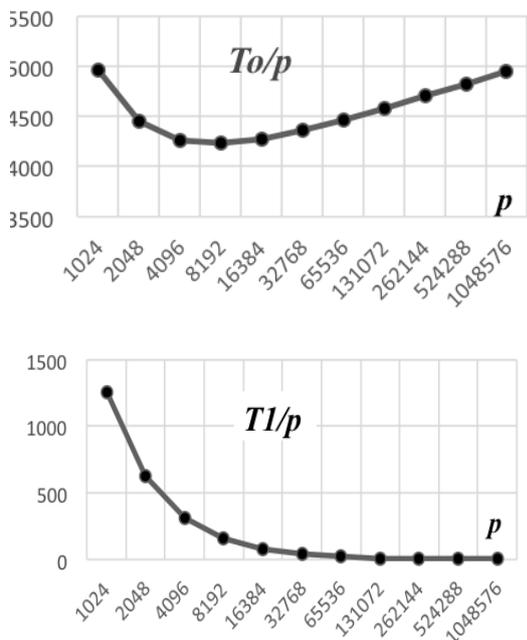


Рисунок 2 – Графики зависимости составляющих общего времени реализации параллельного алгоритма ВЯМРК 4 (5) от числа процессоров для ВС топологии гиперкуб

Тогда, T_p будет вначале уменьшаться, а потом, после некоторого p_0 , будет увеличиваться с ростом p . Определим соотношение между параметрами, которые доставляют минимум функции T_p :

$$\frac{dT_p}{dp} = \frac{d(T_1/p)}{dp} + \frac{d(T_o/p)}{dp} = 0. \quad (9)$$

$$\frac{d(T_1/p)}{dp} = -T_1/p^2, \quad \frac{d(T_o/p)}{dp} = -\frac{T_o}{p^2} + \frac{d}{dp} \frac{T_o}{p},$$

$$-T_1/p^2 - \frac{T_o}{p^2} + \frac{d}{dp} \frac{T_o}{p} = 0.$$

$$\frac{dT_o(T_1, p)}{dp} = \frac{T_1}{p} + \frac{T_o}{p}, \quad \text{но по определению}$$

накладные расходы на параллелизм вычисляются, как: $T_o = pT_p - T_1 \Rightarrow T_p = (T_o + T_1)/p$.

Таким образом, независимо от метода и параллельной вычислительной системы при наличии ограничения $T_o > \Theta(p)$, имеем уравнение (10), решение которого даст значение максимального числа процессоров для получения минимального общего времени реализации параллельного алгоритма:

$$\frac{dT_o(T_1, p)}{dp} = T_p(m, p, t_{op}, t_s, t_w). \quad (10)$$

Обычно функция T_o – сумма термов, каждый из которых является произведением машино-зависимой константы, параметров метода и задачи на некоторую функцию от числа процессоров. Непосредственно для ВЯМРК и вычислительной системы с топологией гиперкуб соотношение (8) имеет вид:

$$\frac{dT_o(T_1, p)}{dp} = (s+1)(\log_2 p + \log 2)t_s + m(s+1)t_w + 2st_{op} =$$

$$(s+1) \cdot [t_s \cdot \log_2 p + t_w \cdot \lceil m/p \rceil \cdot (p-1)] + \lceil m/p \rceil (s+1)T_F +$$

$$+ (s^2 \lceil m/p \rceil + 6s \lceil m/p \rceil + 2s + 4 \lceil m/p \rceil) t_{op} = T_p^H. \quad (11)$$

Решение (11) для ВЯМРК порядка 4(5) для ВС топологии гиперкуб с временными параметрами $t_{op} = 2, t_s = 25, t_w = 4$ приведено на рисунке 3.

