

# Методы адаптации приложений для грид-сред ЕМІ и РГС\*

Н.А. Кутовский<sup>1,2</sup>

e-mail: Nikolay.Kutovskiy@jinr.ru, <sup>1</sup>Лаборатория информационных технологий, Объединенный институт ядерных исследований, Дубна

<sup>2</sup>Национальный научно-образовательный центр физики частиц и высоких энергий Белорусского государственного университета, Минск, Беларусь

**Аннотация.** Грид-технологии стали незаменимым инструментом во многих областях науки. Полноценное их использование специалистами в решении своих прикладных задач предполагает адаптацию соответствующих приложений для работы в конкретной грид-среде. В этой связи является актуальным разработка методов “гридификации” приложений. В данной статье описаны факторы, которые необходимо учитывать при адаптации пакета для работы в той или иной грид-среде, а также методы гридификации приложений для грид-сред на базе промежуточного программного обеспечения ЕМІ и Российской национальной грид-сети.

Процедура адаптации конкретного прикладного программного пакета (ППП) для использования в той или иной грид-инфраструктуре зависит от нескольких факторов, среди которых ключевыми можно назвать следующие:

1. средства взаимодействия пользователя с грид-средой для запуска задач;
2. тип вычислительных ресурсов, пригодных для работы адаптируемого ППП и представленных в конкретной грид-инфраструктуре.

Оба фактора зависят от ППО, на базе которого построена конкретная грид-инфраструктура.

На данный момент существует достаточно много ППО, реализующего концепцию Грид. Оно различается по используемым языкам программирования, по набору сервисов и предоставляемой функциональности, типу вычислительных ресурсов и элементов хранения данных, поддерживаемым операционным системам, условиям распространения и использования и т. д.

Для работы пользователя в грид-среде ему предоставляется один или несколько интерфейсов. Например, интерфейс командной строки, где взаимодействие пользователя с грид-инфраструктурой осуществляется путем вызова команд с определенным набором параметров. Для запуска задачи посредством такого интерфейса, как правило, пользователь должен предварительно составить файл еџ описания на понятном

данной грид-среде языке. В качестве примеров таких интерфейсов можно привести интерфейс командной строки ГридННС – ПИКС, интерфейс командной строки ЕМІ, Globus Toolkit или Российской национальной грид-сети (РГС). Другим возможным способом взаимодействия пользователя с грид-средой является веб-интерфейс, как например ВИГ (веб-интерфейс ГридННС и РГС). Также есть разработки, предоставляющие пользователям единый веб-интерфейс к различным грид-средами. В качестве примера такового можно привести упоминавшийся ранее портал WS-PGRADE.

Помимо этого, грид-сервисы различных ППО имеют интерфейсы программирования приложений (application programming interface, API), используя которые можно создать собственные интерфейсы, обеспечивающие необходимую конечным пользователям функциональность для взаимодействия с грид-средой.

Что же касается вычислительных ресурсов, представленных в грид-инфраструктурах, то в них можно условно выделить следующие типы:

1. суперкомпьютеры – высокопроизводительные вычислительные комплексы для решения сильносвязанных параллельных задач;
2. кластеры – вычислительные системы для обработки независимых последовательных задач;
3. гетерогенные среды, в качестве вычислительных ресурсов которых выступают персональные компьютеры (т. н. “DesktopGrid”).

В конкретной грид-инфраструктуре могут быть представлены как один, так и несколько или даже все из перечисленных выше типов вычислительных ресурсов.

Подходящий тип вычислительных ресурсов (а значит и процедура его интеграции в грид-среду) для конкретного приложения зависит от возможных режимов его работы, а именно:

1. параллельный, когда выполняющие одну задачу процессоры могут обмениваться между собой сообщениями, например, посредством библиотек MPI;

\*Работы выполнены при частичной поддержке грантом ОИЯИ №13-602-03

2. последовательный, когда каждый процессор выполняет независимую часть задачи.

Для использования приложений в параллельном режиме подходят суперкомпьютеры и кластеры с поддержкой счета параллельных задач, тогда как для последовательных задач возможно использование всех типов вычислительных ресурсов, хотя некоторые из них не всегда оптимальны.

Действия по адаптации приложений для работы в среде грид зависят как от режима работы пакета, так и от способа запуска задач.

