Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

# Распараллеливание алгоритма умножения матриц

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование информационных систем
факультета компьютерных наук и информационных технологий
Шершова Дениса Андреевича

Научный руководитель:
Доцент кафедры ИиП, к.э.н., доцент _____ Л.В. Кабанова

подпись, дата

Зав. кафедрой:
к.ф.-м.н _____ М.В. Огнева

подпись, дата

Саратов 2017

**Введение.** Одна из основных задач матричных вычислений — умножение матриц. Эта операция используется практически везде: от компьютерной графики до расчета поверхностей самолетов. Существуют различные алгоритмы [9], применяющиеся для больших матриц. Вопрос о предельной скорости умножения больших матриц, также как и вопрос о построении наиболее быстрых и устойчивых практических алгоритмов умножения больших матриц остается одной из нерешенных проблем линейной алгебры. Существует множество алгоритмов их быстрого перемножения, таких как алгоритм Штрассена, Копперсмита - Винограда и другие.

В данной работе рассматриваются алгоритмы перемножения матриц, возможные подходы к их распараллеливанию с помощью технологии OpenMP и MPI, сравнивается время работы алгоритма при использовании данных технологии, проводится анализ полученных результатов.

**Цель бакалаврской работы** – является изучение и реализация параллельного алгоритма Кэннона умножения матриц средствами OpenMP, MPI и их анализ.

Поставленная цель определила **следующие задачи**:

1. Изучить возможности технологии OpenMP и MPI для распараллеливания алгоритмов перемножения матриц.

2. Изучить основные алгоритмы перемножения матриц и возможности их распараллеливания.

3. Реализовать параллельный алгоритм Кэннона перемножения матриц с использованием технологий OpenMP и MPI.

4. Провести анализ последовательного и параллельного алгоритма, его работы в среде OpenMP и MPI.

Теоретическая значимость работы состоит в рассмотрении основных параллельных алгоритмов перемножения матриц и возможности их распараллеливания на основе OpenMP и MPI.

Практическая значимость работы состоит в реализации параллельного алгоритма Кэннона средствами OpenMP и MPI.

**Структура и объём работы.** Бакалаврская работа состоит из введения, 2 основных разделов, заключения, списка использованных источников и двух приложений. Общий объем работы – 52 страницы, из них 37 страниц – основное содержание, включая 10 рисунков и 3 таблицы, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

**Основное содержание работы. Первая глава «**Основные алгоритмы перемножения матриц и возможности их распараллеливания**»** посвящена изучению алгоритмов перемножения матриц и возможности распараллеливания таких алгоритмов, как алгоритм ленточный, Фокса и Кэннона.

В ленточном алгоритме матрицы разбиваются на непрерывные последовательности строк или столбцов (полосы). В простейшем случае полосой может служить отдельная строка или столбец.

В блочных алгоритмах (Кэннона и Фокса) матрицы разбиваются на блоки, представляющие собой подматрицы исходных матриц.

В алгоритме Фокса вначале производится рассылка в процесс с координатами (i, j) блоков Aij, Bij исходных матриц. Кроме того, выполняется обнуление матрицы Cij, предназначенной для хранения соответствующего блока результирующего произведения AB. Затем запускается цикл по m (m = 0, …, q). После завершения цикла в каждом процессе будет содержаться матрица Cij, равная соответствующему блоку произведения AB. Останется переслать эти блоки главному процессу.

Алгоритм Кэннона отличается от алгоритма Фокса двумя аспектами. Во-первых, начальная пересылка блоков матриц A и B в процессы выполняется таким образом, чтобы получаемые блоки сразу могли быть перемножены без каких-либо пересылок данных. Во-вторых, при организации цикла выполняется циклическая пересылка как блоков матрицы A (по строкам), так и блоков матрицы B (по столбцам).

В вычислительных системах с распределенной памятью процессоры работают независимо друг от друга. Для организации параллельных вычислений в таких условиях необходимо иметь возможность распределять вычислительную нагрузку и организовать информационное взаимодействие между процессорами [1].

