

ПОДХОД К АВТОМАТИЗИРОВАННОМУ РАСПАРАЛЛЕЛИВАНИЮ В СИСТЕМЕ САПФОР

М.С. Клинов

1. Введение

Согласно ресурсу [1]: распараллеливание программ – это процесс адаптации алгоритмов, записанных в виде программ, для их эффективного исполнения на вычислительной системе параллельной архитектуры. Распараллеливание может быть ручным, автоматическим и полуавтоматическим (автоматизированным). С точки зрения получения наиболее эффективных программ ручное распараллеливание является предпочтительным. Сравнение эффективности результатов ручного программирования и автоматизированного имеется в работах [2, 3]. Но вместе с тем, ручное распараллеливание предъявляет к программисту большие требования к владению современными технологиями и языками параллельного программирования, а также к опыту работы с текстами программ, написанными другими программистами, и владению инструментами отладки параллельных программ. При автоматическом распараллеливании от программиста никаких особых навыков не требуется. При автоматизированном распараллеливании либо вообще не требуются знания современных технологий и языков параллельного программирования, либо требуются не в таком объеме как при ручном распараллеливании, но опыт работы с текстами программ и инструментами отладки необходим.

Путем автоматического распараллеливания можно для некоторых программ получить хорошие результаты распараллеливания, например при использовании автоматически распараллеливающего компилятора, входящего в состав системы САПФОР [2, 3], однако для реальных приложений применение средств автоматического распараллеливания дает низкую эффективность. Требуются, как правило, преобразования пользователем последовательных программ и спецификация им ее свойств. Это переводит распараллеливание из области автоматического в область полуавтоматического, которую также называют областью автоматизированного распараллеливания.

На настоящий момент нельзя утверждать, что сам процесс автоматизированного распараллеливания быстрее ручного, подобных исследований пока нет. Понятно, что на некоторых этапах работы, например, на этапе первичного анализа программы или этапе вставки распараллеливающих строк (операций или специальных комментариев) система может сделать это быстрее программиста. Но на этапах преобразования программы или детального анализа программы, при котором автоматическими методами свойства программы не определяются, в выигрышном положении находится программист по сравнению с, как правило, менее опытным в программировании пользователем системы распараллеливания, в т.ч. и за счет того, что программист вынужден более глубоко вникать в текст программы на всех этапах распараллеливания.

При распараллеливании программ существует фактор внесения ошибок в параллельную программу. Пользователь системы распараллеливания может ошибиться в преобразованиях последовательной программы или, что труднее обнаружить, в свойствах последовательной программы. Свойства программы пользователь задает в случае, если система распараллеливания не может их определить автоматически, поэтому система не может их проверить. При ручном распараллеливании программист может ошибиться в распараллеливающих конструкциях. Большинство ошибок распараллеливания можно определить и локализовать путем отладки полученных параллельных программ. Исключения составляют ошибки, которые проявляются при определенных входных данных, для отладки программ на которых инструментами отладки не хватает ресурсов ЭВМ.

Системы автоматизированного распараллеливания могут рассматриваться как средства упрощения процесса распараллеливания по сравнению с ручным распараллеливанием для опытных программистов и как среды разработки параллельных программ для менее опытных программистов.

2. Система САПФОР

Система САПФОР – это система автоматизированного распараллеливания Фортран-программ [4]. Система ориентирована на получение эффективных параллельных программ, под которыми понимаются такие программы, эффективность которых сравнима с эффективностью программ, разработанных при ручном распараллеливании с использованием языков параллельного программирования.

Архитектура системы САПФОР:

- Анализаторы (статический и динамический),
- Блоки распараллеливания,
- Блок создания текста параллельных программ,
- Диалоговая оболочка.

Связующим звеном между блоками является база данных системы САПФОР, которая представлена в системе в виде файла с операциями доступа к нему.

В текущей версии системы САПФОР из анализаторов имеется только статический, среди блоков распараллеливания имеются 4 блока для следующих типов ЭВМ:

- кластер (блок использует для распараллеливания программы язык Fortran-DVM),
- многоядерный процессор (соответственно Fortran-OpenMP),
- многоядерный кластер (соответственно Fortran-DVM/OpenMP),
- кластер с графическими ускорителями (соответственно Fortran-DVMH).

3. Подход системы САПФОР

Подход системы САПФОР основан на следующих тезисах.

