



ИПМ им.М.В.Келдыша РАН

Абрау-2017 • Труды конференции



Н.А. Катаев, А.С. Колганов, А.А. Смирнов

**Поддержка интерактивности в
системе САПФОР**

Рекомендуемая форма библиографической ссылки

Катаев Н.А., Колганов А.С., Смирнов А.А. Поддержка интерактивности в системе САПФОР // Научный сервис в сети Интернет: труды XIX Всероссийской научной конференции (18-23 сентября 2017 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2017. — С. 243-249. — URL: <http://keldysh.ru/abrau/2017/57.pdf> doi:[10.20948/abrau-2017-57](https://doi.org/10.20948/abrau-2017-57)

Размещена также [презентация к докладу](#)

Поддержка интерактивности в системе САПФОР

Н.А. Катаев¹, А.С. Колганов^{1,2}, А.А. Смирнов¹

1 Институт прикладной математики им. М.В. Келдыша РАН

2 Факультет вычислительной математики и кибернетики Московского государственного университета им М.В. Ломоносова

Аннотация. Система САПФОР является в первую очередь системой автоматизированного распараллеливания программ, что предполагает организацию взаимодействия с пользователем для принятия некоторых решений по распараллеливанию. Существующая версия системы обладает средством визуализации автоматически принимаемых решений, а также позволяет пользователю участвовать в распараллеливании через задание специальных указаний в исходном коде программы. При этом возможности, предоставляемые указанными средствами, ограничены и требуют от пользователя либо явного указания всех особенностей программы до начала ее распараллеливания, либо многократного перезапуска всех компонент системы после каждого небольшого уточнения свойств. В данной статье рассматривается новый подход к организации интерактивного взаимодействия с пользователем в системе САПФОР, который позволит предоставлять пользователю информацию о ходе распараллеливания и учитывать его рекомендации по ходу работы системы.

Ключевые слова: интерактивное распараллеливание, автоматизированное распараллеливание, гетерогенный вычислительный кластер.

1. Введение

Выбор оптимизационной последовательности проходов – важная проблема, с которой сталкиваются разработчики оптимизирующих и автоматически распараллеливающих компиляторов [1,2]. Построить универсальную последовательность, пригодную для оптимизации разных участков кода даже в рамках одной программы, может оказаться невозможно. При этом огромные размеры пространства оптимизационных последовательностей не позволяют целенаправленно выбирать проходы для отдельных программ. В совокупности с особенностями, свойственными большим программным комплексам [3], это делает автоматическое распараллеливание таких программ практически невозможным.

Использование интерактивных средств взаимодействия с пользователем в процессе распараллеливания позволяет, с одной стороны, снизить сложность ручного распараллеливания, а с другой, автоматизировать принятие решений в компиляторе, для которых необходимо знание программно-зависимых

особенностей. В зависимости от вклада, вносимого в процесс распараллеливания, можно выделить разные виды интерактивных систем. Одни ориентированы на исследование программы статическими и/или динамическими методами и предполагают, что распараллеливание программы (преобразование исходного кода, расстановка директив параллельного программирования и др.) выполняется пользователем вручную [4]. Другие позволяют разработчику полностью контролировать компиляцию программы: выбирать применяемые преобразования, описывать условия применения оптимизационных проходов, отменять уже сделанные модификации [1]. При этом графический интерфейс позволяет наглядно демонстрировать текущее состояние процесса компиляции, используя различные абстракции для представления программы (дерево циклов, дерево доминаторов и др.). Некоторые системы дают возможность программисту дать ряд указаний, например, выбрать распределение данных, на основе которого будет осуществляться дальнейшее распараллеливание [5].

Система автоматизированного распараллеливания САПФОР [6] нацелена на автоматизацию отображения последовательных программ на гибридные вычислительные кластеры. Распараллеливание выполняется в модели DVMH [7,8]. Можно выделить следующие задачи, для решения которых в системе требуется интерактивность:

- исследование информационной структуры распараллеливаемой программы, представление ее в виде различных абстракций, включающих текстовое и графовое представления;
- задание свойств программы, знание которых необходимо для автоматического принятия решений по распараллеливанию, но которые не могут быть получены автоматически, например, из-за требования консервативности средств статического анализа;
- ручной выбор преобразований программы, которые должны быть выполнены с целью устранения проблем, препятствующих распараллеливанию, если автоматически устранить данные проблемы не удастся;
- частичное распараллеливание программы пользователем, например, за счет выбора распределения данных или ручного распараллеливания отдельных фрагментов программы.

