

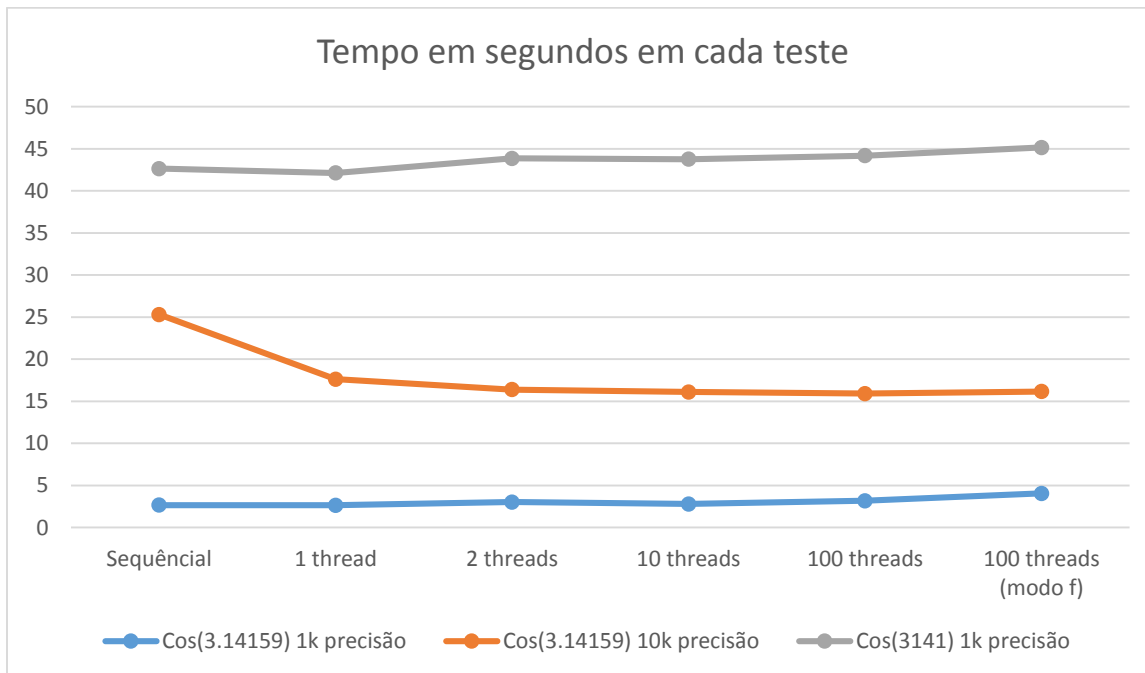
Relatório EP2

Os testes foram executados em uma máquina virtual com o Ubuntu 64 bits instalado, a máquina faz uso de dois núcleos e 3GB de memória.

Foram feitos vários testes, e em cada teste foram executados pelo menos 5 vezes, e assim foi calculado a média do tempo em cada caso. Nos testes realizamos os cálculos do cosseno de 3.14159 e cosseno de 3141 nessas situações:

Tivemos o seguinte resultado em segundos de tempo de execução:

			m	m	m	m	f
		Sequencial	1 thread	2 threads	10 threads	100 threads	100 threads
3.14159	1k	2.670s	2.654s	3.020s	2.797s	3.180s	4.054s
3.14159	10k	25.304s	17.628s	16.388s	16.108s	15.914s	16.170s
3134	1k	42.628s	42.123s	43.857s	43.753s	44.173s	45.150s



O tempo foi medido usando o resultado obtido em "real" da função *time* do Linux, já que o tempo que está em "user" leva em consideração o tempo gasto em cada thread como algo individual e assim não iria satisfazer o objetivo proposto, que era calcular o tempo ganho com a utilização das threads.

O resultado da execução obviamente não foi o que eu estava esperando, já que em alguns casos, o programa executa em modo sequencial levemente mais rápido do que quando threads são criadas com o objetivo de economizar tempo. Não foi o que aconteceu, e também, quanto mais threads são criadas, não há nenhuma mudança significativa no tempo de execução.

A possível causa desse resultado deve ser por causa dos semáforos terem uma ordem a serem executados, e a parte do código que executa em paralelo (seção não crítica) é composta por pouquíssimas operações, enquanto a seção crítica e a barreira são as que consomem mais tempo, fazendo com que todos os outros processos fiquem a maior parte do tempo em espera, podendo apresentar um problema de overhead.

Números de alta precisão

Neste exercício Programa foi usado a biblioteca gmp, em que não há limite de precisão aritmética, sendo o único limite a memória da máquina usada, fizemos bastante o uso da variável `mpf_t` em que ao inicializar as variáveis setamos como 1048576 o mínimo de bits usados para compor a precisão do número, o que é mais que suficiente para o nosso programa, já que no máximo serão impressos 100,000 dígitos de precisão do valor do cosseno.