

ИСПОЛЬЗОВАНИЕ РАСПРЕДЕЛЕННОЙ СРЕДЫ REST-СЕРВИСОВ ДЛЯ ОПТИМИЗАЦИОННОЙ ИДЕНТИФИКАЦИИ УГЛЕРОДИСТЫХ СТРУКТУР ПО РЕНТГЕНОВСКИМ ДИФРАКЦИОННЫМ ХАРАКТЕРИСТИКАМ НАНОМАТЕРИАЛОВ¹.

В.В. Волошинов

Институт системного анализа РАН

vladimir.voloshinov@gmail.com

В.С. Неверов

Институт физики токамаков РНЦ «Курчатовский институт»

vs-never@hotmail.com

Аннотация

В работе [1] был предложен метод определения топологического состава углеродных наноструктур из достаточно широкого набора пространственных форм углерода для задачи интерпретации результатов рентгеновской дифрактометрии углеродистых пленок, образующихся внутри вакуумной камеры токамака Т-10 в результате электрического разряда. Метод включает проведение массовых расчетов кривых рентгеновского рассеяния на упомянутом широком классе углеродных наноструктур с последующим решением оптимизационной задачи определения вероятностей принадлежности элементарного рассеивателя (атома углерода) к наноструктуре определенной топологии и размера.

Необходимость в массивированных вычислениях с последующим применением достаточно производительных пакетов оптимизации для решения задач идентификации (используются три различных критерия качества приближения экспериментальных данных модельными), делает целесообразным перенос вычислений в распределенную вычислительную среду. Для проведения в будущем многовариантных расчетов и предоставления широкого доступа к созданному сценарию численного анализа результатов физического эксперимента, такой подход был реализован нами в системе RESTful-сервисов MathCloud (www.mathcloud.org) [2], [3], [4]. В докладе обсуждаются различные аспекты работ по реализации соответствующего вычислительного сценария, включая возможности программного инструментария MathCloud.

1. Введение. Физическое содержание задачи

Создание новых наноматериалов ставит задачу контроля свойств материалов в нанометровом диапазоне размеров. Одним из наиболее эффективных инструментов, способных решать такие задачи, является рентгеновская дифрактометрия в субнанометровом диапазоне длин волн. Между тем, анализ результатов рентгеновского рассеяния на наноматериалах имеет ряд сложностей, вызванных тем, что с наноструктурной среде пространственное упорядочение порой может сохраняться лишь на нескольких нанометрах, вследствие чего отсутствуют резкие брегговские пики, а имеющиеся широкие пики уже нельзя трактовать как брегговские. Поэтому дифракционный профиль содержит недостаточно информации для восстановления структуры сложного наноматериала. Так, восстановление распределения электронной плотности в среде методами Фурье-анализа не

дает исчерпывающей информации об объекте, поэтому нет возможности получить единственное решение. Стандартные же методы кристаллографии оказываются не применимыми. Мы предлагаем решать обратную задачу, а именно восстанавливать структуры нанобъектов, во-первых, значительно сужая класс возможных решений за счет использования результатов других диагностических методов, и во вторых, используя численное моделирование рентгеновского рассеяния на выбранном классе нанобъектов и сопоставляя расчетные данные с экспериментом.

Такой метод решения был использован в работе [1] для задачи интерпретации результатов рентгеновской дифрактометрии углеродистых пленок, образующихся внутри вакуумной камеры токамака Т-10 в результате электрического разряда. Требовалось определить возможный топологический состав углеродных наноструктур в образце.

2. Методика численного анализа экспериментальных данных

Предложенная в [1] анализа экспериментальных данных рентгеновского рассеяния на углеродных образцах включает проведение массовых расчетов кривых рентгеновского рассеяния на широком классе углеродных наноструктур с последующим решением оптимизационной задачи определения вероятностей принадлежности элементарного рассеивателя (атома углерода) к наноструктуре определенной топологии и размера.

Для расчета дифракционных характеристик наноструктур был создан численный код со следующими основными характеристиками:

- **На Входе.** Число структур в ансамбле ограничено только вычислительными мощностями, каждая структура представляется элементарным блоком (например, элементарная ячейка кристалла). Элементарный блок может иметь ограниченное только вычислительными мощностями количество копий, организованных в регулярные (кристаллы) и нерегулярные структуры. Нерегулярные структуры могут включать в себя регулярные и наоборот.
- **На Выходе.** Одномерный профиль интенсивности рассеяния рентгеновского излучения, рассчитанный по формуле Дебая, с возможностью вывода отдельных интерференционных вкладов. Двумерная дифракционная картина с возможностью усреднения по пространственным углам волнового вектора падающего излучения.
- **Интерфейс.** Простой XML интерфейс. Для генерации начальных данных, обработки и визуализации результатов используется набор скриптов, написанных на Python.
- **Вычисления.** Код написан на C/C++ и распараллелен при помощи как MPI, так и OpenMP.

XML интерфейс делает код удобным для автоматизированной генерации начальных данных, что необходимо при массовых расчетах, в данном случае в Гриде.

Поиск распределения изолированных наноструктур по их топологическим характеристикам и размерам проводился минимизацией следующих ошибок:

$$Z_j(\mathbf{x}, a, b, A) = S_{\text{exp}}(q_j) - \sum_{i=1}^N S_i(q_j) \cdot x_i - a \cdot (f_c(q_j))^2 - A \cdot \sum_{k=1}^{N_{\text{in}}} \alpha_k \cdot (f_k(q_j))^2 - b, \quad (j = 1 : m) \quad (1)$$

где $S_{\text{exp}}(q_j)$ - экспериментальный профиль интенсивности рентгеновского рассеяния, $S_i(q_j)$ - модельный профиль интенсивности рентгеновского рассеяния на i -той углеродной

наноструктуре, деленный на число атомов в ней, j - номер точки в дискретном пространстве модуля вектора рассеяния q , x_i/A - вероятность атома углерода принадлежать i -той наноструктуре, N – полное число выбранных наноструктур, a/A вероятность атома углерода принадлежать аморфной среде, $A = \sum_{i=1}^N x_i + a$, $f_c(q_j)$ - форм-фактор рентгеновского рассеяния на единичном атоме углерода, α_k – отношение числа атомов k -той примеси (включая водород) к числу атомов углерода в образце, взятое из эксперимента (в нашем случае [5]), N_{im} – полное количество значимых примесей в образце, $f_k(q_j)$ форм-фактор рентгеновского рассеяния на единичном атоме k -той примеси, b – неизвестная величина возможного постоянного фона, величина 0.5 отвечает учету средней ошибки экспериментальных данных, m – число точек дискретного пространства модуля вектора рассеяния q .

