

ПРИКЛАДНАЯ МАТЕМАТИКА И МЕХАНИКА

DOI 10.15507/2079-6900.23.202102.193–206

Оригинальная статья

ISSN 2079-6900 (Print)

ISSN 2587-7496 (Online)

УДК 519.6

Метод балансировки загрузки для гетерогенных численных алгоритмов моделирования газодинамических течений

С. А. Суков

Федеральный исследовательский центр Институт прикладной математики им. М. В. Келдыша Российской академии наук (г. Москва, Российская Федерация)

Аннотация. В статье рассматривается проблема балансировки загрузки в параллельных гетерогенных алгоритмах численного моделирования физических процессов на неструктурированных сетках. Описан метод распределения вычислений для кластеров с центральными процессорами и массивно-параллельными ускорителями. Балансировка загрузки включает определение весов вершин и ребер дуального графа, тестирование быстродействия вычислительных устройств и двухуровневую декомпозицию дискретной модели расчетной области на основе метода геометрического параллелизма. Детали предлагаемого подхода рассматриваются на примере параллельного неструктурированного конечно-объемного алгоритма моделирования уравнений Навье-Стокса, разработанного с использованием моделей программирования MPI, OpenMP и CUDA. Приводятся параметры быстродействия и параллельной эффективности вычислений, полученные в процессе расчета задачи сверхзвукового обтекания сферы на смешанной сетке.

Ключевые слова: гетерогенные вычисления, балансировка загрузки, неструктурированные сетки, вычислительная газовая динамика

Для цитирования: Суков С. А. Метод балансировки загрузки для гетерогенных численных алгоритмов моделирования газодинамических течений // Журнал Средневолжского математического общества. 2021. Т. 23, № 2. С. 193–206. DOI: <https://doi.org/10.15507/2079-6900.23.202102.193–206>

1. Введение

Вычислительный эксперимент постепенно становится основным методом предварительной оценки характеристик проектируемых технических объектов и моделирования физических процессов. Качество результатов расчета задач вычислительной газовой динамики (CFD), помимо точности численного алгоритма, зависит от типа и пространственного шага сетки. Для дискретизации двумерных и трехмерных областей со сложной формой границ широко применяются неструктурированные гибридные и смешанные сетки, которые в общем случае состоят из многогранников произвольной формы [1].

Моделирование течений на подробных сетках CFD-алгоритмами повышенного порядка точности требует привлечения ресурсов суперкомпьютерных систем. На текущем этапе развития вычислительной техники производительность кластеров максимизируется за счет перехода к гетерогенной (гибридной) архитектуре узлов. В дополнение



к центральным многоядерным процессорам (CPU) на каждом узле устанавливаются массивно-параллельные ускорители. Это могут быть существенно многоядерные сопроцессоры Intel Xeon Phi или потоковые графические мультипроцессоры (GPU). Аппаратно заложенный в гетерогенной архитектуре многоуровневый параллелизм требует разработки специальных вычислительных алгоритмов, сочетающих модели параллельных вычислений с распределенной (уровень узлов кластера) и общей (уровень CPU-ядер одного узла) памятью, а также параллелизм SIMD (Single Instruction Multiple Data) на уровне сопроцессоров. Описание примеров алгоритмов моделирования задач газовой динамики для гетерогенных систем и их программных реализаций можно найти в работах [2–5].

Эффективность и масштабируемость параллельных алгоритмов для кластерных систем жестко связана с эффективностью применяемого метода балансировки загрузки. Стандартным решением проблемы распределения вычислений является метод геометрического параллелизма (domain decomposition), при котором каждый MPI-процесс обрабатывает ассоциированное с ним подмножество сеточных ячеек. На этапе балансировки топология сетки представляется как нерегулярный граф ее элементов и их связей по вершинам, ребрам или граням. Для разбиения дуального графа используются программные реализации соответствующих методов [6–7]. Корректная постановка задачи декомпозиции требует ввода весов ячеек смешанной сетки, пропорциональных объему вычислений в процессе их обработки. Результат разбиения для вычислений в гетерогенном режиме должен дополнительно учитывать фактическое соотношение производительности программных реализаций CFD-алгоритма для разных устройств.

В настоящей работе на примере параллельного алгоритма моделирования уравнений Навье-Стокса описан эффективный подход к решению проблемы балансировки загрузки для гетерогенных вычислений. Приводятся оценки производительности и параллельной эффективности расчета сверхзвукового обтекания сферы на смешанной сетке с четырьмя типами элементов.

2. Параллельный алгоритм моделирования уравнений Навье-Стокса

Предлагаемый метод статической балансировки загрузки рассматривается применительно к параллельной версии неструктурированного конечно-объемного алгоритма моделирования задач газовой динамики повышенного порядка точности с явным интегрированием по времени. В качестве математической модели течений вязкого сжимаемого теплопроводного газа используется система нестационарных уравнений Навье–Стокса, записанная в виде законов сохранения

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \mathbf{F} = 0. \quad (2.1)$$

Здесь $\mathbf{Q} = (\rho, \rho u, \rho v, \rho w, E)^T$ – вектор консервативных газодинамических переменных; $\mathbf{F} = \mathbf{F}^C + \mathbf{F}^D$ – вектор потока, равный сумме потоков конвективного и диффузионного переноса, соответственно.

