

Modalidade do trabalho: Relatório técnico-científico
Evento: XXIII Seminário de Iniciação Científica

COMPARAÇÃO DE DESEMPENHO ENTRE IMPLEMENTAÇÕES DO ALGORITMO JOGO DA VIDA COM PTHREAD E OPEMMP¹

Márcia Da Silva², Igor Gamste Haugg³, Eliézer Silveira Prigol⁴, Édson L. Padoin⁵, Rogério S. M. Martins⁶.

¹ Trabalho desenvolvido na disciplina de Processamento Paralelo desenvolvido na Unijuí.

² Bolsista voluntária do grupo GCA, aluna do curso Ciência da Computação da Unijuí.

³ Bolsista PIBITI/CNPq, aluno do curso Ciência da Computação da Unijuí.

⁴ Aluno do curso Ciência da Computação da Unijuí.

⁵ Professor do curso Ciência da Computação da Unijuí.

⁶ Professor do Curso Ciência da Computação da Unijuí.

Introdução

Atualmente tem se utilizado cada vez mais processamento paralelo nas implementações para problemas que exigem alto processamento e também aqueles que possuem alto tempo de execução. As threads são uma forma de implementação deste tipo de processamento e podem ser utilizadas de diferentes formas, como por exemplo, pthreads e OpenMP. Quando uma implementação sequencial é executada, normalmente seu tempo de execução é maior comparado a implementações com threads. Pois com processamento paralelo, em máquinas que possuem processadores com múltiplos núcleos, é possível executar tarefas em paralelo, com isso, o tempo de execução é reduzido.

Neste trabalho serão utilizados os conceitos de threads para executar implementações realizadas com pthreads e OpenMP. Estas implementações serão aplicadas no problema jogo da vida, com o objetivo de comparar o tempo de execução obtido de cada implementação.

Estado da Arte

Jogo da vida é um autômato celular desenvolvido pelo matemático John Conway em 1970 (CONWAY, 1970). Esse problema foi apresentado em 1940 por John Von Neumann, o qual tentou construir uma máquina hipotética que construísse cópias dela mesma, o resultado foi atingido quando ele encontrou um modelo matemático para cada máquina com muitas regras complexas em um tabuleiro retangular. O jogo da vida surgiu como a tentativa bem sucedida de Conway de simplificar drasticamente as ideias de Von Neumann (CALLAHAN, 2010). O ambiente do jogo da vida é um tabuleiro bidimensional dividido em células quadradas, cada célula possui dois possíveis estados, vivo ou morto. Todas as células interagem com suas quatro vizinhas, que estão distribuídas horizontalmente e verticalmente. As células vivas são representadas com o número 1, e as mortas com 0. A cada geração, as regras são analisadas. A primeira regra define que qualquer célula viva com menos de dois vizinhos vivos, morre. Na segunda regra qualquer célula viva com dois ou três

Modalidade do trabalho: Relatório técnico-científico
Evento: XXIII Seminário de Iniciação Científica

vizinhos vivos, vive. Na terceira regra qualquer célula viva com mais de três vizinhos, morre. E na quarta regra qualquer célula com exatamente três vizinho vivos se torna uma célula viva. Desta forma, o padrão inicial reflete nas gerações futuras. A primeira geração é criada aplicando as regras simultaneamente em cada quadrante do tabuleiro. Nascimentos e mortes acontecem simultaneamente. As regras continuam a ser aplicadas repetidamente para gerar mais gerações de população (GYMREK, 2010).

Processamento paralelo é uma forma eficiente de processamento com ênfase na exploração de eventos simultâneos na execução de um software. A motivação para o processamento paralelo é a possibilidade de aumentar a capacidade de processamento de uma única máquina. É possível realizar a implementação através da utilização de threads. As threads compartilham entre elas os recursos do processo ao qual pertence. Quando o processo que está vinculado a thread é finalizado, a thread também é finalizada. Uma thread pode ser considerada uma função que é executada de forma independente do programa principal. Cada processo possui atributos que são compartilhados por suas threads (NICHOLS, BUTTLAR, FARRELL, 1996).

