

Обработка данных рентгеновской дифрактометрии наноматериалов в распределенной среде rest-сервисов¹

В.В. Волошинов, В.С. Неверов

Аннотация. Описывается распределенное приложение для обработки экспериментальных данных рентгеновской дифрактометрии углеродистых наноматериалов в среде REST-сервисов MathCloud, включающее массовые расчеты кривых рентгеновского рассеяния отдельными наноструктурами с последующим решением задач оптимизационной идентификации наиболее вероятного распределения наноструктур по топологии и размеру.

Ключевые слова: REST-сервисы, распределенные вычисления, оптимизация, обработка экспериментальных данных, наноматериалы, рентгеновская дифракция.

Введение

Для создания новых наноматериалов необходимо контролировать их характеристики в нанометровом диапазоне размеров. Одним из наиболее эффективных инструментов, способных решать эту задачу, является рентгеновская дифрактометрия в субнанометровом диапазоне длин волн. Между тем, анализ результатов рентгеновского рассеяния на наноматериалах, для которых пространственное упорядочение структуры сохраняется лишь на нескольких нанометрах, имеет ряд сложностей, связанных с существенным отклонением от поликристаллической структуризации. Широкие пики на получаемой в таких случаях кривой рентгеновского рассеяния нельзя трактовать как брегговские, вследствие чего дифракционный профиль содержит недостаточно информации для восстановления структуры сложного наноматериала. Стандартные методы кристаллографии, которые в первую очередь заключаются в восстановлении распределения электронной

плотности в среде методами Фурье-анализа, не дают исчерпывающей информации об объекте. Получить достаточно надежный результат, основываясь только на результатах рентгеновской дифракции, оказывается затруднительным. Мы предлагаем решать обратную задачу, а именно, восстанавливать структурный (пространственный и топологический) состав исследуемых материалов в нанометровом диапазоне размеров, во-первых, значительно сужая класс возможных решений за счет использования результатов других диагностических методов, и, во-вторых, используя численное моделирование рентгеновского рассеяния на возможном классе решений (нанообъектов) для сопоставления расчетных данных с экспериментом. Принципиальная возможность ограничения класса возможных решений является необходимым условием применимости излагаемого ниже метода, использованного в работах [1-3] для задачи определения возможного топологического состава углеродных наноструктур в углеродистых пленках, образующихся внутри вакуумной ка-

¹ Работа поддержана РФФИ (проекты 09-07-00469, 11-07-00543-а) и при поддержке ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы» (Госконтракт № 07.514.11.4024).

меры токамака Т-10, по результатам их рентгеновской дифрактометрии.

1. Методика численного анализа экспериментальных данных

Предлагаемая методика 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 или отдельная наноструктура). Элементарный блок может иметь количество копий, ограниченное только вычислительными мощностями и организованное в регулярные (кристаллы) и нерегулярные структуры. Нерегулярные структуры могут включать в себя регулярные и наоборот.

На Выходе. Одномерный профиль интенсивности рассеяния рентгеновского излучения, рассчитанный в приближении изотропного распределения наноструктур по их ориентации (что позволяет использовать известную формулу Дебая), с возможностью вывода вкладов от интерференции различных типов наноструктур. Двумерная дифракционная картина с возможностью усреднения по пространственным углам волнового вектора падающего излучения.

Интерфейс. Простой XML интерфейс, удобный для автоматизированной генерации начальных данных, необходимой при массовых расчетах, например, в Гриде. Для форирования 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.

Структурный состав определяется по методу оптимизационной идентификации [1-3], основанной на минимизации величин (1) расхождения между экспериментальной и модельной интенсивностями рассеяния, где последняя вычисляется как сумма «вкладов» (с неизвестными коэффициентами x_i ($i = 1 : N$), a , b , A) от однородных аморфных смесей углеродных структур, свободных атомов углерода, примесей и некоторого «фона»:

$$Z_j(\mathbf{x}, a, b, A) \doteq S_{\text{exp}}(q_j) - \sum_{i=1}^N S_i(q_j) \cdot x_i - a \cdot S_C^{\text{amorph}}(q_j) - A \cdot S_{\text{impur}}^{\text{amorph}}(q_j) - b \quad (j = 1:m); \quad (1)$$

