

УДК 519.63

М.С. Вдовенко, Г.А. Доррер

ИССЛЕДОВАНИЕ ДВУХ РАЗЛИЧНЫХ РЕАЛИЗАЦИЙ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ РАСЧЕТА РАСПРОСТРАНЕНИЯ КРОМКИ ЛЕСНОГО ПОЖАРА

Рассмотрены технологические аспекты разработки масштабируемых параллельных программ для кластерных вычислительных систем с использованием библиотеки MPI на примере задачи моделирования распространения лесных пожаров на кластерной вычислительной системе. Получены расчетные значения ускорений, позволяющие оценить масштабируемость алгоритма и его программной реализации. Приведены значения ускорений, полученные в ходе вычислительного эксперимента, которые хорошо согласуются с теоретическими оценками.

Ключевые слова: моделирование распространения лесных пожаров, параллельный алгоритм, ускорение вычислений.

Разработка программных комплексов для проведения крупномасштабных вычислительных экспериментов на параллельных вычислительных системах для решения задач, связанных с моделированием пожарной ситуации в лесах, представляет собой сложную в теоретическом и практическом плане задачу. Разработка параллельных программ практического уровня сложности для задач математической экологии представляет собою многоэтапный технологический процесс, включающий в себя анализ задачи, выбор модели программы, декомпозиция задачи на параллельные процессы, анализ производительности и организации вычислительного эксперимента.

Анализ задачи и выявление ее потенциального параллелизма

Постановка задачи и метод решения

Данная работа основана на полуэмпирической модели процесса распространения лесного пожара [1]. Рассмотрим алгоритм построения горящей кромки, основанный на численном решении уравнений, описывающих распространение процесса горения, по нескольким слоям лесного горючего. Введя в каждом из слоев прямоугольную сетку с шагами по координатам x и y соответственно Δx и Δy , введем сеточные области $D_{\Delta 0i}$, $D_{\Delta 1i}$, $D_{\Delta 2i}$, где $D_{\Delta 0i}$ – область, где горючий материал находится в исходном состоянии, $D_{\Delta 1i}$ – область, где происходит горение, $D_{\Delta 2i}$ – область, где горение невозможно, $i = 1, 2$. Алгоритм состоит в последовательном определении данных областей. Система рассматривается в дискретные моменты $t = 0, 1, 2, \dots$ с шагом Δt .

Уравнение нагрева горючего в области $D_{\Delta 0i}$ имеет вид

$$H_e(i, j, t+1) = H_e(i, j, t) + \Delta t \Delta x \Delta y \sum_{s=1}^2 \sum_{(m,n) \in D_{\Delta s}} \Phi(m, n, t) \times \xi_{sl}(m-i, n-j) + \\ + \Phi_{el}(i, j, t) \Delta t - \Delta t k_l(i, j) \times [H_l(i, j, t) - H_{0l}(i, j)],$$

где $H_e(i, j, t)$ – энтальпия горючего, $\Phi(m, n, t)$ – интенсивность тепловыделения в горящей области, $\Phi_{el}(i, j, t)$ – интенсивность внешних тепловых источников, $\xi_{sl}(m-i, n-j)$ – функция влияния локального пламени в горящей области на горючее в области $D_{\Delta 0i}$, $k_l(i, j)$ – коэффициент тепловых потерь. Система рассматривается при начальных условиях $H_l(i, j, 0) = H_{0l}(i, j)$, $(i, j) \in D_{\Delta 0l}$, $l = 1, 2$.

Условие воспламенения горючего в l -м слое при $t = t^*$: $H_l(i, j, t^*) \geq H_l^*(i, j)$, при этом узел (i, j) исключается из $D_{\Delta 0l}(t^*)$ и присоединяется к $D_{\Delta 1l}(t^*)$.

Уравнение расхода горючего:

$$\omega_l(i, j, t+1) = \begin{cases} \omega_l(i, j, t) - r_l \Delta t & \text{при } \omega_l(i, j, t) > 0, \\ \omega_l(i, j, t) & \text{при } \omega_l(i, j, t) \leq 0 \end{cases}$$

с начальным условием $\omega_l(i, j, t^*) = \omega_{0l}(i, j)$, $(i, j) \in D_{\Delta 1l}$, $l = 1, 2$.

