

康威生命游戏实验报告

CPU 型号：Intel i5-6200U

支持指令集：SSE SSE2 SSE3 等

选用指令集：SSE2

编译选项：使用 Visual Studio 2017,生成编译。

串行主要思路(MyConwayGame)：使用两张全局变量的矩阵，对于上一代中的每一个节点统计周围八个点的状态，更新在下一代的表中；迭代时将下一代表作为上一代表

并行主要思路(MyConwayGameSIMD)：在串行的基础上，每次处理四个数据单元，使用__m128i 存储四个 32 位整形数据

并行主要指令： __mm_loadu_si128 数据载入存储器
__mm_add_epi32 两个 32 位 x4 向量寄存器相加

实验数据：input_50x100 与对应的 output_50x100(_simd)分别在对应程序的.exe 相同目录下，通过命令行输入控制迭代次数。行数列数，输入输出文件名已在程序中默认给出。

运行&成果：

串行版本：

```
C:\Users\Puzzle Yao\Desktop\MyConwayGame>cd Debug
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 100
The program takes about 0.101 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 200
The program takes about 0.199 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 500
The program takes about 0.486 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 1000
The program takes about 0.97 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 2500
The program takes about 2.432 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 5000
The program takes about 4.848 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 10000
The program takes about 9.712 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 20000
The program takes about 19.397 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGame\Debug>MyConwayGame.exe 50000
The program takes about 48.431 seconds!
```

并行版本：

```
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 100
The program takes about 0.007 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 200
The program takes about 0.013 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 500
The program takes about 0.04 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 1000
The program takes about 0.061 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 2500
The program takes about 0.15 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 5000
The program takes about 0.296 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 10000
The program takes about 0.605 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 20000
The program takes about 1.201 seconds!
C:\Users\Puzzle Yao\Desktop\MyConwayGameSIMD\Debug>MyConwayGameSIMD.exe 50000
The program takes about 3.042 seconds!
```

分析表：

| 迭代次数 | 串行时间 (s) | 并行时间 (s) | 加速比 |
|-------|----------|----------|-------|
| 100 | 0.101 | 0.007 | 14.42 |
| 200 | 0.199 | 0.013 | 15.31 |
| 500 | 0.486 | 0.04 | 12.15 |
| 1000 | 0.97 | 0.061 | 15.9 |
| 2500 | 2.432 | 0.15 | 16.21 |
| 5000 | 4.848 | 0.296 | 16.38 |
| 10000 | 9.712 | 0.605 | 16.05 |
| 20000 | 19.397 | 1.201 | 16.15 |
| 50000 | 48.431 | 3.042 | 15.92 |

效率分析：

按照并行的核心部分，加速比应该接近于 4，但是在多次测试保证结果相同的情况下，发现加速比在稳定后可以达到接近于 16。原因之一肯定是一次处理 4 个数据，这也应该是并行的核心。猜测另一个原因应该是因为对于并行减少了边界检测（checkBound()）和串行的状态统计（countLives()）的执行，采用直接相加状态的方式。最后的加速比可以给出初步结论会稳定在 16，体现了并行的高效率。