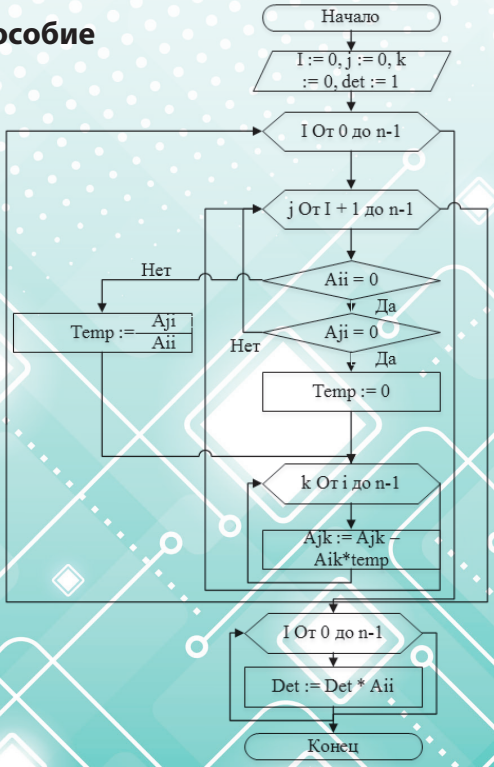
тольяттинский
государственный
университет

Н.В. Хрипунов, А.В. Герасимов

ТЕХНОЛОГИИ МАССИВНО-ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ. ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

Учебно-методическое пособие



Тольятти
Издательство ТГУ
2025

Министерство науки и высшего образования
Российской Федерации
Тольяттинский государственный университет

Н.В. Хрипунов, А.В. Герасимов

ТЕХНОЛОГИИ МАССИВНО-ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ. ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

Учебно-методическое пособие

Тольятти
Издательство ТГУ
2025

УДК 378.091.313(075.8)+004.424(075.8)
ББК 74.480.278я73
Х933

Рецензенты:
канд. экон. наук, доцент, директор Высшей школы интеллектуальных систем и кибертехнологий Поволжского государственного университета сервиса *О.А. Филиппова*;
д-р техн. наук, доцент, профессор кафедры «Прикладная математика и информатика» Тольяттинского государственного университета *С.В. Мкртычев*.

Х933 Хрипунов, Н.В. Технологии массивно-параллельных вычислений. Выполнение курсовой работы : учебно-методическое пособие / Н.В. Хрипунов, А.В. Герасимов. – Тольятти : Издательство ТГУ, 2025. – 29 с. – ISBN 978-5-8259-1782-5.

В пособии приведены методические указания к курсовой работе по курсу «Технологии массивно-параллельных вычислений», описана процедура выбора и формулировки темы курсовой работы, даны требования к структуре и содержанию курсовой работы и к ее оформлению, даны критерии оценки курсовой работы.

Предназначено для студентов, обучающихся по направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем», форма обучения очная и заочная (в том числе с использованием дистанционной образовательной технологии).

УДК 378.091.313(075.8)+004.424(075.8)
ББК 74.480.278я73

Рекомендовано к изданию научно-методическим советом Тольяттинского государственного университета.

ISBN 978-5-8259-1782-5

Нормативные ссылки

В настоящем пособии использованы ссылки на следующие локальные акты ТГУ:

– Положение о курсовой работе (курсовом проекте);

– Методические указания по оформлению выпускных квалификационных работ по программам бакалавриата, программам специалитета, программам магистратуры.

Выбор и формулировка темы курсовой работы

Общий порядок разработки, алгоритм выбора и согласования темы курсовой работы регламентируется Положением о курсовой работе (курсовом проекте).

С целью обеспечения уникальности тем курсовых работ при выборе темы работы используется уникальное сочетание трех смысловых элементов:

– технология 1;
– технология 2;
– задача.

Формулировка темы курсовой работы делается по шаблону «Оценка эффективности параллельных технологий <технология 1> и <технология 2> на примере <задача>», например: «Оценка эффективности параллельных технологий MATLAB и OpenCL на примере решения задачи «Матричное умножение».

Перечень технологий формируется исходя из изучаемых в лекционном материале и на практических занятиях технологий вычислений на графическом процессоре, например CUDA, C++AMP и т. п.

В качестве «технологии 2» допускается технология вычислений на центральном процессоре.

Перечень задач содержит задачи, для решения которых может быть использована как однопоточная, так и многопоточная технология, например, инверсия изображений, умножение матриц, расчет числа Пифагора и т. п.

По согласовании с руководителем курсовой работы допускается обоснованный выбор темы, основанной на стратегии, технологии (технологиях) и задаче (задачах), не включенных в перечни.

Требования к структуре курсовой работы

Общие требования к структуре курсовой работы регламентируются Положением о курсовой работе (курсовом проекте).

Первая – теоретическая – часть работы содержит теоретическую информацию, связанную с темой работы.

Вторая (практическая) часть работы состоит из двух разделов. В первом разделе практической части выполняется разработка программ, во втором – сравнительная оценка эффективности программ.