На учебно-исследовательской и тестовой грид-инфраструктуре ЛИТ ОИЯИ [1] были выполнены работы по разработке методов адаптации приложений для грид-сред на базе промежуточного программного обеспечения (ППО) ЕМІ и РГС [2]. Разработанные методы адаптации прикладных программных пакетов (ППП) для грид-сред на базе ППО ЕМІ и РГС предполагают, что приложения удовлетворяют следующим требованиям:

1. совместимость прикладного пакета с операционной системой (ОС) семейства Linux, т. к. ОС этого типа установлены на рабочих узлах ресурсных центров соответствующих грид-инфраструктур;
2. возможность вызова приложения из командной строки, т. к. непосредственный его вызов на рабочих узлах осуществляется из скриптов;
3. отсутствие необходимости интерактивного ввода какой-либо информации, т. к. у пользователя нет возможность еще предоставить при таком способе запуска;
4. возможность передачи параметров вызова приложения и входных данных через аргументы в командной строке и (или) в одном или нескольких отдельных файлах.

Общую процедуру адаптации ППП для обоих грид-сред можно сформулировать следующим образом:

1. проверка приложения на соответствие указанным выше требованиям;
2. создание вспомогательного скрипта, осуществляющего запуск приложения с нужными параметрами, а также при необходимости выполняющего набор операций до и после вызова прикладного пакета;
3. разработка файла описания задания на соответствующем языке конкретной грид-среды;
4. размещение приложения на рабочих узлах кластера, как минимум, одного грид-сайта;

5. внесение соответствующей информации об установленном пакете в информационную систему грид-сайта.

Однако, каждая из грид-сред имеет свою специфику, в связи с чем есть некоторые отличия в методах адаптации приложений для каждой из них.

**Методы адаптации приложений для грид-среды ЕМІ.** Методы адаптации приложений для грид-сред на базе ППО ЕМІ различаются в зависимости от режима работы портируемого пакета.

Если приложение работает в последовательном режиме, то в общем случае необходимо выполнить следующие шаги.

1. Создать скрипт, предназначенный для исполнения непосредственно на рабочих узлах кластера ВЭ и осуществляющий такие действия, как
  - предварительную подготовку (при необходимости) до вызова приложения (например, извлечение файлов со входными данными из архива, проверку соблюдения обязательных для работы пакета условий и т. п.);
  - запуск приложения с необходимыми параметрами;
  - пост-обработку (упаковка результатов в архив и т. п.).
2. Создать файл описания задачи на языке JDL, где указать в атрибуте Executable имя файла скрипта, сформированного на первом шаге, и его же в атрибуте InputSandbox для передачи на рабочие узлы. Кроме того, в этом же атрибуте необходимо перечислить набор файлов со входными данными либо их архив.
3. Сделать доступным на рабочих узлах адаптируемое приложение (обычно, его размещают в сетевой директории, смонтированной на всех рабочих узлах). Как правило, его устанавливает пользователь, являющийся членом конкретной ВО и имеющий в ней роль менеджера программной группы (software group manager, sgm). У него есть необходимые права доступа в данную директорию, в том числе и на запись.
4. Также этот sgm-пользователь ответственен за внесение соответствующих данных об установленных пакетах, которая затем публично доступна из информационной системы грид-сайта. Для этих целей используются специальные утилиты lcg-tags и lcg-ManageVOTag.

Чтобы не перечислять имена всех файлов со входными данными в файле описания задачи (jdl-файле) удобно их поместить в архив. Использование одного и того же имени архива для разных наборов файлов со входными данными позволяет избавиться от необходимости менять его имя в атрибуте InputSandbox. Если же файл со входными данными для данного приложения один и он относительно небольшого объёма, тогда нет необходимости его архивировать, хотя по-прежнему удобно использовать одно и то же имя, чтобы не редактировать jdl-файл при изменении имени входного файла.

Процедура гридификации параллельных приложений несколько отличается.

Если приложение поддерживает работу в параллельном режиме (например, с помощью библиотеки MPI), тогда для его запуска в среде EMI удобно использовать пакет mpi-start. Он представляет собой набор вспомогательных скриптов, предоставляющих единый интерфейс для запуска параллельных задач в различных средах выполнения и позволяющий пользователю облегчить запуск MPI-приложений в грид-инфраструктурах на базе ППО EMI.

Обычно данный пакет предустановлен на рабочих узлах грид-сайтов EMI, поддерживающий возможность запуска параллельных задач. Однако в случае его отсутствия, пользователь может переслать архив с ним на рабочие узлы вместе со входными файлами для своей параллельной задачи и соответствующим образом задействовать этот пакет.

Использование mpi-start предъявляет определённые требования к составлению файла описания задачи и предполагает применение одного или двух вспомогательных скриптов. Они устанавливают соответствующие переменные среды окружения и обеспечивают вызов mpi-start с параметрами для корректного запуска пользовательского приложения (как правило, данный функционал реализуют в скрипте с именем mpi-start-wrapper.sh, хотя имя жёстко не фиксировано и может быть любым). Также эти скрипты указывают необходимые к выполнению до и после непосредственного вызова приложения команды с нужными параметрами (обычно задаются в файле с именем mpi-hooks.sh).