Уравнение тепловыделения:

$$\Phi_l(i, j, t) = h_l(i, j)[\omega_l(i, j, t) - \omega_l(i, j, t - \Delta t)], \quad (i, j) \in D_{\Delta 1l}, \quad l = 1, 2,$$

где $h_l(i, j)$ – теплотворная способность горючего.

Условие погасания при $t = t_*$: $\omega_l(i, j, t_*) = 0$, $(i, j) \in D_{\Delta 1l}$, при этом точка (i, j) исключается из $D_{\Delta 1l}(t_*)$ и присоединяется к $D_{\Delta 2l}(t_*)$.

Функции влияния $\xi_j(x - x_1, y - y_1, z - z_1)$ задаются формулами с учетом дискретизации по пространству и приводятся в работе [1].

П а р а л л е л и з м

Для распараллеливания процесса вычислений предлагается схема, вытекающая из физического содержания данной задачи. Расчет энтальпии для точки ij на $(t+1)$ -м временном шаге происходит с использованием некоторого количества точек на t -м шаге. Численный расчет ведется итеративно: по имеющимся значениям n -го временного шага выстраиваем $(t+1)$ -й и т.д.

Следовательно, исходную задачу можно разбить на несколько подзадач для областей, пересекающихся только по границе разбиений, независимых друг от друга на каждом расчетном шаге. Исходная область не включает взаимно перекрывающиеся подобласти, поэтому пересчет значений на границах между данными областями предполагает согласно алгоритму суммирование при обмене вычисленными значениями для граничных элементов. Для перехода к следующей итерации необходимо согласование границ, так как первая область должна передать второй ее левую границу для следующего шага по времени, в свою очередь вторая область должна передать первой ее правую границу и т.д.

Определим теперь потенциальный параллелизм алгоритма количественно. Общий объем вычислений V_{calc} в алгоритме определяется соотношением $V_{\text{calc}} = s \cdot k \cdot N^2$, где N^2 – количество расчетных точек в двухмерной области моделирования, k – количество операций, выполняемых в одной расчетной точке, s – количество шагов по времени. Для простоты полагаем, что все операции, в том

числе и обмены, имеют одинаковое время исполнения, равное t . Рассмотрим наихудший случай, когда все операции в одной точке выполняются последовательно. Следовательно, на каждом шаге итерации алгоритм требует для своей реализации k этапов. Тогда средняя степень параллелизма r будет равна $r = s \cdot k \cdot N^2 / (s \cdot k) = N^2$.

Ускорение и масштабируемость

Потенциальное ускорение алгоритма можно оценить исходя из следующего соотношения: $S_p = T_1/T_p$, где T_1 – время вычислений на одном процессоре, T_p – время вычислений на p процессорах. При этом время перевычислений будет определяться следующим образом: $T_c = 2smN$, где m – количество значений для одной расчетной точки, передаваемых соседнему процессу.

Рассмотрим случай с использованием неблокирующих передач. В общем случае обмены с использованием таких передач не зависят от количества используемых процессоров и потребуют $T_c = \text{const}$ единиц времени. Тогда ускорение с использованием неблокирующих передач будет определяться следующим соотношением:

$$S'_p \leq \begin{cases} \frac{1}{1/p + T_c/T_1}, & 1 < p \leq r, \\ \frac{1}{1/r + T_c/T_1}, & p > r. \end{cases}$$

Учитывая, что $T_c = 2smNt$, $T_1 = s \cdot k \cdot N^2 \cdot t$, $r = N^2$, получим окончательно

$$S'_p \leq \begin{cases} \frac{1}{1/p + 2m/kN}, & 1 < p \leq N^2, \\ \frac{1}{1/N^2 + 2m/kN}, & p > N^2. \end{cases}$$

В табл. 1 приведена зависимость ускорения S'_p от количества используемых процессоров при следующих значениях параметров: $N = 500$; $k = 43$; $m = 15$; $r = 250000$.

