



Литература

1. Бретт Маклахлин. Java и XML, 2-е издание, Символ-Плюс, 2016. – 544 страницы.
2. Интеграция XML данных — другой путь // [habrahabr.ru] –2017 – URL: <https://habr.com/ru/post/325186/>
3. Выгрузка условий конкурсов с zakupki.gov.ru // [habrahabr.ru] –2015 – URL: <https://habr.com/post/253201/>
4. Чеботарев, А. XML: свобода, ограниченная только фантазией// Компьютеры+Программы. - 2003. - № 5. - С. 52-55
5. Лыгина, Н.И., Пудич, А.С.. Исследование правильности и эффективности средств парсинга информации на веб-ресурсах// Инновационная наука, №. 3-1, 2017, С. 59-67.

М.А. Кузин¹, Д.Л. Головашкин²

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ БЛОЧНО-ПАРАЛЛЕЛЬНОГО АЛГОРИТМА РАЗНОСТНОГО РЕШЕНИЯ ВОЛНОВОГО УРАВНЕНИЯ

¹ Самарский национальный исследовательский университет имени академика С.П. Королева,

² Институт систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН)

Введение

Актуальность разностного решения волнового уравнения связана с его использованием в современной вычислительной оптике [1,2]. Основной проблемой при этом признается [2] значительная длительность расчетов при реализации такого решения, обусловленная высокой вычислительной сложностью. Популярным в последнее время инструментом решения указанной проблемы, в англоязычной литературе именуемым “tiling” [3], является построение блочных алгоритмов. Классическим подходом к сокращению длительности вычислений при разностном решении волнового уравнения по-прежнему остается разработка параллельных алгоритмов [4]. Объединение перечисленных методик представляется авторам настоящей работы весьма перспективным в силу успешности такого приема при организации вычислений по FDTD (Finite-Difference Time-Domain) методу [5] и несомненной связи обоих численных методов [2,6].

1. Исследование параллельного и блочного алгоритмов

В качестве аппаратной базы для проведения экспериментов была выбрана ЭВМ, оснащенная процессором AMD Ryzen 5 2600, 3,4 ГГц, материнской платой MSI B450 Tomahawk (MS-7C02) с частотой системной шины 8000 МГц и оперативной памятью DDR4-3200МГц объемом 16 Гб. Работающая под управлением операционной системы Ubuntu 18.04.2. Для компиляции кода использо-



вался gfortran 7.4 с ключом компиляции -O3. При параллельной реализации задействовались лишь два вычислительных потока с целью исследования особенностей протекания вычислительного процесса на этом простом примере.

Двумерное волновое уравнение, разностная схема и программная реализация тривиального алгоритма выбирались из работы [7], соответствующий алмазный блочный алгоритм из [2], а параллельный алгоритм и его реализация составлялись посредством авторской актуализации алгоритма и кода из [4], где использовалась библиотека MPI.

Сеточная область с дискретизацией $3e4 \times 3e4$ узлов по пространству ($N=3e4$) и 120 по времени занимала в оперативной памяти 10 Гб при условии хранения двух последних временных сечений сеточной функции.

Декомпозиция сеточной области при составлении параллельного алгоритма производилась вдоль оси, по которой эта же область разбивается (одномерным разбиением) на блоки. От требования систоличности авторы отказались с целью минимизации коммуникационных издержек.

При выбранных параметрах длительность расчетов по тривиальному алгоритму составила 76,93 сек., по параллельному - 47,16 сек. Полученное ускорение, 1,63 раза, заметно меньше ожидаемого для двухзадачного алгоритма. В ходе исследования параллельного алгоритма (комментировались определенные фрагменты кода) было сделано предположение, что в падении ускорения повинны не коммуникационные издержки, а одновременная работа двух вычислительных потоков над общей кэш-памятью процессора.

Использование блочного алгоритма также не привело к ожидаемому результату. Если в [2] его применение, правда на другой аппаратной и системной базе, позволило двукратно ускорить вычисления за счет снижения объема коммуникаций между оперативной памятью и кэш-памятью процессора, то в рассматриваемом случае ускорение составило лишь 1,7 раза. Отметим, что наименьшее время вычислений – 45,24 сек. получено при величине блочного параметра 20 (таблица 1).

Таблица 1 Зависимость длительности вычислений T (сек.) по блочному алгоритму от блочного параметра n.

n	2	4	8	10	12	20	30	40	60
T	59,44	49,46	46,97	46,62	46,77	45,24	77,85	77,84	77,91

Зависимость времени расчетов от значения блочного параметра имеет ожидаемый [2] U-образный вид.

2. Исследование блочно-параллельного алгоритма

Стремясь дополнительно снизить длительность вычислений, авторы составили на основе блочного и параллельного алгоритмов гибридный блочно-параллельный. При оговоренной ранее декомпозиции сеточной области на задачи и разбиении на блоки оказывается возможной организация параллельных вычислений внутри каждого блока при сохранении последовательности их пе-



ребора. Общее количество коммуникаций между задачами при этом увеличивается в N/n раз при неизменном объеме.

