УДК 681.3.068

СОСТОЯНИЕ И ВОЗМОЖНОСТИ ОТКРЫТОЙ РАСПАРАЛЛЕЛИВАЮЩЕЙ СИСТЕМЫ (лето 2006 г.)

Б.Я. Штейнберг, З.Я. Нис, В.В. Петренко,
Д.Н. Черданцев, Р.Б. Штейнберг, А.М. Шульженко

Ростовский государственный университет
Россия, 344006, г. Ростов-на-Дону, ул. Б. Садовая, 105, e-mail: steinb@ns.math.rsu.ru

1. Введение

Открытая распараллеливающая система (ОРС, [1]) предназначена для автоматического распараллеливания программ с процедурных языков программирования на параллельные компьютеры. С помощью ОРС могут быть разработаны: распараллеливающие компиляторы или конверторы в языки высокого уровня, диалоговые распараллеливающие системы, программы для обучения параллельному программированию и т.д. Преобразования из библиотеки ОРС можно будет комбинировать для различных архитектур. На сегодняшний день в ОРС автоматически определяются циклы ParDO, т.е. циклы, итерации которых могут выполняться независимо [3]. Автоматически определяются конвейеризуемые циклы, автоматически рассчитывается синхронизация конвейерных вычислений. Основные части ОРС: внутреннее представление, графовые модели программ, библиотека оптимизирующих/распараллеливающих преобразований программ, дополнительные функции компиляции. Реализуется ОРС на языке C++ и на сегодняшний день объем кода составляет около 100 000 строк.

2. Внутреннее представление ОРС

Внутреннее представление (ВП) – это структура данных, хранящая полную информацию о программе, а также библиотека сервисных функций, облегчающих выполнение преобразований программ [5]. ВП в ОРС разработано так, чтобы быть:

- универсальным, т.е. позволять хранить программы на нескольких процедурных языках: Си, Фортран и Паскаль;
- базисом для построения, анализа информационных зависимостей и написания преобразований программ, при этом похожие свойства процедурных языков унифицировать, а специфичные сохранять;
- расширяемым для подключения компиляторов переднего плана с новых языков программирования к ОРС;
- удобным для генерации машинного кода различных целевых архитектур.

ВП спроектировано с применением Design Patterns [8]. Разработан механизм передачи информации между различными частями ОРС в виде специальных пометок в узлах ВП. Реализованы отладочные средства и средства проверки целостности структуры ВП во время работы программы.

ВП Открытой распараллеливающей системы является связной древовидной структурой и состоит из четырех деревьев: типов, идентификаторов, выражений и операторов. Каждое дерево ВП хранит информацию о соответствующей части программы. ВП проектировалось так, чтобы при преобразованиях программ сохранялась синтаксическая и семантическая целостность.

При разработке ВП были рассмотрены и проанализированы внутренние представления наиболее известных распараллеливающих систем Polaris [10] и SUIF [11], их достоинства и недостатки.

3. Интерфейс, система визуализации и графовые модели программ

Интерфейс и система визуализации позволяют в диалоговом режиме производить цепочки преобразований, позволяют анализировать возможности распараллеливания программ, автоматически выводя на экран графовые модели программ.

На следующей экранной форме показано как в результате последовательности действий (выбрано преобразование «Iteration Space Splitting», отмечено гнездо циклов и два вхождения переменной «a», и затем нажатие кнопки «Transform») в окне справа была получена преобразованная программа.