Дополнительные условия:

$$S_{\text{exp}}(q_j) - a \cdot (f_c(q_j))^2 - A \cdot \sum_{k=1}^{N_{im}} \alpha_k \cdot (f_k(q_j))^2 - b \geq -0.5, (j = 1 : m) \quad (2)$$

$$x_i, a \geq 0 (i = 1 : N) \quad (3)$$

Оптимизация проводится по трем различным критериям:

$$\sum_{j=1}^m |Z_j(\mathbf{x}, a, b, A)| \xrightarrow{\mathbf{x}, a, b, A} \min - L_1: \text{минимизация суммы абсолютных ошибок}, \quad (4)$$

$$\sum_{j=1}^m (Z_j(\mathbf{x}, a, b, A))^2 \xrightarrow{\mathbf{x}, a, b, A} \min - L_2: \text{минимизация суммы квадратов ошибок}, \quad (5)$$

$$\max_{j=1:m} |Z_j(\mathbf{x}, a, b, A)| \xrightarrow{\mathbf{x}, a, b, A} \min - L_{\text{inf}}: \text{минимизация максимальной абсолютной ошибки}. \quad (6)$$

Для случаев L_1 и L_{inf} был использован свободно доступный ЛП-солвер LP_SOLVE, (<http://lpsolve.sourceforge.net>). Для случая L_2 был использован свободно доступный пакет нелинейной оптимизации Ipopt (<http://projects.coin-or.org/Ipopt>).

Следует отметить, что решение, получаемое в результате решения задач (4-6) может быть неединственным. Во избежание этого эффекта можно, либо вводить дополнительное регуляризующее слагаемое в критерии (4-6) (например, $\varepsilon \cdot \sum_{i=1}^N (x_i)^2 + a^2$, с малым коэффициентом ε), либо разумно подходить к выбору шага по параметрам структур, не делая его излишне малым.

Необходимость в массивованных вычислениях с последующим применением достаточно производительных пакетов оптимизации для решения задач идентификации, делает целесообразным перенос вычислений в распределенную вычислительную среду. Для проведения в будущем многовариантных расчетов и предоставления широкого доступа к созданному сценарию численного анализа результатов физического эксперимента, такой подход был реализован нами в системе RESTful-сервисов MathCloud (www.mathcloud.org) [2], [3], [4].

3. Общие сведения об инструментарии MathCloud

Для автоматизации изложенной выше многоэтапной методики оптимизационной идентификации углеродистых наноструктур, был использован программный инструментарий MathCloud [2]-[4], [6]. Этот комплект ПО разрабатывается в Центре Грид-технологий и распределенных вычислений Института системного анализа РАН с целью предоставить в распоряжение разработчиков и пользователей распределенных вычислительных систем удобные в использовании и нетребовательные к ресурсам средства:

- удобного преобразования существующих приложений (в том числе, Грид-приложений) в удаленно доступные, в т.ч. программным способом, сервисы;
- интеграции сервисов в сложносоставные вычислительные сценарии, которые становятся новыми, т.н. «композитивными» сервисами, средствами «визуального программирования»;
- автоматического управления выполнением созданных сценариев, в условиях, когда процесс вычислений может занимать продолжительное время.

Уже созданные элементы ПО свободно доступны, на условиях лицензии Apache License, Version 2.0, <https://www.assembla.com/spaces/mathcloud/documents>. Публикации с подробным описанием принципов работы и применяемых программных технологий (здесь можно выделить работу [3]) свободно доступны на том же сайте, <https://www.assembla.com/wiki/show/mathcloud/Publications>. Не вдаваясь в детали, можно сказать, что инструментарий MathCloud является одним из примеров ПО, позволяющих на практике реализовать модель «научной кооперации в стиле Веб 2.0». Этот подход, по сути, означает, что сообщество пользователей само создает «полезное содержание системы», (или, если использовать «продвинутый», но неточный, термин - «контент») которым, на условиях некоторых дополнительных соглашений, пользуются остальные участники сообщества. В контексте организации совместной научной работы это означает, что исследователи могут сами создавать веб-сервисы, основанные на имеющихся в их распоряжении приложениях. В частности, если это уникальные реализации численных процедур обработки данных, программные модели различных систем и т.п., то речь может идти о новой форме «публикации» части своих результатов в форме *удаленно доступных сервисов*.

Очень часто, «веб-сервисами» называют только веб-приложения, рассчитанные на их использование пользователем-оператором в браузере. Не вдаваясь в философские споры, отметим, браузер – всего лишь (хотя и очень широко распространенное) клиентское приложение, которое взаимодействует с веб-приложением, посредством обмена сообщениями по протоколу HTTP. Это важное обстоятельство позволяет создавать различные клиентские приложения, которые взаимодействуют с веб-приложениями в автоматическом режиме (самый простой пример – программы-краулеры, просматривающие содержимое сайтов для поисковых систем типа Яндекс или Google).

Инструментарий MathCloud рассчитан на то, что создаваемый сервис может быть использован как в «ручном», так и в «автоматическом» режиме. В первом случае - для однократной обработки данных, отправляемых пользователем сервису через браузер. Второй случай соответствует *составным* («композитивным») вычислительным сценариям, где требуется координированная обработка данных на различных сервисах, выполняемая согласно некоторой схеме вычислений. Здесь сервисы взаимодействуют между собой как приложения, без участия оператора, который лишь запускает сценарий, может (но не обязан)

контролировать ход его выполнения и получает доступ к результатам вычислений (в случае их успешного завершения).

На Рис. 1 представлен общий взгляд на «уровни ответственности» архитектуры MathCloud (обведены красной рамкой). На самом нижнем уровне, за рамками MathCloud находятся уже неоднократно упоминаемые «существующие приложения», изначально никак не рассчитанные на использование в MathCloud. Средствами MathCloud они преобразуются в сервисы. Сервисы группируются в разнообразные *приложения MathCloud*, фактически - в сценарии координированной обработки данных. На самом верхнем уровне находятся пользователи, которые используют эти приложения (включая изолированные сервисы).