Наименьшее время вычислений – 24,86 сек. получено как и ранее при $n=20$, ускорение по сравнению с тривиальным алгоритмом достигло значения 3,1 (таблица 2).

Таблица 2 Зависимость длительности вычислений T (сек.) по блочно-параллельному алгоритму от блочного параметра n .

n	2	4	8	10	12	20	30	40	60
T	40,38	29,09	26,27	26,02	26,38	24,86	25,06	28,88	41,71

Любопытно, что несмотря на одинаковое оптимальное значение блочного параметра для блочного и блочно-параллельного алгоритмов, драматический рост длительности вычислений приходится у этих алгоритмов на разные значения n . В таблице 1 это $n=30$ (соответствует объему блока в 10,3 Мб), в таблице 2 указанный эффект наблюдается для $n=60$ (20,6 Мб) при объеме кэш-памяти уровня L3 у выбранного процессора в 16 Мб. Видимо, при реализации не параллельного блочного алгоритма последовательному вычислительному процессу не удастся использовать кэш-память полностью. Уже для двух вычислительных потоков (данный процессор поддерживает 12 потоков) указанное ограничение снимается, что и обуславливает рост ускорения для блочно-параллельного алгоритма по сравнению с блочным. Так, при $n=30$ расчет по блочно-параллельному алгоритму производится в 3,11 раза быстрее, чем по блочному, хотя задействуются всего два вычислительных потока.

Благодарности

Работа выполнена при поддержке гранта РФФИ 19-07-00423 А.

Заключение

Применение представленного блочно-параллельного алгоритма разностного решения волнового уравнения позволяет аккумулировать достоинства различных подходов (блочного и параллельного) к решению задачи ускорения расчетов по разностному методу, нивелируя их недостатки.

Литература

1. Козлова, Е.С. Моделирование предвестников Зоммерфельда и Бриллюэна в среде с частотной дисперсией на основе разностного решения волнового уравнения / Е.С. Козлова, В.В. Котляр // Компьютерная оптика. – 2013. – Т. 37, No 2. – С. 146-154.
2. Яблокова, Л.В. Блочные алгоритмы совместного разностного решения уравнений Даламбера и Максвелла / Л.В. Яблокова, Д.Л. Головашкин // Компьютерная оптика. – 2018. – Т. 42, No 2. – С. 320-327.



3. Perepelkina, A.Yu Diamond Torre Algorithm for High-Performance Wave Modeling/ A.Yu Perepelkina, V.D.Levchenko // Keldysh Institute Preprints. 2015. – Vol. 18. – P. 20.

4. Barney, B Introduction to Parallel Computing/ B. Barney. - Lawrence Livermore National Laboratory, 2010. – 187 p.

5. Orozco, D.A. Mapping the FDTD Application to Many-Core Chip Architectures / D.A. Orozco, G.R. Gao // Parallel Processing. – 2009. – P. 309-316.

6. Maloney, J.G. Adaptation of FDTD techniques to acoustic modeling / J.G. Maloney, K.E. Cummings // 11th Annual Review of Progress in Applied Computational Electromagnetics. – 1995. – Vol. 2. – P. 724-731.

7. Головашкин, Д.Л. Совместное разностное решение уравнений Даламбера и Максвелла. Двумерный случай / Д.Л. Головашкин, Л.В.Яблокова, Е.Ю. Булдыгин // Компьютерная оптика. – 2014. – Т.38, No1. – С. 20-27.

К.С. Кульга, Э.Р. Асбапов

ДИНАМИЧЕСКИЙ АНАЛИЗ КОМПОНОВОК СТАНКОВ НА СТАДИИ ЭСКИЗНОГО ПРОЕКТИРОВАНИЯ С ПОМОЩЬЮ CAD/CAE-СИСТЕМ

(Уфимский государственный авиационный технический университет «УГАТУ»)

Актуальность проблемы. Компоновка мехатронного станка (далее, станок), как правило, состоит из одного стационарного (постоянно неподвижного) и нескольких подвижных элементов, разделённых соединениями (стыками) [1]. Создание (синтез) компоновки осуществляется на ранней стадии проектирования станка после определения его характеристик, выбора метода и движений формообразования, предварительного определения кинематической структуры. Компоновка станка влияет на его основные показатели качества: статические и динамические упругие деформации, термические деформации, на распределение давлений и износ в подвижных соединениях.

Цель научного исследования. Разработка методики динамического анализа компоновок станков с ЧПУ на стадии эскизного проектирования на основе применения программного обеспечения (ПО) CAD (*Computer Aided Design*)/CAE(*Computer Aided Engineering*)-систем и метода конечных элементов (МКЭ).

Теоретическая часть. Совместное использование ПО CAD/CAE-системы позволяет в максимальной степени задействовать достоинства графической среды (параметризация, поверхностное и твердотельное геометрическое моделирование, кинематика сборок) и получить результаты статического, модального и динамического анализов компоновок станков на стадии эскизного проектирования [2].

Рассмотрим этапы реализации предлагаемой методики создания конечно-элементной модели компоновки станка для механической обработки детали,