

Б. Я. ШТЕЙНБЕРГ, Е. Н. КРАВЧЕНКО, Р. И. МОРЫЛЕВ, З. Я. НИС, В. В. ПЕТРЕНКО,
И. С. СКИБА, В. Н. ШАПОВАЛОВ, О. Б. ШТЕЙНБЕРГ, Р. Б. ШТЕЙНБЕРГ

ОСОБЕННОСТИ РЕАЛИЗАЦИИ РАСПАРАЛЛЕЛИВАЮЩИХ ПРЕОБРАЗОВАНИЙ ПРОГРАММ В СИСТЕМЕ ДВОР

Обсуждаются особенности функционирования библиотеки преобразований программ в диалоговом высокоуровневом оптимизирующем распараллеливателе.

Ключевые слова: распараллеливание, преобразование программ, оптимизация.

Диалоговый высокоуровневый оптимизирующий распараллеливатель программ (ДВОР) является развитием Открытой распараллеливающей системы [1]; он позволяет работать с различными языками программирования, поддерживает общую библиотеку и возможность диалоговой компиляции. Для реализации этих функций ДВОР использует высокоуровневое внутреннее представление программы, в котором проводятся оптимизирующие (распараллеливающие) преобразования, что выгодно отличает его от существующих аналогов [2, 3]. Представление является иерархическим, с разной детализацией программы на каждом уровне. Верхний уровень для каждого входного языка индивидуален и ориентирован на удобство разбора программы. Второй уровень, общий для всех входных языков, ориентирован на преобразования программ. На этом уровне разобранная программа хранится в виде четырех деревьев: идентификаторов, типов, операторов и выражений, что удобно для диалоговой формы преобразований.

Диалоговое распараллеливание является компромиссом между ручным и автоматическим. Программисту предоставляются автоматизированные инструменты анализа и преобразования программ, ускоряющие распараллеливание по сравнению с ручной обработкой. Использование автоматизированных инструментов повышает надежность кода и снижает требования к квалификации и опыту программиста, поскольку в диалоговом режиме компиляции возможно запросить недостающую информацию для эффективного распараллеливания программы. Например, для определения информационных зависимостей в программе могут быть полезны диапазоны значений переменных, размеры массивов данных, соотношения между различными переменными, которые в ряде случаев в программе явно не определены.

В целях распараллеливания ДВОР позволяет распознавать характерные фрагменты кода, выполнять над ними соответствующие преобразования, а также осуществлять проверку сохранения синтаксиса и семантики, эквивалентности преобразования данных. Также оцениваются целесообразность преобразования, а при наличии нескольких вариантов — выбор оптимального способа преобразования для заданного фрагмента. Проверки могут выполняться автоматически и в режиме диалога. Для контроля эквивалентности преобразований кроме традиционного графа информационных связей используется решетчатый граф программы [4].

ДВОР поддерживает такие преобразования, как линеаризация выражений, гнездование и разрезание циклов, разрезание условных операторов. При линеаризации выражений в качестве

базовых переменных кроме счетчиков циклов могут выступать и внешние переменные (которые не изменяют своего значения в преобразуемом фрагменте), и имена массивов с различными индексами (при распараллеливании рекуррентных циклов). При линеаризации на этапе компиляции приводятся подобные *члены* с учетом типов данных. Гнездование циклов используется при распараллеливании цикла на параллельной архитектуре класса MIMD. ДВОР поддерживает процедуру проверки структуры преобразуемого фрагмента на возможность применения преобразования гнездования. При данной проверке система получает фрагмент программы и список допустимых в нем единиц языка (список допустимых типов операторов и т.д.) и проверяет, содержатся ли в данном фрагменте только единицы языка из списка. При такой форме проверки в случае последующего расширения внутреннего представления новыми операторами система останется работоспособной без изменения преобразования.

Разрезание циклов в системе ДВОР использует граф информационных связей, по которому строится фактор-граф. В полученном фактор-графе строится правильная нумерация. Далее проводятся анализ применения вспомогательных преобразований и непосредственно разрезание на части, следуемые в порядке правильной нумерации. Преобразование разрезания цикла может иметь разные вариации (например, разрезание в указанном месте или на максимальное количество частей). Разрезание условного оператора заменяет один условный оператор двумя подряд идущими условными операторами, при этом блок ветки *then* исходного условного оператора разбивается на части и каждая из частей становится блоком одного из результирующих условных операторов.

Дополнительно в ДВОР поддерживается расширение скалярной SSA-формы (Static Single Assignment Form, форма со статически однократным присваиванием) [5] на код с массивами. SSA-форма для массивов строится с помощью нечеткого анализа потока данных (Fuzzy Array Dataflow Analysis) [6]. В SSA-форме для массивов некоторые скалярные преобразования расширяют область применения: удаление мертвого кода, протягивание констант, использование общих подвыражений, распределение регистров и др.

Работа выполнена в рамках ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

СПИСОК ЛИТЕРАТУРЫ

1. Открытая распараллеливающая система [Электронный ресурс]: <<http://ops.rsu.ru/>>.
2. [Электронный ресурс]: <<http://www.csrd.uiuc.edu/promis/>>.
3. [Электронный ресурс]: <<http://www-suif.stanford.edu/research/>>.
4. Воеводин В. В. Воеводин Вл. В. Параллельные вычисления. СПб: БХВ-Петербург, 2002. 599 с.
5. Muchnick S. S. Advanced Compiler-Design and Implementation. Morgan Kaufmann Publishers, 1997. 860 p.
6. Collard J.-F., Barthou D., Feautrier P. Fuzzy Array Dataflow Analysis // ACM SIGPLAN Notices. 1995. Vol. 30, Iss. 8.

Сведения об авторах

- Борис Яковлевич Штейнберг** — д-р техн. наук; Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; зав. кафедрой; E-mail: borsteinb@mail.ru
- Евгений Николаевич Кравченко** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; программист; E-mail: peon_sxe@mail.ru
- Роман Игоревич Морылев** — аспирант; Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; E-mail: frg10@yandex.ru
- Зиновий Яковлевич Нис** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; программист; E-mail: irishrover@mail.ru

- Виктор Владимирович Петренко** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; научный сотрудник;
E-mail: vpetrenko@gmail.com
- Иван Сергеевич Скиба** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; программист; E-mail: aidan@pisem.net
- Василий Николаевич Шаповалов** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; младший научный сотрудник;
E-mail: Vasily.Shapovalov@gmail.com
- Роман Борисович Штейнберг** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; научный сотрудник;
E-mail: romanofficial@yandex.ru
- Олег Борисович Штейнберг** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; ассистент;
E-mail: romanofficial@yandex.ru

Рекомендована НИИ НКТ

Поступила в редакцию
15.05.11 г.