- У пользователя системы распараллеливания есть отлаженная последовательная программа в том смысле, что она работает на некоторых входных данных, не содержит ошибок, таких как наличие неинициализированных данных, и результаты ее работы не сильно (приемлемо для пользователя) зависят от опций компиляции.
- В системе есть автоматический анализ последовательной программы.
- Система строит варианты распараллеливания.
- Система оценивает варианты распараллеливания путем прогнозирования характеристик параллельного выполнения.
- Система показывает пользователю причины, по которым некоторые фрагменты программы распараллелить не удалось.
- Пользователь системы выбирает вариант распараллеливания либо по лучшему согласно прогнозу времени параллельного выполнения программы, либо по другим критериям.
- Система строит текст параллельной программы по выбранному пользователем варианту распараллеливания.
- Пользователь может изменять последовательную программу для того, чтобы система могла распараллелить ее целиком или основные ее фрагменты.
- Пользователь имеет возможность задавать свойства последовательной программы, которые не удалось системе определить автоматически.

Сделаем несколько необходимых пояснений к этим тезисам, чтобы раскрыть подход системы САПФОР в полной мере.

К тезису про наличие в системе автоматического анализа последовательной программы относится вопрос ввода программы в систему. Программа может состоять из нескольких файлов, и при анализе одного файла могут потребоваться результаты анализа другого файла. У пользователя системы помимо последовательной программы должен быть еще и проект ее компиляции (проект Visual Studio, makefile, бат-файл), необходимый для запуска программы. В рамках подхода система САПФОР должна поддерживать такие форматы задания программы из нескольких файлов.

Анализ должен записать в базу данных результаты анализа за определенное время, приемлемое для пользователя. Для этого можно иметь несколько видов анализа от более грубого до более точного. При этом результаты анализа могут отсутствовать, если времени не хватило даже для грубого анализа.

Распараллеливание после автоматического анализа вынуждено быть консервативным, т.е. в случае, когда блок анализа не уверен в наличии свойства, блок распараллеливания должен исходить из того, что свойство может как быть в программе, так и не быть в ней. Анализатор должен специальным образом пометить такие свойства. Это полезно при детальном рассмотрении пользователем результатов автоматического анализа.

Блок распараллеливания также должен укладываться в определенное пользователем время работы. Блок распараллеливания не только строит варианты распараллеливания, но и оценивает их путем прогнозирования характеристик параллельного выполнения. При прогнозе учитываются такие аспекты выполнения программы как настройки параметров запуска, например организация процессов в виде логической многомерной решетки процессов, и возможность использования не всех ресурсов ЭВМ, заданных пользователем или определенных путем поиска оптимального размера ресурсов. За счет этого прогноз может быть более грубым или более детальным. При этом, также как и при анализе программ, результаты работы блока могут не успеть сформироваться и записаться в базу данных.

Блок распараллеливания по каждому распараллеленному фрагменту программы, например циклу, записывает в базу данных информацию о дополнительных операциях, необходимых для организации параллельных вычислений, а диалоговая оболочка показывает ее пользователю. Эти операции требуют определенного времени для своего выполнения и с точки зрения программы будут накладными расходами, необходимыми для ее распараллеливания.

После получения пользователем текста программы дальнейшая работа с ней не поддерживается системой САПФОР. Пользователь должен скопировать текст программы на ЭВМ, скомпилировать его на ней, выполнить запуски программы при необходимости с разными настройками параметров запуска, анализировать эффективность выполнения программы, сравнить результаты параллельного и последовательного выполнения программы, и в случае отличия результатов отлаживать программу при помощи инструментов отладки параллельных программ.

Тут есть момент, который может вернуть пользователя к необходимости воспользоваться системой САПФОР. В случае отличия результатов параллельного и последовательного выполнения программы одной из причин может быть неверное указание пользователем свойств программы. В таком случае пользователю нужно указать правильное свойство и воспользоваться системой САПФОР для построения параллельной программы, потому что указанное свойство может повлиять на несколько фрагментов программы.

