ТРУДЫ КОНФЕРЕНЦИИ МОЛОДЫХ УЧЕНЫХ
Новосибирск, 2014 С. 122–133

Средства задания прямого управления во фрагментированных программах и их применение на примере явного метода решения уравнения Пуассона

А. А. Ткачёва

УДК 004.272.2; 004.45

Рассмотрена проблема эффективного выполнения фрагментированной программы (ФП) в системе фрагментрованного программирования LuNA. Для повышения производительности выполнения ФП были разработаны средства задания прямого управления в виде предикатной сети Петри. Реализован модуль прямого управления RuSh. Представлено исследование производительности модуля RuSh на задаче решения уравнения Пуассона явным методом. Результаты сравнивались с реализацией выполнения ФП в системе LuNA и выполнением ФП в системе LuNA совместно с RuSh.

The problem of efficient execution of fragmented program (FP) under the fragmented parallel programming system LuNA is considered. To achieve good performance of FP execution the control flow means was designed in the form of predicate Petri nets. The control flow module RuSh was developed. Evaluation of RuSh performance was examine on the s solution of Poisson equation by explicit method. Obtained results were compared with FP execution under system LuNA and FP execution under system LuNA together with Rush.

Ключевые слова: параллельное программирование, системные алгоритмы параллельной обработки данных, предикатная сеть Петри.

Keywords: parallel programming, system algorithms of parallel processing data, predicate/transition nets.

Введение. Развитие параллельных вычислительных технологий позволяет решать задачи численного моделирования, которые раньше невозможно было реализовать на последовательных компьютерах. Однако возможности параллельных вычислений непосредственно связаны со сложностями параллельного программирования.

Для высокой производительности прикладная программа должна обладать следующими динамическими свойствами: настройка на имеющиеся ресурсы вычислителя, динамическая балансировка загрузки процессоров, осуществление коммуникаций на фоне счета и т. д. Сейчас в большинстве случаев эти свойства реализуются вручную с помощью MPI, но делаются попытки реализации систем программирования и библиотек, в которых динамические свойства обеспечиваются автоматически, например LuNA, Charm++, SMP Superscalar, PLASMA.

Фрагментированное программирование — технология параллельного программирования, предназначенная для реализации больших численных задач на суперкомпьютере. В ней фрагментированная программа (ФП) представляется декларативно в виде множества фрагментов данных (ФД) и фрагментов вычислений (ФВ), и такая фрагментированная структура ФП сохраняется в ходе вычислений, что позволяет автоматически обеспечивать динамические свойства программы. Такой подход позволяет программировать на более высоком уровне и не заботиться о сложностях системного параллельного программирования. При этом описание фрагментированного алгоритма (ФА) является платформенно независимым, а настройку на конкретный вычислитель обеспечивает runtime-система системы фрагментированного программирования. В настоящее время в лаборатории синтеза параллельных программ ИВМиМГ СО РАН разрабатывается система фрагментированного программирования LuNA (Language for Numerical Algorithm) [1, 2].

Runtime-системе LuNA необходимо одновременно выполнять множество функций, таких как задание динамического распределения ресурсов, определение ФВ, готовых к исполнению, и назначение на исполнение одного из них, нескольких, всех или ни одного. Из-за этого на управление вычислениями система LuNA тратит много времени, в зависимости от задачи это может привести к стократному ухудшению времени исполнения ФП по сравнению с ручной реализацией с использованием MPI.

Чтобы снизить накладные расходы, целесообразно определить статически прямое управление там, где это возможно, оставив некоторую необходимую степень недетерминизма выполнения ФП для возможности параллельного исполнения на мультикомпьютере и обеспечения динамических свойств выполнения программы.

Для этих целей был разработан модуль прямого управления RuSh (Runtime Shell), являющийся компонентом системы LuNA. Его задачей является параллельное выполнение некоторого подмножества ФВ, входящего в ФП, под прямым управлением без необходимости дополнительных проверок ФВ на готовность к исполнению.

В работе представлено исследование производительности модуля RuSh на задаче решения уравнения Пуассона явным методом при трехмерной фрагментации области. Результаты сравнивались с реализацией выполнения ФП в системе LuNA и выполнением ФП в системе LuNA совместно с RuSh.

1. Постановка задачи. Задачи численного моделирования отличаются регулярной структурой: массовость данных и операций (одни и те же алгоритм обработки ФД, размер ФД, примерно совпадающий объем вычислений при обработке). Использование этого свойства позволяет уменьшить сложность задачи управления, так как на множестве однородных по количеству вычислений ФВ алгоритма одно решение по управлению может быть использовано для всех ФВ. Такой подход позволяет разбить большую и трудоемкую задачу управления вычислениями во ФП на множество более простых и менее затратных подзадач.