Частичная реализация данных возможностей и применение их при распараллеливании программ [9,10] выявили ряд недостатков, которые учитываются при проектировании обновленной архитектуры системы САПФОР [2].

К ним относится отсутствие интеграции с существующими средами разработки. Это приводит к необходимости разработки отдельного приложения, включающего модули, напрямую не связанные с распараллеливанием (редакторы кода, модули, отвечающие за взаимодействие с файловой системой). С точки зрения пользователя это приводит к отсутствию

полноценной поддержки языков программирования и необходимости использования дополнительных приложений для написания кода программ. Существующая диалоговая оболочка разрабатывалась на языке C++ с применением библиотеки MFC, что предопределило ориентированность САПФОР на вычислительные системы под управлением операционной системы Windows и невозможность установки разрабатываемых инструментов на большинство вычислительных кластеров. Использование директив [9] для описания преобразований программы, хотя и позволило значительно расширить возможности системы и автоматизировать процесс получения последовательной программы, которая может быть автоматически распараллелена, но не является перспективным при преобразовании больших вычислительных комплексов. Даже при преобразовании одного вычислительного цикла объем исходного кода программ, рассмотренных в работе [9], увеличился практически в два раза. В результате последовательного применения различных преобразований становится сложно обзреть вычислительный код и корректно расставить директивы без дополнительной визуальной поддержки со стороны распараллеливающей системы. Еще одной особенностью существующих средств визуализации является ориентированность на устаревшую базу данных САПФОР, обладающую рядом существенных недостатков, препятствующих развитию системы в направлении распараллеливания больших вычислительных комплексов [11].

2 Архитектурные решения

Обновленная архитектура САПФОР предусматривает организацию клиент-серверного взаимодействия между интерактивной оболочкой системы и средствами анализа, преобразования, принятия решений по распределению данных и вычислений и организации межпроцессорных коммуникаций, составляющими ядро системы. Указанное взаимодействие обеспечивается за счет механизма передачи сообщений между клиентской и серверной частью.

Серверная часть может быть развернута как локально, на вычислительной системе пользователя, так и на удаленном сервере. Таким образом, становится возможным задействовать значительные ресурсы, необходимые при распараллеливании программ, и снять нагрузку с компьютера пользователя. Это потенциально позволяет выполнять распараллеливание даже на портативных и маломощных устройствах, что особенно важно при современных темпах и направлениях развития информационных технологий и ориентированности их в направлении облачных решений. Более того, развертывание серверной части на целевых суперкомпьютерах, на которых планируется выполнение распараллеленной программы, позволяет обеспечить тонкую настройку программы и лучше учесть особенности целевой архитектуры.

Еще одним преимуществом выделения интерактивной оболочки в виде клиентского приложения является возможность использовать различные

технологии программирования при разработке клиентской и серверной частей системы. Конечный выбор конкретной технологии обуславливается целями, поставленными перед приложением.

Для серверной части более подходящим является использование низкоуровневых языков программирования таких как С и С++, что необходимо для обеспечения приемлемой производительности при выполнении сложных анализов программ. Данные языки используются при разработке современных компиляторов, таких как GCC и LLVM, но не имеют достаточной поддержки для разработки графических приложений.

В тоже время для клиентского приложения наибольший акцент можно сделать на языки, обеспечивающие удобное отображение информации, например, такие как JavaScript или HTML 5, широко применяемые в браузерах, но также поддерживаемые и при разработке настольных приложений. Унифицированный интерфейс обмена сообщениями между клиентской и серверной частью позволяет разработать целый комплекс интерактивных сред, которые могут запускаться в браузере, на телефоне, или в виде отдельного приложения. Могут быть разработаны плагины по-разному нагруженные функциональностью для таких сред разработки как Microsoft Visual Studio (на базе С#), Microsoft Visual Studio Code (на базе JavaScript), Atom (на базе JavaScript), NetBeans (на базе Java) и др.