К тезису, что пользователь может изменять последовательную программу, относится вопрос определения, какие фрагменты нужно менять, а какие не стоит. Если прогнозируемая эффективность или реальная эффективность (по запускам программы на ЭВМ) не устраивает пользователя, то он начинает просматривать через диалоговую оболочку причины, по которым некоторые фрагменты программы распараллелить не удалось, анализировать степень того, насколько они помешали распараллеливанию программы, и придумывать способы их устранения. Затем изменяет последовательную программу и сравнивает результаты ее работы с исходной программой (или откладывает это на потом в расчете на внесение нескольких правок в программу). При любом изменении программы возникает аспект корректности внесенных изменений. Можно быстро проверить, используя малые размеры входных данных, но это не будет показательным для других входных данных, а в свою очередь верификация по текстам программ или на больших размерах данных может быть затруднительной. Этот вопрос является актуальным и для ручного распараллеливания.

Некоторые преобразования возможно производить автоматически, некоторые можно автоматизировать и выполнять по указаниям пользователя и по заданным им параметрам (например, для изменения формы массива). Количество различных необходимых преобразований является большим и это приводит к тому, что в рамках подхода системы САПФОР не подразумевается автоматизация их всех и отсутствие работы пользователя с текстом программы при его изменении.

При работе с диалоговой оболочкой пользователь может не менять текст программы, а описывать свойства будущей программы без получения ее текста. По полученной информации могут работать блоки распараллеливания и строить оценки параллельного выполнения. Таким образом можно оценить целесообразность изменений программы или сравнить разные способы изменения программы.

Не обязательно устранять и в крайне редких случаях возможно устранить все причины нераспараллеленности всех фрагментов программы. Некоторые причины принципиально не могут быть устранены, например: потоковая зависимость в линейном массиве, ввод-вывод массивов (обычно делается через один процессор ввода-вывода), многоуровневый параллелизм в некоторых языках параллельного программирования запрещен или существенно ограничен (а в некоторых случаях и не целесообразен, например, когда соответствующие циклы требуют больших накладных расходов на их распараллеливание).

К тезису о коррекции результатов анализа подход системы САПФОР предусматривает наличие удобных механизмов, реализованных через возможности диалоговой оболочки. Если система не может автоматически определить свойства последовательной программы, то может по крайней мере удобно визуализировать информацию, например, чтение и запись в элементы массива с учетом вызова процедур, по которой пользователь сможет определить свойство проще, чем только по тексту программы. Свойство можно задать через диалоговую оболочку для определенного фрагмента программы или для нескольких фрагментов сразу. В этом смысле диалоговая оболочка предоставляет более удобные средства. Все заданные свойства можно сохранить через специальные комментарии, которые вставляются в текст программы и прикрепляются, таким образом, к нему для случая последующей модификации текста, чтобы не требовалось снова задавать свойства фрагментов программы.

4. Степень реализации системы САПФОР

Система САПФОР на данный момент не реализована в той мере, чтобы поддерживать изложенный подход полностью. Существует экспериментальная версия для Фортран-программ, в которой реализованы некоторые возможности подхода.

Экспериментальная версия системы САПФОР не оценивает варианты распараллеливания программы для многоядерного процессора, многоядерного кластера и кластера с графическими ускорителями. В процессе работы соответствующих блоков строится по одному варианту распараллеливания и оценка вариантов не является необходимой для выбора варианта. Хотя, безусловно, полезна для пользователя для оценки ускорения. При построении варианта распараллеливания на многоядерный кластер сначала строятся варианты распараллеливания программы по узлам ЭВМ, они оцениваются исходя из использования определенных ресурсов ЭВМ, выбирается вариант с наилучшим временем выполнения, затем для него строится вариант распараллеливания программы внутри узла.

Причины, по которым некоторые фрагменты программы распараллелить не удалось, строятся только для случая распараллеливания программы на кластер.

В экспериментальной версии не реализована поддержка файлов проектов компиляции программы, которые могли бы использоваться ею для определения порядка анализа файлов для случая, когда программа состоит из нескольких файлов.

Блоки анализа и блоки распараллеливания, внутри которых производится в т.ч. оценка вариантов распараллеливания, работают не в режиме постепенного уточнения. В частности, блок распараллеливания может не обработать корректно, если анализ был прерван. Степень реализации блоков анализа и блоков

распараллеливания не позволяет говорить о том, что они успешно отработают на всех программах, с которыми система САПФОР должна на данном этапе развития справляться. А сужать класс программ, которые можно подавать на вход, до очень узкого класса программ, с которыми экспериментальная система гарантировано справлялась бы, нецелесообразно.

