

УДК 004.5, 004.896

© С. П. Копысов, А. К. Новиков, В. Н. Рычков, Ю. А. Сагдеева, Л. Е. Тонков

ВИРТУАЛЬНАЯ ЛАБОРАТОРИЯ КОНЕЧНО-ЭЛЕМЕНТНОГО МОДЕЛИРОВАНИЯ¹

В данной работе формулируются основные задачи, решаемые инструментальной программной системой «Виртуальная лаборатория» для конечно-элементного анализа и параллельных вычислений, и ее место в научно-исследовательской и учебной работе научного коллектива. Система организует эффективное взаимодействие разработчиков, заказчиков, обучение пользователей и студентов и обеспечивает их основными научными сервисами для конечно-элементного моделирования на многопроцессорных вычислительных системах с широким применением web-технологий.

Ключевые слова: web-приложения, метод конечных элементов, параллельные вычисления.

Введение

Информационные технологии, как показывают время и практика, охватывают самые различные аспекты и организационные уровни научной и образовательной сферы. Внедрение информационных технологий в деятельность научного коллектива позволяет выйти за рамки традиционных, достаточно разработанных форм работы и более эффективно осуществлять совместную работу как внутри коллектива, так и вне его — с другими участниками научно-образовательного процесса.

Цель данной работы состоит в разработке Виртуальной лаборатории как программно-аппаратной научно-исследовательской учебной среды, способной поддержать все этапы жизненного цикла задач математического моделирования и параллельных вычислений и предоставить web-интерфейс. Для создания системы применяются web-технологии, новые подходы к разработке и сопровождению программ, технологии обучения. Система Виртуальной лаборатории имеет многозвенную архитектуру: web-браузер и специализированные клиентские приложения; web-сервер, СУБД, файловый сервер; вычислительные программные системы и кластеры.

§ 1. Виртуальная лаборатория как услуга, использующая открытые стандарты и открытое программное обеспечение

Популярность предоставления программного обеспечения как услуги во всем мире постоянно растет. Благодаря web-приложениям функции любой вычислительной программы могут стать доступными через Интернет. Таким образом, такие программные средства как PHP, ASP, JSP, JavaBeans, COM и пр. могут теперь обращаться к какой-либо программе, работающей на другом сервере (то есть к web-сервису), и использовать результат, полученный от нее, в собственном приложении. В этой идее нет ничего нового, но это большой шаг вперед к упрощенному доступу к программам через сеть. Основываясь на стандартных форматах связи, web-приложения могут поменять представление о том, как можно работать над проектами в научном коллективе. Возможность использования данных средств и свойств в сети стала предпосылкой создания Виртуальных лабораторий. Можно привести их следующую классификацию [1]:

- *Виртуальная лаборатория как научная лаборатория* — виртуальный аналог традиционного для академической среды научного коллектива. Интернет-технологии позволяют

¹Работа выполнена при финансовой поддержке РЦП УрО РАН и РФФИ (гранты 09-01-00061, 10-01-96039-урал)

имитировать рабочую среду. Виртуальной лаборатории присущи основные качества исследовательской лаборатории коллективного творчества, а также набор соответствующих технических и аппаратных возможностей;

- *Виртуальная лаборатория как общедоступный интернет-ресурс*, продвигающий программные продукты на рынок и обеспечивающий возможность обучения их использования;
- *виртуальная обучающая лаборатория* — виртуальный аналог комплекса учебных материалов, приборов, позволяющих моделировать эксперименты.

В методическом плане понятие виртуальная научная лаборатория для математического моделирования на параллельных вычислительных системах гораздо шире и включает все вышеперечисленные типы. Виртуальная лаборатория это не просто интегрированная система для математического моделирования, объединяющая открытые научные пакеты прикладных программ, компоненты САД-систем, удаленные вычислительные кластеры, но и основа совместной разработки, сопровождения и использования прикладного программного обеспечения. Кроме того, Виртуальная лаборатория может использоваться в учебно-исследовательских работах преподавателей и студентов.

Рассмотрим существующие программные технологии, которые могут быть использованы для реализации Виртуальной лаборатории. Это, прежде всего, технологии распределенных вычислений (**Grid computing**) [2] — когда большая ресурсоемкая вычислительная задача распределяется для выполнения между множеством компьютеров, объединенных в мощный вычислительный кластер по локальной сети или в несколько кластеров, соединенных через Интернет. Grid-технология позволяет объединить разнородные элементы информационной инфраструктуры — вычислительные ресурсы, средства хранения данных и файловые системы — в единую систему. Архитектура Grid предоставляет многие необходимые технологические компоненты для построения распределенных САД/САМ/САЕ-систем, охватывающих не только совместное проектирование, но и вычислительный уровень [3]. Среди основных проблем, которые необходимо решить в такой среде, — управление ресурсами, обеспечение защиты информации и др.

Вторая технология — это «облачные» вычисления (**Cloud computing**) [4], которые представляют собой динамически масштабируемый способ доступа к внешним вычислительным ресурсам в виде web-сервиса, при этом пользователю не требуется никаких особых знаний об инфраструктуре «облака» или навыков управления этой «облачной» технологией. В общем случае сервисы «облачных» вычислений представляют собой web-приложения, доступ к которым обеспечивается посредством обычного web-браузера. Пользователю совершенно не нужно обладать специальным высокопроизводительным аппаратным обеспечением для запуска пользовательского программного приложения.

В первом приближении обе эти технологии похожи, так как они позволяют решать вычислительные задачи с использованием большого количества распределенных узлов. Принципиальная разница заключается в подходах к решению этих задач. Для работы в Grid необходимо владеть определенными навыками управления обработчиками, очередями, выполняющимися в сети процессами и прочими специфическими системами промежуточного программного обеспечения. В «облачных» системах аналогичного рода задачи решаются намного проще: от пользователя не требуется детальное знание вычислительной платформы, системного и прикладного программного обеспечения. За аппаратно-программную платформу в «облаке» отвечают специалисты.

В расширенном трактовании под термином «облачные» вычисления подразумевают предоставление не только программного обеспечения, но и аппаратных ресурсов и инфраструктуры. Варианты предоставления вычислительных ресурсов существенно отличаются.

- *Программное обеспечение как услуга (SaaS)* или приложения в виде сервисов — вариант,

при котором предлагается совместное использование какого-либо программного обеспечения.

