

THIAGO MORAES

MATRÍCULA 21452625

TP13-LPAV

PTHREADS

1) QUESTÃO

K/N	512	1024	2048	4096	8192	32768
1	0.000146	0.000060	0.000069	0.000068	0.000210	0.000164
2	0.000082	0.000050	0.000088	0.000055	0.000060	0.000175
4	0.000036	0.000092	0.000115	0.000205	0.000135	0.000234
8	0.000060	0.000209	0.000046	0.000074	0.000081	0.000165
16	0.000278	0.000737	0.000065	0.000085	0.000113	0.000158
32	0.000729	0.000157	0.000170	0.000220	0.000259	0.000384
64	0.001567	0.000821	0.001074	0.000374	0.000692	0.002341

Percebi que os tempos computados para processos com duas Threads foram menores na maioria do que com uma só Threads. Mas que processos com 8 Threads em alguns casos foram mais demorados do que com 16 Threads. Consultei um amigo para compararmos dados, e os tempos se distinguiram-se, acredito que seja pelo fato do escalador de processos do SO dele agir diferente do meu.

2) QUESTÃO

K/N	128	256	512	1024	2048
1	0.013544	0.122530	0.904259	14.282906	135.735638
2	0.016466	0.069538	0.405737	6.367467	63.639652
4	0.008970	0.048935	0.264307	1.721363	21.788643
8	0.003386	0.030130	0.107874	1.202496	7.953848
16	0.004018	0.016416	0.066729	0.573516	3.871616
32	0.004120	0.009647	0.048802	0.213075	2.637665
64	0.004324	0.009225	0.026269	0.194182	1.030836

Podemos notar que multiplicação de matrizes mesmo usando Threads foi algo custoso, inicialmente para valores de 128 a 256 não chegamos tão próximo de 1s, mas a partir de 512 com apenas uma Thread chegamos bem próximos de 1s. O que me chamou atenção foi para um valor N = 1024 com apenas uma Threads obtivemos 14.282906s, eu não achava que iria demorar tanto o mesmo vale para 2048 usando apenas uma Thread, mas se

levamos em consideração que multiplicação de matrizes tem complexidade assintótica $O(n^3)$ faz todo sentido a demora. Mas quando dividimos em 64 Threads obtivemos tempos bem pequenos.