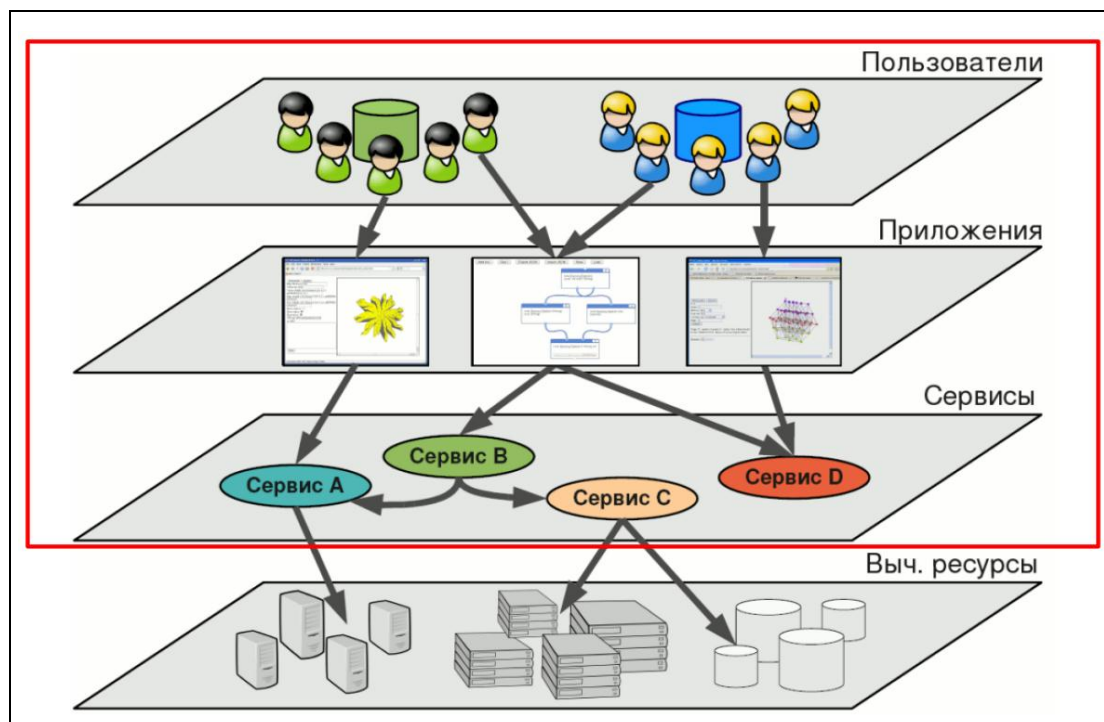


Рис. 1. Общий взгляд на архитектуру MathCloud.

В настоящее время инструментарий MathCloud предусматривает два способа преобразования существующих приложений в RESTful-сервисы: на основе отдельной программной компоненты, т.н. *контейнера* ресурсов Everest [3]; в форме CGI-сценария на языке Perl [6], реализующего MathCloud REST API, который может быть использован в любом Веб-сервере с поддержкой Perl CGI. Для данной работы использовался только контейнер Everest, см. Рис. 2. Этот программный компонент реализован на языке Java и Веб-сервере Jetty. Комплект разработчика содержит специальные модули (plugins), облегчающие подключение приложений трех типов: 1) реализованных (надлежащим образом) в виде набора Java-модулей, а также, на Jython - Java-реализации интерпретатора языка Python; 2) уже используемых в режиме командной строки; 3) уже существующих Грид-приложений, реализованных согласно стандартам Грид-инфраструктуры EGI, European Grid Infrastructure, на основе промежуточного ПО Грид gLite.

Взаимодействие с сервисами по сети осуществляется в форме обмена сообщениями по протоколу HTTP благодаря встроенному в контейнер веб-серверу. В случае изолированного использования в браузере сервисы возвращают сообщения в формате HTML (примеры см. ниже). В автоматическом режиме сервисы обмениваются сообщениями в текстовом формате

JSON (JavaScript Object Notation). Входящее сообщение содержит данные, отправляемые на обработку, включая URL-ссылки на файлы для обмена большими объемами данных. При получении входящего запроса создается *задание*, которое ставится в очередь на выполнение, а клиентскому приложению возвращается уникальный *идентификатор задания* в виде URL. При повторном обращении по этому URL контейнер возвращает сообщение, содержащее *статус* задания (WAITING, RUNNING, DONE, ERROR) и некоторую дополнительную информацию. Если задание выполнено (статус DONE), то возвращаемое сообщение содержит выходные параметры с результатами выполнения задания, включая (если это предусмотрено) URL-ссылки на файлы с результатами, которые сохраняются контейнером в создаваемых автоматически временных папках. Такой асинхронный стиль взаимодействия значительно упрощает выполнение трудоемких заданий и сценариев, когда время обработки данных заранее неизвестно.

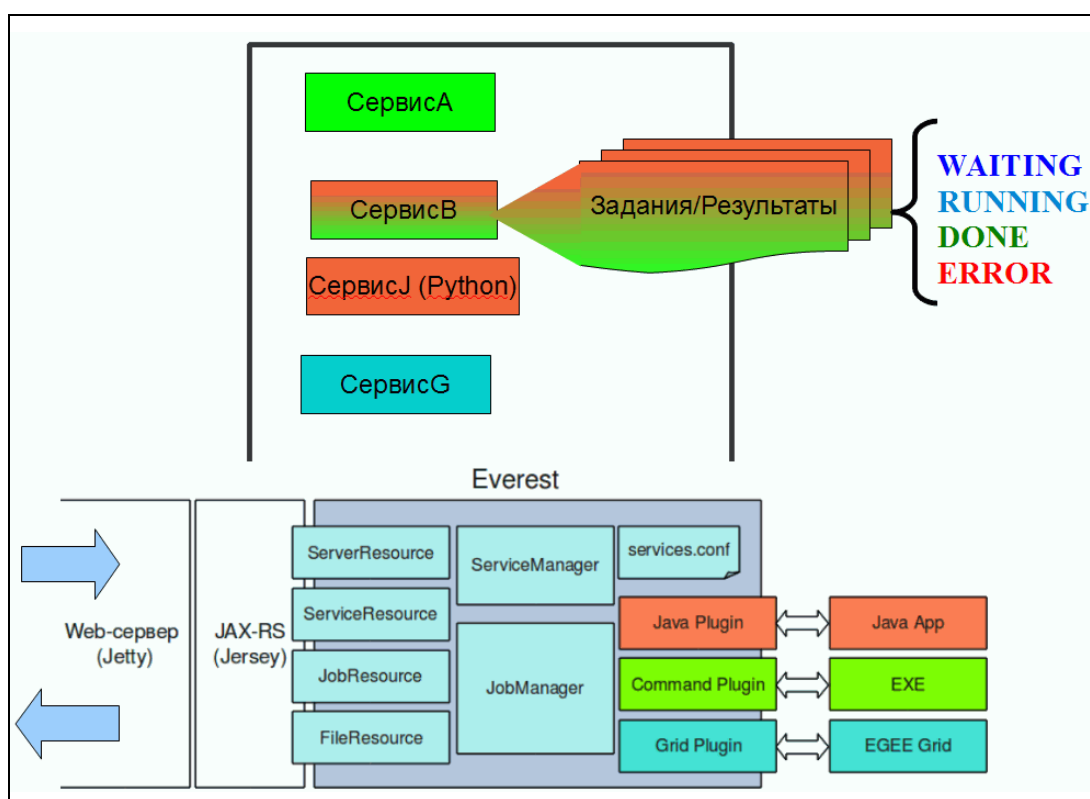


Рис. 2. Контейнер сервисов MathCloud.

