УДК 004.42 + 004.415

ВЕБ-СРЕДА АНАЛИЗА И ПРЕОБРАЗОВАНИЙ ПРОГРАММ В ОПТИМИЗИРУЮЩЕЙ РАСПАРАЛЛЕЛИВАЮЩЕЙ СИСТЕМЕ

А. П. Баглий

Институт математики, механики и компьютерных наук Южного федерального университета, *г. Ростов-на-Дону*

taccessviolation@gmail.com

Аннотация

Описан опыт проектирования различных вариантов веб-среды разработки (IDE) для Оптимизирующей распараллеливающей системы и компилятора на реконфигурируемую архитектуру на основе существующих инструментов, таких, как Jupyter Notebook и Eclise Che. Сформированы требования к инструментам в составе Открытой распараллеливающей системы для поддержки их интеграции в веб-среду разработки, доступную в интернете. Описан процесс создания переносимого окружения для разработки модулей компилятора, демонстрации его работы и обучения навыкам разработки параллельных программ. Приведены примеры использования разработанных преобразований программ, используемых при оптимизации программ для ПЛИС в разработанной веб-среде, и описаны способы визуализации результатов выполнения преобразований и анализа при использовании Jupyter Notebook. Проведенная работа демонстрирует возможность организации удаленного доступа к библиотеке разрабатываемых инструментов оптимизации программ в виде, удобном прикладным разработчикам.

Ключевые слова*: интегрированная среда, распараллеливающий компилятор, преобразования программ, ПЛИС, контейнеризация, интерактивная тетрадь, облачные вычисления.*

ВВЕДЕНИЕ

При разработке сложной системы, например, оптимизирующего компилятора, возникает множество проблем, связанных с организацией как самого процесса разработки, обучением и вовлечением новых программистов, так и

предоставления доступа к результатам извне, например, для демонстрации определенных функций и результатов численных экспериментов. Проблемы, возникающие в процессе разработки компилятора, в чем-то аналогичных тем, которые возникают перед программистом, не знакомым с разработкой программ для программируемых логических схем. На сегодняшний день системы, позволяющие напрямую преобразовывать программы на высокоуровневом языке программирования в схему для ПЛИС, или недостаточно развиты, или недостаточно распространены. Для того чтобы сгладить эту проблему, целесообразно организовать доступ к подобным инструментам через интернет в виде интегрированной среды разработки в браузере, дополненной некоторыми расширениями, позволяющими программисту в диалоговом режиме итеративно модифицировать свою программу и получаемую схему на ПЛИС и оценивать их характеристики. Такой подход может решить некоторые из самых острых проблем, с которыми сталкивается все больше программистов при первых попытках разработки программ для ПЛИС.

Выделим основные проблемы, которые рассматриваются в данной работе:

— быстрое развертывание окружения со всеми необходимыми инструментами для нового члена коллектива разработчиков или нового клиента системы;

— предоставление интуитивно понятного и простого диалогового интерфейса, позволяющего программисту получить программу, оптимизированную для ПЛИС или другой архитектуры по исходной;

— демонстрация некоторых новых возможностей разрабатываемого компилятора, включая оптимизирующие преобразования и алгоритмы анализа программы;

— использование создаваемой системы для обучения разработке параллельных программ и использования преобразований программ;

— использование частей создаваемой системы в научных исследованиях, где требуются предоставление доступа к результатам и удобство их воспроизведения.

Для демонстрации возможностей Оптимизирующей распараллеливающей системы [1] ранее были разработаны веб-интерфейсы для демонстрации ее от-

дельных функций [2, 3]. Общий вид первой версии веб-интерфейса показан на рис. 1.

