



ИПМ им.М.В.Келдыша РАН

Абрау-2019 • Труды конференции



Труды XXI Всероссийской научной конференции

## Научный сервис в сети Интернет

А.П. Баглий

### Web среда анализа и преобразований программ для реконфигурируемых архитектур

#### ***Рекомендуемая форма библиографической ссылки***

Баглий А.П. Web среда анализа и преобразований программ для реконфигурируемых архитектур // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23-28 сентября 2019 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2019. — С. 76-84. — URL: <http://keldysh.ru/abrau/2019/theses/18.pdf> doi:[10.20948/abrau-2019-18](https://doi.org/10.20948/abrau-2019-18)

Размещена также [презентация к докладу](#)

# Web среда анализа и преобразований программ для реконфигурируемых архитектур

А.П. Баглий

*Южный Федеральный Университет*

**Аннотация.** Описывается опыт проектирования различных вариантов web-среды разработки (IDE) для Оптимизирующей распараллеливающей системы и компилятора на реконфигурируемую архитектуру на основе существующих инструментов, таких как Jupyter Notebook и Eclipse Che. Формируются требования к инструментам в составе Открытой распараллеливающей системы для поддержки их интеграции в web среду разработки, доступную в сети Интернет. Описывается процесс создания переносимого окружения для разработки модулей компилятора, демонстрации его работы и обучения навыкам разработки параллельных программ.

**Ключевые слова:** интегрированная среда разработки программ, распараллеливающий компилятор, контейнеризация, Web IDE, облачные вычисления, ПЛИС.

## Web environment for program analysis and transformation onto reconfigurable architectures

A.P. Bagly

*Southern Federal University*

**Abstract.** Experience of designing different outlines 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.

**Keywords:** integrated development environment, parallelizing compiler, containerization, web IDE, cloud computing, FPGA.

## 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.

Возникающие в процессе разработки компилятора проблемы 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 организовывать доступ к таким системам 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, в интерактивном режиме 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:

1. Быстрое развертывание окружения со всеми необходимыми инструментами для нового члена коллектива разработчиков или нового клиента системы.
2. Предоставление интуитивно понятного и простого диалогового интерфейса, позволяющего программисту получить схему для ПЛИС по исходной программе.
3. Организация демонстрации некоторых новых возможностей разрабатываемого компилятора.
4. Использование создаваемой системы для обучения разработке параллельных программ.
5. Использование частей создаваемой системы в научных исследованиях, где требуется предоставление доступа к результатам и удобство их воспроизведения.

Для демонстрации возможностей Оптимизирующей распараллеливающей системы [1] ранее были разработаны web-интерфейсы для демонстрации ее отдельных функций [2][3]. Общий вид первой версии web-интерфейса показан на рис. 1.

## Web-распараллеливатель

Предыдущий шаг Следующий шаг

<b>Шаг 1</b> Режим преобразования	<b>Шаг 2</b> Источник программы	<b>Шаг 3</b> Ввод программы	<b>Шаг 4</b> Выполнение преобразования
--------------------------------------	------------------------------------	--------------------------------	---

**Выберите источник программы**

- Выбор примера из библиотеки
- Загрузка файла<sup>1</sup>
- Ввод текста программы в редакторе<sup>2</sup>

---

<sup>1</sup> - если программа состоит из нескольких файлов, эти файлы можно скачать в zip-архив и загрузить в виде одного файла.  
<sup>2</sup> - если Ваша программа состоит из одного файла, текст этой программы можно ввести через редактор.

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, впрочем, далеко не полностью. Пользователь не имеет возможности напрямую управлять процессором, а от разработчиков требуется создание специальных сценариев использования системы.

Данная работа ставит задачу создания более универсальной среды разработки, доступной в сети Интернет, которая бы:

1. Была основана на существующих наработках, широко применяемых в среде облачных вычислений.
2. Решала остальные указанные проблемы.
3. 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.

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 web-среды разработки для

синтеза программ для ПЛИС, основанный на OPS и существующих облачных IDE с открытым исходным кодом.

Далее в разделе «Постановка задачи» описываются более подробные требования к составным частям компилятора, которые выработаны для создания web-IDE на его основе. В разделе «Использование Jupyterlab в OPS» описывается подход создания среды разработки на основе OPS на языке C++ на основе интерактивных тетрадей в браузере. В разделе «Результаты» описываются основные изменения, которые позволяют создать минимально функциональное окружение разработки программ для ПЛИС на основе разрабатываемого компилятора.

