УДК 519.8

АВТОМАТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРИ ДЕКОМПОЗИЦИИ ДЛЯ ГЕТЕРОГЕННЫХ И ГИБРИДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В МЕТОДИКЕ ТИМ-3D

А. А. Воропинов, Т. Н. Половникова, Ф. О. Голомидов, А. К. Шмелёва

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе рассматриваются вопросы выполнения декомпозиции для гибридных и гетерогенных вычислительных систем в методике ТИМ-3D. Для учета разной производительности вычислительных устройств вводится специальная весовая функция, значения которой используются для определения размеров параобластей на этапе декомпозиции. В докладе рассматривается алгоритм автоматического определения значений данной весовой функции, основанный на расчете небольшой задачи в режиме дублирования без обменов всеми процессами задачи. Величина обратная времени выполнения нескольких шагов выступает в качестве значений весовой функции производительности процессов. Помимо этого данный алгоритм позволяет учесть «шумность» вычислительной системы и выявить «грязные» узлы.

Ключевые слова: декомпозиция, весовая функция производительности, гетерогенные вычислительные системы, методика ТИМ-3D.

Введение

Современные вычислительные системы (ВС), как правило, используют кластерную архитектуру. При этом все большее распространение получают гибридные ВС, когда в рамках одного узла используются вычислительные устройства разного типа, обладающие разнородными характеристиками по:

- скорости выполнения арифметических операций;
- количеству процессорных ядер;
- объему и скорости работы оперативной памяти;
- типу доступа к файловой системе;
- скорости коммуникаций.

Использование гибридных вычислительных систем требует разработки специальных алгоритмов декомпозиции, позволяющие учесть данные аспекты. В данной работе рассматриваются алгоритмы, реализованные в рамках методики ТИМ-3D [1].

В методике ТИМ-3D используется метод трехуровневого распараллеливания [2]. На первом уровне используется распараллеливание по математическим областям, на втором уровне – внутри области по параобластями. На первых двух уровнях используется геометрический принцип распределения работы и модель распределенной памяти (MPI). На третьем уровне распараллеливаются итерации счетных циклов в модели общей памяти с использованием OpenMP. Данный метод подходит и для гибридных ВС.

При проведении тестовых расчетов использовалась ВС «Каскад» ООО «Центр компетенции и обучения» [3].

Учет характеристик гибридных ВС в параллельных алгоритмах ТИМ-3D

С точки зрения метода распараллеливания методики ТИМ-3D наиболее существенным является разная скорость выполнения операций разными вычислительными устройствами. Остальные характеристики вычислительных устройств не накладывают существенных ограничений на алгоритмы.

Так использование OpenMP позволяет скрыть различия по количеству процессорных ядер между вычислительными устройствами. Алгоритмы OpenMP распараллеливания является «самобалансирующимися» – нить выполняет выделенный объем работы и получает следующую порцию вычислений. При этом количество нитей в алгоритмах в явном виде не программируется. В результате различие в количестве доступных ядер определяется параметрами запуска. Таким образом, уровень OpenMP распараллеливания можно в явном виде не учитывать при распределении вычислительной работы.

Скорость работы с оперативной памятью можно считать аспектом общей скорости работы вычислительного устройства. Что касается объема доступной памяти, то на этапе формирования параобластей вводится локальная нумерация элементов сетки. В результате в процессе проведения расчетов не требуется большого объема оперативной памяти. Тем не менее, управляющие операции с математическими областями должны выполняться на вычислительных устройствах с большим объемом памяти.

Аналогично вводится ограничение и в связи с разным типом доступа к файловой системе.

Разная скорость коммуникаций может не учитываться в связи с тем, что большинство параллельных алгоритмов методики использует асинхронные обмены с наложением их с вычислениями. В результате при большом объеме вычислительной работы скорость обмена информацией не оказывает существенного влияния на общую скорость работы.

Весовая функция производительности процессов

Рассматриваемые ниже алгоритмы ориентированы на архитектуру Intel x86, которая реализована в различных универсальных процессорах и сопроцессорах семейства MIC. (Сопроцессоры Intel Xeon Phi позволяют использовать так называемый «симметричный» режим работы, при котором на каждом из них запускается независимый MPI процесс. Это позволяет представлять гибридную ВС как гетерогенную.) При этом разные процессы получают независимый набор процессорных ядер (универсальных или сопроцессорных), работающих на общей памяти. В результате задачу декомпозиции для гетерогенных ВС можно рассматривать без учета особенностей конкретного вычислительного устройства. При этом разница между вычислительными устройствами сводится к разной скорости работы процессов запущенных на соответствующем вычислительном устройстве.

Таким образом каждый MPI процесс можно охарактеризовать некоторым числом, характеризующим производительность выделенных для него вычислительных ядер. Чем выше значение – тем выше производительность соответствующего процесса. При наложении условия, что данные числа используют единую систему единиц можно говорить о введении весовой функции производительности процесса. Для удобства работы на значения данной функции накладывается ограничение – что сумма весов должна быть равна 1, то есть значение весовой функции характеризует долю общей работы, которую может выполнить процесс. Отметим, что данное ограничение не является существенным, так как всегда можно выполнить нормировку значений. Такая схема определения производительности используется, например, в библиотеке ParMeTiS при решении задачи о разрезании графа на подграфы разного размера [4].