Типовая структура основной части работы:

1 Постановка задачи на исследование

1.1 Практическая значимость задачи

1.2 Обзор методов решения задачи

1.3 Реализация последовательной программы

2 Проектирование и разработка параллельных программ

2.1 Обзор технологий разработки параллельного программного обеспечения

2.2 Разработка параллельных алгоритмов

2.3 Реализация параллельной программы <технология 1>

2.4 Реализация параллельной программы <технология 2>

3 Экспериментальное исследование эффективности разработанного программного обеспечения

3.1 Методика оценки эффективности программного обеспечения

3.2 Оценка воспроизводимости эксперимента

3.3 Планирование и проведение экспериментальных вычислений

3.4 Анализ результатов эксперимента

По согласовании с руководителем курсовой работы допускается обоснованное изменение типовой структуры основной части работы.

Требования к содержанию курсовой работы

Работа должна быть изложена академическим стилем повествования, которому, в частности, присущи:

– безличная форма (не «мы рассчитали», а «рассчитано»);
– отсутствие необоснованной оценочной составляющей (не «безусловно самым наилучшим методом расчета является», а «методом расчета является» или «одним из методов расчета является»);
– отсутствие прямого обращения к читателю (не «вы можете видеть», а «видно»).

Заимствованные фрагменты должны быть взяты в кавычки, после которых должна быть ссылка на источник. Сам источник с указанием URL должен присутствовать в списке используемой литературы и используемых источников.

В общем случае результаты расчётов округляют до трех значащих цифр. Исключение составляют случаи, когда значащие цифры после третьей имеют принципиальное значение, например, при оценке точности расчета числа Пифагора.

В конце каждого раздела приводятся выводы по разделу, в которых в краткой форме отражаются результаты, полученные в разделе.

Введение

Введение должно содержать:

– вводный текст (общее описание решаемой задачи, отражающее ее актуальность);
– формулировку предмета исследования (это должен быть процесс, например, «процесс решения...»);
– формулировку объекта исследования (это, как правило, программа, модуль, информационная система);
– цель курсовой работы («оценка эффективности массивно-параллельных вычислений на примере... с использованием технологий...»);
– задачи, решаемые в ходе выполнения курсовой работы для достижения указанной цели (как правило, задачи соответствуют разделам работы);
– краткое описание структуры работы («Работа состоит из трех разделов. В первом разделе выполнено...»).

Пример вводного текста: «Умножение матриц является распространенной операцией компьютерной обработки информации, встречающейся в таких вычислительных задачах, как обработка изображений, машинное обучение и т. п. Производители вычислительных машин стараются увеличивать производительность своих продуктов, они могут увеличивать ускорение вычислений посредством ускорения центральных процессоров или посредством увеличения их количества. Из-за определенных ограничений, таких как лимит мощности потребляемой энергии и предел пропускной способности, изготовители компьютеров стали увеличивать производительность за счет увеличения числа чипов, увеличения числа ядер в одном чипе и переноса вычислений на графические процессоры. В свете изложенного работа, направленная на применение массивно-параллельных технологий к операции матричного умножения, является актуальной».

Постановка задачи на исследование

В подразделе 1.1 («Практическая значимость задачи») необходимо:

– показать актуальность задачи;
– составить обзор литературы для выявления современного уровня знаний по задаче;
– определить области применения задачи.

Пример выполнения подраздела 1.1 (фрагмент): «Обратная матрица – это матрица, при умножении на которую исходная матрица даст единичную матрицу.

Операция вычисления обратной матрицы широко используется в обратных задачах, где значение параметров модели должны быть получены из наблюдаемых данных [...]. Обратная задача часто возникает в таких областях, как геофизика, астрономия, медицинская визуализация, компьютерная томография, дистанционное зондирование Земли, спектральный анализ, теория рассеяния и задачи по неразрушающему контролю [...].

Сами по себе матрицы получили широкое распространение. Матрицу можно использовать везде, где данные можно записать в виде прямоугольной таблицы. Так, в программировании матри-

цы – это двумерный массив, в математике матрицы применяются для компактной записи и решения системы линейных и дифференциальных уравнений [...], в экономике математическую модель экономических объектов и процессов удобно записывать в матричном виде, в реляционных базах данных все данные представляются в виде таблиц».

В подразделе 1.2 («Обзор методов решения задачи») необходимо выполнить математическое описание задачи и методов ее решения.

Пример выполнения подраздела 1.2: «Для того чтобы вычислить определённый интеграл

$$\int_a^b f(x)dx \tag{1}$$

методом Монте-Карло, рассматривают случайную величину u, равномерно распределённую на отрезке интегрирования $[a, b]$. Тогда $f(u)$ также будет случайной величиной, причём её математическое ожидание выражается как

$$Ef(u) = \int_a^b f(x)\varphi(x)dx, \tag{2}$$

где $\varphi(x)$ – плотность распределения случайной величины u, равная $\frac{1}{b-a}$.

