Trabalho prático 2 - Simulação de um Ecossistema

Rafael Sachetto

29 de setembro de 2021

1 Descrição

O problema a ser resolvido consiste na simulação de um ecossistema simples, habitado por duas espécies de animais, raposas e coelhos, cuja existência ao longo de gerações é mutuamente dependente.

A simulação deve ser feita usando uma matriz de L linhas e R colunas (o mundo do ecossistema), onde cada célula representa o espaço do ecossistema no qual as raposas e os coelhos se movem. Em cada geração, cada célula pode estar livre ou ocupada. Uma célula pode ser ocupada por uma raposa, um coelho ou uma rocha. O mundo é inicializado com uma determinada população de raposas, coelhos e rochas que, ao longo das gerações, evoluem de acordo com um conjunto fixo de regras. A simulação começa com um dado ecossistema inicial e constrói as sucessivas gerações de raposas e coelhos até que um determinado número de gerações (N_GEN) seja alcançado.

1.1 Regras para os coelhos

- Os coelhos podem se mover horizontalmente ou verticalmente (ou seja, norte, leste, sul e oeste), mas não na diagonal.
- Em cada geração, os coelhos tentam se mover para uma célula adjacente vazia. Se houver muitas células adjacentes vazias, elas escolherão uma de acordo com um critério comum para selecionar células adjacentes (veja abaixo). Se não houver célula adjacente vazia, eles permanecerão no mesmo local.
- Os coelhos podem procriar sempre que GEN_PROC_COELHOS gerações se passaram desde que nasceram ou desde a última procriação. Sempre que um coelho atinge essa idade (para procriar) e faz um movimento,

deixa em sua última posição um novo coelho e a idade de procriação de ambos os coelhos é definida como zero.

1.2 Regras para as raposas

- As raposas podem se mover horizontalmente ou verticalmente, mas não na diagonal.
- Em cada geração, as raposas tentam comer um coelho movendo-se para uma célula adjacente que é ocupada por um coelho. Se houver muitas células adjacentes ocupadas com coelhos, eles escolherão uma de acordo com os critérios comuns para selecionar células adjacentes (veja abaixo). Se não houver célula adjacente ocupada por um coelho, as raposas tentam se mover para uma célula adjacente vazia, seguindo os mesmos critérios. Se não houver uma célula adjacente ocupada com um coelho ou vazia, elas permanecerão no mesmo local.
- As raposas morrem de fome sempre que GEN_COMIDA_RAPOSAS gerações se passaram desde que nasceram ou desde a última vez que comeram um coelho. As raposas morrem depois de não encontrar um coelho para comer e antes de tentar se mover para uma célula adjacente vazia.
- As raposas podem procriar sempre que GEN_PROC_RAPOSAS gerações se passaram desde que nasceram ou desde a última vez que foram criadas. Sempre que uma raposa atinge essa idade (para procriar) e faz um movimento, deixa em sua última posição uma nova raposa e a idade de procriação de ambas as raposas é definida como zero.

1.3 Regras para as rochas

- As rochas não se mexem e nenhum animal pode ocupar seu espaço.

1.4 Critérios para seleção de células adjacentes

- Seguindo o sentido horário comece a numerar, à partir de 0, as P possíveis células para onde uma raposa/coelho pode se mover (células adjacentes ao norte, leste, sul e oeste).
- Seja G a geração atual e (X, Y) as coordenadas das células onde uma raposa/coelho está posicionada no espaço do ecossistema, então a célula adjacente a ser escolhida é determinada por (G + X + Y) mod P. Suponha que a geração inicial é a 0 e a origem do mundo é (0,0).

1.5 Resolução de conflitos

Em cada geração, comece movendo o conjunto de coelhos e depois o conjunto de raposas. Ao mover o conjunto de coelhos/raposas, você deve assumir que todos os coelhos/raposas se movem ao mesmo tempo, ou seja, a ordem específica pela qual os coelhos/raposas são movidos não deve afetar o resultado final. No entanto, durante o movimento dentro de uma geração, pode acontecer que vários coelhos/raposas tentem se mover para as mesmas células. Quando isso acontece, você deve aplicar as seguintes regras:

- Quando dois ou mais coelhos se moverem para a mesma célula, apenas aquele com a idade de procriação mais antiga (GEN_PROC_COELHOS) é mantido e todos os outros coelhos morrem.

- Quando duas ou mais raposas se deslocam para a mesma célula, apenas a que tem a idade de procriação mais antiga (GEN_PROC_RAPOSAS) é mantida e todas as outras raposas morrem. Se houver duas ou mais raposas com a mesma idade de procriação, mantemos a raposa com menos fome (GEN_COMIDA_RAPOSAS).

1.6 Implementação

Comece implementando uma versão sequencial do programa e só então desenvolva uma implementação paralela para ambientes de memória compartilhada. Para isso, você pode usar processos com segmentos de memória compartilhada ou ou threads usando os modelos de programação Pthreads ou OpenMP. Iremos chamar os processos ou threads envolvidos na paralelização de *Workers*.

Para a implementação paralela, comece com estratégias simples, por exemplo, dando a cada *Worker* uma parte fixa do mundo. Observe que, no final de cada geração, pode ser necessário sincronizar dados sobre raposas e coelhos que se deslocam do espaço de um *Worker* para o espaço de outros *Workers* adjacentes . Você também pode dividir a matriz horizontalmente, verticalmente ou em ambas as dimensões de maneira estática, embora soluções mais eficientes devam ser adaptáveis para equilibrar dinamicamente a carga de trabalho entre os *Workers* em execução.