Дискретизация системы уравнений (2.1) выполняется на смешанных сетках с элементами четырех типов (Рис. 2.1). Многогранная сеточная ячейка C_i имеет объем $|C_i|$ и центр масс \mathbf{r}_i . Ее поверхность $\partial C_i = \bigcup_{j \in I_i} \partial C_{ij}$ состоит из разделяемых с многогранниками множества I_i плоских граней. Общая для контрольных объемов C_i и C_j

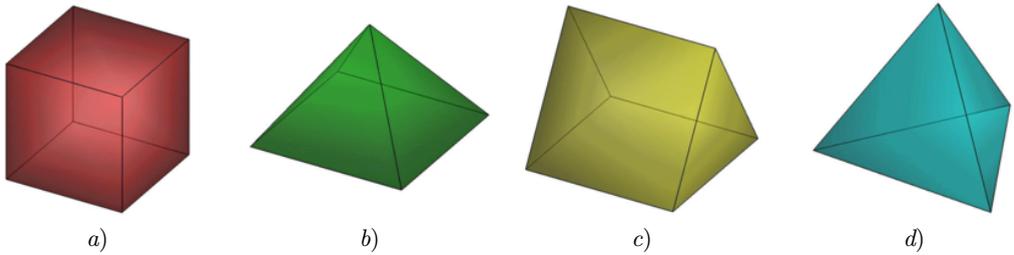


Рис. 2.1. Типы ячеек смешанной сетки: а) гексаэдр; б) четырехугольная пирамида; в) треугольная призма; д) тетраэдр

Fig 2.1. Mixed mesh cell types: а) hexahedron; б) quadrangular pyramid; в) triangular prism; д) tetrahedron

треугольная или четырехугольная грань ∂C_{ij} характеризуется координатой центра \mathbf{r}_{ij} , единичным вектором внешней нормали \mathbf{n}_{ij} и площадью S_{ij} .

Дискретные значения сеточных функций Q_i относятся к центрам масс расчетных ячеек и равны среднему интегральному значению от непрерывно распределённой величины

$$Q_i = \frac{1}{|C_i|} \int_{C_i} Q d\Omega.$$

Поведение функции f внутри контрольного объема реконструируется линейным полином вида

$$f_i^P(\mathbf{r}) = f_i + \Psi_i (\nabla f_i \cdot (\mathbf{r} - \mathbf{r}_i)),$$

где $\nabla f_i = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$ соответствует градиенту f в центре ячейки, и Ψ обозначена функция-лимитер. Компоненты вектора градиента определяются на основе дискретного аналога формулы интегрального представления градиента

$$\nabla f_i = \frac{1}{|C_i|} \sum_{j \in I_i} \left(g_{ij}^i f_i + g_{ij}^j f_j \right) \mathbf{n}_{ij} S_{ij}. \tag{2.2}$$

Геометрические коэффициенты g_{ij}^i и g_{ij}^j в (2.2) обратно пропорциональны расстояниям от центров ячеек до плоскости общей грани и отвечают условию $g_{ij}^i + g_{ij}^j = 1$.

В соответствии с методом контрольного объема для ячеек сетки записывается балансовое соотношение

$$\frac{\partial \mathbf{Q}_i}{\partial t} + \sum_{j \in I_i} (\mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D) S_{ij} = 0.$$

Конвективный поток через грань контрольного объема вычисляется с использованием одной из базовых схем решения задачи Римана о распаде произвольного разрыва

$$\mathbf{F}_{ij}^C = \Phi(\mathbf{Q}_i^P(\mathbf{r}_{ij}), \mathbf{Q}_j^P(\mathbf{r}_{ij}), \mathbf{n}_{ij}).$$

Выбор схемы зависит от параметров моделируемого течения. В качестве аргументов функции Φ подставляются реконструированные значения газодинамических переменных слева и справа от центра рассматриваемой грани.

Необходимые для определения диффузионных потоков первые производные трех компонент скорости и температуры вычисляются как комбинация усредненных градиентов $\overline{\nabla Q_{ij}}$ и производных по направлению

$$\overline{\nabla Q_{ij}} = \frac{\nabla Q_i + \nabla Q_j}{2},$$

$$\nabla Q_{ij} = \overline{\nabla Q_{ij}} - \left[\overline{\nabla Q_{ij}} \mathbf{t}_{ij} - \frac{Q_j - Q_i}{l_{ij}} \right] \mathbf{t}_{ij}.$$

Параметр $l_{ij} = |\mathbf{r}_j - \mathbf{r}_i|$ соответствует расстоянию между центрами соседних ячеек, $\mathbf{t}_{ij} = \frac{(\mathbf{r}_j - \mathbf{r}_i)}{l_{ij}}$ - единичный вектор, сонаправленный с проходящей через центры прямой.

Постановка граничных условий реализуется путем явного задания значений газодинамических переменных \mathbf{Q}_Γ или потоков $\mathbf{F}_{i\Gamma}$ в центрах масс граней $\partial C_{i\Gamma}$, расположенных на границе расчетной области Γ . Для интегрирования системы газодинамических уравнений по времени используется явная четырехшаговая схема Рунге-Кутты (RK) второго порядка точности.

Последовательная программная реализация вычислительного ядра газодинамического алгоритма состоит из трех функциональных блоков:

- блок линейной реконструкции – вычисление градиентов, цикл по контрольным объемам;
- блок вычисления потоков – определение потоков через грани расчетных ячеек, цикл по граням контрольных объемов;
- блок обновления значений переменных – суммирование потоков и вычисление значений газодинамических переменных на новом шаге RK или новом временном слое, цикл по контрольным объемам.