Таблица 1

Значения ускорения в зависимости от количества доступных процессоров

p	1	2	4	8
S'_p	1,000000	1,994434	3,977798	7,911684

Алгоритм и его программная реализация являются масштабируемыми, если ускорение и производительность зависят линейно от количества используемых процессоров [3]: $S_p = O(p)$, где S_p – ускорение, p – количество процессоров. На практике алгоритмы, для которых $S_p = O(p/(\ln p))$, также считаются масштабируемыми.

Полученные результаты показывают, что алгоритм обладает значительным объемом потенциального параллелизма и хорошей, с точки зрения распараллеливания, структурой.

Выбор схемы распараллеливания вычислений и программирование задачи

Возможны две параллельных реализации данной задачи:

1. С одним управляющим узлом, который координирует вычислительный процесс на остальных процессорных узлах.

Управляющий процессор (процессор 0) выполняет:

- считывание данных из файлов;
- разбиение последовательности данных на части и пересылку на соответствующие процессоры блоков последовательности;
- выполнение параллельной обработки исходных данных (эта часть действий является общей для всех процессоров и детально описана в алгоритме работы функциональных процессоров);
- сбор результатов вычислений от всех процессоров;
- запись полученных результатов в выходной файл;
- освобождение занимаемой памяти.

Функциональные процессоры выполняют:

- прием от процессора 0 соответствующих элементов;
- внутреннюю обработку данных, включающую решение уравнений теплопроводности на данном временном шаге, вычисление значения энтальпии и определение воспламенившихся и погасших точек;
- реализацию упаковки информации о воспламенившихся точках;
- обмен данными между соседними процессорами (передача упакованной информации только о воспламенившихся точках);
- распаковку и размещение данных в соответствующую область памяти;
- передача результата на управляющий процессор.

2. Технология «взаимодействующие равные», при которой все процессорные узлы выполняют одинаковый набор действий.

Все процессорные узлы выполняют:

- считывание данных из соответствующих файлов;
- выполнение параллельной обработки исходных данных (решение уравнений теплопроводности на данном временном шаге);
- обмен данными между соседними процессорами (передача информации о величине энтальпии для всех граничных данных);
- вычисление значения энтальпий для каждой граничной точки согласно алгоритму суммирования, определение воспламенившихся и погасших точек;
- запись полученных результатов в выходной файл;
- освобождение занимаемой памяти.

Для данной реализации необходима дополнительная предпроцессорная обработка, заключающаяся в разбиении области данных на отдельные файлы для каждого процессорного узла.

Рассмотрим подробнее второй случай. Реализация параллельной программы осуществилась на языке программирования Си с применением функций библиотеки передачи сообщений MPI [2 – 4]. Алгоритм параллельной программы содержит следующие шаги: исходные данные считываются процессами параллельной программы, а затем на каждом узле параллельная программа обрабатывает свою часть данных. Вычисления проводятся в два этапа.

1. Вычисляется сфера влияния локального пламени в данных условиях:

- a) на каждом узле параллельная программа обрабатывает свою часть данных;

б) по завершению расчетов осуществляется обмен вычисленными значениями на границах.

2. Расчет распространения горящей кромки.

При этом на каждом временном шаге каждый процессорный узел вычисляет для своей части пространства вклад в нагревание горячего каждой горящей точкой. Также необходимо на каждом временном шаге передавать значения энтальпий для левых и правых границ. В конце временного шага по вычисленным значениям энтальпий определяется: воспламенится ли соответствующая негорящая точка.

По окончании вычислений результаты со всех процессорных узлов сохраняются в файле результатов.

Структура обменов является однородной, за исключением первого и последнего процессов.

Проведение вычислительного эксперимента и анализ результатов

Для численного исследования ускорения параллельного алгоритма рассмотрим следующую модельную задачу. Пусть Ω – «квадрат» на сфере: $\Omega = [1, 6\pi] \times [1, 6\pi]$. Возвышения поверхности задаются косинусоидой $= 1 + \cos(x/150)$. В расчетной области построена равномерная квадратная сетка 500×500 , в вычислительных экспериментах было сделано 100 шагов по времени.

