

**О. А. Бессонов**, ст. науч. сотр., bess@ipmnet.ru, Институт проблем механики им. А. Ю. Ишлинского Российской академии наук, Москва,  
**Н. А. Харченко**, вед. специалист, Корпорация "Тактическое ракетное вооружение", Московская обл., Королёв

## Программная платформа для суперкомпьютерного моделирования задач аэротермодинамики

*Представлена программная платформа для суперкомпьютерного моделирования, основанная на новом технологическом подходе, существенно упрощающем распараллеливание задач по работе с неструктурированными сетками. Сформулирован принцип единого математического адресного пространства задачи для всех используемых узлов кластера. Распараллеливание кода в рамках нового подхода проводится с минимальными усилиями, без изменения основных частей программы. Представлены результаты замеров производительности, подтверждающие хорошую масштабируемость метода.*

**Ключевые слова:** вычислительная аэротермодинамика, уравнение Навье—Стокса, неструктурированные сетки, алгоритм Катхилла—Макки, разбиение сетки, распараллеливание, OpenMP, MPI

### Введение

Численное моделирование аэротермодинамических характеристик высокоскоростных летательных аппаратов требует значительных вычислительных ресурсов. Основная причина этого — необходимость в большом числе сеточных элементов (десятки или сотни миллионов ячеек) для разрешения тонких областей течения, пограничных слоев и ударных волн на гиперзвуковых режимах обтекания. Моделирование сложных мультифизических, химически неравновесных и радиационных процессов усложняет структуру числового кода и дополнительно увеличивает требования к вычислительным ресурсам [1, 2].

С распространением высокопроизводительных кластеров и суперкомпьютеров предоставляется возможность значительно ускорить процесс моделирования за счет массового распараллеливания вычислительных кодов [3, 4]. К настоящему времени накоплен большой опыт распараллеливания в моделях с общей и распределенной памятью, а также в гибридной модели OpenMP + MPI. Такая модель является наиболее перспективной для кластеров, построенных из многоядерных процессорных узлов [5]. Отработаны основные принципы распараллеливания с использованием этих моделей.

Однако распараллеливание по-прежнему остается очень сложной задачей и требует больших усилий и опыта, а также обучения программистов и координации действий между ними при создании новых кодов или адаптации старых. В связи с этим обстоятельством возрастает потребность в технологическом подходе, который мог бы упростить разработку и ускорить процесс распараллеливания существующих вычислительных кодов и создания

новых. В рамках этого подхода желательно, чтобы изменения в наследуемом последовательном коде были минимальными, а схема разбиения сетки не влияла на структуру кода. В то же время ради достижения отмеченных целей не следует сильно жертвовать эффективностью распараллеленного кода.

В предыдущей статье авторов по этой тематике [6] рассматривался подход к распараллеливанию в гибридной модели OpenMP + MPI на основе единого адресного пространства для всех подобластей параллельной программы, находящихся в разных узлах кластера. В статье был представлен пример такого распараллеливания для задачи моделирования несжимаемой вязкой жидкости в области регулярной геометрии.

Настоящая статья расширяет положения этого подхода и формулирует принцип единого математического адресного пространства в применении к задачам с неструктурированными сетками. Этот принцип удовлетворяет указанным выше требованиям как в части минимизации изменений, которые вносятся в код программы, так и простоты разбиения используемой для моделирования расчетной сетки. Данный подход реализован для задачи моделирования характеристик летательных аппаратов, однако он может быть применен и к другим задачам, использующим неструктурированные сетки с одномерной нумерацией ячеек. При этом физико-математическая специфика задачи не имеет принципиального значения.

Формализация представленного подхода и реализация его программного базиса в виде структур данных, подпрограмм обмена и правил работы с ними образуют программную платформу, на основе которой строятся параллельные расчетные программы.

## 1. Математическая модель и численный метод

Представленная в данной статье программная платформа использовалась для распараллеливания компьютерного кода ГРАТ [7], предназначенного для проведения расчетов аэротермодинамических характеристик высокоскоростных летательных аппаратов с применением в широком диапазоне чисел Маха, высот и углов атаки. В основе кода ГРАТ лежит решение трехмерной нестационарной системы уравнений движения вязкого, теплопроводного, химически неравновесного реагирующего газа. Система уравнений Навье—Стокса, выражающая законы сохранения массы, импульсов и полной энергии, записывается в следующем виде:

$$\begin{aligned} \frac{\partial \mathbf{w}}{\partial t} + \frac{\partial F^x(\mathbf{w})}{\partial x} + \frac{\partial F^y(\mathbf{w})}{\partial y} + \frac{\partial F^z(\mathbf{w})}{\partial z} = \\ = \frac{\partial G^x(\mathbf{w})}{\partial x} + \frac{\partial G^y(\mathbf{w})}{\partial y} + \frac{\partial G^z(\mathbf{w})}{\partial z}, \end{aligned}$$

где  $\mathbf{w} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$  — вектор консервативных переменных;  $\rho$  — плотность;  $u, v, w$  — компоненты скорости;  $E$  — полная энергия;  $F$  и  $G$  — векторные операторы конвективного и вязкого потоков.

В задачах высокоскоростного моделирования [8—11], не требующих учета физико-химических процессов в газе, решаемая система уравнений замыкается уравнениями состояния совершенного газа. В задачах гиперзвукового моделирования [12], когда рассматривается смесь химически реагирующих газов с конечными скоростями реакций, система уравнений замыкается термическим и калорическим уравнениями состояния химических компонентов газовой смеси.

Численное интегрирование конвективной части системы уравнений газовой динамики проводится с использованием метода HLLC [13], который относится к приближенным методам решения задачи о распаде произвольного разрыва, развивающим идею С. К. Годунова о вычислении потоков через грани конечного объема [14]. Для повышения порядка пространственной аппроксимации решения [15] проводится линейная реконструкция значений газодинамических параметров, используемых для вычисления конвективных потоков через грани конечного объема. Для решения системы уравнений химической кинетики используется двухшаговый метод расщепления по физическим процессам, где образование химических компонентов газовой смеси учитывается неявным образом. Вследствие того, что решение стационарной задачи проводится методом установления, используется явный метод интегрирования по времени.