Таким образом, искомый интеграл

$$\int_a^b f(x)dx = (b-a)Ef(u). \tag{3}$$

Математическое ожидание случайной величины $f(u)$ можно оценить, смоделировав эту случайную величину и посчитав выборочное среднее. Для этого берется N точек, равномерно распределённых на $[a, b]$, для каждой точки u_i вычисляется $f(u_i)$. Затем вычисляется выборочное среднее:

$$\frac{1}{N}\sum_{i=1}^{N} f(u_i). \tag{4}$$

В итоге оценим интеграл:

$$\int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(u_i). \tag{5}$$

Точность оценки зависит только от количества точек N. Этот метод имеет и геометрическую интерпретацию. Он очень похож на описанный выше метод с той разницей, что вместо равномерного разделения области интегрирования на маленькие интервалы и суммирования площадей получившихся «столбиков» область интегрирования забрасывается случайными точками, на каждой из которых строится такой же «столбик» с определением его ширины как $\frac{b-a}{N}$ и суммированием их площади.

Метод Монте-Карло для вычисления одномерных интегралов обычно не применяется, так как для получения высокой точности более удобны квадратурные формулы. Этот метод оказывается более эффективным при вычислении кратных интегралов, когда кубатурные формулы для достижения малой погрешности слишком громоздки и требуют большого объема вычислений.

Для вычисления двойного интеграла формула метода Монте-Карло имеет вид

$$\int_a^b \int_c^d f(x,y)dxdy = \frac{(b-a)(d-c)}{N} \sum_{i=1}^{N} f(x_i, y_i). \qquad (6)$$

Для m-кратного интеграла формула метода Монте-Карло имеет вид

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_m}^{b_m} f(x^1, x^2, \dots, x^m)\, dx^1 dx^2 \dots dx^m \approx$$

$$\approx \frac{(b_1 - a_1)(b_2 - a_2) \dots (b_m - a_m)}{N} \sum_{i=1}^{N} f\left(x_i^1, x_i^2, \dots, x_i^m\right). \qquad (7)$$

Для реализации последовательного алгоритма расчета интеграла по методу Монте-Карло необходимо задать кратность интеграла, область интегрирования и количество итераций вычисления».

В подразделе 1.3 («Реализация последовательной программы») должны присутствовать следующие элементы:

– текстовое описание алгоритма;

– графическое изображения алгоритма;

– текстовое описание выполняемого кода;

– фрагменты кода (полный код приводится в приложении);

– текстовое описание условий тестирования;

– результаты тестирования – снимки экрана, графики, таблицы. Снимки экрана должны быть читаемыми, области, не содержащие информации, необходимо обрезать. Снимки экранов последовательных запусков программы на разных настройках рекомендуется объединять на одном рисунке;

– вывод по результатам тестирования (в тестировании по времени расчета для однопоточной программы следует в качестве базового для тестов многопоточных программ принять результат с временем расчета от 10 до 60 секунд).

Пример текстового описания алгоритма (фрагмент): «На начальном этапе создается матрица размером N на N и заполняется случайными числами. Далее объявляются используемые в вычислениях переменные, после чего начинается обход матрицы в цикле по переменным i от 0 до $N - 1$ и j, которые отвечают за индексы матрицы. Затем следует получить значение переменной, на которое впоследствии будет умножаться элемент матрицы. Если элемент матрицы, на который необходимо делить, окажется равным нулю, то происходит обнуление этой переменной. Если этот элемент не равен нулю, то выполняется деление. После этого шага выполняется обход столбцов матрицы по переменной k от i до $N - 1$. Затем из элемента с индексами j, k вычитается элемент матрицы с индексами i, k, умноженный на посчитанную ранее переменную».

Пример графического изображения алгоритма показан на рис. 1.

Пример текстового описания выполняемого кода и фрагментов кода: «Для реализации последовательной программы на языке C++ необходимо иметь методы для работы с матрицами:

– выделение памяти для матрицы;
– вывод матрицы на экран;
– получение копии матрицы;
– заполнение матрицы случайными числами;
– освобождение памяти, выделенной на матрицу.

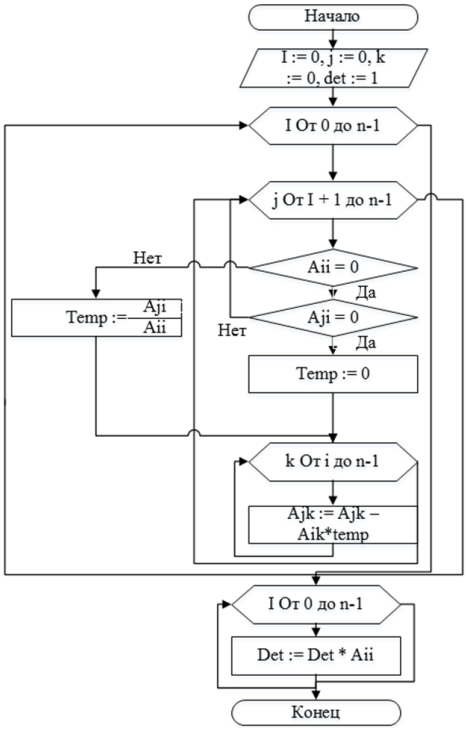


Рис. 1. Последовательное нахождение определителя матрицы

Выделение памяти для матрицы реализуется через функцию `allocateMemory`. Сначала выделяется память под массив указателей, а потом в цикле для каждого указателя выделяется свой участок памяти. В данной функции выделяется память для $N \times N$ элементов матрицы типа `double` (листинг 1).