Решение этих вопросов и обеспечивает интерфейс передачи данных (message passing interface – MPI). MPI – это стандарт, которому должны удовлетворять средства организации передачи сообщений. Так же, MPI – это программные средства, которые обеспечивают возможность передачи сообщений и при этом соответствуют всем требованиям стандарта MPI [13]. Так, по стандарту, эти программные средства должны быть организованы в виде библиотеки MPI и должны быть доступны для наиболее широко используемых алгоритмических языков C и Fortran.

Так же, одним из наиболее популярных средств программирования для компьютеров с общей памятью, базирующихся на традиционных языках программирования и использовании специальных комментариев, в настоящее время является технология OpenMP. За основу берётся последовательная программа, а для создания её параллельной версии пользователю предоставляется набор директив, функций и переменных окружения [10]. Технология OpenMP нацелена на то, чтобы пользователь имел один вариант программы для параллельного и последовательного выполнения [14]. Однако возможно создавать программы, которые работают корректно только в параллельном режиме или дают в последовательном режиме другой результат.

Операции с матрицами большой (тысячи/сотни тысяч) размерности являются основой многих численных методов (например, МКЭ – метода конечных элементов).

**Вторая глава «**Распараллеливание алгоритмов средствами OpenMP и MPI и их анализ**»** посвящена реализации алгоритма Кэннона, в котором поставлена задача C = A * B , где C,A,B - квадратные матрицы размером m*m, тогда для корректной работы алгоритма нам потребуется m*m=p процессоров. Разобьём каждую из матриц A и B на p квадратных блоков и зададим нумерацию процессоров P(i,j), где i = 1,m-1 , j = 1,m-1 и умножение блоков A(i,j) и B(i,j) привяжем к процессору P(i,j) соответственно.

Разбиение матриц на блоки в алгоритме Кэннона осуществляется таким образом, чтобы располагаемые блоки в процессорах могли быть перемножены без каких-либо дополнительных коммуникаций между процессорами, а так же чтобы передача данных между процессорами в процессе выполнения алгоритма могла быть осуществлена с помощью простых коммуникационных операций, таких как циклический сдвиг.

Сам алгоритм состоит из двух больших операций - это инициализация матриц, то есть подготовка к умножению, и основной цикл алгоритма.

Инициализация матриц:

для каждой строки i, кроме первой, циклический сдвиг блоков на i - 1 позиций влево;

для каждого столбца j, кроме первого, циклический сдвиг блоков на j - 1 позиций вверх.

Основной цикл:

Формально определим операции: (*) - циклический сдвиг влево блоков строки на 1 позицию (**) - циклический сдвиг вверх блоков столбца на 1 позицию

Тогда алгоритм выглядит следующим образом:

Для всех строк кроме первой производим коммуникационную операцию (*)

Для всех столбцов кроме первого производим коммуникационную операцию (**)

Вычисляем C(i,j) = C(i,j) + A(i,j) * B(i,j)

for (int i = 0; i <= m - 1; i++)

{

Для всех строк производим коммуникационную операцию (*). Для всех столбцов производим коммуникационную операцию (**). Вычисляем C(i,j) = C(i,j) + A(i,j) * B(i,j)

}

Алгоритм Кэннона реализован средствами OpenMP и MPI.

Производились перемножения матриц размером от 100x100 до 1000x1000 с шагом 100 в среде OpenMP. Эксперимент проводился на 4-х процессах персонального компьютера со следующими характеристиками:

AMD FX-6350 Six-Core processor (3.9GHz, 8MB Cache) Microsoft Windows 10 Pro x64

Компилятор MS Visual Studio 2015

Разница во времени выполнения последовательного и параллельного алгоритма Кэннона на OpenMP при размерности 100x100 составила 0,002405, а при 1000x1000 составила 0,764956. При увеличении размерности в 10 раз разница увеличилась в 1,314 раза.

