

Особенности внутреннего представления системы САПФОР*

Н.А. Катаев

Институт Прикладной Математики имени М.В. Келдыша РАН

Система САПФОР помогает пользователю пройти полный путь распараллеливания программы от исследования последовательной версии программы через ее преобразование и получение потенциально параллельной версии до распараллеливания с использованием моделей DVM, DVMH, OpenMP. Тесное взаимодействие с пользователем на этапах анализа и преобразования последовательной программы предъявляет определенные требования к архитектуре САПФОР. Внутреннее представление системы должно обеспечивать возможность предоставления пользователю исчерпывающей информации о каждом этапе распараллеливания программы. В рамках проводимого исследования было разработано внутреннее представление САПФОР, единое для языков Фортран и Си.

1. Введение

Широкое распространение технологий параллельного программирования и, одновременно, трудность их повсеместного использования при написании прикладных программ ведут к необходимости создания средств параллельного программирования, автоматизирующих процесс разработки параллельных программ. Использование полностью автоматических средств, например, автоматически распараллеливающих компиляторов для мультипроцессоров, поставляемых компаниями Intel, Microsoft, Sun Microsystems, IBM и др. не достаточно для получения максимально эффективного параллельного кода для сложных вычислительных задач.

Возможным решением является создание систем, помогающих пользователю в процессе разработки параллельных программ. Такие системы позволяют контролировать корректность принимаемых пользователем решений, собирают информацию об узких местах в программе, обладающих недостаточным параллелизмом, проблемах, мешающих распараллеливанию. Крайне желательным является прогнозирование характеристик выполнения параллельных программ на вычислительных системах разной конфигурации, подбор различных параметров, например, решетки процессоров, для максимального использования ресурса вычислительных систем.

Системы, автоматизирующие процесс распараллеливания, могут предоставлять пользователю информацию рекомендательного характера относительно принятия решений о распараллеливании или преобразовании последовательной программы с целью устранения проблем, мешающих распараллеливанию.

Одним из средств облегчающих разработку параллельных программ является среда разработки параллельных приложений Intel® Parallel Studio [1]. Ее недостатком является то, что непосредственно распараллеливание выполняет пользователь. На основе проведенного анализа и профилирования последовательной программы среда лишь дает рекомендации и указывает места, на которые надо обратить внимание. Для пользователя этап распараллеливания является наиболее трудоемким, требует детального знания соответствующих технологий параллельного программирования и подвержен ошибкам, наличие которых хотя и может быть установлено средствами Intel® Parallel Studio, устранено все равно должно быть пользователем вручную.

Между тем наличие более точных данных о последовательной программе, полученных в результате анализа статического и динамического и в виде знаний, предоставляемых системе пользователем, позволяют проводить распараллеливание потенциально параллельной программы полностью автоматически. Данный подход применяется в системе САПФОР [2], взаимо-

* Работа поддержана грантом Президента РФ МК-6772.2012.9 и грантами РФФИ № 12-01-33003, 12-07-31205 и 12-07-31260.

действуя с которой пользователь подготавливает прикладную программу к распараллеливанию, не выходя за рамки последовательной версии программы.

2. САПФОР

Система Автоматизированного Распараллеливания Фортран Программ (САПФОР) разрабатывается в Институте Прикладной Математики им. М.В. Келдыша РАН. Входным языком системы является Fortran, а результат распараллеливания представляет собой программу на языке Fortran OpenMP, Fortran DVM/OpenMP или Fortran DVMH [3, 4, 5]. Все эти языки являются расширением стандартного языка Fortran 95 директивами параллелизма. Высокий уровень выходного языка позволяет продемонстрировать пользователю результат распараллеливания в понятных для него терминах, кроме того директивы распараллеливания могут быть проигнорированы компиляторами, не поддерживающими соответствующий язык. Для данных языков существуют развитые средства отладки функциональности и эффективности.

Отличительной особенностью САПФОР является использование автоматически распараллеливающего компилятора [6], преобразующего потенциально параллельную программу в ее параллельную версию для заданной ЭВМ. При этом предварительный анализ программы и приведение программы к потенциально параллельному виду может выполняться в полуавтоматическом режиме. Для уточнения свойств последовательной программы, выявленных анализатором, используется либо диалоговая оболочка, либо специальные аннотации в тексте программы. В САПФОР наиболее сложный этап распараллеливания (получение параллельной версии программы) выполняется полностью автоматически.

