



Веб-лаборатория для суперкомпьютерного многомасштабного моделирования задач напыления

Н. И. Тарасов

Институт прикладной математики имени М. В. Келдыша РАН (ИПМ РАН),
Москва, Российская Федерация

ORCID: 0000-0002-2342-1292, e-mail: nikita_tarasov@imamod.ru

В. О. Подрыга

Институт прикладной математики имени М. В. Келдыша РАН (ИПМ РАН),
Москва, Российская Федерация

ORCID: 0000-0001-7874-6978, e-mail: pvictoria@list.ru

С. В. Поляков

Институт прикладной математики имени М. В. Келдыша РАН (ИПМ РАН),
Москва, Российская Федерация

ORCID: 0000-0003-1859-9034, e-mail: polyakov@imamod.ru

Аннотация: Работа посвящена анализу принципов построения веб-лабораторий, предназначенных для суперкомпьютерного математического моделирования сложных физических процессов и явлений. Основными целями подобных цифровых платформ являются: автоматизация вычислительных экспериментов, формирование базы знаний и обеспечение совместной работы исследователей в выбранной предметной области. В работе рассмотрены существующие сегодня решения и принципы построения предметно-ориентированных платформ. На основе выполненного анализа была разработана и практически реализована веб-лаборатория, связанная с решением задач напыления. В работе приведены архитектура и детали программной реализации платформы. Основными ее преимуществами являются возможность динамического встраивания проблемно-ориентированных приложений и удаленных вычислительных ресурсов. Реализованная веб-лаборатория была протестирована посредством проведения серии вычислительных экспериментов по модельным задачам сверхзвукового холодного газодинамического напыления. Также в работе предлагаются направления для последующего развития оригинальной веб-лаборатории.

Ключевые слова: веб-лаборатория, цифровая платформа, многомасштабные подходы, автоматизация вычислительных экспериментов, суперкомпьютерные вычисления.

Благодарности: Работа выполнена при финансовой поддержке Российского научного фонда (проект № 21-71-20054). Расчеты проведены с помощью гибридного суперкомпьютера K60, установленного в Центре коллективного пользования ИПМ имени М. В. Келдыша РАН.

Для цитирования: Тарасов Н.И., Подрыга В.О., Поляков С.В. Веб-лаборатория для суперкомпьютерного многомасштабного моделирования задач напыления // Вычислительные методы и программирование. 2023. 24, № 4. 463–484. doi 10.26089/NumMet.v24r432.



Web laboratory for supercomputer multiscale modeling of spraying problems

Nikita I. Tarasov

Keldysh Institute of Applied Mathematics, Moscow, Russia
ORCID: 0000-0002-2342-1292, e-mail: nikita_tarasov@imamod.ru

Viktoriia O. Podryga

Keldysh Institute of Applied Mathematics, Moscow, Russia
ORCID: 0000-0001-7874-6978, e-mail: pvictoria@list.ru

Sergey V. Polyakov

Keldysh Institute of Applied Mathematics, Moscow, Russia
ORCID: 0000-0003-1859-9034, e-mail: polyakov@imamod.ru

Abstract: The work is devoted to the analysis of the principles of constructing the web laboratories intended for supercomputer mathematical modeling of complex physical processes and phenomena. The main goals of such digital platforms are the automation of computational experiments, formation of a knowledge base and ensuring collaboration of researchers in a subject area. The work discusses the solutions that exist today and the principles for constructing subject-oriented platforms. Based on the analysis performed, a web laboratory related to solving spraying problems was developed and implemented. The work provides the architecture and details of the software implementation of the platform. Its main advantages are the ability to dynamically integrate problem-oriented applications and remote computing resources. The implemented web laboratory was tested through a series of computational experiments on model problems of supersonic cold gas-dynamic spraying. The work suggests directions for the subsequent development of the original web laboratory.

Keywords: web laboratory, digital platform, multiscale approaches, computational experiments automation, supercomputer calculations.

Acknowledgements: The work was carried out with the support of the Russian Science Foundation, project No. 21–71–20054. The calculations were performed on the hybrid supercomputer K60 installed in the Center of Collective Usage of KIAM RAS.

For citation: N. I. Tarasov, V. O. Podryga, S. V. Polyakov, “Web laboratory for supercomputer multiscale modeling of spraying problems,” *Numerical Methods and Programming*. 24 (4), 463–484 (2023). doi 10.26089/NumMet.v24r432.

1. Введение. Современные исследования сложных физико-химических процессов проводятся на основе решения сопряженных задач в рамках многомасштабных подходов. Это относится и к моделированию процессов напыления. При решении данного класса задач выделяются три масштабных уровня: микроуровень взаимодействия частиц напылителя с подложкой, мезоуровень взаимодействия агломераций частиц, макроуровень динамики многочастичной многофазной среды [1]. При численной и программной реализации такого подхода формируется набор прикладных программ, составляющих вычислительное ядро. Зачастую они представляют собой пакетные приложения с различными описаниями расчетной области, физико-химических свойств веществ и протекающих процессов, сервисных конфигурационных файлов. Проведение вычислительного эксперимента с помощью таких приложений дополнительно усложняется строгой последовательностью и сложной логикой их запуска. К тому же при решении многих практически важных задач по причине их высокой вычислительной емкости приходится использовать мощные вычислительные кластеры или суперкомпьютеры, расположенные удаленно. Существенно упростить и автоматизировать процесс подготовки и проведения вычислительного эксперимента в этом случае могут интерактивные информационные среды, которые объединяют в себе вычислительные ядра и инструменты систем автоматизированного проектирования (САПР).



Текущее состояние веб-технологий позволяет реализовать новые архитектурные принципы работы больших САПР-комплексов. Если ранее подобные комплексы представляли собой локальные программные решения, примерами которых могут выступить так называемые “толстые” клиенты Ansys [2] или FlowVision [3], то сегодня широкое применение находят веб-платформы. Они направлены на решение сложных прикладных задач из различных областей средствами удаленных суперкомпьютерных систем. Преимуществом подобных подходов являются пониженные требования к персональному вычислительному устройству пользователя и установленному на устройстве программному обеспечению. Также преимуществом является реализация посредством удаленного доступа многопользовательского режима. Наиболее интересными в этом случае являются платформы, предоставляющие интерактивный графический интерфейс и позволяющие проводить полный цикл вычислительного эксперимента в рамках единого автоматизируемого и унифицированного процесса взаимодействия пользователя с предлагаемым сервисом и различными вычислительными ресурсами. Подобные программные продукты получили название веб-лабораторий. Они позволяют не только существенно снизить порог вхождения в прикладную область, но и сделать процесс моделирования интерактивным и наглядным, что является актуальным и в образовательной сфере. Многопользовательский подход позволяет сформировать процесс совместной экспертной оценки как при подготовке исходных данных, так и при анализе получаемых результатов. Настоящее исследование посвящено развитию цифровой платформы KIAM Digital Tool [4, 5] в части интеграции программных кодов, подготовки и управления вычислительными экспериментами на удаленных суперкомпьютерах. Прототип системы, описанный в работе [6], был переведен на язык программирования TypeScript [7], кроме того существенной переработке подверглась и серверная часть веб-платформы.

Работа далее структурирована следующим образом. В разделе 2 рассматриваются существующие сторонние решения и подходы к реализации цифровых платформ. Раздел 3 посвящен описанию общих принципов построения цифровой платформы KIAM Digital Tool, включая описание взаимодействия сервера с удаленными суперкомпьютерами и встраивания проблемно-ориентированных приложений, их последующего комплексирования. В разделе 4 приведены результаты тестирования полученной веб-лаборатории на задачах сверхзвукового холодного газодинамического напыления. Раздел 5 содержит итоги исследования и дальнейшие направления развития представленной разработки.

2. Существующие решения. Благодаря богатству и широте возможностей веб-технологии на сегодняшний день существует большое количество подходов и конкретных реализаций цифровых платформ для математического моделирования [8, 9]. Каждая из них, в общем, представляет из себя клиент-серверное приложение. Однако существенными являются детали взаимодействия платформы с приложениями и вычислительными ресурсами. Рассмотрим далее актуальные типы и экземпляры платформ.

В качестве первой группы приведем цифровые платформы, предназначенные для обслуживания конкретной предметной области. Для данной группы характерна модель обслуживания SaaS (программное обеспечение как услуга). Примерами здесь могут выступить Prove Design [10] и Nucleonica [11]. Они предоставляют пользователю развитый интерактивный графический интерфейс, реализованный посредством веб-технологий. Ограничением представителей этой группы является отсутствие возможности встраивания сторонних прикладных программ.

Prove Design предназначен для моделирования прочностных процессов и включает структурный, модальный и термоупругий анализы. Предлагаемый графический интерфейс позволяет задавать материалы исследуемых объектов и граничные условия непосредственно на трехмерной визуализации геометрии расчетной области. Числовые значения величин нагрузок, коэффициентов материалов и других управляющих параметров задаются посредством специальных форм. В рамках единого веб-интерфейса производится запуск задачи на счет, включающий генерацию разностной сетки и проведение расчета, в процессе выполнения которого происходит информирование пользователя о стадиях его выполнения. По завершении вычислительного эксперимента на трехмерной модели объекта исследования возможно отобразить полученные поля цветовой заливкой. Проведенные расчеты сохраняются в базе данных веб-лаборатории, обеспечивая доступ к результатам по требованию пользователя.

Nucleonica [11] представляет собой веб-сайт, посвященный ядерной физике, в рамках которого реализован портал приложений. Данный портал содержит множество прикладных веб-программ, разделенных на следующие категории: расчет, отображение данных, знания и нуклиды. Графический интерфейс, предлагаемый пользователю, зависит от типа и конкретного приложения, обладает высокой специализацией к предметной области и интерактивностью, обеспечивая удобство использования как в исследовательской, так и в образовательной деятельности.

Иной подход представляют цифровые платформы модели PaaS (платформа как услуга), предназначенные для интеграции прикладных программ и автоматизации вычислительных экспериментов. К данному классу можно отнести pSeven Enterprise [12] и Everest [13].

Цифровая платформа pSeven Enterprise предоставляется к установке на выделенной серверной машине. В качестве расчетных узлов в стандартной поставке предлагается использовать сеть Kubernetes, однако существует возможность использования суперкомпьютерных ресурсов с помощью установки на целевом вычислителе специальной программы-агента. pSeven Enterprise предлагает широкие возможности комплексирования проблемно-ориентированных приложений посредством интерактивного графического интерфейса, в рамках которого каждая программа представляется узлом графа. Последующее задание ребер графа позволяет сформировать зависимости входных и выходных данных, на основании которых платформа формирует последовательность исполнения с возможностью параллельного запуска для независимых узлов, автоматизируя процесс проведения вычислительного эксперимента в целом.

Everest предлагает к использованию платформу [14], выступающую в роли центра приложений с возможностью интеграции вычислительных ресурсов различного типа. Регистрация на сервисе является свободной, исходный код закрыт. Интеграция вычислительных ресурсов происходит с помощью установки приложения-агента, обеспечивающего взаимодействие с предустановленной системой прохождения пользовательских заданий (поддерживаются в том числе Slurm Workload Manager и TORQUE). Встраивание приложений реализуется посредством текстовых мета-описаний. Запуск программного обеспечения осуществляется с помощью веб-интерфейса либо Python-скриптов с использованием поставляемой библиотеки-обертки REST-API сервера цифровой платформы. Именно с помощью второго подхода достигаются широкие возможности автоматизации процесса проведения вычислительного эксперимента, однако требующего от пользователя написания сервисного кода.

Широкую популярность обрела программная платформа HUBZero [15], с открытым исходным кодом под лицензией GNU GPL2, предназначенная для конструирования тематических веб-лабораторий и поддерживающая подход свободной интеграции сторонних программных средств. Реализованные на базе HUBZero сервисы, такие как NanoHUB [16] и Nemo5 [17], предлагают не только информационные материалы, но и широкий набор проблемно-ориентированных приложений, называемых в рамках подхода “инструменты” (“Tools”), с возможностью их встраивания. “Инструменты” в HUBZero запускаются в изолированных контейнерах на базе операционной системы Linux, доступ к которым осуществляется с помощью “тонкого” клиента по VNC-протоколу [18]. Допускается использование программного обеспечения, пригодного для запуска на Linux/X11 с графическим пользовательским интерфейсом. При его отсутствии разработчику предлагается реализовать его с помощью библиотеки Rapture Toolkit Application. Обобщенное встраивание приложений в HUBZero может быть специализировано под предметную область посредством обращений к серверу и соответствующей обработки данных. Примером подобной реализации может служить NemoVis [19] для проекта Nemo5. Кроме того, в рамках HUBZero можно осуществить запуск сервера Jupyter Notebook [20].

