

Разделенный подход при решении связанных задач FSI на гибридных вычислительных системах *

С.П. Копысов, И.М. Кузьмин, В.Н. Рычков, Л.Е. Тонков

Институт механики УрО РАН

В статье рассматривается создание прикладного программного интерфейса для решения сопряженных задач взаимодействия конструкции с жидкостью или газом и выделением самостоятельного приложения для синхронизации и обмена данными. Обсуждаются алгоритмы синхронизации решений отдельных физических задач, интерполяции данных между ними и деформирования расчетных сеток. На численных примерах показывается возможность решения ряда задач на кластерах с графическими процессорами.

1. Введение

При решении сопряженных задач взаимодействия потока газа и деформируемой конструкции (FSI) в рамках разделенного подхода, согласование получаемых решений на границах подобластей является неотъемлемым и весьма важным этапом, существенно влияющим на адекватность получаемых результатов. Под согласованием понимается, прежде всего, совокупность операций по обмену результатами вычислений между соответствующими решателями физических подзадач, интерполяции значений между узлами расчетных сеток, деформирование сетки газодинамической области и, возможно, реализация семейства итерационных процедур при жестком связывании. Актуальным является построение параллельных алгоритмов как решения отдельных физических подзадач динамики деформирования и газодинамики, так и непосредственно решающих всю совокупность задач сопряжения (интерполяция, деформирование расчетных сеток, итерационное связывание) и создание прикладного программного интерфейса с выделением самостоятельного приложения для синхронизации и обмена данными, в котором реализуются отдельные подзадачи связывания. Предлагаемый подход к моделированию задач взаимодействия потока газа и деформируемой конструкции/тела основан на выделении этапа согласования и взаимодействий в отдельную программную подсистему, не зависящую от используемых реализаций расчетов, газодинамических полей и напряженно-деформированного состояния. Таким образом, может быть обеспечено взаимодействие для произвольных комбинаций программ решения задач газовой динамики и механики деформирования. В этом случае сопрягаются различные типы сеток, разные виды аппроксимаций. Точность аппроксимации уравнений сохраняется как в областях расчета газодинамики и механики деформирования, так и на границе взаимодействия. В рассматриваемом случае, система уравнений газовой динамики решается произвольным лагранжево-эйлеровым конечно-объемным методом в библиотеке OpenFOAM [1], а на этапе решения задач деформирования применяется конечно-элементный пакет FEStudio [2]. Оба пакета написаны в рамках объектно-ориентированного подхода, что существенно облегчает их совместное использование.

2. Программная модель связывающего приложения для решения сопряженных задач

Процедура решения каждой из задач основывается на максимально полном использовании ранее созданного кода, предназначенного для решения отдельных физических подзадач. При этом каждое отдельное приложение использовало свою модель параллелизма,

*Работа выполнена при финансовой поддержке РФФИ (14-01-00055-а, 14-08-00064-а).

независимую параллельную реализацию численных методов и алгоритмов и свои требования к среде параллельного выполнения. Разработанное программное обеспечение позволяет осуществить решение поставленных FSI задач не нарушая указанных ограничений и с минимальными изменениями исходного кода имеющихся приложений.

Связывающее приложение реализуется в соответствии с моделью «клиент-сервер» на объектно-ориентированном промежуточном ПО Ice ZeroC [3]. Ice поддерживает модели синхронных и асинхронных вызовов. В последнем случае клиент, вызывая объект при помощи прокси, наряду с обычными параметрами операции, передает прокси специальный callback-объект. После вызова прокси управление сразу возвращается клиенту. При завершении вызова объекта, среда выполнения Ice на клиентской стороне вызывает метод callback-объекта, передавая ему результаты вызова или исключения. С точки зрения сервера, обработка синхронных и асинхронных вызовов выглядит абсолютно одинаково. Клиент может и не знать, с помощью какой модели осуществляется обработка его вызова.