Основными компонентами САПФОР являются анализаторы последовательных программ, блоки преобразования последовательных программ в параллельные программы (эксперты), диалоговая оболочка для взаимодействия с пользователем, генератор кода, создающий на основе принятых экспертом решений параллельную версию программы.

Связующим звеном между всеми компонентами системы, обеспечивающим поддержку итерационного распараллеливания прикладных программ является внутреннее представление САПФОР. Схема САПФОР показана на **Рис. 1**.

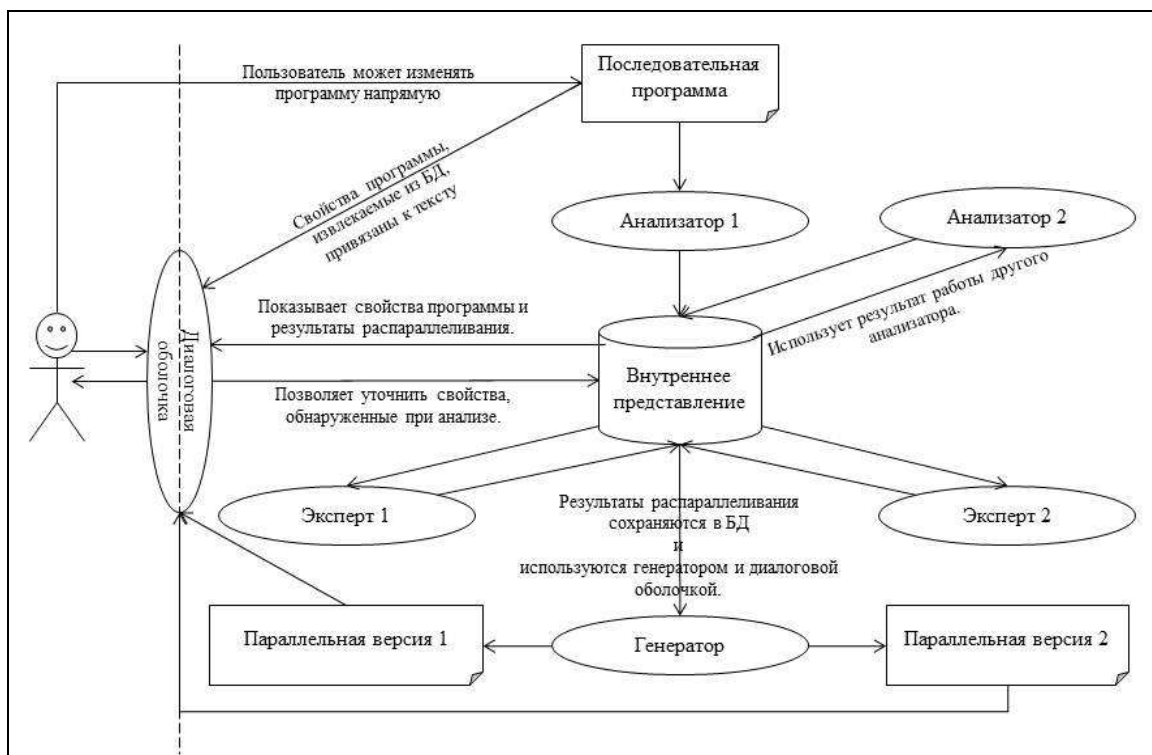


Рис. 1. Схема САПФОР

В статье рассматривается разработанное в рамках данного исследования Универсальное Высокоуровневое Внутреннее Представление (УВВП) процесса распараллеливания программы в системе САПФОР. УВВП САПФОР позволяет использовать в компонентах системы единые алгоритмы функционирования для языков программирования Fortran и C и обеспечивает возможность предоставления пользователю необходимой для распараллеливания информации в терминах исходной программы на протяжении всего пути распараллеливания прикладной программы пользователя.