где $S_{\text{exp}}(q_j)$ — экспериментальный профиль интенсивности рентгеновского рассеяния; $S_i(q_j)$ — модельный профиль интенсивности рентгеновского рассеяния на i -й углеродной наноструктуре, делённый на число атомов в ней; j — номер точки в дискретном пространстве модуля вектора рассеяния q ; x_i/A — вероятность принадлежности атома углерода i -й наноструктуре; N — полное число выбранных наноструктур; a/A — вероятность атома углерода принадлежать аморфной среде; $S_C^{\text{amorph}}(q_j)$ — модельный профиль интенсивности аморфной углеродной среды заданной плотности, приходящийся на один атом углерода; $S_{\text{impur}}^{\text{amorph}}(q_j)$ — суммарный модельный профиль интенсивности от всех возможных примесей (процентное содержание которых в образце должно быть известно из результатов других диагностик образца) для заданной плотности примесей в образце, делённый на интенсивность рассеяния уединённым атомом углерода; b — неизвестная величина возможного постоянного фона; m — число значений модуля вектора рассеяния q в наборе экспериментальных данных.

Дополнительные условия:

$$S_{\text{exp}}(q_j) - a \cdot S_C^{\text{amorph}}(q_j) - A \cdot S_{\text{impur}}^{\text{amorph}}(q_j) - b \geq -0,5 \quad (j = 1:m) \quad (2)$$

$$\sum_{i=1}^N x_i + a = A, \quad x_i \geq 0 \quad (i = 1 : N), \quad a \geq 0 \quad (3)$$

Условие (2) является ограничением на постоянный фон b , а величина 0,5 отвечает учёту средней ошибки экспериментальных данных. Предусмотрены также возможности вручную задавать границы для величины постоянного фона и исключать из оптимизации профили $S_C^{\text{amorph}}(q_j)$ и $S_{\text{impur}}^{\text{amorph}}(q_j)$. В подобной постановке оптимизационная задача может охватывать довольно широкий круг проблем анализа экспериментальных данных рентгеновского рассеяния.

Оптимизация проводится по трем различным критериям (при ограничениях (2, 3)):

$$\sum_{j=1}^m |Z_j(\mathbf{x}, a, b, A)| \xrightarrow{\mathbf{x}, a, b, A} \min \quad (4)$$

L_1 : минимизация суммы абсолютных ошибок,

$$\sum_{j=1}^m (Z_j(\mathbf{x}, a, b, A))^2 \xrightarrow{\mathbf{x}, a, b, A} \min \quad (5)$$

L_2 : минимизация суммы квадратов ошибок,

$$\max_{j=1:m} |Z_j(\mathbf{x}, a, b, A)| \xrightarrow{\mathbf{x}, a, b, A} \min \quad (6)$$

L_{inf} : минимизация максимальной абсолютной ошибки. 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 задач (4-6) может быть не единственным. Во избежание этого эффекта можно либо вводить дополнительное регуляризирующее слагаемое в критерии (4-6) (например, $\varepsilon \cdot \left(\sum_{i=1}^N (x_i)^2 + a^2\right)$, с малым коэффициентом ε , превращая (4-6) в задачи квадратичного программирования), либо разумно подходить к выбору шага по параметрам структур, не делая его излишне малым.

Необходимость в массивированных 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, был реализован подход в

системе RESTful-сервисов MathCloud (www.mathcloud.org) [4-6].

2. Возможности MathCloud

Для автоматизации изложенной выше многоэтапной методики оптимизационной идентификации углеродистых наноструктур был использован программный инструмент MathCloud [4-7]. 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:

- 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, так называемыми «компонентными» сервисами, средствами «визуального программирования»;
- автоматического управления выполнением созданных сценариев в условиях, когда процесс вычислений может занимать продолжительное время.

Уже созданные элементы ПО вместе с документацией свободно доступны, на условиях лицензии Apache License, на Version 2.0, <http://code.google.com/p/websolve>.

Инструментарий MathCloud рассчитан на то, что создаваемый сервис может быть использован как в «ручном», так и в составе «вычислительных сценариев». В первом случае - для однократной обработки данных, отправляемых пользователем сервису через браузер. Второй случай соответствует ситуации, когда требуется координированная обработка данных различными сервисами согласно некоторой схеме вычислений.