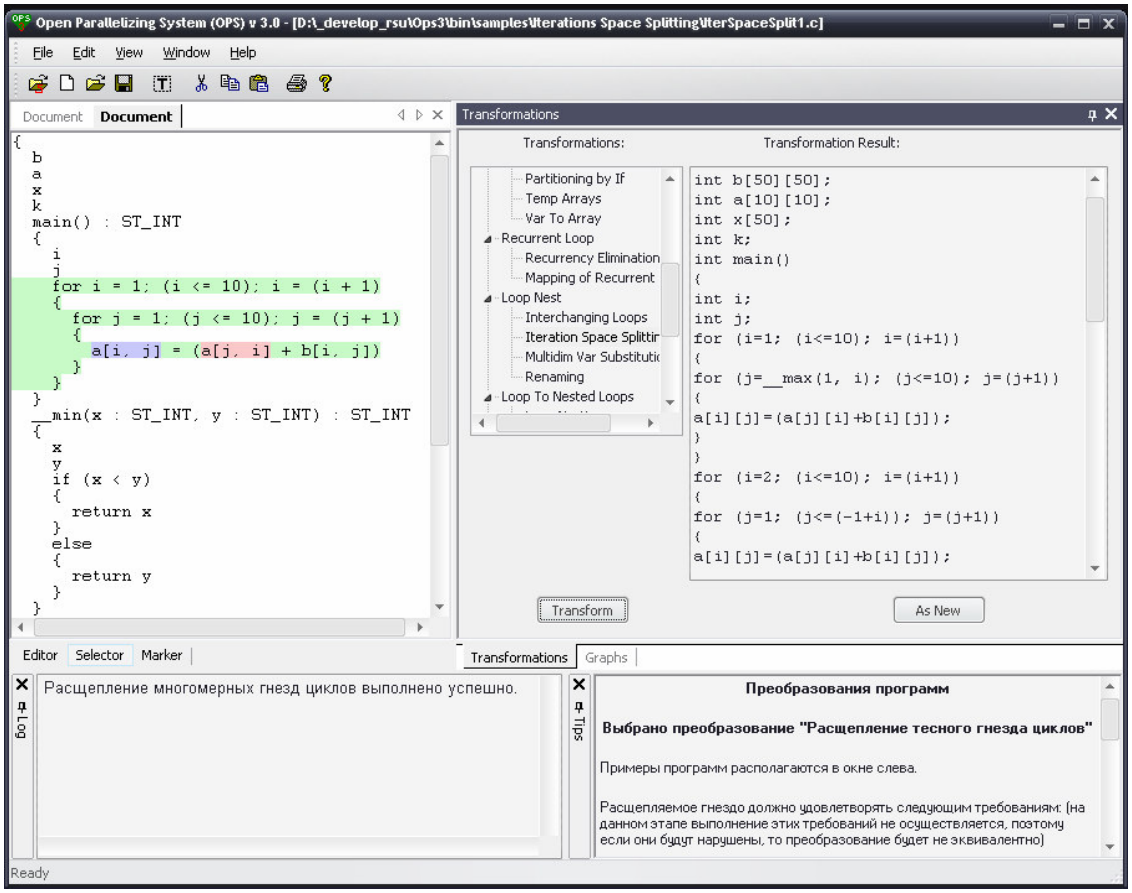

Рис. 1. Интерфейс OPC. Диалоговое выполнение преобразований.

В OPC реализованы и визуализированы следующие графовые модели программ: граф информационных связей, граф вычислений (РГА), решетчатый граф программы, управляющий граф программы, граф вызовов подпрограмм.

4. Семантическая корректность преобразований

Так как большой класс преобразований оперирует фактически исходными текстами программ, то для таких преобразований становится актуальной проблема сохранения статической семантической корректности.

Даже в простых преобразованиях возможны неочевидные проблемы с семантикой. Рассмотрим преобразование «раскрутка цикла», заключающееся в замене цикла со счетчиком N-кратным повторением тела цикла, где N – известное во время компиляции количество итераций. Если тело цикла содержит операторы досрочного перехода к следующей итерации (continue в языке Си), то после применения преобразования эти операторы окажутся

вне цикла, что является нарушением семантических ограничений. При таких преобразованиях, как развертка, расщепление, гнездование или разбиение цикла появляется две или более копий тела исходного цикла с измененными индексными выражениями массивов. Неосторожный разработчик может не предусмотреть возникновение нескольких операторов с одинаковыми метками.

В ОРС ведутся работы по автоматизации контроля статической семантической корректности преобразований программ. Система контроля корректности включает в себя разработку языка описания преобразований и разработку формального описания статической семантики процедурных языков.

5. Библиотека преобразований ОРС.

Преобразования в ОРС делятся на группы, в зависимости от типов фрагментов исходного текста, к которым они применяются. Написанные преобразования проверяют условия эквивалентности, основанные на графе информационных связей или на решетчатом графе. В ОРС реализованы смешанные вычисления и стандартизация выражений, что позволяет повысить широту применимости преобразований.

В ОРС реализованы следующие преобразования,: подстановка вперед, включая интервальную подстановку с оценкой погрешности, вынесение общих подвыражений; растягивание скаляров, расщепление вершин, перестановка операторов, канонизация циклов, разбиение циклов, слияние циклов, перестановка циклов, гнездование циклов, разрыв итераций.

ОРС содержит преобразования, основанные на решетчатых графах, эквивалентность которых не может быть основана на традиционном графе программных зависимостей: расщепление многомерных гнезд циклов в виде последовательности тесных гнезд; подстановка индексных переменных; экспансия массивов, на основе расщепления в виде последовательности тесных гнезд циклов.

В ОРС ведутся работы над распараллеливанием рекуррентных циклов [2], над оптимизацией кода, содержащего условные операторы, над динамическими преобразованиями программ. Ведется работа с индуктивными переменными. Предполагается разработка архитектурно-ориентированных комбинаций преобразований [2].