### **Постановка задачи**

В среде облачных вычислений существует множество широко используемых web-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:

1. Создание и использование контейнеризованных рабочих сред для разработчиков системы с помощью docker решает проблему быстрого развертывания окружения с повторяемыми характеристиками.
2. Управление рабочей средой с помощью расширяемого API решает проблему удобства демонстрации выбранных функций системы.
3. Возможность создания расширений для всех архитектурных частей системы позволяет добавить в нее новые представления для демонстрации важных функций и визуализации результатов.

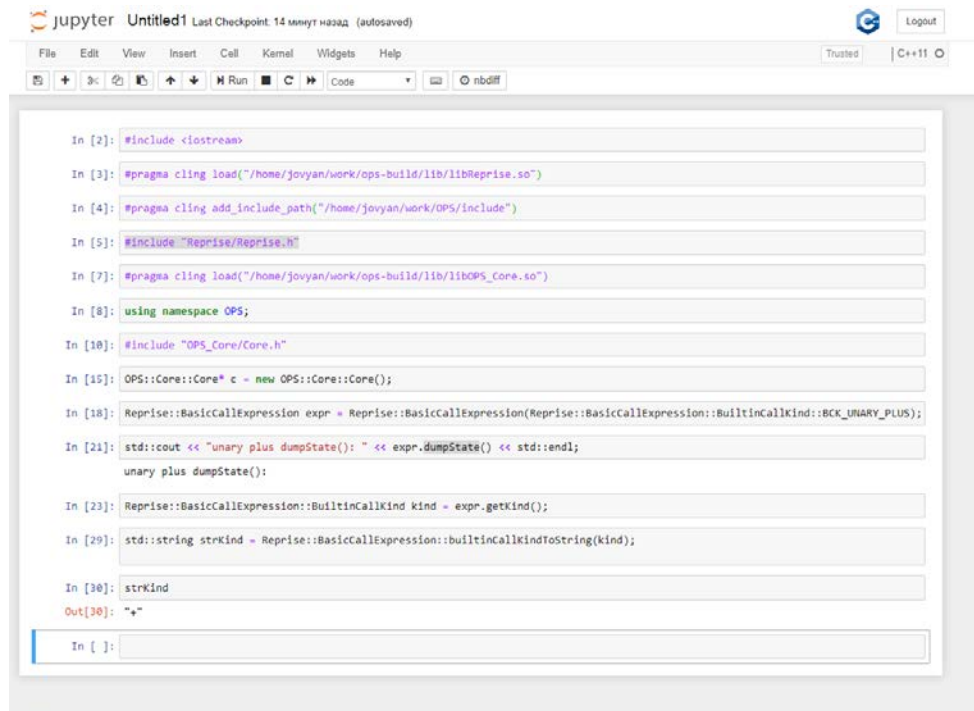
Однако использование данной среды разработки в качестве основы для собственной облачной среды требует модификации как самой Eclipse Che, так и разрабатываемого компилятора.

В общем виде, требуется:

1. Разбиение процесса работы компилятора на этапы, между которыми возможен экспорт или визуализация промежуточных результатов.
2. Добавление новых представлений в интерфейс IDE и соответствующих им методов API рабочей среды.
3. Создание набора контейнеров для обеспечения работы компилятора.
4. Обеспечение более слабых связей между модулями компилятора для удобства их использования со стороны облачной IDE.

## Использование Jupyterlab в OPS

Jupyterlab [9] является на данный момент самым простым в использовании средством интерактивного предоставления результатов вычислений, это web-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 OPS в виде интерактивных тетрадей в браузере, как показано, например на Рис. 2.



```
In [2]: #include <iostream>

In [3]: #pragma cling load("/home/jovyan/work/ops-build/lib/libReprise.so")

In [4]: #pragma cling add_include_path("/home/jovyan/work/OPS/include")

In [5]: #include "Reprise/Reprise.h"

In [7]: #pragma cling load("/home/jovyan/work/ops-build/lib/libOPS_core.so")

In [8]: using namespace OPS;

In [10]: #include "OPS_Core/Core.h"

In [15]: OPS::Core::Core* c = new OPS::Core::Core();

In [18]: Reprise::BasicCallExpression expr = Reprise::BasicCallExpression(Reprise::BasicCallExpression::BuiltinCallKind::BCK_UNARY_PLUS);

In [21]: std::cout << "unary plus dumpState(): " << expr.dumpState() << std::endl;
unary plus dumpState():

In [23]: Reprise::BasicCallExpression::BuiltinCallKind kind = expr.getKind();

In [29]: std::string strKind = Reprise::BasicCallExpression::builtinCallKindToString(kind);

In [30]: strKind
Out[30]: "+"

In [ ]:
```

Рис. 2. Пример выполнения функций OPS 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.

