УДК 004

ЭФФЕКТИВНОСТЬ И МАСШТАБИРУЕМОСТЬ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ БЛОЧНОГО УМНОЖЕНИЯ ПЛОТНО ЗАПОЛНЕННЫХ МАТРИЦ

Лямина О. В., Назарова И.А.
Донецкий национальный технический университет
Кафедра прикладной математики и информатики
E-mail: liamina_olga@mail.ru

Аннотация

Лямина О.В., Назарова И.А. Эффективность и масштабируемость параллельных алгоритмов блочного умножения плотно заполненных матриц*. Рассмотрены параллельные алгоритмы умножения матриц семейства Кэннона. Проведен анализ зависимости времени выполнения и ускорения от числа процессоров и размера матриц. Оценена эффективность алгоритмов.*

Постановка задачи

Матричное умножение – одна из основных операций, которая выполняется при решении различных задач: решение системы линейных алгебраических уравнений, дифференциальных уравнений. Умножение матриц является трудоемким с операционной и коммутативной точки зрения. Поэтому эффективность решения этой задачи – важный фактор.

Для эффективного выполнения умножения матриц используется параллельные алгоритмы. Топология решетка и гиперкуб являются наиболее подходящими для реализации таких операций. Для такой топологии вычисление матричных арифметических операций можно свести к выполнению операций с блоками матриц.

Существует несколько таких алгоритмов. В данной работе рассматривается семейство алгоритмов Кэннона, которое основано на блочном разбиении матриц.

В алгоритме Кэннона две исходные матрицы A и B и матрица результат C разделяются на блоки. Семейства Кэннона изменяет отображения блоков двух из трех матриц, которые берут участие в вычислении произведения.

Пусть количество столбцов/строк матрицы n кратно числу узлов решетки p. Количество узлов решетки по вертикали/горизонтали равно q. Если представить матрицы в виде квадратных блоков размером $k=\dfrac{n}{q}$ элементов, то каждому узлу можно однозначно поставить в соответствие такой блок.

Алгоритм вычисления матричного произведения с сохранением отображения блоков матрицы-результата С.

Алгоритм включает в себя шаги:

1) блоки строк матрицы A сдвигаются циклично влево на i узлов по горизонтали, где i - индекс строки матрицы A.

2) блоки столбцов матрицы B сдвигаются циклично вверх на j узлов по вертикали, где j – индекс столбца матрицы A.

Алгоритм выполняется за q шагов, где q – размерность вычислительной решетки. Каждый шаг состоит из следующих действий:

а) на вычислительном узле решетки с индексами (i, j) производится умножение блоков

A_{ij} и B_{ij}.

b) циклическое смещение блоков матрицы A влево на 1 узел по горизонтали решетки.

c) циклическое смещение блоков матрицы B вверх на 1 узел по вертикали решетки.

Результат умножения матриц хранится в матрицы C, блоки которой не подлежат смещению.

Анализ эффективности

Время выполнения п.1) или п.2) согласно [1] можно рассчитать по формуле:

$$T_{alignAB} = (t_s + t_w \cdot k^2)(q+1) \tag{1}$$

где t_s - латентность, t_w - время передачи слова данных.

Время умножения матриц в одном блоке:

$$T_{AB} = (k^2 \cdot (2k-1) + k^2) \cdot \tau \tag{2}$$

Время циклического сдвига для п. c) и b):

$$T_{rollShift} = (t_s + t_w \cdot k^2)(q+1) \tag{3}$$

Суммарное время выполнения алгоритма:

$$T_{CannonC} = 2qT_{alignAB} + 2T_{rollShift} + pT_{AB} \tag{4}$$

$$T_{CannonC} = (2q+2)(t_s + t_w \cdot k^2)(q+1) + (k^2 \cdot (2k-1) + k^2) \cdot p \cdot \tau \tag{5}$$

Отсюда получаем ускорение параллельного алгоритма и эффективность использования параллельным алгоритмом процессоров при решении задачи:

$$S_{CannonC} = \frac{(k \cdot q)^2 (2k \cdot q - 1)}{(2q+2)(t_s + t_w \cdot k^2)(q+1) + (k^2 \cdot (2k-1) + k^2) \cdot p \cdot \tau} \tag{6}$$

$$E_{CannonC} = \frac{S_{CannonC}}{p} \tag{7}$$

Алгоритм вычисления матричного произведения с сохранением отображения блоков матрицы А.

Алгоритм включает в себя шаги:

1) блоки строк матрицы B сдвигаются циклично вправо на i узлов по горизонтали, где i - индекс строки матрицы B.

2) блоки столбцов матрицы B сдвигаются циклично вверх на j узлов по вертикали, где j – индекс столбца матрицы A.

3) блоки строк матрицы C сдвигаются циклично вправо на i узлов по горизонтали, где i - индекс строки матрицы C.

Алгоритм выполняется за q шагов, где q – размерность вычислительной решетки. Каждый шаг состоит из следующих действий:

a) на вычислительном узле решетки с индексами (i, j) производится умножение блоков A_{ij} и B_{ij}.

b) Циклическое смещение блоков матрицы C вправо на 1 узел по горизонтали решетки.

c) Циклическое смещение блоков матрицы B вверх на 1 узел по вертикали решетки.

Результат умножения матриц хранится в матрицы C, блоки которой подлежат смещению. Поэтому по завершению нужно выровнять матрицу до выходного отображения блоков.

Анализ эффективности

Время выполнения п.1), п. 2) или п.3) просчитывается по формуле (1). Время умножения матриц в одном блоке – формула (2). Время циклического сдвига для п. c) и b) – (3). После выполнения q шагов матрицы B и C необходимо выровнять до выходного отображения блоков на узле вычислительной решетке. Время выполнения рассчитывается по (1).

Суммарное время выполнения алгоритма:

$$T_{CannonA} = 3qT_{alignAB} + 2T_{rollShift} + pT_{AB}$$

$$T_{CannonA} = (3q+2)(t_s + t_w \cdot k^2)(q+1) + (k^2 \cdot (2k-1) + k^2) \cdot p \cdot \tau \quad (8)$$

Отсюда получаем ускорение параллельного алгоритма и эффективность использования параллельным алгоритмом процессоров при решении задачи:

$$S_{CannonA} = \frac{(k \cdot q)^2 (2k \cdot q - 1)}{(3q+2)(t_s + t_w \cdot k^2)(q+1) + (k^2 \cdot (2k-1) + k^2) \cdot p \cdot \tau} \quad (9)$$

$$E_{CannonA} = \frac{S_{CannonA}}{p} \quad (10)$$

Алгоритм вычисления матричного произведения с сохранением отображения блоков матрицы B.

Алгоритм включает в себя шаги:

1) блоки строк матрицы A сдвигаются циклично влево на i узлов по горизонтали, где i - индекс строки матрицы A.
2) блоки столбцов матрицы A сдвигаются циклично вниз на j узлов по вертикали, где j – индекс столбца матрицы A.
3) блоки столбцов матрицы C сдвигаются циклично вниз на i узлов по горизонтали, где i - индекс столбца матрицы C.

Алгоритм выполняется за q шагов, где q – размерность вычислительной решетки. Каждый шаг состоит из следующих действий:

a) на вычислительном узле решетки с индексами (i, j) производится умножение блоков A_{ij} и B_{ij}.
b) Циклическое смещение блоков матрицы A влево на 1 узел по горизонтали решетки.
c) Циклическое смещение блоков матрицы C вниз на 1 узел по вертикали решетки.

Результат умножения матриц хранится в матрицы C, блоки которой подлежат смещению. Поэтому по завершению нужно выровнять матрицу до выходного отображения блоков.

Анализ эффективности

Аналогично алгоритму вычисления матричного произведения с сохранением отображения блоков матрицы A рассчитываем суммарное время выполнения алгоритма:

$$T_{CannonB} = 3qT_{alignAB} + 2T_{rollShift} + pT_{AB}$$

$$T_{CannonB} = (3q+2)(t_s + t_w \cdot k^2)(q+1) + (k^2 \cdot (2k-1) + k^2) \cdot p \cdot \tau \quad (11)$$

Получаем ускорение параллельного алгоритма и эффективность использования параллельным алгоритмом процессоров при решении задачи:

$$S_{CannonB} = \frac{(k \cdot q)^2 (2k \cdot q - 1)}{(3q+2)(t_s + t_w \cdot k^2)(q+1) + (k^2 \cdot (2k-1) + k^2) \cdot p \cdot \tau} \quad (12)$$

$$E_{CannonB} = \frac{S_{CannonB}}{p} \quad (13)$$

Сравнительный анализ алгоритмов

Отличия вышерассмотренных алгоритмов состоит в коммуникационных затратах. Рассмотрим графики поведения алгоритмов, на которых черным цветом представлен алгоритм вычисления матричного произведения с сохранением отображения блоков матрицы-результата C, а серым - алгоритмы вычисления матричного произведения с сохранением отображения блоков матрицы A и B.

На Рис.1 отображается поведение функции времени для фиксированных матриц, $n = 10000$, в зависимости от количества узлов решетки.

На Рис.2 показана зависимость функции времени для фиксированного числа узлов решетки, $p = 10000$, в зависимости от размера матриц.

На Рис.3 показано поведение функции ускорения для фиксированных матриц, $n = 10000$, в зависимости от количества узлов решетки.

На Рис.4 показана зависимость функции ускорения фиксированного числа узлов решетки, $p = 10000$, в зависимости от размера матриц.

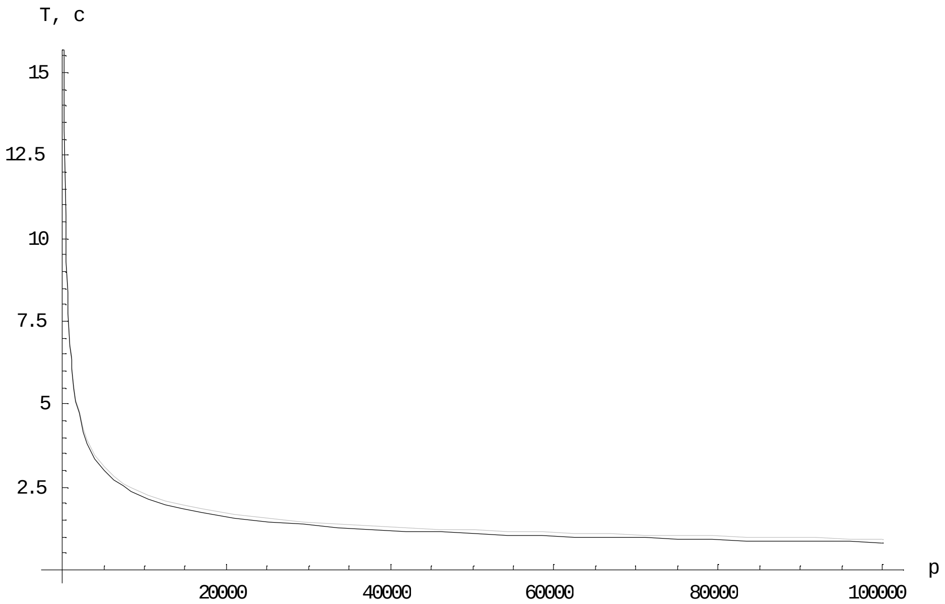

Рис.1 – График зависимости времени выполнения от количества узлов

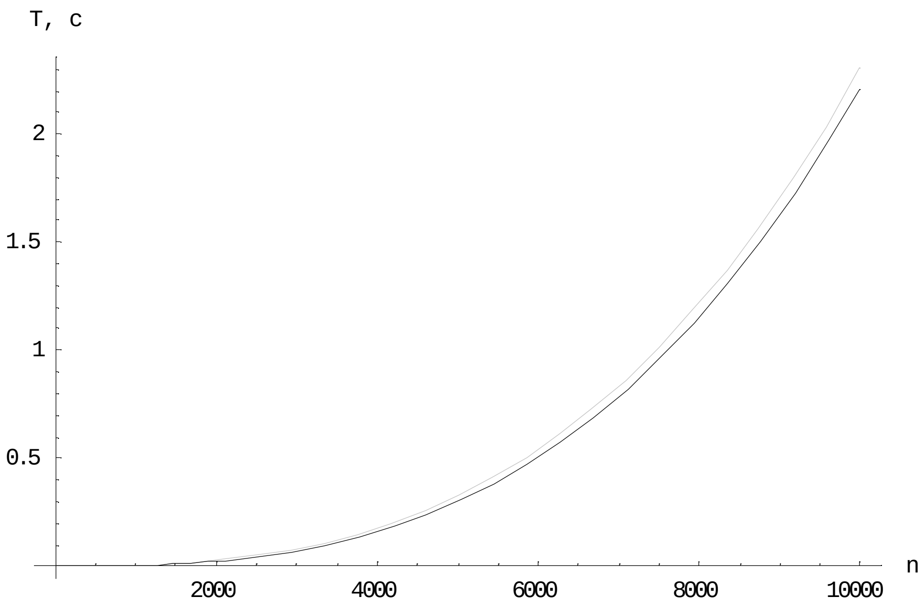

Рис.2 – График зависимости времени выполнения от размера матрицы

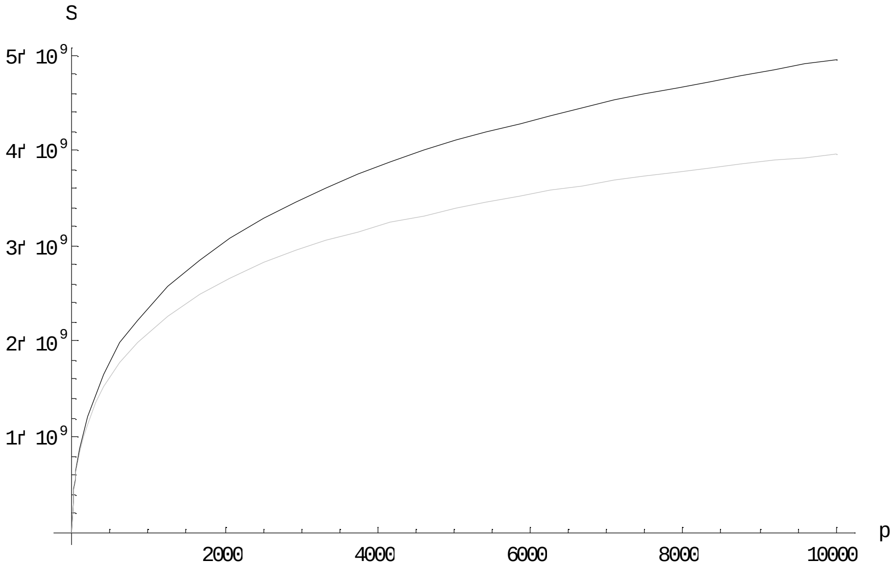

Рис.3 – График зависимости ускорения от количества узлов

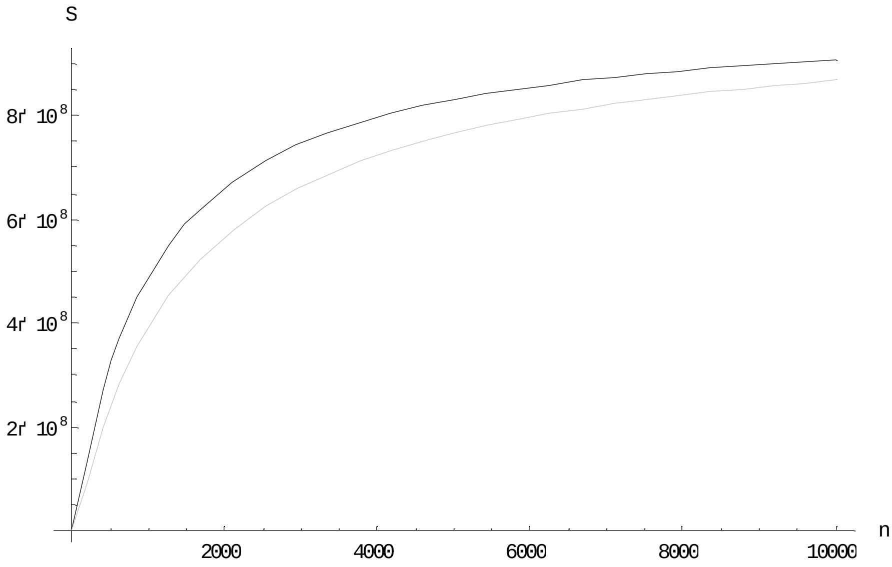

Рис.4 – График зависимости ускорения от размера матрицы

Выводы

В работе были проведены исследования эффективности и масштабируемости алгоритмов семейства Кэннона. Все три алгоритма показывали близкие значения в отдельных ситуациях, однако наиболее эффективным с точки зрения времени выполнения оказался алгоритм вычисления матричного произведения с сохранением отображения блоков матрицы-результата C, что было определено, анализируя графики зависимостей.

Литература

1. Гергель В.П. Теория и практика параллельных вычислений.

2. Интернет-Университет Информационных Технологий - дистанционное образование. http://www.intuit.ru/

3. Gupta A., Kumar V. Scalability of parallel algorithm for matrix multiplication // Technical report TR-91-54, Department of CSU of Minneapolis, 2001