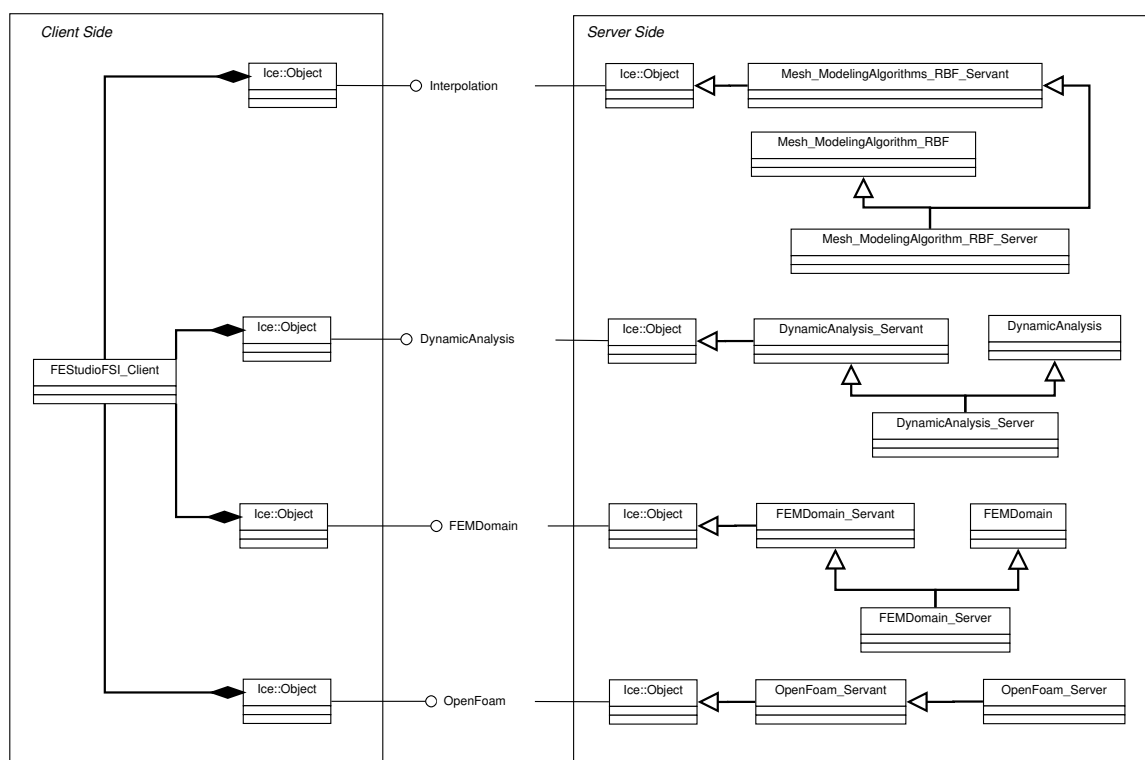


Рис. 1. Программная модель связывающего приложения

Для работы распределенной модели клиент приложения содержит информацию о прокси четырех серверов (рис. 1). Каждый прокси обладает уникальным именем, которое служит идентификатором при обращении к соответствующему серверу. Со стороны OpenFOAM инициализируется сервер содержащий информацию об алгоритме решения, текущем состоянии сетки и решения. Со стороны FESudio инициализируются два сервера: FEMDomain — содержит информацию об расчетной области и ее состоянии и DynamicAnalysis — отвечает за алгоритм решения динамической задачи деформирования. На отдельном узле инициализируется сервер Interpolation, отвечающий за интерполяцию данных между решателями отдельных подзадач. Вызов методов, реализованных со стороны серверов осуществляется посредством созданных прокси. Кроме того, клиент хранит и обновляет вектора перемещения и давления, для их последующего использования в расчете и управляет логикой работы всей программы. Описанный механизм связывания объектно-ориентированной модели и инфраструктуры Ice позволяет, путем замены объектов, получить распределенную объектно-ориентированную программу. Переход к распределенным/параллельным вычис-

лениям осуществляется при минимальной модификации независимых приложений. Подход обеспечивает наибольшую универсальность и позволяет объединять независимые приложения от которых требуется лишь использование соответствующего прикладного программного интерфейса.

3. Параллельные алгоритмы выполняемые на связывающем приложении

Каждая из частей сопряженной задачи – газодинамика (CFD) и динамическая задача деформирования (CSD) решаются независимо, на своей сетке. Задачи решаются на гибридных вычислительных узлах, что позволяет максимально задействовать вычислительные мощности. Решение газодинамической задачи осуществляется на основе библиотеки OpenFOAM. Для ускорения решения системы линейных алгебраических уравнений используется собственная библиотека CUSMA [4] основанная на технологии CUDA. Задача CSD решается методом конечных-элементов реализованном в пакете FESstudio [2]. В этом пакете реализована поддержка технологий OpenMP и CUDA, которые используются на различных этапах решения: формирование элементных векторов нагрузок и внутренних сил, решение системы уравнений.