- *Данные как услуга* — понятие «данные как услуга» с появлением архитектуры для обслуживания широкого круга запросов (SOA) включает стандартизированные процессы для обращения к данным, в то время как фактическая платформа, на которой постоянно находятся данные, не имеет значения.
- *Платформа как услуга* — в отличие от SaaS, предназначенного больше для конечного пользователя, этот вариант — для разработчиков. В «облаке» функционирует некоторый набор программ, основных сервисов и библиотек, на основе которых предлагается разрабатывать свои приложения.
- *Инфраструктура как услуга* — означает предоставление некоторых базовых аппаратных возможностей и ресурсов в виде сервисов. Термин пришел на смену аппаратному обеспечению как услуге. Примером являются средства Виртуализации, балансировщики нагрузки и тому подобные системы, лежащие в основе построения других систем.
- *Рабочее место как услуга* — это предоставление рабочего места как услуги, частный случай Инфраструктуры как услуги.
- и другие.

Все приведенные варианты использования данной технологии предполагают следующее.

- Архитектура приложения скрыта от пользователя. Приложение не требует ничего от пользователя, кроме как запустить браузер и набрать строку URL.
- Пользователь не является администратором, и нет необходимости устанавливать, настраивать, поддерживать и решать возникающие проблемы конкретного ПО. В роли администратора выступает разработчик web-приложения.
- Приложение находится в одном месте, и нет необходимости устанавливать прикладное ПО на свое рабочее место. Достаточно загрузить только ту его часть, которая требуется для выполнения конкретной текущей задачи.
- Проблемы поддержки разных версий не существует, так как web-приложение находится на сервере и есть только одна его копия, которую поддерживает разработчик.

Web-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — web-сервер. Логика web-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по сети.

Открытые протоколы обмена и передачи данных обеспечиваются web-сервисами [5]. Web-сервисы в настоящее время фактически являются универсальной технологией связывания существенно разнородных систем. В ее основе лежит несколько основных стандартов: XML для описания данных, SOAP для передачи информации с одних систем на другие, WSDL для описания сервисов (в том числе задания типов входных и выходных данных) и UDDI для хранения и предоставления по запросу WSDL-описаний. У web-сервиса нет пользовательского графического интерфейса и он не предназначен для обслуживания конечных пользователей. Вместо этого у него есть программный интерфейс, то есть web-сервис реализует функции — web-методы, которые могут быть вызваны удаленно по сети, и предоставляет услуги другим web-приложениям — приложениям с графическим пользовательским интерфейсом или консольным приложениям.

Клиентская часть web-приложения реализует пользовательский интерфейс, который формирует запросы к серверу и обрабатывает ответы от него. Для формирования и обработки запросов, создания интерактивного и независимого от браузера интерфейса используются: Java Applet; JavaScript; Silverlight; ActiveX; Adobe Flash; Adobe Flex и так далее.

Для создания web-приложений на стороне сервера используются разнообразные технологии и языки программирования: C/C++; Java; Python; Perl; PHP и др. Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует web-страницу и отправляет ее клиенту по сети с использованием протокола HTTP.

В будущем, по крайней мере — ближайшем, web-приложения вряд ли смогут заменить все системные программы и выполнять задачи, доступные только консольным приложениям. В рамках Виртуальной лаборатории моделирования, по-видимому, больше преимуществ будет иметь гибридная модель, которая подразумевает как использование локального программного обеспечения, так и получение web-приложений через Интернет. В идеале многие вычислительные программные комплексы должны работать в подобном гибридном режиме.

Общая схема рассматриваемой Виртуальной лаборатории состоит из клиентской и серверной частей. Клиентская часть включает: программные средства совместной работы; Python-интерфейс к расчетным библиотекам; скрипты создания заданий для моделирования; Python-интерфейс к открытым программам визуализации результатов моделирования и так далее. Серверная часть имеет базы данных, библиотеки, расчетные программы, web-сервер. Выполнение расчетов производится в едином пользовательском интерфейсе, построенном на базе web-технологий и доступном через web-браузер. Кроме того, имеются мощные средства разработки дополнений, для которых используется Python.

§ 2. Вычислительные web-приложения в Internet

В настоящее время для работы с вычислительными пакетами широко используются различные варианты клиент-серверных технологий с доступом пользователей к вычислительному ядру того или иного пакета через web-интерфейс. В этом случае на клиентской стороне необходимо лишь интернет-браузер, а вся вычислительная работа выполняется на серверной стороне.

Проприетарные программные продукты MatLab, MathCad, Maple, Mathematica имеют средства удаленной работы с вычислительными пакетами — Matlab Web-server (MWS), Mathcad Calculation Server, MapleNet, webMathematica соответственно. Как показывает практика, все они требуют некоторой доработки скриптов или документов для организации web-интерфейса. Технологически средства организации удаленного доступа несколько различаются, но характерным примером может быть пакет webMathematica, использующий технологию сервлетов и JSP (Java Server Pages).

Что касается вычислительных пакетов с открытым кодом, то развитые средства для организации удаленной работы существуют для системы статистической обработки данных **R**, позволяющие запустить TCP/IP-сервер для доступа других программ к возможностям **R** — это Rserve (<http://www.rforge.net/Rserve/>). Вэб-интерфейс, предназначенный для удаленной работы с Octave (см. <http://hara.mimuw.edu.pl/web octave/web/>) и пакетом Maxima (<http://maximaphp.sourceforge.net/>), по организации и возможностям мало отличается от web-интерфейсов к пакету **R**.

Впечатляющие возможности удаленной работы с вычислительными пакетами обеспечивает Sage (<http://sagemath.org>) — это система компьютерной алгебры, работающая прозрачным образом с другими математическими пакетами. Также доступен web-интерфейс — Sage-notebook, который можно опробовать на сайте разработчиков. При помощи Sage-notebook можно удаленно работать с большим числом математических и графических пакетов, входящих в состав Sage, а также публиковать в Интернете интерактивные документы (используя скрипты для различных пакетов и результаты их работы при помощи Python). Выполнение расчетов производится в едином пользовательском интерфейсе, построенном на базе web-технологий и доступном через web-браузер. Имеются мощные средства разработки дополнений и скриптинга, для которых используется Python, а не специализированный язык, как во многих других

системах. Модель разработки и технологии Sage основаны на большом стремлении к открытости, кооперации и совместной работе.

В силу этого реализуемая в настоящее время инициатива разработчиков Sage по объединению open-source проектов может привести к созданию единой, достаточно функциональной, открытой системы символьных вычислений с удаленным доступом. Sage уже поглотил проекты Axiom, GAP, GP/PARI, Macaulay2, Maxima, Octave, Singular и некоторые другие. Отметим, что создание единой системы пока не приводит к созданию громоздкой системы, которая утратила бы свою гибкость и потенциал развития. В связи с этим следует выделить открытый проект **FEMhub** (<http://femhub.org>) для конечно-элементного анализа, который использует всю инфраструктуру Sage в качестве программного окружения. Данная система имеет интерфейс web-notebook, используемый в Sage, и Python. В FEMhub включены открытые пакеты:

- FiPy (<http://www.ctcms.nist.gov/fipy/>),
- Hermes (<http://hpfem.org/main/hermes.php>),
- Phaml (<http://math.nist.gov/phaml/>),
- SfePy (<http://www.sfepy.org/>);

средства научной визуализации данных:

- mayavi (<http://mayavi.sourceforge.net/>),
- matplotlib (<http://matplotlib.sourceforge.net/>),
- pyglet (<http://www.pyglet.org/>);

Python-библиотеки:

- Scipy (<http://www.scipy.org/>),
- Numpy (<http://numpy.scipy.org/>),
- Sympy (<http://code.google.com/p/sympy/>).

Другим интересным проектом, имеющим единую инфраструктуру, можно назвать проект **MathCloud** (<http://mathcloud.org>), в котором основным средством запуска приложений также является браузер. На уровне сервисов реализуется удаленный программный доступ к востребованной пользователем MathCloud функциональности. По сравнению с технологиями Web-сервисов на основе протокола SOAP (Simple Object Access Protocol) в проекте использована архитектура REST (Representational State Transfer), представляющая собой простой унифицированный интерфейс на основе открытых стандартов HTTP.

Активно разрабатываются расчетные программные комплексы, работающие в режиме online и направленные на решение определенного круга прикладных задач [6–8]. Нельзя не отметить здесь проект по созданию Центра компьютерного моделирования (<http://SciShop.ru>), предназначенного для реализации современных потребностей в распространении научных продуктов. Он направлен на решение проблем, связанных с разнообразными научными и техническими аспектами разработки специализированного Web-ресурса, а также на его продвижение, в том числе и на платной основе. В качестве контента используется группа функциональных программных комплексов. Данный проект характеризуется специализированной инфраструктурой, которая содержит не только собственно программные комплексы как аппаратную платформу проведения компьютерных расчетов, но и поддерживающие их структуры (библиографический раздел, табличные и графические базы данных, содержащие результаты уже проведенных расчетов), обеспечивает систему обучения специалистов и студентов программным комплексам. Данный web-сервис взаимодействует с вычислительными ресурсами суперкомпьютерного центра.

Обзор публикаций указал на присутствие ряда проблем, связанных с реализацией вычислительных web-приложений. В основном это проблемы многопользовательского режима, интерактивности и визуализации расчетных данных.

Открытая информационно-вычислительная система «Виртуальная лаборатория» должна, в первую очередь, обеспечивать: создание, хранение и удаленную визуализацию исходных данных, расчетных САД-моделей и сеток; доступ к вычислительным кластерам и выполнение параллельных распределенных вычислительных программ; совместную разработку и сопровождение программного обеспечения и документации; информационно-образовательную среду.

§ 3. Вычислительный комплекс для конечно-элементного моделирования как основа Виртуальной лаборатории

Основой Виртуальной лаборатории является параллельная распределенная объектно-ориентированная вычислительная система FEStudio для конечно-элементного моделирования в двух- и трехмерных областях [9]. FEStudio состоит из следующих подсистем: *геометрия* [10], *сетка* [11], *расчетная сетка*, *разделенная сетка*, *распределенная сетка* [12] и *расчетные модули* [13–15]. Каждый уровень включает компоненты ввода/вывода, объектно-ориентированной модели и визуализации. В свою очередь, объектно-ориентированная модель разбивается еще на две составляющие: структуры данных и алгоритмы. Деление на уровни отражает принцип декомпозиции сложных задач на более простые компоненты. Каждый уровень допускает ортогональное наращивание функциональности путем добавления модулей, реализующих новые решения. Ядро вычислительной системы платформо-независимо и реализовано на языке C++ с использованием ряда открытых математических библиотек.

Для построения САД-геометрий для системы FEStudio используются две свободно распространяемые системы: FreeCAD (<http://sourceforge.net/projects/free-cad/>) и HeeksCAD (<http://code.google.com/p/heelscad/>). Обе эти системы геометрического моделирования основаны на Open CASCADE. В них реализован импорт и экспорт геометрии в форматах IGES, STEP, STL, dxf и многих других. В состав FEStudio входит также графическое приложение (рис. 1), реализованное с помощью MFC (Microsoft Foundation Classes) и Open CASCADE (<http://www.opencascade.com>). Данное приложение использует функционал Open CASCADE для отображения сложных геометрических объектов (вращение, масштабирование, движение сцены, управление прозрачностью и так далее). Модуль визуализации сетки позволяет отобразить сетку в различных режимах: узлы; ребра; поверхностная сетка; отдельные ячейки сетки; часть сетки, определяемая логическим выражением, зависящим от координат ячейки. Модуль визуализации расчетных данных позволяет задавать различные граничные условия (закрепление, вектор движения, нагрузки и др.). Отображение разделенной сетки дает возможность отображения подсеток разными цветами, а также возможность разнесения подсеток.

В ходе эксплуатации вычислительной системы FEStudio возникла необходимость совместной работы над геометрическими моделями, а также визуальной обработки расчетных сеточных моделей больших объемов. Эти проблемы могут быть решены в рамках проекта Виртуальной лаборатории путем создания web-приложений, обеспечивающих единые базы данных рассчитываемых конструкций, генерацию и обработку сеток через интернет-браузер.

