ДИАЛОГ С ПРОГРАММИСТОМ В СИСТЕМЕ АВТОМАТИЗАЦИИ РАСПАРАЛЛЕЛИВАНИЯ САПФОР

В.А. Бахтин, И.Г. Бородич, Н.А. Катаев, М.С. Клинов, Н.В. Ковалева, В.А. Крюков, Н.В. Поддерюгина

Система САПФОР использует автоматически распараллеливающий компилятор и подсказывает программисту, какие участки программы следуют переписать и где следует указать свойства последовательной программы, определить которые автоматически не удается. Это взаимодействие осуществляется через диалоговую оболочку с графическим интерфейсом.

Введение

Система САПФОР представляет собой систему автоматизированного распараллеливания Fortran-программ [1-4], которая помогает программисту эффективно отображать его программы на многоядерные кластеры.

Подход системы САПФОР – не заставлять программиста принимать непростые решения об организации параллельных вычислений, например, о выборе варианта распределения данных, а делать это автоматически. Таким образом, на программиста остаются:

- модификации последовательной программы (например, разбиение циклов, введение временных массивов, подстановка констант и переменных);
- спецификации свойств программы, определить которые автоматическими методами нельзя за приемлемое время (будем называть их недоступными для анализа).

Без такого участия программиста невозможно говорить о получении эффективных реальных программ для высокопроизводительных ЭВМ.

На выходе системы САПФОР получается параллельная программа на языке Fortran-DVM, Fortran-OpenMP или Fortran-DVM/OpenMP.

Состав системы САПФОР

Система САПФОР состоит из автоматически распараллеливающего компилятора АРК-DVM и диалоговой оболочки, которая осуществляет взаимодействие программиста с компилятором. Компилятор АРК-DVM состоит из двух анализаторов (статического и динамического), трех экспертов (DVM-эксперта, который принимает решения о распараллеливании программы на кластер, OpenMP-эксперта для мультипроцессора, DVM/OpenMP-эксперта для многоядерного кластера) и генератора, который вставляет DVM- и OpenMP-директивы в текст последовательной программы.

Статический анализатор работает по тексту последовательной программы и определяет следующие свойства для циклов с учетом вложенных в них вызовов процедур:

- информационные зависимости между витками цикла (OUTPUT, FLOW, ANTI);
- редукционные зависимости между витками цикла;
- приватные переменные (скаляры и массивы) для каждого витка. На каждом витке, прежде чем использовать приватную переменную, ей присваивается значение, и за пределами цикла тоже всегда есть присваивание перед использованием;
- регулярные зависимости по массивам. Между витками имеет место FLOW зависимость, и виток зависит от некоторого количества (не более константы) соседних предыдущих витков. Такой цикл допускает конвейерное выполнение его витков.

Динамический анализатор работает во время запуска последовательной программы на представительных данных. Он используется для циклов, с которыми не справляется статический анализатор, и способен определять:

- приватные переменные и зависимости (информационные, регулярные) в случаях косвенной индексации и с учетом динамически вводимых данных;
- параметры циклов, прежде всего, количество витков и время выполнения;
- размеры динамических массивов.

Оба метода анализа имеют свои недостатки, в частности, динамический анализ не может определять свойства циклов для всех возможных вводимых данных.

Эксперты помимо принятия решений о распараллеливании программы производят оценку их эффективности, которая представлена набором характеристик. В этом наборе имеются времена выполнения, вычислений, коммуникаций, дублирования вычислений и другие характеристики. Такие характеристики могут быть определены для любого интервала выполнения программы, в системе САПФОР они строятся для всех циклов программы, кроме циклов, вложенных в параллельные циклы.

Набор характеристик получается путем прогнозирования, основанного на моделировании выполнения программы.

Диалоговая оболочка позволяет программисту провести:

- исследование результатов анализа и задание характеристик программы, недоступных для анализа, но необходимых для распараллеливания;
- исследование предлагаемых вариантов распараллеливания и выбор из них наиболее предпочтительных;
- исследование выигрыша при возможных преобразованиях исходной программы.

Диалоговая оболочка системы САПФОР

Исследование результатов анализа может быть частично или полностью произведено сразу после работы анализаторов, но более удобно ее производить после работы экспертов, опираясь на построенные варианты распараллеливания, оценки их эффективности и причины того, что циклы программы остались не распараллеленными.

В любом случае результаты анализа представляют собой огромный объем информации. Поэтому в системе имеется режим постепенной детализации информации: сначала показ наиболее важных свойств и их комбинаций, потом их детализация и показ связанных с этим мест в программе.

Мы предлагаем показывать программисту такие комбинации свойств для циклов:

- межвитковые зависимости OUTPUT по переменной, если переменная не приватная и не редукционная;
- межвитковые зависимости FLOW и ANTI по переменной, если по ней нет зависимости OUTPUT и она не приватная, и не редукционная, указывается регулярная зависимость или нет;
- выход из цикла;
- использование операторов ввода-вывода в цикле.