На Рис.1 представлен общий взгляд на «уровни ответственности» архитектуры MathCloud (обведены рамкой). На самом нижнем уровне, за рамками MathCloud находятся уже «существующие приложения», изначально никак не рассчитанные на использование в MathCloud. Средств-

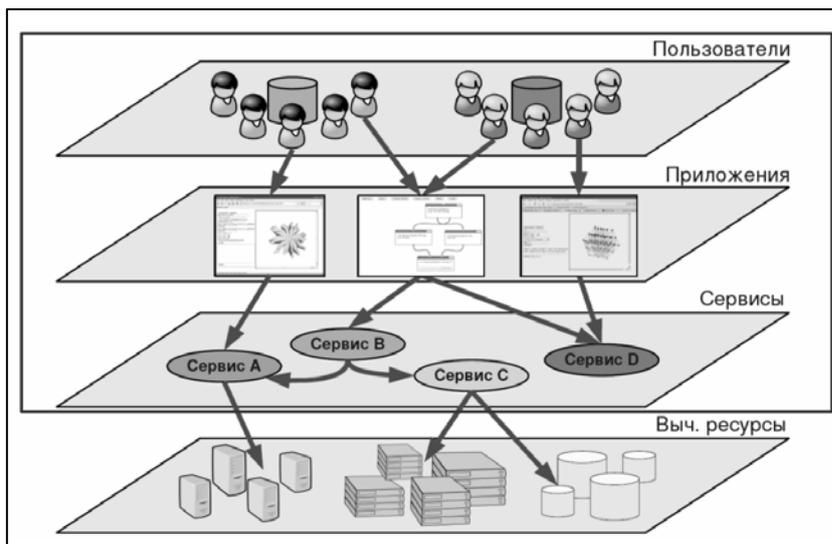


Рис. 1. Общий взгляд на архитектуру MathCloud

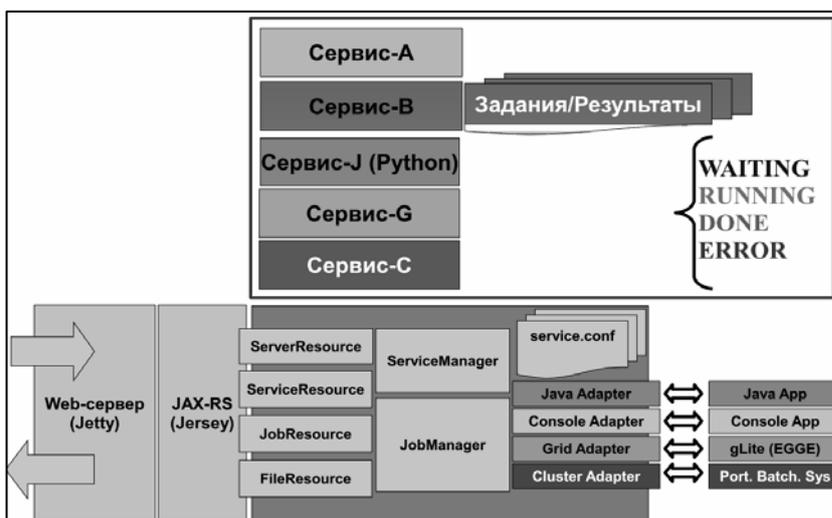


Рис. 2. Архитектура контейнера сервисов Everest MathCloud

вами 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).

В настоящее время инструментарий MathCloud предусматривает два способа преобразования существующих приложений в RESTful-сервисы: на основе отдельной программной компоненты, т.н. контейнера ресурсов Everest [5]; и в форме CGI-сценария на языке Perl [7], реализующего MathCloud REST API, который может быть использован в любом Веб-

сервере с поддержкой Perl CGI. Для данной работы применялся только контейнер Everest (Рис.2). Этот программный компонент реализован на языке Java. Комплект разработчика содержит специальные модули (plugins), обеспечивающие подключение уже существующих приложений четырех типов:

- реализованных (надлежащим образом) в виде набора Java-модулей, а также, на Jython - Java-реализации интерпретатора языка Python (Java Adapter на Рис.2);
- консольных приложений (Console Adapter);
- уже существующих Грид-приложений, реализованных согласно стандартам Грид-инфраструктуры EGI, European Grid Infrastruc-

ture, на основе промежуточного ПО Грид gLite (Grid Adapter);

- консольных приложений для кластеров под управлением TORQUE Portable Batch System (Cluster Adapter).

Взаимодействие с сервисами 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. 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 (Рис.2). При повторном обращении по этому URL контейнер возвращает сообщение со *статусом* задания (WAITING, RUNNING, DONE, ERROR) и некоторой дополнительной информацией. 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 контроль за выполнением трудоемких заданий и сценариев, требующих продолжительного времени.

За создание, хранение и управление ходом выполнения составных сценариев отвечает специальный программный модуль СУС, Служба Управления Сценариями. Этот модуль 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. 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. Важная особенность СУС состоит в том, что для «правильно» спроектированных сценариев лишь относительно небольшой объем данных проходит по сети от сервисов с СУС и обратно. Если некоторый сервис является источником большого объема данных, передаваемых в виде файла, то этот сервис пересылает СУС только строку с URL этого файла. Этот URL передается сервису-получателю данных, который загружает их непосредственно с контейнера-источника.

