

ЭФФЕКТИВНОСТЬ MIMD-РЕШАТЕЛЕЙ ПРОБЛЕМНО ОРИЕНТИРОВАННОЙ ПАРАЛЛЕЛЬНОЙ МОДЕЛИРУЮЩЕЙ СРЕДЫ

Рассмотрена оценка трудоемкости и характеристик распараллеливания MIMD-решателей уравнений на базе MPI- и OpenMP-стандартов для параллельной моделирующей среды. Рассмотрена задача построения и апробации MIMD-решателей уравнений.

параллельные алгоритмы, MPI, OpenMP, моделирующая среда

Введение

Актуальной проблемой технологий параллельного моделирования является выработка рекомендаций по рациональному использованию ресурсов параллельной моделирующей среды (ПМС).

В статье рассматриваются оценка трудоемкости и характеристик распараллеливания MIMD-решателей уравнений на базе MPI- и OpenMP-стандартов для ПМС [1], ориентированных на особенности топологии, аэрогазодинамических процессов сетевых динамических объектов (СДО) с сосредоточенными параметрами, построение и апробация MIMD-решателей уравнений, которые являются ключевой частью проблемно ориентированной ПМС.

Оценка трудоемкости алгоритма решателя уравнений

Решатель уравнений – аппаратно-програмное средство, которое реализует некоторый численный метод [2] решения сформированных уравнений. Рассмотрим реализацию решателя уравнений аэродинамики шахтных вентиляционных сетей (ШВС). Система уравнений аэродинамики ШВС представляется следующим образом:

$$\begin{cases} X = -W \cdot Y \\ \frac{dY}{dt} = TP \cdot H - RU \cdot Z \end{cases}$$

где

$$TP = (S_y \cdot K_y - S_x \cdot K_x \cdot W)^{-1} \cdot S$$

$$RU = (S_y \cdot K_y - S_x \cdot K_x \cdot W)^{-1} \cdot S \cdot R$$

$$W = A_x^{-1} \cdot A_y$$

На подготовительном этапе формируется дерево и антидерево графа ШВС с подвекторами потоков расходов воздуха в ветвях дерева

$X = (X_1, X_2, \dots, X_{n-1})^T$ и антидерева $Y = (Y_1, Y_2, \dots, Y_{m-n+1})^T$. Матрицы инциденций A , независимых контуров S , диагональные матрицы физических параметров ШВС R и K , векторы H и Z упорядочены относительно векторов X, Y .

Алгоритм функционирования решателя уравнений представлен на рис. 1.

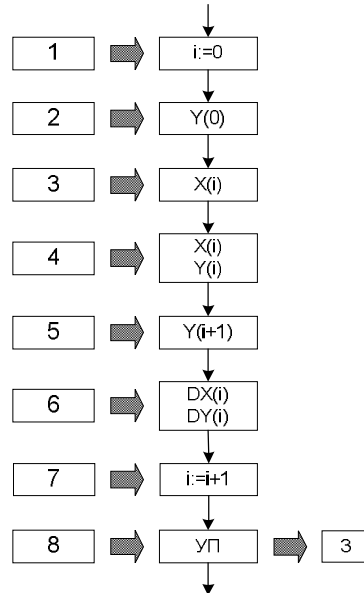


Рисунок 1 – Алгоритм функционирования решателя уравнений

Вычисления расходов воздуха в ветвях дерева графа назовем X -компонентой. Для определения расходов воздуха в ветвях дерева, X -компонента решателя уравнений умножает топологическую коммутирующую матрицу W на вектор значений расходов в ветвях антидерева. Вычисление расходов воздуха в ветвях антидерева назовем Y -компонентой, которая определяет вектор производных $\frac{dY}{dt}$ и с помощью выбранного численного метода находит элементы вектора Y . Таким образом, алгоритм функционирования решателя уравнений состоит из двух частей: вычисление X - и Y -компонент.