Дискретизация расчетной области методом конечного объема проводится на неструктурированных сетках, важным преимуществом которых является автоматизация построения для сложных геометрических форм (рис. 1, а). Возможность построения квазирегулярной призматической сетки позволяет разрешить тонкие области течения, образующиеся в ударном слое и у поверхности обтекаемого тела на гиперзвуковых режимах (рис. 1, б). Дискретизация граничных условий основана на использовании фиктивных ячеек, прилегающих к каждой границе расчетной области.

## 2. Общая организация вычислительного кода ГРАТ

Вычислительный код ГРАТ написан на языке Фортран. Организация и структура вычислительного кода определяется использованием неструктурированной сетки для дискретизации вычислительной области.

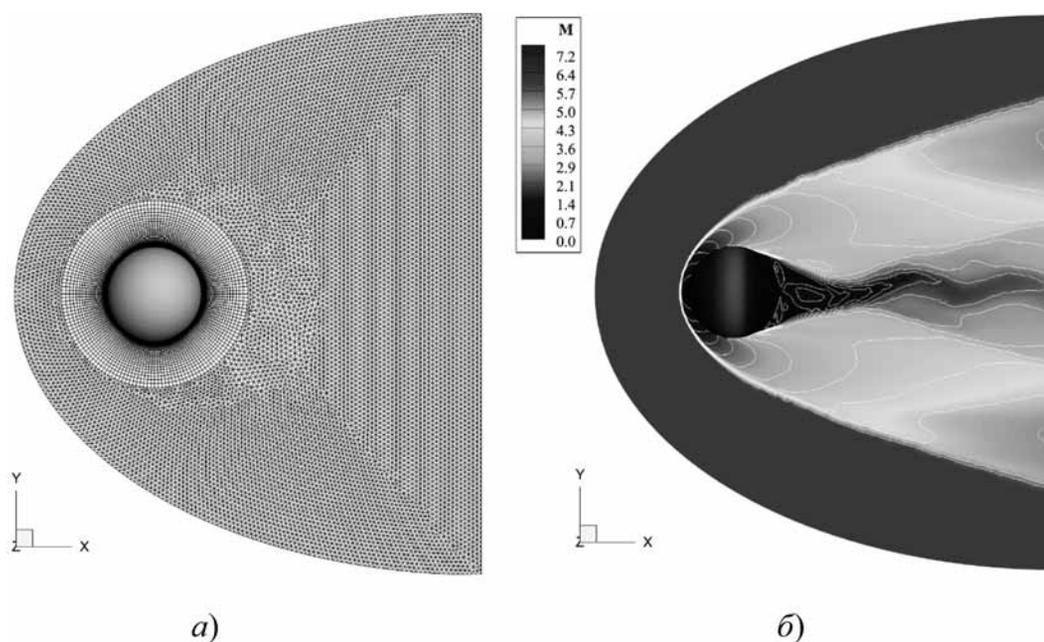


Рис. 1. Конечно-объемная сетка (а) и распределение числа Маха при гиперзвуковом обтекании затупленного тела (б)

Вычисления выполняются в соответствии с порядком нумерации ячеек (контрольных объемов). Основные структуры данных, предназначенные для хранения значений физических величин и вспомогательных переменных, имеют размерность  $(N)$  или  $(M, N)$ , где  $N$  — полное число ячеек, включая фиктивные ячейки за границей расчетной области ( $N_{tot}$ ), либо число внутренних ячеек в области ( $N_{sum}$ ), а  $M$  — число граней у ячейки (4 либо 5). Массивы размерности  $(N)$  используются для хранения значений величин, размещенных внутри ячейки, а размерности  $(M, N)$  — для значений, находящихся на гранях ячейки (например, для численных потоков), либо принадлежащие соседним ячейкам. Все фиктивные элементы в массивах размерности  $(N_{tot})$  расположены в конце массивов, после внутренних элементов. Это позволяет проводить вычисления в режиме сквозного счета в DO-циклах от 1 до  $N_{sum}$ .

Для указания связей ячеек друг с другом используется массив  $NB(M, N)$ , в котором для каждой внутренней ячейки области хранятся номера соседних ячеек (внутренних либо фиктивных). В математическом представлении этот массив является разреженной матрицей, в каждой строке которой содержится 4 или 5 ненулевых элементов. Позиции этих элементов в строках матрицы соответствуют номерам соседних ячеек. При проведении вычислений, затрагивающих соседние ячейки, происходят обращения к элементам различных массивов по этим номерам (индексам). Алгебраически обработку таких массивов можно представить в виде  $\mathbf{y} = \mathbf{A} \otimes \mathbf{x}$ , где  $\mathbf{A}$  — матричное представление массива  $NB$ ;  $\mathbf{x}$  — массив, индексируемый по этой матрице;  $\mathbf{y}$  — результат вычислений и  $\otimes$  — оператор, проводящий вычисления над индексируемыми элементами массива  $\mathbf{x}$ . Такое представление аналогично записи оператора умножения разреженной матрицы на вектор.

В зависимости от способа нумерации ячеек в расчетной области ненулевые элементы в матрице  $\mathbf{A}$  могут быть размещены более или менее регулярным образом. Если такие элементы расположены максимально близко к диагонали матрицы, то индек-

сируемые элементы массивов будут сгруппированы в небольшие подмножества, помещающиеся в кеш-память. Последующие выборки элементов такого подмножества происходят из кеш-памяти, что приводит к снижению времени счета.

На практике все используемые генераторы сеток нумеруют узлы и ячейки неоптимальным образом. В связи с этим обстоятельством проводится предварительная перенумерация ячеек с использованием упрощенного варианта метода Катхилла—МакКи [16]. Благодаря такой перенумерации время счета может уменьшиться на 15...25 % (в зависимости от качества исходной сетки). Другая цель перенумерации — обеспечить эффективное разбиение сетки для кластерного распараллеливания задачи.