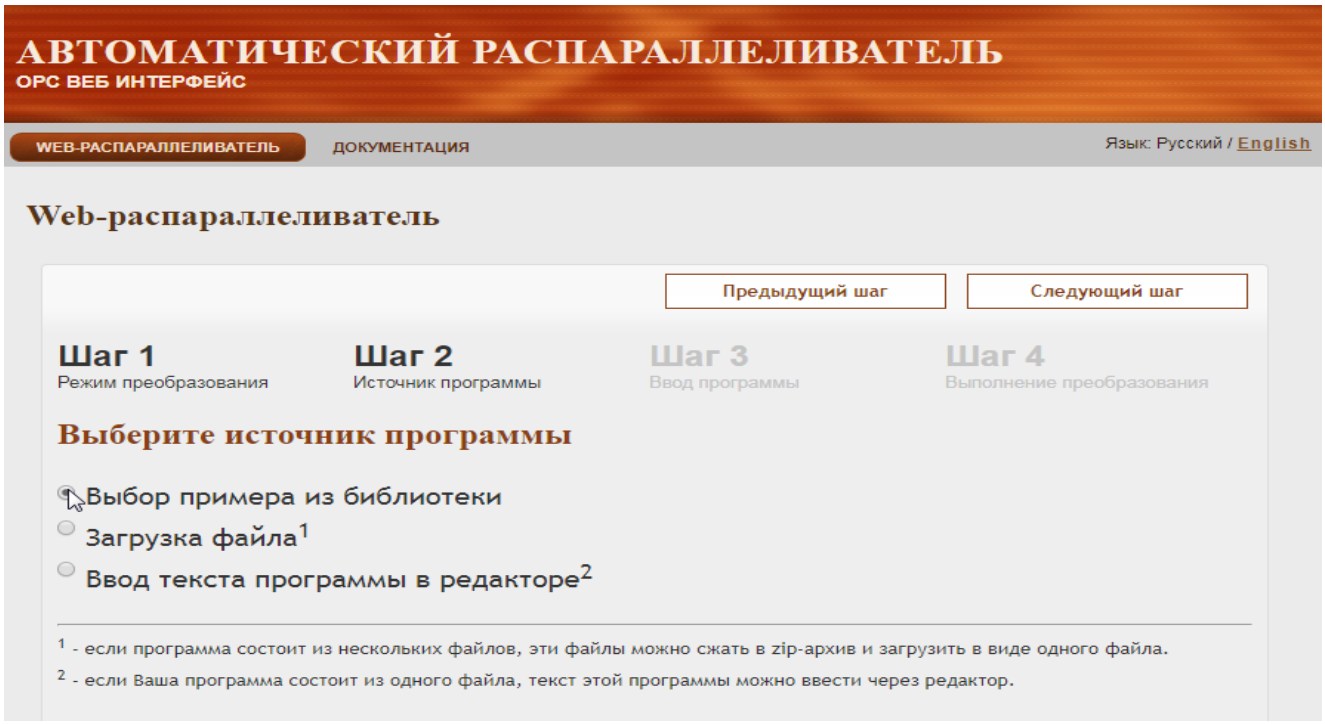

Рис. 1. Окно одной из предыдущих версий веб-интерфейса ОРС

Этот интерфейс позволяет пользователю загрузить исходный текст программы, выбрать действие из заданного набора (преобразование программы, автоматическое распараллеливание), выполнить указанной действие на сервере и скачать результирующий текст программы.

Такой веб-интерфейс решает только проблему №2 из перечисленных выше, впрочем, далеко не полностью. Пользователь не имеет возможности напрямую управлять выполнением преобразований, а от разработчиков требуется создание специальных сценариев использования системы для всех необходимых случаев.

Настоящая работа ставит задачу создания более универсальной среды разработки, доступной в интернете, которая:

— была бы основана на существующих наработках, широко применяемых в среде облачных вычислений;

— решала бы остальные указанные проблемы;

— не требовала бы в дальнейшей перспективе дополнительных усилий от разработчиков для поддержания своей функциональности при модификации самого разрабатываемого компилятора.

На основе ОРС разрабатывается компилятор на реконфигурируемые вычислительные архитектуры [4], для которого указанные проблемы стоят особенно остро, поэтому подразумевается в первую очередь применение описанных подходов к данному компилятору.

На основе перечисленных проблем ставится задача о предоставлении доступа пользователей через веб-браузер к библиотеке интерактивных документов (демонстрационных примеров, программ), которые будут сопровождаться индивидуальным для каждого пользователя окружением разработки на удаленном сервере. Эти окружения должны создаваться по требованию и позволять запускать программы из библиотеки, визуализировать результаты их работы и легко модифицировать программы.

Основными результатами данной работы являются, во-первых, новые архитектурные требования к Оптимизирующей распараллеливающей системе, которые постепенно внедряются в нее, чтобы улучшить ее модульность и дать возможность собирать на ее основе инструменты для компиляции программ, обучения параллельному программированию и программированию для ПЛИС. Во-вторых, результатом работы является прототип веб-среды разработки для синтеза программ для ПЛИС, основанный на ОРС и существующих облачных IDE с открытым исходным кодом.

Далее в разделе «Постановка задачи» описаны более подробно требования к составным частям компилятора, которые выработаны для создания веб-IDE на его основе. В разделе «Jupyterlab и OPS» описан подход к созданию среды разработки на основе OPS на языке C++ на основе интерактивных тетрадей в браузере. В разделе «Визуализация результатов» описаны выбранные способы визуализации результатов запуска преобразований программ. В разделе «Примеры преобразований программ» приведены примеры использования веб-среды для запуска преобразований, предназначенных для оптимизации программ на ПЛИС. В разделе «Результаты» представлены основные изменения, которые позволяют создать минимально функциональное окружение разработки программ на основе разрабатываемого компилятора.

ПОСТАНОВКА ЗАДАЧИ

В среде облачных вычислений существует множество широко используемых веб-ориентированных сред разработки программ с открытым исходным кодом, которые подразумевают создание расширений для той или иной предметной области, например, поддержки новых языков программирования, управления вычислительными кластерами.

Стоит особо выделить среду Eclipse Che [5], архитектура которой позволяет решить почти все проблемы, упомянутые во введении:

— создание и использование контейнеризованных рабочих сред для разработчиков системы с помощью docker[1] решает проблему быстрого развертывания окружения с повторяемыми характеристиками;

— управление рабочей средой с помощью расширяемого API решает проблему удобства демонстрации выбранных функций системы;

— возможность создания расширений для всех архитектурных частей системы позволяет добавить в нее новые представления для демонстрации важных функций и визуализации результатов.

Однако использование данной среды разработки в качестве основы для собственной облачной среды требует модификации как самой Eclipse Che, так и разрабатываемого компилятора.

В общем виде требуются:

— разбиение процесса работы компилятора на этапы, между которыми возможен экспорт или визуализация промежуточных результатов;

— добавление новых представлений в интерфейс IDE и соответствующих им методов API рабочей среды;

— создание набора контейнеров для обеспечения работы компилятора;

— обеспечение более слабых связей между модулями компилятора для удобства их использования со стороны облачной IDE.