За создание, сохранение и управление ходом выполнения составных сценариев отвечает специальный компонент СУС, Служба Управления Сценариями, Рис. 3. Этот компонент основан на том же контейнере Everest и дополнительно включает средства хранения и управления сценариями, в т.ч. специальный визуальный редактор для создания сценариев. Этот редактор реализован в виде библиотек на JavaScript, загружаемых в браузер пользователя системы. Служба хранит сценарии в виде текстовых файлов в формате JSON. Каждый сценарий, в свою очередь, является сервисом и в дальнейшем может использоваться точно так же, как обычный веб-сервис MathCloud.

При получении команды запуска сценария, система сама начинает его автоматическое выполнение, определяя очередность вызова сервисов, включенных в сценарий, согласно *информационным зависимостям* (где выходные данные одного или нескольких сервисов

являются входом для других сервисов), установленным при создании сценария. Важной особенностью СУС является то, что для «правильно» спроектированных сценариев относительно небольшой объем данных проходит по сети от сервисов с СУС и обратно (пунктирные стрелки на Рис. 3). Если некоторый сервис является источником большого объема данных, передаваемых в виде файла, то сервис пересылает СУС только строку с URL этого файла. Эта ссылка передается сервису-получателю данных, который запрашивает их непосредственно у соответствующего контейнера (жирные стрелки в «облаке сервисов»).

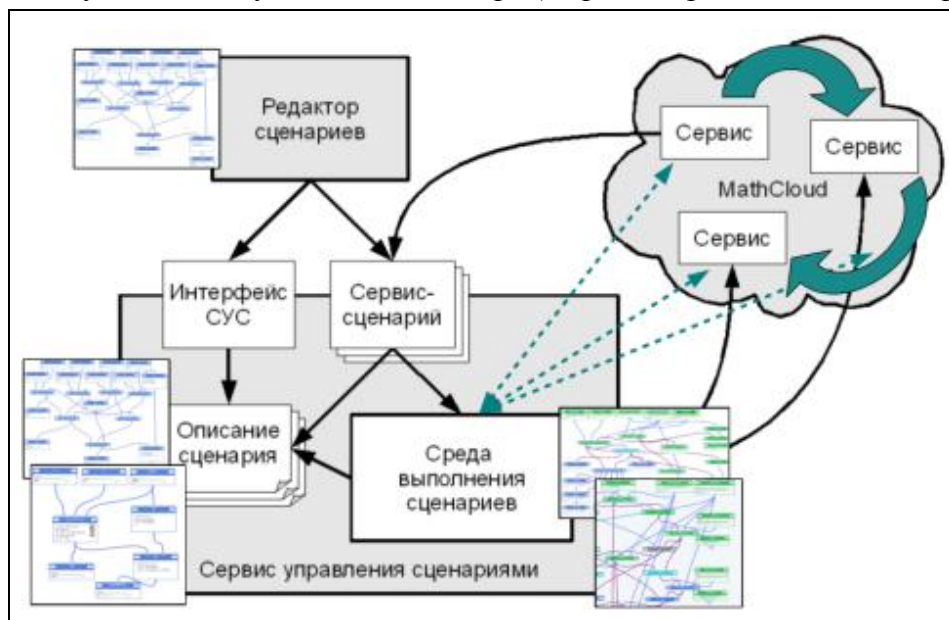


Рис. 3. Служба управления сценариями.

4. Реализация системы анализа экспериментальных данных в MathCloud

До применения MathCloud, обсуждаемый сценарий идентификации углеродистых структур по рентгеновским дифракционным характеристикам был основан на нескольких приложениях, уже упомянутых в разделе 3: грид-приложение, для массивованного вычисления модельных кривых рассеяния; shell-скрипта для вызова пакетов оптимизации, которым передаются (в виде текстовых файлов в формате AMPL) символьное описание оптимизационных задач, данных экспериментальной и модельных кривых рассеяния; нескольких вспомогательных утилит на языке Python (с использованием модулей NumPy, SciPy и Matplotlib) для преобразования текстовых файлов и построения графиков. Вызов всех приложений, кроме «гридовского», происходит в виде командной строки.

Для создания соответствующих сервисов MathCloud требовалось лишь составить специальные JSON-дескрипторы для каждого из указанных приложений. Например, дескриптор для одной из утилит (plotXRD_web.py, на Python), строящей графики с модельными кривыми рассеяния приведен на Рис. 4. В разделах *inputs* и *outputs* указаны входные и выходные параметры. В разделе *implementation*: тип приложения для контейнера («command») и правил соответствия формальных параметров разделов *inputs* и *outputs* фактическим именам файлов и потокам *stdout*, *stderr* консоли Linux. Красным цветом выделена командная строка вызова утилиты. Дескриптор грид-приложения приведен на Рис. 5.

```

name : XRD_Plotter_mini,
description : "Plots the results of x-ray diffraction modelling",
inputs : {
    start : {type:file, title:"Config xml",
             description: "Single xml file or tar.gz archive"},
    results : {type:file, title:"XRD results",
              description: "XRD modelling results (tar.gz archive)"},
    CL : {type:string,title:"Command line options",
         description: "Add parameters here.", optional: true}
},
outputs : {
    pictures : {type:file, title:"Pictures"},
    out : {type:file,title:"Output"},
    err : {type:file,title:"Error"}
},
implementation : {
    type : command,
    command : "python /home/vneverov/XRDnano/plotXRD_web.py $CL",
    input2file:{
        start:start,
        results:results.tar.gz
    },
    file2output:{
        pictures.tar.gz:pictures,
        stdout:out,
        stderr:err
    }
}