3. Реализация системы анализа экспериментальных данных в MathCloud

До применения MathCloud вычислительный сценарий идентификации углеродистых структур по рентгеновским дифракционным характеристикам был основан на нескольких приложениях, упомянутых ранее: грид-приложения, для массового вычисления модельных кривых рассеяния; shell-скрипт для вызова пакетов оптимизации, которым передаются (в виде текстовых файлов в формате AMPL, A Modeling Language for Mathematical Programming, 0) 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. В исходном виде все приложения, кроме «гридовского», работали в режиме «командной строки» (в консоли Linux).

При создании соответствующих сервисов MathCloud требовалось лишь составить специальные JSON-дескрипторы для каждого из указанных приложений. На основе дескрипторов контейнер Everest автоматически создает соответствующие сервисы. Таким образом, преобразование приложений в сервисы MathCloud не

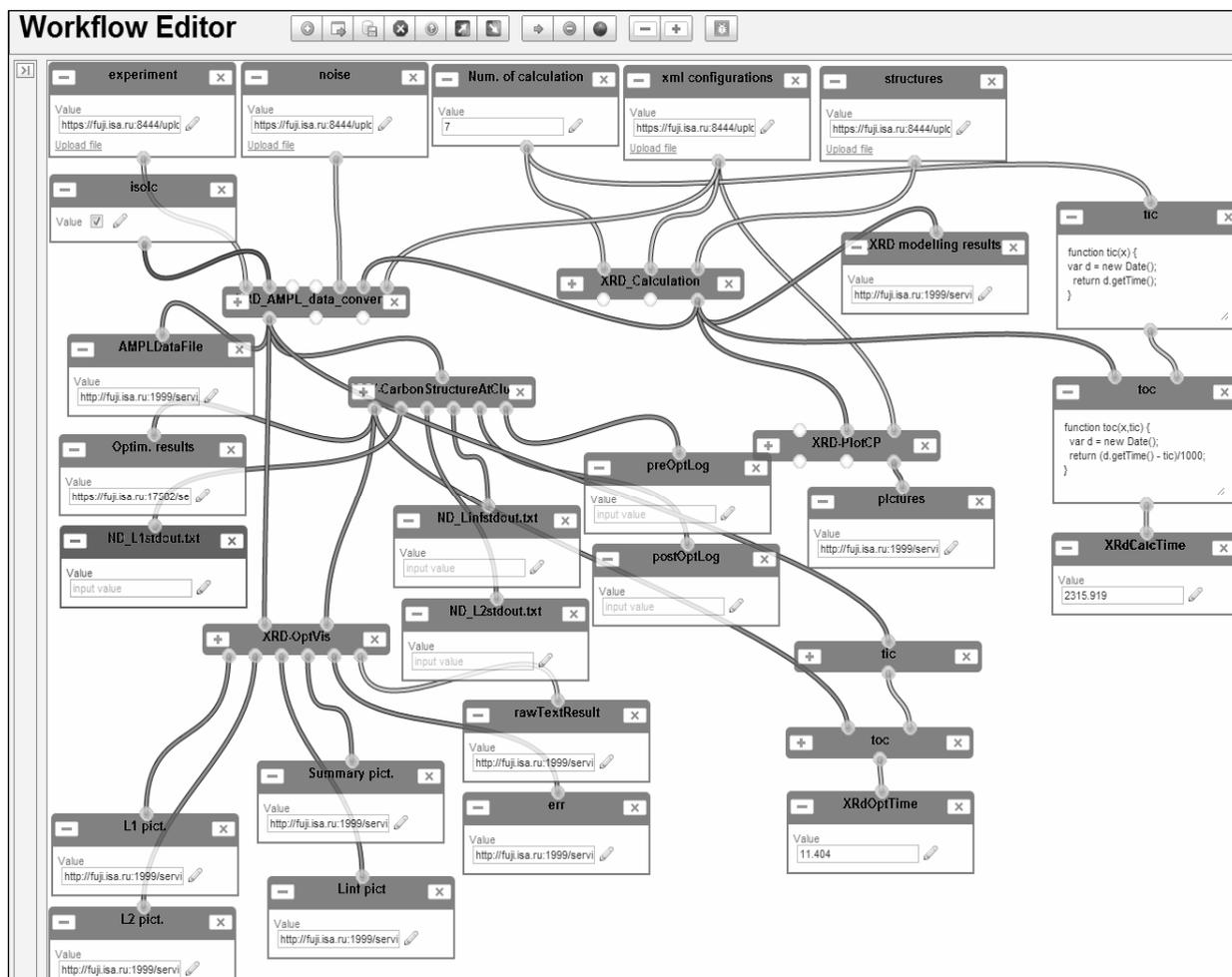


Рис. 3. Схема анализа данных рентгеновской дифракции на наноматериалах в редакторе вычислительных сценариев MathCloud