Оценим трудоемкость вычисления вектора X , каждый элемент которого вычисляется по следующей формуле:

$$X_k = -\sum_{p=1}^{\gamma} W_{kp} \cdot Y_{p1}, k = 1, 2, \dots, n-1$$

Временная сложность этой части алгоритма будет:

$$T_1^X = (n-1) \cdot (\gamma \cdot t_{ym} + (\gamma-1) \cdot t_{cl}).$$

Трудоемкость вычисления компоненты Y зависит от формулы правой части дифференциального уравнения - T_f^Y и от выбранного численного метода. Временная сложность вычисления значения производной функции дифференциального уравнения будет:

$$T_f^Y = m \cdot t_{ym} + \gamma \cdot (m \cdot t_{ym} + (m-1) \cdot t_{cl}) + \gamma \cdot t_{cl}$$

Временная сложность вычисления производной в векторно-матричном виде с помощью численного метода Эйлера будет $\gamma \cdot T_s$, где T_s - временная сложность метода Эйлера $T_s = t_{ym} + t_{cl}$.

Отображение параллельного алгоритма решателя уравнений ну целевую MIMD-систему (ЦПВС)

Процесс построения параллельных РУ можно разделить на два этапа: построение виртуальных параллельных моделей СДО и девиртуализация последних [3]. Под виртуальной будем понимать модель, в которой выделены параллельно функционирующие, относительно независимые процессы, которые совместно могут обеспечить решение задачи. Девиртуализацией назовем процесс отображения виртуальных моделей на целевой параллельный вычислительный ресурс, ту вычислительную систему, которая есть в распоряжении пользователя. Девиртуализация параллельных моделей состоит из следующих этапов: априорная оценка виртуальной модели с учетом балансировки загрузки и обмена данными, объединение виртуальных процессов с целью выравнивания загрузки и уменьшения количества операций обмена данными, анализ характеристик ЦПВС, программная реализация виртуальной модели в условиях выбранной ЦПВС.

Анализ последовательного решателя уравнений аэродинамики ШВС показал, что используется одна и та же последовательность инструкций, но для разных данных. Следовательно, в данном случае распараллеливание целесообразно выполнять по SPMD-принципу.

Так как минимальным моделируемым элементом является ветвь, то рассмотрим виртуальную модель, в которой есть m -виртуальных процессов, каждый из которых вычисляет один элемент X - или Y - компоненты, выделим один процесс на одну ветвь графа. Следовательно, $n-1$ виртуальных процессов вычисляют элементы компоненты X , а остальные $m-n+1$ процессов - элементы Y - компоненты (1-я виртуальная модель).

Выполним девиртуализацию – процесс отображения виртуальной модели на целевую вычислительную машину, т.е. отображение виртуальных процессов на определенное количество реальных процессоров. Тогда для 1-й виртуальной модели процессоры будут разделены на X - и Y -группы. Количество вычисляемых элементов компоненты X на одном процессоре будет равно отношению количества ветвей дерева к количеству процессоров в группе. Соответственно количество вычисляемых элементов компоненты Y на одном процессоре будет равно отношению количества ветвей антидерева к количеству процессоров в группе (рис. 2).

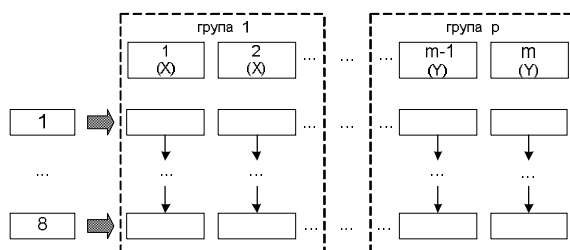


Рисунок 2 – Девиртуализация модели РУ

Рассмотрим виртуальную модель, в которой каждый виртуальный процесс отвечает за вычисление как элемента X - так и элемента Y - компонент. Для этого случая количество виртуальных процессов составляет $m - n + 1$ по количеству ветвей антидерева (2-я виртуальная модель).

Для этой модели на каждом процессоре реализуются виртуальные процессы X - и Y -компоненты. Количество вычисляемых элементов компонент X , Y на процессоре будет равно отношению количества ветвей дерева к общему количеству процессоров (рис. 3) соответственно.