[1] https://www.docker.com/

JUPYTERLAB И OPS

Jupyterlab [9] является на данный момент самым простым в использовании средством интерактивного предоставления результатов вычислений, это веб-приложение и связанный с ним набор инструментов поддерживают множество языков программирования, включая C++, за счет использования Xeus-Cling [10]. Это позволяет организовать доступ ко всем инструментам в составе OPC в виде интерактивных тетрадей в браузере, как показано, например, на рис. 2.

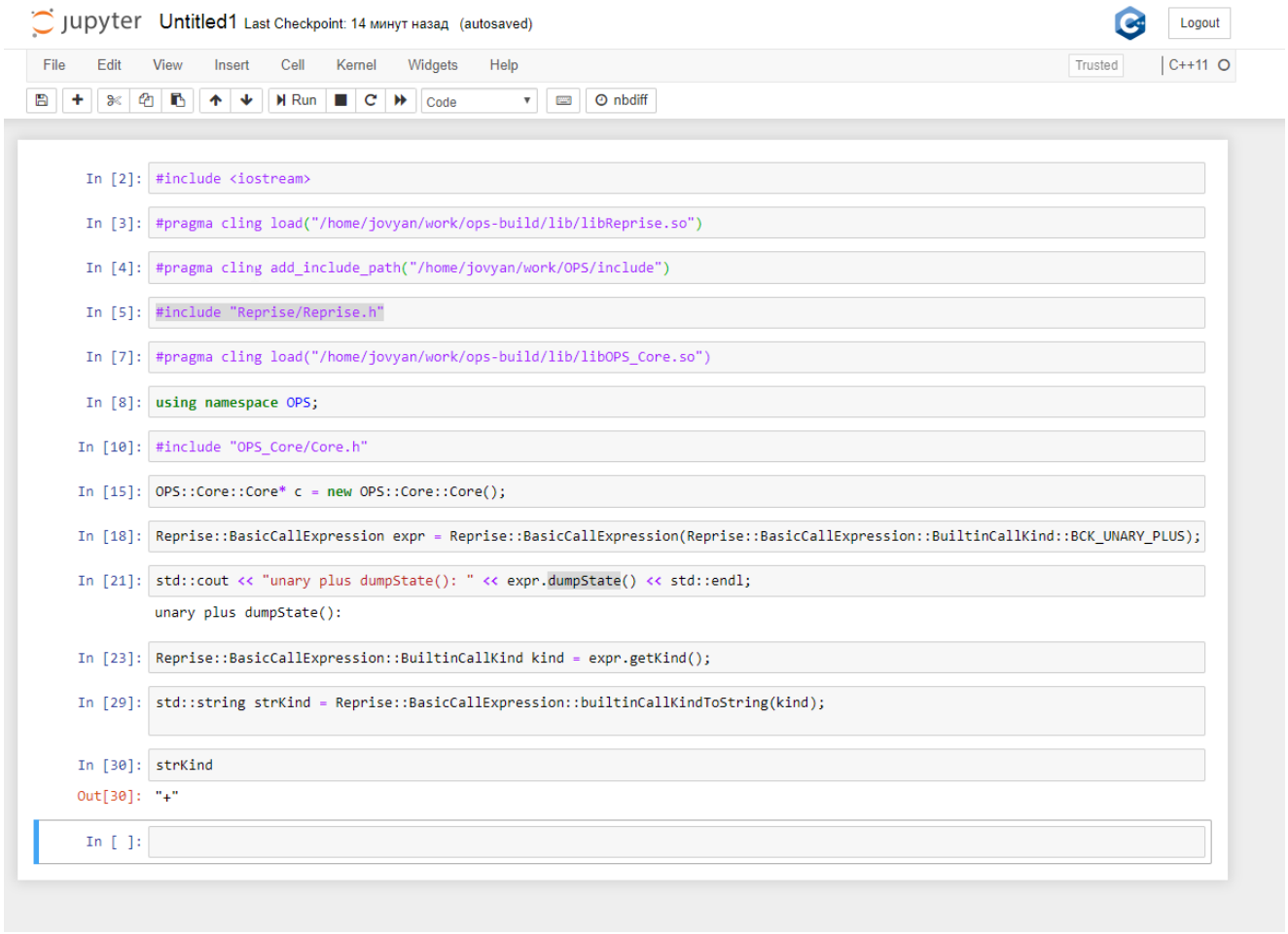

Рис. 2. Пример выполнения функций OPC в Jupyter notebook

За счет того, что в интерактивных тетрадях Jupyterlab получается использовать тот же язык программирования, на котором написана сама система, достигается простота написания сценариев ее использования разными типами пользователей. Этой возможностью уже пользуются различные проекты, например,

система анализа данных ROOT[2], так же использующая C++ и Cling в среде JupyterLab.

При использовании JupyterLab каждый пользователь получает в свое распоряжение интерактивную среду в браузере, внутри которой он может создавать тетради, использующие различные языки программирования, выполнять программы в них и использовать предоставленную библиотеку примеров. Общий вид интерфейса пользователя показан на рис. 3.

