

# Instituto Tecnológico de Aeronáutica

## CE-265: Processamento Paralelo

### Exercício 01: Jogo da Vida Sequencial no SDumont

Bruno Benjamim Bertucci - Turma 23.2

## 1 Introdução

Para fins de familiarização com a utilização do supercomputador SDumont, foi proposta a execução de um programa denominado Jogo da Vida de forma sequencial. O jogo é constituído de um tabuleiro representado por uma matriz quadrada. Cada célula do tabuleiro pode assumir dois valores, representados como viva ou morta. A cada iteração, os estados de todas as células são atualizados, representando uma geração do jogo. Essa atualização se dá segundo as seguintes regras:

- Células vivas com menos de 2 vizinhas vivas morrem por abandono.
- Células vivas com mais de 3 vizinhas vivas morrem de superpopulação.
- Células mortas com exatamente 3 vizinhas vivas tornam-se vivas.
- As demais células mantêm seu estado anterior.

## 2 Execução do jogo para um tabuleiro de tamanho 6

A implementação do jogo, feita utilizando a linguagem C, compreende dois testes, além de *scripts* na linguagem *bash* para a submissão deles para execução no SDumont. Um desses testes considera somente um tabuleiro de tamanho 6, e imprime na tela representações deste a cada geração, tendo como objetivo demonstrar o correto funcionamento da implementação das regras do jogo. A saída desse teste está disponível no Anexo A.

## 3 Complexidade do programa para um tamanho N

Na implementação do jogo, o laço principal que executa as gerações é percorrido  $2(N - 3)$  vezes, onde  $N$  é a dimensão do tabuleiro, sendo que, a cada execução desse laço, duas gerações são executadas. Ou seja, no total, ocorrem  $4(N - 3)$  gerações do jogo antes do término do programa.

Para cada geração processada, há um laço para percorrer cada linha da matriz e, dentro deste, há um segundo laço para percorrer cada elemento da linha, de tal forma que cada elemento da matriz é percorrido. Dentro desses dois laços, estão implementadas as regras para atualização dos estados das células, calculando a quantidade de vizinhos vivos seguidos por uma série de condicionais executando tais regras. Portanto, o bloco de código dentro desses laços tem complexidade assintótica de  $O(1)$ , sendo executado  $N^2$  vezes para cada geração por conta dos laços. Assim, a execução de uma geração tem complexidade de  $O(N^2)$ .

Ainda, ocorrem um total de  $4(N - 3)$  gerações, o que permite concluir que essa etapa do programa tem uma complexidade de  $O(N^3)$ .

## 4 Comparação do teste do jogo para vários tamanhos com a complexidade calculada

O segundo teste implementado executa o jogo repetidas vezes, variando o tamanho do tabuleiro em potências de 2, desde 8 até 512.

O tempo de execução de cada etapa do programa, incluindo a inicialização do tabuleiro, o processamento do laço que avança as gerações, e o término da execução, é calculado e exibido para cada tamanho de tabuleiro testado. O intuito desse teste é mostrar a variação do tempo de execução do jogo, especialmente na segunda etapa, com a variação da dimensão do tabuleiro.

Considerando a complexidade assintótica de  $O(N^3)$  encontrada na seção anterior para essa etapa, pode-se modelar o tempo de execução esperado para um tamanho qualquer  $N$  segundo a expressão:  $T(N) = k \cdot N^3$ , onde  $k$  é uma constante. Para a determinação do seu valor, foi tomado o tempo de execução para o maior tamanho de tabuleiro testado, nesse caso, de 512, pois este se aproxima mais do tempo de execução assintótico. Para o jogo com tabuleiro de dimensão 512, o tempo gasto foi de 11.593071 segundos. Portanto, encontrou-se o valor da constante  $k = 8,64 \cdot 10^{-8}$ . Portanto, segundo o modelo teórico adotado, o tempo de execução deve seguir a Equação 1.

$$T(N) = 8,64 \cdot 10^{-8} \cdot N \quad (1)$$

A saída produzida pela execução desse segundo teste encontra-se no Anexo B, a partir da qual os tempos de processamento da etapa de avanço das gerações foram separados para análise na Tabela 1, junto com os tempos teóricos calculados utilizando o modelo de complexidade assintótica descrito acima.

Tamanho	Tempo de execução (s)	Tempo teórico (s)
8	0,000034	0,000044
16	0,00031	0,00035
32	0,00264	0,00283
64	0,022	0,023
128	0,178	0,181
256	1,44	1,45
512	11,6	11,6

Table 1: Tempos de execução e tempos teóricos para diferentes tamanhos de tabuleiro diferentes.

Observou-se que a complexidade calculada é capaz de prever, com boa acurácia, o tempo de execução para a maioria dos tamanhos de tabuleiro testados, com exceção do teste com tamanho 8. Este apresentou um erro consideravelmente grande, o que pode ser explicado pelo valor pequeno de  $N$  para esse caso levar o tempo de execução desse teste a se distanciar do modelo assintótico. Para os demais casos, porém, o tempo teórico calculado se aproxima do tempo observado na execução, sustentando a complexidade assintótica do tipo  $O(N^3)$  da execução do jogo.

## Anexo A – Etapas do Jogo da Vida para um tabuleiro de dimensão 6

Inicial; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6

=====

.X....

..X...

XXX...

.....

.....

.....

=====

Iter 001; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6

=====

.....

X.X...

.XX...

.X....

.....

.....

=====

Iter 002; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6

=====

.....

..X...

X.X...

.XX...

.....

.....

=====

Iter 003; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6

=====

.....

.X....

..XX..

.XX...

.....

.....

=====

Iter 004; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6

=====

.....

..X...

...X..

.XXX..

.....

.....

=====

Iter 005; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6

=====

.....

.....

.X.X..

..XX..

..X...

.....

=====

```

Iter 006; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....
.....
...X..
.X.X..
..XX..
.....
=====
Iter 007; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....
.....
..X...
...XX.
..XX..
.....
=====
Iter 008; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....
.....
...X..
....X.
..XXX.
.....
=====
Iter 009; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....
.....
.....
..X.X.
...XX.
...X..
=====
Iter 010; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....
.....
.....
....X.
..X.X.
...XX.
=====
Iter 011; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....
.....
.....
...X..
....XX
...XX.
=====
Iter 012; Dump posicoes [1:6, 1:6] de tabuleiro 6 x 6
=====
.....

```

```

.....
.....
....X.
.....X
...XXX
=====
**RESULTADO CORRETO**

```

## Anexo B – Tempos de execução do Jogo da Vida para vários tamanhos de tabuleiro

```

**RESULTADO CORRETO**
tam=8; tempos: init=0.000045, comp=0.000034, fim=0.000019, tot=0.000098
**RESULTADO CORRETO**
tam=16; tempos: init=0.000001, comp=0.000310, fim=0.000001, tot=0.000312
**RESULTADO CORRETO**
tam=32; tempos: init=0.000009, comp=0.002637, fim=0.000004, tot=0.002650
**RESULTADO CORRETO**
tam=64; tempos: init=0.000028, comp=0.021934, fim=0.000014, tot=0.021976
**RESULTADO CORRETO**
tam=128; tempos: init=0.000107, comp=0.178352, fim=0.000055, tot=0.178514
**RESULTADO CORRETO**
tam=256; tempos: init=0.000465, comp=1.442253, fim=0.000214, tot=1.442932
**RESULTADO CORRETO**
tam=512; tempos: init=0.002100, comp=11.593071, fim=0.000851, tot=11.596022

```