

РАЗВИТИЕ ВОЗМОЖНОСТЕЙ АНАЛИЗАТОРА ПОСЛЕДОВАТЕЛЬНЫХ ПРОГРАММ

Т.П. Баранова, В.Ю. Вершубский, К.Н. Ефимкин, В.А. Федосимов

Задача автоматизации распараллеливания последовательных программ решается в различных постановках много лет, как отечественными, так и зарубежными исследователями, однако удовлетворительного, с практической точки зрения, результата пока не получено. Тем не менее, в последнее время расходы за рубежом на исследования по автоматизации распараллеливания сравнимы с расходами на исследования новых архитектур суперкомпьютеров. Это объясняется, по-видимому, возрастающей сложностью ручной разработки эффективных параллельных программ для все более усложняющихся архитектур параллельных компьютеров.

В ИПМ им. М.В. Келдыша РАН ведутся работы по созданию системы автоматизации распараллеливания Фортран-программ [1], в состав которой входит статический анализатор свойств последовательной программы [2,3]. В настоящее время существует две версии статического анализатора. Первая - это диалоговый статический анализатор ВерБа, ориентированный на автоматизацию ручного распараллеливания и поддерживающий анализ свойств последовательной программы, существенных для распараллеливания. Вторая версия – это статический анализатор, функционирующий как консольное приложение в составе системы распараллеливания, без возможности ведения диалога с пользователем.

Обе версии анализатора используют адаптированные парсер и библиотеку классов пакета SAGE [4].

Диалоговый анализатор ВерБа работает независимо от системы автоматизации распараллеливания и поддерживает графический режим представления результатов анализа.

Одна из существенных проблем при создании систем анализа больших программ – обеспечение возможно более быстрой обработки запросов пользователя. С этой точки зрения в анализаторе принята следующий подход: на этапе инициализации строятся основные структуры, в которых хранится информация, нужная в большинстве запросов и наличие которой позволяет осуществлять быструю визуализацию результатов отдельного запроса. Базовая информация помещается в постоянных, относительно сеанса работы, структурах данных.

Ниже кратко описан сценарий работы пользователя с диалоговым анализатором. После запуска анализатора пользователь видит главное окно. В меню этого окна пользователь может традиционным способом открыть проект, связанный с анализируемой последовательной программой.

После открытия проекта становится доступной пункт главного меню Собрать сведения, предназначенный для сбора базовой информации о программах проекта в основных структурах данных. При сборе базовой информации в окне История для файлов проекта выдается информация о времени обработки. В конце обработки всех файлов проекта указывается общее число строк во всех файлах проекта. Ниже на Рис. 1 показано главное окно анализатора после открытия и сбора сведений по проекту max.prj, состоящего из 10 файлов.

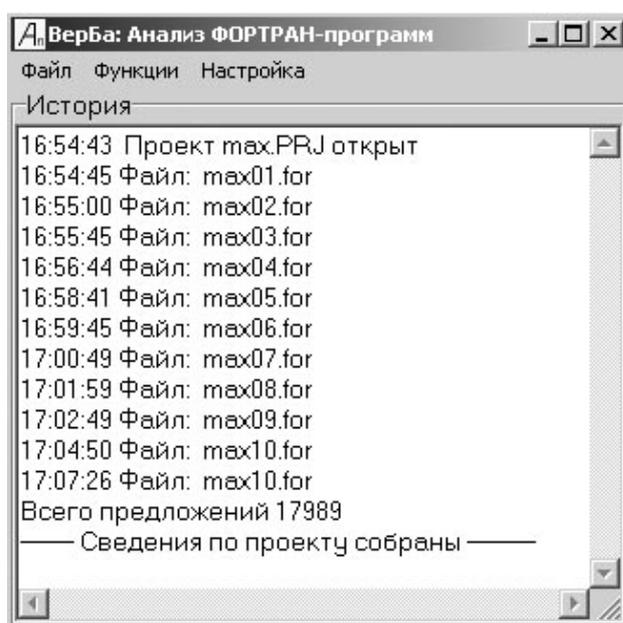


Рис. 1. Вид окна История

После сбора сведений в пункте меню Функции становится доступным подпункт Сведения по процедурам, после выбора которого появляется окно для отображения свойств программных единиц проекта. Ниже на Рис. 2 приведен пример такого окна для проекта max.pj.

