

Инкрементальное распараллеливание программ с использованием DVM-системы

В.А. Бахтин^{1,2}, Д.А. Захаров¹, В.А. Крюков^{1,2}, Н.В. Поддерюгина¹, М.Н. Притула¹
ИПМ им. М.В. Келдыша РАН¹, МГУ им. М.В. Ломоносова²

DVM-система предназначена для разработки параллельных программ научно-технических расчетов на языках C-DVMH и Fortran-DVMH. Эти языки используют единую модель параллельного программирования (DVMH-модель) и являются расширением стандартных языков Си и Фортран спецификациями параллелизма, оформленными в виде директив компилятору. Получаемые DVMH-программы без каких-либо изменений могут эффективно выполняться на кластерах различной архитектуры, использующих многоядерные универсальные процессоры (далее ЦПУ), графические ускорители (далее ГПУ) и сопроцессоры Intel Xeon Phi. Это достигается за счет различных оптимизаций, которые выполняются как статически, при компиляции DVMH-программ, так и динамически. Параллельные программы могут настраиваться при запуске на выделенные для их выполнения ресурсы - количество узлов кластера, ядер, ускорителей и их производительность [1].

Данные оптимизации позволяют добиться высокой эффективности выполнения различных тестовых программ (например, программ из пакета NAS NPB [2-3]) и реальных приложений (получены результаты при использовании более 1000 ГПУ [4]).

Распараллеливание программы в модели DVMH можно разделить на следующие этапы:

- а) Распределение данных (массивов) и вычислений (параллельных циклов).
- б) Определение и спецификация удаленных данных (данные, которые вычисляются на одном процессоре, а используются на других).
- в) Определение вычислительных регионов (последовательности операторов, циклов) для выполнения на ускорителях.
- г) Управление перемещением данных между памятью ЦПУ и памятью ускорителей.

Основная сложность разработки параллельной программы для кластера - необходимость принятия глобальных решений по распределению данных и вычислений с учетом свойств всей программы, а затем выполнения кропотливой работы по модификации программы и ее отладке. Большой объем программного кода, многомодульность, многофункциональность затрудняет принятие решений по согласованному распределению данных и вычислений.

Для решения данной проблемы может использоваться метод инкрементального, или частичного распараллеливания. Идея этого метода заключается в том, что распараллеливанию подвергается не вся программа целиком, а ее части (области распараллеливания) - в них заводятся дополнительные экземпляры требуемых данных, производится распределение этих данных и соответствующих вычислений. Данные области могут быть построены на основе времен, полученных с помощью анализатора производительности, который входит в состав DVM-системы.

Для взаимодействия с теми частями программы, которые не подвергались распараллеливанию, используются операции копирования исходных (нераспределенных) данных в дополнительные (распределенные) данные и обратно. Конечно, операции копирования могут снизить или вообще ликвидировать эффект от распараллеливания. Кроме того, до полного распараллеливания (пока не все массивы программы будут распределены) программе будет требоваться память и для распределенных, и для нераспределенных массивов, что может ограничивать размер решаемой задачи.

К достоинствам инкрементального распараллеливания можно отнести:

- а) Возможность распараллелить не всю программу, а ее времяемкие фрагменты, упрощает работу программиста, так как существенно сокращается объем кода программы для анализа и распараллеливания.
- б) Отказ от распараллеливания сложных фрагментов программы позволяет с большей вероятностью найти хорошие решения для выделенных областей распараллеливания.

в) Найденные решения могут быть использованы в качестве подсказки при распараллеливании других частей программы на следующих этапах.

г) Использование DVMH-модели позволяет быстро проверить эффективность различных вариантов распределения данных, т.к. переход от одного варианта распределения данных на другой, как правило, осуществляется за счет изменения нескольких DVM-указаний и не требует сложной модификации кода программы.

В последних версиях языков C-DVMH [5] и Fortran-DVMH [6] введено понятие нераспределенного параллельного цикла, для которого нет необходимости задавать отображение на распределенный массив. Например, трехмерный параллельный цикл может выглядеть так (рис. 1):

```
#pragma dvm parallel(3) reduction (max(eps))
for (int i = L1; i <= H1; i++)
    for (int j = L2; j <= H2; j++)
        for (int k = L3; k <= H3; k++)
            ...
!DVM$ PARALLEL(I,J,K) REDUCTION (MAX(EPS))
DO I = L1, H1
    DO J = L2, H2
        DO K = L3, H3
```

Рис. 1. Новый режим распределения вычислений

Наличие такой возможности существенно упрощает процесс разработки параллельных программ. До принятия решения о распределении данных, у программиста появляется возможность оценить перспективы ускорения целевой программы на многоядерных процессорах и ускорителях.

На постере представлены этапы инкрементального распараллеливания тестовой программы для решения уравнения теплопроводности, а также программного комплекса для численного исследования динамики потоков плазмы в каналах квазистационарных плазменных ускорителей [7]. Приводятся характеристики эффективности параллельного выполнения получаемых версий DVMH-программ. Оцениваются потери, возникающие на различных этапах инкрементального распараллеливания. С использованием инкрементального распараллеливания разработана DVMH-версия программного комплекса для численного исследования динамики потоков плазмы, которая ускоряется до 24 раз на 32 процессорах вычислительного комплекса К-100 (ИПМ РАН) и до 22 раз на 28 процессорах вычислительной системы MVS1P5 (МСЦ РАН).

Литература

1. Бахтин В.А., Колганов А.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н. Методы динамической настройки DVMH-программ на кластеры с ускорителями. Суперкомпьютерные дни в России: Труды международной конференции (28-29 сентября 2015 г., г. Москва), М.: Изд-во МГУ, 2015, С. 257-268.
2. Алексахин В.Ф., Бахтин В.А., Жукова О.Ф., Колганов А.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Савицкая О.А., Шуберт А.В. Распараллеливание на графические процессоры тестов NAS NPВ 3.3.1 на языке Fortran DVMH // Вестник Уфимского государственного авиационного технического университета. 2015. Т. 19, №1(67). С. 240-250.
3. Алексахин В.Ф., Бахтин В.А., Жукова О.Ф., Колганов А.С., Крюков В.А., Островская И.П., Поддерюгина Н.В., Притула М.Н., Савицкая О.А. Распараллеливание на языке Fortran-DVMH для сопроцессора Intel Xeon Phi тестов NAS NPВ3.3.1. Сборник трудов Международной научной конференции «Параллельные вычислительные технологии 2015», Челябинск: Издательский центр ЮУрГУ, 2015, С. 19-30.
4. Бахтин В.А., Клинов М.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н. Использование языка Fortran DVMH для решения задач гидродинамики на высокопроизводительных

гибридных вычислительных системах // Вестник Южно-Уральского государственного университета, серия «Вычислительная математика и информатика», Челябинск: Издательский центр ЮУрГУ, 2013, Том 2, Выпуск 3, С. 106-120.

5. Язык C-DVMH. C-DVMH компилятор. Компиляция, выполнение и отладка CDVMH-программ. URL: http://dvm-system.org/static_data/docs/CDVMH-reference-ru.pdf (дата обращения: 14.04.2018).
6. Язык Fortran-DVMH. Fortran-DVMH компилятор. Компиляция, выполнение и отладка DVMH-программ. URL: http://dvm-system.org/static_data/docs/FDVMH-user-guide-ru.pdf (дата обращения: 14.04.2018).
7. Kozlov A. N., Kononov V. S. Numerical study of the ionization process and radiation transport in the channel of plasma accelerator // Communications in Nonlinear Science and Numerical Simulation. 2017. Vol. 51. P. 169-179.