Разрабатываемые средства задания прямого управления должны удовлетворять следующим требованиям:

— позволять задавать желаемый порядок для достаточно широкого класса численных алгоритмов; — задаваемое управление должно допускать эффективную реализацию на мультикомпьютере.

2. Подход к решению. Прямое управление [3] может задаваться различными способами:

— как отношение частичного порядка;

— циклами типа while или типа for;

— как сеть Петри.

Отношение частичного порядка хотя и является универсальным способом задания управления, однако такой способ труден для человеческого восприятия. Задание управления с помощью циклов типа while или типа for и сети Петри более наглядно.

Использование циклов типа while или типа for позволяет накладывать управление между итерациями цикла, т. е. использование специализированных циклов, все итерации которых выполняются либо независимо, либо последовательно, одна за другой. Недостат-

ком этого средства задания прямого управления является то, что если тело цикла состоит из ФВ, которые можно выполнять параллельно, эти ФВ будут выполняться все равно последовательно.

Использование в качестве средства задания прямого управления предикатной сети Петри [4] позволяет задавать управление для более общего случая, а именно задавать параллелизм выполнения ФВ и внутри тела циклов, и между итерациями циклов.

Идея использования предикатной сети Петри состоит в том, что каждому ФД соотносится соответствующее место, а ФВ — переход, предикат условия срабатывания ФВ — предикату условия срабатывания перехода. Нахождение фишки в месте означает, что соответствующий ФД вычислен и доступен для множества тех ФВ, для которых ФД является входным ФД. Если во всех входных местах некоторого перехода есть фишки, то это означает, что должен быть запущен на выполнение соответствующий ФВ.

Тогда порядок выполнения ФВ будет определяться предикатной сетью Петри, при этом, в отличие от runtime-системы LuNA необходимо проверять только наличие фишек в местах.

3. Реализация модуля прямого управления RuSh. Средства для задания управления в ФП были реализованы в виде программного модуля RuSh, который обеспечивает параллельное выполнение ФП в общей памяти, в которой порядок выполнения ФВ определяется функционированием предикатной сети Петри.

Особенности и ограничения реализации:

— в разработанном программном модуле поддерживается только задание управления в виде безопасной предикатной сети Петри (не более одной фишки в месте) — это условие ограничивает реализацию обмена ФД между ФВ одним буфером памяти;

— кроме того, можно задавать частичное распределение ресурсов, а именно задавать использование одного буфера памяти для хранения разных ФД.

4. Явный метод решения уравнения Пуассона. Уравнение Пуассона имеет вид

$$\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2} + \frac{\partial^2 F}{\partial z^2} = p,$$

При этом оно аппроксимируется следующей 7-точечной разностной схемой второго порядка:

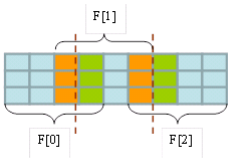

Рис. 1. Схема фрагментации данных при одномерной фрагментации области для явного метода решения уравнения Пуассона

$$\frac{F_{i+1,j,k} - 2F_{i,j,k} + F_{i-1,j,k}}{h_x^2} + \frac{F_{i,j+1,k} - 2F_{i,j,k} + F_{i,j-1,k}}{h_y^2} + \\ + \frac{F_{i,j,k+1} - 2F_{i,j,k} + F_{i,j,k-1}}{h_z^2} = p_{i,j,k}, \quad (1)$$

Явный метод решения уравнения Пуассона является итерационным, на каждой итерации происходит уточнение решения относительно начального для некоторой области моделирования. Формула вычисления значений на новой итерации получается, если выразить $F_{i,j,k}$ из схемы (1). В явном методе для вычисления сеточного значения на новой итерации все используемые в формуле значения берутся с предыдущей итерации. Итерационный процесс продолжается до сходимости $\max_{i,j,k} |F_{i,j,k}^{m+1} - F_{i,j,k}^{m}| < \epsilon$. Здесь индекс m — номер итерации.

В основе выбранного способа распараллеливания явного метода лежит принцип фрагментации области моделирования, который в данном случае эквивалентен пространственной декомпозиции области. При этом область моделирования разбивается на слои, и расчет каждого слоя на каждой итерации выполняется отдельным ФВ.

На рис. 1 приведен пример фрагментации данных для двумерного пространства моделирования при одномерной фрагментации на три фрагмента.