Так же были произведены перемножения матриц размером от 100x100 до 1000x1000 с шагом 100 в среде MPI. Эксперимент проводился на 4-х процессах персонального компьютера со следующими характеристиками:

AMD FX-6350 Six-Core processor (3.9GHz, 8MB Cache) Microsoft Windows 10 Pro x64

Компилятор MS Visual Studio 2015

Разница во времени выполнения последовательного и параллельного алгоритма Кэннона на MPI при размерности 100x100 составила 0,001619, а при 1000x1000 составила 0,070219. При увеличении размерности в 10 раз разница увеличилась в 1,023 раза.

**Заключение.** В данной дипломной работе рассмотрен стандарт для программирования в системах с распределенной памятью MPI и OpenMP; были рассмотрены следующие алгоритмы параллельного перемножения матриц: алгоритм Кэннона, ленточный алгоритм и алгоритм Фокса; алгоритм Кэннона был реализован; целью являлось изучение и реализация параллельного алгоритма Кэннона умножения матриц средствами OpenMP и MPI и их анализ. Изучены возможности технологии OpenMP и MPI для распараллеливания алгоритмов перемножения матриц, так же рассмотрены основные алгоритмы перемножения матриц и возможности их распараллеливания, реализован параллельный алгоритм Кэннона перемножения матриц с использованием технологий OpenMP и MPI и проанализированы результаты работы последовательного и параллельного алгоритма Кэннона в среде OpenMP и MPI, а так же были проведены сравнения этих двух сред, в результате которых выяснилось, что подсчеты в среде OpenMP значительно эффективнее. Цель достигнута, задача была реализована и успешно выполнена.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. Шпаковский Г.И., Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI. – Минск:БГУ, 2002.
2. Букатов А.А., Дацюк В.Н., Жегуло А.И. Многопроцессорные системы и параллельное программирование. – Ростов-на-Дону: РГУ, 2000.
3. Немнюгин С.А. Средства программирования для многопроцессорных вычислительных систем. – СПб: СПбГУ, 2007
4. Антонов А.С. "Параллельное программирование с использованием технологии OpenMP: Учебное пособие".М.: Изд-во МГУ, 2009.
5. C. Snir MPI - The Complete Reference, Volume 1 / C.Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra - The MIT Press, 1998.
6. M. Quinn Parallel Programming in C with MPI and OpenMP / M. Quinn - , 2003.
7. Г. Р. Эндрюс. Основы многопоточного, параллельного и распределенного программирования / Г.Р. Эндрюс – Вильямс, 2003.
8. Беллман Р. Введение в теорию матриц. — М.: Мир, 1969
9. Кибернетический сборник. Новая серия. Вып. 25. Сб. статей 1983 — 1985 гг.: Пер. с англ. — М.: Мир, 1988 — В.Б. Алекссев. Сложность умножения матриц. Обзор.
10. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. - 608 с.
11. Антонов А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. -М.: Изд-во МГУ, 2004. - 71 с.
12. Антонов А.С. Введение в параллельные вычисления. Методическое пособие.-М.: Изд-во Физического факультета МГУ, 2002. - 70 с.
13. Андрейченко Д.К., Ерофтиев А.А., Мельничук Д.В. Распараллеливание параметрического синтеза по схеме «Портфель задач» на основе технологии MPI// Изв. Сарат. ун-та. Нов. сер. 2015. Т. 15. Сер. Математика. Механика. Информатика. Вып. 2. С. 222-228.

14. Левин М.П. Параллельное программирование с использованием OpenMP. Изд-во Бином Лаборатория знаний, 2013.
15. Корнеев В.Д. – Параллельное программирование в MPI. Уч. пособие. Ярославль, 2002.
16. Вранешич З,Заки С, Хамахер К. Организация ЭВМ, СПб.: Питер, 2003
17. Quinn M.J Parallel Programming in C with MPI and OpenMP New York, NY: McGraw-Hill, 2004.
18. Афанасьев К. Е. И др. Многопроцессорные вычислительные системы и параллельное программирование. Кемерово: Кузбассвузидат, 2003.
19. Есаулов А.О, Старченко А.В. Параллельные вычисления на многопроцессорных вычислительных системах. Томск: ТГУ, 2002.
20. Дацюк В.Н. и др. Методическое пособие по курсу «Многопроцессорные системы и параллельное программирование». Ростов-на-Дону: РГУ, 2000.