В разделенном подходе решения сопряженных задач необходимо интерполировать вектор нагрузок и перемещения сопряженной границы задач CSD и CFD. Интерполяция нагрузок происходит по общей границе расчетных областей. При интерполяции перемещений данные необходимо интерполировать с границы расчетной области CSD на всю область CFD. Это связано с тем, что задача динамики деформируемого твердого тела решается в отчетной конфигурации, а газодинамики — в текущей. Как следствие, возникает необходимость в деформировании газодинамической сетки, то есть приведение ее к текущей конфигурации. Таким образом обе задачи (интерполяция нагрузок и деформирование сетки) сводятся к интерполяции расчетных данных.

Связывающее приложение отвечает за алгоритм сопряженной задачи, синхронизацию решений задач газодинамики и динамической задачи деформирования и интерполяцию расчетных данных между ними. Для решения задачи интерполяции рассматривали бессеточные методы: метод Шепарда и метод на основе радиальных базисных функций (РБФ). Рассмотрим этап деформирования сетки и его распараллеливания, так как он является наиболее затратным по времени этапом решения сопряженной задачи [5].

Обозначим через s – узлы сетки динамической задачи деформирования твердого тела, а через f – узлы сетки задачи газодинамики. В соответствии с методом РБФ неизвестные коэффициенты интерполянта находятся, как решение системы уравнений:

$$\begin{bmatrix} \Phi_{ss} & B_s \\ B_s^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} W_s \\ 0 \end{bmatrix}, \quad (1)$$

где Φ_{ss} матрица из элементов $\phi(\|x_{s_i} - x_{s_j}\|)$, B_s матрица со столбцами $[1 \ x_{s_j}]$, и μ – коэффициенты полинома q . Тогда искомые значения интерполяции W_f для газодинамической сетки получаются из матрично-векторного произведения:

$$W_f = [\Phi_{fs} B_f] \begin{bmatrix} \lambda \\ \mu \end{bmatrix} \quad (2)$$

где Φ_{fs} матрица из элементов $\phi(\|x_{f_i} - x_{s_j}\|)$, B_f матрица со столбцами $[1 \ x_{f_j}]$.

Таким образом, задача интерполяции данных сводится к решению системы линейных алгебраических уравнений с матрицей, разреженность которой зависит от вида радиальных базисных функций. Решение системы осуществлялось с помощью библиотеки CUSMA [4].

Оценка вычислительных затрат параллельных алгоритмов производилась на нескольких тестах. Первый тест – движение сетки при повороте недеформируемой пластины, находящегося в центре квадратной области. Граничные узлы свободно перемещаются вдоль границ. Хотя, сам алгоритм не использует структуру связности узлов в ячейках сетки, допустимые деформации определяются с учетом топологии ячейки. Предельные значения скошенности были получены при повороте на 180° методом радиальных базисных функций с $\phi(\|x\|) = (1 - \|x\|)^3(1 + 3\|x\|^3)$.

Следующий тест представляет взаимодействие потока газа с открывающейся заслонкой. Поворот заслонки осуществлялся вокруг оси O_x . Максимальный угол поворота методом Шепарда составил 10° . Вырожденные элементы появляются вблизи оси поворота пластины. Это связано с ограничением влияния интерполирующей функции в области оси вращения. Максимальная деформация достигнута методом радиальных базисных функций с $\phi(\|\mathbf{x}\|) = (1 - \|\mathbf{x}\|/r)^2$. Результаты показали, что деформирование сетки практически не ухудшило мер качества сетки, несмотря на большой угол (60°) поворота заслонки. Дальнейшее увеличение угла поворота порождает вырожденные элементы, также вблизи оси поворота заслонки.

Также проведена оценка времени выполнения методов на гибридном вычислительном кластере. Для метода Шепарда вычислительные затраты на CPU составили примерно 13 секунд. Деформирование на основе метода радиальных базисных функций осуществлялось различными функциями. Расчеты показали, что время решения зависит от выбора базисной функции. Например, для функции $\phi(\|\mathbf{x}\|) = (1 - \|\mathbf{x}\|/r)^2$, время решения одним потоком центрального процессора составило 42 секунды. Использование GPU для решения системы уравнений позволило сократить время расчета до 3 секунд.

4. Численное моделирование

Работа программного комплекса выглядит следующим образом. Каждое из приложений запускается на отдельном гибридном вычислительном узле, содержащем CPU/GPU процессоры. Сначала запускаются серверные объекты принадлежащие OpenFOAM и FESstudio, а в управляющем процессе создается клиентский объект FESstudioFSI, который содержит прокси к соответствующим серверам. Во время запуска серверных объектов инициализируются переменные, необходимые для расчетов каждой из программ OpenFOAM и FESstudio, а также загружаются соответствующие расчетные сетки. Клиент инициализирует необходимые переменные и ждет сигнал о готовности со стороны серверов. Как только сервера сигнализируют о готовности, клиент запускает расчет и обмен данными, выполняя соответствующие запросы серверам. Кроме обеспечения взаимодействия двух серверов клиент, в рамках построенной модели, занимается интерполяцией данных между сетками и деформированием сетки. Время продолжительности расчета определяется клиентом и не зависит от серверов.

