

Подходы к конструированию интеллектуальных высокопроизводительных композитных приложений для моделирования сложных систем

С.В. Ковальчук, С.В. Марьин, А.В. Дунаев, А.В. Бухановский

*НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного
университета информационных технологий, механики и оптики*

sergey.v.kovalchuk@gmail.com, sergey.maryin@gmail.com,

anton.dunaev@gmail.com, avb_mail@mail.ru

Сегодня достаточно обширный класс среди задач, требующих высокопроизводительных вычислений, составляют задачи моделирования сложных систем (complex systems simulation). Под сложной системой подразумевается система, которая (а) состоит из большого числа компонентов; (б) допускает «дальние» связи между компонентами; (в) обладает многомасштабной (в том числе пространственно-временной) изменчивостью [1]. Понятие сложной системы отражает не объективную сложность реального объекта моделирования, а, скорее, структурную сложность и уровень детализации сопоставляемой ему описательной модели. Например, классическая задача прогноза погоды [2] не требует описания сложной системы, в то время как климатическая система («ансамбль погод») с учетом мелкомасштабной, синоптической, сезонной и межгодовой изменчивости составляющих ее процессов, сложной системой является. Аналогично, крайне ресурсоемкие расчеты из первых принципов характеристик атомно-молекулярных систем, состоящих из сотен и тысяч атомов [3], сами по себе к моделированию сложных систем не относятся. Однако их рассмотрение в процессе конструирования макрообъектов (материалов, процессов, устройств) с заданными характеристиками задает сложную нанотехнологическую систему. Поведение сложных систем в силу перечисленных выше особенностей затруднительно изучать посредством физического эксперимента. Потому основным способом их исследования в настоящее время является вычислительный эксперимент, выполняемый обычно на суперкомпьютерах.

Компьютерная реализация моделей сложных систем имеет свои специфические особенности. Так, большое число компонентов системы отражается на ресурсоемкости вычислительных процедур и повышенных требованиях к объемам оперативной памяти для хранения структур данных. Наличие «дальних» связей критически сказывается на возможностях экстенсивного распараллеливания вычислений формальными методами и требует использования специфических алгоритмов декомпозиции. Наконец, учет дальних связей в системе требует предварительного решения проблемы связности, затрудняющей или даже приводящей к невозможности распараллеливания формальными методами. Дополнительно, многомасштабность сложных систем требует использования для их описания комплекса параметрически связанных моделей в смежных диапазонах изменчивости [4]. Как следствие, создание инструментария вычислительного эксперимента со сложными системами требует не только разработки оптимальных (для заданной вычислительной архитектуры) параллельных алгоритмов, но и построения иерархических схем организации вычислений. Такие схемы определяют процесс взаимодействия между одновременно или последовательно исполняемыми вычислительными модулями, каждый из которых, в свою очередь, может выполняться параллельно на одном или нескольких вычислительных комплексах различных архитектур.

Таким образом, с точки зрения задачи оптимизации параллельной производительности, высокопроизводительные программные комплексы моделирования сложных систем сами могут интерпретироваться как сложные системы с присущими им свойствами. Это естественным образом затрудняет процесс проектирования и разработки такого программного обеспечения. По-видимому, в настоящее время проблема проекти-

рования, разработки и внедрения высокопроизводительных программных комплексов данного класса не имеет устоявшегося решения; процесс разработки не автоматизируется и жестко ограничен спецификой соответствующей предметной области.

С другой стороны, в настоящее время достаточной популярностью в применении к построению бизнес-приложений характеризуется класс композитных приложений, использующих для решения поставленной задачи набор готовых программных модулей [5, 6]. В то же время данный класс приложений использует подходы, применение которых будет не менее эффективным при разработке программных комплексов, решающих ресурсоемкие научные задачи, в том числе, задачи моделирования сложных систем. Для решения задач моделирования таких систем зачастую приходится организовывать взаимодействие уже существующих вычислительных модулей в составе единого программного комплекса. Использование готовых вычислительных модулей может быть обосновано рядом причин:

- повторное использование кода позволяет избежать дополнительных затрат на написание соответствующих программных модулей;
- программные модули, отвечающие за моделирование отдельных элементов сложной системы, могут быть слишком трудоемки для реализации в рамках текущего проекта;
- использование существующих программных решений может быть продиктовано требованиями технического задания, предписывающего разработчикам использовать конкретные программные решения (например, в силу их сертифицированности).