Поток через общую грань соседних контрольных объемов вычисляется один раз и буферизуется в локальный массив. При выполнении третьего функционального блока значение потока дважды считывается из буфера с разными знаками. Витки циклов всех функциональных блоков не имеют зависимости по данным.

Параллельная версия алгоритма для гетерогенных кластерных систем с графическими ускорителями реализована с использованием программных моделей MPI, OpenMP и CUDA. Каждый MPI-процесс обрабатывает подмножество сеточных ячеек, называемое доменом. По расположению в пределах домена его ячейки делятся на внутренние и интерфейсные многогранники. Внутренние многогранники не имеют общих граней с многогранниками соседних доменов. Расчетная область MPI-процесса объединяет домен и зону запрашиваемых данных. В нее попадают ячейки доменов других процессов, являющиеся связями первого и второго уровня по дуальному графу для интерфейсных ячеек. Значения сеточных функций в ячейках второго уровня связей учитываются при вычислении градиентов в ячейках первого уровня связей. Сеточные функции и их градиенты в ячейках первого уровня связей используются для определения потоков через границу домена. Пример выделения расчетной области для домена структурированной сетки на плоскости показан на Рис. 2.2.

По типу привязки к центральным процессорам или ускорителям MPI-процессы делятся на вычислительные и управляющие соответственно. Вычислительные MPI-процессы резервируют для выполнения функциональных блоков газодинамического

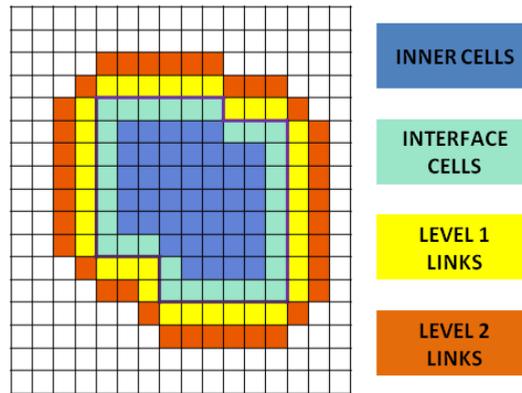


Рис. 2.2. Пример расчетной области MPI-процесса
Fig 2.2. An example of the MPI process computational domain

алгоритма одно или несколько ядер центрального процессора. Распределение вычислений между CPU-ядрами реализовано через постановку директив OpenMP. Управляющий MPI-процесс запускается на одном ядре CPU и устанавливает привязку к графическому ускорителю, куда переносятся основные вычисления. В программных реализациях функциональных блоков для GPU один поток CUDA обрабатывает сеточный элемент или грань. Запущенный на CPU фоновый MPI-процесс управляет постановкой GPU-ядер в очередь на выполнение, передачей MPI-сообщений и копированием данных между процессором и ускорителем.

В параллельном режиме вычислений выполнение функциональных блоков газодинамического алгоритма чередуется с обменом данными. MPI-процесс принимает значения газодинамических переменных в ячейках зоны запрашиваемых данных, а также буферизует и передает значения переменных в собственных ячейках, оказавшихся в зонах запрашиваемых данных других MPI-процессов. Применяемый принцип формирования расчетных областей допускает распараллеливание с одной точкой синхронизации MPI-процессов на каждом шаге RK при условии дублирования вычислений потоков и градиентов вдоль границ разбиения сетки на домены.

Вычислительные процессы работают с явной точкой синхронизации. Функциональные блоки газодинамического алгоритма запускаются строго по завершении процедуры обмена данными. В случае переноса вычислений на графический ускоритель происходит перекрытие обменов данными с вычислениями. Блок-схема алгоритма работы управляющего процесса показана на Рис. 2.3. Вспомогательные операции распараллеливания, в число которых входят буферизация передаваемых данных, копирование данных между CPU и GPU и передача MPI-сообщений, выполняются одновременно с вычислением градиентов во внутренних ячейках домена и потоков через общие грани внутренних расчетных ячеек.

Более подробное описание численного алгоритма и его программной реализации приведено в [8–10].

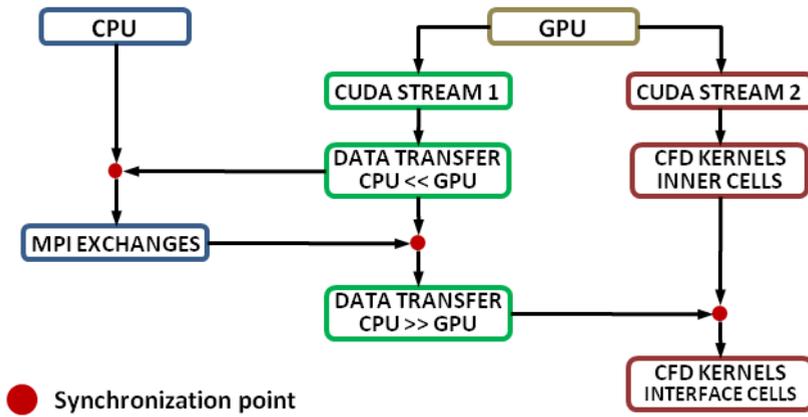


Рис. 2.3. Алгоритм работы управляющего MPI-процесса
 Fig 2.3. Algorithm of the control MPI process

3. Алгоритм балансировки загрузки

Предлагаемый алгоритм статической балансировки загрузки параллельных процессов включает пять этапов:

- определение весов ребер и вершин дуального графа сетки;
- выбор конфигурации запуска;
- определение соотношения между производительностью центральных процессоров и массивно-параллельных ускорителей;
- декомпозиция сетки между узлами кластера;
- декомпозиция подобластей кластеров между вычислительными устройствами.

