

УДК 681.3.06

А. И. Толкачёв

ПРИМЕНЕНИЕ ПРОГРАММНОГО КОМПЛЕКСА РАЗРАБОТКИ КОМПИЛЯТОРОВ ДЛЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ПРОЦЕССЕ ОБУЧЕНИЯ СПЕЦИАЛИСТОВ

Гомельский государственный университет имени Франциска Скорины, Республика Беларусь

Вступление

В последнее время наблюдается большой интерес разработчиков программного и аппаратного обеспечения вычислительных систем к использованию параллельной обработки.

Это обусловлено, во-первых, быстрым развитием сфер применения встроенных систем и специализированных цифровых устройств. К таким системам постоянно предъявляются новые требования производительности и возможности решения различных задач, таких как обработка звука, изображений, видео, шифрование и сжатие данных и т.д.

Во-вторых, использование параллельной обработки является единственным способом решения некоторых задач, требующих огромных объемов вычислений, например, создание качественного видеоизображения, прогнозирование погоды, моделирование физических процессов.

Разработанный нами программный комплекс проектирования компиляторов языков высокого уровня для параллельных вычислительных систем может использоваться для обучения специалистов в области параллельного программирования. В основе программного комплекса лежит алгоритм автоматического выявления параллелизма [1, 2], реализованный в виде универсального модуля. Настройка на целевую платформу осуществляется реализацией *back-end* компилятора, принимающего на входе максимально распараллеленное промежуточное представление.

Проблемы проектирования программного обеспечения для параллельных архитектур

Основная проблемы, встающие при разработке программ для параллельных архитектур, заключаются, во-первых, в эффективности использования имеющихся ресурсов и, во-вторых, в сложности написания программ, максимально эффективно использующих возможности целевой архитектуры.

Системы с распределенными ресурсами включают в себя множество различных вариантов конфигураций вычислительных систем, и для каждой из них существуют различные критерии эффективности. Если для традиционных последовательно работающих процессоров критерии оптимизации примерно одинаковы, то для параллельных систем они могут существенно различаться. Например, в многопроцессорном компьютере может потребоваться оптимизация обращений к общей памяти, а в вычислительной системе, состоящей из множества компьютеров, работающих в локальной сети, узким местом может быть скорость передачи данных между узлами.

Перечисленные проблемы приводят к тому, что для разработки программ для параллельных вычислительных систем используются различные методы и инструменты.

Существующие средства для программирования вычислительных систем с распределенными ресурсами

Существует множество систем для создания программного обеспечения для систем с распределенными ресурсами. Можно выделить несколько классов средств разработки. Самыми многочисленными являются системы, представляющие собой библиотеки для некоторого языка программирования высокого уровня. Они, как правило, используются при программировании для систем, состоящих из множества параллельно работающих компьютеров, и содержат функции для синхронизации процессов и передачи данных между ними. Вся работа по распараллеливанию алгоритма возлагается на программиста. Самой распространенной системой такого типа является *MPI* [6] (*Message*

Passing Interface, інтерфейс для передачі повідомлень).

Друга технологія – *OpenMP* [6], може використовуватися в багатопроцесорних системах з загальною пам'яттю. Ідея заключається в тому, що певні фрагменти програми можуть бути оголошені паралельними, і тільки вони можуть виконуватися одночасно. Завжди існує основний процес, який створює паралельні і очікує закінчення їх роботи. Як і в разі з *MPI*, розпаралелювання здійснюється програмістом.

З систем, використовуючих розширення мов програмування, розглянемо *mpC* [6]. Програма на *mpC* являє собою опис множини процесів, взаємодіючих з допомогою неявної передачі повідомлень. Розподіл процесів між фізичними процесорами відбувається автоматично в процесі виконання програми.

Засоби для програмування суперскалярних і *VLIW* процесорів являють собою оптимізуючі компілятори або транслятори асемблера, розроблювані або налаштовані спеціально для кожної архітектури. Можливості паралельної обробки *VLIW* процесорів мають суттєві обмеження, що дозволяють оптимізувати програму тільки на рівні інструкцій.

Актуальні проблеми, мета дослідження

Актуальною є проблема автоматизації проектування програмного забезпечення для паралельних архітектур. Метою дослідження являлося створення універсальних методів компіляції мов високого рівня для паралельних висувальних систем, а також реалізація програмного комплексу для автоматизованого проектування компіляторів.

В ході дослідження вирішувалися наступні проблеми:
 автоматизація розробки синтаксических аналізаторів входних мов високого рівня;
 створення формату проміжного представлення програми, яке може містити інформацію про паралелізм, використовуватися для застосування різних алгоритмів оптимізації, а також є універсальним по відношенню до входної мови і цільової платформи;
 розробка алгоритмів автоматичного виявлення паралелізму, що визначають можливості паралельного виконання як на рівні окремих інструкцій, так і цілих блоків вихідної програми;
 алгоритми оптимізації, що дозволяють розробляти ефективні компілятори для паралельних висувальних систем з різними обмеженнями і вузькими місцями;
 інтеграція отриманих компіляторів з універсальною відлагодочною середовищем для розробки програмного і апаратного забезпечення вбудованих систем.

