

С.В. Марьин, С.В. Ковальчук, А.В. Ларченко

НИИ наукоемких компьютерных технологий Санкт-Петербургского
государственного университета информационных технологий,
механики и оптики

ИНТЕЛЛЕКТУАЛЬНАЯ ПЛАТФОРМА УПРАВЛЕНИЯ КОМПОЗИТНЫМИ ПРИЛОЖЕНИЯМИ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ

В настоящее время существует обширный класс задач, решение которых требует использования высокопроизводительных вычислительных ресурсов. При этом в последнее время существенным интересом со стороны исследователей характеризуются задачи, обладающие высокой структурной сложностью. Примерами такого рода задач могут служить задачи моделирования сложных систем [1]. Под сложной системой в данном контексте подразумевается система, а) состоящая из большого числа компонентов; б) характеризующаяся наличием дальних связей между компонентами; в) обладающая многомасштабной изменчивостью. Характерными примерами такой системы могут служить климатическая система «океан–атмосфера» с учетом мелкомасштабной, синоптической, сезонной и межгодовой изменчивости, или эпидемиологическая система (от масштабов отдельного вируса до всемирной популяции носителей). Рациональным подходом к решению подобных задач является композитный подход, использующий существующие программные решения в качестве процедурных элементов в составе более сложного алгоритма. Во многих прикладных областях эта задача облегчается за счет возможности использования объемной информационной, алгоритмической и программной базы, разработанной ранее. Тем не менее зачастую такая база характеризуется существенной разнородностью (вследствие независимой разработки отдельных решений), неполнотой (например, в результате неполной реализации цикла разработки программного обеспечения), а зачастую неоднозначностью (в случае существования альтернативных методов, алгоритмов и/или их программных

реализаций). Следствием этой проблемы является сложность конструирования композитных программных решений на основе существующей базы программных компонентов. Другой насущной проблемой является структурное усложнение и рост неоднородности доступных сегодня разработчику программно-аппаратных платформ для высокопроизводительных вычислений: наблюдается рост популярности систем с программно-аппаратной архитектурой, отличной от традиционных суперкомпьютерных решений (ПЛИС, GPGPU); объединение вычислительных ресурсов (в том числе посредством сети Интернет) в единые вычислительные комплексы, зачастую отличающиеся сложной динамической топологией. Таким образом, актуальной становится разработка средств унификации доступа к разнородным вычислительным ресурсам и автоматизации процесса управления вычислениями в такой неоднородной среде. Наиболее остро эта проблема встает в случае организации работы с динамическими композитными приложениями: возникает необходимость отображения композитного приложения на множество вычислительных сервисов в составе используемой вычислительной инфраструктуры таким образом, чтобы построенное отображение характеризовалось наилучшей производительностью в ходе решения поставленной задачи. В рамках решения этой задачи необходимо обеспечение не только технологической, но и информационной интеграции используемых вычислительных ресурсов. Такая интеграция, в свою очередь, позволяет обеспечить: а) непрерывность потока исполнения в рамках композитного приложения при условии автоматизации промежуточных процессов (конвертирования и анализа данных, управления коммуникационной средой и т.п.); б) информационную поддержку пользователя в процессе конструирования и исполнения композитного приложения. При этом наилучшим решением является то, которое максимально изолирует пользователя от технических сложностей вычислительного эксперимента. Идеальная система взаимодействует с пользователем на языке соответствующей предметной области, автоматизируя все технические процессы на основании имеющихся требований.

В данной работе рассматривается структура и базовые принципы работы платформы, за счет привлечения экспертных знаний, обеспечивающей: а) интеллектуальную поддержку процесса конструирования композитных приложений; б) построение оптимального отображения построенного композитного приложения на доступные вычислительные ресурсы; в) реализацию выполнения приложения на выбранных ресурсах с максимальной изоляцией пользователя от технических особенностей этого процесса.