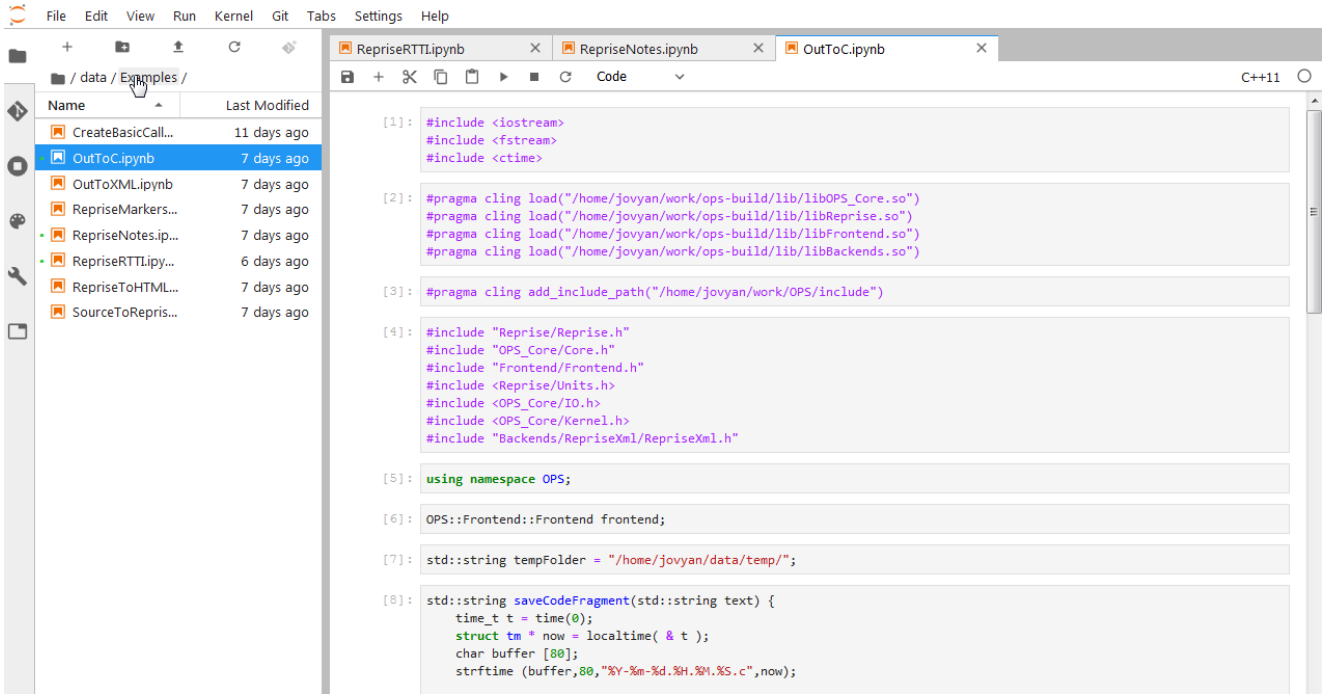

Рис. 3. Пример интерфейса пользователя системы при работе с набором готовых примеров использования функций OPC

Теперь для предоставления различных сценариев использования системы достаточно составить набор интерактивных тетрадей, которые:

— демонстрировали бы ключевые функции модулей системы, например, работу с внутренним представлением, выполнение преобразований программ, что особенно полезно в процессе обучения;

— предоставляли бы пользователям возможность проведения экспериментов по исследованию производительности и эффектов различных преобразований над тестируемой программой с помощью вспомогательных сервисов;

[2] https://root.cern.ch/

— предоставляли бы разработчикам системы простой и удобный интерфейс для обмена результатами работы, проведения тестов.

Кроме этого, использование тетрадей JupyterLab позволяет организовать работу с библиотекой примеров программ, например, используя проект Binder[3]

ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ

JupyterLab позволяет использовать различные форматы[4] содержимого ячеек, что дает возможность визуализировать данные любых типов. При выполнении преобразований программ требуется визуализировать следующие данные:

— тексты программ с пометками операторов, вхождений переменных и т. п., для которых удобно использовать формат HTML;

— графовые представления программ, в которых узлами служат вхождения переменных, выражения или операторы, для которых удобно использовать SVG-графику;

— табличные данные – результаты экспериментов, метрики исходного кода, для которых можно использовать формат Markdown;

— графовые представления программ, сложные для визуализации, например, решетчатый граф программы, для которых возможно использовать растровую графику.

Для визуализации текстов программ, например, исходной и полученной после преобразования, удобно использовать HTML содержимое, как проиллюстрировано на рис. 4., на котором показаны два результата поиска похожих участков в исходной программе для создания расширения набора инструкций процессора на ПЛИС. Кроме пометок операторов цветом возможно использовать интерактивные элементы управления. Использование текстовых форматов преставления, например, HTML или SVG, облегчает разработку модулей визуализации и передачу готовых данных.

