

GOOGLE COLAB EXPERIMENT 1 - (Tesla K80)

Kernel 1 - Simple Copy

No. of elements	Threads / Block	Execution Time (ms)
128 x 10 ⁵	16	2.420
128 x 10 ⁵	32	1.251
128 x 10 ⁵	64	0.526
128 x 10⁵	128	0.436
128 x 10⁵	256	0.439
128 x 10 ⁵	512	0.460
128 x 10 ⁵	1024	0.496
64 x 10⁵	128	0.234
64 x 10⁵	256	0.233
32 x 10⁵	128	0.117
32 x 10⁵	256	0.117
8 x 10⁵	128	0.038
8 x 10⁵	256	0.034

Observation:- Optimal block size is 128 or 256. Too high a thread number reduces the parallelism and too low fails to use all the available resources, especially in size 16 when even the warp size is 32

Kernel 2 - Scalar Multiplication

No. of elements	Threads / Block	Execution Time (ms)
128 x 10 ⁵	16	2.437
128 x 10 ⁵	32	1.261
128 x 10 ⁵	64	0.529
128 x 10⁵	128	0.440

128 x 10⁵	256	0.442
128 x 10 ⁵	512	0.463
128 x 10 ⁵	1024	0.499
64 x 10⁵	128	0.232
64 x 10⁵	256	0.232
32 x 10⁵	128	0.117
32 x 10⁵	256	0.117
8 x 10⁵	128	0.038
8 x 10⁵	256	0.038

Observation:- Not much different from kernel 1 except for a few microsecond of difference, which implies that scalar multiplication is quite fast

Kernel 3 - Addition of 2 arrays

No. of elements	Threads / Block	Execution Time (ms)
128 x 10 ⁵	16	2.492
128 x 10 ⁵	32	1.288
128 x 10 ⁵	64	0.608
128 x 10⁵	128	0.589
128 x 10⁵	256	0.588
128 x 10 ⁵	512	0.596
128 x 10 ⁵	1024	0.612
64 x 10⁵	128	0.301
64 x 10⁵	256	0.302
32 x 10⁵	128	0.150
32 x 10⁵	256	0.151
8 x 10⁵	128	0.043
8 x 10⁵	256	0.046

Observation:- In size 16 and 32, there is not much difference as it initially did not use the resources fully, and with double the elements to load it now fully uses the resources. The rest all shows nearly a 30% increase in time as each warp has to wait for both the operand to load.

Kernel 4 - Addition of sin and cos of 2 arrays respectively

No. of elements	Threads / Block	Execution Time (ms)
128 x 10 ⁵	16	3.582
128 x 10 ⁵	32	1.641
128 x 10 ⁵	64	0.789
128 x 10 ⁵	128	0.779
128 x 10 ⁵	256	0.787
128 x 10 ⁵	512	0.830
128 x 10 ⁵	1024	0.898
64 x 10 ⁵	128	0.401
64 x 10 ⁵	256	0.410
32 x 10 ⁵	128	0.199
32 x 10 ⁵	256	0.204
8 x 10 ⁵	128	0.060
8 x 10 ⁵	256	0.060

Observation:- In size 16, there is over a 40% increase in time, with the rest all have a nearly 30% increase in execution time. This goes to show that sin and cos operations are quite costly