§ 4. Web-приложения для построения САД-моделей и расчетных сеток

В качестве языка расширения системы FEStudio исходно предполагалось использовать Python. Главными аргументами были следующие: Python легко встраивается в программу; в отличие от других языков Python может взаимодействовать с системой как через механизм COM, CORBA, так и через простой, но в то же время мощный, API к C/C++. По мере использования Python выяснилось, что он может быть прекрасно использован не только для управления системой, но и для написания полноценных и весьма сложных подсистем. Кроме того, Python сочетает в себе понятный синтаксис и имеет развитые средства обработки и создания web-приложений. Поэтому естественным был выбор библиотеки pythonOCC (<http://www.pythonOCC.org>), построенной на ядре моделирования Open CASCADE. Библиотека является связующим звеном между открытым ядром 3D геометрического моделирования

Open CASCADE, написанного на языке C++, и интерпретируемым языком программирования Python, который все более широко используется в научных вычислениях и имеет API к многим открытым библиотекам. Отметим, что pythonOCC в настоящее время содержит более 10 000 классов и использует около 90 % функциональности Open CASCADE. Фактически PythonOCC является мультиплатформенной библиотекой для разработки CAD/CAE/PLM, обеспечивающей следующие возможности: трехмерное гибридное и параметрическое моделирование, работа с топологией, обмен данными (поддержка форматов файла STEP/IGES), совместная разработка и web-сервисы, поддержка управления графическим интерфейсом пользователя (wxPython, PyQt, Python-xlib). Кроме того, необходимо отметить, что pythonOCC расширяет ядро OpenCASCADE новой функциональностью: работа с базой знаний, совместная разработка; использование wxPython и wxWidgets позволяет не только создавать графический интерфейс, например, wxWidgets имеет построенную на ODBC библиотеку работы с базами данных, систему обмена данными между процессами, сетевую библиотеку, но и множество классов для работы с различными приложениями.

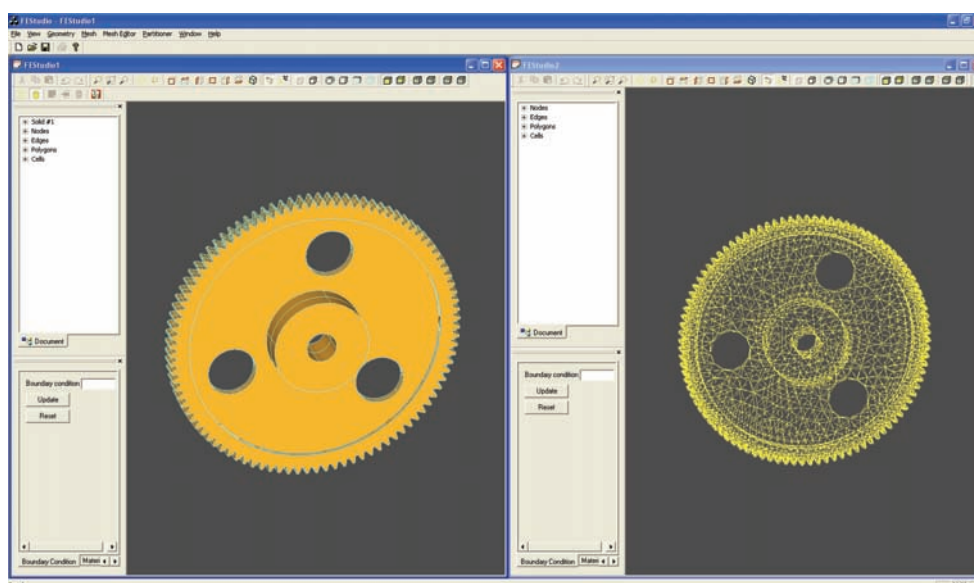


Рис. 1. Визуализация геометрии и сетки в программном комплексе FESstudio

Потенциально интересен модуль pythonOCC, обеспечивающий многопроцессорную обработку (рис. 2) и реализующий обработку CAD-геометрии в параллельном режиме, что дает заметное ускорение в генерации сложной геометрии. Модуль XML/RPC — клиент-серверное приложение, созданное на Python, позволяет реализовать совместную работу над геометрией: таким образом, геометрическая фабрика на сервере выдает модели, отображаемые на клиентских приложениях, что обеспечивает возможность совместной работы с геометрическими моделями (см. рис. 2).

На основе Open CASCADE, pythonOCC, сеточной структуры данных и методов построения и разделения расчетных сеток FESstudio создается визуальное окружение для построения трехмерных и двумерных сеток с web-интерфейсом на основе геометрических CAD-моделей. Построение разделенных сеток и их визуализация реализуется через API Python с возможностью основных методов отображения: вращение, масштабирование, перенос и рендеринг. Реализованы интерфейсы для Python для библиотек построения и разделения конечно-элементных сеток (см. рис. 3). Средства интерактивного пользовательского интерфейса обеспечивают быструю настройку (задание параметров) и непосредственное выполнение вычислений. Программный интерфейс обеспечивает также вызов функций построения сеток и их разделения из оригинальных прикладных программ. Прежде всего, это база данных геометрических CAD-моделей для расчетных задач для обеспечения совместного доступа.

Основным компонентом данной подсистемы является ядро геометрического моделирования

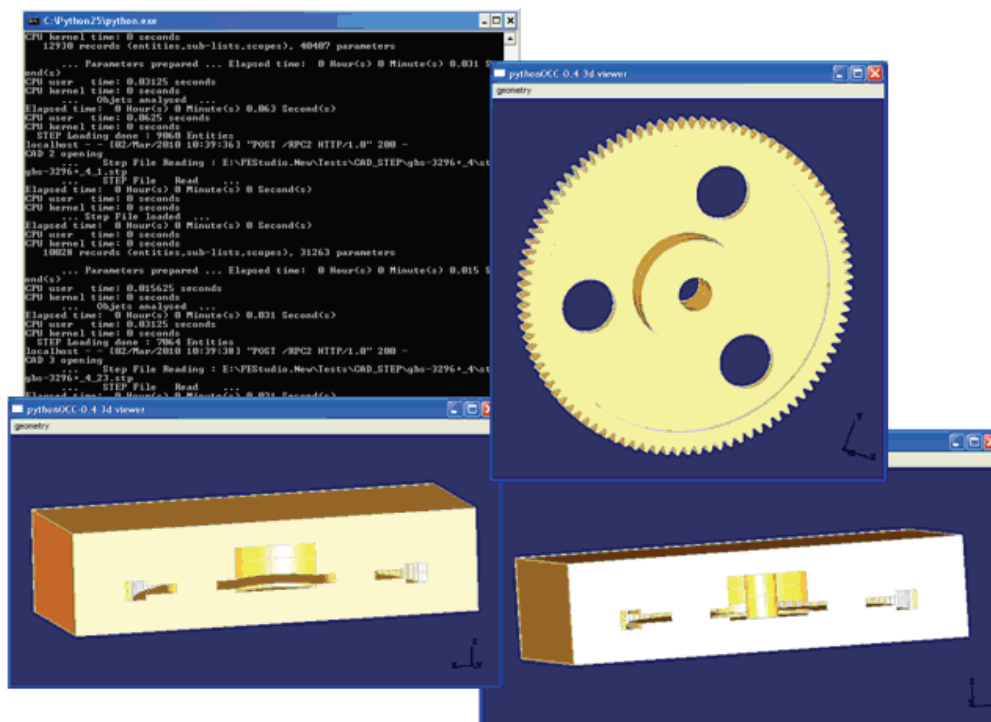


Рис. 2. Удаленное отображение комплексной геометрической модели

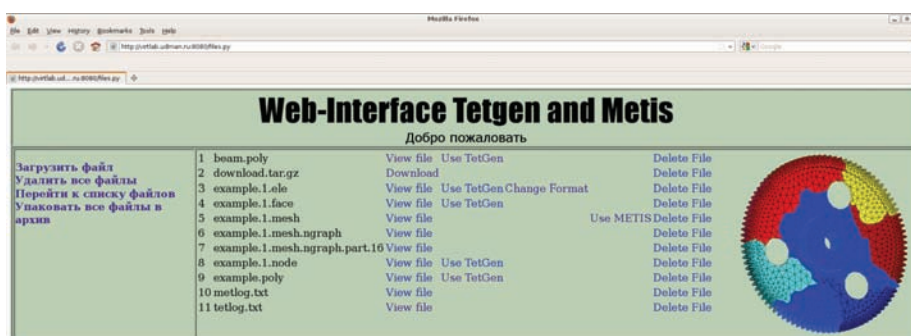


Рис. 3. Web-интерфейс к программам построения и разделения сеток

Open CASCADE. Далее используется программный интерфейс pythonOCC для геометрического ядра Open CASCADE, что обеспечивает построение web-интерфейса к геометрическому ядру для создания, отображения расчетных геометрических моделей и сеток.

Актуальной проблемой для web-лаборатории является создание подсистемы визуализации, которая позволила бы производить распределенную обработку данных большого объема на современных параллельных вычислительных системах. Удаленно в распределенном режиме должна быть также обеспечена возможность анализа и отображения некоторых численных результатов, поскольку манипуляции с расчетными данными становятся затруднительными на персональном компьютере исследователя.

Предполагается проведение анализа и выбора, реализации подсистемы удаленной визуализации на основе API Python для свободно распространяемых систем визуализации, например, Open CASCADE, MayaVi2, Paraview, Visit и др. В качестве обзора по данной проблеме можно выделить работы [16–19].

§ 5. Web-интерфейс к очереди вычислительных заданий на кластере

Обеспечение доступа к кластеру для запуска вычислительных заданий и мониторинг их выполнения удаленно осуществляется на основе PBSWeb и Ganglia.

Кластеры ИПМ УрО РАН (<http://www.udman.ru/cluster>) оснащены системой доступа к очереди заданий pbsweb-lite-0.95 (<http://webdocs.cs.ualberta.ca/~pinchak/PBSWeb>), которая представляет собой web-интерфейс для планировщика OpenPBS и обеспечивает разработку и запуск пакетных заданий в безопасном режиме. Для организации удаленного доступа к планировщику заданий установлен web-сервер Apache. С помощью технологии PHP и HTML форм с динамическими возможностями JavaScript автоматизированы процессы создания, модификации и повторного использования заданий; посредством технологии PHP реализовано взаимодействие с планировщиком заданий OpenPBS: постановка заданий в очередь, контроль за выполнением, получение результатов; сообщения о выполнении вычислительных заданий по e-mail; соединения защищены от несанкционированного доступа протоколом SSL, для запуска заданий требуется авторизация SSH.

Также инсталлирована масштабируемая распределенная система мониторинга Ganglia с web-интерфейсом для кластеров. Кроме того, установлено программное обеспечение для удаленного мониторинга и управления работы системой бесперебойного питания APC UPS вычислительного кластера, обеспечивающая дистанционный мониторинг и управление через простой Web/SNMP интерфейс.

§ 6. Совместная разработка и сопровождение программного обеспечения

Основная работа разработчика (программирование и отладка) выполняется на его рабочем компьютере. Весь инструментарий объединяется в интегрированную среду разработки (IDE — Integrated Development Environment). В первую очередь, интегрированные среды увеличивают продуктивность работы программиста, обеспечивая хорошо продуманный интерфейс к основным инструментам разработки программ: редакторам кода, компиляторам, библиотекам и так далее. Подключив подобную среду к web-сервисам, можно организовать процесс совместной разработки программ. Таким образом, интегрированная среда разработки становится еще одним примером клиентского специализированного приложения Виртуальной лаборатории. В своей работе мы используем открытые системы Eclipse (<http://eclipse.org>) и KDevelop

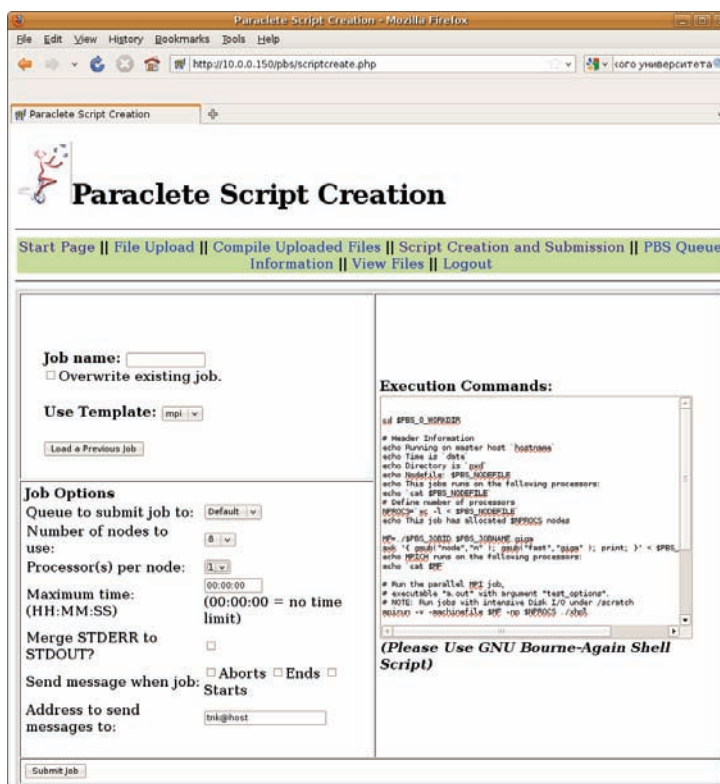


Рис. 4. Web-интерфейс PBSWeb к планировщику очереди заданий для кластера Paraclete

(<http://www.kdevelop.org>). Указанные системы являются клиентами для web-сервисов планирования работ, хранения исходного кода и ведения документации.

Коллективная работа в Виртуальной лаборатории организована с помощью открытой системы Trac (<http://trac.edgewall.org>). Она обеспечивает эффективное взаимодействие между территориально удаленными друг от друга разработчиками программ, а также исследователями и заказчиками. В настоящее время эта система применяется для разработки вычислительного ядра лаборатории <http://udman.ru/projects/FESstudio> и включает в себя следующие модули.

- **Планирование работ.** Это основной модуль системы. Он позволяет распределить задачи и проконтролировать их выполнение. Кроме того, обеспечивается учет ошибок, замечаний, пожеланий и это является основой для поддержки пользователей-исследователей. Данный модуль тесно интегрирован с остальными подсистемами, что позволяет, в частности, адресоваться к конкретным файлам и строкам исходного кода, статьям документации и так далее.
- **Репозиторий исходного кода.** Хранит версии файлов проекта, позволяет сравнивать их между собой, извлекать старые версии, синхронизировать код нескольких разработчиков. Построен на основе системы управления версиями Subversion (<http://subversion.tigris.org>).
- **Документирование.** Позволяет вести документацию по проекту, версиям, этапам разработки. Поддерживает возможность установки ссылок на другие разделы проекта (исходный код, заявки, план задач). Построена на основе технологии Wiki (язык для описания web-страниц с перекрестными ссылками и многопользовательская система для их создания и редактирования). Кроме того, имеется возможность автоматической генерации справочного руководства для разработчиков по комментариям, оставленным в исходном коде. Эта возможность реализована с помощью системы Doxygen (<http://www.stack.nl/~dimitri/doxygen>), предназначенной для автоматического документирования исходного кода, написанного на разных языках программирования.
- **Пакетирование и дистрибуция.** Важной задачей, связанной со сложными комплексами программ, является их установка и адаптация к различным аппаратно-программным платформам. Открытые инструменты Autotools (<http://www.gnu.org/software/autotools/>) решают эту задачу. Данный модуль обеспечивает web-интерфейс к Autotools и хранилище дистрибутивов пакетов прикладных программ, готовых к установке на различных платформах.
- **Список изменений.** Этот модуль позволяет отслеживать все изменения в системе за определенный период с возможностью подписки по протоколу RSS (Really Simple Syndication — семейство форматов для представления и сбора информации в Интернете).

При помощи Wiki (<http://www.udman.ru/wiki>) разработчиками совместно создается справочное руководство по программам и модулям, в том числе FESstudio. Технология обеспечивает возможность совместного редактирования документов, и, кроме того, сохраняется полная история сделанных изменений. Вся информацию, хранящуюся в системе, должны сопровождать сами пользователи — сотрудники Виртуальной лаборатории. Это позволит поддерживать правильность и актуальность информации. В противном случае страдает как содержательность технической документации, так и качество и полнота осмысления проекта его разработчиками, а в дальнейшем и пользователями. Для программных проектов wiki обладает еще одним преимуществом: документация доступна как разработчикам, так и заказчикам через web-интерфейс.

В качестве неотъемлемой части документации на сайте ИПМ УрО РАН представлены публикации, доклады, презентации группы разработчиков (<http://www.udman.ru/iam/ru/group/>

1798) с библиографическим описанием и web-сервис для работы с библиографией в формате bibtex (<http://jabref.sourceforge.net/jws/jabref.jnlp>).

§ 7. Информационно-образовательная среда

Созданная система позволяет естественным образом реализовать обучение основам конечно-элементного моделирования, параллельных вычислений и программному комплексу FEStudio с помощью модульной объектно-ориентированной динамической учебной среды, обеспечивающей организацию взаимодействия между преподавателем и слушателями (<http://www.udman.ru/course>). Программной оболочкой для реализации информационно-образовательной среды выбрана система управления обучением Moodle [20]. В электронный курс легко добавляются различные элементы: лекции, задания, форум, глоссарий, Wiki, чат и так далее. Moodle обладает большим набором средств коммуникации.

При изучении ряда математических курсов, таких как «Проекционно-сеточные методы», «Дополнительные главы вычислительной математики», используется пакет Sage, установленный на сервере Виртуальной лаборатории (<http://virtlab.udman.ru:8000/>). Наличие web-интерфейса к Sage обеспечивает простоту использования пакета в учебном процессе. Унифицированный интерфейс взаимодействия различных систем символьных и численных расчетов позволяет организовать обучение и совместное использование для решения задач на практических занятиях в области алгебры, дифференциального и интегрального исчисления, численного анализа и многих других разделов математики и механики.

Виртуальная лаборатория дает возможность участия в проекте по разработке свободного программного обеспечения в качестве замены традиционным курсовым, дипломным студенческим проектам, и при этом, как правило, не требуется «искусственного» создания примеров и заданий для практических занятий. Студентам, например, предлагается участие в уже существующем открытом проекте по разработке свободного ПО FEStudio. В проекте всегда находится список больших и малых задач, которые нуждаются в разработчиках. В этом случае у студента есть широкий выбор тем для проектов: от визуализации данных, вычислительных методов линейной алгебры до web-приложений. Таким образом, студенты получают навыки совместной работы в распределенном режиме, на практике осуществляется знакомство с основными инструментами (Wiki, системы контроля версий, отслеживания ошибок и другие). В результате отдельные подсистемы проекта и алгоритмы полностью разработаны студентами, отличаются хорошей документированностью и наличием большого числа тестовых примеров, а студенты получили навыки работы без отвлечения на практику в коммерческих структурах.

§ 8. Варианты использования информационно-вычислительной системы

В данном разделе рассматриваются основные варианты применения информационно-вычислительной системы «Виртуальная лаборатория» в научно-исследовательском, инженерном и учебном процессе.

