



Universidade de Aveiro

Computação Paralela — 2019/2020

Recurso Prático do Módulo MPI

A ser entregue (via email) até às 18 horas de 17 de julho

Paralelização MPI do jogo da vida de Conway

Neste trabalho vai escrever um programa paralelizado por MPI do jogo da vida de Conway. O universo deste autómato celular é uma rede quadrada de células. Cada célula tem 8 vizinhos adjacentes: 4 ortogonais e 4 diagonais. Cada célula contém um organismo que pode estar vivo ou morto. O universo evolui de geração para geração. As regras genéticas são as seguintes:

Sobrevivência: Se o organismo de uma célula estiver vivo e tiver 2 ou 3 vizinhos vivos, ele vai continuar vivo na próxima geração.

Morte: Se o organismo de uma célula estiver vivo e tiver menos que 2 ou mais que 3 vizinhos vivos, não vai haver um organismo vivo nessa célula na próxima geração.

Nascimento: Se numa geração não houver um organismo vivo na célula, mas se existirem **exatamente** 3 vizinhos vivos, então na geração seguinte vai haver um organismo vivo nessa célula.

Note que todas as mortes e nascimentos ocorrem em simultâneo, de uma geração para outra.

- Para simplificar, escreva um programa que corra em exatamente 4 processos. O universo deverá ter 100×100 células. Cada processo deverá ser responsável por atualizar o estado de um sub-universo de 50×50 células.
- Use simplesmente um número inteiro para representar o estado do organismo de cada célula, 0 para morto, 1 para vivo.
- Use condições fronteira periódicas nas duas direções. É extremamente fácil definir isto quando se cria o comunicador cartesiano e o resto do código até fica mais simples.
- Crie uma configuração inicial aleatória. O código seguinte mostra como criar uma sequência aleatória de zeros e uns.

```
#include <stdio.h>
#include <stdlib.h>
#define NUM 50
int main()
{
    int vida[NUM];
    double frac = 0.40; /* fracao de valores 1 */
    srand(87145); /* Inicializa seed. Usar so' uma vez. */
    for (int i=0; i < NUM; ++i){
        if(random() < (frac * RAND_MAX)) vida[i] = 1;
        else vida[i] = 0;
        printf("%d \n", vida[i]);
    }
}
```

- Corra o programa por alguns milhares de gerações.
- Use as funções de I/O paralelo do MPI para escrever um ficheiro binário com o estado final do universo de 100×100 células.