На рис. 2 показаны примеры портрета матрицы  $\mathbf{A}$  для исходной и перенумерованной сеток (рис. 2, *a* и *b*, соответственно). Также показана средняя часть перенумерованной матрицы в 25-кратном увеличении (рис. 2, *в*).

Можно заметить, что матрица  $\mathbf{A}$  после перенумерации имеет ленточную структуру. При этом ширина ленты определяет разброс адресов, по которым происходят обращения к памяти. При оптимальной перенумерации максимальная ширина ленты для трехмерной области приблизительно равна  $O(N^{2/3})$ .

Качество перенумерованной матрицы может зависеть от выбора ячейки, с которой начинается новая нумерация. В программном модуле применен адаптивный алгоритм с проверкой трех вариантов начальной ячейки и выбором варианта с минимальной шириной ленты.

### 3. Распараллеливание в модели общей памяти OpenMP

Распараллеливание в модели OpenMP проводится стандартным образом, с использованием операторов "\$OMP DO" языка Фортран. В результате применения этого оператора DO-циклы размерности  $N$  автоматически разбиваются на приблизительно равные части, которые вычисляются в параллельно исполняемых

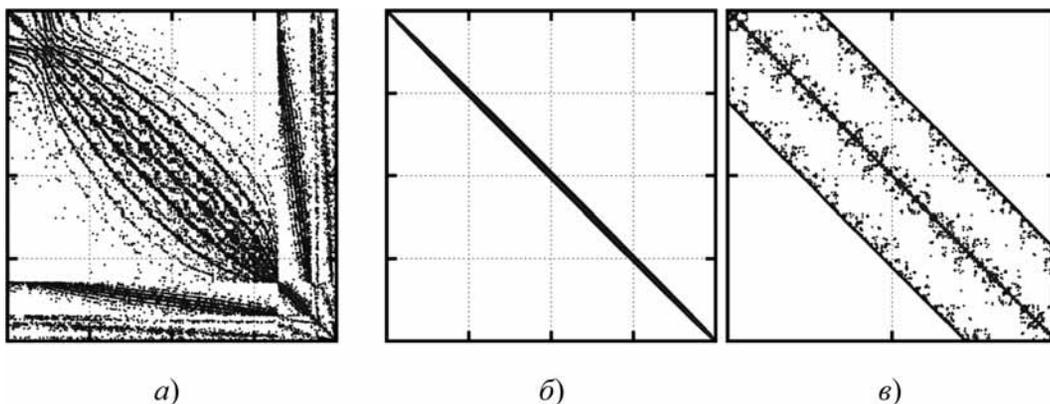


Рис. 2. Иллюстрация оптимизации сетки на примере портрета соответствующей матрицы: *a* — оригинальная; *b* — перенумерованная; *в* — средняя часть перенумерованной матрицы, увеличенная в 25 раз

потоках. При таком разбиении большинство обращений к индексированным элементам массивов благодаря оптимизации сетки происходит в пределах небольшого подмножества, размещенного в кеш-памяти.

Распараллеливание для систем с общей памятью, содержащих два и более процессоров, требует специальной оптимизации в связи с тем, что такие системы имеют неоднородную организацию оперативной памяти (*Non-uniform memory organization*, NUMA). Процессоры в системах типа NUMA объединены высокоскоростными каналами, и к каждому процессору подсоединен свой блок памяти. При выполнении программы обращения к данным, размещенным в памяти своего процессора, происходят с полной скоростью, а к данным в другом процессоре — с существенным замедлением. По этой причине требуется специальная инициализация основных массивов в параллельных циклах, идентичных циклам, в которых проводятся вычисления. В результате физические страницы памяти выделяются в собственной памяти соответствующего процессора.

Следует отметить, что для определенных конфигураций расчетов может оказаться предпочтительным запускать отдельный процесс MPI в каждом процессоре многопроцессорного узла вместо запуска одного процесса на узел.

Запуск расчета в системе с общей памятью должен проводиться с привязкой потоков к вычислительным ядрам. Это особенно важно для систем типа NUMA (например, двухпроцессорных узлов вычислительных кластеров), так как миграция потоков может привести к тому, что потоки и их данные окажутся в разных процессорах и скорость счета заметно снизится. В операционной системе Linux запуск расчетов с привязкой потоков проводится командой "taskset" либо "numactl", а в ОС Windows — командой "start".

#### 4. Технология кластерного распараллеливания в модели единого математического адресного пространства

Распараллеливание задачи для вычислительных систем с распределенной памятью (кластеров) требует разбиения расчетной области на подобласти и организации расчета таким образом, чтобы вычис-

ления внутри отдельных подобластей перемежались с обменами приграничными данными.

Обмены данными между узлами кластера проводятся с помощью вызовов подпрограмм библиотеки MPI. Как правило, для такого распараллеливания создается специальный вычислительный код с организацией независимых структур данных в каждом кластерном узле и с явным вызовом обменов с соседними узлами в соответствии со структурой разбиения расчетной области [17]. При этом разбиение расчетной области означает разделение каждого исходного массива размерности ( $N$ ) на подмассивы с независимой нумерацией элементов, а также подготовку специальных структур данных, описывающих границы между подобластями и стратегии обмена данными между узлами кластера.

##### 4.1. Оптимизация и разбиение расчетной сетки

Для распараллеливания кода в представляемом программном комплексе используется технологический прием, при котором сохраняется единая сквозная нумерация ячеек и соответствующих им массивов [6]. Разбиение на подобласти проводится алгебраически путем разделения каждого исходного массива размерности ( $N$ ) на подмассивы, находящиеся в разных узлах кластера. При этом используется общее пространство индексов подмассивов (единое математическое адресное пространство задачи).

Алгебраическое разбиение массивов данных означает, что в основных вычислениях, которые имеют вид  $y = A \otimes x$ , на границах между подмассивами будет наблюдаться перекрытие. На рис. 3 показаны шаблоны доступа к данным при вычислении  $A \otimes x \rightarrow y$  внутри подмассива (рис. 3, а) и вблизи границы между подмассивами (рис. 3, б).

Можно видеть, что для каждого массива, к которому осуществляется доступ как к вектору  $x$ , при выделении памяти для подмассивов должно быть дополнительно добавлено место для области перекрытия. Размер каждой области перекрытия примерно равен полуширине ленты матрицы  $A$  в этой позиции. После каждого вычисления вида  $y = A \otimes x$  должен быть проведен обмен перекрывающимися областями некоторых массивов между соответствующими узлами кластера. Для экономии времени счета обмен

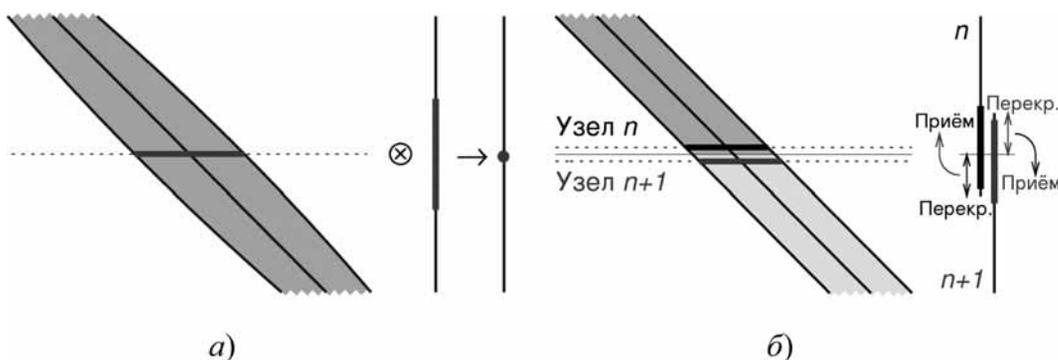


Рис. 3. Иллюстрация характера доступа к данным:

а — внутри подмассива, без перекрытия; б — между подмассивами, находящимися в разных узлах кластера, с перекрытием адресов и обменом данными

данными в противоположных направлениях должен проводиться одновременно (в дуплексном режиме).

В текущей реализации компьютерного кода используются синхронные режимы обмена приграничными данными, что при разбиении на относительно небольшое число подобластей не приводит к существенным накладным расходам. Использование асинхронных режимов обмена данными, происходящих одновременно с вычислениями, потребует более сложных способов обхода массивов и будет реализовано позднее, по мере получения доступа к кластерам с большим числом узлов.

Алгебраический метод разбиения, будучи простым и понятным, не является самым оптимальным. Существует ряд сложных методов декомпозиции доменов, которые минимизируют размер граничных областей между подобластями [18–20]. На рис. 4 приведены примеры почти оптимального разбиения с использованием таких методов.

Однако сложные методы декомпозиции, близкие к оптимальным, имеют ряд недостатков. Как правило, подобласти после такого разбиения имеют большое (и различное) число соседних подобластей с неравными размерами границ. В результате схемы обмена становятся сложными и запутанными, что

усложняет алгоритм распараллеливания и затрудняет его оптимальную программную реализацию. По этой причине применение таких методов оправдано только при разбиении на большое число подобластей.

Вместе с тем при использовании алгебраического метода разбиения каждая подобласть имеет только двух соседей, и размеры границ между подобластями не сильно отличаются. Эти свойства частично компенсируют неоптимальный характер данного метода.

При использовании метода Катхилла—Макки ячейки нумеруются слой за слоем, и характер распространения фронтального слоя по расчетной области зависит от вариаций плотности сетки. При наличии сильных сжатий сетки фронтальный слой может в этих местах искривляться и общее число ячеек в слое может увеличиваться. В местах разбиения расчетной области это значение определяет число граничных ячеек между подобластями и, соответственно, полуширину ленты матрицы  $A$ , которая соответствует объему обменов данными между узлами кластера.

На рис. 5 показаны примеры разбиения трехмерных расчетных областей на восемь подобластей с использованием этого метода. На этих примерах можно видеть искривление границ между подобластями. Число ячеек сетки ( $N$ ) и полуширина ленты матрицы ( $H$ ) для этих примеров следующие:  $N = 12$  млн,  $H = 78\,000$  (рис. 5, а);  $N = 6$  млн,  $H = 28\,000$  (рис. 5, б). Значения параметра  $H/N^{2/3}$ , указывающие относительное (нормализованное к размеру сетки) значение полуширины матрицы, составляют 1,48 и 0,84 соответственно. Это означает, что оба разбиения сетки имеют приемлемое качество, при этом качество второго разбиения немного лучше, чем первого.

#### 4.2. Структура вычислительного кода

Технически расчеты в подобластях организованы следующим образом. Для всех подмассивов данных, совместно использующих единое индексное пространство задачи, определены соответствующие целочисленные переменные, обозначающие диапазоны размещения внутренних (реальных) ячеек,

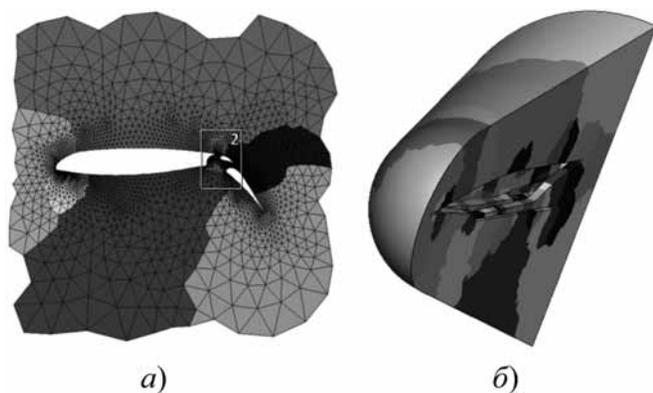


Рис. 4. Иллюстрация почти оптимального разбиения области на 8 подобластей в двумерном случае [20] (а) и на 64 подобласти в трехмерном [17] (б)

