

### Задания

- 1) Умножение разреженных матриц. Матрица хранится ненулевыми элементами строки. Сравнить производительность программы для матриц разного размера и при разном количестве рабочих процессов.
- 2) Пусть дана разреженная матрица, представленная ненулевыми элементами строки. Найти транспонированную матрицу. Сравнить производительность программы для матриц разного размера и при разном количестве рабочих процессов.
- 3) Задача аппроксимации интеграла непрерывной функции  $f(x)=\sin(x)*\exp(x)$  на интервале  $[a,b]$ . Интервал разбить на подынтервалы по количеству процессоров, для каждого использовать алгоритм адаптивной квадратуры. *Алгоритм адаптивной квадратуры:*
  - a) Вычислить  $m$  - середину отрезка  $[a,b]$ .
  - b) По правилу трапеций или по правилу Симпсона найти аппроксимацию интеграла на отрезках  $[a,m]$ ,  $[m,b]$  и  $[a,b]$ .
  - c) Если сумма меньших площадей равна большей с некоторой заданной точностью, то аппроксимацию считаем достаточной.
  - d) Если нет, то процесс повторяется для отрезков  $[a,m]$ ,  $[m,b]$ .
- 4) Задача о 8 ферзях. Нужно расставить на шахматной доске 8 ферзей так, чтобы они не атаквали друг друга. Найти все 92 решения (или требуемое количество), используйте модель «управляющий-рабочие». Управляющий задает начальные позиции ферзей. Рабочие процессы ищет все решения, результат либо передает управляющему, либо выводит в файл.
- 5) Задача подсчета числа уникальных слов в словаре (ни одна буква не повторяется). Использовать модель «управляющий-рабочие» или «взаимодействующие равные». Кроме этого вывести самые длинные слова.
- 6) Задача выделения областей изображения. Представить изображение матрицей чисел, определяющих цвет пикселя. Определить количество областей, составляющие изображение, используя алгоритм пульсации для межпроцессорного взаимодействия. Два пикселя принадлежат одной области, если они являются соседями по горизонтали или вертикали. Для решения задачи можно использовать матрицу меток областей, назначая метке максимальное значение среди соседей.

*Алгоритм пульсации* применяется в итерационных приложениях, параллельных по данным. Данные разделяются между процессорами (разбивая матрицу на полосы или блоки). Работа каждого процессора строится по схеме:

```
Инициализация локальных переменных
While (не выполнено)
{
    send значения соседям;
    Recv значения от соседей
    Пересчитать локальные переменные
}
```
- 7) Задача сглаживания изображения. Представить изображение матрицей чисел, определяющих цвет пикселя (1 – есть цвет, 0 - нет). Нужно убрать «пики» и «зазубрины», затемнить все светлые пиксели, у которых как минимум  $d$  соседей не освещены, тоже сделать и для темных пикселей. Использовать алгоритм пульсации (см задачу 6). Рассмотреть задачу для разных изображений, разного количества процессоров, числа  $d$ .
- 8) Задача «эволюция». Дано двумерное поле клеток, каждая из которых либо содержит организм (1), либо пуста (0). Каждая клетка проверяет состояние своих соседей (их 8) и изменяет свое по правилам:
  - a) Живая клетка, вокруг которой  $<2$  живых клеток, умирает от одиночества
  - b) Живая клетка, вокруг которой есть 2 или 3 живых клеток, выживает
  - c) Живая клетка, вокруг которой  $> 3$  живых клеток, умирает от перенаселения