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

1. Демонстрировали ключевые функции модулей системы, например работу со внутренним представлением, выполнение преобразований программ, что особенно полезно в процессе обучения.
2. Предоставляли пользователям возможность проведения экспериментов по исследованию производительности и эффектов различных преобразований над тестируемой программой с помощью вспомогательных сервисов.
3. Предоставляли разработчикам системы простой и удобный интерфейс для обмена результатами работы, проведения тестов.

## Результаты

Для решения поставленных задач были реализованы следующие нововведения:

1. Набор контейнеров для надежной и повторяемой сборки самой ОРС, компилятора на ее основе, запуска этого компилятора на тестовых программах.
2. Дополнительные сервисы для симуляции выполнения сгенерированных программ на ПЛИС, для тестирования производительности преобразованных программ на языке С по принципу «черного ящика», аналогичные ранее разработанной системе тестирования из [6]
3. Процесс работы компилятора разбит на этапы, которыми можно управлять:
  - а. Выбор участков программы для выполнения на ПЛИС может быть автоматическим или пользователь может выделить нужные фрагменты псевдокомментариями.
  - б. Компилятор принимает на вход набор параметров, описывающих необходимые действия.
  - в. Возможные действия включают экспорт промежуточных результатов для анализа и визуализации, генерацию выходной программы и т.п.
4. Для нескольких выбранных функций компилятора и ОРС созданы соответствующие представления для визуализации данных в среде разработки, например для статического профилировщика [7] и для функции построения обобщенных конвейеров [8]

На рис. 3 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.

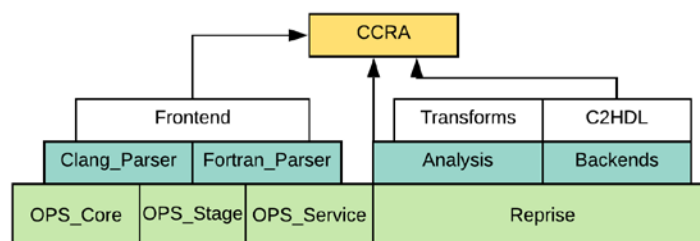


Рис. 3. Набор модульных компонент OPS, включая компилятор для ПЛИС (CCRA)

Дальнейшая работа ведется в двух направлениях:

1. Поддержка работы с кодом компилятора в облачной IDE, то есть обеспечение удобства разработки самого компилятора.
2. Расширение функциональности специализированной облачной IDE, позволяющей использовать разрабатываемый компилятор.

При этом 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.

### **Выводы**

На основе современных облачных интегрированных сред разработки программ возможно создание специализированных инструментов, позволяющих облегчить разработку, использование и внедрение в процесс обучения сложных программных систем.

Для того, чтобы разработать функциональную web-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.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 18-37-00179

### **Литература**

1. Оптимизирующая распараллеливающая система URL: [www.ops.rsu.ru](http://www.ops.rsu.ru) (дата обращения: 25.07.19).
2. Б.Я. Штейнберг, А.Н. Аллазов, Е.В. Алымова, А.П. Баглий, С.А. Гуда, Д.В. Дубров, Е.Н. Кравченко, Р.И. Морылев, А.С. Рошаль, М.В. Юрушкин, Р.Б. Штейнберг Web-ориентированный автоматический распараллеливатель программ // Параллельные вычислительные технологии (ПАВТ'2014), Труды международной научной конференции. Ростов-на-Дону: 1-3 апреля 2014.
3. Е.В. Алымова, Е.Н. Кравченко, Р.И. Морылев, М.В. Юрушкин, Б.Я. Штейнберг 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 2012г, г. Новороссийск). — М.: Изд-во МГУ, 2012.