Скрипт mpi-start-wrapper.sh подходит для запуска большинства параллельных приложений и, как правило, нет необходимости его модифицировать. В файле mpi-hooks.sh в функциях pre\_run\_hook и post\_run\_hook указывают набор команд, которые должны быть выполнены до и после запуска приложения соответственно.

С помощью разработанных методов адаптации

параллельных приложений для грид-среды на базе EMI были портированы следующие пакеты: DL\_POLY, Elmer, Molpro, MEEP и FDS. Запуск задач, использующих эти приложения на рабочих узлах, осуществляется из командной строки интерфейса пользователя.

**Методы адаптации приложений для грид-среды PGC.** Одной из задач в проекте создания российской грид-сети для высокопроизводительных вычислений с целью облегчения работы пользователей в этой инфраструктуре являлась разработка специализированных проблемно-ориентированных веб-интерфейсов (ПОИ) для набора прикладных пакетов, которые бы позволяли упростить формирование и запуск счётных заданий, использующие конкретный ППП, контроль хода их выполнения и получение результатов вычислений.

В инфраструктуре PGC в качестве вычислительных ресурсов используются суперкомпьютеры и кластеры с поддержкой счёта параллельных задач.

Использование ППП в данной среде возможно посредством запуска задач как из интерфейса командной строки, так и с помощью веб-интерфейса либо специальных модулей расширения (плагинов) к нему, специфичных для каждого конкретного прикладного пакета.

Набор действий при адаптации последовательного приложения для работы в PGC аналогичен. Однако есть некоторые отличия, которые представлены ниже.

В PGC различают понятия “задание” и “задача”. Задание — это совокупность задач, представленная в виде направленного ациклического графа. Задачи в рамках одного задания могут быть иметь различные взаимозависимости.

Для составления файлов описания заданий и задач в PGC используется формат JSON, из-за чего их содержимое имеет другой вид.

Так в качестве значения атрибута executable рекомендуется указать абсолютный путь к исполняемому файлу вызова оболочки, для которой написан вспомогательный скрипт. Это связано с тем, что все входящие файлы, указанные в атрибуте input\_files, скачиваются ВЭ с помощью соответствующих утилит по протоколу gsiftp. Полученные таким способом файлы не имеют прав на исполнение (соответствующий бит установлен в 0). В значении атрибута arguments указывается имя вспомогательного файла-скрипта, которое также перечисляется в атрибуте input\_files среди всех входных файлов, включая файлы с данными. Все эти файлы (вспомогательный скрипт, а также файлы со входными данными) должны быть доступны ВЭ

по сети по протоколу gsiftp, а путь к директории с ним необходимо указать либо в значении атрибута `default_storage_base`, либо для каждого файла отдельно в атрибуте `input_files`.

В атрибуте `output_files` нужно перечислить имена всех файлов, которые необходимо сохранить по результатам работы приложения.

Требования к установленному на вычислительном элементе ПО необходимо задать с помощью вложенного атрибута `software` основного атрибута `requirements`.

Помимо этого, прикладное ПО на рабочих узлах ВЭ устанавливает системный администратор соответствующего грид-сайта, а также прописывает профили окружения для этого ПО и заносит необходимую о нём информацию в информационную систему.

Файлы со входными и выходными данными также удобно размещать в архивах, соответствующие операции над которыми выполняет вспомогательный скрипт.

Для запуска MPI-задач в среде PGC предусмотрен атрибут `count`, в качестве значения которого указывается целое число, равное необходимому для работы данного приложения количеству процессоров. Однако в силу того, что в ППО PGC не предусмотрено отдельное выполнение набора команд до и после вызова приложения, то указание в качестве исполняемого файла имени вспомогательного скрипта приводит к одновременному запуску N его копий, где N — количество процессоров, указанных в атрибуте `count`. Т.е. если скрипт выполняет распаковку архива, то эта операция будет запущена на N процессорах одновременно, что не является корректным поведением. Для избежания этого вспомогательный скрипт должен быть написан таким образом, чтобы команды предварительной подготовки и последующей обработки выполнялись одним процессором, а непосредственная работа параллельного приложения — на заданном пользователем их количестве.

Как было упомянуто ранее, помимо запуска заданий из командной строки, PGC предоставляет пользователям возможность взаимодействия с этой средой посредством графического веб-интерфейса. Для удобства работы пользователей с конкретными прикладными пакетами разработчики предусмотрели возможность создания специализированных веб-интерфейсов в виде плагинов к ВИГ, что позволяет использовать его готовые модули по работе с сертификатами и входными/выходными файлами. Такие плагины принято называть ПОИ.

