

Задания по курсу «Введение в суперкомпьютерные вычисления»

Постановка задачи

Замечания. Для всех заданий: Выполните вычислительные эксперименты при различном количестве потоков (для OpenMP) / процессов (для MPI): меньше, равно или больше числа имеющихся вычислительных элементов, и при разном масштабе задач (разных размерах массивов и матриц). Определите время выполнения программ и оцените получаемое ускорение: постройте графики и таблицы зависимости времени и ускорения от количества потоков/процессов и размера задачи.

Требуется подготовить отчет, показать и прокомментировать исходный код - будем обсуждать в индивидуальном порядке с каждым. В отчете по каждому заданию следует коротко описать суть решения, представить исследование производительности - зависимости времени работы/ускорения от количества используемых параллельных потоков/процессов (меньше, равно, больше числа ядер/узлов) и масштаба задачи (размеров массивов, матриц итп), построить графики и таблицы. Коротко привести итоговый анализ полученных результатов - особенно если требуется провести сравнение различных способов решения одной задачи (например, как в задании OpenMP-6,7).

OpenMP

1. Разработайте программу для нахождения минимального (максимального) значения среди элементов вектора. Рассмотреть вариант с использованием редукции (если используемая версия OpenMP поддерживает ее) и без редукции. (*1 балл*)

2. Разработайте программу для вычисления скалярного произведения двух векторов. (*1 балл*)

3. Разработайте программу для задачи вычисления определенного интеграла с использованием метода прямоугольников (*1 балл*)

$$y = \int_a^b f(x) dx \approx h \sum_{i=0}^{N-1} f_i, f_i = f(x_i), x_i = i \cdot h, h = (b - a) / N$$

4. Разработайте программу решения задачи поиска максимального значения среди минимальных элементов строк матрицы (такая задача имеет место для решения матричных игр) (*2 балла*)

$$y = \max_{1 \leq i \leq N} \min_{1 \leq j \leq N} a_{ij}$$

5. Разработайте программу для задачи 4 при использовании матриц специального типа (ленточных, треугольных и т. п.). Определите время выполнения программы и оцените получаемое ускорение. Выполните вычислительные эксперименты при разных правилах распределения итераций между потоками и сравните эффективность параллельных

вычислений (выполнение таких экспериментов целесообразно выполнить для задач, в которых вычислительная трудоемкость итераций циклов различна). (2 балла)

6. Проведите исследование режимов распараллеливания цикла `for` (`static`, `dynamic`, `guided`): придумайте цикл с неравномерной нагрузкой итераций (например, на некоторых итерациях генерировать случайные числа) и посмотрите, как будет изменяться время работы такой программы при различных режимах работы цикла `for`. (2 балла)

7. Реализуйте операцию редукции с использованием разных способов организации взаимоисключения (атомарные операции, критические секции, синхронизацию при помощи замков). Оцените эффективность разных подходов. Сравните полученные результаты с быстродействием операции редукции, выполняемой посредством параметра `reduction` директивы `for`. (2 балла)

8. Разработайте программу для вычисления скалярного произведения для последовательного набора векторов (исходные данные можно подготовить заранее в отдельном файле). Ввод векторов и вычисление их произведения следует организовать как две раздельные задачи, для распараллеливания которых используйте директиву `sections`. (3 балла)

9. Уточните, поддерживает ли используемый Вами компилятор вложенные параллельные фрагменты. При наличии такой поддержки разработайте программу с использованием и без использования вложенного параллелизма (достаточно разработать программу для одной из задач, например, задачи 4) с использованием распараллеливания циклов разного уровня вложенности. Выполните вычислительные эксперименты и оцените эффективность разных подходов. (2 балла)

Итого: до 16 баллов за OpenMP

MPI

Замечания: При выполнении задач на MPI важно помнить теоретические основы построения параллельных программ, условия достижения ускорения параллельной программы, возможные причины замедления. Если ваша программа замедляется при параллельном выполнении, подумайте, что следует учесть и как, возможно, следует изменить условие задачи, чтобы все-таки добиться ускорения.

Обратите внимание, на каких узлах запускаются процессы. На нашем кластере планировщик располагает процессы в соответствии с количеством ядер, таким образом, на одном узле может быть до 16 процессов. Желательно провести эксперименты с процессами, располагающимися на разных узлах, для этого найдите нужный параметр команды `sbatch`.

1. Разработайте программу для нахождения минимального (максимального) значения среди элементов вектора. (1 балл)

2. Разработайте программу для вычисления скалярного произведения двух векторов. (1

балл)

3. Разработайте программу, в которой два процесса многократно обмениваются сообщениями длиной n байт. Выполните эксперименты и оцените зависимость времени выполнения операции передачи данных от длины сообщения. *(1 балл)*
4. Реализуйте алгоритмы умножения матриц: 1) простейший (ленточный); 2) более сложный на выбор (алгоритм Фокса, алгоритм Кэннона или иной). Сравните производительность работы программ на основе разных алгоритмов. *(5 баллов)*
5. Реализуйте параллельную программу с конфигурируемым объемом вычислений и коммуникаций (разные объемы вычислений можно эмулировать, например, с помощью вызова `sleep` на разное количество микросекунд; объемы коммуникаций регулировать размером и количеством пересылаемых сообщений). Исследуйте производительность (ускорение) программы при различном балансе вычислений и коммуникаций. *(3 балла)*
6. Подготовьте варианты ранее разработанных программ (*достаточно одной из программ, например из задания 4*) с разными режимами выполнения операций передачи данных (синхронный, по готовности, буферизованный). Сравните времена выполнения операций передачи данных при разных режимах работы. *(2 балла)*
7. Подготовьте варианты ранее разработанных программ (*достаточно одной из программ, например из задания 5*) с использованием неблокирующего способа выполнения операций передачи данных. Оцените необходимое количество вычислительных операций для того, чтобы полностью совместить передачу данных и вычисления. *(2 балла)*
8. Модифицируйте программу из задания 3, используя операцию одновременного выполнения передачи и приема данных. Сравните результаты вычислительных экспериментов. *(1 балл)*
9. Разработайте реализации коллективных операций (`Broadcast`, `Reduce`, `Scatter`, `Gather`, `AllScatter`, `AllGather`) при помощи парных обменов между процессами. Выполните вычислительные эксперименты и сравните времена выполнения разработанных программ и функций MPI для коллективных операций. *(2 балла)*
10. Разработайте программу для конструирования производных типов данных, для использования упаковки и распаковки производных типов данных, проведите эксперименты, сравните результаты этих вариантов при передаче данных. *(2 балла)*
11. Разработайте программу для представления множества процессов в виде прямоугольной решетки. Создайте коммуникаторы для каждой строки и каждого столбца процессов. Выполните коллективную операцию для всех процессов и для одного из созданных коммуникаторов. Сравните время выполнения операции. *(1 балл)*
12. Разработайте программу для создания декартовой топологии, топологии тора, топологии графа, топологии звезды. *(1 балл)*

Итого: до 22 балла за MPI

Суммарно за задания по обоим модулям возможно набрать до 38 баллов

Оценки:

A – 30 баллов

B – 26 баллов

C – 22 балла

D – 18 баллов

E – 14 баллов

()