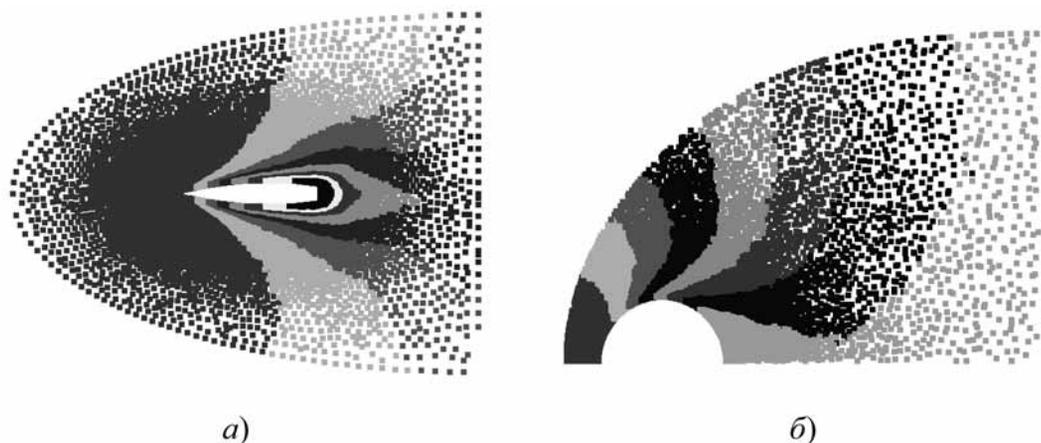


Рис. 5. Примеры применения алгебраического метода разбиения сетки

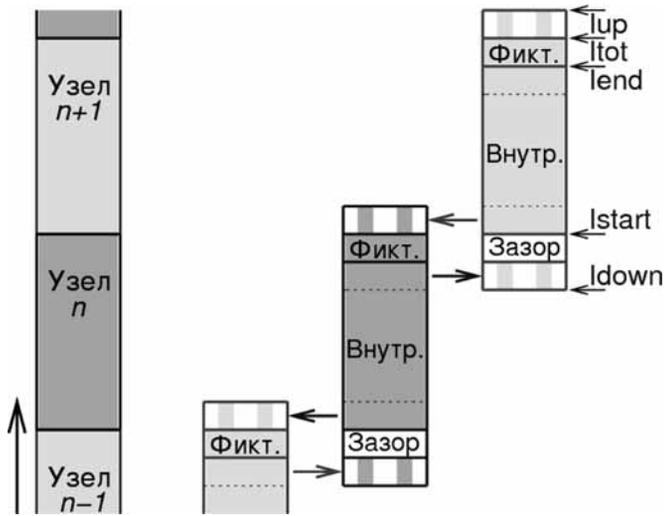


Рис. 6. Структура подмассива в едином пространстве индексов задачи и иллюстрация обменов между перекрывающимися областями

фиктивных ячеек и областей перекрытия. На рис. 6 показана структура подмассива и характер данных, которыми обмениваются подобласти.

Основными диапазонами индексных переменных являются  $Istart$ :  $Iend$  для обработки DO-циклов и  $Idown$ :  $Iup$  для выделения памяти для подмассивов. Индексные пространства подмассивов в соседних узлах кластера перекрываются, при этом соотношение между локальными индексными переменными выглядит следующим образом: значение  $I_{tot} + 1$  для узла  $n$  должно быть равно значению  $Istart$  для узла  $n + 1$ .

Вычисления в узлах кластера выполняются с распараллеливанием в модели OpenMP. После завершения каждого этапа вычислений проводится обмен с обоими соседними узлами. Используется процедура `MPI_SENDRECV`, которая одновременно принимает данные от соседнего узла и передает данные этому узлу.

Технологически все процедуры обмена данными отделены от вычислительного кода и размещаются в специальном файле `XCH.F`. Все основные обмены

между соседними подобластями реализованы в подпрограмме `XCH`, которая вызывается один раз после каждого этапа вычислений (для каждого вновь вычисленного массива, имеющего перекрывающиеся области). Эта подпрограмма самостоятельно определяет, исходя из номера текущего узла и их общего числа, с какими подобластями и в каком порядке проводить операции встречных обменов. В случае, если задача запущена на единственном узле, такая подпрограмма не будет выполнять никакой работы.

На рис. 7, а, показано, как работает подпрограмма `XCH`. Подпрограмма делает два вызова `MPI_SENDRECV` в соответствующем порядке для обмена с обоими соседними узлами кластера.

Аналогичным образом реализованы подпрограммы редукции для вычисления суммы значений переменных во всех узлах, а также их максимального и минимального значений (`XSUM`, `XMAX`, `XMIN`). Для этих вычислений используется алгоритм типа "бабочка" (подобно применяемому в быстром преобразовании Фурье) (рис. 7, б).

Также в коде реализованы унифицированные подпрограммы для распределения данных между узлами кластера после ввода и для сбора данных из узлов перед выводом.

Выделение всех вызовов подпрограмм библиотеки `MPI` в отдельный файл позволяет компилировать основной код обычным компилятором. Компилятор, поддерживающий `MPI`, необходим только для окончательной сборки с использованием этого файла. Также программа может быть собрана без библиотеки `MPI` — для этого случая реализован набор подпрограмм, имитирующих выполнение на одном узле без обменов (`XCH_NOMPI.F`).

Рисунок 8 иллюстрирует эволюцию компьютерного кода в виде этапов — чисто последовательного кода, параллельного кода в модели `OpenMP` и параллельного кода в кластерной модели. На рис. 8 выделены принципиальные изменения основного `DO`-цикла на каждом этапе развития: добавление операторов `!$OMP` для распараллеливания в `OpenMP`, вызов универсальной подпрограммы обмена `XCH` и коррекция диапазона `DO`-цикла для кластерного распараллеливания.