[3] https://mybinder.org/

[4] https://jupyterlab.readthedocs.io/en/stable/user/file_formats.html

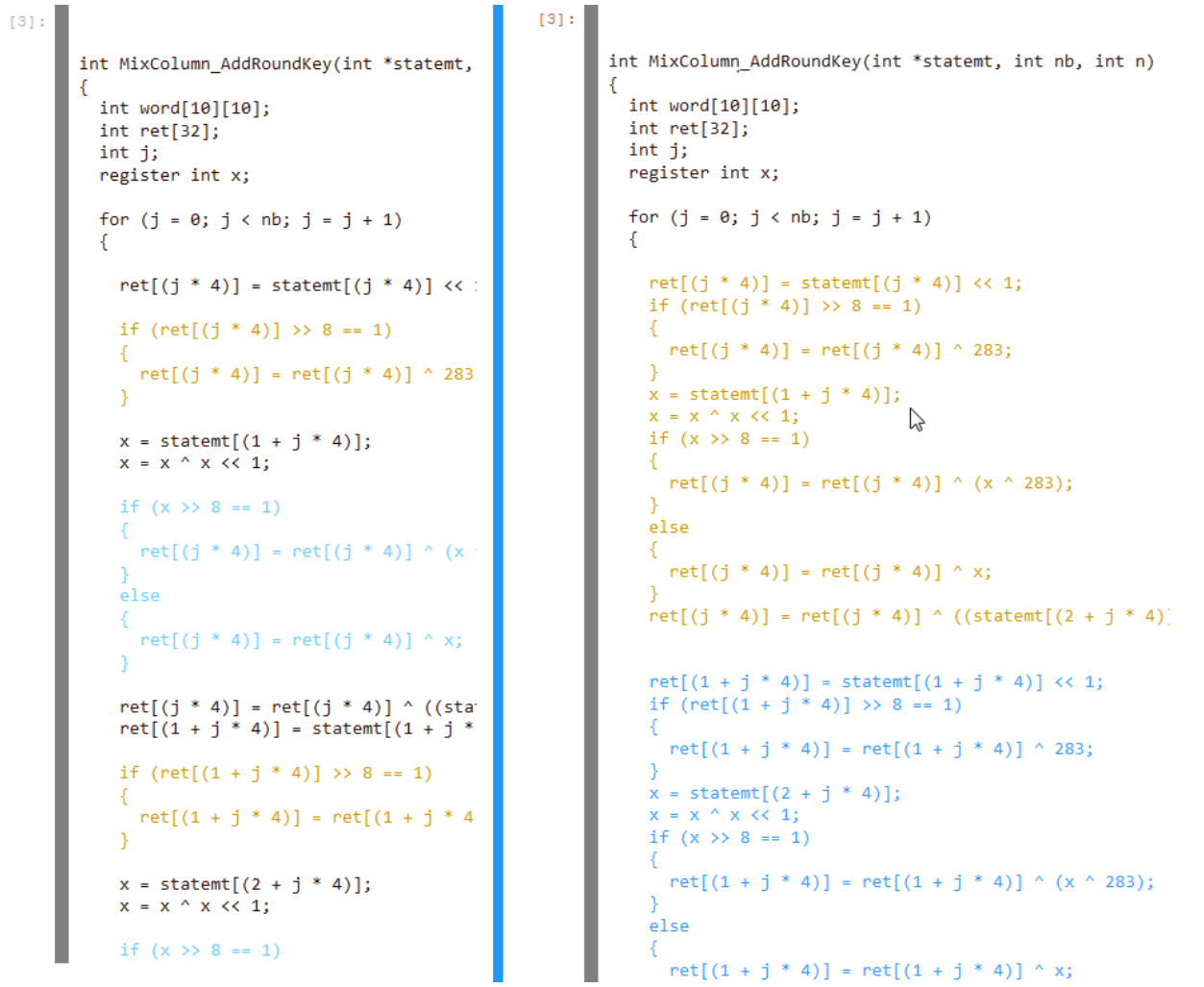

Рис. 4. Пример использования HTML для визуализации размеченного текста программ внутри JupyterLab

ПРИМЕРЫ ПРЕОБРАЗОВАНИЙ ПРОГРАММ

При разработке преобразований программ в Оптимизирующей распараллеливающей системе и оценке их применимости для компиляции программ на ПЛИС требуется удобное окружения для тестирования преобразований в отдельности и в цепочке, которое позволит легко демонстрировать полученные результаты, включать примеры в документацию. Использование JupyterLab с разработанной системой решает эти задачи.

На рис. 5 и 6 показаны примеры запуска преобразований на простых программах. На рис. 5 показано преобразование, конвертирующее функцию в макро-описание для эмулятора операций в системе ТСЕ[5]. На рис. 6 показан пример

[5] http://openasip.org/

запуска преобразования, заменяющего участок кода на вызов функции с соответствующим телом.

```
[26]: "
int main()
{
  int a;
  int b;
  int z;
  a = 0;
  b = 2;
  z = 0;
  z = add(a, b);
  return 0;
}"
```

```
[15]: #include <iostream>
std::ostringstream str;
using namespace OPS;
using namespace OPS::Frontend;
using namespace OPS::Reprise;
OPS::Backends::OutToC printer(str)
OPS::Reprise::Declarations& decls = frontend.getProgramUnit().getUnit(0).getGlobals();
OPS::Reprise::DeclIterator<OPS::Reprise::SubroutineDeclaration> it = decls.getFirstSubr();
OPS::Reprise::SubroutineDeclaration& addSub = *it;
OPS::Backends::OutToTCEBehaviour tceBehaviourGenerator(str);
tceBehaviourGenerator.printOperation(addSub, "OPADDINT");
```

```
[23]: str.str()
```

```
[23]: "#include "OSAL.hh"
OPERATION(OPADDINT)
TRIGGER
int a = INT(1);
int b = INT(2);

  int z;
  z = a + b;
IO(3) = static_cast<int>(z)

return true;
END_TRIGGER;
END_OPERATION(OPADDINT)
"
```

Рис. 5. Пример запуска генерации кода для эмулятора TCE

Подобные примеры позволяют расширить документацию разрабатываемой системы, сделать исходный код более самодокументируемым, дополнить модульные и интеграционные тесты системы более удобными сценариями использования её функций. Это также облегчает процесс экспериментирования с

модулями системы, построения цепочек преобразований и их тестирования.