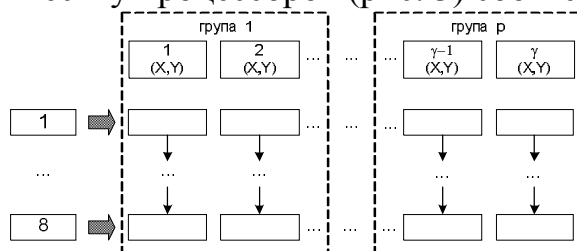


Рисунок 3 – Девиртуализация модели РУ

Таким образом, можно выделить два подхода к распараллеливанию РУ. Подход 1: компоненты X и Y не объединены, предусмотрено m виртуальных процессов, каждый из которых ведет вычисление либо компоненты X , либо Y . После нахождения текущих значений компоненты X выполняется вычисление значений правых частей Y компоненты, в ходе которого происходит обмен информацией между процессами, дальше элементы Y -компоненты передаются виртуальным процессам компоненты X . Девиртуализация для подхода 1 заключается в объединении X -процессов в группы, что составят X -компоненту, а Y -процессы – Y -компоненту, с целью выровнять нагрузку процессоров.

Подход 2: компоненты X и Y объединены и предусмотрено γ виртуальных процессов, каждый из которых ведет вычисление компонент X и Y . После нахождения текущих значений компоненты X выполняется вычисление значений правых частей Y - компоненты, в ходе которого происходит обмен информацией между процессами, дальше виртуальные процессы обмениваются найденными элементами Y -компоненты. Девиртуализация для подхода 2 заключается в том, что виртуальные процессы полностью идентичны, в i -м виртуальном процессе исчисляются X_i, Y_i . В ситуации $\gamma < n - 1$ целесообразно привязывать количество процессов к величине γ , а те $X_{\gamma+1}, \dots, X_{n-1}$, что остались не распределенными,

распределить равномерно среди виртуальных процессов. При $\gamma > n-1$ можно допустить работу части Y - процессов без вычисления компонентов вектора X .

В соответствии с подходом 1, обозначим количество доступных процессоров на целевой параллельной вычислительной системе как s , тогда s_x - количество процессоров, на котором вычисляется X -компонента, а s_y - количество процессоров, на котором вычисляется Y -компонента, т.е. $s = s_x + s_y$. Зная количество процессоров X - и Y -компоненты, можно найти k_x - количество элементов X -компоненты и k_y - количество элементов Y -компоненты, вычисляемые на одном процессоре:

$$k_x = \begin{cases} \left\lfloor \frac{n-1}{s_x} \right\rfloor, ((n-1) \bmod s_x) = 0 \\ \left\lfloor \frac{n-1}{s_x} \right\rfloor + 1, ((n-1) \bmod s_x) \neq 0 \end{cases}$$

$$k_y = \begin{cases} \left\lfloor \frac{\gamma}{s_y} \right\rfloor, (\gamma \bmod s_y) = 0 \\ \left\lfloor \frac{\gamma}{s_y} \right\rfloor + 1, (\gamma \bmod s_y) \neq 0 \end{cases}$$

Определим количество элементов X -компоненты, вычисляемые на одном процессоре и количество элементов Y -компоненты для подход 2:

$$k_x = \begin{cases} \left\lfloor \frac{n-1}{s} \right\rfloor, ((n-1) \bmod s) = 0 \\ \left\lfloor \frac{n-1}{s} \right\rfloor + 1, ((n-1) \bmod s) \neq 0 \end{cases}$$

$$k_y = \begin{cases} \left\lfloor \frac{\gamma}{s} \right\rfloor, (\gamma \bmod s) = 0 \\ \left\lfloor \frac{\gamma}{s} \right\rfloor + 1, (\gamma \bmod s) \neq 0 \end{cases}$$

Рассмотрим время выполнения одной итерации цикла (определение компоненты X и Y) РУ. В общем случае время выполнения $T_s = T_s^X + T_s^Y$ складывается из времен вычисления элементов X - и Y -компонент.

Для подхода 1 время вычисления X компоненты будет

$$T_s^X = k_x \cdot (\gamma \cdot t_{ym} + (\gamma - 1) \cdot t_{cl})$$

Формула Y -компоненты для уравнений аэродинамики (метод Ейлера) будет следующей:

$$T_s^Y = k_x \cdot t_{ym} + \gamma \cdot (k \cdot t_{ym} + (k - 1) \cdot t_{cl}) + k_y \cdot t_{cl}$$

Для подхода 2 формула вычисления общего времени T_s и вычисления компоненты X совпадает с формулами подхода 1. Формула Y компоненты будет:

$$T_f^{Y_s} = t_{ум} \cdot (k_x + k_y) + \gamma \cdot ((k_x + k_y) \cdot t_{ум} + ((k_x + k_y) - 1) \cdot t_{сл}) + k_y \cdot t_{сл}$$

Для оценки целесообразности применения параллельных подходов используются такие характеристики, как ускорение и эффективность. Ускорение определяется как отношение времени выполнения последовательного алгоритма одним процессором к времени параллельной реализации при использовании заданного количества процессоров.

Эффективность определяется как отношение ускорения на заданном количестве процессов к количеству процессоров.

Предложенная модель оценки позволяет анализировать зависимости ускорения и эффективности параллельных подходов от размерности графа и характеристик ЦПВС. С помощью этой модели можно определять не только ускорение и эффективность распараллеливания, но и решать обратную задачу: определять количество процессоров, которое необходимо использовать для достижения требуемого ускорения вычислений и эффективности использования параллельных ресурсов.

Входные параметры для математической модели соответствуют реальной шахтной вентиляционной сети, у которой количество ветвей $m = 117$, узлов- $n = 61$.

На рис. 4 показана зависимость ускорения параллельного алгоритма решателя уравнений от количества процессоров и способа девиртуализации.

На рис. 5 показана зависимость эффективности параллельного алгоритма решателя уравнений от количества процессоров и способа девиртуализации.

Анализ ускорения и эффективности показал, что для графов с количеством ветвей более 50 эффективным является использование 2-го подхода, который подразумевает объединение вычислений элементов компоненты X и Y . При этом эффективность остается в области 0.8, что является хорошим показателем внутреннего параллелизма.

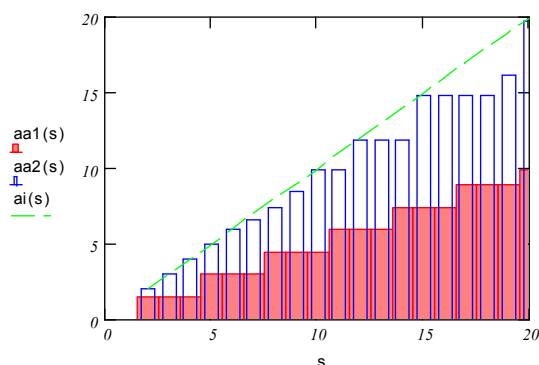


Рисунок 4 – Ускорение параллельного алгоритма решателя уравнений в зависимости от количества процессоров и способа девиртуализации

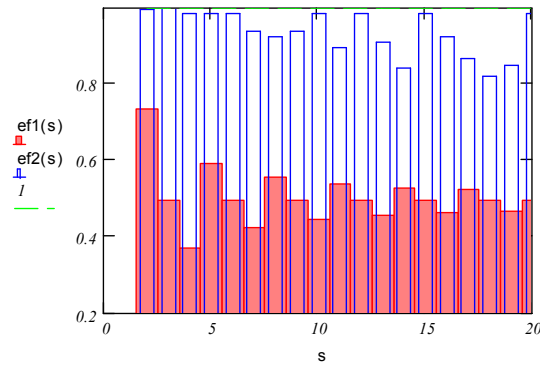


Рисунок 5 – Эффективность параллельного алгоритма решателя уравнений в зависимости от количества процессоров и способа девиртуализации

Экспериментальные исследования решателей уравнений, построенных на базе MPI- и OpenMP- стандартов

В результате моделирования были построены зависимости времени решения уравнений от количества процессоров с учетом времени обмена и определены ускорение (рис. 6) и эффективность (рис. 7) распараллеливания для подходов 1 и 2.

Предложенные подходы к распараллеливанию были реализованы на базе стандартов параллельного программирования MPI и OpenMP. В качестве целевой параллельной машины был взят NEC-SX8 с гибридной архитектурой, в каждом узле которой имеется 8 процессоров, работающие через общую память.

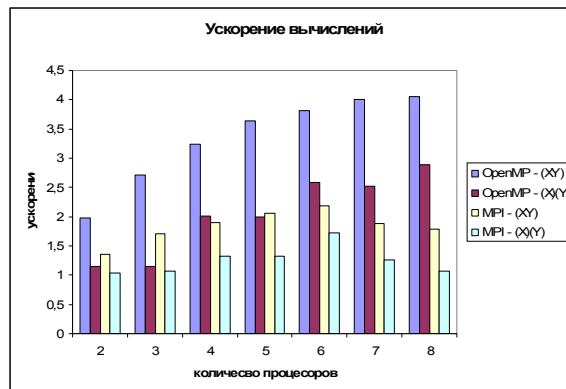


Рисунок 6 – Ускорение параллельного алгоритма решателя уравнений в зависимости от количества процессоров и способа девиртуализации реализованных на базе MPI и OpenMP стандартов

Тесты запускались на одном узле и количество процессоров варьировалось от 2 до 8. Экспериментальные исследования показали, что оба параллельных подхода более эффективны при использовании OpenMP стандарта. Эффективность 2-го подхода лучше 1-го подхода в реализациях на MPI и OpenMP и колеблется от 0.9 до 0.6, что является хорошим показателем. За счет времени обмена сообщениями реализация на MPI стандарте проигрывает OpenMP. Недостатком использования OpenMP

является невозможность запуска параллельных программ на более чем 8-ми процессорах. Можно сделать вывод, что для моделирования задач с размерностью графа $m > 500$ ветвями лучше использовать совместно оба стандарта для реализации гибридных параллельных моделей.

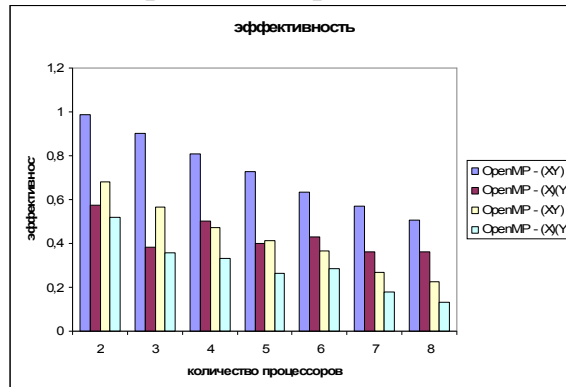


Рисунок 7 – Эффективность параллельного алгоритма решателя уравнений в зависимости от количества процессоров и способа девиртуализации реализованных на базе MPI и OpenMP стандартов

Выводы

Для построения MIMD-решателей уравнений сетевых динамических объектов целесообразно использовать гибридную модель параллельного программирования с использованием MPI- и OpenMP-стандартов в рамках одного параллельного симулятора. Предложенные формулы оценки загрузки процессов виртуальных параллельных моделей позволяют равномерно загружать MIMD-процессоры и эффективно использовать ресурсы распределенной параллельной моделирующей среды.

Список литературы

1. Святний В.А. Проблеми параллельного моделювання складних динамічних систем / В.А. Святний // Научные труды Донецкого национального технического университета. Серия: Информатика, кибернетика и вычислительная техника. – Вып. 6. - 1999. - С. 6-14.
2. Калиткин Н.Н. Численные методы / Н.Н. Калиткин. – М.: Наука, 1978. – 512 с.
3. Святний В.А. Методи розпаралелювання вирішувача рівнянь MIMD-моделі мережних динамічних об'єктів / В.А.Святний, О.М.Смагін, О.М. Солонін // Научные труды Донецкого национального технического университета. Серия: Информатика, кибернетика и вычислительная техника. – 2003. - Вып. 70. - С.20-29.

Надійшла до редакції 15.12.2010.

Рецензент: д-р техн.наук, проф. Святний В.А.

О.М. Смагін

Донецький національний технічний університет

Ефективність MIMD-вирішувачів проблемно орієнтованого параллельного моделюючого середовища. Розглянута оцінка трудомісткості та характеристик розпаралелювання MIMD-вирішувачів рівнянь на базі MPI- та OpenMP-стандартів для

паралельного моделюючого середовища. Розглянуто задачу побудови та апробації MIMD-вирішувачів рівнянь.

паралельні алгоритми, MPI, OpenMP, моделююче середовище

O.M.Smagin

Donetsk National Technical University

Efficiency of MIMD-Solvers of Problem Oriented Parallel Modeling Environment. The evaluation of complexity and characteristics of parallel MIMD-solvers of equations based on the MPI- and OpenMP- standards for parallel modeling environment are considered. The problem of construction and testing of MIMD-solvers of equations is also considered.

parallel algorithms, MPI, OpenMP, simulating environment