



В.А. Бахтин, О.Ф. Жукова, Н.А. Катаев,  
А.С. Колганов, Н.Н. Королев,  
В.А. Крюков, М.Ю. Кузнецов,  
Н.В. Поддерюгина, М.Н. Притула,  
О.А. Савицкая, А.А. Смирнов

**Инкрементальное распараллеливание  
для кластеров в системе САПФОР**

***Рекомендуемая форма библиографической ссылки***

Бахтин В.А., Жукова О.Ф., Катаев Н.А., Колганов А.С., Королев Н.Н., Крюков В.А., Кузнецов М.Ю., Поддерюгина Н.В., Притула М.Н., Савицкая О.А., Смирнов А.А. Инкрементальное распараллеливание для кластеров 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. 48-52. — URL: <http://keldysh.ru/abrau/2017/11.pdf> doi:[10.20948/abrau-2017-11](https://doi.org/10.20948/abrau-2017-11)

Размещена также [презентация к докладу](#)

# Инкрементальное распараллеливание для кластеров в системе САПФОР

В.А. Бахтин<sup>1,2</sup>, О.Ф. Жукова<sup>1</sup>, Н.А. Катаев<sup>1</sup>, А.С. Колганов<sup>1,2</sup>,  
Н.Н. Королев<sup>2</sup>, В.А. Крюков<sup>1,2</sup>, М.Ю. Кузнецов<sup>2</sup>, Н.В. Поддериюгина<sup>1</sup>,  
М.Н. Притула<sup>1</sup>, О.А. Савицкая<sup>1</sup>, А.А. Смирнов<sup>1</sup>

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

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

**Аннотация.** Опыт использования системы САПФОР показал, что при распараллеливании на кластер больших программ и программных комплексов необходимо уметь распараллеливать их инкрементально, начиная с наиболее времяемких ее фрагментов и постепенно добавляя новые фрагменты, пока не достигнем желаемого уровня эффективности параллельной программы. Данная статья посвящена рассмотрению принципов организации инкрементального распараллеливания программных комплексов.

**Ключевые слова:** автоматизация распараллеливания, гетерогенный вычислительный кластер, инкрементальное распараллеливание

## 1. Введение

Особенностью системы САПФОР [1] является использование автоматически распараллеливающего компилятора для систем с распределенной памятью, который переводит последовательную пользовательскую программу в параллельную программу в модели DVMH. Взаимодействие с пользователем осуществляется и на этапе подготовки последовательной программы к распараллеливанию, и в процессе этого распараллеливания.

Инкрементальное распараллеливание широко применяется при разработке программ для мультипроцессора, но при использовании для систем с распределенной памятью его применение наталкивается на значительные трудности. Распределение данных по процессорам влечет за собой накладные расходы на коммуникацию, для эффективной оптимизации которых, как правило, требуется рассмотрение всей программы в целом, а не отдельных ее частей.

Для обеспечения возможности инкрементального распараллеливания на кластеры в системе САПФОР предлагается ввести понятие области распараллеливания. Это позволит последовательно переходить от рассмотрения отдельных небольших областей к более крупным областям, вплоть до целой

программы, сохраняя при этом преемственность ранее принятых решений по распараллеливанию отдельных областей и уточняя их при необходимости.

## **2. Инкрементальное распараллеливание больших программных комплексов**

Как было отмечено выше, описываемая система автоматизированного распараллеливания ориентирована на преобразование уже существующих последовательных программ в эффективные параллельные в модели DVMH. Такие программы могут содержать в себе отдельные модули, решающие определенные задачи, например, модуль ввода/вывода, модуль решения систем линейных уравнений, модуль, отвечающий за подготовительные работы. И в большинстве случаев такие программы не нуждаются в полном распараллеливании, а в некоторых случаях полное распараллеливание такой программы попросту невозможно без существенного переписывания кода.

Но полное распараллеливание часто не требуется. Обычно бывает достаточно распараллелить только модули, производящие сложные вычисления над уже подготовленными данными. С другой стороны, программа может быть настолько большой, что программисту необходимо понять, можно ли вообще получить эффективно работающую программу на конкретной архитектуре. Для выполнения такой оценки достаточно попытаться распараллелить только некоторые модули такой программы, которые могут занимать до 90% времени.

Для решения такой проблемы предлагается метод инкрементального или частичного распараллеливания. Данный метод заключается в том, что программа изначально разбивается на, так называемые, области распараллеливания. Данные области могут быть построены автоматически на основе времен, полученных с помощью профилирования последовательной программы, либо заданы вручную пользователем. Области распараллеливания отражают те участки кода исходной программы, которые будут рассматриваться системой САПФОР. Можно отметить следующие достоинства такого подхода:

- Возможность распараллелить не всю программу, а ее времяемкие фрагменты. Это упрощает работу системы САПФОР и/или программиста, так как существенно сокращается объем кода программы для анализа и распараллеливания. А также позволяет потратить больше времени ЭВМ (и программиста), чтобы найти лучшие схемы распараллеливания времяемких фрагментов.
- Найденные решения для времяемких фрагментов могут быть использованы в качестве подсказки при исследовании оставшихся частей программы на следующих итерациях распараллеливания в системе САПФОР.

