



С.В. Болтанов

АНАЛИЗ ЭФФЕКТИВНОСТИ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ТИПОВОГО РЕШЕНИЯ «МЕТОД ПЕРЕМЕННЫХ НАПРАВЛЕНИЙ»

(ФГБОУ ВПО «Самарский государственный аэрокосмический университет им. академика С.П. Королева (национальный исследовательский университет)»)

В статье развивается подход, описанный в [1], позволяющий расширить и упростить применение высокопроизводительной вычислительной техники в численном моделировании. Ключевыми концепциями данного подхода являются применение типовых решений (паттернов) параллельного программирования, автоматическое распараллеливание и развертывание кода в различных программно-аппаратных архитектурах, разделение ролей системный программист – прикладной программист, использование интегрированных сред разработки (IDE) и методов визуализации кода на основе технологий XML.

Рассмотрим, каким образом может быть осуществлено распараллеливание численных моделей, основанных на методе переменных направлений. Пример также иллюстрирует технологию разработки произвольных последовательно-параллельных программ в нотации Templet [1].

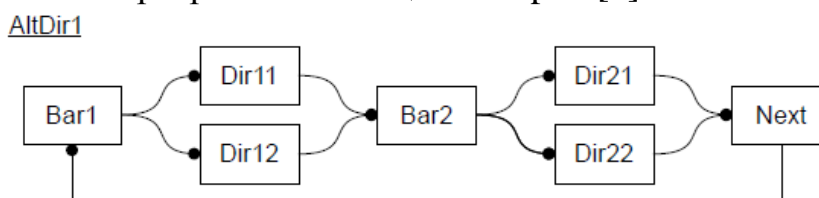


Рис. 1. Композиция процессов в методе переменных направлений с функциональной декомпозицией

Структура процессов для функциональной декомпозиции показана на рисунке 1. Процессы слева Dir11 и Dir12 выполняют вычисления горизонтальных независимых прогонок на $\frac{1}{2}$ шага, процессы справа Dir21 и Dir22 выполняют вычисления вертикальных независимых прогонок при окончании шага (или наоборот: левые процессы – вертикальные, правые – горизонтальные прогонок). Реализация барьера по окончании $\frac{1}{2}$ шага изображена на рисунке 2 (б). Метод bar отправляет сообщения, если закончились все входящие параллельные ветви. Контроль корректности последовательно-параллельной структуры реализуется при помощи ответных сообщений get в каналах Link на рисунке 2 (а).

В программной реализации по модели данные сообщения не обязательны. Другие два барьерных процесса устроены аналогично процессу рисунка 2 (б) [2]. Автором рассматривалась параллельная реализация численного метода, описанного в работе [3].

Для решения задачи нелинейной динамики использовалась схема ADI [2], реализующая метод переменных направлений в общей памяти. В код обычной



схемы ADI были добавлены директивы компилятора OpenMP (Open Multi-Processing), которые предназначены для программирования многопоточных приложений на многопроцессорных системах с общей памятью.

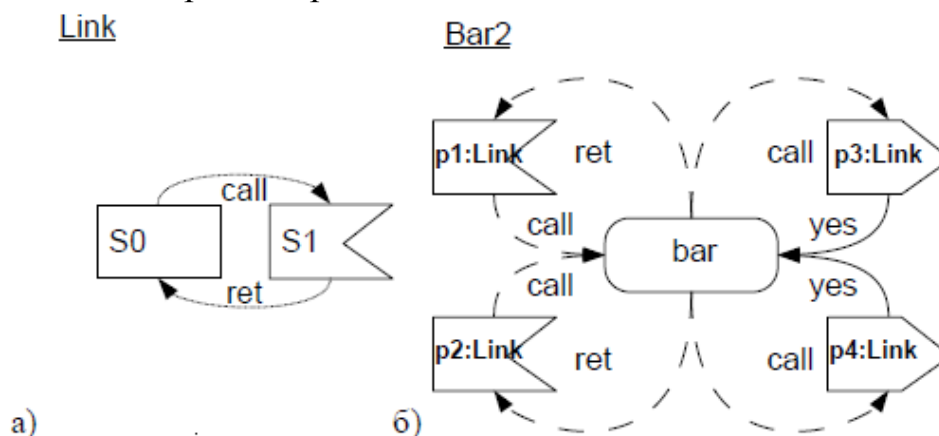


Рис. 2. Описание протокола канала (а) и барьера (б) в методе переменных направлений с функциональной декомпозицией