Система Jupyter Notebook, распространяемая по лицензии “3-Clause BSD”, широко используется сегодня в научных исследованиях, их презентации и образовательном процессе. Данное программное обеспечение представляет собой сервер интерактивных документов, позволяющих включать в описательный текст участки исходного кода и результаты его исполнения. Являясь развитием идей IPython Notebook [21], предназначенной исключительно для языка программирования Python, Jupyter Notebook сегодня поддерживает большее их количество, в том числе C++. Более того, посредством специализированных инструментов, например Voilà [22], на основе интерактивного документа может быть сформировано автономное веб-приложение. Таким образом, Jupyter Notebook может выполнять многие функции веб-лаборатории, включая коллаборацию исследователей и автоматизацию вычислений. Более того, известны возможности его применения на ресурсах суперкомпьютерного уровня [23].

Существует также подход, при котором функциональность веб-лаборатории обеспечивается удаленным рабочим столом (модель DaaS — рабочий стол как услуга). Примером подобных платформ могут выступить UniHUB [24] и KIAM Multilogin [25]. Принцип их работы состоит в предоставлении по требованию пользователя удаленного доступа, в том числе посредством “тонкого” клиента, к контейнеру (виртуальной машины и др.) с операционной системой. При этом, в рамках предметной области, предполагается наличие предустановленного программного обеспечения, направленного на проведение полного цикла вычислительного эксперимента. Основным преимуществом данного подхода являются широкие функциональные возможности, поскольку конечному пользователю предоставляется доступ к операцион-



ной системе. Как следствие этого выявляется и недостаток в повышенных требованиях к квалификации пользователя, поскольку от него требуются навыки использования базовой операционной системы и разнообразного прикладного программного обеспечения.

Подводя итог рассмотрения различных подходов, отметим, что существующие сегодня программные платформы для разработки веб-приложений позволяют реализовывать графические интерфейсы, не уступающие в интерактивности и наглядности “толстым” клиентам. Более того, набор технологий HTML/CSS/JS используют в том числе при разработке “толстых” графических интерфейсов (см. Electron [26]). В связи с этим представляется актуальной реализация веб-клиентов без использования технологии VNC, включающей передачу изображений рабочего стола операционной системы, с обеспечением при этом полноценных возможностей САПР-систем в веб-браузере. Кроме этого, возрастающая вычислительная сложность рассматриваемых задач требует наличия возможности использования множества удаленных суперкомпьютерных ресурсов, что не всегда возможно в уже существующих цифровых платформах, рассмотренных выше. Таким образом, текущее направление нашей разработки KIAM Digital Tool стремится к объединению подходов групп Prove Design и Everest.

3. KIAM Digital Tool.

3.1. Общее описание. Порядок действий при разработке и отладке исследовательских кодов для проведения комплексных вычислительных экспериментов с использованием удаленных суперкомпьютеров представляет собой нетривиальную задачу. Она требует от исследователя не только высокой квалификации в предметной области и разработке программного обеспечения, но и уверенного владения возможностями Unix, поскольку подавляющее большинство вычислителей данного уровня используют именно данное семейство операционных систем [27]. При этом, зачастую, удаленный доступ к суперкомпьютеру реализуется по протоколу SSH посредством терминального управления, с передачей файлов по протоколу SFTP. Иные возможности удаленного доступа могут быть существенно ограничены в целях обеспечения информационной безопасности. Запуск приложения на узлах вычислительного ресурса также осуществляется специальным образом — с помощью системы прохождения пользовательских заданий, отвечающей за организацию распределения времени между пользователями системы и выделение расчетных узлов, примером которой может выступить Система управления прохождением параллельных заданий (СУП-ПЗ) [28]. Таким образом организована работа с суперкомпьютерами Центра коллективного пользования ИПМ имени М. В. Келдыша РАН [29]. В результате исследователь вынужден использовать стандартные средства командной оболочки при подготовке программы, ее конфигурировании и запуске, а также слежении за прохождением задачи при выполнении верификационных и валидационных тестовых расчетов. Последующий анализ, что является особенно критичным для трехмерных расчетов, требует передачи большого количества данных, при использовании стандартных клиентских средств визуализации, либо написания дополнительных утилит обработки. Конечный пользователь проблемно-ориентированных приложений (расчетчик) сталкивается с аналогичными трудностями при проведении серий вычислительных экспериментов. Кроме того, существует необходимость обеспечения длительного доступа и каталогизации проведенных вычислительных экспериментов для обработки и подготовки отчетов на основе полученных материалов.

Рассмотренный процесс взаимодействия с удаленным вычислителем возможно существенно автоматизировать. Для этого сформулируем следующие базовые требования к веб-лаборатории:

- 1) разделение пользователей цифровой платформы по группам ответственности;
- 2) обеспечение унифицированного авторизованного доступа к множеству суперкомпьютеров;
- 3) возможность встраивания прикладных программ, включающая следующие функциональные возможности:
 - хранение исходных кодов приложений;
 - подготовка приложения к запуску, предполагающая отправку и сборку на удаленном вычислительном ресурсе;
 - конфигурирование пользовательского задания;
 - управление и слежение за пользовательским заданием на удаленном вычислительном ресурсе;
 - анализ результатов работы пользовательского задания;
- 4) возможность группировки проблемно-ориентированных приложений в сценарии, направленные на решение конкретной задачи;

- 5) обеспечение хранения и каталогизации проведенных вычислительных экспериментов;
- 6) возможность многопользовательской работы над комплексными физико-математическими постановками.

Решаемые цифровой платформой задачи в рамках серверной части требуют быстрой работы подпрограмм ввода-вывода, поскольку обеспечивают такие функции, как: отправка файлов на удаленный вычислительный ресурс, сборка приложений, слежение за выполнением пользовательского задания, а также сетевое взаимодействие. В связи с этим было решено использовать окружение Node.js [30], обеспечивающее высокую производительность ввода-вывода благодаря неблокирующему исполнению данных операций посредством цикла событий, позволяющее также производить обработку множественных клиентских запросов в одном потоке. Node.js представляет собой расширение языка JavaScript, предоставляющее возможности взаимодействия с операционной системой, в отличие от исполнения в веб-браузере. В связи со сказанным предыдущая версия сервера веб-лаборатории [4] была переработана с использованием программной платформы NestJS [31] на языке программирования TypeScript [7]. Основной целью TypeScript является введение в JavaScript системы типов, существенно облегчающей процесс разработки и отладки приложений, поскольку большинство современных интегрированных сред разработки имеет широкую его поддержку. Дополнительным преимуществом также является наличие этапа компиляции, позволяющего обнаружить основные ошибки программирования до этапа тестирования и использования приложения. При этом исходный код на языке программирования TypeScript в результате работы компилятора переводится в стандартный JavaScript, исполняемый как в среде Node.js, так и в веб-браузере.

Множество возможных архитектурных подходов, доступных при использовании Node.js, приводит к сложности выстраивания процесса и дисциплины разработки, отладки и тестирования. Программная платформа NestJS призвана решить данную проблему посредством строгого определения архитектуры веб-приложения, представляющей собой реализацию слабосвязанных модулей. Данный подход позволяет строить масштабируемые и расширяемые веб-серверы, а широкое использование декораторов позволяет добиться высокой лаконичности исходного кода. Отметим также, что NestJS призван решить вопрос архитектуры — конкретные функциональные возможности обеспечиваются сторонними библиотеками. Например, по умолчанию для маршрутизации и сетевого взаимодействия используется `express.js` [32], однако возможно использование иных решений посредством специальных классов-адаптеров. Аналогичным образом построено взаимодействие с базами данных — функционал реализуется сторонними решениями объектно-реляционного отображения (по умолчанию это `TypeORM` [33], поддерживающая в том числе `sqlite`, `MySQL`, `PostgreSQL` и др.). В рамках текущей реализации были выбраны рекомендуемые NestJS библиотеки, использующие в качестве базы данных `MySQL`.

Клиентская часть веб-лаборатории [5] унаследовала архитектуру одностраничного приложения с применением реактивных компонентов, реализуемых с помощью каркаса `Vue.js` [34]. Реактивный подход заключается в отслеживании изменений данных (при выполнении асинхронного запроса данных с сервера или в результате пользовательского взаимодействия), приводящих к перерисовке элементов графического интерфейса, описанного посредством HTML-подобного шаблона. Программная платформа `Vue.js` реализует этот механизм посредством построения виртуальной объектной модели документа, в которую вносятся зафиксированные изменения данных и которая затем синхронизируется с реальной объектной моделью. С момента начала разработки цифровой платформы была выпущена третья версия `Vue.js`, получившая полную поддержку TypeScript, а также возможность использования программного интерфейса композиции (`Composition API`), наряду с программным интерфейсом свойств (`Options API`). В связи с этим клиентская часть веб-платформы была переведена на актуальную версию `Vue.js` с использованием подхода `Composition API` и языка TypeScript. Преимущества использования TypeScript обсуждены выше при описании изменений серверной части. Подход `Composition API` позволяет, в отличие от `Options API`, обеспечить большую локальность переменных и кода, их использующего, большую его гибкость и переиспользование. Реализация клиента веб-лаборатории использует программную платформу `Quasar` [35], позволяющую использовать единую кодовую базу для одностраничных приложений, серверной отрисовки и даже “толстых” клиентов, полученных с помощью `Electron` [26]. `Quasar` также предоставляет широкий набор реактивных компонентов графического интерфейса и элементов пользовательского управления.

Отметим особенности клиент-серверного взаимодействия `KIAM Digital Tool`. Веб-лаборатория должна обеспечивать пользователей актуальной информацией о состоянии удаленного вычислительного ресурса, его очереди, текущих статусах заданий. Поскольку обновление данных сведений в большей степени зависит от удаленного вычислителя, а не от пользовательского взаимодействия, для их доставки и актуа-



лизации могут быть применены следующие подходы. С одной стороны, возможно генерировать отправку запросов к серверу по таймеру, что приводит к генерации излишней нагрузки. С другой стороны, сегодня нашла широкое распространение технология веб-сокеты [36], позволяющая реализовать событийный механизм в рамках стандартного HTTP-соединения, хорошо интегрируемый в окружение Node.js. В настоящей реализации цифровой платформы веб-сокеты используются также для всех основных клиент-серверных обменов, за исключением авторизации и обеспечения доступа к содержимому удаленных файлов, что позволяет существенно сократить количество сервисной информации — HTTP-заголовок отправляется единожды, в момент установления соединения. Более того, объект веб-сокета на стороне сервера может выступать в роли кэша текущего соединения, храня в себе информацию о пользователе, его роль и тому подобные данные, полученные из базы.

Общая схема цифровой платформы и ее соединений представлена на рис. 1. Масштабирование серверной части достигается с помощью запуска дополнительных процессов посредством утилиты pm2 [37]. Каждый из экземпляров идентичен в части приема и обработки пользовательских запросов и соединений, назначаемых в соответствии с принятым механизмом балансировки. Опрос вычислителей об их текущем состоянии и состоянии пользовательских заданий разделяется между процессами по следующему принципу:

- 1) первый экземпляр сервера по установленному таймеру запрашивает от базы данных идентификаторы требующих обновления данных;
- 2) полученные идентификаторы распределяются по оставшимся процессам с помощью брокера сообщений (в случае текущей реализации цифровой платформы используется Redis [38]);
- 3) каждый из процессов, получивших запрос на обновление, устанавливает соединение с удаленными вычислительными ресурсами и обновляет требуемую информацию в соответствии с установленной системой прохождения пользовательских заданий и операционной системой;
- 4) полученные изменения сохраняются в базе данных, посредством брокера сообщений оповещаются все запущенные серверные процессы, рассылающие полученные обновления присоединенным клиентам.