требует особых усилий и вполне доступно специалисту, который до этого либо создавал, либо уже пользовался этими приложениями. Как отмечалось ранее, каждым из сервисов можно пользоваться изолированно в браузере через веб-форму, которая автоматически создается контейнером при обращении по URL-ссылке сервиса.

На Рис.3, 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.

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

Для замеров продолжительности выполнения заданий в сценарий добавлены блоки «псевдо-сервисов» в виде несложных процедур на JavaScript. На Рис.3 эти процедуры скрыты в блоках «tic», «toc». Здесь провод, входящий в

блок «tic», проведен из блока «Num. of calculation» (передающего в блок массовых расчетов параметр количества расчетов), а левый провод, входящий в блок «toc», - из сервиса массовых расчетов в гриде кривых рентгеновской дифракции «XRD_Calculation». Таким образом, блок «tic» «сработает» перед самым началом вызова сервиса массовых расчетов, а блок «toc» - сразу после его завершения.

4. Подсистема оптимизационной идентификации

Сценарий, представленный на Рис.3, включает в себя самые разнообразные приложения, которые можно разделить на три типа:

- консольные (в своем исходном виде) приложения, осуществляющие преобразования файлов с исходными и промежуточными данными;

- Грид-приложения (в данном примере - одно, представленное блоком «XRD_Calculation», осуществляющее «массированные» параллельные вычисления «модельных» кривых рассеяния $S_i(q_j)$, согласно формуле (1));

- приложения для кластера (отвечающие за решение задач оптимизации) входящие в состав блока «VVV-CarbonStructureAtCluster»). Остановимся подробнее на этом блоке, поскольку он демонстрирует как возможности применения в системе MathCloud составных сценариев в форме «вложенных» процедур, так и новый перспективный подход к созданию высокопроизводительных систем оптимизационного моделирования на основе MathCloud.

Как уже отмечалось выше, исходные данные для трех задач оптимизации (4), (5) и (6) (с ограничениями (2), (3)) подготавливаются в текстовом формате AMPL 0. Этот алгебраический

язык оптимизационного моделирования возник ещё в 1970-е годы и широко применяется во многих высокоуровневых системах оптимизации. Он позволяет записывать исходные данные оптимизационных задач (параметры, переменные, целевую функцию, ограничения и т.д.), допуская разделение «символьного» описания задачи (т.н. «модельное» представление) и набора числовых значений параметров. Применение AMPL предусматривает этапы, описание которых приведено в следующих разделах.

4.1. Подготовка стабов с описаниями оптимизационных задач

На этом этапе специальный транслятор AMPL принимает на вход два текстовых файла: *.mod, содержащего символьные описания всех соотношений одной или нескольких оптимизационных задач, и *.dat, - с численными значениями параметров указанных задач. На выходе - один или несколько (по числу сформированных экземпляров задач) файлов *.nl, содержащих т.н. *стабы*. Эти файлы содержат специальную структуру данных, адаптированную для дальнейшего автоматического дифференцирования и непосредственно передаваемую в пакеты оптимизации, совместимые со стандартом AMPL. Надо отметить, что большинство современных пакетов оптимизации поддерживают указанный формат (в т.ч. и упомянутые выше LP SOLVE и Iopt). Для управления работой AMPL-транслятора удобно использовать специальный, третий, текстовый файл, содержащий команды AMPL. Приведем наиболее существенные для предлагаемого подхода фрагменты этого файла, названного «preOpt_ND-WS-L_1_2_inf.amp» (пояснения - в комментариях после символа «#»).

```
option log_file 'ND-WS.log'; # задание имени вспомогательного текстового файла
                             # для записи «информационных» сообщений по работе
                             # транслятора
model ND-WS.mod; # загрузка файла с символьным описанием задач (4), (5), (6)
data ND-WS.dat; # загрузка файла с общими параметрами задач (4), (5), (6)
...
problem ND_L1; # выбор задачи (4), названной в ND-WS.mod ND_L1
write gND_L1; # команда генерации стаба задачи ND_L1 в виде файла ND_L1.nl
...
problem ND_L2; # выбор задачи (5), названной в ND-WS.mod ND_L2
write gND_L1; # создание стаба задачи ND_L2 в виде файла ND_L2.nl
...
problem ND_Linf; # выбор задачи (6), названной в ND-WS.mod ND_Linf
write gND_Linf; # создание стаба ND_Linf в виде файла ND_Linf.nl
```

Вызов в консоли команды

```
amp1 preOpt_ND-WS-L_1_2_inf.amp
```

(в случае успешного ее завершения) будет иметь своим результатом файл ND-WS.log и три стаб-файла ND_L1.nl, ND_L2.nl, ND_Linf.nl.