1.7 Entrada e Saída

A aplicação a ser desenvolvida deve aceitar como entrada um arquivo com os parâmetros de configuração e a descrição do estado inicial do ecossistema. Essa descrição deve obedecer ao seguinte padrão: a primeira linha de entrada

possui todos os parâmetros de configuração (incluindo um número inteiro N representando o número de objetos no ecossistema inicial), seguido por N linhas, cada uma representando um objeto do ecossistema inicial. Os parâmetros de configuração são os seguintes (em ordem):

- GEN_PROC_COELHOS - número de gerações até que um coelho possa procriar
- GEN_PROC_RAPOSAS - número de gerações até que uma raposa possa procriar
- GEN_COMIDA_RAPOSAS - número de gerações para uma raposa morrer de fome
- N_GEN - número de gerações para a simulação
- L - número de linhas da matriz representando o ecossistema
- C - número de colunas da matriz representando o ecossistema
- N - número de objetos no ecossistema inicial

Os objetos no ecossistema inicial são fornecidos usando o formato OBJETO X Y, em que OBJETO pode ser ROCHA, RAPOSA ou COELHO, e X e Y são as coordenadas em que o OBJETO está posicionado na matriz. Um exemplo de entrada é:

```
2 4 3 6 5 5 9
ROCHA 0 0
COELHO 0 2
RAPOSA 0 4
RAPOSA 1 0
RAPOSA 1 4
ROCHA 2 4
COELHO 3 0
COELHO 4 0
RAPOSA 4 4
```

A aplicação deve escrever na saída padrão a descrição do estado final do ecossistema, seguindo o mesmo formato usado para a entrada. Para o exemplo dado, a saída deve ser:

```
2 4 3 0 5 5 5
ROCHA 0 0
ROCHA 2 4
COELHO 3 0
RAPOSA 4 0
RAPOSA 4 3
```

Mais detalhadamente, para o exemplo dado, a sequência de gerações é a seguinte:

```
Gen 0     Gen 1     Gen 2     Gen 3     Gen 4     Gen 5     Gen 6
-------   -------   -------   -------   -------   -------   -------
|* C R|   |*C R |   |* R  |   |*  R |   |*   R|   |*    |   |*    |
|R   R|   |   R |   |  R  |   |     |   |     |   |     |   |     |
|    *|   |R   *|   |    *|   |    *|   |    *|   |    *|   |    *|
|C    |   | C  R|   |R    |   |     |   | C   |   |C    |   |C    |
|C   R|   | C   |   |  C R|   |RCC  |   | R C |   | RR  |   |R  R |
-------   -------   -------   -------   -------   -------   -------
```

No exemplo acima, * é uma rocha, C um coelho e R raposa. O arquivo de entrada deve ser lido da entrada padrão e os resultados devem ser impressos na saída padrão.

2 Especificação

A partir do problema descrito acima, criar uma versão paralela com memória compartilhada do mesmo. Espera-se que sejam explorados todos os potenciais de melhora na execução do algoritmo. As implementações devem ser feitas em linguagem C. Não devem ser utilizados pacotes ou bibliotecas específicas, a menos que autorizadas pelo professor. Os algoritmos devem ser compiláveis com o gcc.

3 Documentação

A paralelização da aplicação pode ser dividida em 4 partes, as quais devem gerar um relatório técnico. São elas:

1. perfil de desempenho sequencial (usando prof)
2. identficação das oportunidades de paralelização
3. paralelização
4. avaliação dos ganhos da paralelização

3.1 Pefil de desempenho sequencial

O ponto de partida desta tarefa é a execução da aplicação para carga de trabalho proposta para a mesma. Para cada execução deve-se analisar o seu tempo de execução, o tempo de execução em cada uma das suas funções, assim como a variação em consequência da variação de parâmetros e da carga. Mais ainda, deve-se avaliar o uso de recursos computacionais por parte da aplicação, utilizando aplicativos como vmstat. Um dos objetivos da determinação do pefil de desempenho é determinar as porções de código que teriam maior impacto se paralelizadas, por serem as mais demandantes computacionalmente.

3.2 Identificação das oportunidades de paralelização

A identificação das oportunidades de paralelização deve prover as seguintes informações sobre cada uma das oportunidades identificadas:

- qual é a oportunidade, ou seja, qual o arquivo, função e linhas de código que compõem a oportunidade?
- qual a sua significância no perfil sequencial, ou seja, quanto do tempo de execução da aplicação está associado à oportunidade?
- qual a estratégia de paralelização com a sua justificativa (por que você acredita que essa estratégia é escalável e eficiente) e expectativa de eficiência, em termos da fração serial e paralela resultante. A apresentação de pseudo códigos é bem vinda.

3.3 Paralelização

O relatório de paralelização deve documentar todas as atividades de paralelização realizadas no código da aplicação em termos tanto da estratégia utilizada como do código modificado. É uma documentação tradicional de trabalho prático, o que também inclui testes de execução.

3.4 Entrega

Os trabalhos devem ser entregues impreterivelmente até a data marcada no moodle.