Весы ребер и вершин дуального графа сетки устанавливаются с учетом особенностей программной реализации CFD-алгоритма. В рассматриваемом случае объем вычислений в процессе полиномиальной реконструкции и обновления значений переменных в ячейке на новом слое по времени кратен числу граней контрольного объема. Общее число граней, обработанных при вычислении потоков, приблизительно равно половине от общего числа граней ячеек, входящих в расчетную подобласть процесса. Поэтому целочисленный вес сеточной ячейки (вершины дуального графа) приравнивается к числу ее граней. Ребра дуального графа эквивалентны связям ячеек через общую грань и не имеют весов.

Проблема выбора конфигурации запуска приложения заключается в поиске оптимального числа выполняемых на узле кластера вычислительных MPI-процессов. Число управляющих процессов соответствует числу установленных на узле ускорителей. Запуск отдельного вычислительного MPI-процесса на каждом из оставшихся CPU-ядер приводит к росту объема вычислений вследствие увеличения площади границы разбиения на домены. Для алгоритмов с дублированием вычислений вдоль границ доменов более эффективным оказывается гибридное распараллеливание MPI+OpenMP.

В таком случае привязка управляющих MPI-процессов к ядрам центральных процессоров должна учитывать число нитей OpenMP, порождаемых вычислительными MPI-процессами.

На Рис. 3.1 показан пример двух вариантов загрузки узла кластера с двумя шестиядерными CPU и двумя GPU-ускорителями. В первом варианте (Рис. 3.1, а) на узле запускается два вычислительных MPI-процесса с вложенным распараллеливанием циклов между пятью OpenMP-нитеями, во втором – пять вычислительных процессов по две нити OpenMP – (рис. 3.1, б). Число управляющих процессов в обоих случаях равно двум. Обозначенная на рисунке жесткая привязка MPI-процессов и нитей OpenMP к ядрам центрального процессора (например, с использованием пары функций `sched_getaffinity` и `sched_setaffinity`) необходима для стабилизации времени работы параллельного приложения.

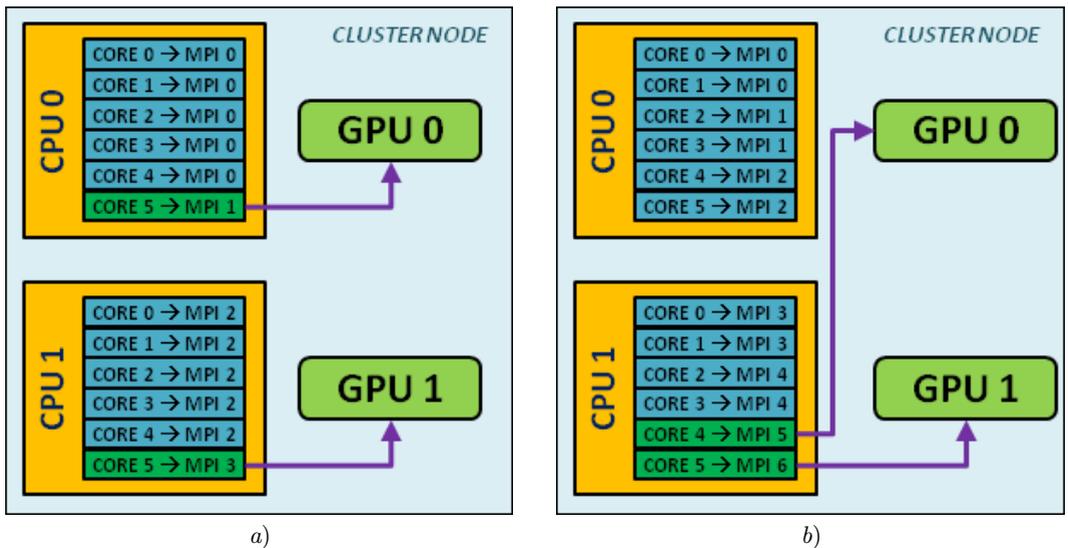


Рис. 3.1. Варианты загрузки узла кластера
Fig 3.1. Examples of run MPI processes on a cluster node

Лучший по быстродействию вариант запуска выявляется экспериментально. Одновременно определяется фактическое соотношение между производительностью вычислений на различных устройствах. При небольшом размере расчетной сетки в качестве исходных данных теста целесообразно рассматривать постановку основной задачи. В противном случае эксперименты проводятся на некотором упрощенном тесте, выполнение которого возможно на одном узле. При первом запуске считается, что быстродействие процессора и ускорителя совпадает. Фиксируется только чистое время выполнения функциональных блоков CFD-алгоритма, а время передачи данных не учитывается. На основе полученных результатов соотношение производительности корректируется. И эксперимент повторяется до тех пор, пока времена выполнения функциональных блоков на различных устройствах не совпадет.