Использование базовой концепции iPSE. Разрабатываемая платформа реализуется на базе концепции iPSE (Intelligent PSE) [3], обеспечивающей должный уровень информационной и технологической интеграции композитных приложений, унифицированный доступ к разнородным высокопроизводительным инфраструктурам, а также интеллектуальную и инструментальную поддержку пользователя в процессе организации и проведения вычислений. Концепция iPSE строится на развитии логики PSE и определяет принципы построения сред для моделирования сложных систем в форме интеллектуальной оболочки управления параллельными вычислительными процессами в распределенной иерархической среде, включающей в себя вычислительные системы различной архитектуры. Такой подход расширяет достоинства традиционных PSE за счет обеспечения эффективного параллельного исполнения композитных приложений в силу того, что использует для управления параллельными вычислениями симбиотические знания об особенностях предметной области и специфике вычислительного процесса:

1. В рамках iPSE формализуются не только методы и вычислительные алгоритмы, но и экспертные знания об организации процесса решения задачи в рассматриваемой предметной области средствами компьютерного моделирования.

2. iPSE предоставляет единый интерфейс взаимодействия для разнородных предметно ориентированных программных модулей и компонентов.

3. iPSE изначально ориентирована на поддержку высокопроизводительных вычислений на базе платформ с различной

архитектурой. При этом централизованное управление эффективностью выполнения сценария позволяет избежать конфликтных ситуаций при разделении ресурсов между различными вычислительными модулями и разными пользователями.

Программная система, построенная на базе концепции iPSE, может быть отнесена к классу интеллектуальных систем, обладающих сложной распределенной структурой (структурная сложность), характеризующихся широким спектром решаемых задач (функциональная сложность), а также учитывающих особенности работы в условиях неопределенности (информационная сложность).

Использование экспертных знаний. Управление разнообразными вычислительными ресурсами, оптимальное построение композитных приложений для решения поставленной задачи, организация информационной и технологической поддержки пользователя в значительной степени базируются на экспертных знаниях. Для эффективной реализации указанных процессов требуется использование соответствующих механизмов формализации и автоматического использования знаний. В рамках концепции iPSE можно выделить два базовых класса знаний, отличающихся своей ролью в процессе работы программного комплекса: предметно-зависимые знания о соответствующей области, к которой принадлежит решаемая задача, и технологические знания о характеристиках и особенностях использования программных и аппаратных средств.

На сегодняшний день существует множество способов описания знаний: правила формальной логики, фреймовые структуры, семантические сети и пр. В рамках концепции iPSE предлагается использовать онтологический подход [4], ориентированный на детальную формализацию избранных областей знаний. Указанный подход в настоящее время отличается высокой популярностью в качестве инструмента формализации описательных знаний о предметных областях. Кроме того, онтологическая структура традиционно представляет собой динамически расширяемое описание, включающее в свой состав

множество отдельных онтологий, что позволяет в рамках предлагаемого подхода осуществить динамическую модификацию доступной базы знаний.

Формируя структуру онтологии, необходимо учитывать ряд особенностей, как проистекающих из требований, определяемых объектом исследования (композитивными высокопроизводительными приложениями с динамической компонентной архитектурой на основе прикладных сервисов), так и определяемых необходимостью универсализации структуры онтологии в предположении о возможности ее адаптации для задач произвольной предметной области. При этом производится декомпозиция онтологии на несколько базовых уровней, различающихся по степени абстракции, с которой рассматривают анализируемое композитное приложение:

1. *Уровень метаописания* представляет собой структурированные знания: а) о процессе функционирования всей системы, целиком описанной на всех уровнях; б) о многоуровневых моделях производительности, учитывающих все уровни абстракции, способах их синтеза и методах использования; в) о способах анализа протекающих в суперкомпьютерном приложении процессов.

2. *Абстрактное описание*. На данном уровне композитное приложение и его компоненты рассматриваются как средства реализации определенных методов компьютерного моделирования, обладающих соответствующим набором параметров.