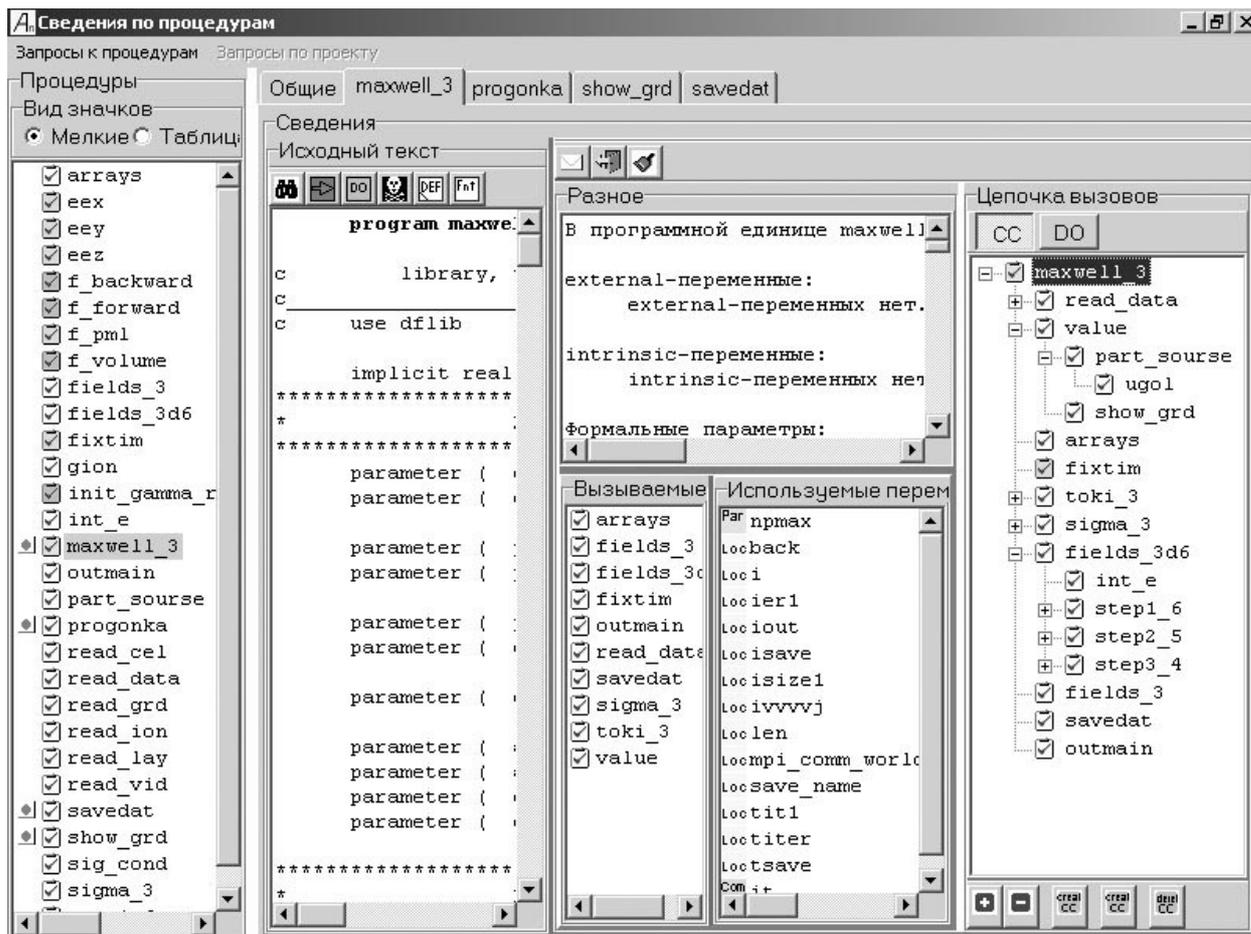


Рис. 2. Окно Сведения по процедурам

В окне отображаются следующие сведения:

Панель Процедуры содержит перечень программных единиц проекта. Программные единицы отмечены разноцветными значками, соответствующими виду программной единицы: program, subroutine, function, а также объявленные в проекте, но не использованные программные единицы. В этой панели для приведенного примера отмечены четыре программных единицы (они отмечены точками). Информация об отмеченных программных единицах располагается на страницах блокнота Сведения. В приведенном примере выбрана страница для программной единицы maxwell_3.

Панель Сведения содержит пять вложенных панелей: Исходный текст, Разное, Цепочка вызовов, Используемые переменные и панель с плавающим названием.

Панель Исходный текст содержит текст программной единицы и набор кнопок, обеспечивающий навигацию по этому тексту.

Панель Разное содержит перечень всех объявленных в программной единице переменных и параметров. Для переменных указан способ объявления (явно, неявно, implicit), глобальность (common), тип данных (real, integer, double precision, и т.п.). Для именованных констант (parameter) - присвоенные им значения.

Панель Цепочка вызовов предназначена для отображения, либо цепочки вызовов программных единиц, начиная от текущей, либо иерархической схемы циклов DO в рамках текущей программной единицы.

Панель Используемые переменные содержит только использованные в программной единице переменные и параметры. Упомянутые в панели имена помечены значками, отражающими разновидность имен (common, локальная, формальный параметр, именованная переменная).

Панель с переменным заголовком содержит сведения, соответствующие запросу, выбранному из меню Запросы к процедурам (программные единицы, вызываемые из данной, программные единицы, вызывающие данную, список используемых в данной программной единице common-блоков, список меток, лишние метки, неиспользуемые переменные, список массивов заданной размерности).

В анализаторе появились возможность представления информации о переменных с индексами, которая собирается по всем программным единицам проекта. Собранная информация помещается в список,

который можно просматривать в специальном окне. Для задания условий просмотра предусмотрено дополнительное всплывающее окно Фильтр для индексных выражений. Окно содержит два класса управляющих элементов: элементы для включения объектов списка в число отображаемых (радио кнопки), и элементы для исключения объектов списка из числа отображаемых (флажки). Ниже на Рис. 3 показан пример этого окна, в случае, когда задан запрос на отображение всех обращений к массиву с именем gm_v_1.

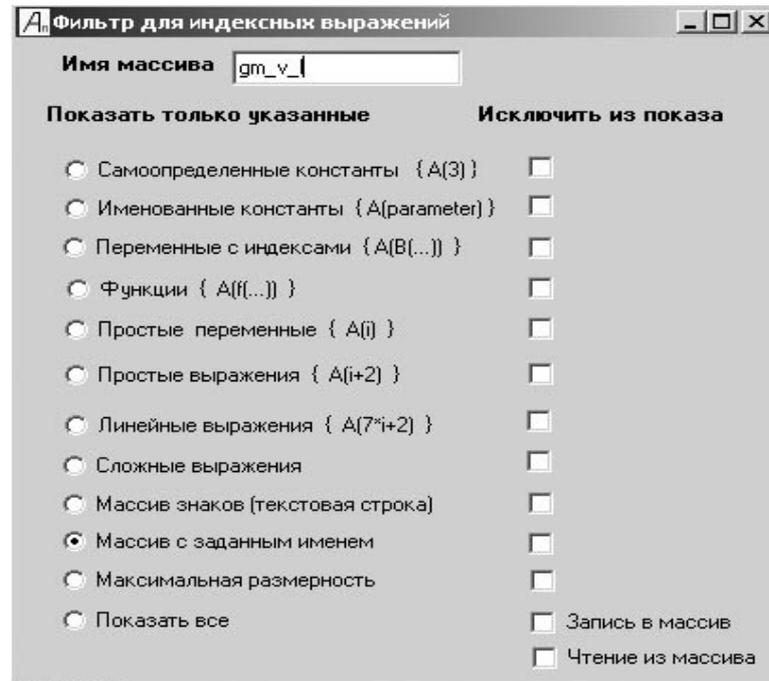


Рис. 3. Окно Фильтр для индексных выражений

На Рис. 4 показан пример отображения всех элементов списка обращений ко всем массивам.