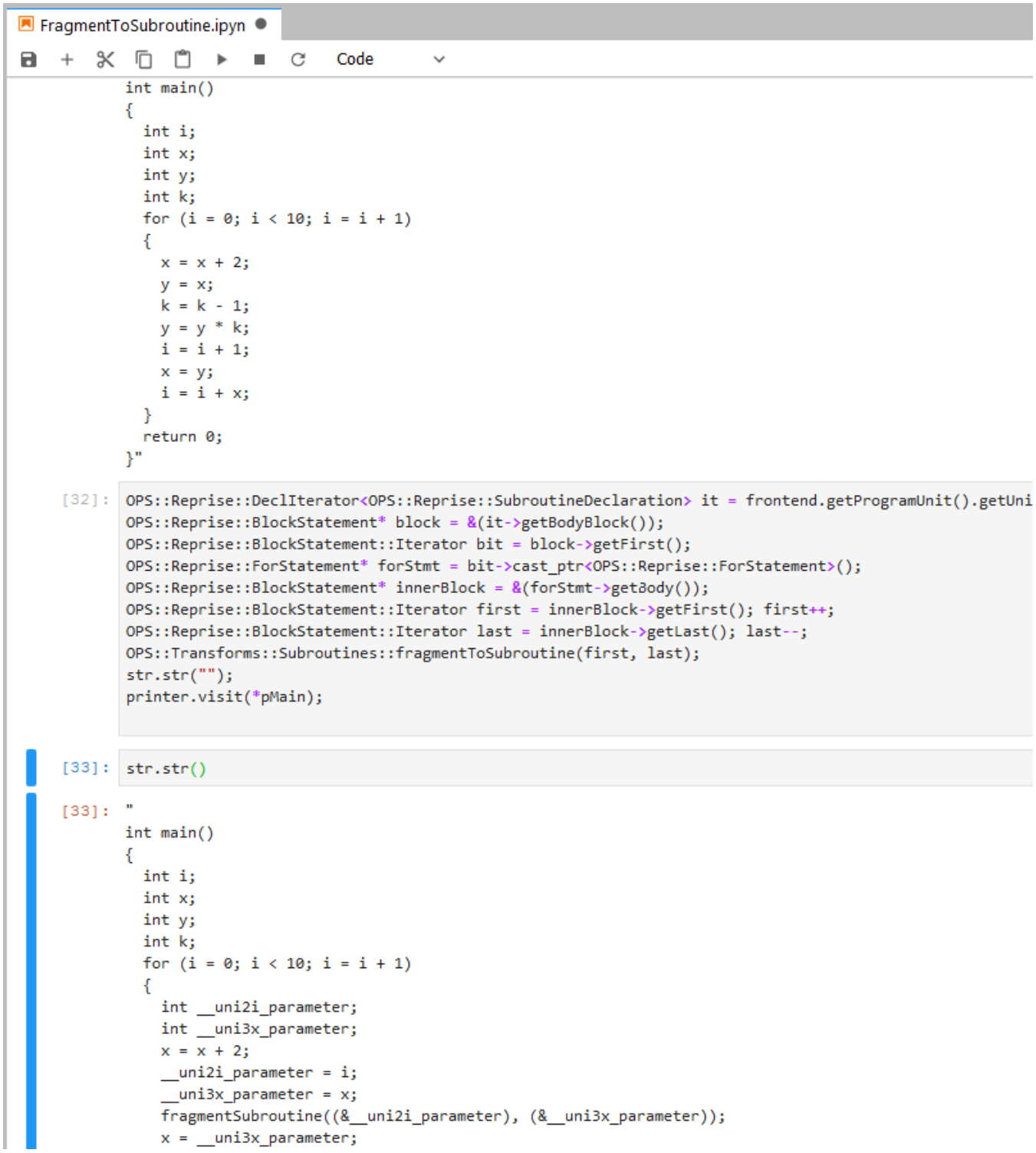

Рис. 6. Пример запуска преобразования «фрагмент в подпрограмму»

РЕЗУЛЬТАТЫ

Для решения поставленных задач в ОРС были реализованы следующие нововведения:

— набор контейнеров для надежной и повторяемой сборки самой ОРС, ее внешних зависимостей, включая компиляторы Clang/LLVM различных версий, компилятора на ее основе, запуска этого компилятора на тестовых программах;

— дополнительные сервисы для симуляции выполнения сгенерированных программ на ПЛИС, тестирования производительности преобразованных программ на языке С по принципу «черного ящика», аналогичные ранее разработанной системе тестирования из [6];

— процесс работы компилятора разбит на этапы, которыми можно управлять:

- выбор участков программы для выполнения на ПЛИС может быть автоматическим, или пользователь может выделить нужные фрагменты псевдокомментариями;
- компилятор принимает на вход набор параметров, описывающих необходимые действия;
- возможные действия включают экспорт промежуточных результатов для анализа и визуализации, генерацию выходной программы и т. п.;

— для нескольких выбранных функций компилятора и ОРС созданы соответствующие представления для визуализации данных в среде разработки, например, для статического профилировщика [7] и функции построения обобщенных конвейеров [8].

На рис. 7 показан набор основных модулей ОРС, которые реализуют плагины системы, независимые друг от друга. Это позволяет предоставлять программисту доступ только к выбранным функциям, упростить и ускорить сборку системы и ее развертывание.