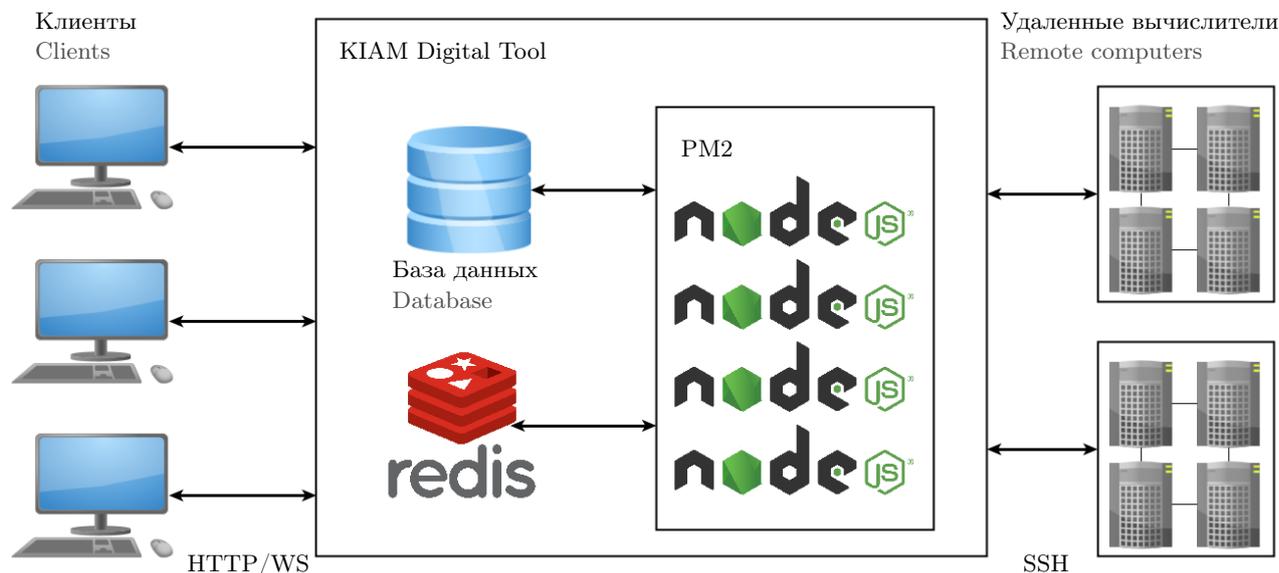


Рис. 1. Схема KIAM Digital Tool

Fig. 1. KIAM Digital Tool scheme

Приведенный набор технологий и общий принцип построения архитектуры KIAM Digital Tool позволяют выполнить поставленные задачи, получив в итоге интерактивную, масштабируемую и современную веб-лабораторию, направленную на решение комплексных многомасштабных задач. Стандартный процесс начала пользовательского взаимодействия включает в себя авторизацию с помощью веб-формы, затем, при успехе, происходит его допуск к основной части графического интерфейса и установление соединения веб-сокета, используемого для обменов между сервером и интерфейсом. Отметим, что при регистрации

каждому пользователю назначается роль (либо список ролей), определяющая возможности его взаимодействия с веб-лабораторией. Далее рассмотрим основные функциональные сущности цифровой платформы, обеспечивающие возможности встраивания проблемно-ориентированных приложений, вычислительных ресурсов и удаленный запуск пользовательских заданий.

3.2. Встраивание прикладных программ. Многომасштабные модели предполагают наличие множества различных расчетных программ и утилит, направленных на решение единой прикладной задачи, включающих приложения для подготовки исходных данных, проведения расчетов, конвертации, а также обработки полученных результатов. В связи с данным фактом цифровая платформа должна обеспечивать возможности динамического расширения набора прикладного программного обеспечения, его версионирования и объединения в исполняемые последовательности. Для обеспечения данных функций КИАМ Digital Tool предлагает использовать сущности “Приложение” и “Сценарий”, динамически модифицируемые в базе данных системы и входящие в область ответственности пользователей с ролью “Разработчик”.

Встраивание в веб-лабораторию прикладного программного обеспечения, используемого на удаленных вычислителях, предполагает обращение к специальной странице — форме создания приложения, изображенной на рис. 2. Поля, подлежащие заполнению, включают в себя:

- имя — отображаемое в графическом интерфейсе название приложения;
- идентификатор — идентификатор приложения, состоящий из латинских букв и цифр, посредством которого происходит обращение к данной записи в сценариях (обсуждаемых далее);
- репозиторий — имя репозитория для хранения исходных кодов;
- разработчики — список пользователей системы, имеющих доступ к возможностям обновления исходного кода и внесению изменений в запись базы данных;
- тип инициализации репозитория — тип инициализации, включающий возможности: создать пустой репозиторий, инициализировать посредством отправления файла-архива и клонировать существующий.

После заполнения и принятия предложенной формы сформированное пользователем описание отправляется на сервер цифровой платформы через веб-сокеты, выполняется сохранение полученной информации в базе данных, а также генерируется директория, представляющая собой репозиторий системы версионирования Git [39] для хранения исходного кода. При успешном создании нового приложения пользователь системы получает доступ к ее странице, пример которой представлен на рис. 3. Последующее взаимодействие с исходным кодом, в том числе его обновление, возможно осуществлять с помощью стандартных средств Git посредством HTTP-запросов к серверу цифровой платформы, имеющему для данной цели выделенный маршрут. Использование системы версионирования позволяет отслеживать внесенные разработчиком в исходный код изменения, создавать ветвления и соблюдать целостность хранимых файлов для последующей отправки на удаленный вычислительный ресурс.

Для корректного встраивания в систему проблемно-ориентированного приложения корень репозитория должен содержать файл мета-описания под названием `passport.yaml` в формате YAML [40]. Данное мета-описание содержит команды сборки и запуска, дерево входных параметров и список выходных данных, с назначением типа и компонента интерфейса для их отображения. Использование TypeScript и вспомогательных библиотек `class-transformer` [41] и `class-validator` [42] позволило построить эффективный, расширяемый и контролируемый процесс вали-

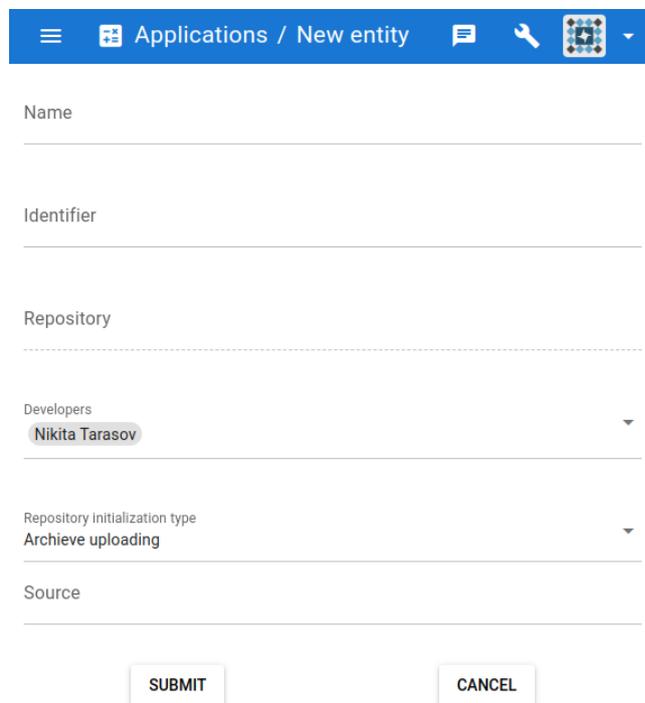


Рис. 2. Форма создания приложения

Fig. 2. Application creation form

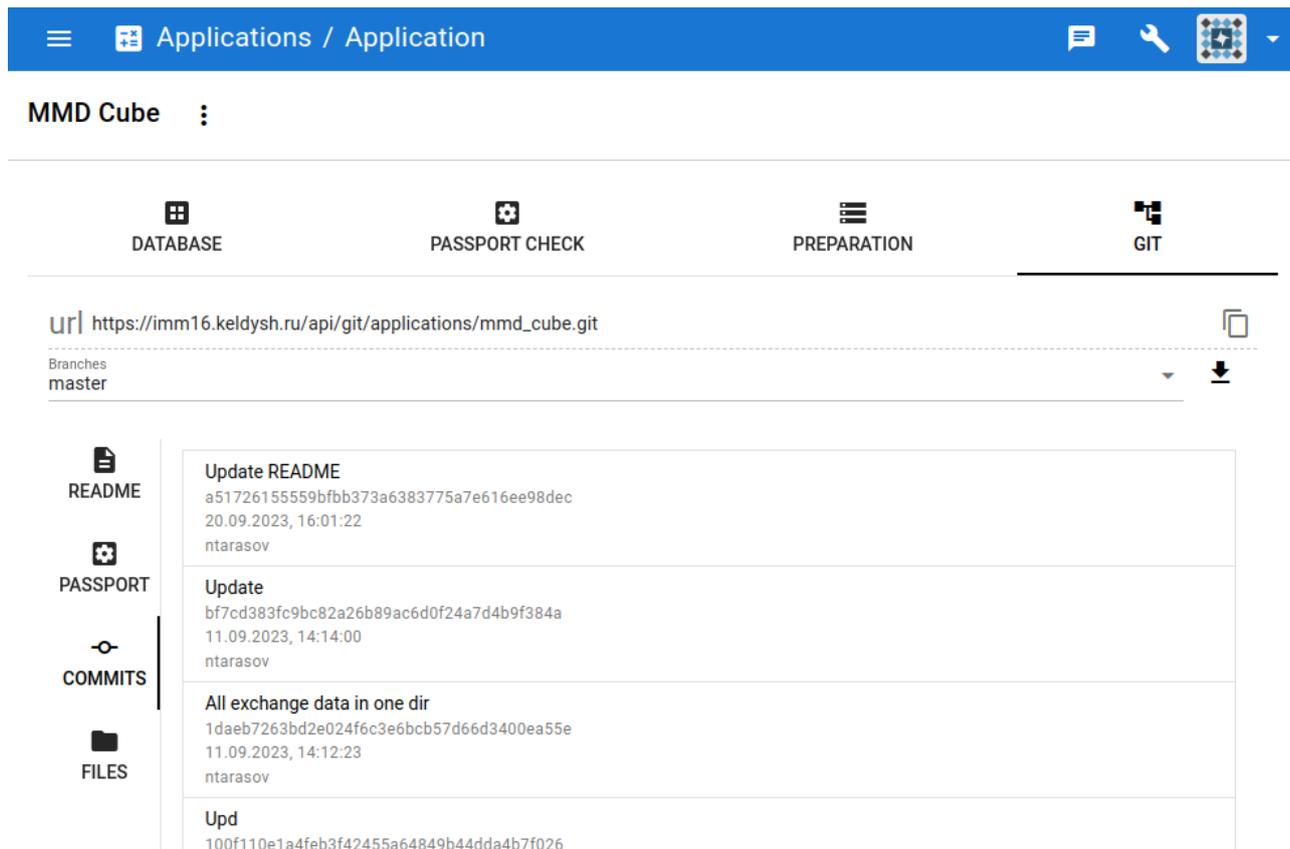


Рис. 3. Страница приложения

Fig. 3. Application page

дании содержимого passport.yaml. Опциональным является файл README.md формата Markdown [43], содержащий текстовое описание приложения с упрощенной разметкой, отображаемое в графическом интерфейсе в качестве справочной информации.

Как было отмечено ранее, существует необходимость в наличии механизма группировки приложений, поддерживающего возможность объединения выходных и входных данных, для получения цепочки исполнения. В качестве примера возможно рассмотреть процесс генерации расчетной сетки, предполагающий последовательное выполнение действий (подготовка и обработка геометрии исследуемой области, генерация дискретизации полученного объекта, разбиение сетки на подобласти для параллельного расчета и конвертация в формат приложения для моделирования) с использованием данных, полученных на предыдущем шаге. Для поддержки данного поведения KIAM Digital Tool предлагает к использованию сущности типа “Сценарий”.

