

# Laboratório III

João Pedro Cavalcante - DRE: 119052154

12 de Abril de 2021

## Dados Coletados

Utilizando o programa feito em  $C$  podemos observar questões interessantes que serão aprofundadas mais a frente.

- Para  $n = 3$

Versão	$n$	Threads	Tempo	Valores de $\pi$
Sequencial	$10^3$	-	0.000026s	3.140592653839794
Concorrente	$10^3$	1	0.000249s	3.140592653839794
Concorrente	$10^3$	2	0.000178s	3.140592653839795
Concorrente	$10^3$	4	0.000212s	3.140592653839791

- Para  $n = 6$

Versão	$n$	Threads	Tempo	Valores de $\pi$
Sequencial	$10^6$	-	0.013138s	3.141591653589774
Concorrente	$10^6$	1	0.012773s	3.141591653589774
Concorrente	$10^6$	2	0.006334s	3.141591653589692
Concorrente	$10^6$	4	0.008055s	3.141591653589781

- Para  $n = 9$

Versão	$n$	Threads	Tempo	Valores de $\pi$
Sequencial	$10^9$	-	12.052161s	3.141592652588050
Concorrente	$10^9$	1	11.656895s	3.141592652588050
Concorrente	$10^9$	2	7.262054s	3.141592652589258
Concorrente	$10^9$	4	7.362327s	3.141592652589210

## Conclusão

Analisando os dados coletados do último tópico e comparando com o valor de  $\pi$  fornecido da biblioteca *math.h* da linguagem de  $C$ , valor de  $\pi$  é 3.141592653589793, podemos notar que quanto maior o  $n$ , ou seja, quanto maior a quantidade de números na soma, maior a precisão de  $\pi$ . Outro aspecto importante seria que para  $n$  pequenos a versão concorrente não se sai melhor do que a versão sequencial, por conta das criações das threads o programa leva um tempo desnecessário e acarreta em uma "lentidão" comparada com a sequencial, e para  $n$  grandes pode-se notar uma melhoria em relação

a versão sequencial, valendo a pena o uso da concorrência. Ambas as versões se aproximam bastante do valor de  $\pi$ , principalmente, a versão concorrente e com mais threads, porém nos teste não coincidiu com o valor de  $\pi$ . Por conta da aproximação de ponto flutuante, e sua imprecisão de certa forma, acarretou nas diferenças entre a versão sequencial e a concorrente, a forma que foi realizada a soma (sequencial foi realizado em uma sequencia, e na concorrente em blocos e depois a soma dos blocos) trouxe as diferenças, e devido a esse modo da soma ser diferente entre as versões a concorrente foi se aproximando cada vez mais de  $\pi$ , trazendo assim consigo resultados mais precisos.

Obs.: Novamente os resultados de 4 threads se assemelharam com os de 2 threads, e eu acredito que tenha sido por conta da máquina utilizada, os detalhes abaixo.

Macbook Air 2017 13"

CPU: 1,8 GHz Dual-Core Intel Core i5