Сложностью использования весовой функции производительности процессов является то, что ее значения трудно получить исходя из общих характеристик алгоритмов. Дополнительную трудность представляет и то, что в гибридных вычислительных системах для разных операций соотношения в производительности существенно различаются. Например, Intel Xeon Phi 7120P обладает высокой скоростью работы векторных операций, однако при работе с памятью существенно проигрывает универсальному процессору. В связи с этим наиболее простым и универсальным представляется определение значений весовых функций исходя из выполнения пробных вычислений по используемым алгоритмам. На основе полученных характеристик можно определить значения весовой функции производительности процессов.

Принципы декомпозиции в методике ТИМ-3D

Трехуровневое распараллеливание методики ТИМ-3D требует выполнения декомпозиции данных в три этапа и введения нескольких весовых функций.

На первом этапе определяется распределение математических областей по процессам с учетом дополнительных ограничений, связанных например с неполным доступом к файловой системе у некоторых процессов. Каждой области присваивается вес, отражающий объем вычислений, связанный с этой областью. В качестве веса используется время на расчет последних 100 шагов данной матобласти, а если оно недоступно (например, в начале расчета), то количество ячеек области. Использование времени за 100 шагов позволяет с одной стороны отразить текущее состояние задачи с точки зрения объема вычислений, с другой стороны позволяет сгладить влияние отдельных шагов.

На втором этапе определяются количество и размер компактов для каждой математической области. Компакт – набор ячеек одной области. На основе компактов формируются параобласти. Области разбиваются на компакты независимо друг от друга. При этом размер и распределение параобластей по процессам формируется с одной стороны таким образом, чтобы получить сбалансированную декомпозицию. С другой стороны количество порождаемых параобластей минимизируется (для уменьшения накладных расходов на организацию параллельных вычислений).

На третьем этапе декомпозиции производится декомпозиция математических областей на компакты количество и размер которых определен на втором этапе. При этом решается задача о разрезании графа на подграфы при помощи алгоритмов библиотек MeTiS [4], SCOTCH [5] или геометрическо-топологической декомпозиции [6]. В качестве вершин графа выступают ячейки сетки, а ребра отражают соседство между ячейками (как правило, через грань сетки). На третьем этапе используются веса ячеек сетки. При определении данного веса возникает ряд дополнительных трудностей. Во-первых, не всегда возможно выделить объем вычислений связанных с конкретной ячейкой сетки (например, при решении систем линейных уравнений). Во-вторых, попытки засекать время вычислений, необходимое для отдельных элементов сетки, приводит к многократному росту времени счета, так как процедура засечки времени сама по себе является достаточно дорогостоящей. В-третьих, хранение времени счета 100 шагов для каждой ячейки требует чрезмерных объемов памяти. По этим причинам была разработана следующая схема:

Засечки делаются для крупных блоков. Время, затраченное на выполнение блока, разделяется между всеми участвующими в нем ячейками сетки поровну.

Для расчета веса в ячейке используется две величины – время счета t на текущем шаге (определяемое по участию ячейки в тех или иных счетных блоках) и непосредственно вес ячейки w.

Суммарное время по всем ячейкам области равно времени расчета области на текущем шаге.

Вес ячейки в конце каждого шага пересчитывается по формуле $w = \alpha\tilde{w} + t$, где $\tilde{w}$ – вес ячейки на предыдущем шаге, $0 < \alpha < 1$ – коэффициент уменьшения влияния веса на предыдущих шагах. Введение параметра α позволяет с одной стороны учесть время, затраченное на предыдущих шагах, с другой стороны позволяет его постепенно уменьшить. Например, при $\alpha = 0{,}955$ вклад в весовую функцию текущего времени t за 100 шагов уменьшится в ~100 раз.

Отметим, что вес области и ячеек связан по правилу: на последнем шаге вес области равен сумме времен t по всем ячейкам этой области.

Декомпозиция для гетерогенных вычислительных систем

Алгоритм первого этапа, на котором по процессам распределяются математические области, для гетерогенных вычислительных систем выглядит следующим образом:

1. Определяется количество доступных процессов для распределения областей, имеющих полный доступ к файловой системе.

2. Производится перенормировка весовой функции для процессов, доступных для распределения областей. (Чтобы суммарное значение их долей было равно единице – это значение весовой функции используется только на первом этапе декомпозиции.)

3. Для каждого процесса определяется вес, который должен быть к нему отнесен (суммарный вес всех областей, умноженный на значение весовой функции процесса) – это число объявляется оставшимся весом процесса.

4. В цикле по количеству областей:
   1) Выбирается не распределенная математическая область с максимальным весом.
   2) Выбранная матобласть относится к процессу с максимальным значением оставшегося веса.
   3) Значение оставшегося веса соответствующего процесса уменьшается на величину вес области.

Данный алгоритм близок к алгоритму «Best-Fit» задачи «О рюкзаке» теории графов. Модификация алгоритма связана с дополнительным выбором наибольшего числа.

На втором этапе для каждой области определяется количество и размер компактов. При мелкозернистом распараллеливании каждая область разбивается на компакты независимо от других областей. Однако при этом количество порождаемых параобластей подбирается таким образом, чтобы с одной стороны нагрузка между процессами была сбалансированной, а с другой стороны количество разбиений было минимальным. Эта задача близка к задаче теории графов о минимизации времени обработки (minimum makespan [7].