Листинг 1 – Функция выделения памяти для матрицы

```
double** allocateMemory(int N) // выделить память под массив
{
   double **array = new double*[N];
   for (int i = 0; i < N; ++i)
       array[i] = new double[N];
       return array;
}
```

Для получения копии матрицы реализована функция `getCopyMatrix`. В данной функции выделяется память для новой

матрицы с помощью функции allocateMemory, новая матрица заполняется элементами изначальной матрицы в двух вложенных циклах (листинг 2).

Листинг 2 – Функция получения копии матрицы

```
double **getCopyMatrix(double **matrix, int N) // копирует
матрицу
{
    double** returnMatrix = allocateMemory(N);
        for (int i = 0; i < N; ++i)
        for (int j = 0; j < N; ++j)
               returnMatrix[i][j] = matrix[i][j];
    return returnMatrix;
}»
```

Пример текстового описания и результатов тестирования: «Для тестирования корректности программы умножения матрицы на вектор использована матрица размером 4×4 и вектор из четырех элементов, заполненные случайными целыми числами от 0 до 50. Результаты расчета (рис. 2) показывают, что умножение матрицы на вектор проведено правильно.

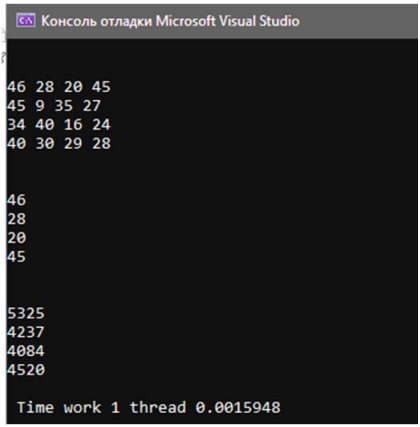


Рис. 2. Результаты тестирования корректности умножения матрицы на вектор

Для тестирования по времени расчёта использован системный таймер Visual Studio и отключен вывод на экран исходных массивов

и результатов расчета. Результаты тестирования по времени расчета представлены в табл. 1.

Таблица 1

Результаты тестирования по времени

Размер матрицы	Время расчёта, с
44	0,00159
400×400	0,0117
2000×2000	0,304
5000×5000	1,89
10 000×10 000	25,9
15 000×15 000	192

По результатам тестирования однопоточной программы умножения матрицы на вектор в качестве базового размера задачи выбран размер матрицы 10 000×10 000 со временем расчета в один поток 25,9 с».

Проектирование и разработка параллельных программ

Подраздел 2.1 «Обзор технологий разработки параллельного программного обеспечения» должен содержать краткий обзор (3–5 страниц) с указанием достоинств и недостатков не менее трех технологий.

Подраздел 2.2 «Разработка параллельных алгоритмов» должен содержать:

– анализ последовательного алгоритма и выбор параллельных секций;
– обоснование при необходимости использования критических секций, условных блокировок, использования библиотек, мьютексов, семафоров, мониторов, блокировок, потокобезопасных коллекций и т. п.

Подраздел 2.3 «Реализация параллельной программы <технология 1>» должен содержать:

– текстовое описание параллельного алгоритма;
– графическое изображение параллельного алгоритма;
– текстовое описание параллельного выполняемого кода;
– фрагменты параллельного кода (полный код приводится в приложении);

– текстовое описание условий тестирования (на корректность и на время расчета);
– результаты тестирования (снимки экрана, графики, таблицы);
– вывод по результатам тестирования.

Пример текстового описания параллельного алгоритма: «Последовательная программа основывается на циклах с параметром. Для параллельного выполнения программы необходимо написать перед циклом `#pragma omp parallel for` (рис. 3). Также необходимо использовать дополнительные директивы `reduction, private, shared`.

Директива `private` применяется к переменным `j,k,temp` для создания локальной копии в каждом потоке. Директива `shared` используется для определения переменной *i* в качестве общей для потоков. Директива `reduction` применяется при вычислении определителя с применением оператора умножения».