Декомпозиция расчетной области в процессе балансировки загрузки происходит в два этапа (Рис. 3.2). Масштабируемость гетерогенного приложения измеряется числом загруженных узлов кластера, производительность которых считается идентичной. Поэтому на первом этапе (Рис. 3.2, а) граф сетки делится на равные по весу домены пер-

вого уровня, ассоциированные с узлами суперкомпьютера. На втором этапе (Рис. 3.2, б) внутри доменов первого уровня выделяются домены второго уровня, обрабатываемые запущенными на узле управляющими и вычислительными MPI-процессами. Веса доменов второго уровня соответствуют производительности вычислительных устройств. Преимущество двухуровневой схемы декомпозиции состоит в минимизации числа передаваемых сообщений между MPI-процессами, запущенными на разных узлах.

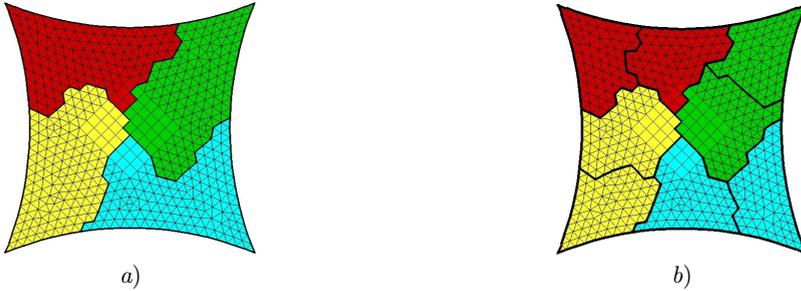


Рис. 3.2. Двухуровневая декомпозиция смешанной сетки: а) декомпозиция между четырьмя узлами на первом уровне; б) последующее разбиение между двумя устройствами на втором уровне

Fig 3.2. Two-level mixed mesh decomposition: a) decomposition between four nodes on the first level; b) partitioning between two devices on the second level

4. Результаты вычислительных экспериментов

Эффективность метода балансировки загрузки исследовалась на примере расчета сверхзвукового обтекания единичной сферы при $M = 2$ и $Re = 300$. Геометрия расчетной области и структура сетки отображены на Рис. 4.1.

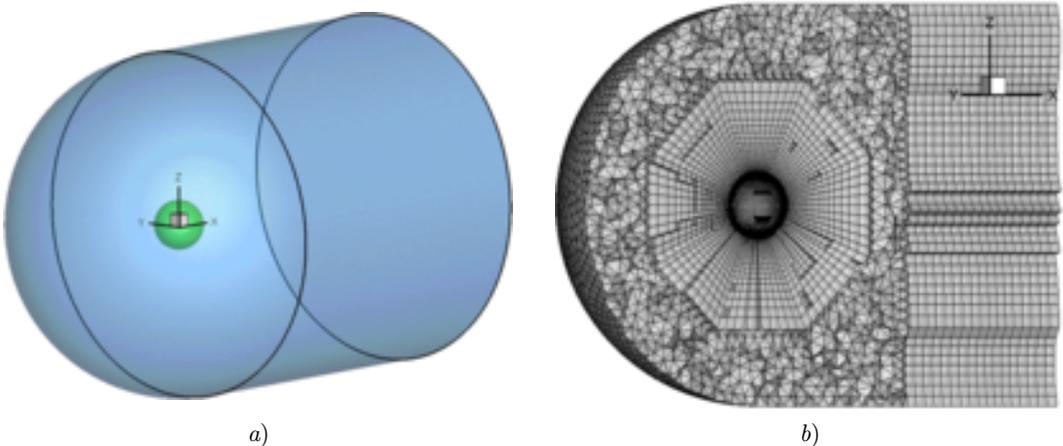


Рис. 4.1. а) расчетная область; б) структура смешанной сетки
Fig 4.1. а)Computational domain; б) mixed mesh structure

В процессе генерации сетки область решения задачи была разделена на 54 подобласти, внутри которых построены согласованные по границам сеточные блоки различных

типов. Плотность пространственной дискретизации увеличивается вблизи поверхности сферы. Зона разрешения пограничного слоя заполнена гексаэдрами и треугольными призмами. В общей сложности сетка состоит из 1 039 620 узлов и 2 160 156 элементов: 1 238 224 тетраэдра, 33 276 четырехугольных пирамид, 172 800 треугольных призм и 715 856 гексаэдров. Выбранная размерность сетки по числу элементов позволяет как запускать расчет на одном узле кластера, так и оценивать масштабируемость гетерогенного алгоритма.

На Рис. 4.2 показана визуализация картины стационарного течения, построенная по результатам расчета. Контрольные параметры течения (например, коэффициент сопротивления $C_d = 1.401$) согласуются с опубликованными в [11] данными.

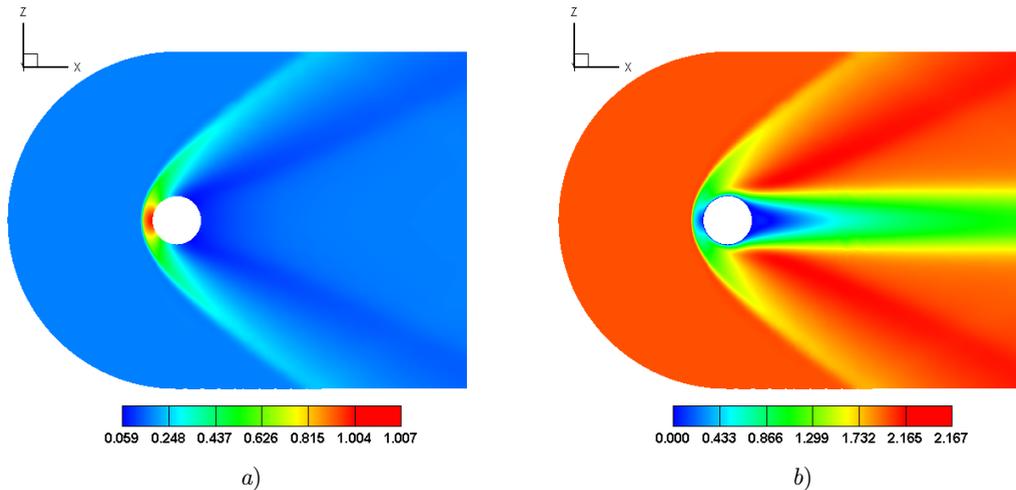


Рис. 4.2. а) поле безразмерного давления; б) числа Маха

Fig 4.2. a) dimensionless pressure distribution; b) Mach number distribution

Для проведения экспериментов по оценке эффективности параллельных вычислений использовался гетерогенный суперкомпьютер К-100, установленный в ИМП им. М. В. Келдыша РАН. Данный кластер состоит из объединенных высокоскоростной коммуникационной системой узлов с несимметричной архитектурой: два центральных шестиядерных процессора Intel Xeon X5670 и три графических ускорителя NVIDIA 2050. Эксперимент включает серию запусков программы на различном числе узлов в трех режимах: загрузка только центральных процессоров, загрузка только графических ускорителей и гетерогенный режим распараллеливания. При запуске каждой конфигурации фиксировалось время выполнения 200 шагов газодинамического алгоритма (800 вызовов функциональных блоков). В качестве начальных данных использовалось поле сеточных функций установившегося течения.

В представленных ниже результатах быстродействие выполнения программы на N узлах характеризуется скоростью вычислений R_N , которая измеряется числом шагов интегрирования по времени в секунду. Ускорение и эффективность распараллеливания вычисляются относительно времени выполнения теста на одном узле.

В режиме загрузки графических ускорителей на узле кластера запускается три управляющих MPI-процесса с привязкой к трем ядам первого CPU. График изменения R^{GPU} показан на Рис. 4.3 зеленой линией; там же красной линией отображен график изменения скорости вычислений в случае балансировки загрузки путем декомпозиции

дуального графа без весов. Из полученных данных следует, что вариант балансировки с разбиением взвешенного графа позволяет снизить время работы программы на 10-18 %. При этом эффективность распараллеливания в обоих случаях не опускается ниже 80 %.

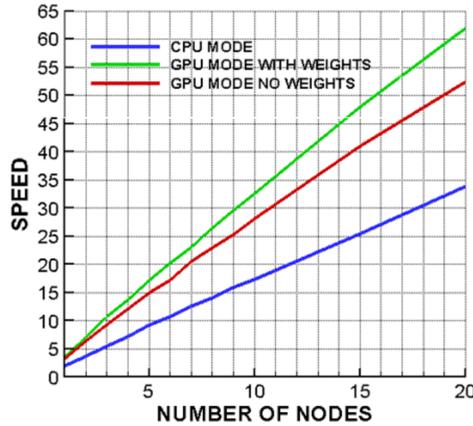


Рис. 4.3. Скорость вычислений
Fig 4.3. Computation speed

В режиме загрузки центральных процессоров на узле запускается четыре вычислительных MPI-процесса с внутренним распараллеливанием функциональных блоков между тремя нитями OpenMP. Полученная скорость вычислений R^{CPU} (синяя линия на Рис. 4.3) оказывается в среднем в 1.88 раза ниже R^{GPU} при практически совпадающей эффективности распараллеливания.

В гетерогенном режиме на узле выполняется три вычислительных и три управляющих MPI-процесса (Рис. 4.4).

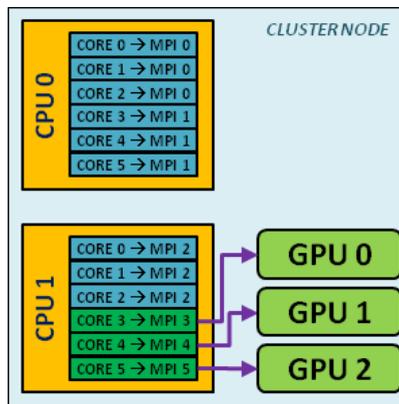


Рис. 4.4. Конфигурация запуска
Fig 4.4. Program launch configuration

При первичном запуске программы на одном узле кластера с равномерной балан-

сировкой загрузки между вычислительными и управляющими MPI-процессами минимальное и максимальное время выполнения теста функциональных блоков газодинамического алгоритма отличаются приблизительно в 2.43 раза, что соответствует соотношению производительности 0.41:1. При последующих настроенных запусках, уже с обмени данными, соотношение меняется до 0.35:1. Корректировка в меньшую сторону относительной производительности вычислительных MPI-процессов объясняется падением производительности параллельных вычислений на CPU вследствие блокировки запуска функциональных блоков CFD-алгоритма до завершения процедуры обмена данными.

График скорости вычислений в гетерогенном режиме R^{HET} приведен на Рис. 4.5. Для сравнения приводится график теоретически ожидаемой по результатам предшествующих запусков скорости вычислений $R_N^{OPT} = R_N^{GPU} + 0.75R_N^{CPU}$. Поправочный коэффициент при R_N^{GPU} учитывает использование трех ядер центрального процессора для запуска управляющих MPI-процессов.