Недостатки существующих представлений программ

В статье [7] был рассмотрен статический анализатор САПФОР, выполняющий анализ исходной программы на основе информации, содержащейся только в базе данных системы. Подчеркивалось, что данный анализатор потенциально не зависит от языка, на котором написана анализируемая программа. В анализаторе использовались общие алгоритмы анализа частных скалярных переменных, применимые как для программ, написанных на языке программирования Fortran, так и для программ, написанных на языке программирования C.

Применение единых алгоритмов для различных языков программирования возможно также при выявлении других особенностей программы [7], требующихся для эффективной работы эксперта САПФОР. Решение о распараллеливании прикладной программы также можно принимать независимо от исходного языка. Такой подход используется, например, при оптимизации программ в системе компиляторов GCC [8] и LLVM [9].

Использование особенностей соответствующего языка (синтаксиса, семантики) необходимо на стадии взаимодействия с пользователем и при получении конечной версии параллельной программы. В этом состоит отличие внутреннего представления САПФОР от низкоуровневых представлений GIMPLE GCC и IR LLVM. Низкоуровневые представления должны быть адаптированы для извлечения информации о программе в терминах языка программирования, на котором написана прикладная программа пользователя.

Попытка использования представления GIMPLE GCC совместно с существующей базой данных САПФОР была первым шагом к использованию единого внутреннего представления. Был разработан анализатор [10], использующий в качестве основного средства анализа Универсальную Библиотеку Трансляции (УБТ) [11] и использующий ее для заполнения базы данных САПФОР. В УБТ анализ программ выполняется на основе внутреннего представления GIMPLE системы компиляторов GCC.

В GIMPLE GCC одному оператору исходной программы в общем случае соответствует группа операторов GIMPLE. Более того использование формы однократного присваивания (SSA) и операторов простого вида приводит к появлению дополнительных переменных. При сохранении информации в базе данных все временные переменные должны быть удалены, а операторы приведены к виду операторов исходной программы.

В GIMPLE представлении все использующиеся в программе массивы линейризованы, но для корректного распараллеливания и распределения данных в языке Fortran DVM (C DVM) необходимо знать размерности массивов и используемые по каждому измерению индексы.

Также необходимо определение индуктивных переменных с границами и шагом и наличия тесной вложенности циклов. Для последнего необходимо восстановление высокоуровневых заголовков циклов программы из соответствующих им групп операторов GIMPLE.

Внутреннее представление должно отражать полный путь распараллеливания программы от исследования последовательной версии программы через ее преобразование и получение потенциально параллельной версии до распараллеливания с использованием моделей DVM, DVMH, OpenMP. Это делает невозможным использование GIMPLE GCC или IR LLVM без организации дополнительного средства хранения информации, отражающей результаты взаимодействия с пользователем, решения принимаемые компонентами САПФОР, выявленные особенности прикладной программы.

Для применения единых алгоритмов информации, содержащейся в существующей базе данных, было не достаточно. Основными недостатками использующейся базы данных были:

1. Отсутствие полной информации об исполняемых операторах программы. Для операторов сохранялась семантика доступа к памяти программы, но семантика испол-

няемых операций отсутствовала. В результате было невозможно, используя только базу данных, проанализировать ветвления в программе, оценить вероятность выполнения различных ветвей, выполнить распространение констант, определить индукционные переменные циклов.

2. Ориентированность базы данных на язык Fortran.
3. Невозможность хранения информации о точках и участках программы (например, циклах), когда данная информация зависит от пути выполнения программы, по которому данная точка или цикл были достигнуты. Путь выполнения программы определяется графом управления и графом вызовов,

В результате было разработано Универсальное Высокоуровневое Внутреннее Представление (УВВП) системы САПФОР, обладающее следующими особенностями:

1. Является связующим звеном между всеми компонентами системы.
2. Обеспечивает поддержание итерационного процесса распараллеливания программы. УВВП САПФОР содержит информацию о каждом этапе распараллеливания программы (результаты анализа: статического и динамического, выполняемые преобразования, указания пользователя, принятые экспертом решения о распараллеливании). С помощью диалоговой оболочки системы данная информация может быть наглядно представлена пользователю в терминах исходной программы. УВВП САПФОР содержит полное описание программы и позволяет восстановить исходный код программы, возможно, с учетом преобразований последовательной программы и директив распараллеливания, указанных экспертом САПФОР.
3. Не зависит от языка программирования. В качестве поддерживаемых языков рассматриваются языки Fortran и C, при этом используемые структуры данных предусматривают возможность адаптации УВВП САПФОР к другим языкам программирования.