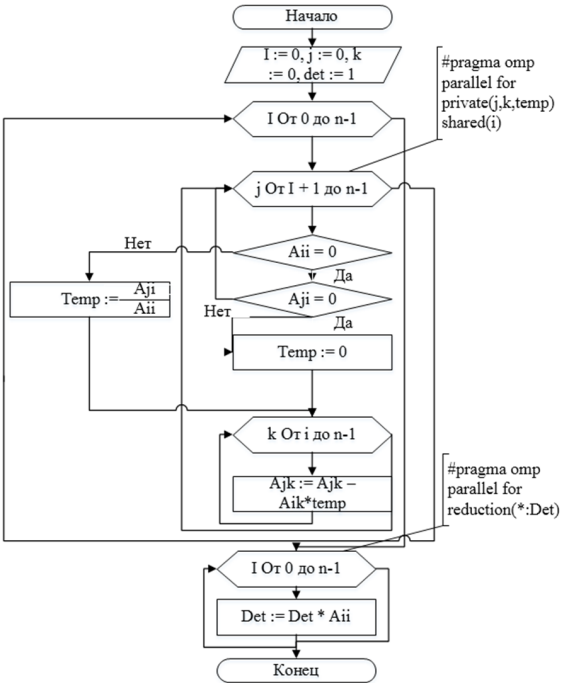


Рис. 3. Схема параллельного алгоритма расчета определителя на OpenMP

Пример описания тестирования (фрагмент): «Для тестирования параллельной программы численного интегрирования методом трапеций выбран интеграл <...> в пределах от <...> до <...> Аналитическое решение данного интеграла равно 2,5.

Для тестирования на корректность расчёта выбран диапазон интервалов от 5 до 10 с шагом 1. Результаты расчета приведены в табл. 2.

Таблица 2

Результаты тестирования на корректность

Число интервалов	Результат расчета	Абсолютное отклонение
5	2,04	0,46
6	2,15	0,35
7	2,24	0,26
8	2,64	0,14
9	2,61	0,11
10	2,44	0,06

Успешность теста оценивается графически (рис. 4) по схождению расчетного значения интеграла к аналитическому значению.

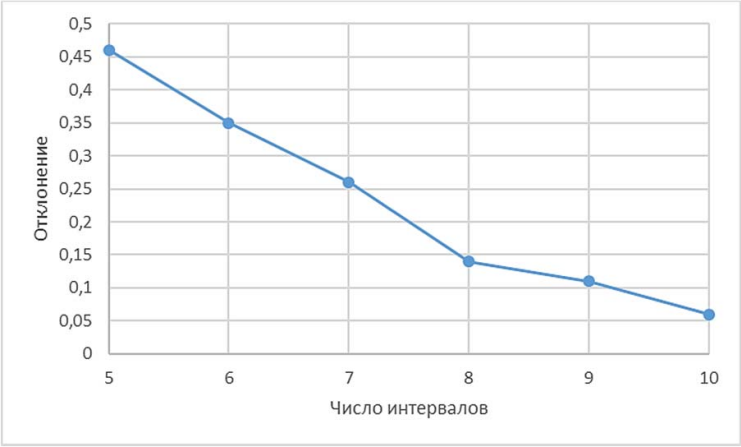


Рис. 4. Зависимость отклонения от числа интервалов

Для тестирования по времени расчета выбрано число интервалов 20 000. Время расчета составило 16 с. Снимок экрана с результатом показан на рисунке ... При тестировании однопоточной программы время расчета составило 32 с.

По результатам тестирования многопоточной программы расчета интеграла методом трапеций по технологии OpenMP установлено:

– многопоточная программа корректна – результат приближается к аналитическому при увеличении числа интервалов;

– многопоточная программа эффективна – расчет выполняется в 2 раза быстрее, чем в один поток».

Подраздел 2.4 «Реализация параллельной программы <технология 2>» по структуре аналогичен подразделу 2.3.

Экспериментальное исследование эффективности разработанного программного обеспечения

Подраздел 3.1 «Разработка методики оценки эффективности программного обеспечения» должен содержать:

– текстовое описание содержания работ;
– описание программно-аппаратного обеспечения (конфигурация компьютера и используемая ОС, характеристики графического и центрального процессоров);
– варьируемые параметры, пределы и интервалы варьирования;
– измеряемые величины (время выполнения программы);
– расчётные величины (коэффициент эффективности, коэффициент ускорения).

В текстовом описании содержания работ приводится информация по сравниваемым программам и цель экспериментального исследования.

Пример текстового описания: «Для оценки эффективности представлены три программы:

1) программа численного интегрирования методом трапеций последовательная;
2) программа численного интегрирования методом трапеций параллельная по технологии OpenMP;

3) программа численного интегрирования методом трапеций параллельная по технологии C++AMP.

Целью экспериментального исследования является оценка эффективности использования технологий параллельных вычислений OpenMP и C++AMP в задаче численного интегрирования методом трапеций».

В ходе описания программно-аппаратного обеспечения можно использовать как встроенные средства операционной системы, так и внешние программы, например CPU-Z и GPU-Z. Информация о системе, на базе которой выполняется эксперимент, может быть представлена в текстовом виде или в форме снимков экрана соответствующих приложений.

Варьируемые параметры, пределы и интервалы варьирования задаются относительно базового размера задачи (задач), относительно которого было рассчитано время выполнения при тестировании программ в разделах 1 и 2 курсовой работы. При выборе пределов варьирования рекомендуется путём предварительного прогона программ на предельных настройках убедиться, что время расчета не превышает 2 минут и не менее 1 секунды.