3 Апробация предложенной архитектуры

Текущая реализация графического интерфейса является плагином к быстро развивающемуся кроссплатформенному редактору Microsoft Visual Studio Code [12]. Выбор данного инструмента был обусловлен удобным механизмом разработки плагинов, открытыми исходными кодами, кроссплатформенностью, возможностью использовать магазин плагинов для распространения конечного продукта. В качестве языка разработки используется JavaScript или его расширение TypeScript [13], которое компилируется в JavaScript и при этом отличается от него лучшим контролем типов и поддержкой объектно-ориентированного программирования. Использование данных скриптовых языков позволит снизить трудности при портировании плагина для запуска в браузерах и создании облачного сервиса разработки параллельных программ.

В данный момент обеспечивается анализ последовательных программ на языке программирования Си 99, клиентская и серверная части запускаются на локальной машине пользователя, а взаимодействие основано на использовании механизма сокетов. Для разработки инструментов применяются языки программирования JavaScript, TypeScript, С++ 11, визуализация данных осуществляется в виде HTML-страниц. В качестве системы поддержки JavaScript используется Node.js [14]. Схема взаимодействия между клиентской и серверной частями приведена на рис. 1.

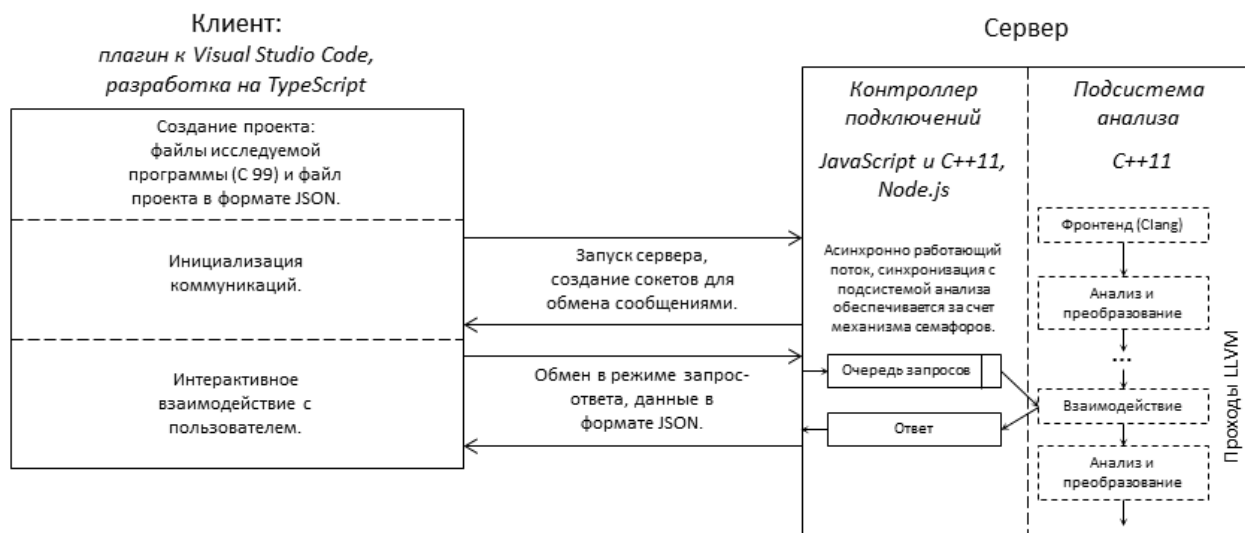


Рис. 1. Схема взаимодействия между клиентской и серверной частями САПФОР

Для пользователя использование системы САПФОР выглядит следующим образом. Сначала создается проект, описывающий распараллеливаемую программу. Проект может включать один или несколько файлов с исходным кодом, а также проектный файл в формате JSON, описывающий правила сборки и анализа программы. После инициализации сеанса связи с серверной частью системы взаимодействие между клиентом и сервером осуществляется с помощью механизма передачи сообщений, закодированных в формате JSON. Как отмечалось в статье [2] процесс распараллеливания программы в разрабатываемой версии системы САПФОР заключается в определении последовательности проходов анализа, преобразования, распределения данных и вычислений, ведущих к получению параллельной версии исходной программы. Для взаимодействия с клиентской частью предусматривается отдельный вид проходов, которые принимают запросы от диалоговой оболочки, а затем подготавливают и отправляют на них ответы. Данные проходы контролируют поведение системы в зависимости от входящих запросов.