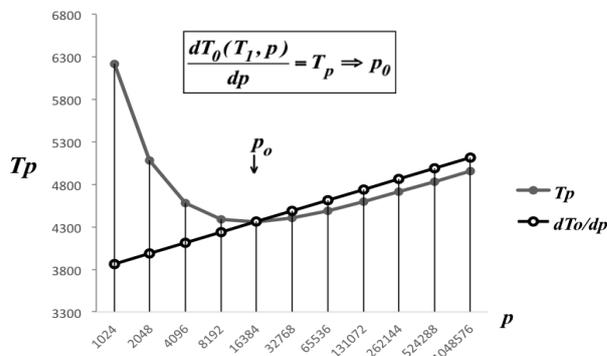


Рисунок 3 – Графики зависимости общего времени реализации параллельного алгоритма ВЯМРК 4(5) и производной от накладных расходов при вариации числа процессоров

Уравнение дает математическое значение для p_0 , но полученное число может превышать максимальную степень параллелизма алгоритма Dop , поэтому на практике в качестве пиковой точки для p при определении минимального общего времени параллельной реализации алгоритма выбирают следующую величину (при фиксированном размере входа):

$$p_{max} = \text{Min}(p_0, Dop). \quad (12)$$

Для описанного параллельного алгоритма максимальная степень параллелизма равна размерности СОДУ: $Dop = m$, поэтому $p_{max} = \text{Min}(p_o, Dop) = Dop$, коэффициенты ускорения и эффективности в пиковой точке могут быть вычислены по формулам:

$$S_p = \frac{T_1 \cdot Dop}{T_1 + T_o(T_1, p_{max})} = \frac{T_1 \cdot Dop}{T_1 + T_o(T_1, Dop)},$$

$$E_p = \frac{T_1}{T_1 + T_o(T_1, Dop)}.$$

Для конкретного примера на рисунках 1 и 4 приведены зависимости общего времени выполнения и эффективности параллельного алгоритма ВЯМРК 4(5) от числа процессоров p для ВС топологии гиперкуб с параметрами $t_{op} = 2, t_s = 25, t_w = 4$ с одинаковой пиковой точкой ($p \approx 16901$), но $p_{max} = Dop = m = 512$.

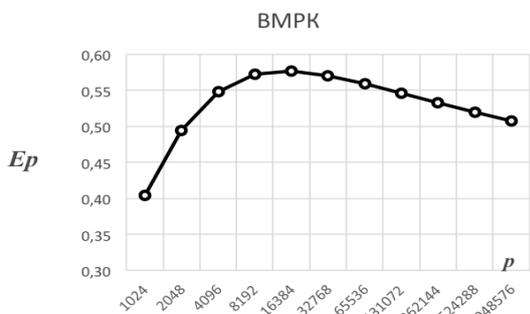


Рисунок 4 – График зависимости коэффициента эффективности алгоритма ВЯМРК от числа процессоров p для ВС топологии гиперкуб

Построение и анализ функции изоэффективности для вложенных стадийных явных численных схем

Основное соотношение изоэффективного анализа для параллельного алгоритма ВЯМРК с отображением на ВС топологии гиперкуб имеет вид:

$$T_1 = K \cdot T_o^H \Rightarrow$$

$$T_1 = (s + 1)mT_F + (s^2m + 6sm + 2s + 4m)t_{op} =$$

$$= K \cdot T_o^H = K[(s + 1)(p \log_2 p) \cdot t_s +$$

$$+ m(s + 1)(p - 1) \cdot t_w + 2s(p - 1) \cdot t_{op}]$$

Определим функцию изоэффективности, то есть получим зависимость $m = f_E(p)$, на основе которой можно вычислить, как необходимо увеличивать размер задачи при увеличении количества процессоров для поддержки постоянной эффективности:

$$m = f_E(p; T_F, t_{op}, t_s, t_w; s; K).$$

$$m = \frac{K[(s + 1)(p \log_2 p) \cdot t_s + 2s(p - 1) \cdot t_{op}] - 2s \cdot t_{op}}{(s + 1) \cdot T_F + (s^2 + 6s + 4) \cdot t_{op} - K(s + 1)(p - 1) \cdot t_w},$$

где $K = E / 1 - E$ и выполняется неравенство:

$$p < \frac{(s + 1) \cdot T_F + (s^2 + 6s + 4) \cdot t_{op} + K(s + 1) \cdot t_w}{K(s + 1) \cdot t_w}.$$

Исходя из полученных выше соотношений и согласно общепринятой классификации [1-2], рассмотренный параллельный алгоритм ВЯМРК является квазилинейным по отношению к числу задействованных процессоров, то есть имеет достаточно высокую масштабируемость. Рисунок 5 представляет коэффициент ускорения при различных значениях размерности задачи для асинхронных параллельных систем с низкоскоростными коммуникационными сетями: высокая латентность и небольшая скорость передачи данных для сложных правых частей СОДУ. Как и ожидалось, для малой размерности $m \leq O(100t_{op})$ ускорение достигает максимума уже при небольшом числе процессоров: $p \approx 100$. За этой точкой увеличение размера процессорного поля существенного влияния на ускорение не дает. С другой стороны, ускорение для задач большой размерности со сложными правыми частями СОДУ близко к линейному.