В сценарии на Рис.4 данный этап представлен сервисом «preOptCarbonStructure» с одним входным параметром - файлом с данными ND-WS.dat, который подготавливается сервисом «XRD_AMPL_data_converter» (Рис. 3) по результатам вычисления модельных кривых рассеяния. Неизменный файл ND-WS.mod с символьным описанием моделей (4)-(6) «скрыт» от пользователя и является «частью» реализации сервиса «preOptCarbonStructure». Ниже приведен дескриптор этого сервиса в формате JSON, содержание которого практически не требует пояснений. Стоит, пожалуй, только обратить внимание на простоту «превращения» консольной команды (выделена ниже жирным шрифтом) в удаленно доступный REST-сервис.

```
{ name: preOptCarbonStructure,
  description: "Prepare identification of carbon structures composition (Err
L_1_2_inf to min)",
  inputs: { data: {type: file, title: "x-Ray scattering data (experiment & mod-
els)" } },
  outputs: {
    log: {type: file, title: "Runtime log", description: "The raw output of
script"},
    L1_stub: {type: file, title: "L1 problem stub"},
    Linf_stub: {type: file, title: "Linf problem stub"},
    L2_stub: {type: file, title: "L2 problem stub"}
  },
  implementation: {
    type: command,
    command: "amp1 preOpt_ND-WS-L_1_2_inf.amp",
    input2file: { data: ND-WS.dat },
    file2output : { ND_L1.nl: L1_stub, ND_Linf.nl: Linf_stub, ND_L2.nl: L2_stub,
ND-WS.log: log }
  }
}
```

В заключение обратим внимание на то, что сервис должен быть размещен на хосте, где установлен AMPL-транслятор, fuji.isa.ru (Рис.4).

4.2. Решение оптимизационных задач

Созданные транслятором AMPL на предыдущем этапе файлы «со стабами» можно непосредственно передать пакетам оптимизации, указав в параметрах командной строки соответствующую опцию, например, вызов пакета Ipopt для решения задачи квадратичного программирования (5) в командной строке выглядит так:

```
ipopt ND_L2.nl -AMPL
```

Немного забегаая вперед, отметим, что опция «-AMPL» обеспечивает запись полученного решения в файл с расширением *.sol, формат которого также соответствует стандартам AMPL, что позволяет на следующем этапе обработать этот файл транслятором AMPL и преобразовать результаты решения в форму, необходимую для дальнейших действий (в нашем сценарии - это построение графиков). В данном разделе остановимся на ином важном аспекте, связанном с возможностью одновременного решения всех трех, не зависящих друг от друга, задач (4)-(6).

Для этого в сценарии был использован специальный плагин для преобразования в REST-сервис приложений для кластера (Cluster Adapter на Рис.2). Для этого, на головном хосте кластера grid.isa.ru был запущен сервис «opt-cluster», с дескриптором:

```
{
  "name" : "opt-cluster",
  "description" : "Submits a job to solve AMPL stub with a selected solver",
  "inputs" : {
    "stub" : { "type" : "file", "title" : "AMPL stub" },
    "solver_name" : { "type" : "string", "title" : "Solver to be used. Possible values:
ipopt, lpsolve.", "enum" : ["lpsolve", "ipopt"] } },
  "outputs" : { "result.sol" : { "type" : "file", "title" : "Solution found" },
    "stdout.txt" : { "type" : "file", "title" : "Solver's stdout output" },
    "stderr.txt" : { "type" : "file", "title" : "Solver's stderr output" } },
  "implementation": { "adapter": "cluster", "job-file": "services/opt-luster/solve.job",
    "input2file" : { "stub" : "stub.nl", "solver_options" : "ipopt.opt" },
    "file2output" : { "stub.sol" : "result.sol", "stdout" : "stdout.txt", "stderr" :
"stderr.txt" } }
}
```

