

УДК 519.612

О. И. Овчаренко

Таганрогский институт управления и экономики, Россия
347900, г. Таганрог, ул. Петровская, 45

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА ЦИКЛИЧЕСКОЙ РЕДУКЦИИ НА НРС-КЛАСТЕРЕ

O. I. Ovcharenko

Taganrog Management and Economics Institute, Russia
347900, 45, Petrovskaya Street, Taganrog

PARALLEL IMPLEMENTATION OF THE CYCLIC REDUCTION HPC-CLUSTER

O. I. Овчаренко

Таганрозький інститут управління та економіки, Росія
347900, м. Таганрог, вул. Петровського, 45

ПАРАЛЛЕЛЬНА РЕАЛІЗАЦІЯ МЕТОДУ ЦИКЛІЧНОЇ РЕДУКЦІЇ НА НРС-КЛАСТЕР

В статье представлена модель для оценки эффективности алгоритмов при реализации на многоядерных вычислительных системах. Получение оценок эффективности и ускорения параллельного алгоритма рассматриваются на примере решения систем с трехдиагональной матрицей методом циклической редукции. При оценке учитываются аппаратные характеристики НРС-кластера, который представляет собой многопроцессорную вычислительную систему.

Ключевые слова: multiprocessor computing systems, cluster, parallel algorithm.

The article presents a model for evaluating the effectiveness of the implementation of algorithms on multicore computing systems. Getting performance evaluations and accelerate parallel algorithm considers the example of solving systems with tridiagonal matrix by cyclic reduction. When evaluating the hardware specifications of the HPC-cluster, which is a multiprocessor computer system.

Key words: Fast direct methods, parallel algorithms.

У статті представлена модель для оцінки ефективності алгоритмів при реалізації на багато-ядерних обчислювальних системах. Отримання оцінок ефективності і прискорення паралельного алгоритму розглядаються на прикладі рішення систем з трьохдіагональною матрицею методом циклічної редукції. При оцінці враховуються апаратні характеристики НРС-кластера, який є багатопроцесорною обчислювальною системою.

Ключові слова: багатопроцесорні обчислювальні системи, кластер, паралельний алгоритм.

Использование вычислительного потенциала многоядерных процессоров в настоящее время является важнейшей составляющей инновационного развития страны. Но, чтобы в полной мере воспользоваться этим вычислительным потенциалом, необходима разработка соответствующего программного обеспечения.

При реализации прямых методов на отдельных этапах возникает потребность в решении систем линейных алгебраических уравнений (СЛАУ) с трехдиагональной матрицей, для решения которых существует достаточно много методов. В данной работе выбор остановлен на часто используемом на практике методе циклической редукции (cyclic reduction (CR)) [1]. Данный метод является экономичным, т.к. обладает вычислительной сложностью порядка $O(N)$.

В качестве исходных данных при оценке эффективности построенного параллельного алгоритма циклической редукции [2] использовались параметры, характеризующие выбранную задачу, а также аппаратные характеристики НРС-кластера (ЮФУ, г. Таганрог), который представляет собой многопроцессорную вычислительную систему (МВС) с распределенной памятью, состоящую из 128 вычислительных узлов, объединенных сетью InfiniBand, с пропускной способностью 20 Гбит/с. В качестве вычислительных узлов используется 128 однотипных 16-ядерных Blade-серверов HP ProLiant BL685c, каждый из которых оснащен четырьмя 4-ядерными процессорами AMD Opteron 8356 2.3GHz и объемом ОЗУ 32 Гб [3]. Параллельные расчеты произведены с применением технологий MPI на МВС и с учетом параметров, полученных при тестировании данной системы пакетами Linpack и Pallas MPI benchmark.

В качестве величин, характеризующих «распараллеливаемость» алгоритмов, использовались коэффициенты ускорения $S_p = T_1/T_p$ и эффективности алгоритма $E_p = S_p/p$. Величина T_1 характеризует время выполнения последовательного алгоритма, а T_p – время выполнения алгоритма с использованием p -вычислительных ядер.

При построении модели для оценки эффективности построенного алгоритма, учитывалось следующее.

1. Время выполнения последовательного алгоритма T_1 складывается из времени необходимого для выполнения вычислений в одном процессорном ядре, а также времени, которое тратится на чтение данных, необходимых для вычислений, из оперативной памяти в кэш-память.

2. Для оценки времени выполнения параллельного алгоритма T_p также учитывались накладные расходы на организацию параллелизма (организация и закрытие параллельных секций) и необходимость обмена числами (данными) между узлами кластера. Далее под числом будем понимать вещественную переменную со стандартной двойной точностью в 64-битовом представлении с плавающей запятой, что является необходимым при решении, например, СЛАУ большой размерности.