Так как для расчета $F_{i,j,k}$ используется 7-точечный шаблон, то для параллельной реализации решения уравнения Пуассона требуется обмен теневыми гранями, а именно обмен значениями $F_{i,j,k}$ на границах разрезания слоев между соседними слоями области моделирования.

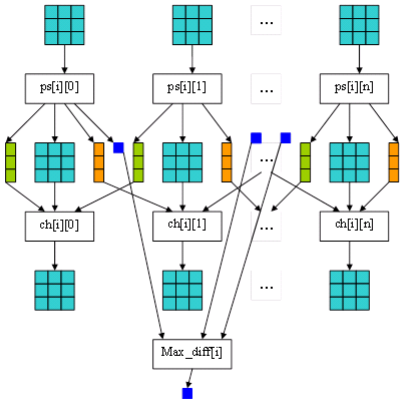

Рис. 2. Схема ФА i-й итерации явного метода решения уравнения Пуассона для двумерной области моделирования и одномерной фрагментации

Фрагментированный алгоритм i-итерации явного метода решения уравнения Пуассона изображен на рис. 2, где n — количество фрагментов, на которые разбивается область моделирования; $ps[i][j](j = 0, ..., n)$ — ФВ, обрабатывающий $j-$ слой области моделирования на i-й итерации, $in = (F[i][j])$, $out = (F'[i][j], right[i][j-1], left[i][j+1], mx[i][j])$; $ch[i][j](j = 0, ..., n$ — ФВ, вычисляющий обмен теневыми гранями для j-го слоя, $in = (F'[i][j], left[i][j], right[i][j])$, $out = (F[i+1][j])$; $Max_diff[i]$ — ФВ, вычисляющий редукцию невязки на i-й итерации, $in = (mx[i][j], j = 0..n)$, $out = (mxm[i+1])$.

Использование сети Петри для реализации ФА явного метода решения уравнения Пуассона, приведенной на рис. 3, ограничено памятью одного узла, так как программный модуль RuSh работает в общей памяти мультикомпьютера, в то время как система LuNA может работать как с общей, так и с распределенной памятью.

На рис. 3 изображена сеть Петри для i-й итерации явного метода решения уравнения Пуассона, где $pu[i][j]$, $(j = 0, ..., n)$ — переход, срабатывание требует выполнения ФВ $ps[i][j]$; $c[i][j]$, $(j = 0, ..., n)$ — переход, срабатывание которого требует выполнения ФВ $ch[i][j]$; $calc[i]$ — переход, срабатывание которого требует выполнения ФВ $Max_diff[i]$.

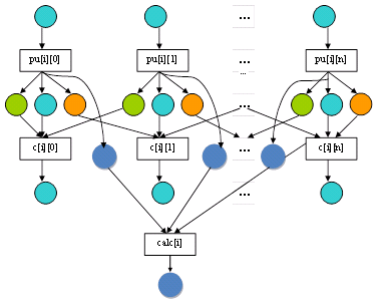

Рис. 3. Сеть Петри для i-й итерации явного метода решения уравнения Пуассона (двумерная область моделирования и одномерная фрагментация области)

Поэтому для параметров задачи, при которых данные не помещаются в память одного узла, имеет смысл использовать комбинацию базового алгоритма выполнения ФП LuNA и модуля RuSh для выполнения ФП в общей памяти.

Структурированный ФВ (СФВ) является аналогом процедуры в языках программирования, для него определяется множество входных и выходных ФД и всех ФВ, входящих в СФВ и выполняющихся в рамках одного узла вычислителя.

Все ФВ, вычисляющие j-й слой на i-й итерации ($j = 0, ..., n$) объединим в группу СФВ фиксированного размера, и внутри СФВ порядок выполнения ФВ определяется функционированием предикатной сети Петри и выполняется в RuSh, а обмен данными между СФВ осуществляется с помощью базового алгоритма выполнения ФП системы LuNA.