Структура внутреннего представления

УВВП САПФОР состоит из четырех уровней, независимых от используемого в распараллеливаемой программе языка программирования:

1. Уровень исходного кода программ.
2. Уровень особенностей программы.
3. Уровень преобразования программы.
4. Уровень распараллеливания программы.

Уровень (1) исходного кода программ описывает прикладную программу в том виде, в каком она была написана пользователем системы САПФОР. Данного уровня достаточно для восстановления исходного кода программы. Уровень является единственным, заполнение которого зависит от используемого в распараллеливаемой программе языка программирования. Для заполнения данного уровня достаточно средств, разбирающих структуру программы (выполняющих лексический, синтаксический и семантический анализ), но не использующих сложные алгоритмы анализа, такие как поиск зависимостей по данным, поиск приватных переменных и др. В качестве front-end строящего первый уровень внутреннего представления для Fortran используется библиотека Sage++ [12], для C предполагается использовать возможности Clang [13] компилятора LLVM.

Уровень (2) особенностей программы описывает свойства программы, выявленные анализаторами системы (статическим и динамическим). Для всех циклов программы необходимо выявлять информацию следующего вида: зависимости по данным (flow, anti, output), регулярные зависимости по массивам, редукционные зависимости между витками цикла, приватные переменные (скаляры и массивы), индукционные переменные, время последовательного выполнения циклов (а также отдельных витков), обнаружение неявных циклов.

Для повышения эффективности проводимого распараллеливания, анализатором строится расширенное дерево циклов, в котором для каждой вершины-цикла прослеживается путь от входной точки программы. Каждому потенциально возможному выполнению цикла, определяемому последовательностью вызовов процедур (путем графа вызовов), соответствует отдельная вершина в расширенном дереве циклов. Для обнаруженных особенностей программы в УВВП

САПФОР осуществляется связывание с соответствующими вершинами расширенного дерева циклов, для которых данные особенности имеют место.

При необходимости анализ выполняется отдельно для каждой вершины расширенного дерева циклов. С точки зрения полноты информации, использование расширенного дерева циклов эквивалентно выполнению инлайн подстановки, при этом, данная операция скрыта от пользователя и ему предоставляется информация в терминах его исходной программы, но с указанием пути выполнения программы, по которому может быть достигнут соответствующий цикл.

Уровень преобразования программы описывает преобразования, последовательной версии программы, которые необходимы для выполнения наиболее эффективного распараллеливания. Для разных вариантов распараллеливания возможно использования различных наборов преобразований, приводящих программу к потенциально параллельному виду, пригодному для использования автоматического распараллеливающего компилятора, входящего в состав САПФОР. Необходимость выполнения тех или иных преобразований определяется тем вкладом, который они могут внести в повышение эффективности распараллеливания, оцениваемой экспертом при построении соответствующих вариантов распараллеливания. По каждому преобразованию может быть получена информация, объясняющая пользователю необходимость его применения.

Уровень распараллеливания программы содержит варианты распараллеливания, построенные экспертом, оценки эффективности параллельных версий программ, описание целевой архитектуры, рекомендованных параметров запуска (например, описание решетки процессоров), описание причин, снижающих эффективность распараллеливания программы.

На **Рис. 2** приведено соответствие между компонентами системы САПФОР и уровнями УВВП, за построение которых они отвечают.