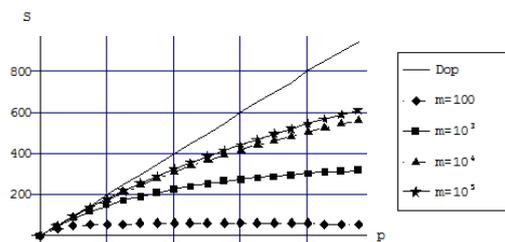


Рисунок 5 – Графики зависимости ускорения ВЯМРК от числа процессоров для ВС топологии гиперкуб с коммуникационной средой высокой латентности при $T_F \gg t_{op}$

Для МИМД-систем с быстрыми каналами связи масштабируемость алгоритма увеличивается, то есть ускорение с ростом числа процессоров становится практически линейным, особенно для задач большой размерности.

Аналогично выводятся выражения для функций изоэффективности ВМРК с топологиями соединения процессоров в параллельной архитектуре: 2D-тор и кольцо (рис. 6). Очевидно, что размерность решаемой СОДУ должна возрастать намного быстрее на торе, чем на гиперкубе, чтобы поддерживать идентичную эффективность. Топология "кольцо" для исследуемых параллельных алгоритмов имеет худшие характеристики масштабируемости.

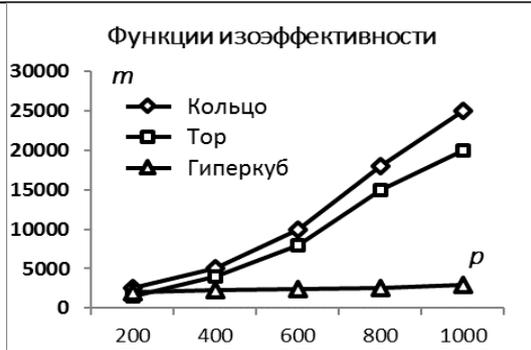


Рисунок 6 – Функции изоэффективности для МІМД-систем топологии гиперкуб, 2D-тор и кольцо для ВЯМРК при $E = 0,9$

Изоэффективный анализ явных одношаговых методов параллельного решения СОДУ со встроенными алгоритмами оценки локальной погрешности

Применение теории изоэффективного анализа позволяет сделать выводы относительно качества параллельного алгоритма на основе единого аналитического выражения, сократив тем самым многочисленные эксперименты при различных комбинациях параметров от которых зависят характеристики параллелизма. Составляющие динамических характеристик для явных параллельных одношаговых методов типа Рунге-Кутты со встроенными способами оценки локальной апостериорной погрешности:

а) коэффициент при T_F :

- 1) правило Рунге: $(3s - 1) \frac{m}{p}$;
- 2) технология ЛЭР: $\left(\frac{s^2 - 2s + 4}{2} \right) \cdot \frac{m}{p}$;
- 3) вложенные методы: $(s + 1) \frac{m}{p}$;

б) коэффициент при t_{op} :

- 1) правило Рунге: $3 \frac{m}{p} s^2 + 6 \frac{m}{p} s + 6s - 6$;
- 2) технология ЛЭР: $\left[\left(\frac{s^2 - 2s + 4}{2} \right) + \left(\frac{21}{8} s^2 - \frac{42}{8} s + 8 \right) \frac{m}{p} \right]$;
- 3) вложенные методы: $\frac{m}{p} s^2 + 6 \frac{m}{p} s + 2s + 4 \frac{m}{p}$;

в) коэффициент при t_s :

- 1) правило Рунге: $3s \log_2 p$;
- 2) технология ЛЭР: $\left(\frac{s^2 - 2s + 4}{2} \right) \log_2 p$;
- 3) вложенные методы: $(s + 1) \log_2 p$;

г) коэффициент при t_w :

- 1) правило Рунге: $3s \frac{m}{p} (p - 1)$;

2) технология ЛЭР: $\left(\frac{s^2 - 2s + 4}{2} \right) \frac{m}{p} (p - 1)$;

3) вложенные методы: $(s + 1) \frac{m}{p} (p - 1)$.