A programação paralela com thread é facilitada pela existência de um único espaço de endereçamento através de todo o sistema de memória. Assim, cada processo pode ler e escrever em todas as posições de memória. A programação com threads pode ser implementada de diferentes formas, seja com uso do padrão pthread ou da API OpenMP. A primeira, pthread é uma interface de manipulação de threads padronizada em 1995 pelo IEEE, sendo denominada POSIX threads (Pthreads). Este tipo de thread é definida como um conjunto de tipos de dados em C e um conjunto de rotinas (NICHOLS, BUTTLAR, FARRELL, 1996). A segunda, OpenMP é uma interface de programação (API), portátil, baseada no modelo de programação paralela de memória compartilhada para arquiteturas de múltiplos processadores. Em OpenMP, a região paralela é a estrutura básica de paralelismo no OpenMP. Uma região paralela define uma seção do programa. Os programas começam a execução com uma única thread (denominada thread master). Quando a primeira região paralela é encontrada, a thread master cria as novas threads. Cada thread executa as sentenças que estão dentro da região paralela. No final da região paralela, a thread master espera pelo término das outras threads e continua então a execução de outras sentenças (DAGUM, 1998).

Metodologia

A realização deste trabalho foi dividida em três partes. Inicialmente foi implementado o jogo da vida nos formatos sequencial e paralelo, utilizando pthread e OpenMP. Com os algoritmos implementados, foram realizadas as execuções 10 vezes e realizada uma média sobre os tempos de execução. Após os testes, foi realizado uma comparação dos resultados.

Para realização dos testes utilizamos um notebook com processador AMD Phenom II X3 N830, com 3 núcleos, sistema operacional Linux, kernel versão 3.13, linguagem de programação C, utilizando o compilador GCC de versão 4.8.2.

Modalidade do trabalho: Relatório técnico-científico
Evento: XXIII Seminário de Iniciação Científica

Resultados e Discussões

Nesta seção, apresenta-se os testes realizados nas implementações do algoritmo jogo da vida com pthreads, OpenMP e também a execução na forma sequencial.

Os testes realizados neste artigo utilizaram como base, um tabuleiro de tamanho 5000x5000, o qual foi percorrido 100 vezes. Executando de forma sequencial e sem flags de compilação, o tempo médio de execução foi de 84,16 segundos. Com a utilização da flag “-O3”, o tempo médio de execução do algoritmo sequencial reduziu para 28,6 segundos, o que representa um Speed-up de 0,34, tendo uma melhora de 66% em relação ao algoritmo sequencial sem flags. Também foi executado o algoritmo sequencial utilizando a flag “-march=amdfam10 -O2”, a qual é uma flag específica dos processadores da AMD e está disponível no site do fabricante. Nesta execução o tempo foi de 28,8 segundos, sendo inferior ao alcançado com a flag “-O3”. Na Tabela 1 apresentamos os resultados da execução da implementação utilizando pthreads, com diferentes números de threads.

pthreads				OpenMP			
Threads	Tempo(s)	Speed-up	Eficiência	Threads	Tempo(s)	Speed-up	Eficiência
1	28.67	1.00	1.00	1	24.44	1.00	1.00
2	19.02	0.66	0.33	2	17.22	0.70	0.35
3	15.45	0.54	0.18	3	14.85	0.61	0.20
4	15.66	0.55	0.14	4	15.17	0.62	0.16
5	14.39	0.50	0.10	5	14.89	0.61	0.12
6	13.72	0.48	0.08	6	14.68	0.60	0.10
7	13.26	0.46	0.07	7	14.56	0.60	0.09
8	12.69	0.44	0.06	8	14.65	0.60	0.07
9	12.18	0.42	0.05	9	14.52	0.59	0.07
10	11.97	0.42	0.04	10	14.51	0.59	0.06
20	10.18	0.35	0.02	20	14.42	0.59	0.03

Tabela 1. Resultados dos testes utilizando pthreads e OpenMP

A partir dos dados obtidos na Tabela 1, foi possível compreender que conforme aumentamos o número de threads, o tempo de execução do algoritmo pthread é reduzido, chegando ao tempo de 10.18 segundos com a utilização de 20 threads. Também foi possível compreender que ao utilizarmos 3 threads na implementação com OpenMP, o tempo reduziu de forma significativa, passando de 24,4 segundos para 14,8 segundos, isto ocorre porque o computador utilizado possui um processador com somente três núcleos. Com OpenMP, conforme aumentamos o número de

Modalidade do trabalho: Relatório técnico-científico
Evento: XXIII Seminário de Iniciação Científica

threads, o tempo de execução permaneceu entre 14 e 15 segundos, sem conseguir diminuir o tempo de execução significativamente.

Comparando as duas execuções, podemos identificar que para o jogo da vida a implementação pthread trouxe melhores resultados quando utilizamos um maior número de threads. Esta comparação pode ser vista na Figura 1. Onde o tempo de execução foi diminuindo conforme aumentamos o número de threads.