Пример 1. Исторически первым и основным предназначением Виртуальной лаборатории является обеспечение научных исследований, проводимых в ИПМ УрО РАН. Пользователем системы в данном сценарии является научный сотрудник. Его работа в системе контролируется администратором сервера баз данных и кластера. Процесс создания расчетной модели и параллельных вычислений состоит из следующих основных этапов:

1. *Построение геометрической модели.* Инженер-исследователь работает на персональном компьютере с системами автоматизированного проектирования FreeCAD или HeeksCAD, которые представляют собой специализированные клиентские приложения Виртуальной лаборатории.
2. *Размещение построенной модели в базе данных.* Загрузка модели производится через web-браузер на сервер баз данных. Модели, хранящиеся в базе, могут быть повторно

использованы для построения более сложных моделей, для экспорта данных в другие форматы и так далее.

3. *Генерация сетки и задание граничных условий.* Используя либо визуальное приложение FEStudio, либо web-интерфейс FEStudio, исследователь строит расчетную модель либо в локальном, либо в удаленном режиме соответственно. Подробность разбиения сетки, построенной в локальном режиме, зависит от производительности персонального компьютера исследователя. Удаленное построение выполняется на высокопроизводительной системе и поэтому обеспечивает более высокую точность расчетных моделей. Расчетные модели, включающие в себя исходную геометрию, сетку и граничные условия, также сохраняются в базе данных.
4. *Выполнение расчетов на кластере.* Web-интерфейс кластера включает список расчетных пользовательских приложений и средства их запуска в многопользовательском режиме. Он позволяет исследователю запустить пользовательское приложение с заданными параметрами, одним из которых является расчетная модель, проследить за ходом ее выполнения и сохранить результаты в базе данных.
5. *Визуализация результатов.* Для визуализации результатов исследователь может использовать специализированные клиентские приложения Paraview, Visit. Однако результаты могут быть настолько объемными, что персональный компьютер исследователя может не справиться с их обработкой и визуализацией за приемлемое время. Поэтому данный этап может выполняться в удаленном режиме через web-браузер, когда картинка необходимой детализации генерируется удаленно высокопроизводительной графической станцией и затем передается по сети.

Пример 2. Следующий вариант описывает процесс разработки прикладных программ, с помощью которых выполняются расчеты в Виртуальной лаборатории. Основным пользователем в данном случае является программист-разработчик. Результат его работы — прикладная программа — устанавливается на кластере и становится доступной исследователям. Тем самым обеспечивается взаимосвязь между разработчиками и исследователями. Процесс разработки в Виртуальной лаборатории осуществляется посредством целого набора специальных инструментов, которые так или иначе связаны с web-сервисами лаборатории, и включает следующие шаги:

1. *Планирование работы.* Руководитель проекта составляет план работ, связанных с созданием нового и модификацией старого кода. План составляется с учетом ошибок, замеченных на предыдущих этапах разработки, и пожеланий пользователей-исследователей. В ходе работы над проектом руководитель видит прогресс каждого из разработчиков.
2. *Программирование и отладка.* Данный этап выполняется преимущественно на рабочих станциях разработчиков, которые могут быть распределены географически. Работа программистов синхронизируется посредством web-сервиса Репозиторий исходного кода. Функции разработчиков могут различаться. Например, некоторые из них могут отвечать только за реализацию численных методов, проектирование графического интерфейса, модульное тестирование или пакетирование.
3. *Документирование.* В ходе проектирования и реализации программ разработчики готовят документацию, которая включает в себя детальное описание структур данных и функций (документация для разработчиков), а также примеры и учебные пособия (документация для исследователей).
4. *Пакетирование и дистрибуция.* Основой пакета является набор скриптов, который позволяет сконфигурировать и установить программы на той или иной аппаратно-программной платформе. Скрипты разрабатываются по той же схеме, что и исходный код программ (см. шаг 2). Дистрибутивы пакетов размещаются через web-сервис.

5. *Сопровождение.* Исследователь имеет доступ как к стабильным версиям программ, так и к самым последним разработкам. Он сообщает об ошибках и оставляет свои замечания через Интернет. Получив подобное сообщение, разработчик модифицирует код и обновляет содержимое репозитория и дистрибутивов.

Пример 3. Последний из рассматриваемых вариантов показывает, как Виртуальная лаборатория может быть использована в процессе обучения. Пользователями системы являются преподаватели и студенты. Этим группам пользователей доступны все подсистемы Виртуальной лаборатории. В процессе обучения студенты могут выступать в роли исследователей, выполняя практические задания по моделированию на кластере, или разработчиков, реализовывая новые методы решения новых задач.

1. *Подготовка материалов учебного курса.* Преподаватель готовит материал лекционных и практических занятий. Будучи доступным онлайн, курс может предметно обсуждаться и корректироваться на заседании преподавательского коллектива, например, кафедры университета. Легко могут быть переработаны электронные материалы прошлых лет.
2. *Планирование учебной нагрузки.* Преподаватель распределяет лекционные и практические занятия на учебный семестр. Студенты видят объем работы на текущий учебный период.
3. *Раздача общих и индивидуальных заданий.* Общие и индивидуальные задания в электронном виде распределяются между студентами. Виртуальная лаборатория позволяет синхронизировать работу нескольких преподавателей, например, лектора, преподавателя практики и лаборанта. Все они имеют доступ к плану занятий, спискам студентов и их индивидуальных проектов.
4. *Контроль за выполнением заданий.* Web-лаборатория обеспечивает функции журнала успеваемости и упрощает обмен информацией между преподавателем и студентом через сеть Интернет, предоставляя средства проведения дополнительных консультаций и обсуждений.

§ 9. Заключение

Разрабатываемая Виртуальная лаборатория обеспечивает доступ удаленного пользователя к проблемно-ориентированному программному обеспечению и высокопроизводительным вычислительным ресурсам. Создание Виртуальной лаборатории математического моделирования, предназначенной для проведения научных вычислительных экспериментов, не ограниченных ни расстояниями, ни способом доступа и осуществляемых, возможно, в режиме реального времени, является актуальной научной задачей, имеющей важное практическое значение. Современная информационно-вычислительная инфраструктура — Виртуальная лаборатория с обеспечением доступа к открытым программным продуктам через сеть Интернет — может способствовать интенсификации деятельности научных коллективов, связанной с выполнением прикладных исследований.