3. *Реализация*. Промежуточное описание, связывающее вычислительный модуль, доступный посредством сервиса с индивидами классов, находящихся на уровне абстрактного описания. Совокупность индивидов данного уровня определяет структуру доступных вычислительных модулей, а также допустимые процедуры работы с ними.

4. *Ресурс*. Наиболее низкоуровневое описание сервисов, предоставляющее информацию о конкретных вычислительных сервисах, их характеристиках, платформах их исполнения. Этот уровень описания независим от предметной области, поскольку

основное множество его классов соответствует понятиям из области информационных технологий.

Работа с композитными приложениями. Реализуя работу с композитными приложениями в рамках концепции iPSE, интеллектуальная платформа обеспечивает поддержку основных этапов построения композитного приложения. На начальном этапе пользователь описывает постановку задачи в терминах своей предметной области, создавая таким образом семантическое описание композитного приложения, или *meta-workflow* (MWF), без указания условий ее выполнения в распределенной вычислительной среде. Второй этап проектирования состоит в переходе к описанию в терминах *абстрактного workflow* (AWF). AWF отличается от MWF тем, что оперирует конкретными методами, допускающими реализацию с использованием доступных вычислительных сервисов. Тем не менее в рамках AWF эти методы не ассоциируются с какими-либо вычислительными сервисами и определяются преимущественно в терминах предметной области. Выбранные методы подбираются посредством поиска на основе пользовательских критериев (требований к функциональным характеристикам, ресурсоемкости, точности и пр.) Третьим этапом проектирования является детализация *workflow* до уровня *конкретного workflow* (CWF) с целью оптимизации структуры и характеристик композитного приложения по времени выполнения. Базовыми процедурами построения CWF являются: выбор конкретных вычислительных сервисов, выбор эффективных способов распараллеливания по каждому из сервисов (внутренний параллелизм), выбор эффективных способов распараллеливания на уровне композитного приложения, динамическая модификация структуры CWF (в том числе миграция задач) и пр. Описание приложения в форме CWF эквивалентно алгоритмической записи, что позволяет автоматически генерировать сценарий исполнения, который может быть использован непосредственно для запуска приложения под управлением интеллектуальной платформы.

Управление вычислительной инфраструктурой. Одной из ключевых подсистем разрабатываемой платформы является

подсистема управления распределенными платформами, доступными программному комплексу. Основными ее задачами являются организация эффективного унифицированного доступа к различным программно-аппаратным платформам и множеству вычислительных сервисов, выполняющихся на этих платформах, а также планирование процесса вычислений. Архитектура данной подсистемы и ее связь с остальными логическими подсистемами в рамках концепции iPSE представлена на рисунке.

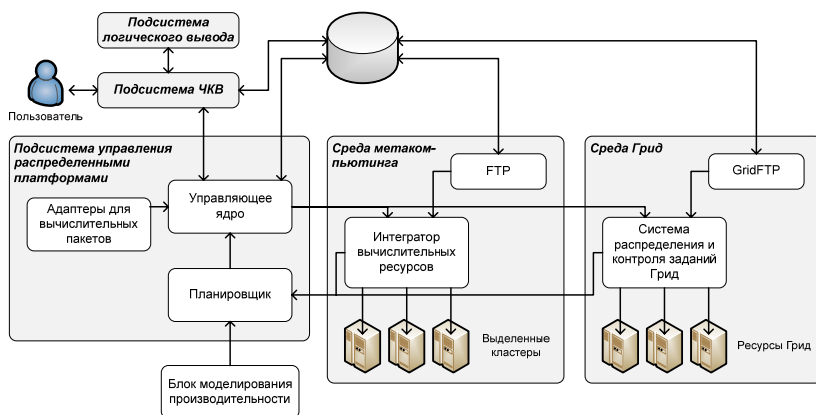


Рис. Функционирование подсистемы управления распределенными платформами в рамках концепции iPSE

Управляющее ядро подсистемы управления распределенными платформами получает описание решаемой пользователем задачи в виде абстрактного WF, контролирует исполнение элементов, входящих в его состав, и предоставляет возможности мониторинга процесса вычислений. Кроме того, управляющее ядро при помощи адаптеров вычислительных пакетов формирует корректные входные файлы для выбранных пакетов на основе данных, предоставленных пользователем.