Использование директив компилятора OpenMP позволило выполнять процессы слева Dir11 и Dir12, вычисляющие горизонтальные независимые прогонки на $\frac{1}{2}$ шага, и процессы справа Dir21 и Dir22, вычисляющие вертикальные независимые прогонки при окончании шага, параллельно. За счет этого время работы программы значительно сократилось.

Был разработан второй вариант схемы ADI. Во-первых, описание всех переменных, констант и методов были вынесены в один класс. Во-вторых, каждому методу добавлен модификатор inline, т.е. тело метода подставляется в каждую точку вызова, вместо того, чтобы генерировать код вызова. Причина использования inline-методов заключается в их эффективности. Всякий раз, когда вызывается метод, необходимо выполнить серию инструкций для формирования вызова метода, вставки аргументов в стек и возврата значения. В некоторых случаях для этого приходится использовать много тактов центрального процессора.

В экспериментах на многоядерной настольной системе решалась задача на области размером 500×500 по пространству и 100 отсчетов по времени. Использовался компьютер Samsung NP300E5A (Intel Core i5-2450M 2.50GHz) с установленной ОС Windows 7 Home Basic 64 bits.

Результаты эксперимента сведены в таблицу 1.

Таблица 1. Время вычислений измененной схемы ADI

| Время вычислений (вариант 1), сек. | Время вычислений (вариант 2), сек. |
|---------------------------------------|---------------------------------------|
| 13.060 | 12.673 |
| 12.675 | 12.286 |
| 12.272 | 12.716 |
| 12.441 | 12.551 |
| 12.352 | 12.629 |



| | |
|--------|--------|
| 12.500 | 12.790 |
| 12.276 | 12.877 |
| 12.705 | 13.002 |
| 12.468 | 12.912 |
| 12.338 | 12.766 |

Из таблицы 1 видно, что среднее время вычисления в первом и втором вариантах схемы ADI примерно равно. Результаты этого эксперимента можно сравнить с результатом вычисления по обычной схеме ADI без добавлений директив компилятора OpenMP, модификаторов Inline и т.п.

Таблица 2. Время вычисления обычной схемы ADI

| Время вычисления, сек. |
|------------------------|
| 29.694 |
| 29.344 |
| 29.299 |
| 29.134 |
| 29.499 |
| 30.306 |
| 29.716 |
| 29.923 |
| 29.507 |
| 28.856 |

Таким образом, применение директив компилятора OpenMP ускорило вычисления более чем в 2 раза: при применении типовой и измененной схемы ADI эффективность распараллеливания остается достаточно высокой. Однако использование типовой схемы обладает некоторыми преимуществами. Во-первых, вносится наглядность и удобство для разработчика численных методов. Во-вторых, появляется возможность скрыть от пользователя метод распараллеливания. Код системы Graphplus templet зарегистрирован Федеральным институтом промышленной собственности и открыт для некоммерческого применения (<http://graphplus.ssau.ru>).

Литература

1. Востокин С.В. Визуальное моделирование в разработке параллельных алгоритмов. Метод и программные средства. LAMBERT Academic Publishing, 2011. 304 с.
2. Востокин С.В., Хайрутдинов А.Р., Литвинов В.Г. Программный комплекс параллельного программирования Graphplus templet // Вестник СамГТУ: Сер. Физико-математические науки, 2011. – № 4 (25). – с. 146 – 153
3. Курушина С.Е. Аналитическое исследование и численное моделирование контрастных диссипативных структур в поле флуктуаций динамических переменных // Изв. вузов: Прикладная нелинейная динамика, 2009. – № 6. – с. 125 – 138