6. Оценки точности и частичные вычисления

В рамках ОРС реализован особый режим компиляции с автоматической оценкой погрешности начальных данных и округления. Получен конвертор, преобразующий пользовательскую программу на языке Си в новую программу, которая помимо запрограммированных пользователем вычислений выдает оценки погрешностей проведенных вычислений.

Для расчета погрешности производится замена входной переменной парой чисел (например, переменная A заменяется на пару (A, Aerr), где A – приближение к истинному значению, а Aerr – оценка относительной погрешности) и переопределением арифметических операций. При построении оценок для выражений, содержащих вызовы функций, применяется механизм порождения новых вспомогательных функций, которые вычисляют погрешность функций, которым они соответствуют.

Частичные вычисления позволяют еще на этапе компиляции упрощать и уменьшать время выполнения функций, если известны значения хотя бы части фактических аргументов. В ОРС реализован моновариантный метод частичных вычислений [12, 7].

И оценки точности и частичные вычисления используются при протягивании интервальных констант в функции.

7. Конвейерные и многоконвейерные вычисления

На данный момент в ОРС реализован расчет таких параметров конвейеризации циклов как задержки и интервал инициализации итераций. По одномерному циклу определяется

наличие зависимостей, препятствующих конвейеризации. В случае их отсутствия, строится регулярный граф алгоритма (РГА), определяется критический путь на этом графе и обратные дуги, порожденные регулярными истинными зависимостями. ОРС позволяет проанализировать все эти характеристики визуально, с помощью модуля визуализации РГА. Также ОРС позволяет сгенерировать таблицу расписания по описанным выше характеристикам цикла и его РГА, а затем визуализировать эту таблицу. Модуль расчета синхронизации конвейера ориентирован на архитектуру перестраиваемого конвейера [9], но может быть адаптирован и к любой другой конвейерной архитектуре.

В группе ОРС ведутся исследования в области многоконвейерных вычислительных систем. Исследуемая в работах модель многоконвейерной вычислительной системы допускает выполнение нескольких программных фрагментов на нескольких конвейерах одновременно. На данный момент получены формулы вычисления задержек между моментами начала работы конвейеров [4]. Описаны некоторые высокоуровневые преобразования, приводящие фрагмент программы к виду, удобному для отображения на многоконвейерную архитектуру. Программный модуль автоматического расчета задержек между стартами конвейеров находится в процессе разработки и в ближайшее время будет подключен к ОРС.

Список литературы:

1. www.ops.rsu.ru
2. Штейнберг Б.Я. Математические методы распараллеливания рекуррентных программных циклов на суперкомпьютеры с параллельной памятью // Ростов-на-Дону, Издательство Ростовского университета, 2004 г., 192 с.
3. Шульженко А.М. Автоматическое определение циклов ParDo в программе // Известия вузов. Северокавказский регион. Естественные науки. Приложение 11’05. с. 77-88.
4. Штейнберг Р.Б. Вычисление задержки в стартах конвейеров для суперкомпьютеров со структурно процедурной организацией вычислений // Искусственный интеллект. Научно-теоретический журнал. Институт проблем искусственного интеллекта НАНУ. Украина, Донецк, ДонДИШИ, “Наука и Освита”, № 4, 2003, с. 105-112.
5. Петренко В.В. Внутреннее представление ОРС3 // Научный сервис в сети Интернет: технологии распределенных вычислений: Труды Всероссийской научной конференции. (19-24 сентября 2005 г., г. Новороссийск). – М.: Изд-во МГУ, 2005, с. 106-108.
6. Черданцев Д.В. Автоматическая оценка погрешности // Научный сервис в сети Интернет: технологии распределенных вычислений: Труды Всероссийской научной конференции. (19-24 сентября 2005 г., г. Новороссийск). – М.: Изд-во МГУ, 2005, с. 81-82.
7. Ершов А.П. Смешанные вычисления: потенциальные применения и проблемы исследования. http://ershov.iis.nsk.su/archive/eaindex.asp?lang=1&did=2596
8. Гамма Э., Хэлм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования // СПб: Питер, 2001. – 368 с.: ил.
9. Каляев А.В. Программирование виртуальных параллельных проблемно-ориентированных суперкомпьютеров в структуре универсальных суперкомпьютеров с массовым параллелизмом// Международная научно-техническая конференция <Интеллектуальные многопроцессорные системы>/ 1-5 сентября, 1999, Таганрог, Россия, Сборник трудов, с.27-39.
10. K. Faigin, J. Hoeflinger, D. Padua, P. Petersen, S. Weatherford. The Polaris Internal Representation.
11. G. Aigner, A. Diwan, D. Heine, M. Lam, D. Moorey, B. Murphy, C. Sapuntzakis. The SUIF Program Representation.
12. N.D. Jones, C.K. Gomard, P. Sestoft "Partial evaluation and automatic program generation", Prentice-Hall, Inc., 1993.