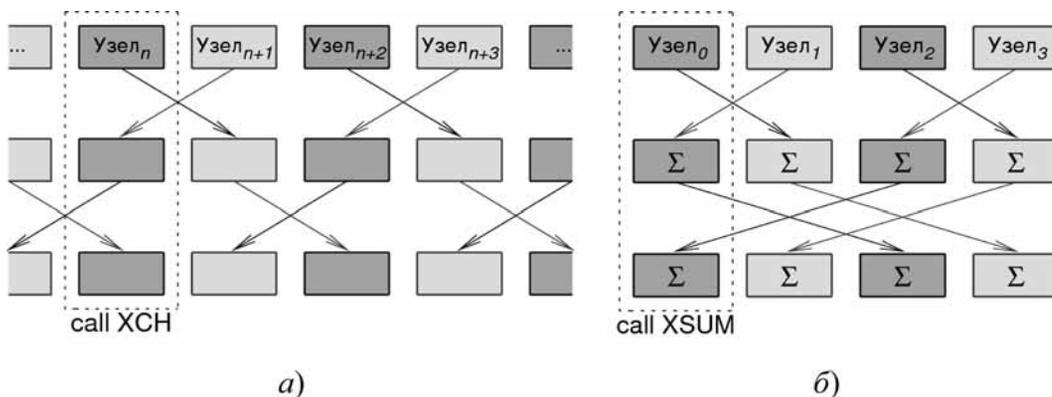


Рис. 7. Иллюстрация основных типов обменов между узлами кластера:

а — обмен с соседними узлами по завершении этапа вычислений; б — алгоритм обменов логарифмической сложности для операций редукции

```

do I=1,Nsum
...
A(I)= ...
B(K,I)= ...
...
enddo

```

a)

```

!$OMP DO
do I=1,Nsum
...
A(I)= ...
B(K,I)= ...
...
enddo
!$OMP END DO

```

б)

```

!$OMP DO
do I=Istart,Iend
...
A(I)= ...
B(K,I)= ...
...
enddo
!$OMP END DO
!$OMP SINGLE
call XCH(...,A)
call XCH(...,B)
!$OMP END SINGLE

```

в)

**Рис. 8. Эволюция вычислительного кода:**  
a — последовательный код; б — OpenMP; в — OpenMP + MPI

Таким образом, технология кластерного распараллеливания с единым математическим адресным пространством задачи позволяет реализовать унифицированный расчетный код для всех режимов — однопоточного, многопоточного в модели общей памяти OpenMP, и кластерного в гибридной модели OpenMP + MPI. При этом распараллеливание кода для кластерного режима обеспечивается минимальными усилиями, без переделки основных частей программы, и при единственном условии — наличии естественного потенциала распараллеливания (например, при использовании явного метода интегрирования по времени). Метод распараллеливания обеспечивает достаточно хорошее качество разбиения расчетной области, при котором число граничных ячеек между подобластями не превышает уровня  $O(N^{2/3})$ .

### 5. Измерения производительности

Новый алгоритм распараллеливания был протестирован на кластере, построенном из двухпроцессорных узлов на процессорах Intel Xeon, содержащих 16 ядер с тактовой частотой 2,6...2,9 ГГц. Узлы кластера соединены между собой каналами Omni-Path со скоростью обмена 100 Гбит/с. Результаты распараллеливания при решении задачи аэротермодинамики с сеткой, содержащей 33 млн ячеек, показаны на рис. 9.

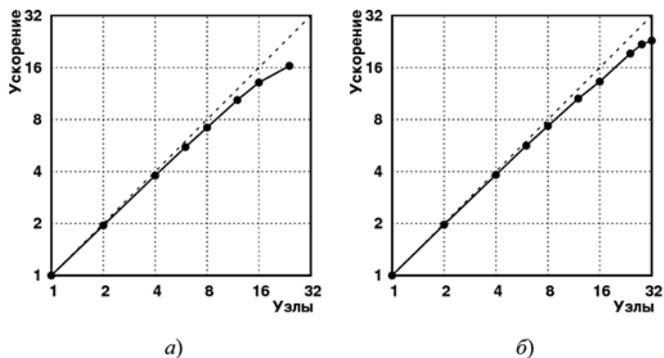
На рис. 9, а, показано ускорение при включенном режиме Hyper-threading и запуске одного процесса MPI на узел в режиме NUMA. В этом сочетании режимов эффект насыщения параллельной производительности начинает наблюдаться на 24 узлах (768 процессорных ядер, 1536 потоков). График на рис. 9, б, соответствует отключенному режиму Hyper-threading и запуску отдельного процесса MPI в каждом процессоре. Здесь насыщение достигается на 32 узлах (1024 ядра). Данное сравнение показывает, что при разбиении на большое число подобластей начинают проявляться эффекты вследствие замедленного доступа к нелокальной памяти в режиме NUMA.

В таблице показаны ускорение и эффективность распараллеливания задачи для этих режимов (режим 1 — один процесс на узел, Hyper-threading включен; режим 2 — два процесса на узел, Hyper-threading отключен).

Приведенные выше результаты демонстрируют хорошую масштабируемость метода распараллеливания

**Ускорение и эффективность распараллеливания задачи**

Число узлов	Режим 1		Режим 2	
	Ускорение	Эффективность, %	Ускорение	Эффективность, %
1	1	100	1	100
2	1,95	97,4	1,97	98,6
4	3,79	94,7	3,83	95,6
6	5,54	92,3	5,64	94,1
8	7,18	89,8	7,34	91,8
12	10,4	86,4	10,6	88,2
16	13,1	83,1	13,3	82,9
24	16,4	68,2	19,3	80,3
28	—	—	21,9	77,8
32	—	—	22,9	71,5



**Рис. 9. Ускорение при распараллеливании задачи с сеткой, содержащей 33 млн ячеек, в различных сочетаниях режимов Hyper-threading и MPI**

ния. Основываясь на 23-кратном ускорении на одном узле кластера по сравнению с одним потоком, общее ускорение на 32 узлах составляет 520. Таким образом, эти результаты иллюстрируют широкий спектр вычислительных платформ, на которых метод может эффективно использоваться.

## Заключение

Представлен новый технологический подход к реализации эффективного кластерного распараллеливания газодинамических и аэродинамических задач, использующих неструктурированные сетки, и программная платформа, построенная на его основе. Этот подход основан на едином математическом адресном пространстве данных задачи для всех используемых узлов кластера задачи. Благодаря рассматриваемому подходу распараллеливание кода осуществляется с минимальными усилиями, без изменения основных частей программы. Кроме того, такой подход значительно упрощает разбиение расчетной сетки.