Пример описания варьируемых параметров, пределов и интервалов: «В качестве варьируемых параметров выбрано число интервалов. Тестирование многопоточных программ, проведенное в разделе 2 при базовом значении $n_Б = 10\ 000$ показало, что время расчета по технологии OpenMP составляет 6,81 секунды, по технологии C++AMP – 3,99 секунды. Однопоточная программа, как показано при тестировании в разделе 1, при $n_Б = 10\ 000$ выполняется за 20,49 секунды.

При выборе пределов варьирования начальное значение:

– верхний предел по числу интервалов $n^+ = 30\ 000$;

– нижний предел по числу интервалов $n^- = 5000$.

Значения интервалов и пределов варьирования параметров приведены в табл. 3.

Таблица 3

Варьируемые параметры

Характеристика варьирования	Число интервалов n
Базовый уровень	10 000
Верхний предел варьирования	30 000
Нижний предел варьирования	5000
Интервал варьирования, верхний/нижний	20 000/5000

Параметр «Число интервалов n» варьируется на трех уровнях».

При описании измеряемых величин (в данной курсовой работе это время выполнения программы) необходимо описать программные конструкции по определению времени выполнения или используемые средства измерения времени, такие как профайлеры.

Расчетные величины это:

– время, затраченное на реализацию последовательного алгоритма T_1;
– время, затраченное на реализацию параллельного алгоритма T_p;
– коэффициент ускорения параллельных вычислений $R_p = T_1 / T_p$.

Подраздел 3.2 «Оценка воспроизводимости эксперимента» должен содержать:

– результаты повтора экспериментов в центре плана;
– анализ воспроизводимости эксперимента.

Целью оценки воспроизводимости является получение информации о разбросе результатов при повторе экспериментов на одних исходных данных. Чаще всего значительный разброс при повторах имеет место, когда исследуются программы сортировки при рандомном задании исходных массивов.

Результаты повтора экспериментов в центре плана должны содержать результаты замеров времени выполнения программы на исходных данных, соответствующих базовым. Должно быть выполнено не менее пяти повторов.

На основании результатов повторов в центре плана выполняется анализ воспроизводимости. В случае если размах повторов в центре плана не превышает 1 % от среднего значения, отклонения

на повторах следует полагать незначительными и строить план эксперимента без повторов. В случае если размах составляет от 1 до 5 %, отклонения на повторах следует полагать существенными и строить план эксперимента с тремя повторами. В случае если размах составляет более 5 %, отклонения на повторах следует полагать значительными и строить план эксперимента с пятью повторами (табл. 4).

Таблица 4

Пример результатов повторных расчетов численного интегрирования методом трапеций при числе интервалов 10 000

Наименование	Один поток	Многопоточное, технология OpenMP	Многопоточное, технология C++AMP
Время повтора 1, с	20,15	6,77	6,95
Время повтора 2, с	20,44	7,01	7,07
Время повтора 3, с	20,67	6,92	6,91
Время повтора 4, с	20,45	6,85	7,09
Время повтора 5, с	20,67	6,92	6,95
Среднее время, с	20,51	6,85	6,96
Размах, с	0,52	0,24	0,18
Отношение размаха к среднему, %	2,55	3,45	2,64

Подраздел 3.3 «Планирование и автоматизация экспериментальных вычислений» должен содержать:

– описание плана эксперимента;

– план и результаты эксперимента.

Пример подраздела 3.3 (фрагмент): «Экспериментальное исследование эффективности параллельных программ вычисления интеграла методом трапеций состоит в последовательном расчете значений интеграла ... при варьировании числа интервалов и количества потоков в соответствии с уровнями и интервалами варьирования (таблица ...) по три повтора каждого эксперимента».

Примерный план эксперимента и результаты замеров времени выполнения программ представлены в табл. 5–7.

Таблица 5

План эксперимента и время выполнения программы в один поток C++

Номер опыта	Число интервалов *n*	Уровень варьирования	Повтор 1, с	Повтор 2, с	Повтор 3, с	Среднее, с
1	5000	–	5,04	5,27	4,95	5,09
2	10 000	0	20,51	20,79	20,63	20,64
3	30 000	+	184,62	184,72	184,56	184,63

Таблица 6

План эксперимента и время выполнения программы по технологии OpenMP

Номер опыта	Число интервалов *n*	Уровень варьирования	Повтор 1, с	Повтор 2, с	Повтор 3, с	Среднее, с
1	5000	–	3,14	3,18	3,22	3,18
2	10 000	0	12,83	12,89	12,88	12,86
3	30 000	+	102,72	102,50	102,40	102,54

Таблица 7

План эксперимента и время выполнения программы по технологии C++AMP

Номер опыта	Число интервалов *n*	Уровень варьирования	Повтор 1, с	Повтор 2, с	Повтор 3, с	Среднее, с
1	5000	–	3,20	3,17	3,17	3,18
2	10 000	0	13,18	13,28	13,34	13,27
3	30 000	+	104,12	104,16	104,66	104,31

Подраздел 3.4 «Анализ результатов эксперимента» должен содержать:

– расчет показателей эффективности по результатам измерений;

– графическое представление и анализ результатов.