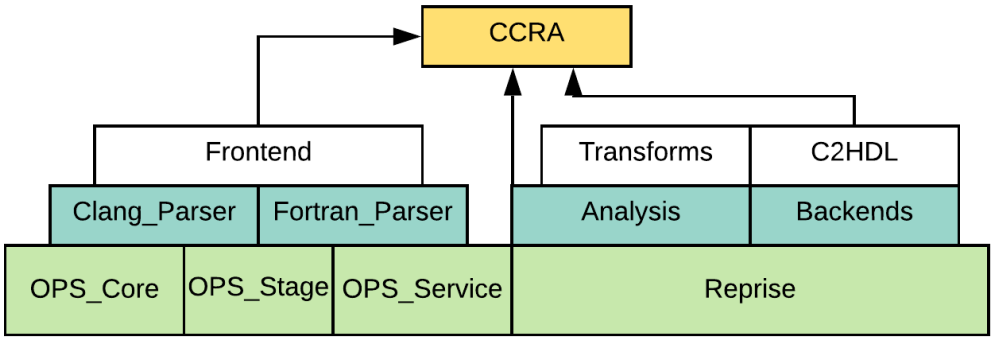

Рис. 7. Набор модульных компонент OPS, включая компилятор для ПЛИС (CCRA)

Дальнейшая работа ведется в следующих направлениях:

— поддержка работы с кодом компилятора в облачной IDE, то есть обеспечение удобства разработки самого компилятора;

— расширение функциональности специализированной облачной IDE, позволяющей использовать разрабатываемый компилятор;

— создание библиотеки примеров разного уровня сложности, документирующих использование разрабатываемых преобразований программ и позволяющих составлять из них более сложные модули.

Для начала знакомства с исходным кодом OPC и компилятора теперь достаточно открыть URL в браузере, после чего будет создано новое рабочее место с полностью сконфигурированным окружением для разработки. Это особенно облегчает непосредственное знакомство с исходным кодом системы и уменьшает время на первоначальную настройку при обучении.

ЗАКЛЮЧЕНИЕ

На основе современных облачных интегрированных сред разработки программ возможно создание специализированных инструментов, позволяющих облегчить разработку, использование и внедрение в процесс обучения сложных программных систем, включая оптимизирующие компиляторы для реконфигурируемых архитектур (ПЛИС). При этом интерактивные среды выполнения программ в браузере, ранее поддерживавшие исключительно интерпретируемые

языки, теперь возможно использовать для работы с системой значительного размера, написанной на языке C/C++.

Для того чтобы разработать функциональную веб-среду разработки на основе переносимого компилятора, требуется решить ряд задач по улучшению модульности кода, управлению внешними зависимостями системы, улучшению ее переносимости на другие платформы и предоставлению более универсальных программных интерфейсов для использования функций системы. Выполнение этих требований позволяет упростить как процесс разработки всей системы, так и использование системы, в том числе, для обучения программированию, демонстрации ключевых возможностей и визуализации результатов для пользователя web-IDE в диалоговом режиме.

На основе разработанной многопользовательской среды разработки и интерактивного выполнения программ возможно построение библиотеки примеров программ, применение которой облегчает все этапы разработки и использования сложной системы.

Благодарности

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект № 18-37-00179.

СПИСОК ЛИТЕРАТУРЫ

1. Оптимизирующая распараллеливающая система URL: www.ops.rsu.ru (дата обращения: 25.07.19).

2. *Штейнберг Б.Я., Аллазов А.Н., Алымова Е.В., Баглий А.П., Гуда С.А., Дубров Д.В., Кравченко Е.Н., Морылев Р.И., Рошаль А.С., Юрушкин М.В., Штейнберг Р.Б.* Web-ориентированный автоматический распараллеливатель программ // Параллельные вычислительные технологии (ПАВТ'2014). Труды международной научной конференции. Ростов-на-Дону: 1–3 апреля 2014.

3. *Алымова Е.В., Кравченко Е.Н., Морылев Р.И., Юрушкин М.В., Штейнберг Б.Я.* Распараллеливание и оптимизация программ с помощью Web-ускорителя OPC // Научный сервис в сети Интернет: поиск новых решений. Труды XIV Международной суперкомпьютерной конференции (17–22 сентября 2012 г., г. Новороссийск). М.: Изд-во МГУ, 2012.

4. *Steinberg B.Y., Bugliy A.P., Dubrov D.V., Mikhailuts Y V., Steinberg O.B., Steinberg R.B.* A Project of Compiler for a Processor with Programmable Accelerator // Procedia Computer Science. 2016. No 101. P. 435–438.

5. Localhost is Killing Software Delivery // Codenvy blog URL: https://blog.codenvy.com/localhost-is-killing-software-delivery-8c93cd49328 (дата обращения: 20.11.19).

6. *Штейнберг Б.Я., Алымова Е.В., Баглий А.П., Морылев Р.И., Нис З.Я., Петренко В.В., Штейнберг Р.Б.* Автоматизация тестирования элементов высокопроизводительного программного комплекса // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность. Труды Всероссийской суперкомпьютерной конференции (21–26 сентября 2009 г., г. Новороссийск). М.: МГУ им. М.В. Ломоносова, 2009. С. 287–292.

7. *Полуян С.В.* Профилирование и его применение в диалоговом оптимизирующем распараллеливателе // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: Труды Международной суперкомпьютерной конференции (20-25 сентября 2010г., г. Новороссийск). М.: Изд-во МГУ, С. 652–653.