```

Рис. 4. Пример дескриптора утилиты на языке Python.

```

name : XRD_Calculation,
description : "X-ray diffraction modeling (calc. in GRID)",
inputs : {
    N : {type: integer, title:"Number of calculations"},
    start : {type:file, title:"Config xml",
            description: "Single xml file or tar.gz archive"},
    structures : {type:file, title:"Structures archive",
                description: "tar.gz archive"}
},
outputs : {
    results : {type:file, title:"Results"},
    out : {type:file,title:"Output"},
    err : {type:file,title:"Error"}
},
implementation : {
    type : grid,
    vo : gilda,
    job-file : "/home/vneverov/XRDnano/startMass.jdl",
    pre : "python /home/vneverov/XRDnano/preXRD_calc.py",
    post : "/bin/bash /home/vneverov/XRDnano/postXRD_calc.sh",
    input2file:{
        start:start,
        structures:structures.tar.gz
    },
    file2output:{
        results.tar.gz:results,
        stdout.tar.gz:out,
        stderr.tar.gz:err
    }
}

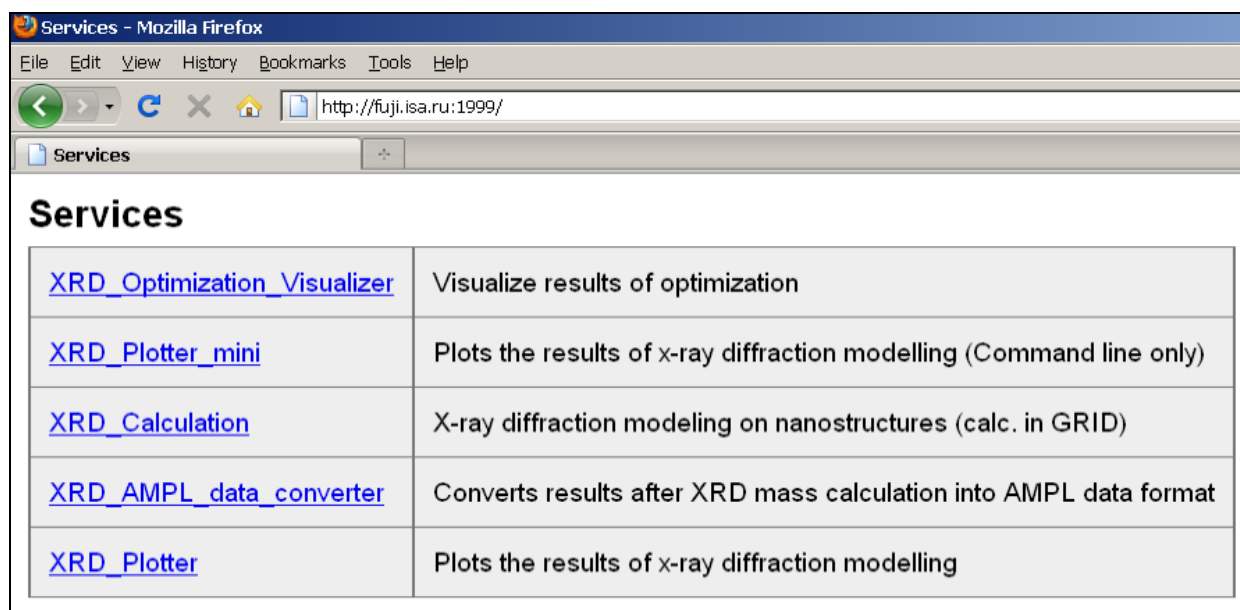
```

Рис. 5. Дескриптор грид-приложения.

Указанные дескрипторы заносятся в конфигурационный файл контейнера Everest, который автоматически создает соответствующие сервисы. Таким образом, преобразование

приложений в сервисы MathCloud не требует особых усилий и вполне доступно специалисту, который до этого либо создавал, либо уже пользовался этими приложениями.

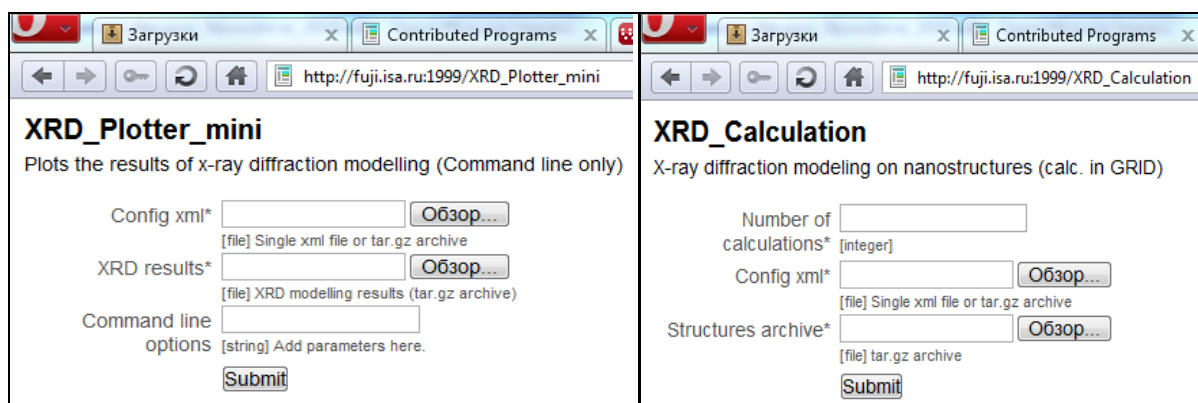
Запущенный MathCloud-контейнер, при обращении к нему в браузере, возвращает HTML с таблицей установленных на нем сервисов, Рис. 6.



Services	
XRD_Optimization_Visualizer	Visualize results of optimization
XRD_Plotter_mini	Plots the results of x-ray diffraction modelling (Command line only)
XRD_Calculation	X-ray diffraction modeling on nanostructures (calc. in GRID)
XRD_AMPL_data_converter	Converts results after XRD mass calculation into AMPL data format
XRD_Plotter	Plots the results of x-ray diffraction modelling

Рис. 6. Список сервисов на одном из контейнеров MathCloud.

Как уже отмечалось ранее, каждым из сервисов можно пользоваться изолированно в браузере через веб-форму, которая автоматически создается контейнером при обращении по URL-ссылке сервиса, Рис. 7.



XRD_Plotter_mini
Plots the results of x-ray diffraction modelling (Command line only)

Config xml*
[file] Single xml file or tar.gz archive

XRD results*
[file] XRD modelling results (tar.gz archive)

Command line options
[string] Add parameters here.