В статье также представлены приемы, упрощающие организацию обменов между узлами кластера и облегчающие программирование. В частности, все процедуры обмена между узлами, содержащие вызовы подпрограмм библиотеки MPI, отделены от основного кода.

В результате применения описанных выше подходов создается единый вычислительный код для всех режимов счета — последовательного, многопоточного (в модели общей памяти OpenMP) и кластерного (в модели распределенной памяти OpenMP + MPI).

Представленный подход может быть применен к различным задачам, использующим неструктурированные сетки с одномерной нумерацией ячеек и допускающим распараллеливание на основе разбиения расчетной области.

Разработанная программная платформа использовалась для распараллеливания компьютерного кода ГРАТ. Результаты измерений производительности кода демонстрируют хорошую масштабируемость предложенного метода распараллеливания.

*Работа выполнена при частичной поддержке из средств госбюджета в рамках государственного задания АААА-А20-120011690131-7.*

## Список литературы

1. Shang J. S., Surzhikov S. T. Simulating nonequilibrium flow for ablative Earth reentry // *J. Spacecraft Rockets*. — 2010. — Vol. 47. — P. 806–815.
2. Zheleznyakova A. L., Surzhikov S. T. Application of the method of splitting by physical processes for the computation of a hypersonic flow over an aircraft model of complex configuration // *High Temp*. — 2013. — Vol. 51. — P. 816–829.
3. Afzal A., Ansari Z., Faizabadi A. R., Ramis M. K. Parallelization strategies for computational fluid dynamics software:

state of the art review // *Arch. Computat. Methods Eng.* — 2017. — Vol. 24. — P. 337–363.

4. Probst A., Knopp T., Grabe C., Jägersküpper J. HPC requirements of high-fidelity flow simulations for aerodynamic applications // *In Proc. Euro-Par 2019, Lecture Notes in Computer Science*. — 2019. — Vol. 11997. — P. 375–387.

5. Rabenseifner R., Hager G., Jost G. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes // *In 17th Euromicro international conference on parallel, distributed and network-based processing*. — 2009. — P. 427–436.

6. Bessonov O. Technological aspects of the hybrid parallelization with OpenMP and MPI // *In Proc. PaCT 2017, Lecture Notes in Computer Science*. — 2017. — Vol. 10421. — P. 101–113.

7. Харченко Н. А., Рыбаков А. Н. Аэротермодинамика спускаемого космического аппарата "Аполлон-4" // *Точно в цель*. — 2020. — № 1. — С. 77–88.

8. Kharchenko N., Kryukov I. Aerothermodynamics calculation of the EXPERT reentry flight vehicle // *J. Phys.: Conf. Ser.* — 2018. — Vol. 1009. — Article 012004.

9. Kharchenko N., Kotov M. Analysis of the High Speed Gas Flow over a Sphere in the Range of Mach Numbers 2–12 // *J. Phys.: Conf. Ser.* — 2018. — Vol. 1009. — Article 012007.

10. Koryukov I., Kryukov I. Three-dimensional calculation of the aerothermodynamics of a double cone 25°/55° on an unstructured grid // *J. Phys.: Conf. Ser.* — 2018. — Vol. 1009. — Article 012003.

11. Макенч Г. С., Харченко Н. А., Крюков И. А. Расчет аэродинамики и динамики полета спускаемого летательного аппарата EXPERT // *Физико-химическая кинетика в газовой динамике*. — 2017. — Т. 18, Вып. 2. URL: <http://chemphys.edu.ru/issues/2017-18-2/articles/738/>

12. Kharchenko N., Kotov M. Aerothermodynamics of the Apollo-4 spacecraft at earth atmosphere conditions with speed more than 10 km/s // *J. Phys.: Conf. Ser.* — 2019. — Vol. 1250. — Article 012012.

13. Einfeldt B., Munz C. D., Roe P. L., Sjögreen B. On Godunov-type methods near low densities // *J. Comput. Phys.* — 1991. — Vol. 92. — P. 273–295.

14. Годунов С. К. Разностный метод численного расчета разрывных решений уравнений гидродинамики // *Матем. сб.* — 1959. — Т. 47 (89), № 3. — С. 271–306.

15. Крюков И. А., Иванов И. Э., Ларина Е. В. Программный комплекс расчета высокоскоростных течений hySol // *Физико-химическая кинетика в газовой динамике*. — 2021. — Т. 22, Вып. 1. URL: <http://chemphys.edu.ru/issues/2021-22-1/articles/902/>

16. Cuthill E., McKee J. Reducing the bandwidth of sparse symmetric matrices // *In ACM'69: Proc. 24th Nat. Conf.* — 1969. — P. 157–172.

17. Ermakov M., Kryukov I. Supercomputer modeling of flow past hypersonic flight vehicles // *J. Phys.: Conf. Ser.* — 2017. — Vol. 815. — Article 012016.

18. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs // *SIAM J. Sci. Comput.* — 1998. — Vol. 20, No. 1. — P. 359–392.

19. Karypis G., Kumar V. Multilevel  $\kappa$ -way partitioning scheme for irregular graphs // *J. Parallel Distrib. Comput.* — 1998. — Vol. 48. — P. 96–129.

20. Железнякова А. Л. Эффективные методы декомпозиции неструктурированных адаптивных сеток для высокопроизводительных расчетов при решении задач вычислительной аэродинамики // *Физико-химическая кинетика в газовой динамике*. — 2017. — Т. 18, Вып. 1. URL: <http://chemphys.edu.ru/issues/2017-18-1/articles/673/>

# Software Platform for Supercomputer Modeling of Aerothermodynamics Problems

O. A. Bessonov, bess@ipmnet.ru, Ishlinsky Institute for Problems in Mechanics RAS, Moscow, 119526, Russian Federation, N. A. Kharchenko, kharchenko.na@phystech.edu, Tactical Missiles Corporation, Moscow district, Korolev, 141080, Russian Federation

Corresponding author:

Bessonov Oleg A., Senior Research Scientist, Ishlinsky Institute for Problems in Mechanics RAS, Moscow, 119526, Russian Federation  
E-mail: bess@ipmnet.ru