- d) Пустая клетка, рядом с которой равно 3 живых соседа, оживает  
С помощью алгоритмы пульсации (см. задачу 6) показать  $n$  шагов эволюции жизни.  
Константы 2 и 3 можно заменить своими значениями.
- 9) Задача «жизнь волков и зайцев». Дано двумерное поле клеток, каждая из которых либо содержит волка (1), либо зайца (2), либо пуста (0). Каждая клетка проверяет состояние своих соседей (их 8) и изменяет свое по правилам:
- Волки и зайцы живут по правилам задачи 8, для зайцев число 3 можно заменить на 4 или 5
  - Волк съедает одного зайца-соседа, в любом порядке на Ваш выбор, за один шаг
  - Порядок следования правил также выбирается Вами (волк ли съест быстрее зайца, или заяц умрет сам!)
  - Использовать алгоритм пульсации (задача 6)
- 10) Задача «пожар». Дано двумерное поле клеток, каждая из которых либо содержит дерево (1), либо куст (2), либо пуста (0). Пожар начинается в одной клетке (либо выбираем случайным образом, либо задаем явно). Теперь клетка может гореть (5), или быть потушенной(0). Каждая клетка проверяет состояние своих соседей(их 4) и изменяет свое по правилам:
- Куст или дерево загорается, если любой из соседей горит
  - Куст горит 1 поколение, дерево 2 поколения.
  - Куст или дерево загорается, если его окружают 3 горящие клетки по диагонали
- С помощью алгоритма пульсации (задача 6), показать картину пожара поля, пока пожар не закончится.
- 11) Нахождение простых чисел  $\leq n$  решетом Эратосфена. Использовать конвейер процессов- фильтров, каждый получает поток чисел от соседа слева и отправляет поток соседу справа (последний - первому). Первое число, полученное фильтром – будет простым; отсылаем соседу числа, не кратные первому. Конец списка можно отслеживать специальным маркером – тогда процесс завершает работу.
- 12) Генерация простых чисел  $\leq n$  с помощью портфеля задач. Используется алгоритм «управляющий-рабочие». Управляющий передает рабочим процессам числа, кандидаты в простые (например 5, 7, 9, ...). Рабочие процессы имеют начальный массив простых чисел (2,3), получает кандидата на простоту, деля его на меньшие простые числа, полученное простые числа передаются управляющему, который и может рассылать их остальным процессам.
- 13) Решение СЛАУ методом исключений Гаусса. Сравнить производительность программы для систем разного размера и при разном количестве рабочих процессов.
- 14) Выполнить LU-разложение матрицы A. Проверить корректность вычислений, т.е., что  $L*U=A$ . Здесь L – нижняя треугольная матрица, U- верхняя треугольная матрица. Сравнить производительность программы для матриц разного размера и при разном количестве рабочих процессов.
- 15) Найти обратную матрицу. Проверить корректность вычислений. Сравнить производительность программы для матриц разного размера и при разном количестве рабочих процессов.
- 16) Гравитационная задача n- тел. Дано большое число астрономических тел галактики (звезд, пылевых облаков, черных дыр). Каждое тело имеет массу, начальное положение (на плоскости, т.е. задача двухмерная) и скорость. Гравитация вызывает перемещение и ускорение тел. Показать эволюцию системы, т.е. движение тел за заданное число шагов изменения времени. *Величина силы гравитации* между двумя телами  $i$  и  $j$  вычисляется по формуле  $F = \frac{Gm_i m_j}{r^2}$  где  $m_i, m_j$  массы тел,  $r$  – расстояние между ними, значение гравитационной постоянной  $G = 6,67 \cdot 10^{-11}$ . *Направление силы* задается единичным вектором от  $i$  к  $j$  и противоположным вектором от  $j$  к  $i$ . *Общая сила*, действующая на тело, есть сумма сил воздействия всех остальных тел. Приведем

формулы для вычисления скорости и положения тела: ускорение:  $a = F/m$ , изменение скорости:  $dv_i = a_i dt$ , изменение положения тела:  $dp_i = (v_i + \frac{dv_i}{2})dt$  на интервале времени  $dt$ . При реализации задачи использовать модель «взаимодействующие равные», или «круговой конвейер», или алгоритм Барнса – Хата (Barnes - Hut), или быстрый метод мультиполей.

- 17) Промоделируйте движение шаров на бильярдном столе. Задать начальные положения и скорости всех шаров, считая движение идеальное, без трения.
- 18) Решить уравнение Лапласа (эллиптическое дифференциальное уравнение в частных производных) методом Якоби. Известны значения в точках на границе, нужно аппроксимировать решение во внутренних точках области, покрыв область равномерной сеткой точек (сеточные вычисления). В методе Якоби новое значение в каждой точке сетки равно среднему от предыдущих значений ее соседних точек (4 точки). Вычисления выполняются по достижению заданной точности (разница между текущей и предыдущей итерацией). Использовать алгоритм пульсации (см. задачу 6).