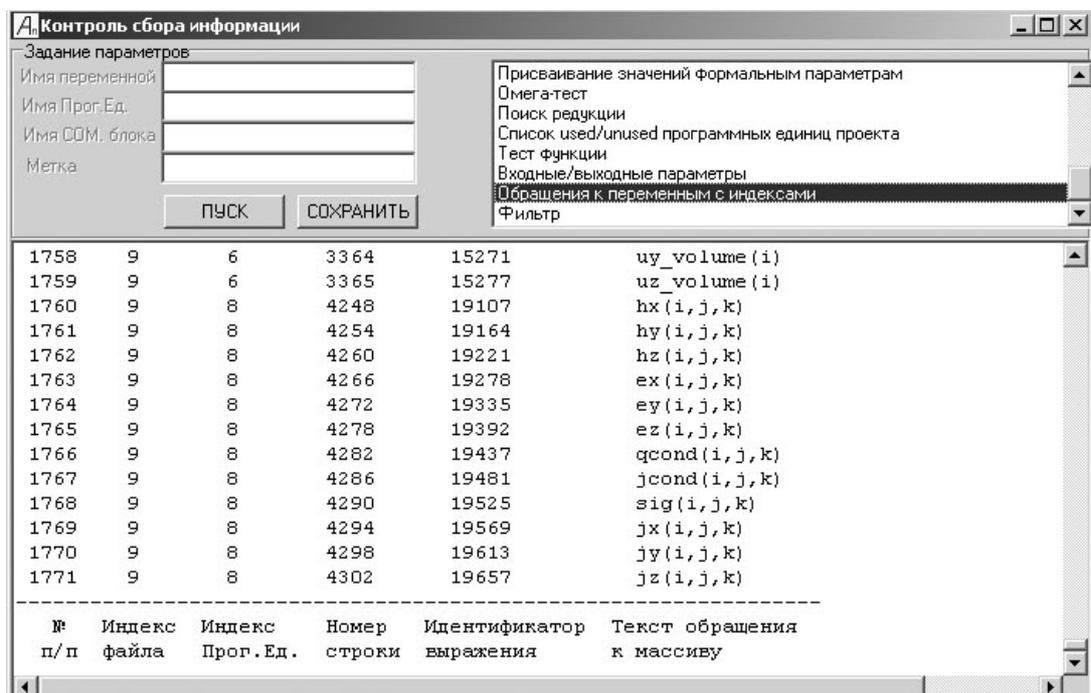


Рис. 4. Вид отображения обращений к массивам

На Рис. 5 показан результат фильтрации заданной на Рис. 3.

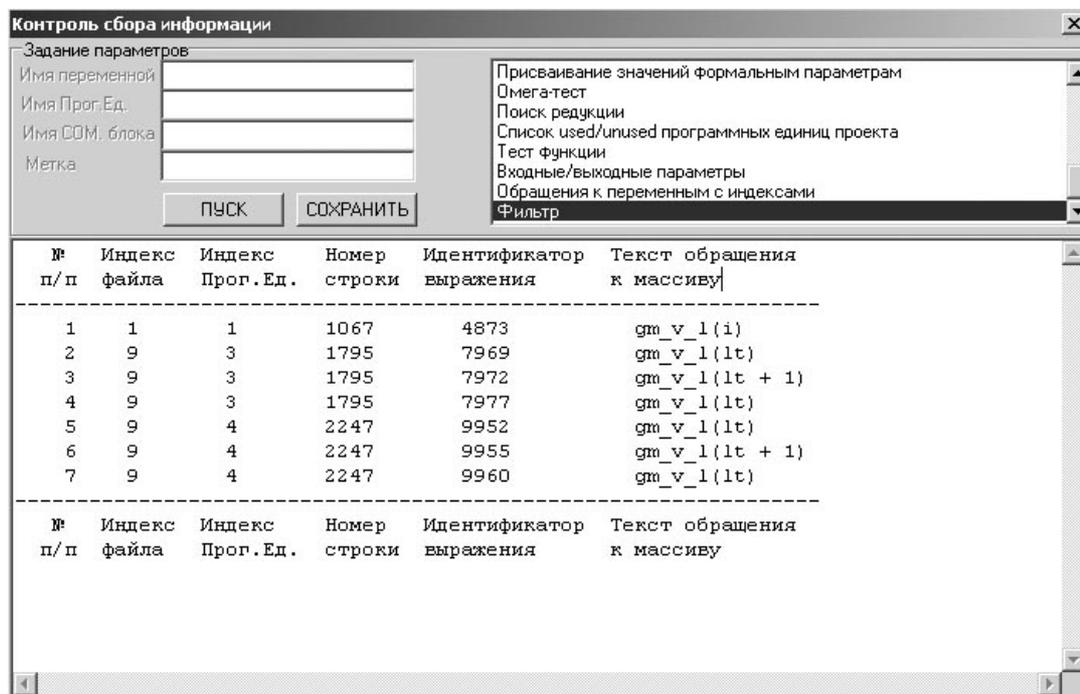


Рис. 5. Обращения к массиву с именем gm_v_1 (фильтрация)

В дальнейшем предполагается расширить функциональность анализатора: - для выводимых в окне обращений определить некоторые действия, например, перейти к строке исходного текста программы, в которой находится данное обращение к массиву, и т.п.

Обе версии анализатора (диалоговая и консольная) обеспечивают запись собранной информации об анализируемой программе в базу данных, основанную на СУБД sqlite (для последующего использования в системе распараллеливания). Помимо функции передачи данных между компонентами системы эта база данных может служить также для долговременного хранения. Данные, записанные в базу анализатором, в дальнейшем используются другими компонентами системы распараллеливания (экспертом, генератором выходной программы, визуализатором).

Записываемую анализатором в базу данных информацию условно можно разделить на два типа: сведения о проекте в целом и сведения о каждой программной единице, входящей в проект.

Для проекта в целом указываются:

- режимы работы анализатора: подавлять сбор сведений о частных переменных; сведений о редуцированных переменных; выбор сведений об ARRAY-зависимостях из структур, полученных Омега-тестом [5].
- имена всех COMMON-блоков проекта;
- список файлов проекта;
- перечень программных единиц проекта.

Для каждой программной единицы указываются:

- все переменные, объявленные в программной единице: параметры, формальные параметры, скаляры и массивы, при этом указывается имя переменной (для массивов записывается размерность, значения верхней и нижней границ размерностей) а также один из следующих атрибутов переменной: DATA, SAVE, EQUIVALENCE, ASSOCIATION, в случае, если переменная указана, соответственно, в операторе DATA, SAVE, EQUIVALENCE, или, в последнем случае, одноименные COMMON переменные в разных программных единицах описаны по разному;
- сведения по операторам циклов: иерархическая схема циклов DO; для каждого цикла указывается вся информация об итерационной переменной цикла; для каждой используемой в теле цикла переменной – ее имя, тип, тип использования (чтение-запись), для переменных-массивов - описание индексных выражений; наличие зависимостей в цикле - редуцированные зависимости, частные переменные, межвитковые зависимости типа ARRAY;
- сведения о вызовах других подпрограмм;
- сведения по COMMON-переменным;
- сведения по специальным комментариям;
- сведения (особенности) по операторам ввода/вывода.

НЕКОТОРЫЕ ХАРАКТЕРИСТИКИ ВРЕМЕНИ ВЫПОЛНЕНИЯ ЗАПРОСОВ

Для проекта max.rg, состоящего из 10 файлов, 36 программных единиц, 1772 строки исходного текста на компьютере с процессором AMD Athlon XP-2500+, ОЗУ – 1 Гб (два модуля по 512 Мб с расщеплением памяти), частота процессора 1,4 ГГц:

- время сбора сведений по проекту 12 минут, 41 секунда.
- время выполнения запроса на отображение обращений к массиву – 3 секунды.

Работа выполнена при частичном финансировании по гранту РФФИ N 07-07-00221.

ЛИТЕРАТУРА:

1. М.С. Клинов, В.А. Крюков. Система автоматизированного распараллеливания программ на языке Фортран. DVM-эксперт. // В сб. трудов Всероссийской научной конференции Научный сервис в сети Интернет: многоядерный компьютерный мир, сентябрь 2007 г., г. Новороссийск, с. 95-97.
2. Т.П. Баранова, В.Ю. Вершубский. Анализатор программ, написанных на языке Фортран. Препринт Института прикладной математики им. М. В. Келдыша РАН № 124, Москва, 2005, 27 стр.
3. Т.П. Баранова, В.Ю. Вершубский, К.Н. Ефимкин, В.А. Федосимов. Статический анализатор последовательных программ. // В сб. трудов Всероссийской научной конференции Научный сервис в сети Интернет: многоядерный компьютерный мир, сентябрь 2007 г., г. Новороссийск, с. 67-70.
4. www.extreme.indiana.edu
5. www.cs.umd.edu/projects/omega