Ранее в методике ТИМ-3D, при декомпозиции для однородных вычислительных систем при решении этой задачи предполагалось что компакты, порождаемые для области, имеют равный вес (примеры таких декомпозиций на рис. 2, рис. 3). Это несколько упрощало использование алгоритмов решения задачи о разрезании графа на подграфы.

Однако для гетерогенной вычислительной системы такой подход становится неприемлемым. Очевидно, что даже для одно областного случая выполнение требования о равности компактов на гетерогенной вычислительной системе приводит к возникновению дисбаланса. По этой причине от данного ограничения пришлось отказаться и ввести возможность декомпозиции на неравные по весу компакты. Помимо возможности осуществления декомпозиции для гетерогенных вычислительных систем введение такой возможности позволяет в большинстве случаев добиться лучшей сбалансированности, чем по старому алгоритму (см. таблица 1).

Таблица 1

Сравнение алгоритмов декомпозиции с одинаковым и разным размером компактов

Узлов	**Количество параобластей**		**Разбалансированность, %**		**Эффективность, %**	
	равные	неравные	равные	неравные	равные	неравные
2	–	2	–	1 %	–	87,54 %
3	2	3	7 %	1 %	79,89 %	81,03 %
4	3	3	17 %	1 %	74,13 %	81,87 %
5	3	3	19 %	1 %	64,15 %	79,05 %
6	4	5	7 %	1 %	73,98 %	76,80 %
7	5	6	9 %	9 %	69,56 %	77,21 %
8	5	7	14 %	6 %	74,15 %	71,91 %
9	6	9	19 %	7 %	71,37 %	74,55 %
10	8	12	19 %	1 %	70,40 %	73,25 %
12	11	14	22 %	1 %	64,18 %	76,98 %
14	13	16	11 %	1 %	66,25 %	64,22 %
16	15	18	6 %	1 %	61,19 %	64,20 %
18	17	20	7 %	1 %	53,34 %	64,17 %
20	19	22	9 %	1 %	53,09 %	61,66 %

Пример работы алгоритмов с равным и не равным размером компактов для однородной ВС представлен на рис. 1. В примере использован двумерный расчет из 6 одинаковых по количеству ячеек областей, декомпозиция построена для 8 процессов. В алгоритме с равным размером компактов сбалансированная декомпозиция получается при разбиении каждой области на 8 компактов (64 парообласти), при декомпозиции с неравными компактами каждая область разбивается на 2 компакта (12 парообластей). Благодаря уменьшению количества парообластей уменьшается и количество обменов, в результате скорость счета возрастает на ~25 %.

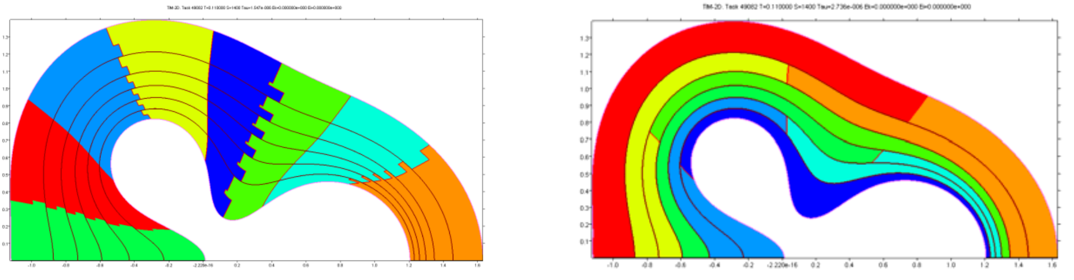


Рис. 1. Сравнение результаты работы алгоритмов декомпозиции с равным и неравным размером компактов

При работе алгоритма на втором этапе декомпозиции работа идет также с числами – весами матобластей, которые распределяются по группам (соответствующие уже всем процессам). При необходимости число может быть разделено на фрагменты (порождение парообластей), каждый из которых относится в свою группу.

Алгоритм для гетерогенных ВС выглядит следующим образом:

1. Определяется суммарный вес (сумма чисел).
2. Определяется усредненный вес каждой группы (суммарный вес, умноженный на долю группы от общей суммы чисел).
3. Определяется максимальный вес для каждой группы (усредненный вес, увеличенный на максимальную допустимую долю разбалансировки).
4. Распределяются «маленькие» числа. Алгоритм распределения «маленьких» чисел близок к алгоритму «best-fit», используемый на первом этапе декомпозиции. В цикле по общему количеству чисел:
   a. Выбираем наибольшее не распределенное число.
   b. Ищем группу с минимальным свободным весом, вмещающим рассматриваемое число без деления.
   c. Если группа не найдена – число отмечается как «большое».
   d. Если группа найдена – число помещается в эту группу. У группы увеличивается задействованный вес.
5. Переходим к «большим» числам, в случае если таковые остались. Если больших чисел не осталось, то это эквивалентно пообластному режиму распараллеливания.
6. Определяется доля от общего веса тех групп, которые заняты еще не полностью. (Занятыми полностью на данном этапе считаются группы с загрузкой более 95 % от «рекомендованной»).
7. Рассчитывается суммарный вес S не распределенных чисел, вместе с весом «маленьких» чисел в группах занятых не полностью.
8. Для не полностью занятых групп определяем, что новый вес не будет превышать долю от оставшегося. Если превышает, то такая группа, из состава не полностью занятых, исключается.
9. Для остальных не полностью занятых групп рассчитывается новый максимальный вес (аналогично шагу 3, но по величинам из п. 7 и доли от не полностью занятых групп).
10. В цикле по числам («маленькие» числа которые уже распределены по группам – пропускаются) – в нем вложенный цикл по всем не занятым группам (полностью занятые группы пропускаются).
    1) От числа «отрезается» кусок соответствующий заполнению группы до конца, либо остаток числа.

2) Заполняется коэффициент фрагмента числа.
3) Увеличивается наполненность группы.
4) Если исходное число (до нарезки) не было отнесено ни к одной группе, и значение весовой функции текущей группы положительно, то исходное число относится в эту группу (это соответствует определению управляющего процесса для матобласти).
5) Если число использовано полностью, то цикл по группам прекращается – переходим к следующему большому числу.

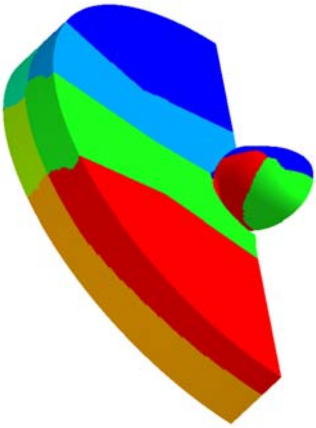


Рис. 2. Пример декомпозиции многообластной задачи для однородной ВС (SCOTCH)

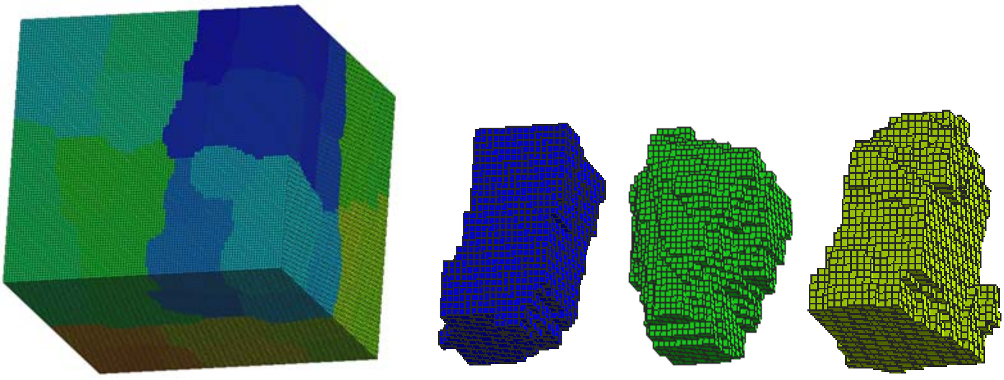


Рис. 3. Пример декомпозиции одной области и вид отдельных компактов (MeTiS)

Примеры декомпозиций для гетерогенных вычислительных систем с принудительно заданным соотношением производительности представлены на рис. 4.

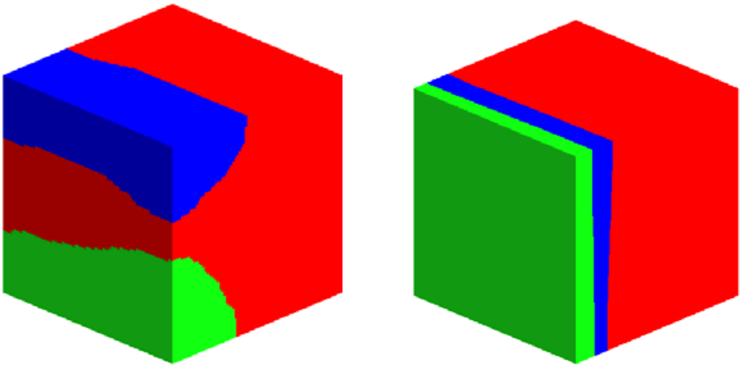


Рис. 4. Декомпозиция области из миллиона шестигранных ячеек на 3 компакта
с соотношение 0,8:0,1:0,1 (слева – MeTiS, справа – SCOTCH)

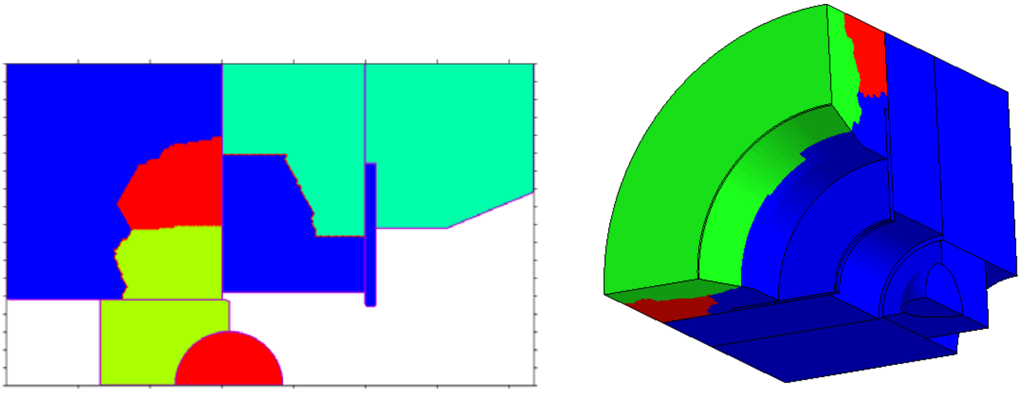


Рис. 5. Пример гетерогенной декомпозиции многообластной задачи на 4 процесса с соотношением 0,4: 0,4:0,1:0,1