XRD_Calculation
X-ray diffraction modeling on nanostructures (calc. in GRID)

Number of calculations*
[integer]

Config xml*
[file] Single xml file or tar.gz archive

Structures archive*
[file] tar.gz archive

Рис. 7. Веб-формы для изолированного использования сервисов в браузере.

Ниже, на Рис. 8, приведена реализация сценария автоматического выполнения всего цикла вычислений в визуальном редакторе MathCloud. Этот сценарий фактически является приложением MathCloud. Каждый блок здесь соответствует либо входным и выходным параметрам, либо сервисам MathCloud. Входные параметры отображаются на блоках верхними «контактами», выходные – нижними. Эти «контакты» соединены «проводами» информационных зависимостей типа «поставщик-получатель» данных. Одни и те же данные могут отправляться нескольким получателям. Система сама будет отслеживать порядок вызова сервисов согласно установленным информационным зависимостям.

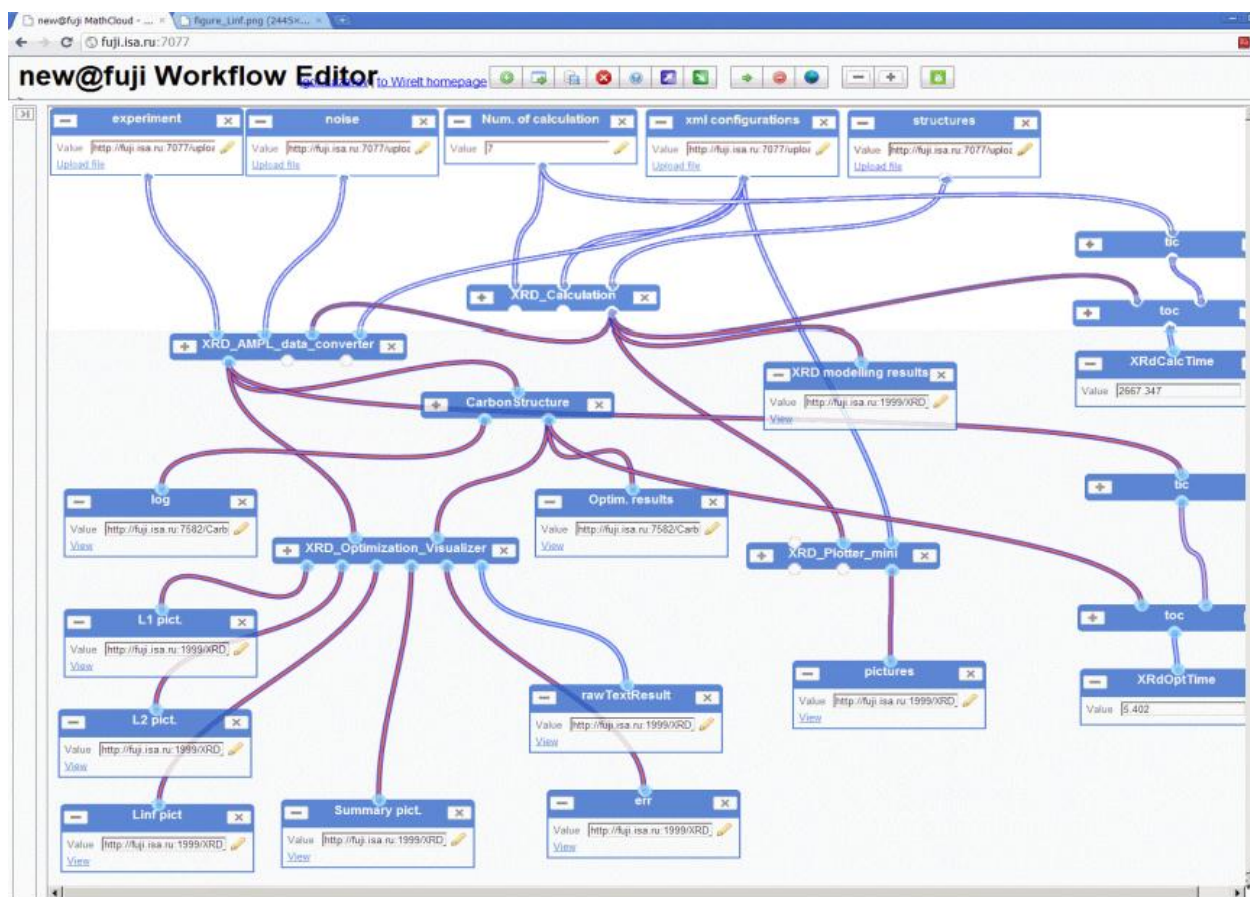


Рис. 8. Система структурного анализа в MathCloud.

Созданное приложение, в свою очередь, является составным сервисом, который можно использовать через автоматически созданную веб-форму, Рис. 9. Этой формой можно пользоваться для проведения многовариантных расчетов, подавая на вход приложения различные сочетания параметров моделирования и экспериментальных данных.

Загрузив параметры задания в поля формы, Рис. 9, и нажав кнопку Submit, пользователь получает в ответ форму с уникальным URL задания (в адресной строке браузера), позволяющую наблюдать за ходом вычислений, см. Рис. 10. Периодически заходя по этому адресу, пользователь может отслеживать статус задания. Если задание выполнено, то пользователь получит в браузере итоговую таблицу, Рис. 11, в частности содержащую ссылки на файлы с результатами вычислений, предусмотренных в сценарии. Автору сценария могут потребоваться и другие данные. Например, на Рис. 11 можно видеть (в поле «XRdCalcTime») продолжительность выполнения грид-задания (расчет модельных кривых рассеяния сервисом «XRD_Calculation», Рис. 7) и продолжительность решения оптимизационных задач (в поле XRdOptTime) сервисом «CarbonStructure».

Для подобных замеров в сценарий добавлены блоки «псевдо-сервисов» в виде несложных процедур на JavaScript. На Рис. 8 эти процедуры скрыты в блоках «tic», «toc», которые в развернутом виде представлены на Рис. 12. Здесь (см. Рис. 8) провод, входящий в блок «tic», проведен из сервиса «XRD_AMPL_data_converter» (готовящего входные данные для сервиса оптимизации), а левый провод, входящий в блок «toc» - из блока оптимизации CarbonStructure. Таким образом, блок «tic» «сработает» перед самым началом вызова сервиса оптимизации, а блок «toc» - сразу после завершения оптимизации.

Neverov-CarbonStructuresIdentification - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://fuji.isa.ru:7077/services/649acf03-f26e-49b5-93cd-c3b1aa06d32c

Neverov-CarbonStructuresIden...

