Кемеровский государственный университет

Учебный курс

Параллельное программирование

Тема №8

Семестровые задания

Лектор

Стуколов Сергей Владимирович

Содержание

- Разбор семестровых заданий
- Формулировка требований к выполнению заданий

Задание №1: Об “обедающих философах”

- Для реализации потребуется пять процессоров.
- Суть задачи следующая: пять философов сидят за круглым столом. Они проводят жизнь, чередуя приемы пищи и размышления.
- В центре стола находится большое блюдо спагетти. Философам, чтобы съесть порцию спагетти, требуется две вилки.
- Вилок всего пять: между каждой парой философов лежит по одной вилке.
- Каждому философу дозволительно пользоваться только вилками, которые лежат рядом с ним (слева и справа).

Задание №1: Об “обедающих философах”

Задание №1: Об “обедающих философах”

- Задача – написать программу, моделирующую поведение философов. Программа должна избегать ситуации, в которой все философы голодны, то есть ни один из них не может взять себе две вилки (например, когда каждый философ держит по одной вилке и не хочет отдавать ее). Раз вилок всего пять, то одновременно могут есть не более, чем двое философов. Два сидящих рядом философа не могут есть одновременно. Предположим, что периоды раздумий и приемов пищи различны – для их имитации в программе можно использовать генератор случайных чисел. Имитация поведения каждого философа может быть разбита на следующие блоки: поразмыслить, взять вилки, поесть, отдать вилки. Вилки являются разделяемым ресурсом.

Задание №1: Об “обедающих философах”

- Запрограммируйте остановку алгоритма по достижении контрольного времени. Выведите в файл результатов общее время реализации параллельного алгоритма, количество приемов пищи для каждого философа, постройте схему работы алгоритма. Например:

Этап	1 философ	2 философ	3 философ	4 философ	5 философ
1	размышление	размышление	размышление	размышление	размышление
2	Взял левую вилку	размышление	Взял левую вилку	размышление	размышление
3	Взял правую вилку	размышление	Взял правую вилку	размышление	Взял левую вилку
4	поел	размышление	поел	размышление	размышление
5	Отдал вилки	размышление	Отдал вилки	размышление	Взял правую вилку

Задание №2: “о восьми ферзях”

- Нужно расставить на шахматной доске восемь ферзей так, чтобы они не атаковали друг друга. разработайте программу, которая строит все 92 решения этой задачи.
- Обнаружив часть решений, процессы должны передать их на 0-й процесс.
- Программа должна найти все решения и завершить работу. Управляющий процесс должен все решения записать в файл результатов.

Задание №3: “о пяти ферзях”

- Запрограммируйте параллельную программу, реализующую задачу о расстановке пяти ферзей на шахматной доске, при которой каждое поле будет находиться под ударом одного из них.

Задание №4: "о коммивояжере"

- Эта классическая задача имеет практическое применение, например, при планировании обслуживания населения городским общественным транспортом. Дано N городов и симметричная матрица A(N,N). Значением элемента A(i,j) является расстояние между городами i и j. Коммивояжер начинает путь в городе 1 и должен посетить по одному разу каждый город, закончив свой путь снова в городе 1. Требуется найти путь, минимизирующий расстояние, которое придется проехать коммивояжеру, а результат сохранить в векторе B(N). Значением вектора должна быть перестановка целых чисел от 1 до N, соответствующая порядку посещения городов коммивояжером. Разработайте параллельную программу для решения поставленной задачи, используя модель "управляющий-рабочие".

Задание №5: “игра в жизнь”

- Многие биологические и физические системы можно смоделировать в виде набора объектов, которые с течением времени циклически взаимодействуют и развиваются. Некоторые простейшие системы можно моделировать с помощью клеточных автоматов. Основная идея разделить пространство физической или биологической задачи на отдельные клетки. Каждая клетка – это конечный автомат. После инициализации все клетки сначала совершают один переход в новое состояние, затем второй переход и т.д. Результат каждого перехода зависит от текущего состояния клетки и ее соседей.

Задание №5: “игра в жизнь”

- Дано двухмерное поле клеток. Каждая клетка либо содержит организм (жива), либо пуста (мертва). Каждая клетка имеет восемь соседей, которые расположены сверху, снизу, слева, справа и по четырем диагоналям от нее. Игра “жизнь” происходит следующим образом. Сначала поле инициализируется: определяются мертвые и живые клетки (для этой цели в программе можно использовать генератор случайных чисел). Затем каждая клетка проверяет состояние свое и своих соседей и изменяет свое состояние в соответствии со следующими правилами:
  - живая клетка, возле которой меньше двух живых клеток, умирает от одиночества;
  - живая клетка, возле которой есть две или три живые клетки, выживает еще на одно поколение;
  - живая клетка, возле которой находится больше трех живых клеток, умирает от перенаселения;
  - мертвая клетка, рядом с которой есть ровно три живых соседа, оживает.
- Этот процесс повторяется заданное число шагов (поколений). Поле требуется разделить на полосы или блоки клеток, при этом каждая полоса или блок клеток обрабатываются отдельным процессором. Для реализации алгоритма потребуется организовать передачу данных о состоянии пограничных клеток.

Задание №6: “решето Эратосфена”

- Запрограммируйте параллельную программу, реализующую алгоритм “решето Эратосфена” для нахождения всех простых чисел меньше *n*. Решетом Эратосфена называют следующий способ.
  - Выпишем подряд все целые числа от 2 до *n*.
  - Первое простое число 2. Запомним его, а все большие числа, кратные 2, вычеркнем.
  - Первое из оставшихся чисел 3. запомним его, а все большие числа, кратные 3, вычеркнем.
  - Первое из оставшихся чисел 5 (4 уже вычеркнуто как кратное 2), запомним его, а все большие числа кратные 5, вычеркнем и т.д.

