Доскоч Роман 13 группа 3 курс ТП Домашнее задание от 12.11.2021

Исследовать функцию std::generate

template< class ExecutionPolicy, class ForwardIt, class Generator >  
void generate( ExecutionPolicy&& policy, ForwardIt first, ForwardIt last, Generator g );

Код

#include <execution>

#include <algorithm>

#include <iostream>

#include <vector>

#include <chrono>

#include <concepts>

enum class Policy { seq, par, par\_unseq };

// Функция выбора метода выполнения : последовательно или параллельно.

template<class F>

auto maybe\_parallel(F f, const Policy p) {

switch (p) {

case Policy::seq: return f(std::execution::seq);

case Policy::par: return f(std::execution::par);

default: return f(std::execution::par\_unseq);

}

}

// Обертка функции std::generate() с параметром Policy.

auto policy\_generate(std::vector<int>& vec, Policy p) {

return maybe\_parallel(

[&](auto& pol)

{

return std::generate(pol, vec.begin(), vec.end(),

[x = 2]()mutable

{

return x = pow(cos(x) \* sin(x),2);

});

}, p);

}

// Функция тестирования.

auto generate\_test(const Policy policy, int n) {

std::vector<int>v(n);

auto start = std::chrono::high\_resolution\_clock::now();

policy\_generate(v, policy);

auto stop = std::chrono::high\_resolution\_clock::now();

return std::chrono::duration\_cast<std::chrono::microseconds>(stop - start).count() / 1e6;

}

int main() {

setlocale(LC\_ALL, "ru");

for (auto n : { 10000, 1000000, 1000000000 }) {

auto time = generate\_test(Policy::seq, n);

auto par\_time = generate\_test(Policy::par, n);

std::cout << "Размерность " << n << ": \n"

<< "Линейный метод: " << time << std::endl

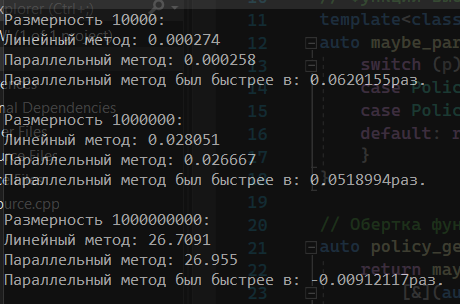
<< "Параллельный метод: " << par\_time << std::endl

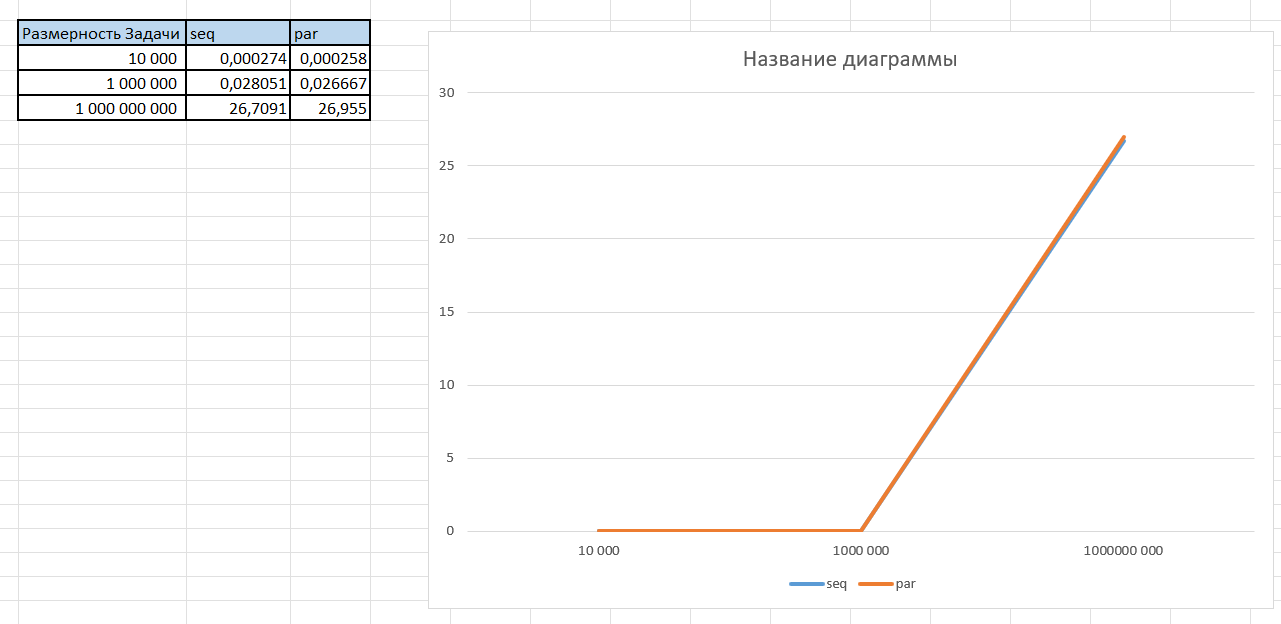
<< "Параллельный метод был быстрее в: "

<< (time - par\_time) / par\_time << "раз.\n\n";

}

}





Вывод: видно, что использование параллельной реализации метода generate, мало чем отличается от последовательного подхода. Объясняется это тем, что алгоритм генерации сложен для распараллеливания. Время так же зависит от функции, на которой будет генерироваться массив. Ее сложность прямой показатель времени выполнения.