

понятно: если коэффициент использования АУ меньше 10 % (остальные 90 % простаивают), то исправление этой ситуации есть новая математическая задача, причем зачастую далеко не простая.

Классические математические качества численных методов: погрешность аппроксимации, устойчивость, точность, экономичность, — остаются, естественно, в силе. Можно только добавить, что теперь ужесточаются требования к выполнению законов сохранения: энергии, импульса и других инвариантов решения, — т.е. максимальное наследование свойств исходных дифференциальных уравнений. Получил распространение новый термин — алгоритмы высокого разрешения, который фактически означает переход на новый уровень моделирования. Здесь большую роль играет, конечно, и количественный фактор: суперкомпьютерная память, объём которой всегда растёт пропорционально скорости выполнения операций, позволяет на порядок увеличивать число узлов расчётных сектор. А это открывает возможности «ухватывать» в численных экспериментах такие тонкие эффекты, которые и не снились исследователям лет 10—20 назад.

Конвергенция алгоритмических структур и компьютерных архитектур

Главный сложившийся принцип и даже лозунг распараллеливания — это отбраживание алгоритмов на архитектуру ЭВМ. Действительно, переделать алгоритм или программу — вопрос нескольких человеко-недель, а проектирование и создание нового компьютера занимает человеко-годы. Однако в действительности в истории вычислительных машин распараллеливание работы различных устройств (например, АУ и средства ввода-вывода данных) присутствовало изначально. Если вспомнить последнюю четверть XX века, то можно обнаружить поразительное богатство изобретательских компьютерных идей: матричные ЭВМ, специализированные процессоры, транспьютеры, экзотические архитектуры типа «шафл», гиперкуб и т.д.

Однако в начале нового века мировой рынок оказался побежден коммерчески наиболее выгодными унифицированными МКС кластерного типа, когда однотипные многопроцессорные и/или многоядерные вычислительные узлы (с общей памятью) соединены шинами в единую сеть, физические соединения между которыми осуществляются специальными коммутаторами. Работа всех устройств контролируется одним управляющим узлом, а информационные обмены между всеми узлами осуществляются тривиальной передачей сообщений, реализуемой программной системой MPI (Message Passage Interface). В общем случае используется так называемое гибридное программирование с применением распределённой и общей памяти: на основе MPI — распараллеливание «по узлам», а в каждом из них — организация многопоточковых вычислений над общей памятью.

Правда, последние годы положение стало меняться. Для быстрой визуализации изображений были разработаны графические процессорные элементы (GPU — Graphic Processor Unit) с ограниченным набором операций и большим количеством вычислительных ядер над общим кэшем (cache — сверхбыстрая оперативная память). По соотношению «производительность/цена» они намного превосходят универсальные центральные процессорные элементы (CPU — Central Processor Unit) и даже стали применяться не только для выполнения графических операций. Сами графические ускорители начали резко эволюционировать, появились новые поколения GPU с быстро развивающимся языком и соответствующей системой программирования CUDA (подмножество языка общего назначения C++), и даже появился новый термин, внутренне противоречивый по своему смыслу: GPU общего назначения, или GP GPU (GP — аббревиатура от General Purpose). Теперь одним из генеральных направлений компьютерных архитектур являются гетерогенные МКС, в которых вычислительные узлы, кроме обычных процессоров и/или ядер, содержат GPU.

Однако данный вектор развития — это частный случай концепции программируемых логических интегральных схем (ПЛИС, или FPGA — Field Programmable Gate Array), заключающейся в проектировании и автоматизированном изготовлении кремниевых кристаллов с электронной схемой, ориентированной на быструю реализацию конкретного алгоритма. Эта идея создания специализированного процессора «под задачу» витает в воздухе

уже много лет (около 30 лет назад существовал термин «кремниевый компилятор»), но только в последнее десятилетие технологические достижения делают эту проблему экономически оправданной.

Поскольку революционные преобразования, в том числе компьютерные, прогнозировать на короткий исторический срок невозможно, можно считать, что пост-петафлопные и первые экзафлопные МКС будут иметь рассмотренную гетерогенно-кластерную структуру. Очевидно также, что межузловые и даже межпроцессорные коммуникации будут слабым звеном, тормозящим общее быстрое действие суперкомпьютера.

В этой связи представляется очень естественным рассмотреть возможные взаимоотношения между структурами алгоритмов и компьютерными архитектурами. МКС может быть представлена односвязным ненаправленным графом, вершины которого соответствуют вычислительным узлам, а рёбра — информационным связям. С другой стороны, важно отметить, что главным принципом распараллеливания для многомерных задач является метод декомпозиции областей (МДО): расчётная область разбивается на подобласти, в каждой из которых формулируются подзадачи, взаимосвязанные между собой через вспомогательные краевые условия на смежных границах. Все подзадачи могут решаться одновременно на соответствующих вычислительных узлах, а решение общей задачи ищется последовательными приближениями «по подобластям», путем итерационного пересчёта смежных краевых условий, через которые осуществляются взаимосвязи подзадач (как раз эта процедура и требует коммуникаций между узлами). Очевидно, что расчётная схема МДО описывается также с помощью графа, и для идеального отображения данного алгоритма на архитектуру МКС их графы должны совпадать.

Исходная идея МДО восходит к концу XIX века и называется альтернирующим методом Шварца. Его можно представить математически с помощью матричного оператора, в котором каждый диагональный блок соответствует «своей» подобласти, а ненулевые внедиагональные блоки обозначают взаимосвязи смежных подобластей. Таким образом, граф МДО описывает матричный портрет оператора А.

Представление расчётной области совокупностью взаимосвязанных подобластей можно назвать также макросетью, в отличие от микросети, которая строится при генерации расчётной сетки, используемой в основных современных методах конечных разностей, конечных объёмов и конечных элементов (МКР, МКО, МКЭ) для приближённого решения задач математического моделирования. Важно отметить, что топологически трехмерные макро- и микросеть эквивалентны в обобщённом смысле, т.е. состоят из объектов одинаковых типов: вершины, рёбра, грани и объёмы. Системы алгебраических уравнений, которые получаются, например, после аппроксимации дифференциальных задач при использовании сеточных методов, описываются с помощью разреженных матриц, в которых диагональные элементы соответствуют узлам сетки, а внедиагональные элементы отличны от нуля только в позициях, отвечающих за взаимосвязи геометрически соседних узлов.

Таким образом, системы сеточных уравнений и их матричные «портреты» также представляются с помощью графов. А в матричной форме МДО после этапа дискретизации каждый матрично-операторный элемент превращается в матричный блок большой размерности, определяемой числом узлов в соответствующих подобластях. Если общее количество узлов в расчётной области зафиксировано, а число вычислительных узлов МКС и число подобластей растёт, то в каждой из них узлов становится меньше и меньше. В пределе мы можем получить компьютерную сеть с одним процессором на каждом узле.

Отнюдь не факт, что такая МКС будет оптимальной с коммерческой точки зрения. Скорее всего, истина находится в «золотой середине», а приведенные рассуждения призваны показать, что развитие алгоритмических структур и компьютерных архитектур — это дорога с двухсторонним движением. С точки зрения быстрого решения одной «большой» задачи, перспективным являются следующие направления эволюции архитектур:

- макросети МКС различной размерности (1-D, 2-D и трёхмерные);
- динамическая реконфигурация МКС с возможностью реализации различных сеточных шаблонов;

— быстрые ближние связи с синхронизацией обменов и вычислений.

Всё это — простейшие иллюстрации к очень разнообразным вопросам оптимизации МКС. Можно добавить, что если рассматривается общая производительность ЦОД коллективного пользования, то сама постановка проблемы значительно меняется и требует учета распределения глобальных ресурсов между потоками задач.

Концепция базовой системы моделирования

Традиционные формы программного обеспечения математического моделирования — это библиотеки алгоритмов и пакеты прикладных программ (ППП), ориентированные на решение конкретного класса задач определенным кругом пользователей, которые определяют требования к интерфейсу и функциональным характеристикам. Примерами таких успешных коммерческих продуктов являются ANSYS, FEMLAB, NASTRAN, MATLAB и многие другие. Слабым местом этих крупных разработок, имеющих свою длительную историю, является инерционность жёстких структур данных и программных технологий, делающих неясными перспективы их перехода к новым стратегиям массивного распараллеливания на базе МКС с миллионами процессоров и ядер.

Для выхода из фактически сложившейся кризиса прикладного программирования кардинальным представляется переход от конкретных классов задач к ориентации на общедоступные наборы инструментов, поддерживающих различные технологические этапы моделирования (генераторы сеток, аппроксиматоры уравнений, алгебраические решатели и т.д.), взаимодействующих между собой на основе разработки согласованных структур данных (геометрических, сеточных, алгебраических, графических). Такой модульный принцип позволит осуществить независимую реализацию, развитие и переиспользование продуктов коллективов разработчиков, а также их адаптацию к непрерывно появляемым платформам МКС.

Очень важно, что здесь изначально могут быть заложены гибкие стратегии и тактики параллелизма на различных стадиях вычислительного эксперимента. Кроме того, эта модель ориентирована на сборку конкретных приложений из представительного набора программных блоков наподобие детского конструктора LEGO, что должно обеспечить такие трудно совместимые категории, как универсальность и эффективность. Разумеется, достижение этой цели требует координации усилий вычислительного сообщества, но примеры успешного сотрудничества такого рода уже имеются: в области линейной алгебры общепринятыми являются форматы и стили пакетов BLAS, SPARSE BLAS, а также коллекции типовых матриц для сравнительного тестирования.

В качестве унифицированной технологической среды решения широкого круга прикладных математических задач можно рассмотреть Базовую Систему Моделирования (БСМ), состоящую из следующих основных компонентов, охватывающих все главные этапы крупномасштабного вычислительного эксперимента, сформированные около 30 лет назад в работах Г.И. Марчука, А.А. Самарского, Н.Н. Яненко и их учеников:

— система геометрического и функционального моделирования, реализующая входной пользовательский интерфейс и формирующая в результате множественные структуры геометрических и функциональных данных (ГСД и ФСД);

— библиотека генераторов квазиструктурированных сеток, работающая на основе анализа ГСД, а также ФСД, и формирующая, в свою очередь, сеточную структуру данных (ССД), которая полностью и однозначно опеределает итоги дискретизации расчётной области, включая взаимосвязи сеточных объектов с функциональными;

— система программной реализации различных аппроксимационных алгоритмов исходной задачи, на основе уже построенных ССД и ФСД, результатом исполнения которой является алгебраическая структура данных (АСД), представляющая дискретизованную модель исходной задачи;

— библиотека алгебраических методов, осуществляющая на базе ССД решение систем линейных или нелинейных алгебраических уравнений, проблемы собственных значений или вычисление рекуррентных последовательностей в нестационарных задачах;

— постобработка и визуализация результатов моделирования, ориентированная на

совокупность конкретных способов представления сеточных функций в рамках БСМ и использующая имеющиеся в широком распространении внешние графические редакторы, а также системы автоматизации проектирования (CAD, CAM, CAE, PLM);

— набор оптимизационных процедур для решения обратных задач идентификации параметров модели (условная минимизация целевых функционалов) на основе реализации последовательности прямых задач; средства управления сложными вычислительными процессами.

Данные компоненты являются самодостаточными, автономными и могут разрабатываться независимыми группами специалистов в различных алгоритмических областях на основе согласования множества информационных форматов. Структура создаваемых программных компонент предусматривает гибкое взаимодействие с внешними прикладными и системными разработками, в том числе — включение в библиотеки БСМ алгоритмических процедур других авторов, для чего создается совокупность конверторов (переходников) структур различных форматов данных. Помимо расширяемого набора вычислительных инструментов, БСМ включает средства конфигурации (сборки) проблемно-ориентированных пакетов прикладных программ для конкретных производственных областей.

По своей архитектуре БСМ представляет собой интегрированную среду вычислительно-информационных инструментов для сообщества прикладных математиков и программистов, которые могли бы не только использовать готовые компоненты для своих проектов, но и привносить инновации в общую копилку. Эта концепция широкой скоординированной кооперации естественно согласуется с X-STACK — так названо программное обеспечение экзамазтабного назначения в «дорожной карте» IESР.

Описанные принципы построения базовой системы моделирования касаются только содержательных аспектов. При этом многие актуальные вопросы требуют дальнейших исследований: поддержка коллективной разработки, сопровождения, эксплуатации и развития программных комплексов с длительным жизненным циклом, обеспечение крупномасштабируемого параллелизма на МКС экстремальной производительности, оптимизация программного кода на алгоритмическом и системном уровне, организация массовых экспериментов на центрах коллективного пользования по технологиям облачных вычислений.

Об авторе:

Валерий Павлович Ильин — главный научный сотрудник Института вычислительной математики и математической геофизики СО РАН, доктор физико-математических наук, профессор, ведущий специалист в области прикладной математики и информатики, автор более 300 научных работ (из них 12 монографий), обладатель трёх авторских свидетельств.

Основные научные результаты В.П. Ильина: экономичные прямые и итерационные алгоритмы решения систем алгебраических уравнений и проблемы собственных значений, компактные методы конечных разностей, конечных объёмов и конечных элементов повышенной точности для аппроксимации смешанных краевых задач, эффективные многосеточные методы и декомпозиция областей для решения многомерных уравнений математической физики, вычислительно-информационные технологии математического моделирования в электрофизике и других прикладных областях, распараллеливание алгоритмов и их отображение на архитектуру многопроцессорных вычислительных систем.

Под его руководством в течение многих лет выполняются работы по грантам РФФИ и РАН, международным и региональным проектам. В.П. Ильин многократно участвовал в международных конференциях, читал лекции по приглашениям ведущих зарубежных университетов и научных центров.

С 1964 года ведет преподавательскую работу в Новосибирском государственном университете и является профессором кафедры «Вычислительная математика». 35 его учеников защитили кандидатские диссертации, шесть из них стали докторами наук.

В.П. Ильин — член редколлегии «Сибирского журнала вычислительной математики», журнала «Прикладная физика» и «Сибирского журнала индустриальной математики». В.П. Ильин является руководителем проекта РФФИ №11-01-00205.