Вычисления выполнялись на 48-процессорном кластере ИВМ СО РАН в г. Красноярске. Кластер МВС-1000/ИВМ (собственная сборка ИВМ СО РАН) содержит 24 вычислительных узла AMD Athlon64/3500+/1Гб (однопроцессорные, одноподъёмные); 12 вычислительных узлов AMD Athlon64 X2 Dual Core/4800+/2Гб (однопроцессорные, двухъёмные); управляющий узел, сервер доступа и файловый сервер Athlon64/3500+/1Gb с общей дисковой памятью 400 Гб. Управляющая сеть кластера FastEthernet (100 Мбит/сек), сеть передачи данных GigaEthernet (1000 Мбит/сек). Необходимо отметить, что кластер является гетерогенным, поскольку некоторые вычислительные узлы двухъёмные. Временные характеристики выполнения программы осреднялись по результатам нескольких десятков расчетов.

В табл. 2 приведены реальные значения ускорений, полученные в ходе вычислительного эксперимента.

Таблица 2

Значения ускорения показанного на кластере ИВМ СО РАН

Параметры	Количество процессоров			
	1	2	4	8
Область	500×500	500×500	500×500	500×500
Количество шагов	100	100	100	100
<i>Схема с управляющим узлом</i>				
Время выполнения основного цикла, T[c]	827,861564	777,038914	610,502950	902,534109
Ускорение, S	1,000000	1,065406	1,356032	0,917263
<i>Схема «взаимодействующие равные»</i>				
Время выполнения основного цикла, T[c]	700,868975	349,848900	197,077725	65,259700
Ускорение, S	1,000000	2,003348	3,556307	10,739690

Полученные результаты немного расходятся с теоретическими оценками ускорений, что объясняется затратами времени на пересылку данных. Схема с управляющим узлом является неэффективной при использовании 8 процессоров, так как время обмена начинает значительно превышать время вычислений. Как следует из табл. 2, для схемы «взаимодействующие равные», при использовании двух процессорных узлов, все данные находятся в оперативной памяти и нет необходимости считывания/записи с жесткого диска, что ведет к значительному ускорению по сравнению с использованием одного процессора.

Для схемы «взаимодействующие равные» полученная программа является эффективной и масштабируемой. Следует также отметить, что при фиксированном количестве узлов сетки, дальнейшее увеличение числа процессоров несколько снижает эффективность распараллеливания из-за возникающего преобладания времени обменов над временем локального счета.

В рамках настоящей работы на примере задачи моделирования распространения лесных пожаров на кластерной вычислительной системе, были рассмотрены технологические аспекты разработки масштабируемых параллельных вычислений для кластерных вычислительных систем с использованием библиотеки MPI.

Получены расчетные значения ускорений, позволяющие оценить масштабируемость алгоритма и его программной реализации. Эти результаты показывают, что алгоритм обладает значительным объемом потенциального параллелизма и хорошей, с точки зрения распараллеливания структурой, что позволяет надеяться на получение ускорений близких к линейным в зависимости от количества используемых процессоров для кластерных вычислительных систем.

Приведены значения ускорений, полученные в ходе вычислительного эксперимента, которые хорошо согласуются с теоретическими оценками.

ЛИТЕРАТУРА

1. Доррер Г.А. Динамика лесных пожаров. Новосибирск: Изд-во СО РАН, 2008. 404 с.
2. Корнеев В.Д. Параллельное программирование в MPI. 2-е изд. Новосибирск: Изд-во ИВМиМГ СО РАН, 2002. 215 с.
3. MacDonald N., Minty E., Harding T., Brown S. Writing Message-Passing Parallel Programs with MPI. [Электронный ресурс]. Режим доступа: <http://www.hpc.nw.ru/PS/mpi-course.zip>
4. McBryan O.A. An overview of message passing environments // Parallel Computing. 1994. V. 20. P. 417 – 441.

Статья принята в печать 18.07.2008 г.