

Laboratório 2

Computação Concorrente - UFRJ

Renan Mendanha - 118055604

Os testes foram conduzidos em uma máquina virtual AMD FX-6350 com 3 cores e 3 Threads(1 thread per core).

Medições do algoritmo para multiplicação de matrizes:

- Dimensão das matrizes: 500
 1. Thread:
1.602830 s.
1.564615 s.
1.552902 s.
 2. Threads:
0.805666 s.
0.834030 s.
0.839056 s.
- Dimensão das matrizes: 1000
 1. Thread:
22.442952 s.
22.812716 s.
22.100957 s.
 2. Threads:
11.319730 s.
11.407234 s.
11.303751 s.
- Dimensão das matrizes: 2000
 1. Thread:
185.498727 s.
184.685481 s.
186.681357 s.
 2. Threads:
96.189731 s.
94.926647 s.
96.369219 s.

Speedup:

Matriz de dimensão 500:

$$S = \frac{T(1)}{T(2)} = \frac{1.552902}{0.805666} = 1.927476$$

Matriz de dimensão 1000:

$$S = \frac{T(1)}{T(2)} = \frac{22.100957}{11.303751} = 1.955187$$

Matriz de dimensão 2000:

$$S = \frac{T(1)}{T(2)} = \frac{184.685481}{94.926647} = 1.945559$$

Conclusão:

Podemos observar, após o registro de tempo durante as execuções da multiplicação, que ao dobrarmos o número de threads conseguimos cortar aproximadamente pela metade o tempo necessário para a computação, ou seja, temos um speedup de aproximadamente 2(dobro em desempenho) em relação à execução sequencial.