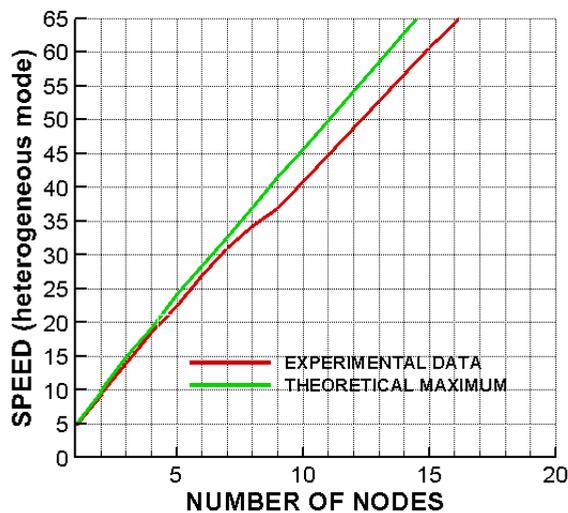


Рис. 4.5. Скорость вычислений в гетерогенном режиме
Fig 4.5. Computation speed for heterogeneous mode

Соотношение между фактической и ожидаемой скоростью вычислений находится на отрезке $\frac{R_N^{HET}}{R_N^{OPT}} \in [0.91; 0.96]$. С ростом числа узлов эффективность распараллеливания плавно опускается с 97 до 82 %. На пределе масштабируемости задачи (20 узлов, 120 MPI-процессов, 240 процессорных ядер, 60 ускорителей) время выполнения теста составляет 2.527 с. Другими словами, один шаг RK занимает порядка 0.003 с, включая обмены данными. Представленные результаты могут рассматриваться как подтверждение эффективности описанного в статье метода балансировки загрузки.

Благодарности. Расчёты проводились на суперкомпьютерах ЦКП ИПМ им. М. В. Келдыша РАН.

СПИСОК ЛИТЕРАТУРЫ

1. Liou M.-S., Kao K.-H. Progress in grid generation: from chimera to DRAGON grids. Cleveland: National Aeronautics and Space Administration, 1994. 26 p.
2. Zaspel P., Griebel M. Solving incompressible two-phase flows on multi-GPU clusters // Computers and fluids, 2013, v. 80, p. 356–364. DOI: <https://doi.org/10.1016/j.compfluid.2012.01.021>
3. Davydov A. A., Chetverushkin B. N., Shilnikov E. V. Simulating flows of incompressible and weakly compressible fluids on multicore hybrid computer systems // Comput. Math. and Math. Phys. Vol. 50. pp. 2157–2165 (2010). DOI: <https://doi.org/10.1134/S096554251012016X>
4. Krasnov M. M., Kuchugov P. A., Ladonkina M. E., Tishkin V. F. Discontinuous Galerkin method on three-dimensional tetrahedral grids: Using the operator programming method // Math. Models Comput. Simul. 2017. Vol. 9, Issue 5. pp. 529–543. DOI: <https://doi.org/10.1134/S2070048217050064>
5. Gorobets A., Soukov S., Bogdanov P. Multilevel parallelization for simulating turbulent flows on most kinds of hybrid supercomputers // Computers and Fluids. 2018. Vol. 173. pp. 171–177. DOI: <https://doi.org/10.1016/j.compfluid.2018.03.011>
6. Schloegel K., Karypis G., Kumar V. Parallel multilevel algorithms for multi-constraint graph partitioning // Euro-Par 2000 Parallel Processing / ed by. A. Bode, T. Ludwig, W. Karl, R. Wismuller. Berlin-Heidelberg: Springer, 2000. Vol. 1900. pp. 296–310. DOI: https://doi.org/10.1007/3-540-44520-X_39
7. Golovchenko E. N., Kornilina M. A., Yakobovskiy M. V. Algorithms in the parallel partitioning tool GridSpiderPar for large mesh decomposition // Proceedings of the 3rd International Conference on Exascale Applications and Software (EASC 2015). Edinburgh: University of Edinburgh, 2015. pp. 120–125.
8. Blazek J. Computational fluid dynamics: Principles and applications. Amsterdam: Elsevier, 2001. 470 c.
9. Kim S., Caraeni D., Makarov B. A multidimensional linear reconstruction scheme for arbitrary unstructured grids : technical report // AIAA 16th Computational Fluid Dynamics Conference. 2003. DOI: <https://doi.org/10.2514/6.2003-3990>
10. Soukov S. A., Gorobets A. V., Bogdanov P. B. Portable solution for modeling compressible flows on all existing hybrid supercomputers // Math. Models Comput. Simul. 2018. Vol. 10. pp. 135–144. DOI: <https://doi.org/10.1134/S2070048218020138>
11. Nagata T., Nonomura T., Takahashi S., Mizuno Y., Fukuda K. Investigation on subsonic to supersonic flow around a sphere at low Reynolds number of between 50 and 300 by direct numerical simulation // Physics of Fluids. 2016. Vol. 28. pp. 056101-1–056101-20. DOI: <https://doi.org/10.1063/1.4947244>

*Поступила 18.03.2021; доработана после рецензирования 11.04.2021;
принята к публикации 6.05.2021*

Информация об авторе:

Суков Сергей Александрович, старший научный сотрудник, ИПМ им. М. В. Келдыша РАН (125047, Россия, г. Москва, Миусская пл., д. 4), кандидат физико-математических наук, ORCID: <http://orcid.org/0000-0002-0667-6955>, ssoukov@gmail.com

Автор прочитал и одобрил окончательный вариант рукописи.