На рис. 4 приведена схема ФА одной итерации явного метода решения уравнения Пуассона в случае комбинированного использования алгоритмов выполнения LuNA и RuSh, где m — количество СФВ, на которые разбиваются все ФВ, вычисляющие j-й слой на i-й итерации ($j = 0, ..., n$); $L = n/m$ — количество ФВ, объединенных в рамках одного СФВ; $P[i][k]$ ($k = 0, ..., m$) — СФВ, вычисляющий несколько сгруппированных ФВ на i-й итерации. $in = (F[i][j]|j{=}k{*}L..(k{+}1) * L)$, $out = (F'[i][k * L]$, $F[i+1][j]|j{=}k * L + 1..(k+1) * L - 1$, $F'[i][(k{+}1) * L], left[i][(k{+}1){*}$ $L + 1], right[i][k * L - 1])$. В рамках СФВ порядок определяется функционированием предикатной сети Петри (см. рис. 3); $[i][k]$,

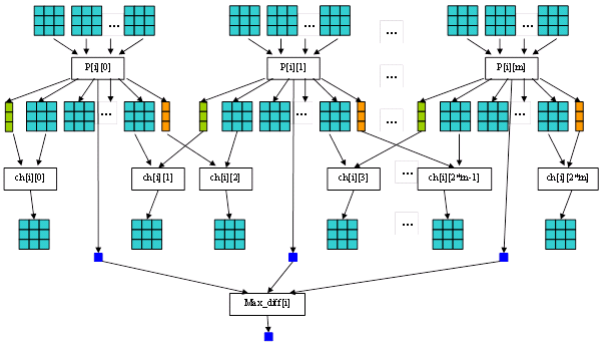

Рис. 4. Схема ФА i-й итерации явного метода решения уравнения Пуассона при использовании структурированных ФВ

$(k = 0, ..., m)$ — ФВ, вычисляющий обмен теневыми гранями для тех слоев области моделирования, которые находятся на границе. $in = (type, F'[i][j], border[i][j])out = (F[i+1][j])$.

4. Исследования производительности исполнения ФП. Для тестирования в общей памяти использовался 6-ядерный вычислитель со следующими параметрами: Intel Xeon CPU X5600 2.8Ghz. При запуске программ использовались четыре рабочих потока в LuNA и в RuSh.

В качестве прикладной задачи для исследования производительности был выбран явный метод решения уравнения Пуассона со следующими параметрами: трехмерная область моделирования, 7-точечный шаблон, трехмерная фрагментация данных.

Необходимо было провести следующие тесты.

1. Изменение накладных расходов в модуле RuSh в зависимости от количества ФВ, для которых задано прямое управления предикатной сетью Петри.

2. Сравнение времени работы распараллеленной ручной реализации прикладной задачи с использованием MPI и ФП прикладной задачи в системе LuNA в зависимости от размера задачи.

3. Сравнение производительностей следующих реализаций:

— выполнение ФП решения прикладной задачи в системе LuNA;

— выполнение ФП решения прикладной задачи с заданным прямым управление с помощью предикатной сети Петри в модуле RuSh;

Рис. 5. Зависимость удельного времени работы на вычисление одного ФВ от количества ФВ, обрабатываемых в RuSh. Размер ФД $1 \times 1 \times 1$ и 100000 итераций

— выполнение ФП решения прикладной задачи комбинированно в LuNA и RuSh, а именно: все ФВ, отвечающие за расчет i-й итерации, делятся на группы СФВ (для каждого СФВ порядок выполнения ФВ определен сетью Петри) и исполняются в RuSh; обмен данными между СФВ осуществляется с помощью LuNA.

4. Изменение времени на вычисление одного ФВ в зависимости от размера СФВ.

Тест 1. На рис. 5 представлены результаты теста 1. Заметим, что с увеличением количества ФВ, на которых задано прямое управление, время не увеличивается для достаточно большого числа ФВ.

Тест 2. На рис. 6 видно, что на достаточно большом размере прикладной задачи время работы в системе LuNA сравнимо по уровню с ручной реализацией в MPI.

Тест 3. На рис. 7 представлены результаты теста 3. Если сравнить время, затрачиваемое на выполнение ФП в системе LuNA и в модуле RuSh, то выполнение ФП в RuSh занимает намного меньше времени из-за отсутствия накладных расходов связанных с управлением. Рассмотрим теперь комбинированный случай: с одной стороны время работы увеличивается по сравнению с использованием только RuSh, так как тратится время на организацию управления вычисле-

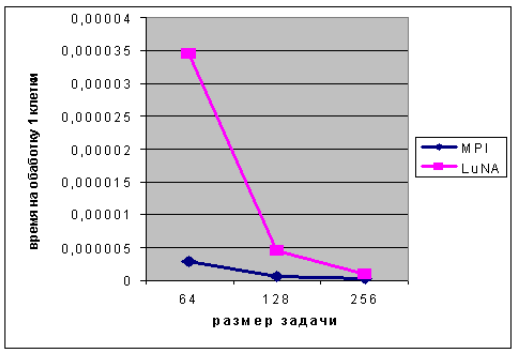

Рис. 6. Сравнение времени работы реализаций прикладной задачи в MPI и LuNA (8 процессов, количество ФД 8, итераций 10) в зависимости от размера задачи