Процедура добавления нового сценария во многом соответствует таковому для приложений, включая создание репозитория для хранения файлов. Однако в качестве мета-описания используется файл script.yaml, на содержании которого необходимо остановиться подробнее. Объект, получаемый при анализе script.yaml, должен включать используемые приложения, этапы исполнения и связи данных, а также иметь опциональную структуру общих входных данных. Используемые приложения описываются словарем, содержащим в качестве ключа некоторый псевдоним, а в качестве значения — идентификатор приложения, зарегистрированного в системе. В данном случае псевдонимы выступают аналогом экземпляра приложения, используемым для обращения к нему в рамках сценария. Применение подобного метода позволяет использовать одно приложение в разных режимах работы, а также осуществлять раздельное задание параметров и последовательный запуск. Список этапов исполнения состоит из строк, подобных стандартному вызову функций в языке программирования C, в качестве имени ожидающий предусмотренное действие (на текущий момент актуально ключевое слово “run”, обеспечивающее запуск с переходом к следующему этапу по завершению выполнения текущего), а в качестве первого параметра — псевдоним

запускаемого приложения. Связывание используемых в сценарии программ осуществляется с помощью соответствующего словаря, в качестве ключей которого задаются источники данных, а значения представляют собой цель либо список целей связывания.

Таким образом, в рамках представляемого подхода организована возможность встраивания широкого набора проблемно-ориентированных приложений в веб-лабораторию, не требующая внесения изменений в исходный код и ограничивающаяся подготовкой мета-описаний. При этом все подготовленные разработчиками программы доступны другим пользователям системы для получения информационной справки на соответствующих страницах, а также для проведения вычислительных экспериментов.

3.3. Интеграция удаленных суперкомпьютеров. Одной из основных целей предлагаемого решения является обеспечение унификации взаимодействия с широким набором удаленных вычислительных ресурсов. Для обеспечения данной возможности пользователь с ролью “Администратор” имеет право доступа к странице регистрации нового вычислителя, изображенной на рис. 4, содержащей форму, предлагающую следующие поля к заполнению:

- имя — название вычислительного ресурса, отображаемое в графическом интерфейсе;
- хост — имя хоста или IP-адрес для доступа к удаленному ресурсу;
- порт — порт для взаимодействия цифровой платформы с удаленным вычислителем;
- тип доступа — протокол доступа к удаленному вычислителю, на текущий момент реализована поддержка SSH;
- тип очереди — предустановленная на удаленном вычислителе система прохождения пользовательских заданий, в настоящее время поддерживается стандартный запуск Unix и СУППЗ;
- переменные окружения — список элементов ключ-значение, устанавливаемый перед выполнением каждой из удаленных команд.

Отметим, что при разработке платформы предусматривается возможность расширения поддерживаемых протоколов взаимодействия с удаленным вычислительным ресурсом и систем прохождения пользовательских заданий. Кроме того, для обеспечения универсальности, было принято решение избежать необходимости установки приложений-агентов на суперкомпьютерах для выделенного взаимодействия с веб-лабораторией, обеспечив требуемые возможности формированием запросов на стороне сервера.

После добавления нового вычислительного ресурса он становится доступен для дальнейшей персональной настройки под каждого пользователя цифровой платформы. В случае взаимодействия по SSH-протоколу пользователь должен добавить публичный `id-rsa` ключ, подготовленный системой при регистрации, в список доверенных на удаленном компьютере. Кроме того, посредством специальной веб-формы необходимо предоставить имя пользователя и путь к рабочей директории вычислительного ресурса. В результате выполнения предложенных действий пользователь получает возможность подготовки приложений на удаленном компьютере. Для этого необходимо обратиться либо к странице приложения, активировав вкладку подготовки (тогда графический интерфейс предложит осуществить выбор ветки приложения для каждого из доступных пользователю вычислительных ресурсов), либо к странице вычислительного ресурса с подготовкой всех доступных приложений (см. рис. 5). После осуществления выбора и его подтверждения система осуществит отправку ис-

Рис. 4. Форма регистрации вычислителя

Fig. 4. Computer registration form

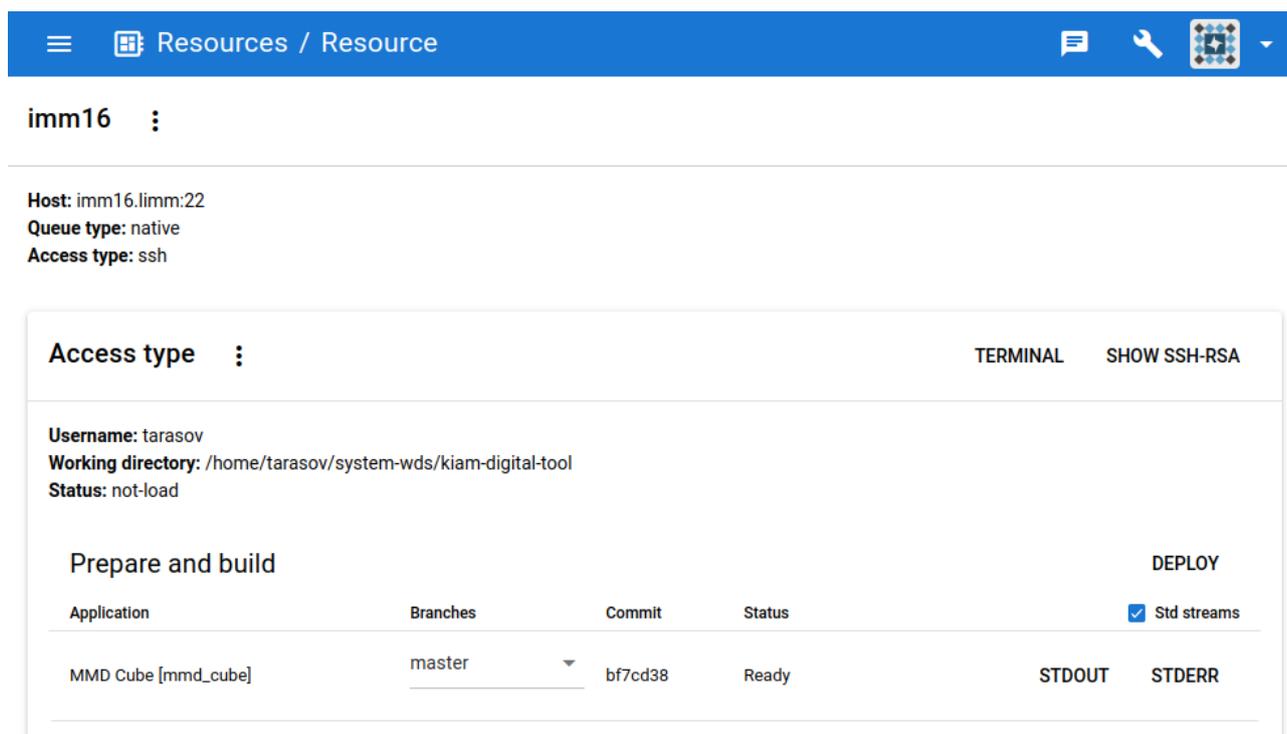


Рис. 5. Страница вычислительного ресурса

Fig. 5. Compute resource page

ходного кода на целевой компьютер из репозитория (с закреплением текущей версии ветки в базе данных) в установленную при настройке рабочую директорию. Также, в соответствии с паспортом приложения, система вызовет команду сборки, в результате которой будет подготовлен исполняемый файл.

Предложенный подход интеграции удаленных вычислительных ресурсов позволяет избежать установки стороннего программного обеспечения на целевой платформе, а также ограничиться пользовательскими правами доступа для запуска зарегистрированных в системе приложений прикладного уровня. В итоге выполнения описанных действий пользователи веб-лаборатории получают необходимый для проведения вычислительных экспериментов набор программного обеспечения в собственных директориях, при этом информация о версиях подготовленных приложений будет доступна цифровой платформе.

3.4. Проведение вычислительных экспериментов. В рамках цифровой платформы вычислительные эксперименты представлены сущностями типа “Расчет”, группируемыми посредством сущностей типа “Проект”. При создании проекта (действие, доступное пользователю с ролью “Расчетчик”) веб-форма требует задания названия, текстового описания и списка участников. Каждый из участников имеет возможность создания расчета, связанного с текущим проектом. При обращении к форме создания нового расчета требуется задать его название, текстовое описание, а также выбрать сценарий и его версию. На основании мета-описания сценария пользователю будет предложено выбрать вычислительный ресурс для каждого из используемых приложений в соответствии с псевдонимами (рис. 6). После подтверждения выбора платформа подготовит рабочие и выходные директории используемых приложений, а также символические ссылки на исполняемые файлы. Благодаря выбору вычислительных ресурсов появляется возможность определить используемую версию приложения, а также актуальный паспорт. На основании мета-описаний, используемых в сценарии приложений, платформой будут сформированы вкладки задания исходных данных (рис. 7), запуска этапов исполнения (рис. 8) и доступа к выходным данным (рис. 9).

Графический интерфейс, подготовленный на основании описания входных параметров паспорта приложения, подобный рис. 7, поддерживает категоризацию и возможность формирования конфигурационного файла, представляющих возможный тип корневого элемента дерева входных параметров. Его листьями являются поля (которые могут быть текстовыми или числовыми, флагом, выпадающим списком) и файлы данных. Последние задаются либо непосредственной отправкой, либо связыванием с выходны-

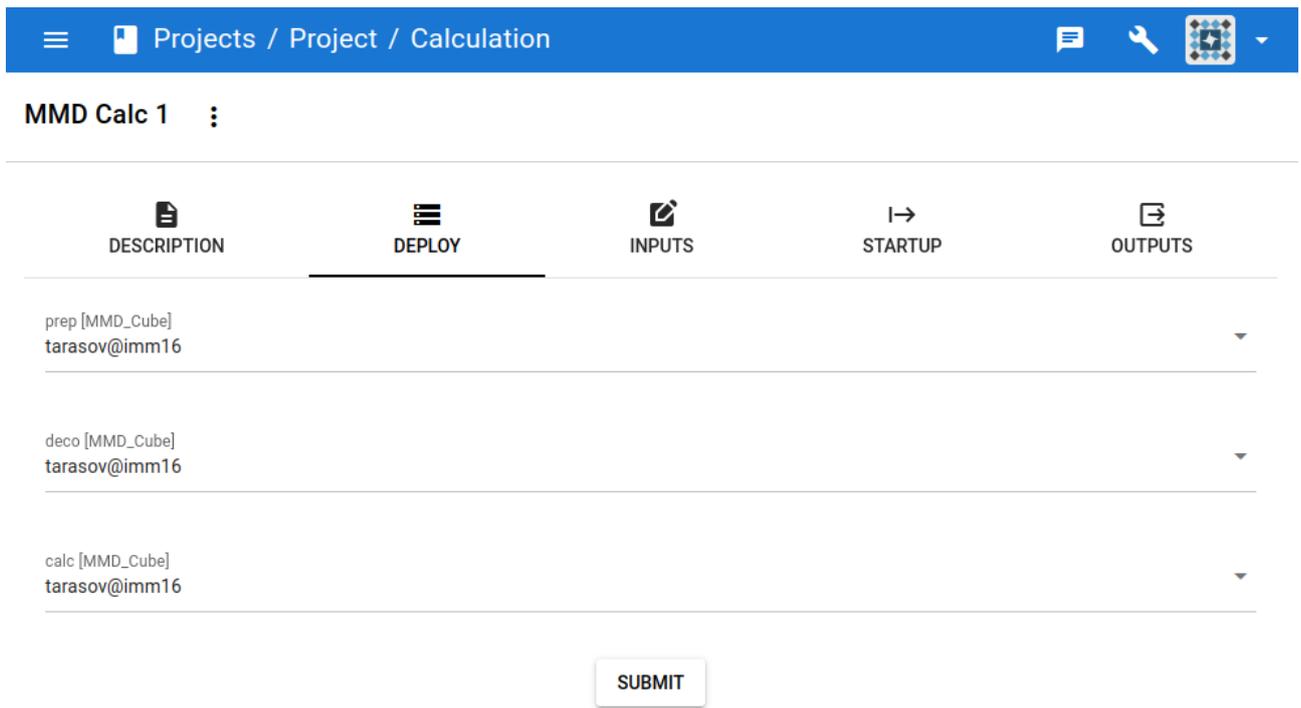


Рис. 6. Страница расчета. Подготовка приложений
Fig. 6. Calculation page. Application preparation

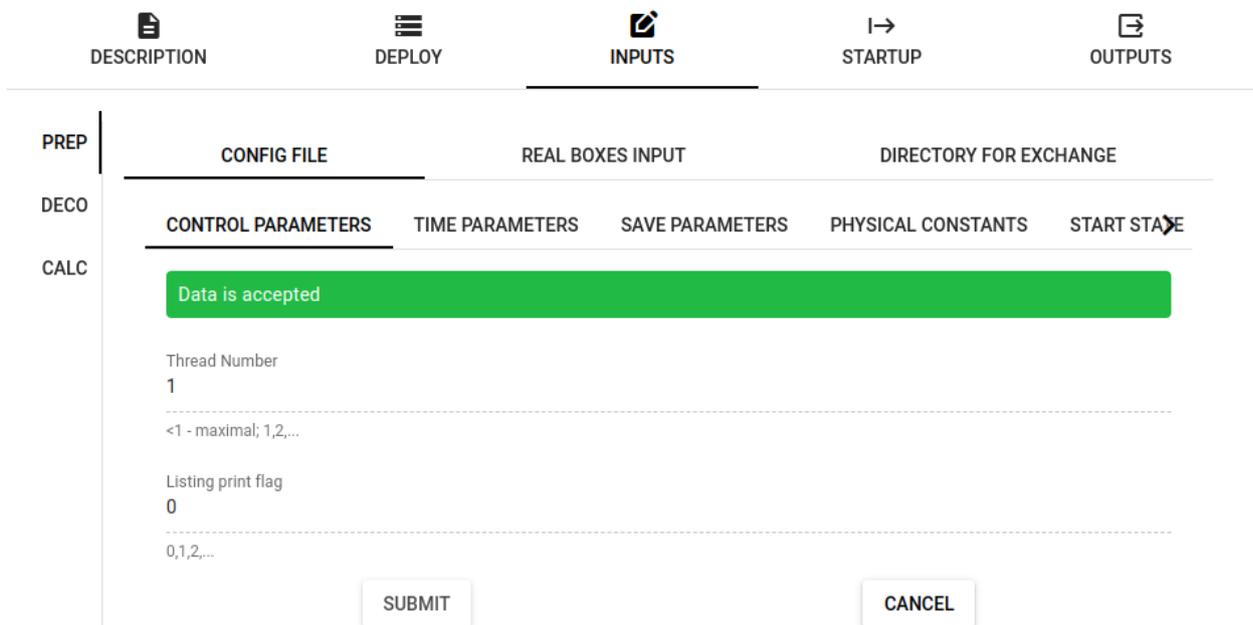


Рис. 7. Фрагмент страницы расчета. Входные параметры
Fig. 7. Calculation page fragment. Inputs

ми данными, полученными в других расчетах. Родительскими элементами полей должны быть формы, добавляющие отображение статуса и кнопки сохранения и отмены. Родителями форм могут выступать категории и конфигурационные файлы. При сохранении данных платформа формирует локальные файлы для каждой из форм, используемые в последующем при запуске или повторном обращении к странице расчета.



DESCRIPTION
DEPLOY
INPUTS
STARTUP
OUTPUTS

Aliases list

Application alias	Run parameters				<input checked="" type="checkbox"/> Std streams	
prep	-np 1	-ppn	-maxtime 10	-termtime	STDOUT	STDERR
deco	-np 1	-ppn	-maxtime 10	-termtime	STDOUT	STDERR
calc	-np 1	-ppn	-maxtime 700	-termtime	STDOUT	STDERR

Stage: 3, Status: Finished ▶ ■

Stage	Status	
run(prepare)	1106756: Finished	▶ ■
run(deco)	2136726: Finished	▶ ■
run(calc)	1107263: Finished	▶ ■

Рис. 8. Фрагмент страницы расчета. Запуск заданий

Fig. 8. Calculation page fragment. Tasks startup

DESCRIPTION
DEPLOY
INPUTS
STARTUP
OUTPUTS

- ▼ calc
 - Start Real Boxes
 - ▶ Directory for exchange
 - ▶ Dat directory
 - ▶ Plt directory
 - ▶ Txt directory
 - ▶ Backup directory
- ▶ deco
- ▶ prep

Type: Directory

Description: Backup directory

Datatype: Binary

Access time: 11.09.2023, 14:44:59

Modification time: 11.09.2023, 17:39:47

Size: 20.48 KB

OPEN
DOWNLOAD
DELETE

Рис. 9. Фрагмент страницы расчета. Выходные данные

Fig. 9. Calculation page fragment. Outputs

После задания исходных параметров предполагается выполнение этапов исполнения, представленных в мета-описании сценария, посредством элементов управления, пример которых представлен на рис. 8. Данный интерфейс предоставляет возможности запуска и остановки как каждого этапа в отдельности, так и сценария в целом, а также информацию о текущем статусе пользовательских задач. Для псевдонимов приложений возможно получить содержание потоков стандартного вывода и ошибки. Также, на основании описания запуска, указанного в паспорте приложения, и типа системы прохождения пользовательских заданий выбранного ресурса, интерфейс дополнится полями основных управляющих параметров очереди, позволяющих задать (на примере СУППЗ): количество параллельных процессов, количество процессов на узел, время расчета и др. В итоге пользователь, изменяя параметры запуска, может проводить вычисления либо на одном узле выбранного им кластера, либо на нескольких. Сами узлы подбирает СУППЗ. Это не является критичным, поскольку чаще всего используются вычислительные кластеры с одинаковой архитектурой узлов. Однако при необходимости возможна и более тонкая настройка системы запуска. Например, на используемом нами гибридном суперкомпьютере К60 имеется секция с графическими ускорителями K60GPU. Администратор платформы может добавить данную секцию в качестве независимого вычислительного ресурса. При ее использовании пользователь может выбрать три режима вычислений: расчеты на CPU, расчеты на GPU, расчеты в совместном режиме. Для реализации такой возможности ему необходимо иметь соответствующий расчетный код. При этом выбор режима вычислений будет определен идентификатором этого расчетного кода или версией программы, например, GIMM_APP_MD_CPU, GIMM_APP_MD_GPU, GIMM_APP_MD_HYBRID. Далее, при запросе запуска, цифровая платформа подготавливает входные параметры для используемого в рамках этапа приложения и отправляет их на удаленный ресурс, если входным параметром является файл данных или конфигурационный файл. После чего, в соответствии с установленной системой прохождения пользовательских заданий, цифровая платформа формирует команду запуска, сохраняя идентификатор и статус задания в базе данных. Дальнейшее обновление информации о задании происходит по таймеру в соответствии с принципом масштабирования цифровой платформы.

Обращение к списку выходных данных с помощью интерфейса, подобного изображенному на рис. 9, возможно как в процессе выполнения удаленных вычислений, так и после их завершения. Цифровая платформа предоставляет подробную информацию о данных (входных, выходных, информационных) проблемно-ориентированных приложений, а также обеспечивает возможности их удаления и загрузки. Если в качестве выходных данных программа генерирует директорию, веб-лаборатория предусматривает возможность доступа к ее содержимому. Также предусмотрена возможность открыть файлы непосредственно в веб-интерфейсе с помощью компонентов, соответствующих типу данных. На текущий момент реализованы компоненты отображения текстовых файлов, изображений и видеозаписей.

Итак, посредством представленного графического веб-интерфейса достигается процесс единообразной процедуры проведения вычислительного эксперимента, включающий подготовку исходных данных, унифицированное взаимодействие с вычислительными ресурсами, а также доступ к выходным данным для их последующей обработки и анализа.

4. Тестирование цифровой платформы. Основной задачей, решаемой посредством цифровой платформы KIAM Digital Tool, является обеспечение проведения полного цикла вычислительных экспериментов, посвященных многомасштабному моделированию задач сверхзвукового газодинамического напыления. Примененный при построении веб-лаборатории подход достигает большей гибкости благодаря свободному встраиванию проблемно-ориентированных приложений и формированию функциональных блоков посредством сценариев исполнения. Однако именно модели напыления позволяют протестировать разработанное в рамках исследования программное обеспечение, выбранные архитектурные решения и набор технологий. Ключевым “узким” местом, в данном случае, становятся обмены данными между выполняемыми на удаленных вычислителях пакетными приложениями, ведь при решении практически важных задач объемы файлов, которыми оперируют прикладные программы, могут достигать нескольких терабайтов.

В рамках настоящего исследования тестирование веб-лаборатории производилось посредством использования расчетных компонент комплекса GIMM_NANO [44], обеспечивающих как многомасштабное, так и прямое молекулярное моделирование процесса напыления нанокластеров на подложку. Данный программный комплекс разрабатывается с 2012 года коллективом авторов (в том числе авторами данной работы) Института прикладной математики имени М. В. Келдыша РАН и постоянно обновляется путем пополнения коллекции расчетных кодов. Комплекс поддерживает параллельность как на уровне потоков



(с помощью OpenMP), так и на уровне процессов (посредством OpenMPI). Среди использованных в данной работе расчетных компонент комплекса GIMM_NANO отметим следующие основные программные модули:

- GIMM_APP_QGD_CPU — расчет газодинамических течений на основе квазигазодинамической системы уравнений [45];
- GIMM_APP_QGD_MD_CPU — совместный расчет процессов напыления [1, 46] с помощью моделей квазигазодинамики и молекулярной динамики [47];
- GIMM_APP_MD_CPU_Gas_Metal — молекулярно-динамический расчет взаимодействий газов с металлическими поверхностями [48].

Отметим, далее, что первый из вышеуказанных программных модулей является по сути классическим сеточным солвером, предназначенным для расчета макроскопических параметров газовой среды. Подготовка к его работе связана с формированием геометрии, генерацией сетки и заданием исходных данных на этой сетке, декомпозицией сетки для реализации параллельных вычислений. Во время и после проведения расчетов требуются постобработка и визуализация рассчитанных распределенных данных. Вся эта процедура поддерживается стандартным сценарием для решения динамических задач вычислительной гидродинамики (например, как в пакетах Ansys Fluent [49] или Comsol Multiphysics [50]) разработанной веб-среды.

Второй программный модуль отличает более тонкая настройка граничных условий. В веб-среде для этого приходится использовать модифицированный сценарий задания исходных данных в граничных ячейках сетки. Также первый и второй модули требуют во время расчета доступа к базе данных молекулярных расчетов (БДМР) в части формирования и использования макроскопических параметров уравнений состояния газовой среды, кинетических коэффициентов, характеризующих такие свойства материалов, как диффузия, вязкость, теплопроводность, а также термоупругость. Этот доступ генерируется при сборке вышеуказанных модулей.

Более сложным оказалась интеграция в веб-среду третьего модуля, который реализует самые разнообразные возможности молекулярного моделирования. Во-первых, такое моделирование является многоэтапным, а, во-вторых, поддерживает следующие функции: а) расчет свойств как чистых, так и композитных материалов методами молекулярной динамики; б) наполнение базы данных молекулярных расчетов; в) конструирование новых сложных многокомпонентных систем частиц на основе проведенных вычислений; г) решение сопряженных задач микрофлюидики и микромеханики.

Подробнее остановимся на работе третьего модуля при прямом молекулярном моделировании задачи напыления. Оно, как указано выше, является многоэтапным и наиболее вычислительно емким. Для его реализации приходится проводить целый ряд отдельных расчетов по подготовке газовой среды, напыляемых частиц и подложки. Опишем здесь лишь заключительную фазу расчета, связанную со взаимодействием нанокластера, доставленного газовым потоком, с поверхностью подложки (расчеты по подготовке и доставке нанокластера обсуждались в работах [51, 52] на примере азота и никеля).