Для ЯМРК с правилом Рунге и ЛЭР, как и в случае вложенных методов, имеет место соотношение: $T_o > \Theta(p)$, то есть скорость роста накладных расходов на параллелизм выше, чем скорость роста числа процессоров. Таким образом, для этих методов также существует нижняя граница для T_p , после которой общее время на реализацию параллельного алгоритма не только не уменьшается с ростом p , но и начинает расти (рис. 7).

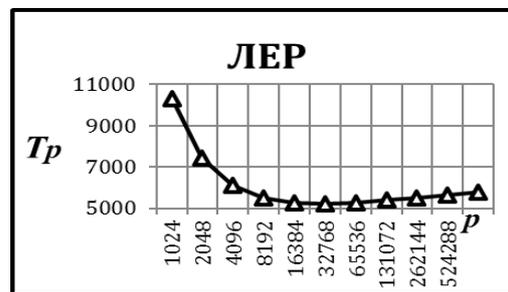


Рисунок 7 – Графики зависимости общего времени на реализацию параллельного алгоритма ЯМРК от числа процессоров p для ВС топологии гиперкуб с параметрами $t_{op} = 2, t_s = 25, t_w = 4$

Максимальная степень параллелизма для описанных методов равна размерности СОДУ: $p_0 > Dop$, поэтому $p_{max} = \text{Min}(p_0, Dop) = m$, ускорение и эффективность в пиковой точке определяются следующим образом:

$$S_p = \frac{T_1 \cdot Dop}{T_1 + T_o(T_1, Dop)}, E_p = \frac{T_1}{T_1 + T_o(T_1, Dop)}$$

Общие накладные затраты на параллелизм вычисляются по формуле: $T_o = pT_p - T_1$ и для наиболее прогрессивной топологии гиперкуб составляют:

- 1) правило Рунге:
 $T_o = 3s(p \log_2 p) \cdot t_s + 3ms(p - 1) \cdot t_w + 6(s - 1)(p - 1) \cdot t_{op}$;

2) локальная экстраполяция Ричардсона:

$$T_o = \left(\frac{s^2 - 2s + 4}{2} \right) (p \log_2 p) \cdot t_s + m \left(\frac{s^2 - 2s + 4}{2} \right) (p - 1) \cdot t_w + \left(\frac{s^2 - 2s + 4}{2} \right) (p - 1) \cdot t_{op}.$$

Для сравнения качества двух или большего числа параллельных алгоритмов теория изоэффективного анализа предлагает выполнить сравнение не только коэффициентов ускорения и эффективности, но в первую очередь функций общих накладных расходов $T_o^{Alg 1} \cong T_o^{Alg 2} \cong T_o^{Alg 3}$, как в целом так и по отдельным термам (t_s, t_w, t_{op}) (рис. 8-9).

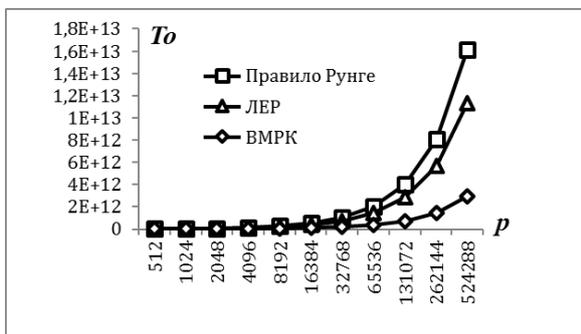


Рисунок 8 – Графики зависимости накладных расходов параллельных алгоритмов ЯМРК со встроенными способами оценки локальной погрешности от числа процессоров p для ВС топологии гиперкуб

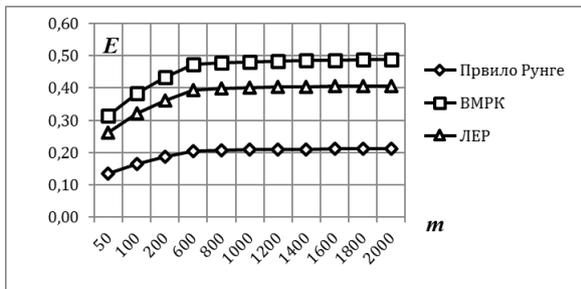


Рисунок 8 – Графики зависимости коэффициента эффективности параллельных алгоритмов ЯМРК со встроенными способами оценки локальной погрешности от размерности задачи для ВС топологии гиперкуб