- Появляется возможность ручного распараллеливания некоторых фрагментов программы и учета принятых программистом решений при распараллеливании других фрагментов системой САПФОР.

### **3. Схема обеспечения инкрементального распараллеливания**

Система автоматизированного распараллеливания САПФОР содержит совокупность блоков, которые выполняют статический и динамический анализ, необходимый для трансформации последовательной программы в параллельную программу в модели DVMH. Статический анализ реализуется в виде набора проходов по файлам входной последовательной программы, которые выполняют разные функции. Например, есть проход, который строит граф вызовов процедур, или проход, который строит связи между используемыми массивами в программе. Все проходы делятся на три типа. К первому типу относятся проходы, которые выполняют проверку заложенных в систему САПФОР ограничений, ко второму – проходы, которые выполняют анализ программы, выявляют ее свойства, и к третьему – проходы, которые выполняют генерацию полученного после распараллеливания системой САПФОР кода.

Динамический анализ реализуется в виде отдельной программной компоненты, которая пытается получить некоторые свойства программы для системы САПФОР. Наличие зависимостей в циклах (например, редуccionных, зависимостей по массивам и прочие) является самым главным ограничением для их распараллеливания. Очевидно, что ни один статический анализ не сможет полностью обеспечить нужды САПФОРА для автоматизации преобразования и распараллеливания произвольной последовательной программы. Динамический же анализ выполняется достаточно долго и позволяет получить некоторые свойства программы для ограниченного набора входных данных.

Применение динамического и статического анализов позволяет более точно обнаруживать некоторые классы зависимостей, которые поддерживаются в модели DVMH. К таким зависимостям можно отнести зависимости по приватизируемым переменным, редуccionным переменным, а также регулярные зависимости по элементам массивов.

Все описанные подходы хорошо работают для небольших программ, которые уже были преобразованы в потенциально параллельные программы пользователем или некоторым автоматом. Но если рассматривать большие комплексы, можно натолкнуться на серьезные трудности.

Как было отмечено выше, во многих случаях нет необходимости в распараллеливании всего комплекса целиком. Чтобы обрабатывать часть комплекса, возникают некоторые вопросы, например, как сообщить системе САПФОР о том, что необходимо анализировать, а что – нет; как учитывать разные решения между разными частями комплекса.

Для решения таких проблем, предлагается ввести понятие области распараллеливания. Каждая область распараллеливания имеет свой уникальный идентификатор и позволяет выделить ту часть кода, которую необходимо анализировать на предмет распараллеливания. Каждая область может состоять из нескольких областей, которые должны быть помечены одним и тем же идентификатором. Вложенные области не допускаются. Таким образом, всю программу можно представить в виде совокупности областей, каждую из которых система САПФОР будет рассматривать как отдельную программу. Если в программе не заданы области распараллеливания, то система САПФОР пытается распараллелить всю программу целиком.

Использование одних и тех же массивов в разных областях распараллеливания не является проблемой для системы САПФОР, так как программа переводится в параллельную программу в модели DVMH, в которой специальные директивы перераспределения данных позволяют распределить одни и те же массивы программы для каждой из областей распараллеливания необходимым образом.

### **Заключение**

Ручное распараллеливание больших программ и программных комплексов является очень трудоемким процессом. Подход к автоматизации распараллеливания, предложенный авторами статьи, заключается в инкрементальном выполнении процесса распараллеливания для кластеров и расширении этого итерационного процесса возможностью интерактивного взаимодействия с пользователем. В докладе будут рассмотрены особенности реализации предложенного подхода и проиллюстрированы первые результаты его применения.

Работа выполнена при поддержке РФФИ (проекты № 16-07-01067, 16-07-01014, 17-01-00820).

### **Литература**

1. 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.
2. Юрушкин М.В., Петренко В.В., Штейнберг Б.Я., Алымова Е.В., Абрамов А.А., Баглий А.П., Гуда С.А., Дубров Д.В., Кравченко Е.Н., Морылев Р.И., Нис З.Я., Полуян С.В., Скиба И.С., Шаповалов В.Н., Штейнберг О.Б., Штейнберг Р.Б. Диалоговый высокоуровневый оптимизирующий распараллеливатель (ДВОР) // Труды Международной суперкомпьютерной конференции “Научный сервис в сети Интернет: суперкомпьютерные

центры и задачи”, Новороссийск, сентябрь 2010 г., – М.: Изд-во МГУ, 2010. С. 71-75.

3. ParaWise – Widening Accessibility to Efficient and Scalable Parallel Code. // Parallel Software Products White Paper WP-2004-01, 2004.
4. Intel Parallel Studio. URL: <http://software.intel.com/en-us/intel-parallel-studio-home> (дата обращения 18.04.2017)