Этот сервис имеет два входных параметра: стаб-файл задачи, «stub.nl», и текстовый параметр для выбора одного из пакетов оптимизации, предварительно установленного на кластере. Фактически, данный сервис предоставляет доступ к простому кластерному приложению следующего вида (файл solve.job), где параметр \$SOLVER_NAME\$ может значения «lpsolve» или «ipopt»:

```
case $SOLVER_NAME$ in
  lpsolve)
    /share/solvers/bin/lpsolve $JOB_DIR$/stub.nl -AMPL
    ;;
  ipopt)
    /share/solvers/bin/ipopt $JOB_DIR$/stub.nl -AMPL
    ;;
  *)
    echo "Unknown solver"
    ;;
esac
```

Здесь предполагается, что пакеты LP_SOLVE и Ipopt были установлены на головном хосте кластера в «общей» папке /share/solvers/bin/, доступной для всех рабочих хостов. При наличии достаточного количества свободных процессоров на кластере решение всех трех задач происходит параллельно.

4.3. Обработка файлов с решениями и подготовка общего файла с результатами

На третьем, завершающем этапе, полученные файлы с решениями ND_L1.sol, ND_L2.sol и ND_Linf.sol передаются сервису «postOptCarbonStructure» (Рис.4), установленному на хосте fuji.isa.ru, который, по аналогии с сервисом «postOptCarbonStructure», выполняет AMPL-скрипт «postOpt_ND-WS-L_1_2_inf.amp». Ниже приведены его основные фрагменты, существенные для пояснения сути предлагаемого подхода.

```
option log_file 'ND-WS.log'; # задание имени вспомогательного текстового файла
                             # для записи «информационных» сообщений по работе
                             # транслятора
model ND-WS.mod; # загрузка файла с символьным описанием задач (4), (5), (6)
data ND-WS.dat; # загрузка файла с общими параметрами задач (4), (5), (6)
...
param x_Vals{i in I, t in {1..3}}; # объявление параметров для записи оптимальных
                                   # значений переменных всех трех задач, здесь I -
                                   # множество индексов переменных x[i] (см. (1)-(3))
...
problem ND_L1; # выбор задачи (4), названной в ND-WS.mod ND_L1
solution ND_L1.sol; # команда чтения решения из файла ND_L1.sol
...
for {i in I} { let x_Vals[i,1] := x[i]; }; # сохранение переменных в x_Vals
...
problem ND_L2; # выбор задачи (5), названной в ND-WS.mod ND_L2
```

```

solution ND_L2.sol;    # чтение решения из файла ND_L2.sol
...
for {i in I} { let x_Vals[i,2] := x[i]; }; # сохранение переменных в x_Vals
...
problem ND_Linf; # выбор задачи (6), названной в ND-WS.mod ND_Linf
solution ND_Linf.sol;    # чтение решения из файла ND_Linf.sol
...
for {i in I} { let x_Vals[i,3] := x[i]; }; # сохранение переменных в x_Vals
...
printf ... > ND-WS-out.txt # вывод x_Vals и иных значений в файл ND-WS-out.txt

```

Таким образом, этот «консольный» сервис принимает на вход три файла с решениями *.sol и файл с данными ND-WS.dat (тот же, что и сервис «preOptCarbonStructure»). На выходе - вспомогательный лог-файл ND-WS.log и основной файл ND-WS-out.txt, с надлежащим образом оформленными результатами оптимизации.

Заключение

В работе демонстрируется методика интеграции разнородных приложений в распределенную систему обработки экспериментальных данных рентгеновской дифрактометрии углеродистых наноматериалов на основе унифицированных REST-сервисов, реализованных с использованием инструментария MathCloud. Созданное распределенное приложение включает массовые расчеты кривых рассеяния отдельными наноструктурами с последующим решением оптимизационных задач для определения наиболее вероятного распределения наноструктур по топологии и размеру в исследуемом образце.

Применение системы оптимизационного моделирования AMPL в сочетании с AMPL-совместимыми пакетами оптимизации и возможностями программного инструментария MathCloud позволило создать высокопроизводительную подсистему оптимизации, интегрированную в общий вычислительный сценарий.