Планировщик получает информацию о вычислительных ресурсах, их загрузке, выполняющихся задачах, оценивает посредством моделей производительности время выполнения за-

дания на каждом вычислительном ресурсе, и с использованием возможностей блока моделирования производительности принимает решение, какое задание на каком ресурсе запустить.

В соответствии с решением, выработанным планировщиком, управляющее ядро осуществляет непосредственный запуск задач, а также контроль их исполнения и постобработку. В том числе реализуются следующие функции:

- возможность формирования CWF на основе полученного AWF и дополнительных параметров предметной области;
- автоматическое определение свободных вычислительных ресурсов оптимальных для указанной задачи (т.е. позволяющих минимизировать время счета);
- расчет приблизительного времени вычисления для указанной задачи;
- формирование параметров запуска (команды запуска, скриптов, входных файлов) для указанной задачи;
- запуск решения задачи на выбранных ресурсах;
- предоставление внутрисистемного интерфейса для получения сообщения о завершении процесса вычислений (успешном или неуспешном);
- выдача информации о состоянии задачи.

В результате проведенных работ была разработана интеллектуальная платформа управления композитными приложениями в распределенных вычислительных средах, обеспечивающая:

- а) эффективное использования экспертных знаний в процессе управления распределенными вычислениями с целью повышения производительности реализуемых композитных приложений;
- б) поддержку процесса поэтапного конструирования структуры композитного приложения в соответствии с требованиями пользователя и критериями эффективности, предъявляемыми к высокопроизводительным приложениям;
- в) интеграцию разнородных вычислительных ресурсов и их эффективное применение в ходе выполнения композитных приложений.

Разработанная платформа была реализована в рамках концепции iPSE, что позволяет рассматривать ее как технологическую базу для практической реализации проектов на основе данной концепции.

В ходе работ были проведены испытания разработанной платформы на базе вычислительных инфраструктур, работающих в рамках моделей метакомпьютинга и Грид. Результаты испытаний показали возможность эффективного использования разработанной платформы в качестве надстройки над вычислительными ресурсами, доступными в рамках данных моделей.

Список литературы

1. Boccara N. Modeling Complex Systems. // Springer New York, 2004. — 397 p.
2. Rice J.R., Boisvert R. F. From Scientific Software Libraries to Problem-Solving Environments // IEEE Computational Science & Engineering. – 1996. – Vol. 3. No. 3. – P. 44–53.
3. Бухановский А.В., Ковальчук С.В., Марьин С.В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. высших учебных заведений. Приборостроение. – 2009. – № 10. – С. 5–24.
4. What Are Ontologies, and Why Do We Need Them? / B. Chandrasekaran, J.R. Josephson, V.R. Benjamins // IEEE Intelligent Systems. 1999. – Vol. 14, No. 1. – P. 20–26.

¹А.Г. Масич, ¹Г.Ф. Масич, ¹Р.А. Степанов, ²В.А. Шапов

¹Институт механики сплошных сред УрО РАН, г. Пермь

²Пермский государственный технический университет

СКОРОСТНОЙ И/О-КАНАЛ СУПЕРВЫЧИСЛИТЕЛЯ И ПРОТОКОЛ ДЛЯ ОБМЕНА ИНТЕНСИВНЫМ ПОТОКОМ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Общие сведения об эксперименте

Экспериментальная установка (ЭУ) использует метод PIV (Particle Image Velocimetry) [4] – оптический метод измерения полей скорости жидкости или газа в выбранном сечении потока. Принцип метода: импульсный лазер создает тонкий световой