Comparando a forma de realizar as duas implementações, OpenMP é de fácil implementação, pois é uma API que já contém muitas funções pré-programadas. Isto torna o código mais simples e compacto. Na implementação com pthreads não se tem disponível muitas funções, então é necessário implementar todas as etapas para o funcionamento das threads.

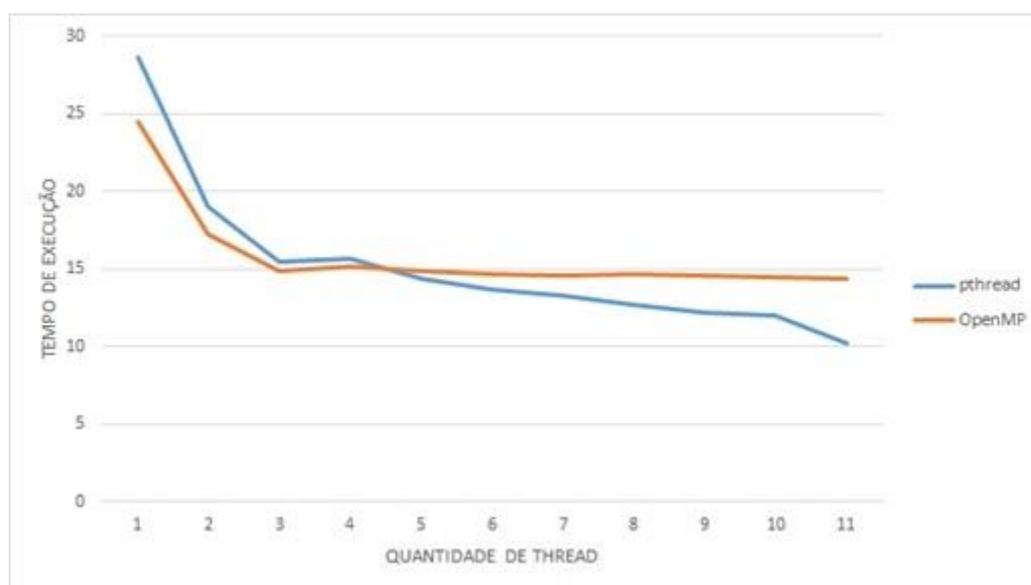


Figura 1. Comparação entre pthread e OpenMP

Conclusões

Esse artigo apresentou uma análise de desempenho entre as versões paralelas do algoritmo jogo da vida implementado com OpenMP e pthread. A partir dos testes realizados, foi possível concluir que para o algoritmo jogo da vida, o OpenMP possui melhores resultados até a utilização de 3 threads. Isto pode acontecer porque o computador utilizado para a realização dos testes, possui somente 3 núcleos no processador. Conforme aumentamos o número de thread, a implementação com pthread possui resultados mais satisfatórios.

Modalidade do trabalho: Relatório técnico-científico

Evento: XXIII Seminário de Iniciação Científica

Também foi possível entender e aplicar os conhecimentos adquiridos em aula sobre threads, e verificar a importância da utilização desta forma de programação. Pois a implementação de forma sequencial tem um tempo de execução muito maior do que na utilização de threads, independentemente da forma que implementamos as threads. Como atualmente está cada vez mais sendo buscado a melhora na performance dos algoritmos, e sempre com o objetivo de reduzir o tempo de execução, a realização deste trabalho foi muito importante, pois conseguimos aprender a utilizar estes tipos de implementação.

Como trabalhos futuros, é possível utilizar outra forma de implementação do jogo da vida. Existem diferentes estilos de implementação, que podem ser testados com a utilização de threads. Também pode ser adicionadas novas regras, para que além da verificação vertical e horizontal, sejam verificados os vizinhos diagonais.

Palavras-Chave

Processamento Paralelo; Threads; Autômato celular.

Agradecimentos

Agradecemos pelo auxílio dos professores no desenvolvimento deste trabalho.

Referências Bibliográficas

CALLAHAN, Paul. What is the Game of Life. World of Math, 2000.

CONWAY, John. The game of life. Scientific American, v, n. 4, p. 4, 1970.

DAGUM, Leonardo; ENON, Rameshm. OpenMP: an industry standard API for shared-memory programming. Computational Science & Engineering, IEEE, n. 1, p. 46-55, 1998.

GYMREK, Melissa. Conway's Game of Life. 2010.

NICHOLS, Bradford; BUTTLAR, Dick; FARRELL, Jacqueline. Pthreads programming: A POSIX standard for better multiprocessing. " O'Reilly Media, Inc.", 1996.