Реализация вычислительного сценария идентификации структурного состава углеродных наноструктур по рентгеновским дифрак-

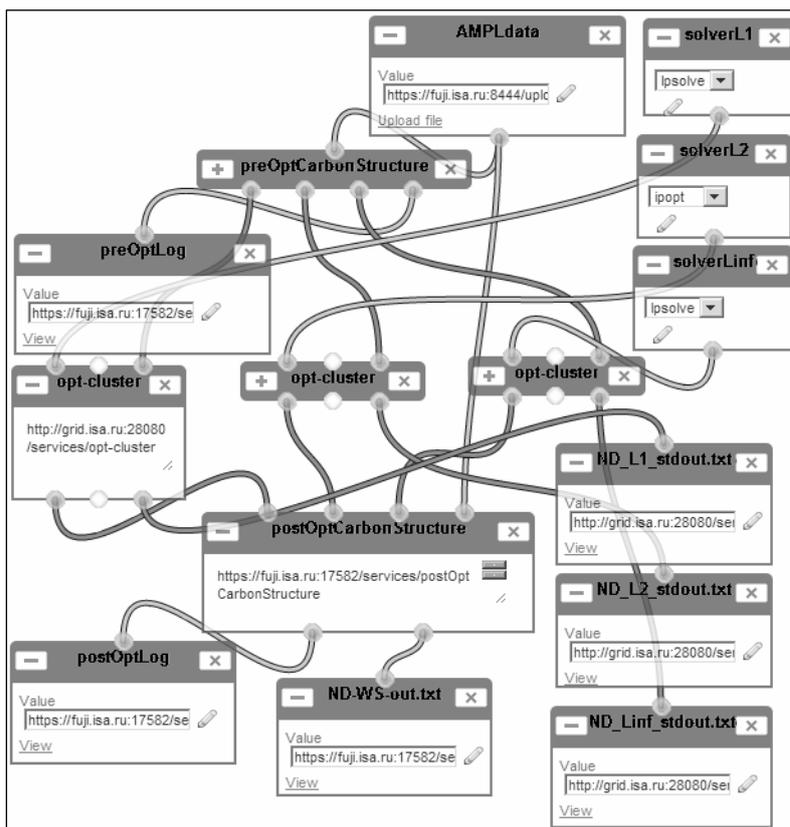


Рис. 4. «Оптимизационный» сценарий (блок «VVV-CarbonStructureAtCluster»)

ционным 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 распределенное приложение.

Авторы выражают благодарность за помощь в работе А.Б. Кукушкину, В.А. Вознесенскому, Н.Л. Марусову, И.Б. Семенову, О.В. Сухорослову. Авторы также благодарны А.А. Велиг-

жанину, Я.В. Зубавичусу, Н.Ю. Свечникову и В.Г. Станкевичу за сотрудничество в [1-3], стимулировавшее выполнение настоящей работы, и А.П. Афанасьеву - за поддержку работы.

Литература

1. A.B. Kukushkin, V.S. Neverov, N.L. Marusov, I.B. Semenov, B.N. Kolbasov, V.V. Voloshinov, A.P. Afanasiev, A.S. Tarasov, V.G. Stankevich, N.Yu. Svechnikov, A.A. Veligzhaniin, Ya.V. Zubavichus and L.A. Chernozatonskii, Few-nanometer-wide carbon toroids in the hydrocarbon films deposited in tokamak T-10, *Chem. Phys. Lett.*, 506, 2011, pp. 265-268, doi:10.1016/j.cplett.2011.03.036.
2. 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.
3. 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, Численное моделирование эффектов интерференции рентгеновского рассеяния углеродными наноструктурами в осаждённых плёнках из токамака Т-10, *Вопросы атомной науки и техники, Сер. Термоядерный синтез*, 2011, том. 1, стр. 13-23.
4. 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., Новороссийск)*. – М.: Изд-во МГУ, 2009. С. 463-467
5. 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
6. 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
7. 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
8. Biegler, L.T., Zavala V.M. Large-Scale Nonlinear Programming with IPOPT: An Integrating Framework for Enterprise-Wide Optimization // *Comp. Chem. Eng.* - 2009. - V. 33, N. 3. - P. 575-582.
9. Fourer R., Gay D.M., Kernighan B.W. *AMPL: A Modeling Language for Mathematical Programming*, second edition. Duxbury Press / Brooks/Cole Publishing Company, 2002. — 538 p.

Волошинов Владимир Владимирович. Заведующий лабораторией Института системного анализа РАН. Окончил Московский физико-технический институт в 1984 году. Кандидат физико-математических наук. Автор более 50 печатных работ. Область научных интересов: теория и численные методы оптимизации, программно-алгоритмическое обеспечение научных исследований, технологии распределенных вычислений. E-mail: vladimir.voloshinov@gmail.com, vv_vol@isa.ru.

Неверов Владислав Сергеевич. Аспирант Института физики токамаков НИЦ «Курчатовский институт». Автор 11 печатных работ. Область научных интересов: физика плазмы, распределенные вычисления, численное моделирование физических процессов. E-mail: vs-never@hotmail.com.