Необходимо провести анализ полученных результатов и прокомментировать выявленное поведение контролируемых факторов при изменении варьируемых параметров.

Расчет показателей эффективности по результатам измерений выполняется от средних значений, полученных в ходе экспериментов. Результаты расчета представляются в табличной форме (табл. 8, 9).

Таблица 8

Коэффициенты ускорения работы программы по технологии OpenMP

Число интервалов *n*	Время расчета, с		Коэффициент ускорения
5000	5,09	3,18	1,60
10 000	20,64	12,86	1,60
30 000	184,63	102,54	1,80

Таблица 9

Коэффициенты эффективности работы программы по технологии C++ AMP

Число интервалов *n*	Время расчета, с		Коэффициент ускорения
5000	5,09	3,18	1,60
10 000	20,64	13,27	1,56
30 000	184,63	104,31	1,77

Графическое представление и анализ результатов должны выполняться в соответствии с целью сравнения технологий.

При оформлении графиков обязательно должны указываться названия осей и шкалы. Выбор предельных значений по осям должен обеспечивать хорошую визуализацию данных, например, для представления на графике ряда 0,96, 0,96, 0,95, 0,94 пределы по вертикальной оси следует задать от 0,9 до 1. Пределы от 0 до 1 сделают плохо видимым наклон кривой.

Не следует приводить на одном графике слишком много кривых. Чаще всего в ходе анализа выполняются парные сравнения. Следует там, где это возможно, использовать усредненные данные.

Пример графического представления и анализа результатов: «В ходе анализа необходимо рассмотреть и сравнить многопоточные технологии OpenMP и C++AMP по результатам эксперимента.

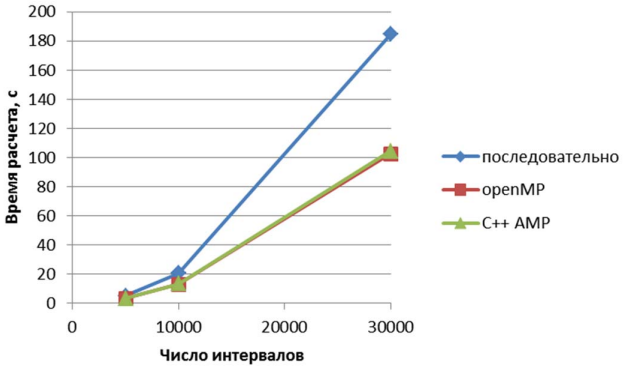


Рис. 5. Зависимость времени расчета от размера задачи

Средние значения коэффициентов эффективности свидетельствуют о том, что по результатам эксперимента исследуемые технологии имеют приблизительно равную эффективность. Технология OpenMP несколько более эффективна (рис. 5).

Влияние варьируемых факторов на коэффициент ускорения показано на рис. 6. На графиках приведены усредненные значения коэффициентов ускорения из табл. 8 и 9.

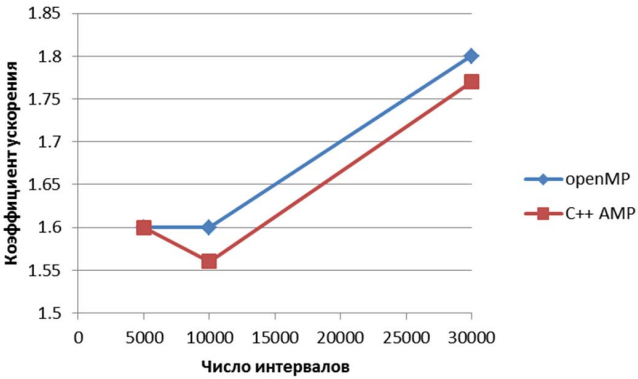


Рис. 6. Влияние числа интервалов на коэффициент ускорения

На графиках видно, что для обеих исследованных технологий с ростом размера задачи коэффициент ускорения увеличивается приблизительно в равной степени».

Заключение

Заключение должно содержать:

– общее описание работы;
– выводы по работе;
– констатацию факта достижения цели работы.

Список источников

Все приведенные в списке источники должны иметь ссылки в тексте работы. Допускается ссылаться сразу на несколько источников, например, «вопросы применения технологии OpenMP рассмотрены в работах [2; 8; 16]».

Приложения

В приложениях к курсовой работе приводится полный код разработанных программ.

Требования к оформлению курсовой работы

В соответствии с п. 4.2 Положения о курсовой работе (курсовом проекте) одной из задач выполнения курсовой работы является «формирование культуры написания выпускной квалификационной работы». В связи с этим при оформлении текста, рисунков, таблиц, заголовков разделов и подразделов курсовой работы необходимо соблюдать правила оформления, изложенные в Методических указаниях по оформлению выпускных квалификационных работ.