Received on April 15, 2021

Accepted on June 21, 2021

Numerical solution of aerothermodynamics problems requires a lot of processor time, and for this reason, the numerical codes for such simulations must be efficiently parallelized. This paper presents a software platform based on a technological approach that greatly simplifies the parallelization of problems with unstructured grids. The paper formulates the principle of a unified mathematical address space of the problem for all used cluster nodes. The formalization of the presented technological approach and the implementation of its software basis in the form of data structures, exchange routines and work rules form a software platform on the basis of which parallel computational programs can be built. This approach is implemented and tested for the problem of modeling the characteristics of aircrafts, but it can be applied to other problems using unstructured grids with one-dimensional cell numbering. In this case, the physical and mathematical specifics of the problem are not important. Parallelization of the code with the new approach is carried out with minimal effort, without changing the main parts of the program. As a result, a single computational code can be created for all regimes — sequential, multi-threaded, and cluster. The performance results obtained with the new code confirm the good scalability of the parallelization method.

**Keywords:** computational aerothermodynamics, Navier—Stokes equation, unstructured grids, Cuthill—McKee algorithm, unified mathematical address space, grid partitioning, parallelization, OpenMP, MPI, performance evaluation

## Acknowledgements:

This work was partially supported by the Russian State Assignment under Contract No. AAAA-A20-120011690131-7.

For citation:

Bessonov O. A., Kharchenko N. A. Software Platform for Supercomputer Modeling of Aerothermodynamics Problems, *Programmnyaya Ingeneriya*, 2021, vol. 12, no. 6, pp. 302—310.

DOI: 10.17587/prin.12.302-310

## References

1. Shang J. S., Surzhikov S. T. Simulating nonequilibrium flow for ablative Earth reentry, *J. Spacecraft Rockets*, 2010, vol. 47, pp. 806—815.
2. Zheleznyakova A. L., Surzhikov S. T. Application of the method of splitting by physical processes for the computation of a hypersonic flow over an aircraft model of complex configuration, *High Temp.*, 2013, vol. 51, pp. 816—829.
3. Afzal A., Ansari Z., Faizabadi A. R., Ramis M. K. Parallelization strategies for computational fluid dynamics software: state of the art review, *Arch. Computat. Methods Eng.*, 2017, vol. 24, pp. 337—363.
4. Probst A., Knopp T., Grabe C., Jägersküpper J. HPC requirements of high-fidelity flow simulations for aerodynamic applications, *In Proc. Euro-Par 2019, Lecture Notes in Computer Science*, 2019, vol. 11997, pp. 375—387.
5. Rabenseifner R., Hager G., Jost G. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes, *In 17th Euromicro international conference on parallel, distributed and network-based processing*, 2009, pp. 427—436.
6. Bessonov O. Technological aspects of the hybrid parallelization with OpenMP and MPI, *In Proc. PaCT 2017, Lecture Notes in Computer Science*, 2017, vol. 10421, pp. 101—113.
7. Kharchenko N. A., Rybakov A. N. Aerothermodynamics of the Apollo-4 re-entry vehicle, *Tochno v cel'*, 2020, no. 1, pp. 77—88 (in Russian).
8. Kharchenko N., Kryukov I. Aerothermodynamics calculation of the EXPERT reentry flight vehicle, *J. Phys.: Conf. Ser.*, 2018, vol. 1009, 012004.
9. Kharchenko N., Kotov M. Analysis of the High Speed Gas Flow over a Sphere in the Range of Mach Numbers 2—12, *J. Phys.: Conf. Ser.*, 2018, vol. 1009, article 012007.
10. Koryukov I., Kryukov I. Three-dimensional calculation of the aerothermodynamics of a double cone 25°/55° on an unstructured grid, *J. Phys.: Conf. Ser.*, 2018, vol. 1009, article 012003.
11. Makeich G. S., Kharchenko N. A., Kryukov I. A. Aerodynamics and flight dynamics simulation of EXPERT re-entry vehicle, *Fiziko-khimicheskaya kinetika v gazovoy dinamike*, 2017, vol. 18, no. 2, available at: <http://chemphys.edu.ru/issues/2017-18-2/articles/738/> (in Russian).
12. Kharchenko N., Kotov M. Aerothermodynamics of the Apollo-4 spacecraft at earth atmosphere conditions with speed more than 10 km/s, *J. Phys.: Conf. Ser.*, 2019, vol. 1250, article 012012.
13. Einfeldt B., Munz C. D., Roe P. L., Sjögren B. On Godunov-type methods near low densities, *J. Comput. Phys.*, 1991, vol. 92, pp. 273—295.
14. Godunov S. K. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, *Matematicheskij sbornik*, 1959, vol. 47 (89), no. 3, pp. 271—306 (in Russian).
15. Kryukov I. A., Ivanov I. E., Larina E. V. Software package hySol for the Numerical Simulation of High-speed Flows, *Fiziko-khimicheskaya kinetika v gazovoy dinamike*, 2021, vol. 22, no. 1, available at: <http://chemphys.edu.ru/issues/2021-22-1/articles/902/> (in Russian).
16. Cuthill E., McKee J. Reducing the bandwidth of sparse symmetric matrices, *In ACM'69: Proc. 24th Nat. Conf.*, 1969, pp. 157—172.
17. Ermakov M., Kryukov I. Supercomputer modeling of flow past hypersonic flight vehicles, *J. Phys.: Conf. Ser.*, 2017, vol. 815, article 012016.
18. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.*, 1998, vol. 20, no. 1, pp. 359—392.
19. Karypis G., Kumar V. Multilevel  $\kappa$ -way partitioning scheme for irregular graphs, *J. Parallel Distrib. Comput.*, 1998, vol. 48, pp. 96—129.
20. Zheleznyakova A. L. Effective domain decomposition methods for adaptive unstructured grids applied to high performance computing for problems in computational aerodynamics, *Fiziko-khimicheskaya kinetika v gazovoy dinamike*, 2017, vol. 18, no. 1, available at: <http://chemphys.edu.ru/issues/2017-18-1/articles/673/> (in Russian).