Тем не менее, ввиду высокой вычислительной сложности решаемых задач, разрабатываемый программный комплекс, использующий сторонние вычислительные модули, должен отвечать требованиям эффективности по использованию параллельных вычислительных ресурсов платформы, на которой производится исполнение. Достижение поставленной цели возможно при условии обеспечения масштабируемости разрабатываемого программного комплекса как цельного программного решения. Для этого необходимо (а) обеспечить наиболее эффективное использование возможностей распараллеливания, предоставляемых каждым из используемых программных модулей, (б) добиться наиболее эффективного функционирования совокупности программных модулей как единого программного комплекса, решающего конкретную задачу предметной области.

Специфика использования сторонних вычислительных модулей

Использование сторонних вычислительных модулей приводит к необходимости учитывать особенности каждого из них в процессе интеграции в состав программного комплекса. К таким особенностям следует отнести:

- *Ограничения предметной области*, очерчивающие круг задач, для решения которых подходит данное программное средство.
- *Особенности используемых технических решений*: язык программирования; аппаратная платформа и операционная система, на которой изначально предполагалось функционирование модуля; множество используемых системных и прикладных библиотек; используемые технологии распараллеливания; способ вызова и использования.
- *Алгоритмические особенности*, характерные как, в общем, для используемого алгоритма, так и для конкретной его реализации.
- *Возможность модификации исходных текстов программного средства* может быть ограничена рядом причин: наличием лицензионных ограничений, сертифицированностью данной реализации, сложностью для понимания структуры исходного

текста, отсутствием в команде разработчиков специалистов по данному языку или технологии и т.п.

- *Юридические, политические и другие внешние ограничения*, накладываемые на данное программное средство.

Для многих программных модулей указанные особенности проистекают из того, что первоначально их разработка преследовала цели решения конкретной задачи, поставленной перед специалистами-разработчиками предметной области. Как следствие, зачастую специалистам, занимающимся организацией повторного использования различных программных средств, приходится сталкиваться с большим разнообразием языков программирования, стилей написания исходных текстов (если предполагается возможность их модификации), платформ исполнения и т.п.

Дополнительную сложность может представлять необходимость использования в составе программного комплекса вычислительных модулей, изначально разработанных только для внутреннего использования специалистами предметной области. В этом случае может потребоваться реализация внешних по отношению к программному модулю механизмов обработки ошибок, контроля стабильности работы программного средства и т.п.

Каждый из используемых вычислительных модулей, как указывалось выше, может использовать свою технологию распараллеливания и свой вариант ее применения к конкретному алгоритму. При использовании же вычислительного модуля в составе программного комплекса возникает задача обеспечения максимальной эффективности распараллеливания программного модуля с учетом особенностей программно-аппаратной платформы, используемой для ее запуска. Одним из способов решения данной задачи является использование моделей, позволяющих оценить параллельную производительность вычислительных модулей до их непосредственного запуска. Простейшим примером аналитической модели, позволяющей оценить эффективность параллельной реализации алгоритма, является закон Амдала. Но для более эффективного анализа параллельной производительности алгоритма необходимо использовать систему моделей, учитывающих:

- актуальные особенности программно-аппаратной платформы, на которой производится запуск приложения;
- параллельную структуру исследуемого алгоритма;
- характеристики входных данных для запуска программы.

Учитывая указанные параметры, можно оценить время выполнения приложения в заданных условиях в зависимости от параметров параллельного запуска. Более того, по результатам моделирования можно определить оптимальные параметры распараллеливания, позволяющие достичь наилучшего времени работы и загрузки вычислительных ресурсов.

Интеллектуальные технологии интеграции вычислительных модулей

Для успешной работы вычислительного модуля в составе программного комплекса необходимо обеспечить унифицированный интерфейс доступа к нему со стороны управляющей подсистемы. Использование концепции сервисно-ориентированной архитектуры [7] позволяет добиться организации унифицированного доступа к вычислительным модулям, что, в свою очередь, существенно упрощает процесс их интеграции. Таким образом, доступ к каждому вычислительному модулю осуществляется посредством сервиса, осуществляющего запуск и контроль работы вычислительного модуля, организацию потоков входных и выходных данных, вторичную обработку ошибок, произошедших в процессе работы вычислительного модуля.

Использование сервисно-ориентированной архитектуры позволяет обеспечить эффективную работу с отдельными вычислительными модулями. Тем не менее, для

обеспечения эффективного функционирования комплекса в целом необходим совместный анализ характеристик вычислительных модулей, целью чего является определение оптимальных параметров запуска. При этом важно учитывать ряд факторов, влияние которых возрастает с переходом к задаче комплексного анализа системы вычислительных модулей:

- Возможность взаимозаменяемости модулей аналогичной функциональности. Данная возможность может быть актуальной, когда один из пакетов позволяет проводить расчеты, которые невозможно выполнить альтернативным пакетом, а также в случае обеспечения одним из пакетов лучшего времени расчетов или точности для данных условий запуска.
- Необходимость обработки данных, передаваемых управляющей системой от одного пакета другому, с учетом их преобразования. Этот фактор важен, поскольку в условиях оценки производительности программного комплекса при решении сложной задачи, необходимо оценивать полные цепочки исполнения, включающие как этапы непосредственной работы вычислительных модулей, так и промежуточные этапы, включающие и обработку данных.

С учетом перечисленных выше факторов актуальной становится задача разработки интеллектуальных технологий интеграции вычислительных модулей, то есть технологий, активно использующих экспертные знания. В данном случае знания, в первую очередь, определяют закономерности использования предметно-ориентированных вычислительных модулей. Структурируя экспертные знания, можно выделить три категории знаний о прикладном вычислительном ПО:

- Знания о способах сопряжения с другими вычислительными модулями. Примером таких знаний может служить набор «эталонных» задач предметной области, решаемых среди прочих с использованием и рассматриваемого ПО.
- Знания о способах использования пакета для решения отдельных задач предметной области. Предполагает классификацию задач, решаемых с помощью данного ПО, с указанием соответствующих режимов и параметров запуска.
- Знания об особенностях внутреннего распараллеливания, необходимые для обеспечения эффективного использования доступных вычислительных ресурсов.

Проводя наиболее общую классификацию ролей экспертов, знания которых используются при построении процедуры интеграции вычислительных модулей, можно выделить две категории носителей знаний:

- Эксперты в области параллельных вычислений, обладающие комплексом знаний для моделирования параллельной производительности как отдельных вычислительных модулей, так и комплекса в целом. Кроме того, на основании знаний экспертов данной области производится анализ результатов моделирования с последующим принятием решения о параллельном запуске вычислительных модулей.
- Эксперты предметной области, располагающие информацией о способах использования вычислительных модулей, требованиях к входным и выходным данным, принципиальных особенностях реализованного в них алгоритма и т.п.

Очевидно, что наилучшим экспертом по специфическому проблемно-ориентированному вычислительному модулю является его разработчик. Тем не менее, во многих ситуациях процедура получения знаний непосредственно от такого эксперта является трудноосуществимой. В этих случаях наиболее эффективным источником знаний об использовании вычислительного модуля может стать его опытный пользователь. Кроме того, во многих случаях для получения качественных знаний об особенностях параллельной реализации данного программного средства необходима работа с отдельным экспертом в области параллельных вычислений.

Формализация экспертных знаний

Важной задачей при построении интеллектуальных методов управления параллельными вычислениями является формализация знаний экспертов. Одним из способов формализации, приемлемой при построении подобной базы знаний является фреймовая структура представления знаний [8], включающая в себя как знания о способах применения вычислительных модулей, так и знания об их параллельной структуре. Следует отметить, что существенный интерес при построении высокопроизводительных (а, значит, ресурсоемких) приложений представляют знания, применимые для анализа параллельной производительности. Частным случаем таких знаний является система аналитических моделей производительности, описанная выше.

Приобретение новых знаний

Процедура приобретения новых знаний обретает существенное значение с учетом динамики программно-аппаратной базы вычислительного комплекса, а также с учетом субъективности знаний, полученных от экспертов. К наиболее значимым методам приобретения новых и обновления существующих знаний можно отнести:

- встраивание новых программных модулей с одновременным расширением существующей базы знаний;
- обновление базы знаний с учетом результатов мониторинга процесса использования комплекса (переоценка программных компонентов и квалификации экспертов);
- имитационное моделирование поведения вычислительных компонентов на специальном стенде (в случаях, когда логический вывод на основе стандартных моделей затруднен).