Блок распараллеливания не записывает в базу данных в удобном для диалоговой оболочки виде информацию об операциях, необходимых для организации параллельных вычислений. Он записывает ее в виде текста специальных комментариев и диалоговая оболочка не показывает их пользователю.

Экспериментальная версия показывает пользователю причины, по которым некоторые фрагменты программы распараллелить не удалось, но не позволяет ему удобными средствами оценить, насколько каждая из причин помешала распараллеливанию программы. Нет в системе автоматических или автоматизированных механизмов преобразования текста последовательной программы, не поддерживаются механизмы оценки целесообразности изменений последовательной программы.

В экспериментальной версии нет удобных для пользователя механизмов автоматизированного поиска свойств программы и их задания. Некоторые свойства система может обнаружить, остальные свойства пользователь вынужден определять по тексту программы.

На этом описание нереализованных идей подхода заканчивается.

На основе реализованной экспериментальной версии системы САПФОР получены результаты распараллеливания нескольких программ, некоторые из которых являются реальными приложениями. Эффективность полученных программ была близка к эффективности вручную распараллеленных программ [2, 3]. В качестве вручную распараллеленных программ были взяты программы, разработанные при помощи языков Fortran-DVM и Fortran-DVMH. В качестве входных программ были использованы либо Fortran-DVM-программы с убранными из них DVM-директивами, либо последовательные программы пользователя, которые в процессе работы пользователя с системой САПФОР преобразовывались им в несколько этапов.

5. Заключение

Система автоматизированного распараллеливания должна быть удобной для пользователя и предоставлять ему все возможные инструменты, которые могут существенно облегчить ему работу.

В рамках подхода системы САПФОР пользователь работает в терминах последовательной программы: не имеет возможности управлять процессом при построении вариантов распараллеливания, выбирает из уже построенных вариантов, может изменять последовательную программу и специфицировать в необходимом объеме ее свойства. Построение вариантов распараллеливания делает система.

Реализация такого подхода с учетом необходимости разработки инструментов и удобной диалоговой оболочки является объемной задачей. На данный момент эта задача решена частично в виде экспериментальной версии системы САПФОР, которая была опробована при распараллеливании реальных приложений. Результаты распараллеливания являются сравнимыми по эффективности с результатами ручного распараллеливания.

Экспериментальная версия системы САПФОР способна, гарантировано, эффективно распараллеливать очень узкий класс программ, формулировать который, однако, бессмысленно, потому что в него не попадают большинство реальных приложений, на которые ориентирована система. Для поддержки реальных приложений требуется отладка блоков анализа и блоков распараллеливания для стабильной их работы, возможно за счет менее эффективных алгоритмов. Такие работы ведутся в рамках реализации новой версии системы САПФОР.

В настоящее время ведутся работы по реализации системы САПФОР для поддержки распараллеливания программ на языке Си, а не только программ на языке Фортран, и исследование возможностей автоматического определения и выполнения преобразований фрагментов текстов последовательных программ, необходимых для эффективного распараллеливания программ при помощи системы САПФОР.

Работа поддержана грантами Президента РФ МК-6772.2012.9 и НШ-4307.2012.9, грантами РФФИ № 12-01-33003, № 12-07-31205, № 12-07-31260 и № 13-07-00402.

ЛИТЕРАТУРА:

1. “Распараллеливание программ”, URL: http://ru.wikipedia.org/wiki/Распараллеливание_программ (дата обращения 27.05.2013)
2. В.А. Бахтин, М.С. Клинов, В.А. Крюков, Н.В. Поддерюгина “Автоматическое распараллеливание последовательных программ для многоядерных кластеров” // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: Труды Международной суперкомпьютерной конференции (20-25 сентября 2010 г., г. Новороссийск), М.: Изд-во МГУ, 2010, С. 12-15
3. В.А. Бахтин, И.Г. Бородич, Н.А. Катаев, М.С. Клинов, В.А. Крюков, Н.В. Поддерюгина, М.Н. Притула, Ю.Л. Сазанов. “Распараллеливание с помощью DVM-системы некоторых приложений гидродинамики для кластеров с графическими процессорами” // Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск), М.: Изд-во МГУ, 2012, С. 444-450
4. В.А. Бахтин, И.Г. Бородич, Н.А. Катаев, М.С. Клинов, Н.В. Ковалева, В.А. Крюков, Н.В. Поддерюгина. “Диалог с программистом в системе автоматизации распараллеливания САПФОР” // Вестник