Пользовательский интерфейс, приведенный на рис. 3, 6, 7, 8 и 9, представляет результат встраивания в платформу приложения GIMM_APP_MD_CPU_Gas_Metal. Процесс исполнения рассматриваемого кода предполагает несколько этапов запуска: предварительную генерацию на основе БДМР систем частиц (кластера и подложки), разбиение системы частиц на расчетные домены, проведение расчетов взаимодействия. Данные этапы производятся выбранным приложением, запускаемым в различных режимах, задаваемых с помощью соответствующих ключей конфигурационного файла. Предложенный процесс исполнения позволяет сформировать сценарий, использующий три экземпляра одного приложения, запускаемых последовательно. При этом каждый из этапов зависит от данных, подготовленных на предыдущем шаге и требующих их передачи по потоку исполнения. И конечно, при запуске расчетов на различных вычислительных ресурсах, отправки больших данных избежать не удастся. Однако, в случае использования одного кластера или суперкомпьютера, веб-лаборатория воспользуется возможностью создания символической ссылки, что позволяет, с одной стороны, сократить время подготовки приложения к запуску, с другой — уменьшить количество используемой памяти. Результатом выполнения полного цикла вычислительного эксперимента являются директории, содержащие файлы, описывающие состояния общей системы частиц (нанокластера и подложки), а также усредненные на декартовой сетке макропараметры (плотность, температура, давление) этой системы.

Последующая обработка полученных результатов также является “узким” местом. Стандартный процесс визуализации данных с использованием локальных решений предполагает наличие на пользовательском устройстве анализируемых файлов, что приводит к необходимости скачивания полного массива информации с удаленного вычислительного ресурса, а также существенно увеличивает системные требования, накладываемые на клиентский компьютер. Поскольку веб-лаборатория поддерживает встраивание сторонних приложений, можно избежать использования “толстого” клиента путем разработки пакетного приложения обработки и визуализации. Однако данный подход не является интерактивным и требует окончания выполнения расчета перед анализом результатов. Для преодоления представленных недостатков предлагается использование KitWare ParaViewWeb Visualizer [53], являющегося клиент-серверным приложением интерактивной веб-визуализации, пример использования которого представлен на рис. 10.

Основной особенностью данного решения в плане визуализации является серверная обработка файлов данных и генерация растровых изображений, передаваемых посредством веб-сокета в графический интерфейс клиентской части, позволяющий осуществлять интерактивное управление камерой, а также запрашивать у сервера дополнительный анализ области исследования, в том числе построение сечения, разреза, фильтрацию по значению и других инструментов визуального исследования. Являясь частью дистрибутива ParaView [54], сервер поддерживает большое количество форматов научных данных непосредственно, а также широкий набор методов их обработки. Также данное программное обеспечение позволяет группировать наборы данных как временные серии с возможностью запуска анимации.

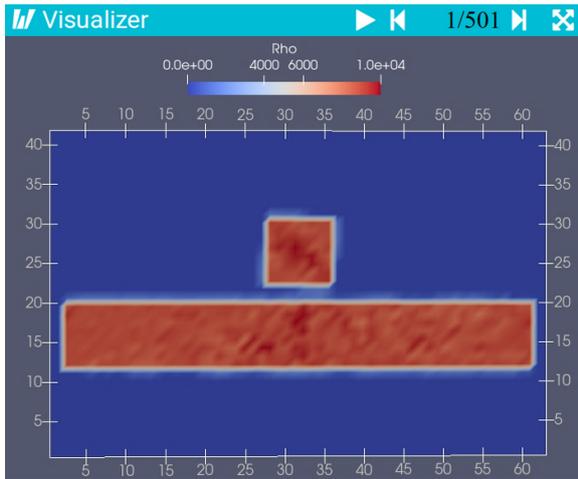
В рамках цифровой платформы предлагается следующая процедура использования Visualizer. При получении соответствующего запроса от авторизованного пользователя системы осуществляются:

- 1) запуск серверной части Visualizer на удаленном вычислительном ресурсе с явным указанием директории выходных данных и ограничением на прослушиваемые адреса (127.0.0.1);
- 2) запуск SSH-туннеля от вычислительного ресурса к цифровой платформе;
- 3) доступ к интерфейсу Visualizer через сервер веб-лаборатории, выступающий в данном случае в качестве прокси;
- 4) завершение работы серверной части Visualizer и закрытие SSH-туннеля при получении соответствующего сигнала завершения пользовательского взаимодействия.

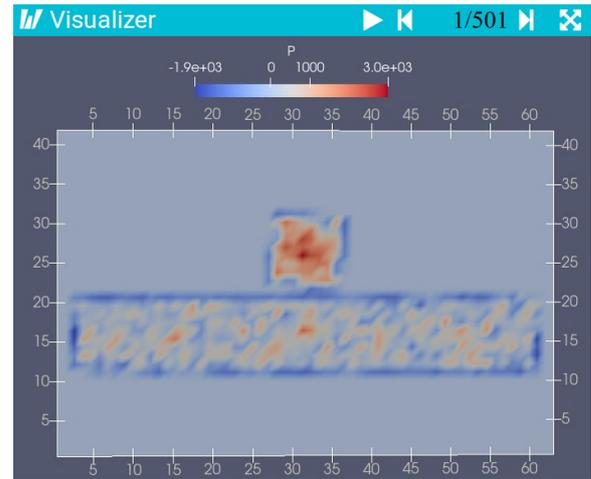
Предложенный процесс позволяет обеспечить контролируемый, авторизованный доступ к данным и соблюсти требования безопасности удаленного вычислительного ресурса (при наличии ограничений на доступные пользователям порты).

Итак, для тестирования цифровой платформы была проведена серия вычислительных экспериментов с заданием различных значений входных параметров, иллюстрирующих процесс сверхзвукового холодного газодинамического напыления. В частности, процесс рассчитывался для трех значений скорости набегания кластера на пластину. Также рассматривались кластеры кубической и сферической формы. Приведем в качестве примера задачу напыления никелевого нанокластера кубической формы, налетающего со скоростью 1000 нм/нс на тонкую никелевую подложку. Моделирование проводилось в полной трехмерной постановке. Расчет проводился на одном расчетном узле с использованием 28 потоков. На рис. 10 показаны центральные сечения системы кластер–подложка на финальной стадии процесса напыления. При выбранных параметрах кластер был успешно внедрен в материал подложки без повреждения последней. Результаты расчета соответствуют теоретическим представлениям о процессе, а именно: в расчете точно выполняются законы сохранения массы, импульса и энергии системы кластер–пластина на каждом шаге по времени, в том числе до и после столкновения.

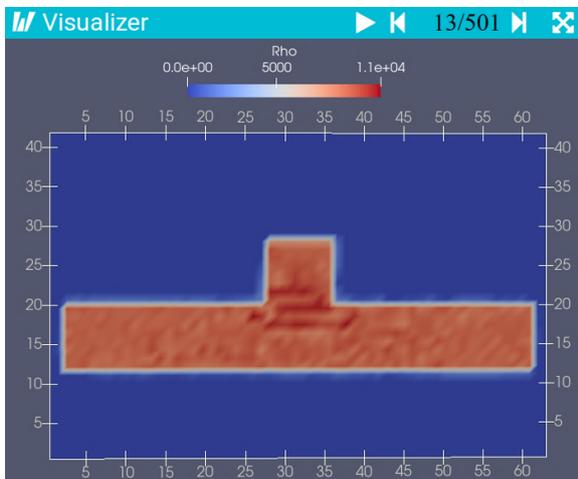
Завершая данный раздел, можно сделать вывод: разработанная цифровая платформа позволяет выполнить весь цикл вычислительных экспериментов в выбранной предметной области. Тем самым подтверждается общая состоятельность принятых архитектурных решений и робастность их программной реализации. Отметим, что хотя основными задачами, решаемыми разработанной в рамках исследования веб-лаборатории, являются задачи суперкомпьютерного математического моделирования процесса напыления, цифровая платформа, вследствие использованного подхода встраивания расчетных компонент, допускает интеграцию приложений для решения задач из иных предметных областей. Генерация веб-интерфейса для проведения вычислительного эксперимента при этом происходит динамически, на основании текстовых файлов — паспортов приложений и сценариев исполнения. Это позволяет получить предметно-ориентированный вариант веб-лаборатории без изменения ее управляющей части. Основными задачами, решаемыми платформой в данном случае, являются: обеспечение унифицированного до-



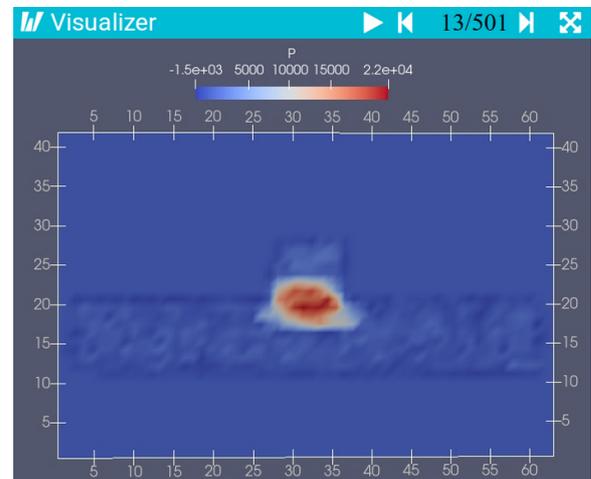
a)



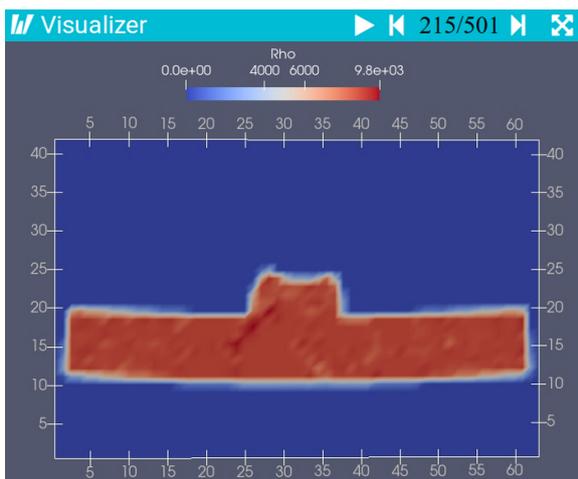
b)



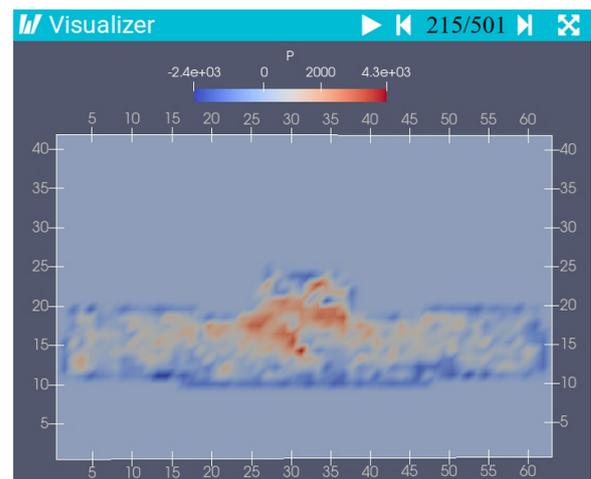
c)



d)



e)



f)

Рис. 10. Веб-визуализация: начальное распределение а) плотности и б) давления; распределение с) плотности и д) давления при начале взаимодействия; распределение е) плотности и ф) давления при внедрении кластера

Fig. 10. Web visualization: initial distributions of a) density and b) pressure; distribution of c) density and d) pressure at the beginning of interaction; distribution of e) density and f) pressure at the cluster implantation

ступа к удаленным вычислительным ресурсам; подготовка и управление пользовательскими заданиями; обеспечение каталогизации, совместного проведения вычислительных экспериментов; анализ полученных результатов. При адаптации веб-лаборатории к другой предметной области необходимо добавить новые расчетные коды и интерфейсы их взаимодействия с управляющей частью (паспорта и сценарии). Также, возможно, понадобится реализация поддержки новых форматов данных для последующего их анализа и визуализации в веб-среде, что может быть достигнуто путем погружения программ конвертации.

5. Заключение. В работе рассмотрены существующие решения и конкретные реализации различных предметно-ориентированных цифровых платформ. При их анализе были сформулированы требования и задачи к авторской разработке подобной платформы. Также было сформировано описание принципов построения функционального ядра и архитектуры веб-платформы KIAM Digital Tool. Реализация этих принципов потребовала модернизации текущей версии платформы. В итоге она поддерживает динамическое встраивание проблемно-ориентированных приложений и их группировку в функциональные модули, регистрацию удаленных вычислительных ресурсов (кластеров и суперкомпьютеров), а также проведение сложных длительных и трудоемких вычислительных экспериментов. Последние включают подготовку исходных данных, запуск и мониторинг группы прикладных программ, осуществляют возможность доступа к результатам моделирования. Цифровая платформа была протестирована на модельных задачах сверхзвукового газодинамического напыления путем проведения серии вычислительных экспериментов и подтвердила свою состоятельность.