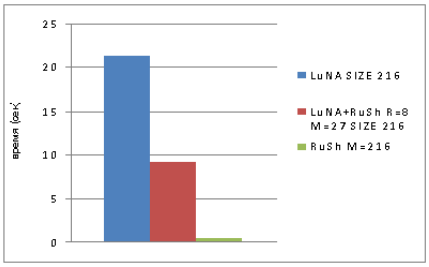

Рис. 7. Сравнение времени, затраченного на выполнение ФП с использованием LuNA, LuNA+RuSh, RuSh для степени фрагментации 216 для явного метода решения уравнения Пуассона (размер сетки: $100 \times 100 \times 100$, 40 временных итераций). SIZE — количество ФД, на которые разбивается область моделирования; M — количество ФВ внутри СФВ; R — количество СФВ, каждый такой СФВ обрабатывается на одном узле в RuSh

ниям между СФВ, с другой — даже частичное использование RuSh позволяет почти в два раза повысить производительность выполнения ФП.

Тест 4. На рис. 8 видно, что увеличение количества ФВ, обрабатываемых в рамках одного СФВ, приводит к росту производительности исполнения ФП.

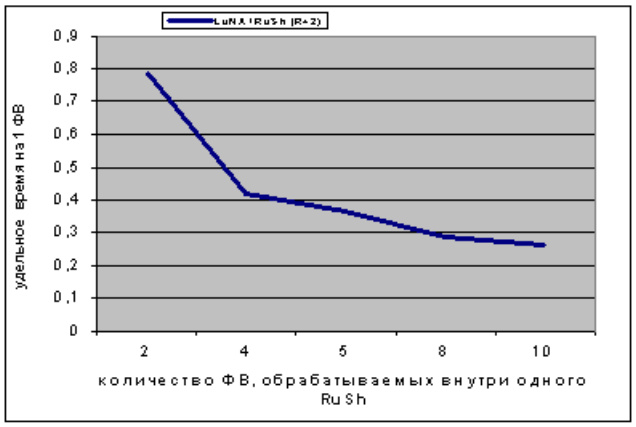

Рис. 8. Зависимость времени обработки одного ФВ от размера СФВ (одномерная фрагментация, размер ФД $20 \times 20 \times 20$, 400 итераций)

5. Результаты тестирования. Результаты тестов показали, что для задачи достаточно большого размера производительность системы LuNA сравнима по уровню с ручной реализацией в MPI. А использование модуля прямого управления RuSh для исполнения СФВ, управления в которых задано предикатной сетью Петри, повышает производительность системы LuNA в общей памяти. При этом накладные расходы модуля RuSh на управление выполнением СФВ в несколько раз меньше, чем в системе LuNA, и чем больше количество ФВ, обрабатываемых в рамках одного СФВ, тем больше выигрыш по производительности.

6. Заключение. В работе исследованы способы применения средств задания прямого управления в ФП одновременно с базовым алгоритмом выполнения ФП системы LuNA на примере решения уравнения Пуассона явным методом. Были проведены тесты производительности в общей памяти, которые позволяют говорить о том, что можно повысить производительность выполнения ФП на мультикомпьютере, используя прямое управление.

В дальнейшем предполагается провести аналогичное тестирование в распределенной памяти и исследовать применимость прямого управления для решения других задач численного моделирования.

Научный руководитель — д.т.н., проф. В. Э. Малышкин.

Список литературы

[1] Malyshkin V. E., Perepelkin V. A. LuNA fragmented programming system, main functions and peculiarities of run-time subsystem // Proc. of the 11-th conf. on parallel computing technologis LNCS 6873. Springer, 2011. P. 53–61.

[2] Kraeva M. A., Malyshkin V. E. Assembly technologyfor parallel realization of numerical models on MIMD-Multicomputers // J.l on Future Generation Computer Systems. Elsiever Sci. 2001. V. 17, N. 6. P. 755–765.

[3] Малышкин В. Э., Корнеев В. Д. Параллельное программирование мультикомпьютеров. Сер.: "Учебники НГТУ". Новосибирск: Изд-во НГТУ, 2006. 296 стр.

[4] Genrich H. G. Predicate/Transition nets // Lect. Notes in Computer Science 254. Springer-Verlag. 1987. P. 207–247

Ткачёва Анастасия Александровна —
мл. науч. сотр. Ин-та вычислительной математики
и математической геофизики СО РАН
e-mail: tkacheva@ssd.sscc.ru