Заметим, что оценки общих накладных расходов являются абсолютными, так как получены на основе соответствующих последовательных алгоритмов, которые были распараллелены, а не наилучшего с точки зрения временных затрат последовательного метода решения исходной задачи Коши – вложенного ЯМРК. Не смотря на этот факт, полностью очевидно, что по всем динамическим характеристикам с учетом влияния всех параметров вложенные методы являются наименее затратным способом оценки локальной

погрешности для явных одношаговых численных схем решения СОДУ. Аналогичный вывод можно получить при анализе составляющих сложности и масштабируемости ВМРК с правилом Рунге и ЛЭР отдельно по временным термам. Напомним, что в данном исследовании речь идет о динамических характеристиках численных методов интегрирования СОДУ с точки зрения качества распараллеливания, эффективности использования параллельной архитектуры, масштабирования вычислений.

Заключение

В данной статье приведены результаты исследований, посвященных анализу масштабируемости и эффективности параллельных приложений в совокупности с архитектурой, на которой они реализованы, с использованием аппарата изоэффективного анализа. Проведена оценка степени масштабируемости параллельных алгоритмов решения задачи Коши для СОДУ на основе явных одношаговых численных схем путем построения функции изоэффективности в зависимости от размера решаемой задачи, характеристик методов (порядок, число стадий), параметров ВС: типа топологии соединения процессоров в коммуникационной сети и временных машинно-зависимых констант. Получены динамические характеристики методов, исследован характер поведения времени параллельной реализации алгоритма, общих накладных затрат на параллелизм, ускорения, эффективности параллельного оборудования, получена оценка для максимального значения числа используемых процессоров.

Научная новизна предложенной методики оценки качества и масштабируемости параллельных численных методов заключается в повышении эффективности решения задачи Коши за счет подбора оптимальной комбинации алгоритм/архитектура.

Применение математического аппарата изоэффективного анализа позволило перевести исследования качества и масштабируемости параллельных алгоритмов при варьировании многочисленных динамических параметров задачи, метода и параллельной системы из области многоразового эксперимента для больших размерностей в аналитическую область.

Практическая значимость оценки масштабируемости состоит в построении оптимальных параллельных алгоритмов для заданных условий реализации, а также в формировании областей приоритетного использования для различных алгоритмов решения одной и той же задачи.

Список литературы

1. Гергель В.П. Теория и практика параллельных вычислений. Москва, ИНТУИТ, Лаборатория знаний «Бином», 2007, 423 с.
2. Grama, A., Gupta, A., Kumar, V., Karypis, G. (2003), *Introduction to Parallel Computing*, Addison Wesley, 856p.
3. Фельдман Л.П., Назарова І.А. Паралельні однокрокові методи чисельного розв'язання задачі Коші. Донецьк, ДонНТУ, 2011. 185 с.
4. Фельдман Л.П., Назарова І.А. Современные параллельные методы численного решения задачи Коши. Донецк, ДонНТУ, 2013. 206 с.
5. Назарова І.А., Чорна А.О. Дослідження масштабованості паралельних систем на основі функції ізоефективності. Наукові праці ДнНТУ, серія «Інформатика, кібернетика та обчислювальна техніка», №2 (21), 2015. ДонНТУ, Красноармійськ, с. 56-62.
6. Назарова І.А. Анализ масштабируемости параллельных алгоритмов численного решения задачи Коши. Наукові праці ДнНТУ, серія «Інформатика, кібернетика та обчислювальна техніка», №10, 2009. ДонНТУ, Донецьк, с. 21-26.

Надійшла до редакції 18.03.2017

І.А. НАЗАРОВА, Л.П. ФЕЛЬДМАН

Донецький національний технічний університет (Україна)

МАСШТАБОВАНІСТЬ ПАРАЛЕЛЬНИХ АЛГОРИТМІВ ТА АРХІТЕКТУР ПРИ ЧИСЕЛЬНОМУ ВИРІШЕННІ ЗАДАЧІ КОШІ НА ОСНОВІ ЯВНИХ РІЗНИЦЕВИХ СХЕМ

Наведено методику оцінки масштабованості паралельних алгоритмів і архітектур на базі теорії ізоефективного аналізу. Визначено динамічні характеристики паралельних однокрокових багатостадійний явних методів вирішення задачі Коші. Проведено ізоефективний аналіз паралельних алгоритмів інтегрування СЗДР з альтернативними способами оцінки локальної апостеріорної похибки: дублювання кроку, вкладені форми і екстраполяція Річардсона.