3. Поскольку для передачи данных могут использоваться как механизмы синхронных (блокирующих), так и асинхронных (неблокирующих) передач, то в дальнейшем при оценке операций приема-передачи данных будем рассматривать операции блокирующего типа, которые являются более надежными и не позволяют совместить процессы передачи данных и вычисления.

4. Передача данных от одного процессора к остальным может производиться одновременно (в стандарте MPI предусмотрена процедура рассылки от одного процессора всем имеющимся процессорам Broadcast).

5. Процессор может принимать информацию от других процессоров последовательно (в стандарте MPI предусмотрена процедура сборки в одном процессоре локальных значений из других процессоров Reduce).

6. Прием и передача информации между процессорами одновременно не допускается. Данная процедура выполняется за два последовательных шага: на первом шаге процессоры с нечетными номерами отправляют данные, а процессоры с четными номерами осуществляют прием этих данных. На втором шаге роли процессоров меняются – четные выполняют передачу, нечетные – прием. Для поддержки данной процедуры в стандарте MPI предусмотрена процедура Sendrecv [4].

Для обозначения алгоритмов введем индекс, характеризующий способ реализации: (1) – последовательный, (p) – параллельный.

Выполним оценку времени выполнения последовательного алгоритма метода циклической редукции, которая складывается из времени вычислений и времени доступа к оперативной памяти:

$$T_{CR}^l = T_{ap}^l + T_{on}^l, \quad (1)$$

где T_{ap}^l – время выполнения арифметических операций;

T_{on}^l – время доступа к памяти.

Общее время выполнения вычислений складывается из времени вычисления коэффициентов на прямом ходе и времени вычисления неизвестных на обратном ходе и определяется следующим образом:

$$\begin{aligned} T_{ap}^l &= T_{np}^l + T_{об}^l = 12t_{ap} \sum_{l=1}^{\log_2 N-1} \left(\frac{N-2^l}{2^l} \right) + 5t_{ap} \sum_{l=0}^{\log_2 N-1} \left(\frac{N}{2^l 2} \right) = \\ &= 12 t_{ap} (N - \log_2 N - 1) + 5 t_{ap} (N - 1), \end{aligned} \quad (2),$$

где t_{ap} – среднее время выполнения базовой арифметической операции;

N – размерность задачи.

Время доступа к оперативной памяти T_{on} представим как результат деления объема извлекаемых данных на пропускную способность памяти β [4]. Также для получения более точной оценки необходимо учитывать параметры конкретной вычислительной системы, а именно: размер кэш-строки (64 байта); латентность доступа к памяти α и частоту возникновения кэш промахов γ ($0 \leq \gamma \leq 1$):

$$T_{on}^l = (9N - 4 \log_2 N - 9) \cdot \gamma (\alpha + 64/\beta). \quad (3).$$

Тогда время выполнения последовательного алгоритма будет получено суммированием выражений (2) и (3):

$$T_{CR}^l = 16 t_{ap} (N - \log_2 N - 1) + 6 t_{ap} (N - 1) + (9N - 4 \log_2 N - 9) \cdot \gamma (\alpha + 64/\beta). \quad (4).$$

Для параллельного алгоритма метода циклической редукции, время выполнения составит:

$$T_{CR}^p = T_{ap}^p + T_{on}^p + T_{обм} + \tau, \quad (5),$$

где T_{ap}^p – время выполнения арифметических операций на p -вычислительных элементах;

T_{on}^p – время доступа к оперативной памяти;

$T_{обм}$ – время выполнения операций обмена информацией между узлами;
 τ – время, необходимое на организацию и закрытие параллельных секций.
 Время выполнения арифметических операций определяется по формуле:

$$\begin{aligned}
T^p_{ap} &= T^p_{np} + T^p_{o\delta} = 12t_{ap} \sum_{l=1}^{\log_2 N-1} \left[\frac{N}{2^l p} \right] + 5t_{ap} \sum_{l=0}^{\log_2 N-1} \left[\frac{N}{2^l 2p} \right] = \\
&= 12 t_{ap}(N/p - \log_2 N - 1) + 5 t_{ap}(N/p - 1), \tag{6}
\end{aligned}$$

где p – общее количество вычислительных ядер; квадратные скобки в формуле означают операцию округления до целого, не превосходящее данное.