Дальнейшее развитие предложенной цифровой платформы будет проводиться по нескольким направлениям. Первое направление связано с расширением возможностей по заданию исходных данных, в том числе компонент задания расчетной области и ее разметки, генерации и редактирования больших распределенных молекулярных структур. Второе направление связано с совершенствованием методик интерактивного анализа результатов расчета в рамках единого графического интерфейса. И, наконец, не менее важным направлением является расширение возможностей разработчика, вплоть до реализации аналога интегрированной среды разработки с использованием для тестирования и отладки удаленных вычислительных ресурсов, что позволит существенно ускорить предоставление обновленных версий приложений конечному пользователю.

Список литературы

1. Podryga V.O., Polyakov S.V., Tarasov N.I., Usachev V.A. Mathematical modeling the processes of supersonic cold gas dynamic spraying of nanoparticles on substrates // Lobachevskii J. Math. 2023. 44, № 8. 1918–1928. doi 10.1134/S1995080223050487.
2. Ansys Workbench. Simulation integration platform. <https://www.ansys.com/products/ansys-workbench>. (Дата обращения: 6.12.2023).
3. Аксенов А.А. FlowVision: индустриальная вычислительная гидродинамика // Компьютерные исследования и моделирование. 2017. 9, № 1. 5–20. doi 10.20537/2076-7633-2017-9-5-20.
4. Подрыга В.О., Поляков С.В., Тарасов Н.И. Сервер цифровой платформы для суперкомпьютерного моделирования процессов напыления наночастиц на подложки KIAM_DIGITAL_TOOL_SERVER, версия 1. Свидетельство о государственной регистрации программ для ЭВМ 2022666958, 12.09.2022.
5. Подрыга В.О., Поляков С.В., Тарасов Н.И. Веб-клиент цифровой платформы для суперкомпьютерного моделирования процессов напыления наночастиц на подложки KIAM_DIGITAL_TOOL_CLIENT, версия 1. Свидетельство о государственной регистрации программ для ЭВМ 2022666957, 12.09.2022.
6. Тарасов Н.И., Подрыга В.О., Поляков С.В., Тимаков А.В. Цифровая платформа для суперкомпьютерного математического моделирования процессов напыления // Электронные библиотеки. 2022. 25, № 6. 697–721. doi 10.26907/1562-5419-2022-25-6-697-721.
7. TypeScript: JavaScript with Syntax for Types. <https://www.typescriptlang.org/>. (Дата обращения: 6.12.2023).
8. Polyakov S.P., Demichev A.P., Kryukov A.P. Web toolkit for scientific research: state of the art and the prospect for development // Procedia Computer Science. 2015. 66. 429–438. doi 10.1016/j.procs.2015.11.049.
9. Kryukov A.P., Demichev A.P., Polyakov S.P. Web platforms for scientific research // Programming and Computer Software. 2016. 42, № 3. 129–141. doi 10.1134/S036176881603004X.
10. Prove.Design. <https://prove.design>. (Дата обращения: 6.12.2023).
11. Magill J., Dreher R., Soti Z., Lasché G.P. Nucleonica: web-based software tools for simulation and analysis // Proc. 21st International Conference on Nuclear Energy for New Europe. Ljubljana, Slovenia. September 5–7, 2012. https://arhiv.djs.si/proc/nene2012/Publication_datoteke/Proceedings/1303.pdf. (Дата обращения: 6.12.2023).



12. pSeven Enterprise. <https://www.pseven.io/product/pseven-enterprise/>. (Дата обращения: 7.12.2023).
13. Sukhoroslov O., Volkov S., Afanasiev A. A web-based platform for publication and distributed execution of computing applications // Proc. 14th International Symposium on Parallel and Distributed Computing (ISPDC), Limassol, Cyprus, June 29–July 2, 2015. doi 10.1109/ISPDC.2015.27.
14. Everest. <https://everest.distcomp.org/apps/list>. (Дата обращения: 7.12.2023).
15. McLennan M., Kennell R. HUBzero: a platform for dissemination and collaboration in computational science and engineering // Comput. Sci. Eng. 2010. 12, N 2. 48–53. doi 10.1109/MCSE.2010.41.
16. Madhavan K., Zentner L., Farnsworth V., et al. nanoHUB.org: cloud-based services for nanoscale modeling, simulation, and education // Nanotechnology Reviews. 2013. 2, N 1. 107–117. doi 10.1515/ntrev-2012-0043.
17. Steiger S., Povolotskyi M., Park H.-H., et al. NEMO5: a parallel multiscale nanoelectronics modeling tool // IEEE Transactions on Nanotechnology. 2011. 10, N 6. 1464–1474. doi 10.1109/TNANO.2011.2166164.
18. Alber A., Nabrzyski J., Wright T. The HUBzero platform: extensions and impressions // Proc. 3rd International Workshop on Science Gateways for Life Sciences (IWSG 2011). London, United Kingdom, June 8–10, 2011. <https://ceur-ws.org/Vol-819/paper2.pdf>. (Дата обращения: 7.12.2023).
19. Mejia D., Kubis T., Klimeck G. NemoViz: a visual interactive system for atomistic simulations design // Visualization in Engineering. 2018. 6. Article Number 6. doi 10.1186/s40327-018-0067-4.
20. Granger B.E., Pérez F. Jupyter: thinking and storytelling with code and data // Comput. Sci. Eng. 2021. 23, N 2. 7–14. doi 10.1109/MCSE.2021.3059263.
21. Pérez F., Granger B.E. IPython: a system for interactive scientific computing // Comput. Sci. Eng. 2007. 9, N 3. 21–29. doi 10.1109/MCSE.2007.53.
22. GitHub — voila-dashboards/voila: Voilà turns Jupyter notebooks into standalone web applications. <https://github.com/voila-dashboards/voila>. (Дата обращения: 7.12.2023).
23. Thomas R., Cholia S. Interactive supercomputing with Jupyter // Comput. Sci. Eng. 2021. 23, N 2. 93–98. doi 10.1109/MCSE.2021.3059037.
24. Самоваров О.И., Гайсарян С.С. Архитектура и особенности реализации платформы UniHUB в модели облачных вычислений на базе открытого пакета OpenStack // Труды Института системного программирования РАН. 2014. 26, № 1. 403–420. doi 10.15514/ISPRAS-2014-26(1)-17.
25. Поляков С.В., Выродов А.В., Пузырьков Д.В., Якововский М.В. Облачный сервис для решения многомасштабных задач нанотехнологии на суперкомпьютерных системах // Труды института системного программирования РАН. 2015. 27, № 6. 409–420. doi 10.15514/ISPRAS-2015-27(6)-26.
26. Electron. Build cross-platform desktop apps with JavaScript, HTML, and CSS. <https://www.electronjs.org/>. (Дата обращения: 7.12.2023).
27. TOP500. The List. June 2023. <https://www.top500.org/lists/top500/2023/06/>. (Дата обращения: 7.12.2023).
28. СУППЗ. Система управления прохождением параллельных заданий. <http://suppz.jscs.ru/>. (Дата обращения: 7.12.2023).
29. ЦКП ИПМ РАН. <https://ckp.kiam.ru/?home>. (Дата обращения: 7.12.2023).
30. Node.js. <https://nodejs.org/>. (Дата обращения: 7.12.2023).
31. NestJS: A progressive Node.js framework. <https://nestjs.com/>. (Дата обращения: 7.12.2023).
32. Express: Node.js web application framework. <https://expressjs.com/>. (Дата обращения: 7.12.2023).
33. TypeORM: Amazing ORM for TypeScript and JavaScript. <https://typeorm.io/>. (Дата обращения: 7.12.2023).
34. Vue.js: The Progressive JavaScript Framework. <https://vuejs.org/>. (Дата обращения: 7.12.2023).
35. Quasar Framework. <https://quasar.dev/>. (Дата обращения: 7.12.2023).
36. Request for Comments: 6455. The WebSocket Protocol. <https://datatracker.ietf.org/doc/html/rfc6455>. (Дата обращения: 7.12.2023).
37. PM2 - Home. <https://pm2.keymetrics.io/>. (Дата обращения: 7.12.2023).
38. Redis. <https://redis.io/>. (Дата обращения: 7.12.2023).
39. Git. <https://git-scm.com/>. (Дата обращения: 7.12.2023).
40. The Official YAML Web Site. <https://yaml.org/>. (Дата обращения: 7.12.2023).
41. Decorator-based transformation, serialization, and deserialization between objects and classes. <https://github.com/typestack/class-transformer>. (Дата обращения: 7.12.2023).
42. Decorator-based property validation for classes. <https://github.com/typestack/class-validator>. (Дата обращения: 7.12.2023).

43. Markdown Guide. <https://www.markdownguide.org/>. (Дата обращения: 7.12.2023).
44. Бондаренко А.А., Кононов Э.М., Косолапов О.А., Поляков С.В., Якобовский М.В. Программный комплекс GIMM_NANO // Международная суперкомпьютерная конференция “Научный сервис в сети Интернет: все грани параллелизма”, Новороссийск, Россия, 23–28 сентября, 2013. <http://agora.guru.ru/abrau2013/pdf/333.pdf>. (Дата обращения: 7.12.2023).
45. Elizarova T.G. Quasi-gas dynamic equations. Berlin: Springer, 2009. doi 10.1007/978-3-642-00292-2.
46. Podryga V.O., Polyakov S.V. Multiscale mathematical modeling of the metal nanoparticles motion in a gas flow // Lecture Notes in Computer Science. Volume 11386. Cham: Springer, 2019. 387–394. doi 10.1007/978-3-030-11539-5_44.
47. Haile J.M. Molecular dynamics simulations. Elementary methods. New York: Wiley, 1992.
48. Подрыга В.О., Поляков С.В., Пузырьков Д.В. Суперкомпьютерное молекулярное моделирование термодинамического равновесия в микросистемах газ–металл // Вычислительные методы и программирование. 2015. 16, № 1. 123–138. doi 10.26089/NumMet.v16r113.
49. Ansys Fluent. Fluid Simulation Software. <https://www.ansys.com/products/fluids/ansys-fluent>. (Дата обращения: 7.12.2023).
50. CFD Module. Simulate Single-Phase and Multiphase Flow. <https://www.comsol.com/cfd-module>. (Дата обращения: 7.12.2023).
51. Поляков С.В., Подрыга В.О., Пузырьков Д.В., Кудряшова Т.А. Суперкомпьютерное молекулярное моделирование газодинамического напыления наночастиц на подложку // Труды международной конференции “Суперкомпьютерные дни в России”, 24–25 сентября 2018. М.: Изд-во МГУ, 2018. 782–792. <http://russianscdays.org/files/pdf18/782.pdf>. (Дата обращения: 7.12.2023).
52. Podryga V., Polyakov S. Atomistic modeling of metal nanocluster motion caused by gas flow impact // Lobachevskii Journal of Mathematics. 2019. 40, № 11. 1987–1993. doi 10.1134/S1995080219110210.
53. Visualizer. <https://kitware.github.io/visualizer/>. (Дата обращения: 7.12.2023).
54. Ayachit U. The ParaView guide: a parallel visualization application. Clifton Park: Kitware Inc., 2015.

Поступила в редакцию
1 ноября 2023 г.

Принята к публикации
4 декабря 2023 г.

Информация об авторах

Никита Игоревич Тарасов — к.ф.-м.н., научн. сотр.; Институт прикладной математики имени М. В. Келдыша РАН (ИПМ РАН), Миусская пл., д. 4, 125047, Москва, Российская Федерация.

Виктория Олеговна Подрыга — д.ф.-м.н., доцент, вед. научн. сотр.; Институт прикладной математики имени М. В. Келдыша РАН (ИПМ РАН), Миусская пл., д. 4, 125047, Москва, Российская Федерация.

Сергей Владимирович Поляков — д.ф.-м.н., старший научный сотрудник, вед. научн. сотр.; Институт прикладной математики имени М. В. Келдыша РАН (ИПМ РАН), Миусская пл., д. 4, 125047, Москва, Российская Федерация.