Отличие методов адаптации приложений в виде ПОИ состоит в следующем. На основании

данных, заданных пользователем в проблемно-ориентированном веб-интерфейсе, этот ПОИ формирует вспомогательный скрипт, сохраняет его в отдельный файл на сервере GridFTP, а файл с описанием задания создаётся в виде объекта и при отправке задания на счёт соответствующей функцией передаётся необходимой компоненте.

Для данной среды с помощью описанных выше методов был адаптирован набор приложений [3], информация о которых представлена в таблице 1.

Таблица 1. Информация об адаптированных для PGC прикладных пакетах

Название пакета	Интерфейс запуска	Режим работы	Описание области применения пакета
DL_POLY	командная строка, ПОИ	параллельный	молекулярная динамика
Elmer	ПОИ	параллельный	инженерные расчёты
FDS	ПОИ	параллельный	моделирование задымлений и пожаров
GEANT4-DNA	командная строка, ПОИ	последовательный	моделирование биологических повреждений, вызванных ионизирующей радиацией на клеточном или субклеточном уровне
ZondGeoStat	ПОИ	последовательный	обработка геофизических, геохимических полей и данных дистанционных зондирований

Разработанные ПОИ (снимки экранов некоторых из них представлены на рис. 1) позволяют пользователю выполнять следующие действия:

- указать название задания (по умолчанию ПОИ автоматически генерирует в соответствующем поле название, состоящее из имени пакета, текущей даты и времени);
- выбрать требуемое окружение на вычислительных узлах;
- задать необходимое количество процессоров (только для DL\_POLY, Elmer и FDS, способных работать в параллельном режиме) либо начальный размер сканирующего окна, шаг его изменения, количество окон и функцию статистики (для ППП ZondGeoStat);

- выбрать архив входных данных, которые должны быть подготовлены теми же средствами, что и при работе с пакетом вне грид-среды;
- указать имя файла для архива с выходными данными (если имя в поле “Архив выходных данных” пользователем не задано, тогда будет использоваться имя output.tar.gz);
- запускать задания на счёт путём нажатия на кнопку ПОИ “Запустить” (обязательные поля должны быть заполнены);
- просматривать результаты операций в окне “Журнал работы”.

В пункте “Запуск и контроль” основного меню ВИГ можно отслеживать статус отправленных заданий.

ла задач, соответствующего указанному в поле “Количество окон” значению. В отличие от параллельных приложений, эти задачи не взаимодействуют между собой во время вычислений, а каждая из них обрабатывает входные данные согласно заданным параметрам алгоритма и формирует свой набор файлов с результатами.

## Заключение

На учебно-исследовательской и тестовой грид-инфраструктуре ЛИТ ОИЯИ были выполнены работы по разработке методов адаптации приложений для грид-сред на базе ППО ЕМІ и РГС. С использованием разработанных методов был адаптирован набор приложений для обоих грид-сред.



Рисунок 1. Снимки экрана ПОИ адаптированных для РГС приложений

Т.к. ППП DL\_POLY, Elmer и FDS имеют одинаковый набор входных параметров, то и ПОИ для этих пакетов тоже имеют одинаковый вид, а также код модуля расширения ВИГ.

Пакет GEANT4-DNA может работать только в последовательном режиме. Поэтому его ПОИ не имеет поля “Количество процессоров”, т. к. оно всегда равно единице.

ПОИ ZondGeoStat несколько отличается от ПОИ предыдущих ППП, адаптированных для работы в РГС, в силу специфики самого пакета. Она заключается в том, что входные данные должны обрабатываться одним и тем же выбранным алгоритмом, но с меняющимся параметром (размером сканирующего окна), диапазон изменения которого задаётся пользователем в соответствующих полях ПОИ: “Сторона первого (начального) окна”, “Шаг изменения стороны окна”, “Количество окон”. По этой причине ПОИ формирует задание, состоящее из определённого чис-

## Список литературы

- [1] N.A. Kutovskiy Cloud autonomous grid infrastructures for training, research, development and testing, *ibid*, pp. ????
- [2] V. V. Korenkov, N. A. Kutovskiy, R. N. Semenov An experience of the application software packages adaptation for running in grid environments // *Computer Research and Modeling*, 2012, vol. 4, no. 2, pp. 339–344 (in Russian)
- [3] Kutovskiy N.A., Lensky I.I., Semenov R.N. Problem-oriented web-interfaces for Russian grid network // *The 5th International Conference “Distributed Computing and Grid-technologies in Science and Education”* (Dubna, 16-21 July 3, 2012): proceedings. Dubna, 2012. P. 186-188