Параллельная реализация игры «Жизнь» средствами OpenMP
ЛАБОРАТОРНАЯ РАБОТА

http://www.kimrt.ru
Онлайн-курс
Суперкомпьютерные технологии в задачах моделирования

Один из вариантов параллельного алгоритма игра Жизнь приведен в листинге 1. Переработайте этот алгоритм на случай периодических граничных условий. Кроме того, подумайте, как оптимизировать алгоритм, например, хранить информацию об игровом поле в одномерном массиве. Тогда, чтобы перебрать все ячейки в поле не нужно будет использовать вложенные циклы. Проход будет осуществляться одним циклом, в котором можно будет распараллелить итерации по потокам. У пользователя должна быть возможность задавать размеры решетки. Промежуточные данные о решетке должны сохраняться в файлы через некоторые периоды, состоящие из N временных шагов. Задание заключается в исследовании зависимости времени расчета от различных параметров и в сравнении со временем расчета последовательной версии алгоритма (закомментировать #pragma). Нужно рассмотреть такие параметры как число временных шагов (200, **1000**, 3000), количество строк в матрице (10^3, **10^4**, $3*10^4$) и число используемых потоков в параллельной версии (2, **3** и 4). Варьируется поочередно только один параметр при фиксированных значениях остальных (выделены полужирным). Каждое испытание повторяется 5 раз и усредняется в Excel. Необходимо заполнить таблицы, построить соответствующие графики и написать краткий отчет с выводом.

Листинг 1. Игра Жизнь на языке C++ с использованием OpenMP

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <ctime>
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <random>
#include <omp.h>
#include <vector>
#define SIZEX 50
#define SIZEY 10000
#define xPlus1 (x+1)%SIZEX
#define xMinus1 (x-1+SIZEX)%SIZEX
```

```
#define yPlus1 (y+1)%SIZEY
#define yMinus1 (y-1+SIZEY)%SIZEY
using namespace std;

int main ()
{
	omp_set_num_threads(4);
	setlocale(LC_ALL,"Russian");
	std::random_device rd;
	 std::mt19937 gen(rd());
	 std::uniform_int_distribution<> dis(1, 10000);
	//int tmp[SIZEX][SIZEY], world[SIZEX][SIZEY];
	std::vector<std::vector<int> > tmp(SIZEX,std::vector<int>(SIZEY));
	std::vector<std::vector<int> > world(SIZEX,std::vector<int>(SIZEY));
	 int x, y, n=0;//координаты и количество соседей
	FILE * f;
	f = fopen("test.txt","wt");
	//начальная генерация
	 for (y=0;y<=SIZEY-1;y++)
		for (x=0;x<=SIZEX-1;x++){
			int num = dis(gen);
			world [x][y] = num % 2;
		}
   int d, i;
   printf("Количество поколений: ");
   scanf("%i", &d);
	clock_t jobTime = clock();//время работы программы
   for (i=1;i<=d;i++)//цикл по временным шагам
   {
      system("cls");
		//запоминаем предыдущее состояние матрицы
      for (y=0;y<=SIZEY-1;y++)
			for (x=0;x<=SIZEX-1;x++) tmp[x][y] = world[x][y];
//Перебираем все ячейки и при необходимости меняем значения
for (x=0;x<=SIZEX-1;x++)
      {
         #pragma omp parallel for firstprivate(n)
        for (y=0;y<=SIZEY-1;y++)
         {
            if (tmp[x][y] == 0)
            {
               if (tmp[xMinus1][yMinus1] == 1) n++;
               if (tmp[x][yMinus1] == 1) n++;
               if (tmp[xPlus1][yMinus1] == 1) n++;
               if (tmp[xPlus1][y] == 1) n++;
               if (tmp[xPlus1][yPlus1] == 1) n++;
               if (tmp[x][yPlus1] == 1) n++;
               if (tmp[xMinus1][yPlus1] == 1) n++;
               if (tmp[xMinus1][y] == 1) n++;
               if (n == 3) world[x][y] = 1;
               n = 0;
            } else
            {
```

```
                if (tmp[xMinus1][yMinus1] == 1) n++;
                if (tmp[x][yMinus1] == 1) n++;
                if (tmp[xPlus1][yMinus1] == 1) n++;
                if (tmp[xPlus1][y] == 1) n++;
                if (tmp[xPlus1][yPlus1] == 1) n++;
                if (tmp[x][yPlus1] == 1) n++;
                if (tmp[xMinus1][yPlus1] == 1) n++;
                if (tmp[xMinus1][y] == 1) n++;
                if ((n == 2) || (n == 3))
                {
                   world [x][y] = 1;
                   n = 0;
                } else
                {
                   world [x][y] = 0;
                   n = 0;
                }
             }
          }
       }
    }
for (y=0;y<=SIZEY-1;y++) {
                      for (x=0;x<=SIZEX-1;x++) fprintf( f, "%d ", world[x][y] );
                      fprintf( f, "\n");
}
fclose(f);
jobTime = clock() - jobTime;//время в тактах микропроцессора
cout << "Execution time: " << (double)jobTime / CLOCKS_PER_SEC
                                << " seconds" << endl;
system("pause");
 return 0;
}
```

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. https://neerc.ifmo.ru/wiki/index.php?title=%D0%98%D0%B3%D1%80%D0%B0_%C2%AB%D0%96%D0%B8%D0%B7%D0%BD%D1%8C%C2%BB
2. https://pikabu.ru/story/programmiruem_zhizn_6389491
3. http://knoppix.ru/sentinel/021017.html

4. http://rosettacode.org/wiki/Conway%27s_Game_of_Life#Simple_Without_Classes
5. http://rosettacode.org/wiki/Conway%27s_Game_of_Life