Полуавтоматическое определение значений весовой функции производительности процессов

Ручное задание соотношений производительности процессов при запуске на большое количество узлов оказывается достаточно трудоемко, так как значение функции надо указать для каждого процесса в отдельности. При этом как правило количество типов вычислительных устройств не велико и значения весовых функций можно привязать к этим типам. В связи с этим разработан алгоритм полуавтоматического задания производительности процессов. Для этого введены следующие правила:

- вычислительное устройство определяется по имени, возвращаемом функцией MPI_GET_PROCESSOR_NAME;
- цифровые символы данного имени определяют номер устройства;
- остальные символы определяют тип устройства.

Например, при запуске на вычислительном устройстве «abc.12_34» номером устройства будет считаться «1234», а его тип «abc._». Такое разделение позволяет задавать значение весовой функции по типам, а не по отдельным процессам. Важным моментом является случай, когда на одном вычислительном устройстве запускается несколько процессов. В этом случае считается, что процессы делят данное устройство в равных долях. Например, при запуске 2 MPI процессов на универсальном узле, и по 1 процессу на сопроцессор, с соотношением значений производительности 8 к 1 реальные значения весовой функции будут определены как 0,4 для процессов на универсальной части и 0,1 для сопроцессоров (пример на рис. 5).

Описанная схема позволяет быстро изменять значения весовой функции. Результаты полученных замеров с вариацией значений весовой функции по типам вычислительных устройств представлены в таблица 2. Как видно из таблицы введение весовой функции производительности позволило достичь ускорения 1,4. Пример одной из декомпозиций по полученным значениям весовой функции производительности для бруска представлен на рис. 6.

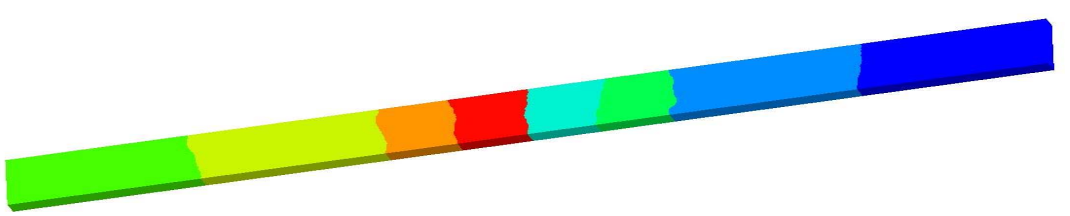


Рис. 6. Вид декомпозиции для двух узлов со значением весовой функции производительности 0,18 Host и 0,07 MIC

Таблица 2

Ускорение счетных блоков газодинамики при вариации значений весовой функции производительности процессов

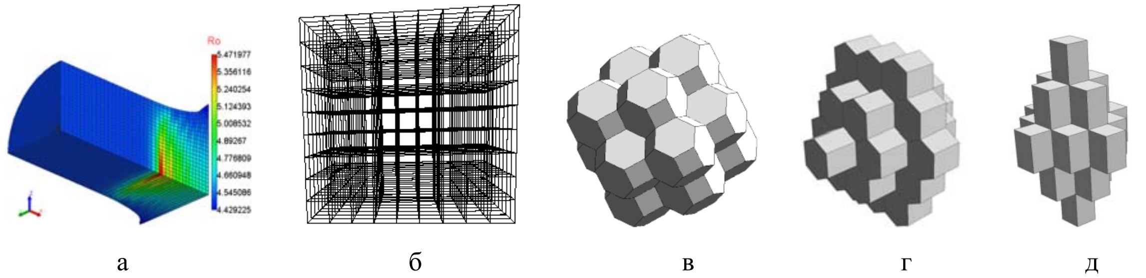
Узлов	Host		MIC		Весовая функция производительности		Время	Ускорение
	MPI	OpenMP	MPI	OpenMP	Host	MIC		
2	2	14	–	–	1	–	223,845	1,00
2	2	14	1	240	0,12	0,13	213,315	1,05
2	2	14	1	240	0,14	0,11	181,702	1,23
2	2	14	1	240	0,15	0,10	171,736	1,30
2	2	14	1	240	0,16	0,09	168,304	1,33
2	2	14	1	240	0,17	0,08	164,275	1,36
2	2	14	1	240	0,18	0,07	160,998	1,39
2	2	14	1	240	0,19	0,06	178,701	1,25

Автоматическое определение производительности процессов

Полуавтоматическое определение значений функции производительности процессов хотя и упрощает управление запуском задачи, однако не лишено недостатков. При расчете реальных задач необходимо выполнить ряд пробных запусков по подбору оптимальных значений весовой функции производительности процессов. Такие запуски с одной стороны увеличивают время на подготовку расчета, с другой стороны подобранные вручную значения могут обладать не высокой точностью. По этим причинам были разработаны алгоритмы автоматического определения весовой функции производительности. Эти алгоритмы заключаются в расчете нескольких шагов малой задачи в режиме дублирования вычислений, без выполнения обменов. Величина обратная к времени выполнения этих шагов отражает производительность и после нормировки может быть использована в качестве весовой функции производительности процессов.