С другой стороны, знания экспертов предметной области нуждаются в оценке (и переоценке) их достоверности. Можно предложить следующую систему коэффициентов достоверности представленных экспертом предметной области знаний:

- Самооценка эксперта предметной области. Оценивается, исходя из анкетирования эксперта в предположении честности его ответов на поставленные вопросы. Система вопросов и формула вывода коэффициента определяется экспертом в области параллельных вычислений.
- Сложность представленной экспертом модели. Как и предыдущий показатель, оценивается экспертом в области параллельных вычислений и позволяет определить компетентность эксперта предметной области в области параллельных вычислений и проделанную работу по анализу параллельной реализации вычислительных модулей.
- Непротиворечивость предоставленных знаний. Оценивается корректность результата работы построенной экспертом модели на конкретных запусках вычислительной системы.
- Устойчивость. Также оценивается на основании запусков вычислительной системы на конкретных данных. Определяет стабильность характеристик модели по отношению к вариациям входных параметров.

Использование предложенного подхода

Предложенный подход был использован при реализации прототипа программного комплекса в проекте «Высокопроизводительный программный комплекс моделирования и прогноза экстремальных гидрометеорологических явлений и расчета воздействий на морские объекты и сооружения» [9], что обеспечило эффективное решение комплексом поставленной перед ним задачи с достижением достаточно высоких показателей масштабируемости (было получено общее ускорение около 84 раз на 128 вычислителях). В настоящее время подход применяется в процессе работы над проектом «Разработка высокопроизводительного программного комплекса для квантово-механических

расчетов и моделирования наноразмерных атомно-молекулярных систем и комплексов» [10].

Выводы

Процесс использования программных модулей тесно сопряжен с извлечением знаний экспертов различных категорий. При этом необходимо учитывать, с одной стороны, специфические особенности конкретных вычислительных модулей, а, с другой стороны, следует принимать во внимание индивидуальные особенности представления знаний различными экспертами. Таким образом, для достижения оптимальной производительности при решении ресурсоемких задач моделирования сложных систем, требующих построения композитных приложений, необходима организация процесса получения, формализации, использования и модификации набора разнородных экспертных знаний.

Литература

1. Boccaro N. Modeling Complex Systems. // Springer New York, 2004. — 397 p.
2. Монин А.С. Гидродинамика атмосферы, океана и земных недр // СПб, Гимиз, 1999, 524 с.
3. Маслов В.Г. Теор. и эксп. химия, 1984, 20, №3, С.288-298.
4. Иванов С.В. Идентификация параметрически связанных моделей сложных систем. // Научно-технический вестник СПбГУ ИТМО. Выпуск 54. Технологии высокопроизводительных вычислений и компьютерного моделирования. – СПб.: СПбГУ ИТМО, 2008. – С. 100-107
5. Сипл Р. Десять правил создания композитных приложений // PC Week/RE (556) 46, 2006.
6. Граничин О.Н., Кияев В.И. Информационные технологии в управлении. Бином. Лаборатория знаний Интернет-Университет Информационных Технологий, 2008. – 336 с.
7. Lublinsky B. Defining SOA as an architectural style. 9 January 2007. <http://www.ibm.com/developerworks/architecture/library/ar-soastyle/>
8. Minsky M. A Framework for Representing Knowledge. The Psychology of Computer Vision. [ed.] Patrick Henry Winston. New York : McGraw-Hill, 1975.
9. Высокопроизводительный программный комплекс моделирования экстремальных гидрометеорологических явлений. Часть I: Постановка задачи, модели, методы и параллельные алгоритмы / А.В. Бухановский, С.В. Ковальчук [и др.] // Научно-технический вестник СПбГУ ИТМО. Выпуск 54. Технологии высокопроизводительных вычислений и компьютерного моделирования. – СПб.: СПбГУ ИТМО, 2008. – С. 56-63.
10. Высокопроизводительный программный комплекс моделирования наноразмерных атомно-молекулярных систем / В.Н.Васильев [и др.] // Научно-технический вестник СПбГУ ИТМО. Выпуск 54. Технологии высокопроизводительных вычислений и компьютерного моделирования. – СПб.: СПбГУ ИТМО, 2008. – С. 3-12.