Первая тестовая задача описывает взаимодействие нестационарного сверхзвукового потока воздуха и деформируемой консоли в ударной трубе. Подробное описание задачи, а также результаты экспериментальных и численных исследований можно найти в [6], [7]. Физические особенности моделируемого явления допускают применение последовательного алгоритма слабого связывания для отыскания решения. Полученное численное решение этой задачи хорошо согласуется с данными экспериментальных исследований [6]. Сравнение выполнялось как для газодинамической картины падающих и отраженных ударных волн, так и для смещений пластины (результаты и их обсуждение приведены в [7]).

Для тестирования алгоритма сильного связывания была рассмотрена в трехмерной постановке задача обтекания потоком несжимаемой вязкой жидкости гибкой пластины, закрепленной на неподвижном основании [8]. Гидродинамическая картина течения определяется движением вихрей, образующихся при обтекании жесткого бруса. Вихри зарожда-

ются попеременно на противоположных гранях основания, что приводит к возникновению колебаний давления и вынуждающей силы, отклоняющей пластину. Для удовлетворения условия сопряжения на границе раздела требовалось от 5 до 7 итераций внутреннего цикла связывания. Сравнение полученных результатов с данными работы [8] показало хорошее совпадение амплитуд и периодов колебаний трех контрольных точек, расположенных на нижней по потоку грани пластины.

5. Выводы

Предлагаемый подход к моделированию задач взаимодействия потока газа и деформируемой конструкции/тела основан на выделении этапа согласования и взаимодействий в отдельную программную подсистему, не зависящую от используемых реализаций расчетов газодинамических полей и напряженно-деформированного состояния. Создан прикладной программный интерфейс FEStudioFSI с выделением самостоятельного (параллельного) приложения, непосредственно решающего всю совокупность задач связывания (синхронизация, обмен данными, интерполяция, деформирование расчетных сеток). Таким образом, можно считать, что мультидисциплинарный программный комплекс построен по архитектуре клиент-сервер, где каждое приложение-клиент оптимальным образом распараллелено, решает свою узкоспециальную прикладную задачу и обменивается расчетными данными через сервер с другими клиентами. Предлагаемый подход обеспечивает наибольшую универсальность и позволяет объединять независимые приложения от которых требуется лишь использование соответствующего прикладного программного интерфейса.

Литература

1. Weller H.G., Tabor G., Jasak H., Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques // *Computers in Physics*. 1998. V. 12, N. 6. P. 620–631.
2. Копысов С.П., Кузьмин И.М., Недожогин Н.С., Новиков А.К., Рычков В.Н., Сагдеева Ю.А., Тонков Л.Е. Параллельная реализация конечно-элементных алгоритмов на графических ускорителях в программном комплексе FEStudio // *Компьютерные исследования и моделирование*. 2014. № 1, Т. 6. С. 79–97.
3. Henning M. A new approach to object-oriented middleware // *Internet Computing, IEEE*. 2004. N. 1, V. 8. P. 66–75.
4. Kopysov S.P., Kuzmin I.M., Nedozhogin N., Novikov A.K., Sagdeeva Y.A. Hybrid multi-GPU solver based on Schur complement method // *Lecture Notes in Computer Science*. 2013. V. 7979. P. 65–79.
5. Копысов С.П., Кузьмин И.М., Тонков Л.Е. Методы деформирования сеток в сопряженных задачах // *Вычислительные методы и программирование*. 2013. № 3, Т. 14. С. 269–278.
6. Giordano J., Jourdan G., Burtschell Y., Medale M., Zeitoun D. E., Houas L. Shock wave impacts on deforming panel, an application of fluid-structure interaction // *Shock Waves*. 2005. V. 14. P. 103–110.
7. Копысов С.П., Кузьмин И.М., Тонков Л.Е. Моделирование взаимодействия сверхзвукового потока и деформируемой панели в ударной трубе // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*. 2012. № 2. С. 156–165.
8. Vaassen J.-M., De Vincenzo P., Hirsch C., Leonard B. Strong coupling algorithm to solve fluid-structure interaction problems with a staggered approach // *11th Int. WS on Simulation and EGSE facilities for Space Programmes*. 2010. V. 692. P. 117–123.