Эти комбинации вносят существенные особенности в организацию параллельного выполнения циклов, которые могут привести к тому, что циклы нецелесообразно распараллеливать. В случае ЭВМ с распределенной памятью выполнение цикла всеми процессорами приведет в системе САПФОР к размножению модифицируемых в цикле данных, что приведет к нецелесообразности распараллеливания других циклов. Важный аспект заключается в том, что программист может уточнять результаты анализа, например, система распараллеливания может быть не уверена в точном наличии какого-то свойства (например, информационной зависимости), а программист может указать, что его точно нет.

После автоматического построения вариантов распараллеливания программист получает оценки их эффективности (наборы характеристик выполнения для циклов программы, для программы в целом по каждому варианту) и причины того, почему циклы остались не распараллеленными. Оценка зависит от процессорной решетки (логическое объединение процессоров), ее вида и количества процессоров в ней. Перебрав некоторое количество решеток при фиксированном количестве процессоров в них, можно сравнивать схемы с разным распределением данных. В системе САПФОР имеется алгоритм поиска оптимальной решетки процессоров для заданного количества процессоров, но есть возможность задать свою решетку процессоров. Имея оценки вариантов и причины отказа от распараллеливания циклов, программист может ограничиться уточнением свойств только наиболее существенных циклов.

При распараллеливании на кластер, в отличие от мультипроцессора, может возникнуть ситуация, когда абсолютно все циклы могут быть распараллелены, но не удается найти такой вариант распределения данных, чтобы время выполнения параллельной программы было бы меньше, чем время выполнения исходной последовательной программы. Как убедить программиста, что такого варианта распределения данных не существует? Для этого случая предусмотрена возможность для программиста задать вариант распределения массивов, а система должна показать, что заданный вариант распределения не лучше тех, что были найдены ею.

В ряде случаев для получения эффективных программ от программиста потребуется не только задание свойств программы, но и модификация исходной программы. Например, в качестве такой модификации можно рассматривать следующие действия: разбить цикл на несколько, избавиться от зависимостей в циклах, завести дополнительные переменные, сделать тесно-вложенные циклы, когда телом одного цикла является другой цикл, сделать подстановку констант и переменных.

В системе распараллеливания есть возможность оценить выигрыш от преобразований последовательной программы без их проведения – путем изменения свойств программы. Эта возможность быстрой оценки преобразований является очень важной, особенно для сложных программных комплексов.

Еще один важный аспект работы программиста с системой САПФОР заключается в том, что программист может допустить ошибки, например, при спецификации свойств циклов программы, а система может ему об этом сообщить.

После получения текста параллельной программы программисту следует проверить ее с помощью инструментов DVM-системы (динамический контроль корректности и сравнительный отладчик) [5]. Инструменты покажут ему ошибку в терминах параллельной программы, но он сможет связать ее с исходной,

потому что в системе САПФОР параллельная программа (для кластера, для мультипроцессора или для многоядерного кластера) текстуально близка к исходной.

В настоящее время система САПФОР и ее компоненты представляют собой экспериментальные версии, которые используются разработчиками для исследований и не полностью реализуют функциональность системы распараллеливания. На рисунке 1 представлено окно диалоговой оболочки системы на примере теста LU из пакета тестов NAS [6]. В нем показан список программных единиц (процедур), текст программы, список циклов с характеристиками выполнения и свойствами. Для циклов указаны: итерационная переменная цикла, ее начальное и конечное значение (в виде выражения), шаг цикла, в какой программной единице цикл описан, в каком файле, на какой строке этого файла начинается цикл, номер цикла (по степени их вложенности), время последовательного и параллельного выполнения (на некотором числе процессоров, в данном случае 8 процессорах), ускорение, спрогнозированное время коммуникаций в цикле, наличие регулярной зависимости между витками цикла и переменные, которые читаются в цикле.

