

## **Интеграция системы MATLAB в распределенную вычислительную среду**

Д. А. Хуторной

### **Введение**

В связи с решением сложных вычислительных задач в распределенной среде возникает проблема использования возможностей уже существующих информационно-алгоритмических ресурсов (ИАР [6]), таких как пакеты решения оптимизационных задач, дифференциальных уравнений, символьных вычислений и т. п. При этом необходимо обеспечивать прозрачный доступ к таким ресурсам на основе распространенных технологий распределенных вычислений. Под ИАР понимается типизированный ресурс, предоставляющий внешний интерфейс, соответствующий его типу и функциональности. Реализация такого унифицированного интерфейса позволяет интегрировать ресурс в распределенную вычислительную среду и многократно использовать его как готовый компонент. Однако большинство существующих программных пакетов, предназначенных для научных расчетов, не предоставляют готовых реализаций каких-либо интерфейсов, позволяющих обращаться к ним как компонентам в рамках известных технологий распределенных вычислений. Тем не менее, взаимодействие внешних программ с такими пакетами зачастую может быть осуществлено посредством предоставляемого программного интерфейса (API) или иного механизма, например путем ввода и вывода данных через файлы. Эти возможности могут быть использованы для создания специальных программных адаптеров, позволяющих организовать доступ к такому ресурсу в соответствии с определенным типом. В терминологии разрабатываемой платформы IARnet [6], такие адаптеры называются агентами удаленного доступа. Задачей агентов доступа является предоставить унифицированный интерфейс для интеграции ИАР в распределенную вычислительную среду, скрывая в тоже время индивидуальные особенности взаимодействия с конкретным ресурсом.

Система научных расчетов MATLAB предоставляет богатые возможности для решения широкого круга вычислительных задач, например:

- Optimization Toolbox — предоставляет набор функций для решения различных задач оптимизации;
- Symbolic Math Toolbox — позволяет выполнять символьные вычисления (такие как дифференцирование и интегрирование, решение

алгебраических уравнений, упрощение выражений, вычисление численных значений выражений);

- множество других полезных функций, входящих в ядро MATLAB (функции для операций с матрицами, численного решения дифференциальных уравнений и др.).

В настоящей статье описан агент, предназначенный для организации удаленного доступа к системе MATLAB и позволяющий использовать ее возможности в качестве информационно-алгоритмического ресурса при распределенном решении вычислительных задач, в частности задач оптимального управления.

## Принципы реализации

Система MATLAB предоставляет ряд внешних интерфейсов (*external interfaces* [2]), а также некоторые возможности для сетевого взаимодействия. Остановимся кратко на некоторых из них.

### MATLAB Web Server

MATLAB Web Server позволяет создавать приложения MATLAB, доступ к которым может осуществляться удаленно посредством обычного Web browser-а. Ввод данных осуществляется пользователем через HTML-формы, затем они передаются непосредственно приложению, которое также ответственно за вывод результирующих данных в виде HTML для отображения пользователю.

Этот механизм не пригоден для включения MATLAB в распределенную вычислительную среду, так как не предоставляет никаких программных интерфейсов, которые позволили бы использовать MATLAB в качестве компоненты для построения распределенных приложений.

### Web-сервисы

MATLAB предоставляет набор функций, позволяющих осуществлять запросы к Web-сервисам. С помощью этих функций можно сгенерировать класс по заданному описанию интерфейса Web-сервиса на языке WSDL, а затем вызывать его методы для формирования запросов. Соответствие типов данных MATLAB и WSDL при этом фиксировано, и преобразование осуществляется автоматически.

Механизм взаимодействия MATLAB с Web-сервисами также не позволяет использовать MATLAB в распределенной среде, так как предоставляет лишь клиентский интерфейс для вызова Web-сервисов, в то время как более интересным представляется использование самой среды MATLAB в качестве Web-сервиса.

### COM/DCOM

Доступ к MATLAB может быть организован на основе технологии COM/DCOM. MATLAB позволяет использовать его в качестве COM-компоненты, для сетевого доступа при этом может быть использован

механизм DCOM. Очевидным недостатком такого подхода является платформозависимость: технология DCOM рассчитана только на ОС Windows.

### **Distributed Computing Toolbox и Distributed Computing Engine**

В ноябре 2004 г. компания MathWorks выпустила Distributed Computing Toolbox, позволяющий выполнять алгоритмы, реализованные в среде MATLAB, на объединенных сетью компьютерах, на которых также работает MATLAB. Для этого пользователь MATLAB должен создать набор задач (*tasks*), объединить их в задание (*job*) и передать задание для выполнения менеджеру заданий (*job manager*), который является частью Distributed Computing Engine. Менеджер заданий выполняет задания по очереди, распределяя содержащиеся в них задачи по известным ему процессам-исполнителям (*workers*).