8. *Баглий А.П., Дубров Д.В., Штейнберг Б.Я., Штейнберг Р.Б.* Повторное использование ресурсов при конвейерных вычислениях // 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. 43–46.

9. *Kluyver T et al*. Jupyter Notebooks – a publishing format for reproducible computational workflows // Positioning and Power in Academic Publishing: Players, Agents and Agendas, IOS Press Ebooks. P. 87–90.

10. Xeus-cling на Github. URL: https://github.com/QuantStack/xeus-cling (дата обращения: 20.11.19)

WEB BASED SYSTEM FOR PROGRAM ANALYSIS AND TRANSFORMATION IN OPTIMIZING PARALLELIZING SYSTEM

A. P. Bagly

Institute of mathematics, mechanics and computer science, Southern federal university, Rostov-on-Don

taccessviolation@gmail.com

Abstract

Experience of designing different variants for web-based development environment (IDE) for Optimizing parallelizing system and compiler for reconfigurable architecture is described. Designed system is based on existing tools and frameworks such as Jupyter Notebook and Eclipse Che. Set of requirements for Optimizing parallelizing system components is developed to make it possible to integrate them into web-based development environment accessible through the Internet. Designing portable environment for compiler development, compiler technology demonstration and teaching parallel program development is also described. Examples of performing newly developed program transformations are shown to be used during program optimizations for FPGA inside the designed web environment. Means of program transformation visualization are described for use with Jupyter Notebook. The work shown demonstrates possibility to organize remote access to library of instruments and tools for program optimizations currently under development that would be convenient for application developers.

Keywords*: integrated environment, parallelizing compiler, program transformations, FPGA, containerization, interactive notebook, cloud computing*

REFERENCES

1. Optimizing parallelizing system. URL: http://www.ops.rsu.ru/.

2. *Shteinberg B.Ya., Allazov A.N., Alymova E.V., Bagly A.P., Guda S.A., Dubrov D.V., Kravchenko E.N., Morylev R.I., Roshal A.S., Iurushkin M.V., Shteinberg R.B.* Web-orientirovannyi avtomaticheskii rasparallelivatel programm // Parallelnye vychislitelnye tekhnologii (PAVT'2014), Trudy mezhdunarodnoi nauchnoi konferentsii. Rostov-na-Donu: 1–3 aprelia 2014.

3. *Alymova E.V., Kravchenko E.N., Morylev R.I., Iurushkin M.V., Shteinberg B.Ya.* Rasparallelivanie i optimizatsiia programm s pomoshchiu Web-uskoritelia ORS // Nauchnyi servis v seti Internet: poisk novykh reshenii, Trudy XIV Mezhdunarodnoi superkompiuternoi konferentsii (17–22 sentiabria 2012 g/, g. Novorossiisk). M.: Izd-vo MGU, 2012.

4. *Steinberg B.Ya., Bugliy A.P., Dubrov D.V., Mikhailuts Y.V., Steinberg O.B., Steinberg R.B.* A Project of Compiler for a Processor with Programmable Accelerator // Procedia Computer Science. 2016. No 101. P. 435–438.

5. *Localhost is Killing Software Delivery* // Codenvy blog URL: https://blog.codenvy.com/localhost-is-killing-software-delivery-8c93cd49328

6. *Shteinberg B.Ya., Alymova E.V., Baglii A.P., Morylev R.I., Nis Z.Ia., Petrenko V.V., Shteinberg R.B.* Avtomatizatsiia testirovaniia elementov vysokoproizvoditelnogo programmnogo kompleksa // Nauchnyi servis v seti Internet: masshtabiruemost, parallelnost, effektivnost. Trudy Vserossiiskoi superkompiuternoi konferentsii (21–26 sentiabria 2009 g., g. Novorossiisk). M.: MGU im. M.V. Lomonosova, 2009. S. 287–292.

7. *Poluian S.V.* Profilirovanie i ego primenenie v dialogovom optimiziruiushchem rasparallelivatele // Nauchnyi servis v seti Internet: superkompiuternye tsentry i zadachi: Trudy Mezhdunarodnoi superkompiuternoi konferentsii (20–25 sentiabria 2010 g., g. Novorossiisk). M.: Izd-vo MGU, 2010. S. 652–653.

8. *Baglii A.P., Dubrov D.V., Shteinberg B.Ya., Shteinberg R.B.* Povtornoe ispolzovanie resursov pri konveiernykh vychisleniiakh // Nauchnyi servis v seti Internet: trudy XIX Vserossiiskoi nauchnoi konferentsii (18–23 sentiabria 2017 g., g. Novorossiisk). M.: IPM im. M.V. Keldysha, 2017. S. 43–46.

9. *Kluyver T. et al.* Jupyter Notebooks – a publishing format for reproducible computational workflows // Positioning and Power in Academic Publishing: Players, Agents and Agendas. IOS Press Ebooks, P. 87–90.

10. Xeus-cling na Github. URL: https://github.com/QuantStack/xeus-cling

СВЕДЕНИЯ ОБ АВТОРЕ

БАГЛИЙ Антон Павлович – ассистент Института математики, механики и компьютерных наук Южного федерального университета, специалист в области системного программирования и разработки компиляторов.

Anton Pavlovich BAGLY – teaching assistant at Institute of mathematics, mechanics and computer science, Southern federal university, a specialist in system programming and compiler development.

email: taccessviolation@gmail.com

Материал поступил в редакцию 16 ноября 2019 года