Операция автоматического определения весовой функции производительности процессов выполняется «на лету» при инициализации счета. При расчете малой задачи используются те же параметры, что и при проведении основного расчета.

Для расчета малой задачи может быть использована сетка, считанная из разреза или быстро построенная по алгоритмам диаграммы Вороного или регулярной шестигранной сетки (представленной в неструктурированном виде) в кубе. Примеры сеток представлены на рис. 7. Отметим, что в качестве сетки из дополнительного разреза может быть использована любая сетка. На рисунке представлен пример, использованный в последующих тестовых расчетах. Дополнительный разрез может быть получен специально при подготовке задачи и быть близким к основному расчету по типу сетки и моделируемым приближениям. Это позволяет более точно определить значения весовой функции производительности процессов для конкретной задачи. С другой стороны прямое построение сеток позволяет быстро получать значения данной весовой функции без дополнительных действий со стороны пользователя.



а б в г д

Рис. 7. Примеры сеток для расчета «малой задачи»: а – из дополнительного разреза; б – шестигранная сетка; в-д – диаграмма Вороного с различной расстановкой центров: в – тела Вороного; г – шестиугольные призмы; д – прямоугольные параллелепипеды

Дополнительным преимуществом автоматического определения весовой функции производительности процессов является возможность учета «шумности» ВС (операционная система может одновременно со счетом задачи выполнять системные задачи, что несколько снижает производительность), а также выявлять «грязные» узлы («зависшие» процессы от предыдущих задач – в этом случае производительность может снижаться многократно).

Поскольку автоматический расчет весовой функции производительности процессов выполняется «на лету» на этапе инициализации, то он должен производится максимально быстро. Для этого с одной стороны ограничено максимальное количество используемых ячеек – не более 1 млн., и количество рассчитываемых шагов – не более 10. Введение расчета нескольких шагов позволяет с одной стороны сгладить влияние отдельного шага из-за спонтанных «шумов» или «плавающей» вычислительной нагрузки. С другой стороны позволяет исключить влияние первого шага, на котором выделяется память для рабочих данных, в результате чего время на расчет может быть существенно выше, чем у последующих шагов.

Время на получение сеток разного типа представлено в таблица 3. Как видно из представленных результатов построение сетки диаграммы Вороного (призматическая и из тел Вороного) требует больших временных затрат и в связи с этим на практике может использоваться для количества точек не более 10 тыс. Построение шестигранной сетки по регулярным алгоритмам оказывается даже более быстрым, чем чтение малой задачи из файла.

Таблица 3

Время на получение сеток разного типа

Тип получения малой сетки	Кол-во точек / Тип вычислительного устройства / Время					
	10 000		100 000		1 000 000	
	Host	MIC	Host	MIC	Host	MIC
Из разреза	1,542	1,554	5,658	5,663	46,450	46,454
Шестигранная	0,017	0,378	0,170	2,645	1,730	24,699
Призматическая	4,353	84,269	36,621	748,96	347,52	>1000
Тела Вороного	10,66	217,18	115,57	>1000	>1000	>1000

Вторая часть накладных расходов состоит в расчете нескольких шагов малой задачи. Время на их выполнение для шестигранной сетки из 1 млн. ячеек для 1 и 5 узлов ВС «Каскад» представлено в табл. 4. Как видно из результатов на сопроцессорах MIC время счета шага в ~30 раз дороже последующих, на универсальном процессоре первый шаг дороже всего на 20 %. Времена последующих шагов близки между собой. По этой причине при вычислении значений весовой функции производительности процессов время на расчет первого шага исключается. Из результатов запуска на 5 узлах можно сделать вывод о достаточно большой «шумности» для универсальных процессоров – разница в скорости счета составляет 12 %. В связи с этим учет шумности может оказывать существенное влияние на общую скорость счета.

Таблица 4

Время на расчет шагов малой задачи и получаемые значения весовой функции производительности для одного узла ВС «Каскад»

№ шага	Host	MIC1	MIC2	Различие в производительности MIC, %	Соотношение между HOST и MIC, раз
1	0,01645	1,48681	1,61394	8,1 %	94,25
С 2 по 5 (среднее)	0,01205	0,05564	0,05527	5,5 %	4,60
Знач. весовой функции	20,75125	4,50825		0 %	4,60

Таблица 5

Время на расчет шагов малой задачи и получаемые значения весовой функции производительности для пяти узлов ВС «Каскад»

№ шага	**Тип**	**Сред.**	**Макс.**	**Миним.**	**Различие, %**	**Соотнош., раз**
1	Host	0,016694	0,01898	0,01505	26,1 %	95,3
	MIC	1,590478	1,66674	1,50255	10,9 %	
С 2 по 5 (среднее)	Host	0,014433	0,01508	0,01344	12,2 %	3,8
	MIC	0,054911	0,05561	0,05443	2,2 %	
Знач. весовой функции, Host		17,33651	18,59919	16,57389	12,2 %	3,8
Знач. весовой функции, MIC		4,553019	4,59323	4,49555	2,2 %	

Комбинированный режим расчета весовой функции производительности

В некоторых случаях соотношение производительности между вычислительными устройствами должно быть задано жестко, но при этом необходимо учесть «шумность» вычислительной системы. То есть значения производительности заданное пользователем используется в качестве среднего значения для данного типа вычислительных устройств и при этом изменяется в соответствии с шумностью. Например, пусть запущенно 4 процесса по 2 устройства разного типа, между устройствами задано соотношение 4 к 1. Разница в производительности из-за «шума» для устройств первого типа составила 10 %. В случае если шумность не учитывается, то значение весовой функции будет 0,4 : 0,4 : 0,1 : 0,1. Если же шумность учесть, то значение весовой функции составит 0,421 : 0,379 : 0,1 : 0,1.