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Original article

MSC2020 65Y05

Load balancing method for heterogeneous CFD algorithms

S. A. Soukov

Federal Research Center Keldysh Institute of Applied Mathematics, Russian Academy of Sciences (Moscow, Russian Federation)

Abstract. The problem of load balancing for unstructured heterogeneous numerical algorithms for simulation of physical processes is considered. A computational distribution method for hybrid supercomputers with multicore CPUs and massively parallel accelerators is described. The load balancing procedure includes determination of dual graph vertices and edges weights, devices' performance test and two-level decomposition of the computational mesh based on domain decomposition method. First level decomposition involves the graph partitioning between supercomputer nodes. On the second level node subdomains are partitioned between the MPI-processes running on the nodes. The details of the proposed approach are considered on the example of an unstructured finite-volume algorithm for modeling the Navier-Stokes equations with polynomial reconstruction of variables and explicit time integration scheme. The parallel version of the algorithm is developed using the MPI, OpenMP and CUDA programming models. The parameters of performance, parallel efficiency and scalability of the heterogeneous program are given. The results mentioned are obtained during the simulation of a supersonic flow around a sphere on a mixed mesh consisting of tetrahedrons, triangular prisms, quadrangular pyramids and hexagons.

Key Words:

For citation: S. A. Soukov. Load balancing method for heterogeneous CFD algorithms. *Zhurnal Srednevolzhskogo matematicheskogo obshchestva*. 23:2(2021), 193–206. DOI: <https://doi.org/10.15507/2079-6900.23.202102.193-206>

REFERENCES

1. M.-S. Liou, K.-H. Kao, *Progress in grid generation: from chimera to DRAGON grids*, National Aeronautics and Space Administration Publ., Cleveland, 1994, 26 p.
2. P. Zaspel, M. Griebel, “Solving incompressible two-phase flows on multi-GPU clusters”, *Computers and Fluids*, **80** (2013), 356–364. DOI: <https://doi.org/10.1016/j.compfluid.2012.01.021>
3. A. A. Davydov, B. N. Chetverushkin, E. V. Shilnikov, “Simulating flows of incompressible and weakly compressible fluids on multicore hybrid computer systems”, *Comput. Math. and Math. Phys.*, **50** (2010), 2157–2165. DOI: <https://doi.org/10.1134/S096554251012016X>

4. M. M. Krasnov, P. A. Kuchugov, M. E. Ladonkina, V. F. Tishkin, “Discontinuous Galerkin method on three-dimensional tetrahedral grids: Using the operator programming method”, *Math. Models Comput. Simul.*, **9**:5 (2017), 529–543. DOI: <https://doi.org/10.1134/S2070048217050064>
5. A. Gorobets, S. Soukov, P. Bogdanov, “Multilevel parallelization for simulating turbulent flows on most kinds of hybrid supercomputers”, *Computers and Fluids*, **173** (2018), 171–177. DOI: <https://doi.org/10.1016/j.compfluid.2018.03.011>
6. K. Schloegel, G. Karypis, V. Kumar, “Parallel multilevel algorithms for multiconstraint graph partitioning”, *Euro-Par 2000 Parallel Processing*, **1900**, eds. A. Bode, T. Ludwig, W. Karl, R. Wismuller, Springer, Berlin-Heidelberg, 2000, 296–310 DOI: https://doi.org/10.1007/3-540-44520-X_39.
7. E. N. Golovchenko, M. A. Kornilina, M. V. Yakobovskiy, “Algorithms in the parallel partitioning tool GridSpiderPar for large mesh decomposition”, *Proceedings of the 3rd International Conference on Exascale Applications and Software (EASC 2015)*, University of Edinburgh Publ., Edinburgh, 2015, 120–125.
8. J. Blazek, *Computational fluid dynamics: principles and applications*, Elsevier, Amsterdam, 2001, 470 p.
9. S. Kim, D. Caraeni, B. Makarov, “A multidimensional linear reconstruction scheme for arbitrary unstructured grids”, technical report, *AIAA 16th Computational Fluid Dynamics Conference* (Orlando, Florida American Institute of Aeronautics and Astronautics. June 2003), 2003 DOI: <https://doi.org/10.2514/6.2003-3990>.
10. S. A. Soukov, A. V. Gorobets, P. B. Bogdanov, “Modeling Compressible Flows on All Existing Hybrid Supercomputers”, *Math. Models Comput. Simul.*, **10** (2018), 135–144. DOI: <https://doi.org/10.1134/S2070048217050064>
11. T. Nagata, T. Nonomura, S. Takahashi, Y. Mizuno, K. Fukuda, “Investigation on subsonic to supersonic flow around a sphere at low Reynolds number of between 50 and 300 by direct numerical simulation”, *Physics of Fluids*, **28** (2016), 056101-1–056101-20. DOI: <https://doi.org/10.1063/1.4947244>

Submitted 18.03.2021; Revised 11.04.2021; Accepted 6.05.2021

Information about the author:

Sergey A. Soukov, Senior Researcher, Federal Research Center Keldysh Institute of Applied Mathematics, Russian Academy of Sciences (4 Miusskaya Sq., Moscow 125047, Russia), Ph.D. (Physics and Mathematics), ORCID: <http://orcid.org/0000-0002-0667-6955>, ssoukov@gmail.com

The author have read and approved the final manuscript.

Conflict of interest: The author declare no conflict of interest.