Distributed Computing Toolbox и Distributed Computing Engine могут работать на различных платформах, поддерживаемых MATLAB (Windows, Unix, Macintosh). Из недостатков описанного инструментария можно отметить то, что не предусмотрена возможность загрузки на удаленные узлы определенных пользователем функций, а также нет механизма для прямого взаимодействия процессов-исполнителей. Существенным также является то, что данный подход позволяет объединять в распределенную вычислительную среду только ресурсы MATLAB и не предусматривает взаимодействия с ресурсами других типов.

### **C/C++ API и технология CORBA**

Система MATLAB предоставляет интерфейсы, которые могут быть использованы внешними программами, написанными на языках программирования C/C++, FORTRAN. Эти интерфейсы позволяют использовать функциональность ядра MATLAB (*engine*), работать с его структурами данных (*mxArray*), файлами данных в формате MATLAB (*mat*-файлами).

Агент доступа реализован на основе стандарта CORBA [4] и внешнем интерфейсе MATLAB для языка C [2] (C Engine Functions, C MX-Functions и C MAT-File Functions) на языке C++. В качестве реализации CORBA использован свободно распространяемый комплект ACE-5.4 + TAO-1.4 [3] для C++.

Реализованы два основных программных интерфейса:

- Интерфейс *MatlabEngine* — предназначен для выполнения команд в среде MATLAB, загрузки переменных и получения их из рабочего пространства MATLAB; реализация основана на использовании C Engine и C MAT-File функций из MATLAB API. Методы этого интерфейса описаны в табл. 1.
- Интерфейс *MxArray* — предназначен для представления данных в формате MATLAB (в виде структур *mxArray*) и доступа к ним; реализация основана на C MX-функциях — части внешнего интерфейса MATLAB для языка C. Методы интерфейса *MxArray* описаны в табл. 2.

Таблица 1

Описание методов интерфейса MatlabEngine

Метод/атрибут	Комментарий
attribute long outputBufferLength;	Атрибут, задающий размер буфера вывода MATLAB
string evalString (in string str);	Метод, осуществляющий выполнение выражения, содержащегося в строке str, в среде MATLAB. Возвращает содержимое буфера вывода после выполнения выражения
Void putVariable (in string name, in MxArray mxArr);	Метод для загрузки переменной mxArr в рабочее пространство MATLAB Engine. Переменной присваивается имя name
MxArray getVariable (in string name);	Метод для получения переменной с именем name из рабочего пространства MATLAB Engine
Void putMFile (in string name, in ByteSeq dt);	Метод для загрузки m-файла в директорию, доступную MATLAB Engine. Файлу присваивается имя name
Void putMatFile (in ByteSeq dt);	Метод для загрузки всех переменных из заданного mat-файла в рабочее пространство MATLAB Engine
Void close();	Метод для закрытия сессии работы с MATLAB Engine. Закрывает Engine и уничтожает соответствующий CORBA-объект

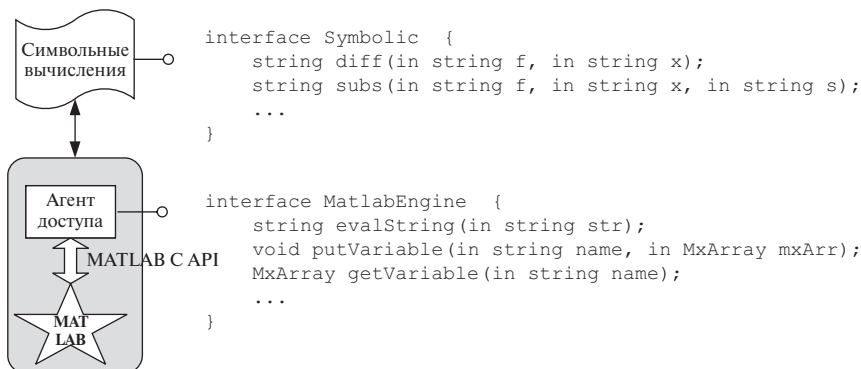


Рис. 1

Таблица 2

Описание методов интерфейса MxArray

Метод	Комментарий
<code>unsigned short getNumberOfDimensions();</code>	Метод для получения числа измерений массива
<code>UshortSeq getDimensions();</code>	Метод для получения размерности массива по каждому из измерений
<code>DoubleSeq getReal();</code>	Метод для получения вещественных частей данных, хранящихся в массиве
<code>DoubleSeq getImag();</code>	Метод для получения мнимых частей данных, хранящихся в массиве
<code>Double getScalar();</code>	Метод для получения вещественной части первого числа, хранящегося в массиве
<code>Void setReal(in DoubleSeq r);</code>	Метод для присваивания новых значений вещественных частей данных, хранящихся в массиве
<code>Void setImag(in DoubleSeq i);</code>	Метод для присваивания новых значений мнимых частей данных, хранящихся в массиве
<code>Void destroyArray();</code>	Метод для уничтожения массива

Предусмотрены два дополнительных интерфейса:

- EngineFactory — интерфейс-фабрика, содержащий метод для создания нового экземпляра MatlabEngine.
- MxArrayFactory — интерфейс-фабрика, содержащий метод для создания нового экземпляра MxArray.