СПИСОК ЛИТЕРАТУРЫ

1. Зинатуллин М. М. Создание виртуальной лаборатории проектирования тестового обеспечения // Научно-технический вестник СПбГИТМО (ТУ). Выпуск 10. Информация и правление в технических системах. — СПб.: СПбГИТМО(ТУ), 2003. — С. 120–125.
2. Grid Computing: Making the Global Infrastructure a Reality. Edited by F. Berman, G. Fox and T. Hey. — John Wiley and Sons, 2003. — 1012 p.
3. Копысов С. П., Пономарёв А. Б., Загребин М. Э. Проектирование CAD/CAE-системы как распределенной информационно-вычислительной Grid // Сб. статей «Численные методы в математике и механике». Ижевск: УрО РАН, 2007. — С. 47–50.
4. Miller M. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. — Que Publishing: Indianapolis, 2008. — 292 p.

5. Ньюкомер Э. Веб-сервисы: XML, WSDL, SOAP и UDDI. — С.-Петербург: Питер, 2003. — 256 с.
6. Емельянов В. Н., Плетнев И. В., Чугреев А. Л. Информационно-вычислительная система на основе технологий internet для химии и смежных областей // Препринт ИПМ № 100, Москва, 2003 г.
7. Астафьев А. С., Афанасьев А. П., Лазарев И. В., Сухорослов А. С., Тарасов О. В. Научная сервис-ориентированная среда на основе технологий Web и распределенных вычислений // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции (21–26 сентября 2009 г., г. Новороссийск). — М.: Изд-во МГУ, 2009. — 524 с.
8. Тарнавский Г. А. Облачные вычисления: контент, инфраструктура и технологии организации информационных потоков Центра компьютерного моделирования SciShop.ru // Исследовано в России. — 2010. — Т. 13. — С. 1–29.
9. Копысов С. П., Новиков А. К., Сагдеева Ю. А., Пономарёв А. Б., Рычков В. Н. Программная среда построения расчетных моделей метода конечных элементов для параллельных распределенных вычислений // Информационные технологии. — 2008. — № 3. — С. 75–82.
10. Копысов С. П., Пономарёв А. Б., Рычков В. Н. САД-ориентированный подход для построения САЕ-системы параллельных распределенных вычислений // Актуальные проблемы математики, механики, информатики: Сб. науч. тр. Пермь, 2008. — С. 101–106.
11. Копысов С. П., Новиков А. К., Пономарёв А. Б. О параллельном построении неструктурированных сеток // Параллельные вычислительные технологии (ПаВТ'2007): Сб. трудов Межд. конф. 29 янв.–2 фев. 2007, г. Челябинск: Изд. ЮУрГУ, 2007. — Т. 2. — С. 65–76.
12. Копысов С. П., Пономарёв А. Б., Рычков В. Н. Параллельная распределенная модель расчетной сетки // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. — 2008. — № 2. — С. 194–196.
13. Копысов С. П., Красноперов И. В., Рычков В. Н. Объектно-ориентированный метод декомпозиции области // Вычислительные методы и программирование. — 2003. — Т. 4, № 1. — С. 176–193.
14. Копысов С. П., Новиков А. К. Метод декомпозиции для параллельного адаптивного конечно-элементного метода // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. — 2010. — № 3. — С. 141–152.
15. Тонков Л. Е. Численное моделирование динамики капли вязкой жидкости методом функции уровня // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. — 2010. — № 3. — С. 134–140.
16. Бахтерев М. О., Васёв П. А., Казанцев А. Ю., Манаков Д. В. Система удаленной визуализации для инженерных и суперкомпьютерных вычислений // Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта–3 апреля 2009 г.). Челябинск: Изд. ЮУрГУ, 2009. — С. 382–388.
17. Iakobovski M., Nesterov M., Krinov P. Large distributed datasets visualization software, progress and opportunities // Computer graphics and geometry. — 2007. — V. 9, № 2. — P. 1–19.
18. Потехин А. Л., Тарасов В. И., Фирсов С. А. и др. ScientificView — система параллельной постобработки результатов, полученных при численном моделировании физических процессов // Вопросы атомной науки и техники. Сер: Матем. модел. физ. процессов. — 2008. — № 4. — С. 37–45.
19. Карташева Е. Л., Багдасаров Г. А., Болдарев А. С. и др. Визуализация данных вычислительных экспериментов в области 3D моделирования излучающей плазмы, выполняемых на многопроцессорных вычислительных системах с помощью пакета MARPLE // Научная визуализация. — 2010. — Т. 2, № 1. — С. 1–25.
20. Андреев А. В., Андреева С. В., Доценко И. Б. Практика электронного обучения с использованием Moodle. — Ростов-на-Дону: Изд-во ТТИ ЮФУ, 2008. — 146 с.

S. P. Kopysov, A. K. Novikov, V. N. Rychkov, U. A. Sagdeeva, L. E. Tonkov
Virtual laboratory for finite element modelling

In this paper, the features of the information-computing software system «Virtual laboratory for finite element simulation and parallel computing» are defined. We discuss how it can be used in research, development and education. This system supports collaboration between researchers and research customers, higher education and industrial training. Using Internet technologies, the Virtual laboratory provides scientific services for finite element simulation on multiprocessor platforms.

Keywords: web-based application, finite-element methods, parallel calculation.

Mathematical Subject Classifications: 68U35, 68W27, 68W10

Копысов Сергей Петрович, д. ф.-м. н., профессор кафедры Вычислительной механики Удмуртского государственного университета, ведущий научный сотрудник, Учреждение РАН Институт прикладной механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34, E-mail: s.kopysov@gmail.com

Новиков Александр Константинович, к. ф.-м. н., доцент кафедры Вычислительной механики Удмуртского государственного университета, старший научный сотрудник, Учреждение РАН Институт прикладной механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34, E-mail: an@udman.ru

Рычков Владимир Николаевич, к. ф.-м. н., доцент кафедры Вычислительной механики Удмуртского государственного университета, старший научный сотрудник, Учреждение РАН Институт прикладной механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34, E-mail: bob.r@mail.ru

Сагдеева Юлия Альбертовна, к. ф.-м. н., доцент кафедры Вычислительной механики Удмуртского государственного университета, старший научный сотрудник, Учреждение РАН Институт прикладной механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34, E-mail: sagdeeva@gmail.com

Тонков Леонид Евгеньевич, к. ф.-м. н., доцент кафедры Вычислительной механики Удмуртского государственного университета, старший научный сотрудник, Учреждение РАН Институт прикладной механики УрО РАН, 426067, Россия, г. Ижевск, ул. Т. Барамзиной, 34, E-mail: tnk@udman.ru