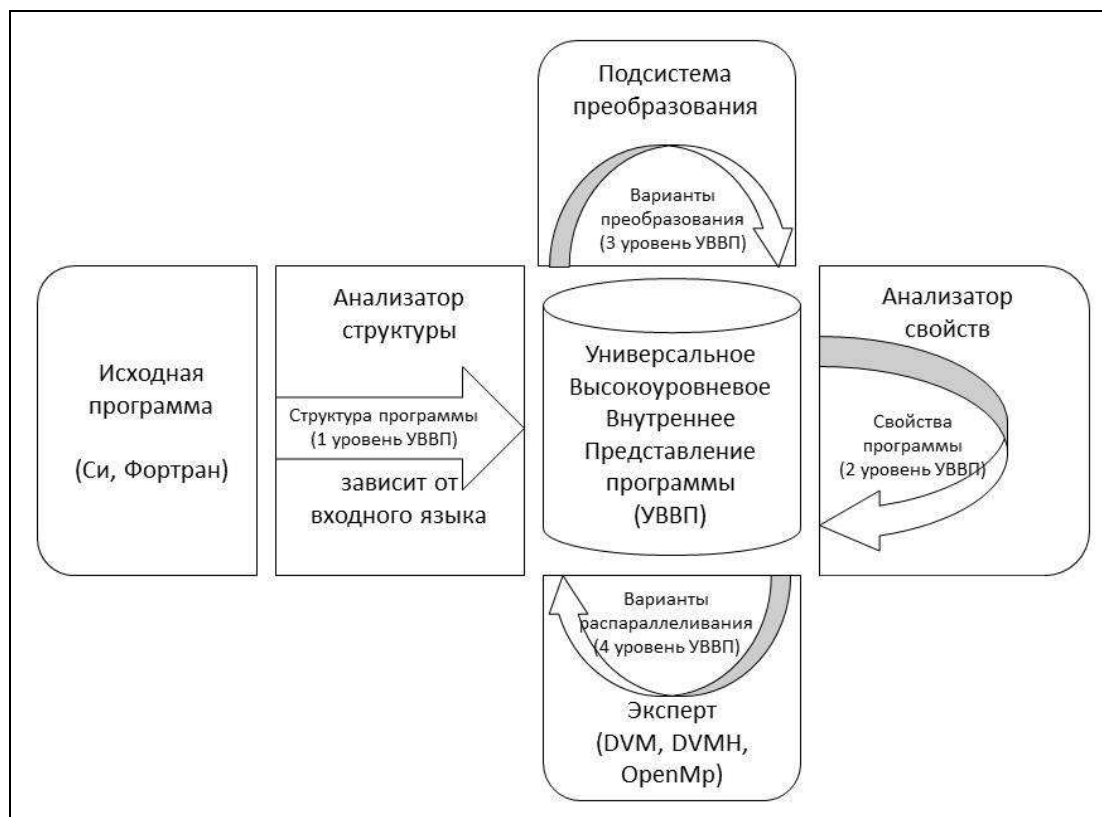


Рис. 2. Уровни УВВП САПФОР

Программная реализация

УВВП САПФОР хранится в виде SQL базы данных, построенной на основе свободно распространяемой библиотеки SQLite [14]. Такой способ реализации УВВП обусловлен тем, что:

1. Система САПФОР образована разными программными компонентами, не имеющими общего адресного пространства и асинхронно взаимодействующими между собой. Компоненты САПФОР могут быть разнесены в пространстве, например, динамический анализ можно запускать на суперкомпьютере для повышения производительности и точности анализа.
2. Во внутреннем представлении содержится большой объем количественных характеристик и других особенностей программы, к которым необходимо обеспечить удобный доступ, для чего может быть использован язык запросов SQL.
3. Использование механизмов SQLite позволяет контролировать непротиворечивость внутреннего представления при его модификации компонентами системы.

Для доступа к базе данных внутри компонент системы САПФОР разработан высокоуровневый интерфейс, представляющий собой надстройку над функциями библиотеки SQLite. Интерфейс реализован на языке программирования C++ в объектно-ориентированном стиле с применением идей метапрограммирования и скрывает от пользователей необходимость использования низкоуровневого механизма доступа к базе данных, предоставляемого библиотекой SQLite.

Интерфейс является отдельной библиотекой и может быть использован с другими базами данных. Интерфейс может быть легко адаптирован к изменениям в структуре базы данных САПФОР и позволяет использовать для доступа к информации, хранящейся в базе данных, имена таблиц и содержащихся в них столбцов, не задумываясь об особенностях реализации базы данных. Это позволяет обеспечить обратную совместимость предыдущих версий компонентов САПФОР и внутреннего представления системы при его модификации. Например, добавление новых таблиц или столбцов в таблицы не повлияет на работу компонентов системы. Интерфейс настраивается под особенности компонентов системы.

Все структуры данных разработанного интерфейса сопровождаются комментариями в формате, поддерживаемом средством автоматической генерации документации Doxygen.