Пропонується технологія проектування компіляторів для систем з розподіленими ресурсами

Основна ідея проектування компіляторів для висувальних систем з розподіленими ресурсами заключається в створенні універсальної незалежної від вихідної мови і цільової платформи частини компілятора. Цей блок компілятора (розпаралелюючий компілятор) здійснює аналіз програми і виявляє всі можливості паралельного виконання як на рівні простіших операторів (арифметических, логіческих), так і цілих блоків програми (циклів, функцій і др.). Вхідними даними для роботи розпаралелюючого компілятора є проміжне представлення вихідної програми, отримане в результаті роботи *front-end* компілятора. Отримане на виході проміжне представлення виконуваного коду містить інформацію про всі знайдені можливості паралельного виконання. Генерацію виконуваного коду для заданої цільової платформи виконує *back-end* компілятор, що отримує розпаралелене представлення програми.

Пропонується схема універсального компілятора зображена на малюнку 1.

Розроблений метод проектування компіляторів для висувальних систем з розподіленими ресурсами покриває весь цикл розробки компілятора, починаючи з синтаксического аналізу вихідної програми і закінчуючи генерацією виконуваного коду.

Основними етапами проектування компілятора є:
 Розробка компілятора вихідного ЯВУ в послідовне проміжне представлення. Для реалізації синтаксического аналізатора вихідної мови використовується програмний комплекс *Unisan* [3].

Реалізація алгоритмів розпаралелювання. Результатом їх роботи є проміжне представлення програми, що містить інформацію про всі знайдені можливості паралельного виконання.

Створення платформено – незалежного оптимізатора проміжного представлення, що містить інформацію про паралелізм.

Реалізація оптимізатора, що враховує особливості цільової архітектури.

Разработка генератора исполняемого кода для целевой платформы.

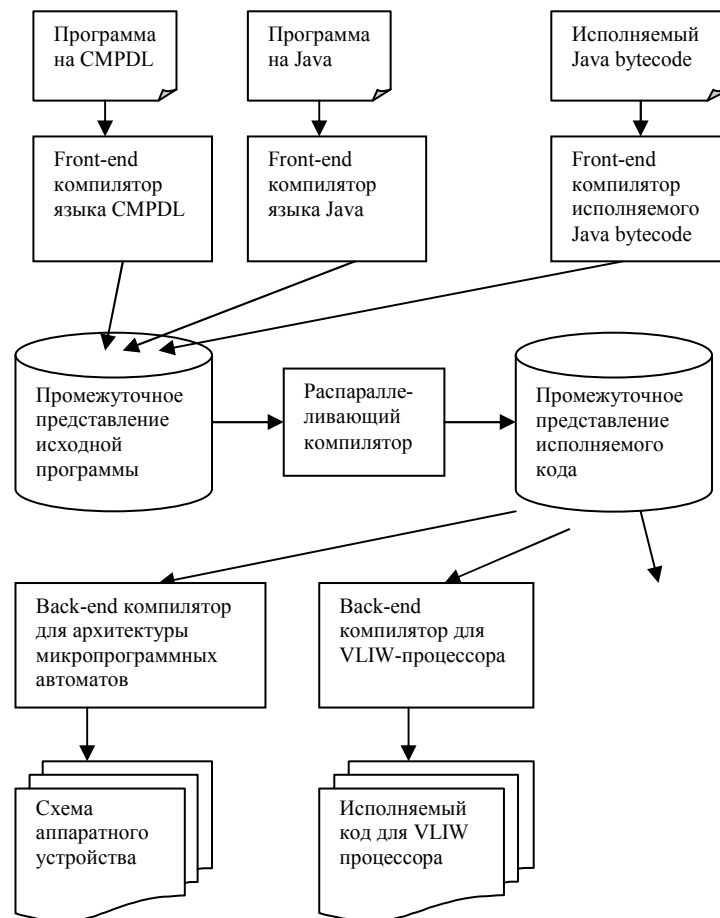


Рис. 1. Схема настраиваемого компилятора языков высокого уровня для параллельных вычислительных систем

Новизна метода проектирования распараллеливающих компиляторов заключается в следующем:

Алгоритм факторизации синтаксических диаграмм в универсальном синтаксическом анализаторе *Unisan* позволяет значительно расширить допустимый класс грамматик, и, соответственно, уменьшить трудоемкость описания синтаксиса входного языка.

Универсальные алгоритмы распараллеливания позволяют выявлять возможности параллельного исполнения как на уровне отдельных инструкций, так и целых блоков исходной программы.

Формат промежуточного представления программы может содержать информацию о параллелизме, использоваться для применения различных алгоритмов оптимизации, а так же является универсальным по отношению к входному языку и целевой платформе.

Получаемая в результате работы распараллеливающего компилятора информация может быть использована для генерации исполняемого кода для широкого класса архитектур с распределенными ресурсами.

Алгоритмы оптимизации позволяют разрабатывать оптимизирующие компиляторы для параллельных вычислительных систем с различными ограничениями и узкими местами.

Интеграция с универсальной отладочной средой для разработки программного и аппаратного обеспечения встроенных систем снижает трудоемкость и стоимость разработки программного или аппаратного обеспечения, использующего параллельную обработку.

Применение разработанных средств для обучения специалистов

Благодаря возможности повторного использования универсальных блоков компилятора значительно снижены необходимые объем работы и трудоемкость реализации оптимизирующих компиляторов для вычислительных систем с распределенными ресурсами. Это позволяет применять программный комплекс для обучения специалистов в области проектирования компиляторов для параллельных вычислительных систем.

В качестве апробации разработанных средств в Гомельском государственном университете реализован компилятор языка *CMPDL* [4] (подмножество языка Си) для архитектуры микропрограммных автоматов [4]. Результатом работы компилятора является синтезируемое описание аппаратной схемы, которая реализует описанный алгоритм с использованием параллельной обработки. Интеграция компилятора с системами высокоуровневого проектирования аппаратного и программного обеспечения встроенных систем *Winter* и *IEESD* [5] позволяет разработчикам моделировать и отлаживать работу устройства с использованием программных моделей процессора микропрограммных автоматов и устройств окружения, а затем получать синтезируемое описание устройства на языке *Vhdl* [7]. Синтезируемое описание может быть аппаратно реализовано в ПЛИС или СБИС.

Программный комплекс успешно используется в процессе обучения студентов специальности “Прикладная математика” по курсу “Проектирование цифровых устройств”.

Выводы, перспективы дальнейшего развития

Разработанная технология проектирования компиляторов для систем с распределенными ресурсами может применяться для создания компиляторов практически любых языков программирования. В качестве целевой платформы возможно использование широкого класса вычислительных систем с распределенными ресурсами, в частности, VLIW процессоров, многопроцессорных систем, генератора аппаратных схем и др.

Практическая значимость заключается в значительном снижении трудоемкости реализации оптимизирующих компиляторов для параллельных вычислительных систем по сравнению с другими средствами проектирования компиляторов. Разработанный программный комплекс может использоваться для обучения специалистов.

Ограничение технологии состоит в отсутствии встроенных средств коммуникации при использовании многопроцессорных целевых архитектур или вычислительных систем, состоящих из множества компьютеров. При создании компиляторов для таких архитектур необходимо дополнительно реализовывать коммуникационные средства для взаимодействия компонент вычислительной системы. Разработка и программная реализация универсальных средств коммуникации является одной из перспектив развития.

Другим направлением развития является создание универсальных средств, автоматизирующих разработку эффективных генераторов исполняемого кода для целевых архитектур со специфическими инструкциями, не имеющими аналогов в промежуточном представлении.

Список литературы

1. Tolkachiov A., Dolinsky M. Hardware Implementation of Complex Data Processing Algorithms // Proceedings of the Work in Progress Session held in Connection with the 29th EUROMICRO Conference EUROMICRO 2003 and the EUROMICRO Symposium on Digital System Design DSD 2003, Belek (Turkey), September 2003, Published by Institute of System Science Johannes Kepler Universitat Linz, Austria, ISBN 3-902457-21-X.
2. Толкачѳв А.И. Комплекс для проектирования аппаратных решений, эффективно реализующих сложные алгоритмы обработки данных // Известия Гомельского государственного университета имени Ф.Скорины. – 2003. – №3(18), – с.153-157.
3. Толкачѳв А.И. Использование программного комплекса автоматизированной разработки трансляторов в учебном процессе // Интернет-Образование-Наука-2002: Материалы 3-й международной конференции – Винница: Універсум-Вінниця – с. 100-102.
4. Долинский М., Коршунов И., Толкачев А., Ермолаев И., Литвинов В. Технология разработки алгоритмически сложных цифровых систем с помощью автоматического синтеза микропрограммных автоматов // Компоненты и технологии. – 2003. – №8. – с.110-114.
5. Долинский М.С., Ермолаев И.Ю., Гончаренко И.И., Толкачѳв А.И. Winter - среда отладки программного обеспечения мультипроцессорных систем // Компоненты и технологии. – 2003. №2, – с.63-69.
6. В.В. Воеводин, Вл.В.Воеводин Параллельные вычисления. – СПб.: БВХ-Петербург, 2002.
7. Chang K.C. Digital Design and Modeling with VHDL and Synthesis // IEEE Computer Society Press, Los Alamitos, California. - Los Alamitos, California. – 1997.

Алексей Иванович Толкачѳв, Гомельский Государственный университет имени Франциска Скорины, ул. Советская, 104, Гомель, 246000, Республика Беларусь, Tel. +(375) 232 578256. E-mail: tolkachyov@gsu.unibel.by