4. B. Y. Steinberg, A. P. Buglii, D. V. Dubrov, Y. V. Mikhailuts, O. B. Steinberg, R. B. Steinberg A Project of Compiler for a Processor with Programmable Accelerator // Procedia Computer Science. 2016. №101. с. 435-438.
5. Localhost is Killing Software Delivery // Codenvy blog URL: <https://blog.codenvy.com/localhost-is-killing-software-delivery-8c93cd49328> (дата обращения: 25.07.19).
6. Б.Я. Штейнберг, Е.В. Алымова, А.П. Баглий, Р.И. Морылев, З.Я. Нис, В.В. Петренко, Р.Б. Штейнберг Автоматизация тестирования элементов высокопроизводительного программного комплекса // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность. Труды Всероссийской суперкомпьютерной конференции (21-26 сентября 2009г., г. Новороссийск) М.: МГУ им. М.В. Ломоносова, 2009, с.287-292.
7. С.В. Полуян 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. А. П. Баглий, Д. В. Дубров, Б. Я. Штейнберг, Р. Б. Штейнберг Повторное использование ресурсов при конвейерных вычислениях // Научный сервис в сети Интернет: труды XIX Всероссийской научной конференции (18-23 сентября 2017г., г. Новороссийск) М.: ИПМ им. М.В. Келдыша, 2017, с.43-46. — doi:10.20948/abrau-2017. — URL: <http://keldysh.ru/abrau/2017/proc.pdf>
9. Thomas Kluyver 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. — doi: 10.3233/978-1-61499-649-1-87
10. Проект Xeus-cling на Github // url: <https://github.com/QuantStack/xeus-cling> (дата обращения: 25.07.19)

## References

1. Optimiziruiushchaia rasparrallelivaiushchaia sistema URL: [www.ops.rsu.ru](http://www.ops.rsu.ru) (data obrashcheniia: 25.07.19).
2. B.Ia. Shteinberg, A.N. Allazov, E.V. Alymova, A.P. Baglii, S.A. Guda, D.V. Dubrov, E.N. Kravchenko, R.I. Morylev, A.S. Roshal, M.V. Iurushkin, R.B. Shteinberg Web-orientirovannyi avtomaticheskii rasparrallelivatel programm // Parallelnye vychislitelnye tekhnologii (PAVT'2014), Trudy mezhdunarodnoi nauchnoi konferentsii. Rostov-na-Donu: 1-3 apreliia 2014.
3. E.V. Alymova, E.N. Kravchenko, R.I. Morylev, M.V. Iurushkin, B.Ia. Shteinberg Rasparrallelivanie i optimizatsiia programm s pomoshchiu Web-uskoritelia ORS // Nauchnyi servis v seti Internet: poisk novykh reshenii, Trudy XIV

- Mezhdunarodnoi superkompiuternoi konferentsii (17-22 sentiabria 2012g, g. Novorossiisk). — M.: Izd-vo MGU, 2012.
4. B. Y. Steinberg, A. P. Bugliy, D. V. Dubrov, Y. V. Mikhailuts, O. B. Steinberg, R. B. Steinberg A Project of Compiler for a Processor with Programmable Accelerator // *Procedia Computer Science*. 2016. №101. s. 435-438.
  5. Localhost is Killing Software Delivery // *Codenvy blog* URL: <https://blog.codenvy.com/localhost-is-killing-software-delivery-8c93cd49328> (data obrashcheniia: 25.07.19).
  6. B.Ia. Shteinberg, E.V. Alyмова, A.P. Baglii, R.I. Morylev, Z.Ia. Nis, V.V. Petrenko, R.B. Shteinberg Avtomatizatsiia testirovaniia elementov vysokoproizvoditelnogo programmnoho kompleksa // *Nauchnyi servis v seti Internet: masshtabiruemost, paralelnost, effektivnost. Trudy Vserossiiskoi superkompiuternoi konferentsii (21-26 sentiabria 2009g., g. Novorossiisk) M.: MGU im. M.V. Lomonosova, 2009, c.287-292.*
  7. S.V. Poluian Profilirovanie i ego primenenie v dialogovom optimiziruiushchem rasparallelvatele // *Nauchnyi servis v seti Internet: superkompiuternye tsentry i zadachi: Trudy Mezhdunarodnoi superkompiuternoi konferentsii (20-25 sentiabria 2010g., g. Novorossiisk).* — M.: Izd-vo MGU, c. 652-653.
  8. A. P. Baglii, D. V. Dubrov, B. Ia. Shteinberg, R. B. Shteinberg Povtornoie ispolzovanie resursov pri konveiernykh vychisleniiax // *Nauchnyi servis v seti Internet: trudy XIX Vserossiiskoi nauchnoi konferentsii (18-23 sentiabria 2017g., g. Novorossiisk) M.: IPM im. M.V. Keldysha, 2017, c.43-46.* — doi:10.20948/abrau-2017. — URL: <http://keldysh.ru/abrau/2017/proc.pdf>
  9. Thomas Kluyver 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.* — doi: 10.3233/978-1-61499-649-1-87
  10. Project Xeus-cling na Github // url: <https://github.com/QuantStack/xeus-cling> (data obrashcheniia: 25.07.19)