Estratégias de Paralelização

Solução 1

Introdução

Na solução 1 o objetivo foi paralelizar o loop que contém o número de execuções. Para facilitar o acesso cada thread possui uma população individual.

Foram feitos testes com 3 configurações de threads: 2, 4, e 6. Em relação ao tamanho a única variável que mudou foi n_trials.

Metodologia

Na solução 1 foi buscado paralelizar o for relacionado ao número de experimentos já que para se obter maior precisão na pesquisa deve-se aumentar o valor de ‘n_trials’.

O método de schedule foi o dynamic.

Resultados

No slide a seguir estão os resultados de cada teste: Cada par de tabela é composto por uma tabela relativa as configurações do problema e a de baixo com o número de threads.

Resultados

Pop. Size	N. Trials	N. Probs
40	500	100

N. Threads	Time (seg)	Speedup (seg)
1	26	-
2	26	1
4	29	0.9
6	23	1.13

Pop. Size	N. Trials	N. Probs
40	500	100

N. Threads	Time (seg)	Speedup (seg)
1	144	-
2	135	1.06
4	115	1.25
6	114	1.26

Pop. Size	N. Trials	N. Probs
40	500	100

N. Threads	Time (seg)	Speedup (seg)
1	642	-
2	569	1.12
4	564	1.13
6	566	1.13

- Config. Pequena:

  Como pode ser visto pelos resultados, com uma configuração do problema muito pequena é difícil saber a melhor solução.

- Config. Média e Grande:

  Já com uma configuração do problema maior, quanto mais threads mais rápido. Mas devido às limitações de hardware (2 cores, 4 threads) e software (Oracle VM) após 4 threads o tempo ficou virtualmente o mesmo.

Solução 2

Introdução

Na solução 2 a paralelização ocorreu no loop principal, o qual depende da variável n_probs.

Foram feitos testes com 3 configurações de threads: 2, 4, e 6. Em relação às configurações do problema, a variável population_size foi modificada.

Metodologia

Na solução 2 foi buscado paralelizar o for relacionado ao número de probabilidades pois esse é o loop mais externo, logo não é necessário ficar criando threads a cada iteração.

O método de schedule foi dynamic, com chunck de n_probs/n_threads.

Resultados

No slide a seguir estão os resultados de cada teste: Cada par de tabela é composto por uma tabela relativa as configurações do problema e a de baixo com o número de threads.

Resultados

Pop. Size	N. Trials	N. Probs
50	100	100

N. Threads	Time (ms)	Speedup (ms)
1	8979	-
2	8371	1.07
4	9543	0.95
6	10413	0.86

Pop. Size	N. Trials	N. Probs
100	100	100

N. Threads	Time (seg)	Speedup (seg)
1	67	-
2	49	1.37
4	42	1.6
6	44	1.52

Pop. Size	N. Trials	N. Probs
150	100	100

N. Threads	Time (seg)	Speedup (seg)
1	209	-
2	148	1.41
4	118	1.77
6	116	1.8

- Config. Pequena:

  Como pode ser visto pelos resultados com um configuração do problema muito pequena o overhead de criar threads afeta consideravelmente o tempo e só 2 threads é mais vantajoso que sequencialmente.

- Config. Média e Grande:

  Já com uma configuração do problema maior, quanto mais threads mais rápido. Mas devido às limitações de hardware (2 cores, 4 threads) e software (Oracle VM) após 4 threads o tempo estagnou e até ficou pior em no caso médio.