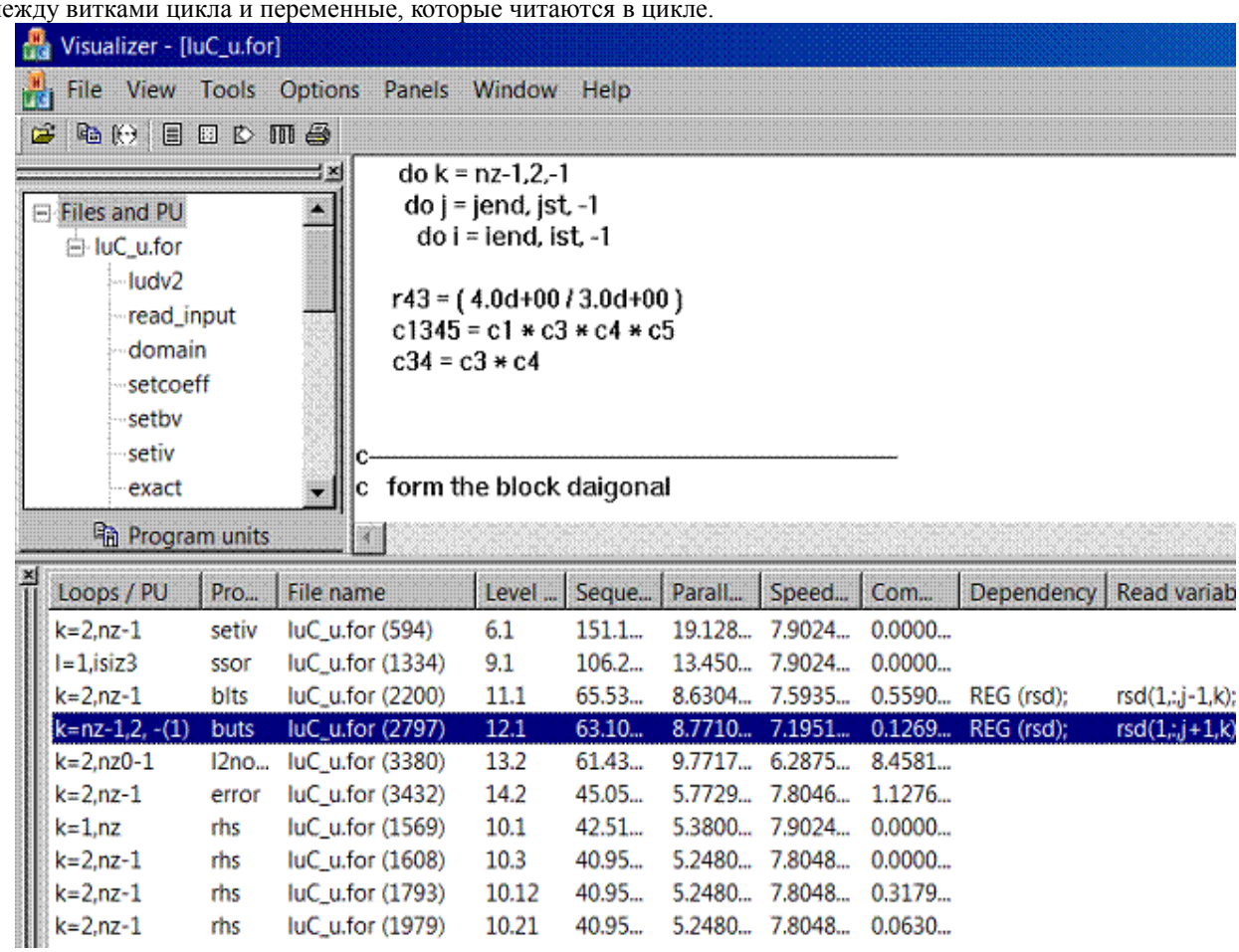

Рис. 1. Окно диалоговой оболочки системы САПФОР на примере теста LU.

Заключение

Практическое использование автоматически распараллеливающих компиляторов для получения эффективных программ затруднительно без участия программиста и дополнительных инструментов, таких как удобная диалоговая оболочка и средств отладки параллельных программ.

В настоящее время ведется развитие этих инструментов и всех компонентов компилятора APK-DVM для расширения входного языка Fortran 77 до языка Fortran 95.

Эти работы поддержаны Программами президиума РАН №14, №15 и №17, грантом РФФИ № 10-07-00211 и грантом Президента РФ МК-3324.2010.9.

Их успешное выполнение позволит перейти к практическому внедрению системы автоматизированного распараллеливания программ на языке Fortran и ее дальнейшему развитию применительно к кластерам с неоднородными узлами (содержащими в качестве ускорителей графические процессоры).

ЛИТЕРАТУРА:

1. Система САПФОР: сайт. URL: http://www.keldysh.ru/dvm/SAPFOR/ (дата обращения 16.05.2011)
2. М.С. Клинов, В.А. Крюков “Автоматическое распараллеливание Фортран-программ. Отображение на кластер” // Вестник Нижегородского университета им. Н.И. Лобачевского, 2009, № 2, с. 128–134

3. М.С. Клинов “Прогнозирование характеристик эффективности выполнения DVM-программ на кластере” // Вестник Нижегородского университета им. Н.И. Лобачевского, 2009, №. 4, с. 190-197
4. В.А. Бахтин, М.С. Клинов, В.А. Крюков, Н.В. Поддерюгина “Автоматическое распараллеливание последовательных программ для многоядерных кластеров” // Материалы Всероссийской научной конференции “Научный сервис в сети Интернет: суперкомпьютерные центры и задачи”, Новороссийск, 20-25 сентября 2010 г., Изд-во Московского Университета, с. 12-15
5. Отладка DVM-программ. URL: http://www.keldysh.ru/dvm/dvmhtm1107/rus/usr/debug/debugUGr.html (дата обращения 16.05.2011)
6. The NAS Parallel Benchmarks: сайт. URL: http://www.nas.nasa.gov/Resources/Software/npb.html (дата обращения 24.10.2009)