Задание №7: “Задача о многопроцессорном расписании”

- Даны **m** одинаковых процессоров и **n** независимых задач, каждая из которых может решаться на любом процессоре. Время решения каждой задачи равно **ti**, **i = 1…n**. Как распределить задачи по процессорам таким образом, чтобы выполнение всех задач было завершено в кратчайший срок?

Задание №8:сортировка последовательности чисел

- Дан ряд случайных чисел, написать параллельную программу сортировки данных чисел.

Задание №9: Определение частоты события

- Дан текст, определить частоту встречи слов в тексте.

Задание №10: метод Монте-Карло

- Запрограммируйте параллельную программу, реализующую алгоритм метода Монте-Карло для вычисления площади круга $x^2 + y^2 \leq 1$
- Методы Монте-Карло – это общее название группы методов для решения различных задач с помощью случайных последовательностей. Эти методы (как и вся теория вероятностей) выросли из попыток людей улучшить свои шансы в азартных играх. Этим объясняется и тот факт, что название этой группе методов дал город Монте-Карло – столица европейского игорного бизнеса.

Задание №10: метод Монте-Карло

- Суть метода Монте-Карло для приближенного вычисления значения кратного интеграла $$I = \iint\limits_D f(x,y)dxdy$$
- где *D* - область интегрирования, вписанная в прямоугольник $$D' = (a \le x \le b, c \le y \le d)$$
- сводится к следующему: выбирается случайным образом *n* точек $z_1 \dots z_n$ положенных в прямоугольнике D’
- вычисляется приближенное значение интеграла по формуле $$I \approx \frac{(b-a)(d-c)}{n} \sum_{k=1}^{m} f(z_k)$$
- где *m* - количество точек, попавших внутрь *D*. Для подсчета площади круга требуется найти интеграл:

$$I = \iint\limits_{x^2+y^2 \le 1} dxdy$$

Задание №10: метод Монте-Карло

- При реализации параллельного алгоритма каждый процессор генерирует свое пространство случайных чисел и их обрабатывает. Запишите в файл результат выполнения параллельной программы, содержащий общее количество сгенерированных случайным образом точек *n* , количество попавших в заданную область точек *m*, приближенное вычисленное значение площади круга, погрешность вычисления.

Задание №11: Умножение матрицы на вектор

- Реализуйте последовательный алгоритм умножения матрицы на вектор, получите зависимость времени реализации алгоритма от размера матрицы. Реализуйте параллельный строчно-ориентированный алгоритм умножения матрицы на вектор, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 1000x1000 до 5000x5000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы. Реализуйте параллельный столбцово-ориентированный алгоритм умножения матрицы на вектор, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 1000x1000 до 5000x5000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы. Проведите сравнение параллельных алгоритмов (строчно- и столбцово-ориентированного) по ускорению и эффективности.

Задание №12: Умножение матрицы на матрицу

- Реализуйте последовательный алгоритм умножения матрицы на матрицу, получите зависимость времени реализации алгоритма от размера матрицы. Реализуйте параллельный алгоритм умножения матрицы на матрицу в случае, когда 1 матрица строчно-слоисто, а 2 - целиком распределены по процессорам, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 1000x1000 до 5000x5000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы.

Задание №13: Метод Якоби решения СЛАУ

- Реализуйте последовательный алгоритм решения СЛАУ методом простой итерации, получите зависимость времени реализации алгоритма от размера матрицы. Реализуйте параллельный алгоритм решения СЛАУ методом простой итерации, когда матрица строчно-слоисто распределена по процессорам, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 100x100 до 1000x1000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы.

Задание №14: Метод Гаусса решения СЛАУ (строчный)

- Реализуйте последовательный алгоритм решения СЛАУ прямым методом Гаусса с выбором ведущего элемента, получите зависимость времени реализации алгоритма от размера матрицы. Реализуйте параллельный строчно-ориентированный алгоритм решения СЛАУ прямым методом Гаусса с выбором ведущего элемента, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 1000x1000 до 5000x5000.

Задание №15: Метод Гаусса решения СЛАУ (столбцовый)

- Реализуйте параллельный столбцово-ориентированный алгоритм решения СЛАУ прямым методом Гаусса с выбором ведущего элемента, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 1000x1000 до 5000x5000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы и выбранного хранения матрицы по процессорам.

Задание №16: Метод Гаусса (усовершенствованный)

- Реализуйте параллельный алгоритм решения СЛАУ прямым методом Гаусса с выбором ведущего элемента с применением приема “опережающего вычисления и опережающей рассылки”, получите зависимость времени реализации алгоритма от размера матрицы, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 1000x1000 до 5000x5000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы и выбранного хранения матрицы по процессорам.

Требования

При реализации алгоритмов на параллельных компьютерах для анализа эффективности варианта распараллеливания можно порекомендовать проводить исследование по следующей схеме:

1) написать последовательный алгоритм решения задачи;
2) определить время, затраченное на реализацию последовательного алгоритма $T_1(n)$;
3) написать параллельный алгоритм решения задачи;
4) определить время, затраченное на реализацию параллельного алгоритма $T_p(n)$ (p – количество процессоров);
5) вычислить ускорение параллельного алгоритма по сравнению с последовательным $S_p(n) = \frac{T_1(n)}{T_p(n)}$;
6) вычислить эффективность параллельного алгоритма $E_p(n) = \frac{S_p(n)}{p}$.

Спасибо за внимание!