

CSC 64 Exercício 01

CE 265 Exercício 01

1. Finalidade

Familiarizar-se com a versão sequencial do Jogo da Vida e com a compilação e execução de programas no SDumont. Demonstrar familiaridade com complexidade assintótica do tempo de execução.

2. Jogo da Vida

O Jogo da Vida¹, criado por John H. Conway, utiliza um autômato celular para simular gerações sucessivas de uma sociedade de organismos vivos. É composto por um tabuleiro bidimensional de células idênticas. Cada célula tem exatamente oito células vizinhas (todas as células que compartilham, com a célula original, uma aresta ou um vértice). Cada célula está em um de dois estados: viva ou morta. Uma geração da sociedade é representada pelo conjunto dos estados das células do tabuleiro. Sociedades evoluem de uma geração para a próxima aplicando simultaneamente, a todas as células do tabuleiro exceto as bordas, regras que estabelecem o próximo estado de cada célula. As bordas permanecem no estado inicial. As regras são:

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

3. Arquivos Fornecidos

Obtenha o *tarball* `JogoDaVidaSequencialNoSDumont.tgz` nesta atividade. Transfira o *tarball* (sem desempacotar) para sua conta no Santos Dumont. Mova o arquivo para o diretório `$$SCRATCH`.

Desempacote utilizando `"tar -xvzf JogoDaVidaSequencialNoSDumont.tgz"`. Cria-se o diretório `Exercicio01`, contendo três arquivos fonte (`wall_time.c`, `ModVida.c` e `Tempo.c`), `Makefile` e *script* de execução preparado para o sistema de submissão *batch* do Santos Dumont (`Xmit.sh`).

Os arquivos fonte implantam o Jogo da Vida em C sobre um tabuleiro finito, $TAM \times TAM$, orlado por células eternamente mortas. O tabuleiro é representado por um *array* unidimensional de inteiros alocado com tamanho $(TAM+2) \times (TAM+2)$. A posição (i, j) do tabuleiro 2D, com $i, j = 1, \dots, TAM$ é mapeada na posição do *array* 1D pelo macro `ind2d(i,j)`. As demais posições do *array* são as bordas que não são alteradas. Gerações sucessivas do tabuleiro são representadas por dois *arrays* denominados *Tabulln* e *TabulOut*. Células vivas são representadas pelo inteiro 1 e células mortas pelo inteiro 0.

O arquivo `ModVida.c` contém um procedimento que avança uma geração do tabuleiro (`UmaVida`), um procedimento que imprime trecho selecionado do tabuleiro (`DumpTabul`), um procedimento que inicializa tabuleiros (`InitTabul`), colocando um "veleiro" no canto superior esquerdo do tabuleiro e um procedimento (`Correto`) que verifica se o "veleiro" está no canto inferior direito ao final da computação, após computar $4(TAM - 1)$ gerações. Estude os programas fonte, particularmente o procedimento `UmaVida`. Os procedimentos `DumpTabul` e `Correto` não são utilizados neste exercício, mas serão utilizados em exercícios futuros.

O arquivo `Tempo.c` é o programa principal que inicializa os tabuleiros e avança o número necessário de gerações para mover o "veleiro" do início ao fim da diagonal do tabuleiro. Observe

como o tempo de execução é medido e que medimos apenas tempo do avanço das gerações do tabuleiro.

O script “Xmit.sh” submete o executável à fila de execução adequada no SDumont. Para executá-lo, basta digitar “./Xmit.sh” na linha de comando no diretório onde se encontra.

Acompanhe a execução (rápida) pelo comando “squeue”. Entenda o script, que gera arquivo “ssub.sh” com os comandos SLURM e submete esse arquivo ao pelo comando “sbatch”.

4. Seu Trabalho

Compile os programas fonte utilizando make. Resulta arquivos executável “Tempo.exe”.

O script “Xmit.sh” submete o executável Tempo.exe variando o tamanho do tabuleiro de 2^3 a 2^9 em potências de 2 e medindo o tempo de execução em função do tamanho do tabuleiro. Produz o arquivo de saída “OutTempo”, que reporta os tempos medidos.

Após instalar e compilar, faça as seguintes atividades:

1. Execute Xmit.sh. Reserve o arquivo de saída;
2. Componha relatório do exercício com os seguintes itens:
 - a. A complexidade assintótica teórica do tempo de execução (“big O”) das $4(N-3)$ iterações do Jogo da Vida em tabuleiro de tamanho N , desprezando as bordas. Para tanto, basta examinar o texto do programa. A complexidade assintótica deve ser expressa em função de N ;
 - b. Demonstre que os tempos de execução contidos no arquivo OutTempo (tempos rotulados “comp”) resultantes da tarefa 1 seguem a complexidade calculada no item anterior. Cuidado com a demonstração.

Gere versão em pdf do relatório e entregue a versão em pdf até a meia noite anterior à próxima aula, como atividade realizada no site do curso.

[1] – Martin Gardner, “Mathematical Games – The fantastic combinations of John Conway’s new solitaire game life”, Scientific American 223, Oct. 1970, pp 120-123, or http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm