Jogo Paralelo da Vida

Gabriel Duarte Soares - 122078892 , Henrique Lima Cardoso - 122078397

Relatório Parcial Programação Concorrente (ICP-361) — 2024/2

1

1. Descrição do problema geral

Descrever o problema escolhido:

O projeto consiste na simulação de autômatos celulares, um Jogo de Zero Jogadores onde o espaço é dividido em células (ou ”automatos”) que mudam de estado ao longo do tempo, seguindo um conjunto de regras específicas. Nós simularemos o Game of life elaborado pelo John Conway, onde cada célula representa uma célula ”viva” ou ”morta”, e seu estado em cada passo do tempo depende do estado das células vizinhas.

Na versão sequencial, a simulação de autômatos celulares funciona da seguinte forma: Um grid bidimensional (ou tridimensional, para uma versão mais avançada) é inicializado, onde cada célula possui um estado. A cada frame, o programa avalia o estado de cada célula e de suas vizinhas, com base em regras predefinidas. Isso é repetido por um número de iterações definido pelo usuário - podendo ser infinito - gerando uma sequência de grids que representam a evolução do sistema ao longo do tempo.

Dados entrada:

- Configuração Inicial das células, representado por um array que guarda o estado de cada uma delas.
- Dimensão do conjunto de células - se é bi, tri ou n-dimensional, e o tamanho do universo.
- Números de iterações: por quanto tempo o autômato deve rodar, ou se deve rodar infinitamente.
- Regras de Atualização: Conjunto de regras que determinam como uma célula muda de estado com base nos estados das células vizinhas. (No jogo do Conway, já é determinado)
- (Opcionais) Velocidade e Bufferização

O programa deve mostrar a evolução do jogo ao longo do tempo. Um registro das configurações do grid em cada passo, que pode ser exibido como uma sequência de imagens ou como uma animação. (Teremos que decidir a melhor forma de fazer isso)

2. Projeto da solução concorrente

Descrever o projeto da solução concorrente para o problema:

Ao implementar uma solução concorrente para a simulação de autômatos celulares, a tarefa principal – atualização do grid ao longo do tempo – pode ser dividida entre diferentes threads ou rotinas paralelas. Acelerando significativamente o programa.

Inicialmente, dividiremos os espaço da maneira mais simples que conheçemos - assim como nas primeiras aulas - designaremos blocos para cada thread, de forma a isolar o espaço de atuação de cada uma delas. Nesse quesito, esse programa é embaraçosamente

paralelo. Posteriormente, podemos estudar formas mais avançadas de otimização do Jogo da Vida e ver as possibilidades de paralelismo nelas.

Entre essas formas mais avançadas estão:

- Uso de outras estruturas de dados como Matrizes Esparças
- Aceleração com GPU, milhares de células podem ser atualizadas paralelamente
- Detecção de áreas de atividade, ignorando regiões vazias.
- Balanceamento dinâmico do workload das threads a partir da atividade dos blocos
- Usar hashing para acelerar computacionalmente o cálculo de configurações conhecidas (Hashlife), mantendo uma estrutura leitor/escritor.
- Entre muitos outros.

Algo que seria muito divertido, mas parece ser muito difícil é considerar topologias distintas!

3. Casos de teste de corretude e desempenho

Descrever como o programa será testado:

Para avaliar o desempenho, os testes devem medir o tempo de execução da solução concorrente para diferentes tamanhos de grids e números de threads. Espera-se que o tempo de execução diminua significativamente com o aumento do número de threads.

Como o output do programa é uma sequência de tabelas - ou estado do universo. Quando testarmos a corretude do programa, teremos que comparar a solução paralela com a solução sequencial para cada frame. Isso será fácil de fazer, pois não envolve a parte gráfica - da visualização do programa.

Para testar o desempenho, olharemos para grids com dimensões distintas (pequeno até enorme). Podemos querer saber também a interferência na densidade de células vivas inicialmente.

Espera-se que o programa tenha quase 100% de eficiência! Isto é: o speedup é linear em relação ao número de threads. Mas em grids pequenos, a sobrecarga de comunicação pode limitar os ganhos, diminuindo a eficiência.

Se tivermos a oportunidade de implementar técnicas mais avançadas para a atualização do autômato, será divertido também observar qual é o impacto da paralelização.

4. Referências bibliográficas

- https://en.wikipedia.org/wiki/Conway's_Game_of_Life
- https://en.wikipedia.org/wiki/Embarrassingly_parallel
- https://rbeaulieu.github.io/3DGameOfLife/3DGameOfLife.html?
- https://kodub.itch.io/game-of-life-3d
- https://en.wikipedia.org/wiki/John_Horton_Conway
- https://en.wikipedia.org/wiki/Zero-player_game
- https://en.wikipedia.org/wiki/Hashlife
- Slides da professora Silvana Rossetto