

# Jogo da Vida com OpenMP

- A atividade proposta tem como objetivo fazer uso das diretivas **#pragma omp critical** e **#pragma omp for reduction()** para realizar a contagem do número de células vivas do jogo da vida após a execução de 1000 gerações.
- O tabuleiro utilizado para tal fim consiste em uma matriz bidimensional de dimensões 2048 x 2048.
- A seguir é mostrado o tempo de execução e o speedup para as duas versões onde foi contabilizado o tempo de execução do trecho ao qual se aplicou às diretivas.

# Resultados com o uso de critical

Número de Threads	Tempo de Processamento (s)	Speedup
1	0,019656	1,000
2	0,0196380	1,000
4	0,052258	0,376
8	0,102910	0,191

# Resultados com o uso de reduction

Número de Threads	Tempo de processamento (s)	Speedup
1	0,016076	1,000
2	0,008511	1,889
4	0,006148	2,615
8	0,006417	2,505

# Comparação entre as duas versões

- Pelos resultados obtidos com o uso da diretiva `#pragma omp critical` fica evidente que a melhor estratégia é executar o trecho de código com apenas 1 thread.
- O custo computacional da criação das threads, somado ao custo da manutenção da região crítica protegida fez com que o speedup diminuísse ao se aumentar o número de threads enquanto o resultado desejado era que o speedup aumentasse com o aumento do número de threads.

# Comparação entre as duas versões

- A tabela com os resultados usando a diretiva `#pragma omp for reduction()` mostra um ganho no valor do speedup.
- Usando esta abordagem vemos que é uma boa estratégia executar o trecho de código com 2 threads pois é onde o ganho de speedup foi mais significativo.
- Pela análise dos resultados obtidos podemos concluir que é mais vantajoso em termos de tempo computacional o uso da diretiva `#pragma omp for reduction()` ao invés da diretiva `#pragma omp critical`.