Заключение

Для поддержания процесса распараллеливания при активном участии пользователя в системе САПФОР было разработано Универсальное Высокоуровневое Внутреннее Представление (УВВП) единое для языков программирования Fortran и C.

Внутреннее представление реализовано в виде базы данных, что позволяет использовать SQL-запросы для быстрого поиска необходимой информации о программе и поддерживать непротиворечивость структур, описывающих состояние процесса распараллеливания.

Высокий уровень внутреннего представления позволяет реализовать взаимодействие с пользователем в терминах его исходной программы, и обеспечивает удобство использования системы САПФОР при распараллеливании прикладных программ.

Независимость внутреннего представления от языка программирования позволяет использовать общие механизмы анализа, преобразования и распараллеливания для языков программирования Fortran и C. Это необходимо для развития системы САПФОР, изначально ориентированной на язык программирования Fortran, в направлении автоматизированного распараллеливания программ, написанных на языке программирования C.

Литература

1. Intel® Parallel Studio. URL: <http://software.intel.com/en-us/intel-parallel-studio-home> (дата обращения: 01.12.2012)
2. Система Автоматизированной Параллелизации Фортран Программ. URL: <http://www.keldysh.ru/dvm/SAPFOR/> (дата обращения: 01.12.2012).
3. Бахтин В.А., Коновалов Н.А., Крюков В.А., Поддерюгина Н.В. Fortran OpenMP/DVM – язык параллельного программирования для кластеров // Материалы второго Международ-

ного научно-практического семинара “Высокопроизводительные параллельные вычисления на кластерных системах”, г. Нижний Новгород, 26-29 ноября 2002 г., с.28-30.

4. Бахтин В.А., Коновалов Н.А., Поддерюгина Н.В., Устюгов С.Д. Гибридный способ программирования DVM/OpenMP на SMP-кластерах // Труды Всероссийской научной конференции “Научный сервис в сети Интернет: технологии параллельного программирования” (сентябрь 2006 г., г. Новороссийск), Изд-во Московского Университета, с.128-130.
5. Бахтин В.А., Клинов М.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Сазанов Ю.Л. Расширение DVM-модели параллельного программирования для кластеров с гетерогенными узлами // Труды Международной суперкомпьютерной конференции “Научный сервис в сети Интернет: экзафлопсное будущее” (19-24 сентября 2011 г., г. Новороссийск). - М.: Изд-во МГУ, с. 310-315.
6. Крюков В.А., Клинов М.С., Бахтин В.А., Поддерюгина Н.В. Автоматическое распараллеливание последовательных программ для многоядерных кластеров // Труды Международной суперкомпьютерной конференции “Научный сервис в сети Интернет: суперкомпьютерные центры и задачи” (20-25 сентября 2010 г., г. Новороссийск). - М.: Изд-во МГУ, с. 12-15.
7. Катаев Н.А. Статический анализ последовательных программ в системе автоматизированного распараллеливания САПФОР // Труды международной научной конференции “Параллельные вычислительные технологии (ПаВТ’2012)” (Новосибирск, 26 – 30 марта 2012 г.) – Челябинск: Издательский центр ЮУрГУ, 2012, с. 179-190
8. GCC, the GNU Compiler Collection. URL: <http://gcc.gnu.org/> (дата обращения: 01.12.2012).
9. The LLVM Compiler Infrastructure. URL: <http://llvm.org/> (дата обращения: 01.12.2012).
10. Катаев Н.А. Анализ последовательных программ с помощью средств УБТ // Труды международной научной конференции “Параллельные вычислительные технологии (ПаВТ’2011)” (Москва, 28 марта – 1 апреля 2011 г.) – Челябинск: Издательский центр ЮУрГУ, 2011, с. 697.
11. Optimizing Technologies. URL: <http://www.optimitech.com/> (дата обращения: 01.12.2012).
12. pC++/Sage++ Home Page. URL: <http://www.extreme.indiana.edu/sage/> (дата обращения: 01.12.2012).
13. Clang. URL: <http://clang.llvm.org/> (дата обращения: 01.12.2012).
14. SQLite. URL: <http://www.sqlite.org/> (дата обращения: 01.12.2012).