Ключові слова: масштабованість, функція ізоефективності, загальні накладні витрати, задача Коші.

I.A. NAZAROVA, L.P. FELDMAN

Donetsk National Technical University (Ukraine)

SCALABILITY OF PARALLEL ALGORITHMS AND ARCHITECTURES IN THE NUMERICAL SOLUTION OF THE CAUCHY'S PROBLEM ON THE BASIS OF EXPLICIT DIFFERENCE SCHEMES

A methodology for evaluating the scalability of parallel algorithms and architectures based on the theory of isoeffective analysis is presented. Dynamic characteristics of parallel one-step multi-stage explicit methods for solving the Cauchy's problem are determined. An isoeffective analysis of parallel algorithms for integrating SODE with alternative methods for estimating local a posteriori error is performed: step duplication, embedded forms, and Richardson extrapolation.

This article presents the results of the analysis of the scalability of parallel applications in combination with the architecture on which they are implemented. The degree of scalability of parallel algorithms for solving the Cauchy problem for SODE is calculated on the basis of explicit one-step numerical schemes by constructing the isoefficiency function depending on the size of the problem being solved, the characteristics of the methods, the parameters of the parallel systems: the type of the processor connection topology in the communication network and the time machine -dependent constants. The dynamic characteristics of the methods are obtained, the character of the time behavior of the parallel implementation of the algorithm, the general overhead costs for parallelism, the speedup, the efficiency of parallel equipment, the estimate for the maximum value of the number of processors used are obtained. The scientific novelty of the proposed method for assessing the quality and scalability of parallel numerical methods is to improve the efficiency of solving the Cauchy problem by selecting the optimal algorithm/architecture combination. The application of the isoefficient analysis made it possible to translate the quality and scalability studies of parallel algorithms by varying the numerous dynamic parameters of the problem, method, and parallel system from the region of the reusable experiment to the analytic domain. The practical significance of scaling is to construct optimal parallel algorithms for given implementation conditions, as well as to formulate priority use areas for different algorithms for solving the same problem.

Key words: scalability, function isoefficiency, total overhead, Cauchy's problem.

REFERENCES

1. Gergel, V. P. (2007), *Theory and practice of parallel computing [Teoriya i praktika parallel'nykh vychisleniy]*, INTUIT. Binom. Laboratory of Knowledge, Moscow, 423p.
2. Grama, A., Gupta, A., Kumar, V., Karypis, G. (2003), *Introduction to Parallel Computing*, Addison Wesley, 856p.
3. Feldman, L. P., Nazarova, I. A. (2011), *Parallel one-step methods for the numerical solution of the Cauchy's problem [Paralel'ni odnokrokovyi metodi chisel'nogo rozv'yazannya zadachi Koshi]*, Donetsk National Technical University, Donetsk, 185p.
4. Feldman, L. P., Nazarova, I. A. (2013), *Modern parallel methods for the numerical solution of the Cauchy's problem [Sovremennye parallel'nye metody chislennogo resheniya zadachi Koshi]*, Donetsk National Technical University, Donetsk, 206p.
5. Nazarova, I. A., Chorna, A. O. (2015), *Research scalability of parallel systems based on function of izoefficiency [Doslidzhennya masshtabovanosti paralel'nykh sistem na osnovi funktsii izoefektivnosti]*, Naukovi pratsi DonNTU: Informatyka, Kybernetyka ta obchysliuvalna tekhnika, №2 (21), Donetsk National Technical University, Krasnoarmiysk, P. 56-62.
6. Nazarova, I. A. (2009), *Analysis of the scalability of parallel algorithms for the numerical solution of the Cauchy problem [Analiz masshtabiruemosti parallel'nykh algoritmov chislennogo resheniya zadachi Koshi]*, Naukovi pratsi DonNTU: Informatyka, Kybernetyka ta obchysliuvalna tekhnika, № 10, Donetsk, P. 21-26.