References

1. V. O. Podryga, S. V. Polyakov, N. I. Tarasov, and V. A. Usachev, “Mathematical Modeling the Processes of Supersonic Cold Gas Dynamic Spraying of Nanoparticles on Substrates,” Lobachevskii J. Math. 44 (8), 1918–1928 (2023). doi 10.1134/S1995080223050487.
2. Ansys Workbench. Simulation Integration Platform. <https://www.ansys.com/products/ansys-workbench>. Cited December 6, 2023.
3. A. A. Aksenov, “FlowVision: Industrial Computational Fluid Dynamics,” Comput. Res. Model. 9 (1), 5–20 (2017). doi 10.20537/2076-7633-2017-9-5-20.
4. V. O. Podryga, S. V. Polyakov, and N. I. Tarasov, *Digital Platform Server for Supercomputer Modeling of Nanoparticle Spraying Processes on Substrates KIAM_DIGITAL_TOOL_SERVER, Version 1*, Certificate of RF Registration of Computer Program No. 2022666958. Date of Registration: September 12, 2022.



5. V. O. Podryga, S. V. Polyakov, and N. I. Tarasov, *Digital Platform Web-client for Supercomputer Modeling of Nanoparticle Spraying Processes on Substrate KIAM_DIGITAL_TOOL_CLIENT, Version 1*, Certificate of RF Registration of Computer Program No. 2022666957. Date of Registration: September 12, 2022.
6. N. I. Tarasov, V. O. Podryga, S. V. Polyakov, and A. V. Timakov, “Digital Platform for Supercomputer Mathematical Modeling of Spraying Processes,” *Russ. Digit. Libr. J.* **25** (6), 697–721 (2022). doi [10.26907/1562-5419-2022-25-6-697-721](https://doi.org/10.26907/1562-5419-2022-25-6-697-721).
7. TypeScript: JavaScript with Syntax for Types. <https://www.typescriptlang.org/>. Cited December 6, 2023.
8. S. P. Polyakov, A. P. Demichev, and A. P. Kryukov, “Web Toolkit for Scientific Research: State of the Art and the Prospect for Development,” *Procedia Comput. Sci.* **66**, 429–438 (2015). doi [10.1016/j.procs.2015.11.049](https://doi.org/10.1016/j.procs.2015.11.049).
9. A. P. Kryukov, A. P. Demichev, and S. P. Polyakov, “Web Platforms for Scientific Research,” *Program. Comput. Softw.* **42** (3), 129–141 (2016). doi [10.1134/S036176881603004X](https://doi.org/10.1134/S036176881603004X).
10. Prove.Design. <https://prove.design>. Cited December 6, 2023.
11. J. Magill, R. Dreher, Z. Sóti, and G. P. Lasché, “Nucleonica: Web-Based Software Tools for Simulation and Analysis,” in *Proc. 21st Int. Conf. on Nuclear Energy for New Europe, Ljubljana, Slovenia, September 5–7, 2012*. https://arhiv.djs.si/proc/nene2012/Publication_datoteke/Proceedings/1303.pdf. Cited December 6, 2023.
12. pSeven Enterprise. <https://www.pseven.io/product/pseven-enterprise/>. Cited December 7, 2023.
13. O. Sukhoroslov, S. Volkov, and A. Afanasiev, “A Web-Based Platform for Publication and Distributed Execution of Computing Applications,” in *Proc. 14th Int. Symposium on Parallel and Distributed Computing (ISPDC), Limassol, Cyprus, June 29–July 2, 2015*. doi [10.1109/ISPDC.2015.27](https://doi.org/10.1109/ISPDC.2015.27).
14. Everest. <https://everest.distcomp.org/apps/list>. Cited December 7, 2023.
15. M. McLennan and R. Kennell, “HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering,” *Comput. Sci. Eng.* **12** (2), 48–53 (2010). doi [10.1109/MCSE.2010.41](https://doi.org/10.1109/MCSE.2010.41).
16. K. Madhavan, L. Zentner, V. Farnsworth, et al., “nanoHUB.org: Cloud-Based Services for Nanoscale Modeling, Simulation, and Education,” *Nanotechnol. Rev.* **2** (1), 107–117 (2013). doi [10.1515/ntrev-2012-0043](https://doi.org/10.1515/ntrev-2012-0043).
17. S. Steiger, M. Povolotskiy, H.-H. Park, et al., “NEMO5: A Parallel Multiscale Nanoelectronics Modeling Tool,” *IEEE Trans. Nanotechnol.* **10** (6), 1464–1474 (2011). doi [10.1109/TNANO.2011.2166164](https://doi.org/10.1109/TNANO.2011.2166164).
18. A. Alber, J. Nabrzyski, and T. Wright, “The HUBzero Platform: Extensions and Impressions,” in *Proc. 3rd Int. Workshop on Science Gateways for Life Sciences (IWSG 2011), London, United Kingdom, June 8–10, 2011*. <https://ceur-ws.org/Vol-819/paper2.pdf>. Cited December 7, 2023.
19. D. Mejia, T. Kubis, and G. Klimeck, “NemoViz: a Visual Interactive System for Atomistic Simulations Design,” *Vis. Eng.* **6**, Article Number 6 (2018). doi [10.1186/s40327-018-0067-4](https://doi.org/10.1186/s40327-018-0067-4).
20. B. E. Granger and F. Pérez, “Jupyter: Thinking and Storytelling with Code and Data,” *Comput. Sci. Eng.* **23** (2), 7–14 (2021). doi [10.1109/MCSE.2021.3059263](https://doi.org/10.1109/MCSE.2021.3059263).
21. F. Pérez and B. E. Granger, “IPython: a System for Interactive Scientific Computing,” *Comput. Sci. Eng.* **9** (3), 21–29 (2007). doi [10.1109/MCSE.2007.53](https://doi.org/10.1109/MCSE.2007.53).
22. GitHub — voila-dashboards/voila: Voilà turns Jupyter notebooks into standalone web applications. <https://github.com/voila-dashboards/voila>. Cited December 7, 2023.
23. R. Thomas and S. Cholia, “Interactive Supercomputing with Jupyter,” *Comput. Sci. Eng.* **23** (2), 93–98 (2021). doi [10.1109/MCSE.2021.3059037](https://doi.org/10.1109/MCSE.2021.3059037).
24. O. I. Samovarov and S. S. Gaysaryan, “The Web-laboratory Architecture Based on the Cloud and the UniHUB Implementation as an Extension of the OpenStack Platform,” *Proceedings of the Institute for System Programming of the RAS.* **26** (1), 403–420 (2014). doi [10.15514/ISPRAS-2014-26\(1\)-17](https://doi.org/10.15514/ISPRAS-2014-26(1)-17).
25. S. V. Polyakov, A. V. Vyrodov, D. V. Puzyrkov, and M. V. Yakobovskiy, “Cloud Service for Decision of Multiscale Nanotechnology Problems on Supercomputer Systems,” *Proceedings of the Institute for System Programming of the RAS.* **27** (6), 409–420 (2015). doi [10.15514/ISPRAS-2015-27\(6\)-26](https://doi.org/10.15514/ISPRAS-2015-27(6)-26).
26. Electron. Build cross-platform desktop apps with JavaScript, HTML, and CSS. <https://www.electronjs.org/>. Cited December 7, 2023.
27. TOP500. The List. June 2023. <https://www.top500.org/lists/top500/2023/06/>. Cited December 7, 2023.
28. SUPPZ. <http://suppz.jsc.ru/>. Cited December 7, 2023.
29. CKP KIAM RAS. <https://ckp.kiam.ru/?home>. Cited December 7, 2023.
30. Node.js. <https://nodejs.org/>. Cited December 7, 2023.
31. NestJS: A progressive Node.js framework. <https://nestjs.com/>. Cited December 7, 2023.
32. Express: Node.js web application framework. <https://expressjs.com/>. Cited December 7, 2023.

33. TypeORM: Amazing ORM for TypeScript and JavaScript. <https://typeorm.io/>. Cited December 7, 2023.
34. Vue.js: The Progressive JavaScript Framework. <https://vuejs.org/>. Cited December 7, 2023.
35. Quasar Framework. <https://quasar.dev/>. Cited December 7, 2023.
36. Request for Comments: 6455. The WebSocket Protocol. <https://datatracker.ietf.org/doc/html/rfc6455>. Cited December 7, 2023.
37. PM2 - Home. <https://pm2.keymetrics.io/>. Cited December 7, 2023.
38. Redis. <https://redis.io/>. Cited December 7, 2023.
39. Git. <https://git-scm.com/>. Cited December 7, 2023.
40. The Official YAML Web Site. <https://yaml.org/>. Cited December 7, 2023.
41. Decorator-based transformation, serialization, and deserialization between objects and classes. <https://github.com/typestack/class-transformer>. Cited December 7, 2023.
42. Decorator-based property validation for classes. <https://github.com/typestack/class-validator>. Cited December 7, 2023.
43. Markdown Guide. <https://www.markdownguide.org/>. Cited December 7, 2023.
44. A. A. Bondarenko, E. M. Kononov, O. A. Kosolapov, et al., “Software Package GIMM_NANO,” in *Proc. Int. Supercomputer Conf. Scientific Service on the Internet: All Facets of Parallelism, Novorossiysk, Russia, September 23–28, 2013*. <http://agora.guru.ru/abrau2013/pdf/333.pdf>. Cited December 7, 2023.
45. T. G. Elizarova, *Quasi-Gas Dynamic Equations* (Springer, Berlin, 2009). doi 10.1007/978-3-642-00292-2.
46. V. O. Podryga and S. V. Polyakov, “Multiscale Mathematical Modeling of the Metal Nanoparticles Motion in a Gas Flow,” in *Lecture Notes in Computer Science* (Springer, Cham, 2019), Vol. 11386, pp. 387–394. doi 10.1007/978-3-030-11539-5_44.
47. J. M. Haile, *Molecular Dynamics Simulations: Elementary Methods* (Wiley, New York, 1992).
48. V. O. Podryga, S. V. Polyakov, and D. V. Puzyrkov, “Supercomputer Molecular Modeling of Thermodynamic Equilibrium in Gas–Metal Microsystems,” *Numerical Methods and Programming (Vychislitel’nye Metody i Programirovanie)* **16** (1), 123–138 (2015). doi 10.26089/NumMet.v16r113.
49. Ansys Fluent. Fluid Simulation Software. <https://www.ansys.com/products/fluids/ansys-fluent>. Cited December 7, 2023.
50. CFD Module. Simulate Single-Phase and Multiphase Flow. <https://www.comsol.com/cfd-module>. Cited December 7, 2023.
51. S. V. Polyakov, V. O. Podryga, D. V. Puzyrkov, and T. A. Kudryashova, “Supercomputer Molecular Modeling of Gas-Dynamic Deposition of Nanoparticles onto a Substrate,” in *Proc. Int. Conf. on Russian Supercomputing Days, Moscow, Russia, September 24–25, 2018* (Mosk. Gos. Univ., Moscow, 2018), pp. 782–792. <http://russianscday.s.org/files/pdf18/782.pdf>. Cited December 7, 2023.
52. V. Podryga and S. Polyakov, “Atomistic Modeling of Metal Nanocluster Motion Caused by Gas Flow Impact,” *Lobachevskii J. Math.* **40** (11), 1987–1993 (2019). doi 10.1134/S1995080219110210.
53. Visualizer. <https://kitware.github.io/visualizer/>. Cited December 7, 2023.
54. U. Ayachit, *The ParaView Guide: A Parallel Visualization Application* (Kitware Inc., Clifton Park, 2015).

Received
November 1, 2023

Accepted for publication
December 4, 2023

Information about the authors

Nikita I. Tarasov — Ph.D., Scientist; Keldysh Institute of Applied Mathematics, Miusskaya ploshchad’ 4, 125047, Moscow, Russia.

Viktoriiia O. Podryga — Dr. Sci., Assistant Professor, Leading Scientist; Keldysh Institute of Applied Mathematics, Miusskaya ploshchad’ 4, 125047, Moscow, Russia.

Sergey V. Polyakov — Dr. Sci., Senior Researcher, Leading Scientist; Keldysh Institute of Applied Mathematics, Miusskaya ploshchad’ 4, 125047, Moscow, Russia.