Величина T^p_{on} составит:

$$T_{on}^p = (9N - 4 \log_2(N/p) - 9) \cdot \gamma(\alpha + 64/\beta). \quad (3).$$

Чтобы оценить время выполнения операций обмена информацией, обрабатываемой в различных узлах вычислительной системы, введем следующие обозначения: k_u – количество узлов; k_p – количество однотипных ядер в узле. Тогда время на обмен между узлами определяется следующим образом:

$$T_{об.м} = 9 \log_2(N-1) (\alpha_{inf} + 1/\beta_{inf}) \quad \text{при } N/k_u = k_p;$$

$$T_{обм} = (8(\log_2 N - l) + \log_2 p) (\alpha_{inf} + l/\beta_{inf}) \quad \text{при } N/k_u > k_p,$$

где α_{inf} – латентность (*latency*), определяющая время подготовки к передаче данных по сети;

β_{inf} – пропускная способность сети (*bandwidth*), задающая объем передаваемых по сети за 1 секунду данных.

На рис. 1 представлена схема информационных потоков, которыми необходимо обмениваться узлам вычислительной системы на прямом и обратном ходах выполнения алгоритма.

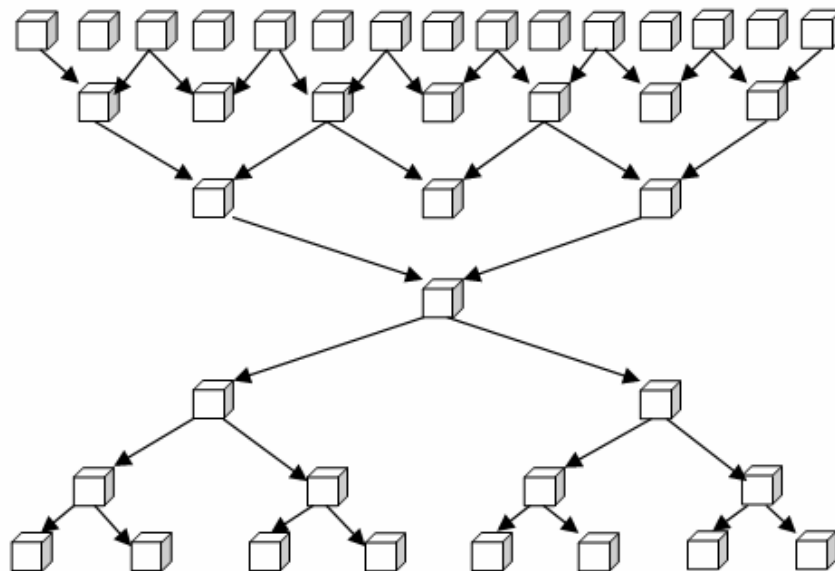


Рисунок 1 – Структура информационных связей циклической редукции

Для оценки качества параллельного алгоритма используются характеристики, введенные ранее. График изменения коэффициента ускорения S_p в зависимости от размерности задачи N и числа вычислительных элементов (ядер) p представлен на рис. 2.

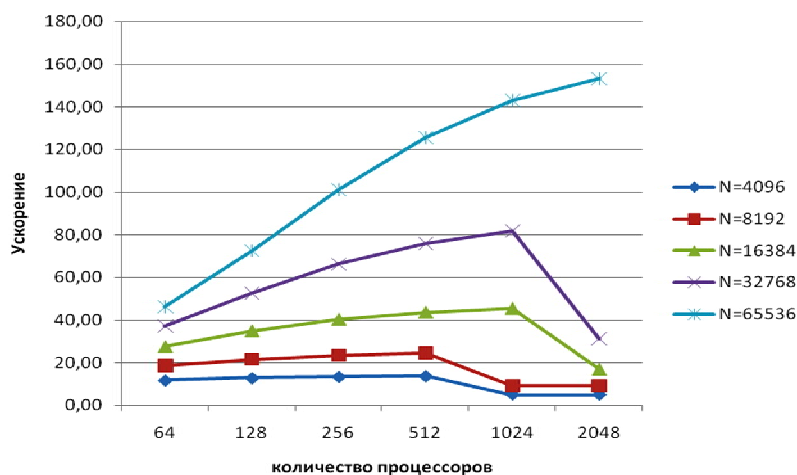


Рисунок 2 – Графики изменения величины ускорения S_p в зависимости от N и p

На рис. 3 представлены графики изменения величины E_p , характеризующей среднюю долю времени выполнения алгоритма, в течение которой процессоры реально используются для решения задачи, в зависимости от размерности задачи N и числа вычислительных элементов (ядер) p .

