

## CE 265 2011

### Exercício 2

O Jogo da Vida<sup>1</sup>, criado por John H. Conway, utiliza um autômato celular para simular gerações sucessivas de uma sociedade de organismos vivos.

É composto por um tabuleiro bi-dimensional, infinito em qualquer direção, de células quadradas 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, regras que estabelecem o próximo estado de cada célula. 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.

Sua tarefa é paralelizar OpenMP a versão sequencial fornecida, utilizando o CESUP ou a Cray.

Utilize um dos “tarball” JogoDaVidaNoCESUP.tgz ou JogoDaVidaNoCrow.tgz, armazenados no site, diretório Semana2, que implanta o Jogo da Vida em Fortran 90 sobre um tabuleiro finito,  $N \times N$ , orlado por células eternamente mortas. Transfira para sua conta na máquina selecionada e desempacote o “tarball” com o comando `tar -xzf JogoDaVidaNoXXXX.tgz` em diretório de sua escolha.

Resultam três arquivos fonte (ModVida.f90, Funcionamento.f90 e TempoExec.f90), Makefile e dois “scripts” de execução preparados para o site (XmitFunciona.sh e XmitTempo.sh).

O arquivo ModVida.f90 contém um procedimento que avança uma geração do tabuleiro (UmaVida), um procedimento que imprime trecho selecionado do tabuleiro (DumpTabuleiro), um procedimento que inicializa tabuleiros (InitTabuleiros), colocando um “veleiro” no canto superior esquerdo do tabuleiro e um último procedimento (Correto) que verifica se o “veleiro” está no canto inferior esquerdo ao final da computação. Estude os fontes, particularmente o procedimento UmaVida.

Os arquivos Funcionamento.f90 e TempoExec.f90 são programas principais que invocam os procedimentos de ModVida.f90 o número de gerações necessárias para mover o “veleiro” pela diagonal do tabuleiro. Divergem no tamanho do tabuleiro, na impressão (ou não) do tabuleiro e na medição (ou não) do tempo de execução. Os dois programas verificam se o resultado final é correto. Observe os laços principais desses dois programas.

O programa “Funcionamento” utiliza  $N=6$ . Imprime o tabuleiro inicial e ao fim de cada iteração.

O programa “TempoExec” mede o tempo de execução para diversos valores de  $N$ , sem imprimir o tabuleiro. Executa  $4(N-3)$  iterações. Imprime  $N$  e o tempo medido (em segundos). Observe que o tempo é medido por invocações de `SYSTEM_CLOCK`, procedimento intrínseco de Fortran 90. Entenda como isso é feito.

Compile os programas fonte utilizando “make”. Resultam dois arquivos executáveis: “Funciona” e “Tempo”.

Os scripts “XmitFunciona.sh” e “XmitTenpo.,sh” submetem os respectivos executáveis à fila adequada da máquina selecionada. Requerem, como único argumento, a quantidade de threads desejada na execução. Embora os programas fonte sejam sequenciais, já estão preparados para o paralelismo OpenMP a inserir, assim como os scripts. Produzem os arquivos de saída “Funciona\_X\_Threads.out” e “Tempo\_X\_Threads.out”, onde *X* é o número de threads utilizado.

Por exemplo, para executar o script “XmitFunciona.sh” com quatro threads, use “./XmitFunciona.sh 4”. Estude os scripts.

Após instalação, execute as seguintes tarefas:

1. Demonstre a correção da instalação, executando o script XmitFunciona.sh com uma única thread e anexando a saída ao seu relatório;
2. Paralelize OpenMP o procedimento de ModVida.f90 que executa uma iteração do Jogo da Vida. Certifique-se da correção do seu trabalho executando XmitFunciona.exe com duas ou mais threads. Anexe o ModVida.f90 assim paralelizado e o arquivo de saída da execução ao seu relatório.
3. Meça o tempo de execução de sua versão paralela de 1 a 4 threads, executando “XmitTempo.sh” com 1 a 4 threads. Calcule o speed-up e entregue tabela com os tempos de execução e com os speed-ups. Analise os resultados, bons ou ruins. Não se preocupe com os ganhos (ou perdas) atingidos. Concentre-se na correção e no entendimento completo do experimento.

Entregue relatório até a véspera da próxima aula (14 de março), pelo TIDIA.

[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](http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm)