

Первое практическое задание по OpenMP: параллельная генерация вектора случайных чисел

Введение

В этом задании вам необходимо реализовать первое свое параллельное приложение, которое будет генерировать случайные числа в нескольких потоках. Перед началом выполнения данного задания мы вам настоятельно рекомендуем просмотреть все видео второго модуля нашего курса, если вы еще этого не сделали, и пройти тест второго модуля.

Для начала выполнения данного задания вам необходимо скачать архив со стартовым проектом и распаковать его в директорию, в который вы будете выполнять практическое задание. Для выполнения данного задания вам понадобится Microsoft Visual Studio 2015. Инструкцию по установке данного программного продукта вы можете найти в вводном модуле нашего курса.

Файлы входящие в задание

Папки:

Ex1 – основная папка всего решения;

random_vector – папка, содержащая исходные файлы.

Файлы с исходным кодом:

main.cpp – файл, содержащий основную функцию main;

vec_random.cpp – файл, содержащий заготовку функции vec_random.

vec_random_norm.cpp – файл, содержащий заготовку функции vec_random_norm.

Заголовочные файлы:

ex_function.h – заголовочный файл, содержащий описание основных функций vec_random и vec_random_norm.

1. Генерация случайных чисел несколькими потоками

В этом задании необходимо реализовать генерацию случайных чисел с использованием нескольких потоков. При выполнении этого задания нормализовать случайные числа не надо. Данную задачу необходимо реализовать в функции vec_random. Заготовка данной функции есть в файле vec_random.cpp. Функция принимает в качестве параметра количество чисел, которые необходимо сгенерировать, и логическую переменную, которая определяет, нормализованный или ненормализованный вектор случайных чисел должен быть создан. Данная функция возвращает указатель на массив целых чисел, сгенерированных случайным образом. Каждый поток должен сгенерировать одно число и записать его в массив под номером, соответствующим номеру потока. **Внимание!** Для успешного прохождения проверки выполнения задания в параллельной области обязательно должна вызываться функция Submit_test.

Рекомендации

- Для генерации случайных чисел вы можете воспользоваться функцией rand, а для инициализации генератора псевдослучайных чисел воспользуйтесь функцией srand.
- Для того чтобы узнать номер потока, воспользуйтесь функцией omp_get_thread_num.

- Не забудьте указать количество потоков, которые должны выполнять параллельную область, оно соответствует количеству генерируемых случайных чисел.
- Не забудьте включить поддержку openmp в свойствах проекта (C/C++ – Language – Open MP Support).
- Если вы не хотите, чтобы после завершения программы сразу закрывалось консольное окно, то запускайте приложение без отладки (Start Without Debugging или горячие клавиши Ctrl+F5).

После того как выполните задание и проверите корректность работы, раскомментируйте вызов функции `Submit_Part1()` в файле `main.cpp`, перекомпилируйте программу и запустите ее. После завершения программы в папке `ex1` будет создан файл `submit1.bin`, который надо загрузить на платформу Coursera для проверки.

2. Нормализация случайных чисел

В этом задании необходимо реализовать минимаксную нормализацию случайных целых чисел с использованием нескольких потоков.

2.1. Нормализация

Нормализация позволяет преобразовать диапазон изменения значений числового признака в другой диапазон, более удобный для применения к данным тех или иных аналитических алгоритмов, а также согласовать диапазоны изменений различных признаков. Часто имеющийся диапазон данных «сжимается» в интервал $[0; 1]$. Особенно важно произвести правильную нормализацию данных в алгоритмах Data Mining, которые основаны на измерении расстояния между векторами объектов в многомерном пространстве признаков (например, в кластеризации).

Минимаксная нормализация

$$V'(i) = \frac{V(i) - \min(V(i))}{\max(V(i)) - \min(V(i))},$$

где $V(i)$ – исходное значение, $V'(i)$ – нормализованное значение

2.2. Реализация

Данную задачу необходимо реализовать в функции `vec_random_norm`. Заготовка данной функции есть в файле `vec_random_norm.cpp`. Функция принимает в качестве параметра указатель на массив чисел типа `float` и размерность массива.

Сначала необходимо найти максимум и минимум в массиве, это должен сделать только один поток. Затем все потоки должны вычислить нормализованное значение для элемента с номером, соответствующим номеру потока.

Рекомендации

- Для того чтобы узнать номер потока, воспользуйтесь функцией `omp_get_thread_num`.
- Не забывайте, что у вас функция `vec_random_norm` вызывается внутри параллельной секции, поэтому все вновь объявленные переменные будут приватными для каждого потока.

- Для того чтобы выполнить поиск минимума и максимума только в одном потоке, воспользуйтесь директивой `single`, а также опцией `copyprivate`.

После того как выполните задание и проверите корректность работы, раскомментируйте вызов функции `Submit_Part2()` в файле `main.cpp`, перекомпилируйте программу и запустите ее. После завершения программы в папке `ex1` будет создан файл `submit2.bin`, который надо загрузить на платформу Coursera для проверки.

Оценка заданий

После выполнения всех частей убедитесь, что вы загрузили все файлы в систему проверки заданий и все задания у вас были засчитаны.

В таблице 1 представлены названия заданий и баллы за каждую часть.

Таблица 1

Название части	Баллы
Генерация случайных чисел несколькими потоками	50
Нормализация случайных чисел	50
Итого	100