Neverov-CarbonStructuresIdentification

Config xml* Browse...
[file] Single xml file or tar.gz archive

Number of calculations*
[integer]

Structures archive* Browse...
[file] tar.gz archive

Experimental profile* Browse...
[file]

Noise profile* Browse...
[file]

Рис. 9. Веб-форма системы структурного анализа в MathCloud.

Neverov-CarbonStructuresIdentification - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://fuji.isa.ru:7077/services/649acf03-f26e-49b5-93cd-c3b1aa06d32c/job6312603119259612568

Neverov-CarbonStructuresIden...

State: RUNNING

Info: Workflow state: RUNNING

Result:

workflowId: 649acf03-f26e-49b5-93cd-c3b1aa06d32c

start-time: 13 Nov 2010 18:09:27 416

tasks: [{"name":"task0 (XRD_Plotter_mini)",
"state":"WAITING"},
{"name":"task2 (XRD_Calculation)",
"state":"RUNNING",
"info":"Service request state: RUNNING. Info: Submitting job to grid",
"start-time":"13 Nov 2010 18:09:27 716"},
{"name":"task6 (XRD_AMPL_data_converter)",
"state":"WAITING"},
{"name":"task9 (CarbonStructure)",
"state":"WAITING"},
{"name":"task10 (XRD_Optimization_Visualizer)",
"state":"WAITING"},
{"name":"task20 (tic)",
"state":"DONE",
"start-time":"13 Nov 2010 18:09:27 717",
"finish-time":"13 Nov 2010 18:09:27 720"},
{"name":"task21 (toc)"}]

Рис. 10. Наблюдение за ходом выполнения сценария на странице задания.

Наблюдать за ходом выполнения сценария можно и непосредственно в окне визуального редактора в браузере. Например, на Рис. 13 приведен скриншот окна редактора во время выполнения сценария определения наноструктурного состава углеродного образца по его дифракционным характеристикам оптимизационными методами. Уже завершённые блоки сценария выделены зеленым (статус DONE). Те, которые в данный момент выполняются (статус RUNNING), выделены светло-голубым цветом. Ожидающие выполнения – синим. На этом рисунке дополнительно размещены условные обозначения типа хостов, где происходит фактическая работа приложений («кластеры» для грид-приложения и «небольшие серверы» – для всех остальных).

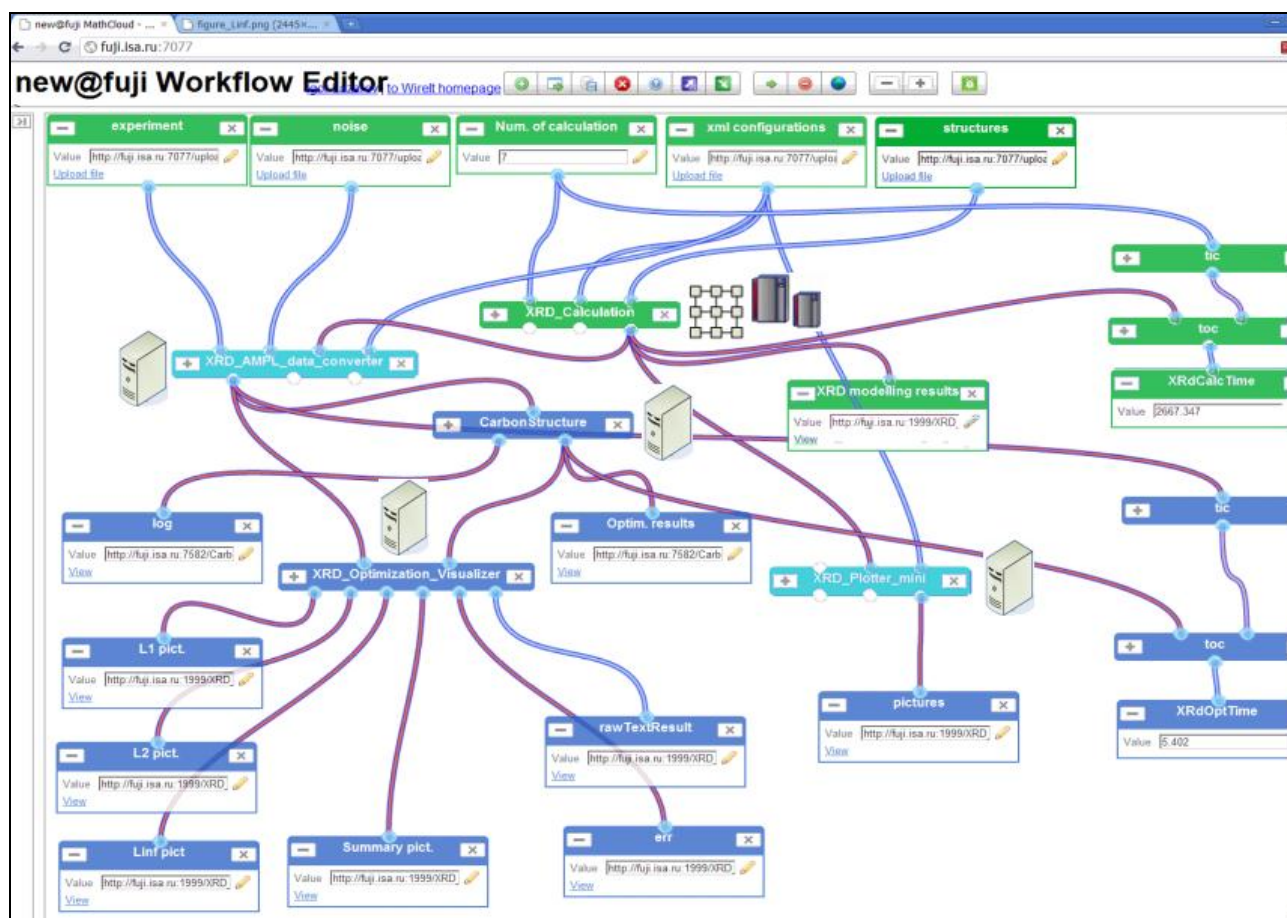


Рис. 13. Наблюдение за ходом выполнения сценария в редакторе.

По завершению вычислений все блоки приобретают зеленый цвет, и пользователь получает доступ ко всем значениям выходных параметров. В рассматриваемом случае – к текстовым и графическим файлам.