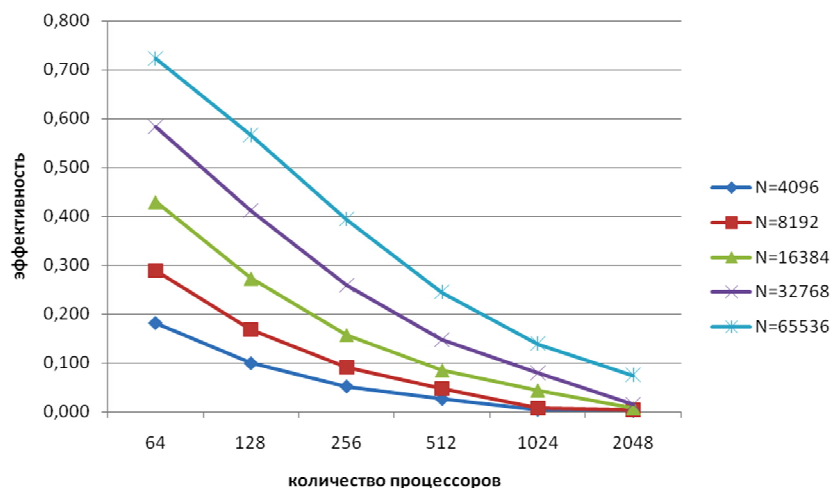


Рисунок 3 – Графики изменения величины эффективности E_p в зависимости от N и p

Анализ представленных графиков показывает, что при небольших значениях N , ускорение растет до определенного момента. Далее при увеличении числа вычислительных элементов (ядер) ускорение начинает падать. Это говорит о том, что по сравнению с объемом вычислений, приходящихся на одно ядро, время на обмены между узлами существенно увеличивается. Аналогично обстоит дело с эффективностью алгоритма (рис. 3). Если считать приемлемыми значениями эффективности величины, попадающие в диапазон от 1 до 0,5, то совершенно очевидно, что использование алгоритма эффективно только при решении задач большой размерности с использованием количества вычислительных элементов не более 128 (для рассматриваемой системы это 8 узлов по 16 ядер в каждом).

Выводы

Построен параллельный алгоритм метода циклической редукции для решения СЛАУ с трехдиагональной матрицей, потребность в решении которых возникает при реализации отдельных этапов быстрых прямых методов. Получены оценки эффективности и ускорения параллельного алгоритма, учитывающие аппаратные характеристики НРС-кластера, который представляет собой многопроцессорную вычислительную систему. Полученные оценки позволяют выбрать наиболее оптимальное количество вычислительных элементов в зависимости от размерности задачи.

Список литературы

1. Самарский А. А. Методы решения сеточных уравнений / А. А. Самарский, Е. С. Николаев. – М. : Наука, 1978. – 592 с.
2. Овчаренко О. И. Параллельная реализация быстрых прямых методов решения сеточных эллиптических уравнений / О. И. Овчаренко // Известия ЮФУ. Технические науки. – Таганрог : Изд-во ЮФУ, 2008. – № 9(86).
3. Многопроцессорная вычислительная система ТТИ ЮФУ: материалы Всероссийской суперкомпьютерной конференции «Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность». / Лапин Д.В., Сухинов А.И., Целых А.Н. и др. – М. : Изд-во МГУ, 2009.
4. Гергель В.П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем : учебное пособие / Гергель В.П. – Нижний Новгород : Изд-во ННГУ им. Н.И.Лобачевского, 2010.

References

1. Samarsky A.A., Nikolayev E.S. Methods for solving difference equations. M.: Nauka, 1978 - 592s.
2. Ovcharenko O.I. Parallel implementation of fast direct methods for solving grid elliptic equations // News of the SFU. Technical science. Taganrog: Publishing house of the SFU, 2008. №9 (86).
3. Lapin D.V., Sukhinov A.I. and others. Multiprocessor system TTI SFU: Proceedings of supercomputing conference "Scientific service in the Internet: scalability, parallelism, efficiency." M.: MGU, 2009.
4. Gergel V.P. High-performance computing for multi-core multi-processor systems: the manual. Nizhny Novgorod: Publishing House of the NNSU N.I.Lobachevsky, 2010.

RESUME

O.I.Ovcharenko

Parallel implementation of the method of cyclic reduction in the HPC-cluster

Background: This article is designed parallel algorithm cyclic reduction method for solving systems of linear algebraic equations with a tridiagonal matrix.

Materials and methods: Estimations of efficiency and acceleration characterize the quality of development of parallel algorithms. When evaluating the hardware specifications of the HPC accounted cluster, which is a multiprocessor computer systems [3].

Results: These estimates allow us to choose the most optimal number of computing elements, depending on the dimension of the problem.

Conclusion: Parallel algorithm of cyclic can be used in the implementation of individual stages of fast direct methods [2] relating to the decision SLAE with tridiagonal matrix.

Статья поступила в редакцию 22.11.2015.