Алгоритм расчета весовой функции производительности для комбинированного режима заключается в независимом расчете весовой функции для каждого типа вычислительных устройств и их перенормировки на значения, заданные пользователем.

Доработки счетных алгоритмов

Расчет значений весовой функции производительности производится «на лету» на этапе инициализации. Это потребовало доработки ряда программ и алгоритмов методики.

Во-первых, введен режим дублирования вычислений. В этом режиме процессы дублируют все данные задачи в полном объеме и производят ее расчет без какого-либо взаимодействия между собой. Во-вторых, расчет весовой функции производительности должен быть скрыт от пользователя, для этого в счетные блоки введен «тихий» режим работы при котором отсутствуют какие-либо выдачи на экран и работа с файловой системой (на этапе расчета весовой функции производительности внешние файлы еще не открыты). Исключение составляет чтение разреза с «малой задачей», если она используется для автоматического определения значений весовой функции производительности. В-третьих, отключены различные сервисные операции. В-четвертых, отключено определение весов областей и ячеек для декомпозиции, так как их расчет на данном этапе не имеет смысла.

Результаты тестирования многообластной задачи

Для тестирования алгоритмов была использована одна из многообластных задач, рассчитываемых по методике ТИМ-3D, состоящая из 5 математических областей с общим количеством точек около миллиона. Сетка неструктурированная преимущественно шестигранная, распределение точек и вес областей представлен в таблица 6.

Таблица 6

Характеристики многообластного расчета

№ области	Ячеек	Граней	Узлов	Вес области
1	42 856	133 366	47 748	386,159
2	726 544	2204 992	752 176	7609,565
3	3 388	11 990	5 283	60,881
4	141 193	436 598	154 392	1410,758
5	132 505	405 568	140 688	1277,344
Всего	1046 486	3192 514	1100 287	10744,707

Для полуавтоматического определения производительности процессов использовалось соотношение между Host и MIC – 6 к 1. Результаты замеров в различных режимах определения весовой функции производительности представлены в таблица 7. Как видно из представленных результатов, автоматический расчет значений весовой функции производительности процессов позволяет достичь производительности не хуже, чем при использовании полуавтоматического режима. Наилучшая производительность достигается при использовании внешнего разреза, так как по набору моделируемых процессов он наиболее близок к основной задаче. В этом случае производительность выше на 16 %. Учет шумности вычислительной системы, как правило, также позволяет повысить производительность на 4 – 5 %.

Таблица 7

Результаты замеров скорости счета с различным способом определения значений весовой функции производительности процессов

№	Режим счета	Среднее время счета шага	Ускорение, %
1	Полуавтоматическое с соотношением 1 к 6	3,45	–
2	Автоматическое (внешний разрез) с усреднением по типам	2,884	16,4 %
3	Автоматическое (внешний разрез) без усреднения по типам процессоров	2,756	20,1 %
4	Автоматическое (шестигранная сетка 100 тыс.) с усреднением по типам процессоров	3,380	2,0 %
5	Автоматическое (шестигранная сетка 100 тыс.) без усреднения по типам процессоров	3,200	7,2 %
6	Автоматическое (сетки из призм 10 тыс.) с усреднением по типам по типам процессоров	3,310	4,1 %
7	Автоматическое (сетка из призм 10 тыс.) без усреднения по типам процессоров	3,167	8,2 %
8	Автоматическое (сетка из тел Вороного 10 тыс.) с усреднением по типам процессоров	3,458	– 0,2 %
9	Автоматическое (сетка из тел Вороного 10 тыс.) без усреднения по типам процессоров	3,459	– 0,3 %
10	Комбинированный с соотношением 1 к 6 (шестигранная сетка 100 тыс.)	3,225	6,5 %
11	Комбинированный с соотношением 1 к 6 (сетка из призм 10 тыс.)	3,483	– 1,0 %
12	Комбинированный с соотношением 1 к 6 (сетка из тел Вороного 10 тыс.)	3,401	1,4 %

Заключение

В докладе рассмотрены алгоритмы выполнения декомпозиции для гибридных и гетерогенных вычислительных систем, реализованные в методике ТИМ-3D. Для учета разной производительности вычислительных устройств вводится специальная весовая функция производительности процессов, значения которой используются для определения размеров параобластей на этапе декомпозиции. В докладе рассматривается алгоритм полуавтоматического, автоматического и комбинированного определения значений данной весовой функции. Полуавтоматический режим основан на задании значений по типам процессов. Автоматический основан на расчете небольшой задачи в режиме дублирования без обменов всеми процессами задачи. Комбинированный режим сочетает эти два для учета шумности ВС.

Тестовые расчеты проведены на ВС «Каскад». Использование весовой функции производительности процессов позволило достичь ускорения в 1,4 для гибридного режима счета на одной из тестовых задач газовой динамики. Автоматическое определение значений функции производительности процессов позволяет, как правило, повысить скорость счета на величину до 20 %.

Литература

1. Соколов С. С., Панов А. И., Воропинов А. А., и др. Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов. – 2005. – Вып. 3. – С. 37 – 52.

2. Воропинов А. А., Соколов С. С. Метод трехуровневого распараллеливания методики ТИМ-2D // Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов. – 2013. – Вып. 4. – С. 70 – 77.

3. ООО «Центр компетенций и обучения». [Электронный ресурс]. Режим доступа: http://compcenter.org.

4. ParMETIS – Parallel Graph Partitioning and Fill-reducing Matrix Ordering [Electronic resource]. Mode of access: http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview.

5. Pellegrini F. SCOTCH: Static Mapping, Graph, Mesh and Hypergraph Partitioning, and Parallel and Sequential Sparse Matrix Ordering Package [Electronic resource]. – Mode of access: http://www.labri.fr/perso/pelegrin/scotch/.

6. Соколов С. С., Новиков И. Г., Воропинов А. А., Половникова Т. Н. Методы балансировки вычислительной нагрузки в методике ТИМ // XI Международная конференция «Параллельные вычислительные технологии ПаВТ-2017». – Казань, 3 – 7 апреля 2017. Короткие статьи и описание плакатов. Челябинск: Издательский центр ЮУрГУ, 2017. – С. 536.

7. A Polynomial Time Approximation Scheme for Minimum Makespan [Electronic resource]. – Mode of access: http://www.cs.ust.hk/mjg_lib/Classes/COMP572_Fall07/Notes/Min_Makespan.pdf

AUTOMATIC EVALUATION OF PERFORMANCE DURING THE DECOMPOSITION FOR HETEROGENEOUS AND HYBRID COMPUTING SYSTEMS IN THE TIM-3D CODE

A. A. Voropinov, T. N. Polovnikova, F. O. Golomidov, A. K. Shmeleva

Russian Federal Nuclear Center –
All-Russian Research Institute of Experimental Physics, Sarov

The paper considers the decomposition issues for hybrid and heterogeneous computing systems using the TIM-3D code. A special weighting function is introduced to account for the difference in the performance of computing devices, its values are used to find the para-domain sizes in the decomposition stage. The algorithm of automatically finding the weighting function values based on solving a small problem in the duplication mode without communications between all processes of the problem is desacribed. The quantity inverse to the runtime of several steps represents the values of the process performance weighting function. Besides, the algorithm allows accounting for the «noisiness» of computing system and revealing «dirty» nodes.

Key words: decomposition, heterogeneous computing systems, performance weighting function, the TIM-3D code.

УДК 517.9+533.7

НЕКОТОРЫЕ НЕРЕШЕННЫЕ ПРОБЛЕМЫ ЗАДАЧ ФИЗИЧЕСКОЙ КИНЕТИКИ

В. А. Галкин

Сургутский государственный университет, Сургут

Рассмотрен класс стационарных задач с источником частиц для уравнений физической кинетики с операторами стокновений больцмановского типа. Предложен алгоритм построения диссипативных решений, позволяющий отыскать точные решения нестационарного уравнения Смолуховского для пространственно однородных систем с мультипликативным ядром слияния частиц. Обоснован алгоритм построения таких решений. Важной спецификой выделенного класса решений является то, что на них оператор столкновений является разрывным в норме соотношения сохранения. Сформулирована проблема существования таких решений в случае оператора столкновений Больцмана кинетической теории газов, что позволило бы дополнить класс точных решений А. В. Бобылева диссипативными решениями.

Ключевые слова: физическая кинетика, операторы столкновений Больцмана и Смолуховского, разрывные операторы, законы сохранения, точные решения, диссипативные решения.

Математические модели физических систем, состоящих из статистически большого количества частиц (разреженные газы, дисперсные системы, плазма), а также модели механики сплошной среды основываются на фундаментальных соотношениях баланса, носящих общее название – *законы сохранения*. Значительное количество современных исследований по теории законов сохранения связано с вопросами корректности задач для систем нелинейных дифференциальных и интегродифференциальных уравнений

$$\frac{\partial f^{(\omega)}(x,t)}{\partial t}+\sum_{j=1}^{n}\frac{\partial F_j^{(\omega)}(f,x,t)}{\partial x_j}=S^{(\omega)}(f,x,t), \tag{1}$$

$$x\in\mathbb{R}_n,\quad t>0,\quad \omega\in\Omega,$$

где $f=\{f^{(\omega)}\}$ – неизвестная вектор-функция, вид потоков F_j и оператора столкновений $S(f)$ считаются заданными характером моделируемого физического процесса, $x\in\mathbb{R}_n$ – пространственные координаты, t – время, Ω – параметры, нумерующие уравнения.

Приложения этих уравнений широко известны, в частности, в связи с газодинамикой и гидродинамикой, физической кинетикой. Система законов сохранения (1) дополняется начальными данными

$$f\,|_{t=0}=f_0,\quad x\in\mathbb{R}_n,\quad \omega\in\Omega\,, \tag{2}$$