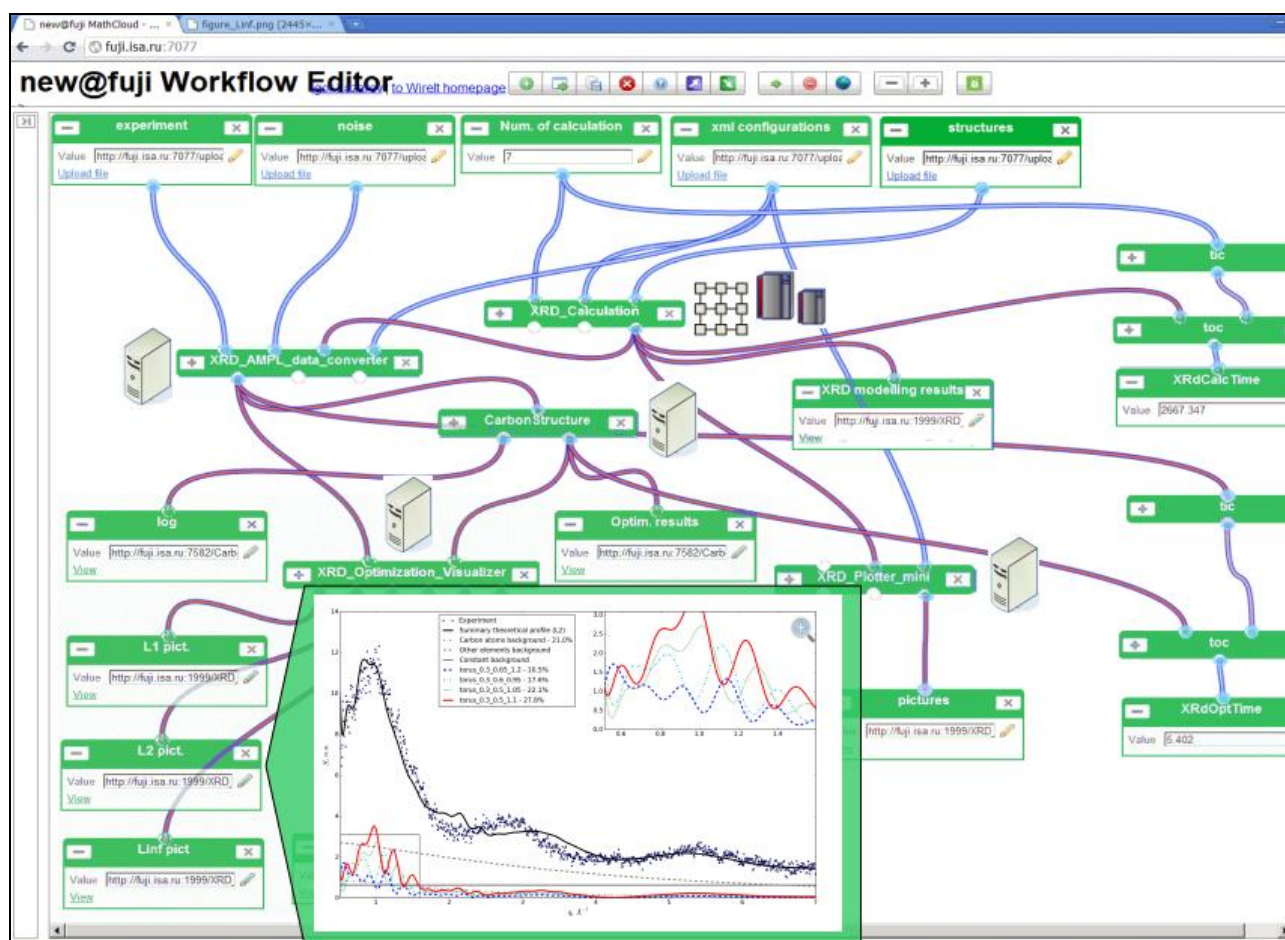


Рис. 14. Завершенный сценарий.

Закключение

Реализация вычислительного сценария идентификации углеродистых структур по рентгеновским дифракционным характеристикам наноматериалов в MathCloud, как мы надеемся, позволит в будущем сторонним исследователям использовать наш подход при обработке соответствующих экспериментальных данных.

Имеющиеся средства инструментария MathCloud для модификации вычислительных сценариев позволяют потенциальным пользователям легко совершенствовать и дополнять созданное распределённое приложение.

ЛИТЕРАТУРА

- [1] В.С. Неверов, А.Б. Кукушкин, Н.Л. Марусов, И.Б. Семёнов, В.В. Волошинов, А.П. Афанасьев, А.С. Тарасов, А.А. Велигжанин, Я.В. Зубавичус, Н.Ю. Свечников, В.Г. Станкевич, Моделирование рентгеновской дифракции на углеродных наноструктурах и определение их возможного топологического состава в осажденных пленках из токамака Т-10, Вопросы атомной науки и техники, Сер. Термоядерный синтез, 2010, том. 1, стр. 7-21.
- [2] Астафьев А.С., Афанасьев А.П., Лазарев И.В., Сухорослов О.В., Тарасов А.С. 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. С. 463-467

[3] Сухорослов О.В. Унифицированный интерфейс доступа к алгоритмическим сервисам в Web. // Проблемы вычислений в распределенной среде / Под ред. С.В. Емельянова, А.П. Афанасьева. Труды ИСА РАН, Т. 46. - М.: КРАСАНД, 2009. С. 60-82

[4] Лазарев И.В., Сухорослов О.В. Реализация распределенных вычислительных сценариев в среде MathCloud. // Проблемы вычислений в распределенной среде / Под ред. С.В. Емельянова, А.П. Афанасьева. Труды ИСА РАН, Т. 46. - М.: КРАСАНД, 2009. С. 6-23

[5] N.Yu. Svechnikov, V.G. Stankevich, K.A. Men'shikov, et al., J. Surf. Invest., 2 (2008) 826-835

[6] Астафьев А.С. Унифицированная оболочка для доступа к ресурсам распределенной математической среды // Труды 52-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук». Часть IX. Инновации и высокие технологии. - М.: МФТИ, 2009. С. 15-17

ⁱАвторы выражают благодарность за помощь в работе:

А.Б. Кукушкину, В.А. Вознесенскому, Н.Л. Марусову, И.Б. Семенову, О.В. Сухорослову, А.А. Велигжанину, Я.В. Зубавичусу, Н.Ю. Свечникову и В.Г. Станкевичу.

Работа поддержана РФФИ (проекты 09-07-00469, 08-07-00430-а) и Европейским проектом EGEE-III (Enabling Grids for E-sciencE, "Развёртывание Гридов для развития вычислительноёмкой науки")