Для решения конкретных вычислительных задач требуется использование ИАР соответствующего типа, предоставляющих специализированный интерфейс, например ИАР, выполняющий символьные вычисления (рис. 1). Поэтому не удобно использовать в качестве компоненты распределенной системы самого агента доступа в описанном виде. Компоненты со специализированным интерфейсом могут быть реализованы в виде «адаптеров», скрывающих взаимодействие с агентом доступа к MATLAB, на любом языке программирования, поддерживаемом CORBA, например, на языке CorbaScript [5].

### Консоль для удаленной работы с MATLAB

Для тестирования агента написано клиентское консольное приложение, позволяющее удаленно использовать возможности MATLAB в режиме командной строки. Приложение создает свою копию объекта, реализу-

ющего интерфейс MatlabEngine, для доступа к MATLAB. С его помощью в среде MATLAB выполняются команды, введенные пользователем в командной строке. Результаты выполнения команд выводятся на экран.

Предусмотрена специальная команда для загрузки m-файлов с функциями на языке MATLAB на сервер, что позволяет в дальнейшем использовать эти функции:

```
mload <имя загружаемого файла> [<имя файла на сервере>]
```

Также есть команда, позволяющая загрузить в рабочее пространство MATLAB Engine все или только отдельные переменные из mat-файла с данными в формате MATLAB, расположенного на клиентской стороне:

```
matload <имя mat-файла> [<имя переменной>*]
```

и команда для сохранения переменных из рабочего пространства MATLAB Engine в mat-файл на клиентской стороне:

```
matsave <имя mat-файла> [<имя переменной>*]
```

Ниже приведен пример работы с консолью. В качестве аргумента приложению должна быть передана ссылка на объект-фабрику типа EngineFactory, в данном случае это адрес объекта на сервисе именовании:

```
D:\>MatlabConsole -MEFRef
corbaname::dcs.isa.ru:30002#IARnet/test/MatlabEngineFactory
MATLAB console
>>
```

Загружаем m-файл, содержащий функцию для вычисления значений правых частей системы ОДУ:

```
>>mload mfile.m mf.m
loading m-file... ok
>>
```

Решаем систему с помощью функции MATLAB ode45:

```
>>[t, y] = ode45(@mf, [0 1])
t = 0
0.0250
...
y = 1.0000 1.0000
...
>>
```

Сохраняем результаты расчетов в файл и выходим:

```
>>matsave data.mat t y
writing data to mat-file... ok
>>exit
MATLAB session closed.
```



операции — вычисления значения функции, входящей в уравнение — они могут обращаться к ресурсу, хранящему символьное выражение функций и имеющему возможность вычисления его значения при заданных значениях переменных.

Процедура распределенного решения задачи ОУ, схематически изображенная на рис. 2, описана в [1]. При ее реализации удобно использовать ресурсы, предоставляющие специализированные интерфейсы для решения подзадач одного из указанных типов, которые в свою очередь могут использовать вычислительные возможности MATLAB посредством взаимодействия с агентом доступа.

## Выводы

Из краткого обзора и приведенных результатов работы следует, что на основе созданного агента доступа можно организовать доступ к вычислительным ресурсам системы MATLAB, используя различные экземпляры MATLAB Engine, снабженные агентами доступа, для параллельного и координированного решения разнотипных математических задач в рамках распределенного решения более крупных вычислительных задач.

## Литература

1. *Афанасьев А. П., Хуторной Д. А.* Решение задачи синтеза оптимального управления в распределенной среде // Проблемы вычислений в распределенной среде. М.: УРСС, 2004.
2. Документация MATLAB ([www.mathworks.com/access/helpdesk/help/techdoc/matlab.html](http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html)).
3. ACE TAO ([www.theaceorb.com](http://www.theaceorb.com)).
4. The Common Object Request Broker: Architecture and Specification. OMG Revision 2.4. October, 2000.
5. CORBA Scripting Language Specification. Version 1.1. February 2003. OMG formal/03-02-01.
6. *Афанасьев А. П., Волошинов В. В., Кривцов В. Е., Рогов С. В., Сухорослов О. В.* Использование информационно-алгоритмических ресурсов для организации распределенных вычислений // Проблемы вычислений в распределенной среде. М.: УРСС, 2004.
7. *Афанасьев А. П., Дикусар В. В., Милютин А. А., Чуканов С. А.* Необходимое условие в оптимальном управлении. М.: Наука, 1990.