Нумерация рисунков, таблиц и листингов сквозная в пределах работы.

Листинги оформляются моноширинным шрифтом (например, Courier New), кегль 12 пт, через одинарный интервал. Заголовок пишется над листингом и содержит номер и название листинга.

Пример оформления листинга в тексте курсовой работы: «Код функции (листинг 1), приводящий матрицу к треугольному виду, содержит внешний цикл от 0 до $n - 1$, где n – количество элементов матрицы.

Листинг 1 – Код функции приведения матрицы к треугольному виду

```
int ComputeGaussianElimination(double **a,int n)
{
   int kolvoPerestanovok = 0;
   double pivot,gmax,pmax,temp;
   int pindmax,gindmax,i,j,k;

   for (k = 0 ; k < n-1 ; k++)
   {
      gmax = 0.0;
// поиск в столбце максимального по модулю элемента и нахождение его индекса строки
            for (i = k ; i < n ; i++)
            {
                  temp = abs(a[i][k]);
                  if (temp > pmax)
                  {
                        gmax = temp;
                        gindmax = i;
                  }
   }
```

```
        if (gmax == 0) return false; // Если матрица
сингулярная, установите флаг & quit.
        if (gindmax != k) // Поменять местами строки при
необходимости
        {
                kolvoPerestanovok++;

                for (j = k; j < n; j++)
                {
                        temp = a[gindmax][j];
                        a[gindmax][j] = a[k][j];
                a[k][j] = temp;
                }
        }

        pivot = -1.0/a[k][k]; // Вычислить точку поворота

        for (i = k+1 ; i < n; i++)
        {
                temp = pivot*a[i][k];
                for (j = k ; j < n ; j++)
                {a[i][j] = a[i][j] + temp*a[k][j];
                }
        }
    }
    return kolvoPerestanovok;
}
```

Дальнейший текст курсовой работы».

В основном тексте работы нужно приводить относительно небольшие фрагменты листинга программ в объеме, необходимом для пояснения текста. Полные листинги необходимо вынести в приложения.

Критерии оценки курсовой работы

К проверке принимаются работы, полностью соответствующие требованиям по оформлению. При отклонении от требований работа возвращается студенту для приведения в соответствие с требованиями по оформлению.

Первым критерием оценки курсовой работы является реализация в работе многопоточности. Если снижение времени расчета при переходе от однопоточной к параллельным программам не продемонстрировано, то цель работы не достигнута и работа получает оценку «неудовлетворительно».

Обязательным требованием является связность результатов – данные, представленные на снимках экрана, в тексте работы, в таблицах и на графиках, должны совпадать.

К защите допускаются курсовые работы, имеющие положительную предварительную оценку. Окончательная оценка выставляется по результатам защиты курсовой работы.

Организация защиты курсовых работ регламентирована Положением о курсовой работе (курсовом проекте).

Рекомендуемая литература

1. Формы, утвержденные Положением о курсовой работе (курсовом проекте) // ТГУ. Тольяттинский государственный университет : [сайт]. – URL: www.tltsu.ru/upravlenie/educational-methodical-management/forms/forms-approved-by-the-regulations-on-course-work-course-project/?ysclid=l8h2l1wif9492728431 (дата обращения: 24.09.2024).
2. Положение о курсовой работе (курсовом проекте) : утверждено решением учёного совета от 19 декабря 2019 года № 262 / Тольяттинский государственный университет. – Тольятти, 2019. – 15 с. – URL: old.tltsu.ru/upravlenie/educational-methodical-management/regulatory-documents-of-educational-process/?ysclid=l8h3ue7t1x546842592 (дата обращения: 24.09.2024).
3. Методические указания по оформлению выпускных квалификационных работ по программам бакалавриата, программам специалитета, программам магистратуры : утверждены приказом от 30 января 2020 года № 145 : с изменениями, утвержденными приказом от 17 июня 2021 года № 1180 / Тольяттинский государственный университет. – Тольятти, 2021. – 39 с. – URL: old.tltsu.ru/upravlenie/educational-methodical-management/regulatory-documents-of-educational-process/?ysclid=l8h3ue7t1x546842592 (дата обращения: 01.09.2022).

Содержание

Учебное издание

Хрипунов Николай Владимирович
Герасимов Антон Владимирович

ТЕХНОЛОГИИ МАССИВНО-ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ. ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

Учебно-методическое пособие

Редактор *О.И. Елисеева*
Технический редактор *Н.П. Крюкова*
Компьютерная верстка: *Л.В. Сызганцева*
Художественное оформление: *И.И. Шишкина*

При оформлении обложки использовано изображение от vector_corp (сайт ru.freepik.com)

Подписано в печать 14.11.2025. Формат 60×80/16.
Печать оперативная. Усл. п. л. 1,68.
Тираж 100 экз. Заказ 1-31-24.

Издательство Тольяттинского государственного университета
445020, г. Тольятти, ул. Белорусская, 14,
тел. 8 (8482) 44-91-47, www.tltsu.ru