4 Заключение

В данной статье рассмотрен новый подход к организации интерактивного взаимодействия с пользователем в системе САПФОР. Данный подход может быть использован для организации удаленного распараллеливания и предоставления функциональности системы как облачного сервиса, доступ к которому будет организован по сети Интернет. В данный момент разработана демонстрационная версия, которая обладает ограниченным функционалом, но при этом может быть использована для оценки возможностей данного подхода. Разработка клиентской части, обеспечивающей отображение информации, и

серверной части может идти независимо, в том числе с использованием разных технологий, что существенно облегчает процесс разработки и повышает его гибкость. Интерактивность на основе системы проходов и использование отдельных проходов, отвечающих за взаимодействие с пользователем и корректирующую поведение системы в зависимости от его запросов, позволяет добавить отсутствующую ранее гибкость в систему САПФОР, поддержать возможность итерационного распараллеливания, уменьшив размер каждой итерации, и более точно реагировать на запросы пользователя. Данные возможности необходимы для организации инкрементального распараллеливания, основанного на механизме областей [2].

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 17-01-00820-а.

Литература

1. Kulkarni P., Zhao W., Moon H., Cho K., Whalley D., Davidson J., Bailey M., Paek Y., Gallivan K. Finding effective optimization phase sequences // Proceedings of the 2003 ACM SIGPLAN Conference on Languages, Tools, and Compilers for Embedded Systems. 2003. P. 12-23.
2. Бахтин В.А., Жукова О.Ф., Катаев Н.А., Колганов А.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Савицкая О.А., Смирнов А.А. Автоматизация распараллеливания программных комплексов // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19-24 сентября 2016 г., г. Новороссийск), М.: ИПМ им. М.В.Келдыша, 2016, С. 76-85
3. В. Armstrong and R. Eigenmann. Challenges in the automatic parallelization of large-scale computational applications. // In Proceedings of SPIE/ITCOM 2001, vol. 4528. August, 2001. P. 50-60.
4. Intel Parallel Studio. URL: <http://software.intel.com/en-us/intel-parallel-studio-home>
5. ParaWise – Widening Accessibility to Efficient and Scalable Parallel Code. // Parallel Software Products White Paper WP-2004-01, 2004.
6. Бахтин В.А., Бородич И.Г., Катаев Н.А., Клинов М.С., Ковалева Н.В., Крюков В.А., Поддерюгина Н.В. Диалог с программистом в системе автоматизации распараллеливания САПФОР // Вестник Нижегородского университета им. Н.И. Лобачевского. – Н. Новгород: Изд-во ННГУ, №5 (2), 2012. С. 242– 245.
7. Коновалов Н.А., Крюков В.А., Михайлов С.Н., Погребцов А.А. Fortran DVM — язык разработки мобильных параллельных программ // Программирование, № 1, 1995, С. 49-54.
8. 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. – Вестник Южно-Уральского

государственного университета, серия "Математическое моделирование и программирование", №18 (277), выпуск 12 – Челябинск: Издательский центр ЮУрГУ, 2012. С. 82-92

9. Baranov M.S., Ivanov D.I., Kataev N.A., Smirnov A.A. Automated parallelization of sequential C-programs on the example of two applications from the field of laser material processing // CEUR Workshop Proceedings 1st Russian Conference on Supercomputing Days 2015. Vol. 1482, 2015, P. 536
10. Катаев Н.А., Колганов А.С., Титов П.А. Автоматизированное распараллеливание задачи моделирования распространения упругих волн в средах со сложной 3D геометрией поверхности на кластеры разной архитектуры // Параллельные вычислительные технологии – XI международная конференция, ПаВТ'2017, г. Казань, 3–7 апреля 2017 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2017, С. 341-355
11. Катаев Н.А. Особенности внутреннего представления системы САПФОР. // Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2013, 1 - 5 апреля 2013 г., г. Челябинск) – Челябинск: Изд-во ЮУрГУ, 2013, с. 380-386.
12. Visual Studio Code. URL: <https://code.visualstudio.com/>
13. Type Script. URL: <https://www.typescriptlang.org/>
14. Node.js. URL: <https://nodejs.org>