



**Tecnológico
de Monterrey**

Campus Puebla

Multiprocesadores

Actividad 1.4

Jorge Martínez Hernández – A01731762

Profesor: Dr. Emmanuel Torres Rios

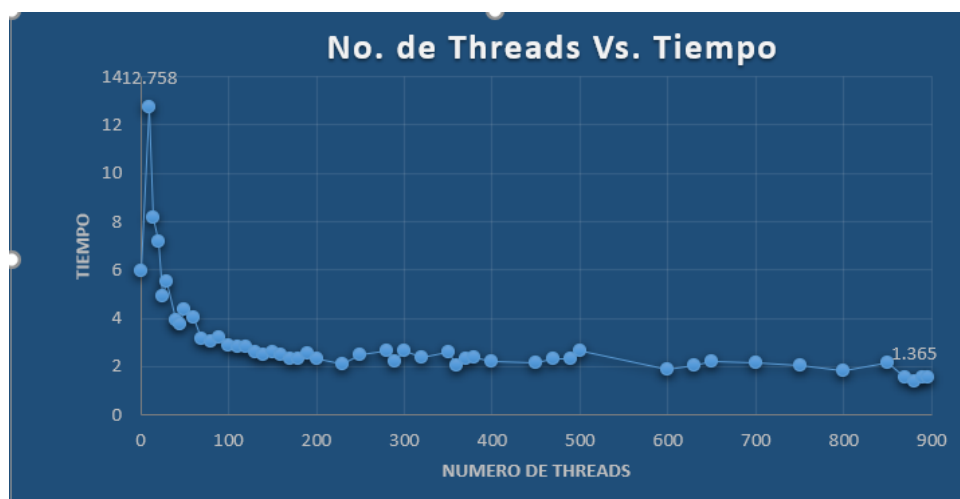
Introducción

En esta actividad se busca encontrar el mejor tiempo usando un programa usando el concepto de paralelización, previamente se busco el numero de pasos para encontrar el valor mas preciso para pi. Dándonos cuenta de que con 10×10^9 es el numero mas optimo y en un tiempo adecuado.

Desarrollo

En la tabla 1 se muestran las corridas de programa en el cual se fueron variando gradualmente los valores de threads y posteriormente obtuvimos el comportamiento del tiempo conforme la cantidad de threads asignados.

	Threads	Tiempo									
1	1	5.98	13	90	3.22	25	230	2.107	37	470	2.331
2	10	12.758	14	100	2.886	26	250	2.477	38	490	2.306
3	15	8.182	15	110	2.81	27	280	2.663	39	500	2.634
4	20	7.192	16	120	2.816	28	290	2.193	40	600	1.887
5	25	4.888	17	130	2.622	29	300	2.633	41	630	2.032
6	30	5.516	18	140	2.476	30	320	2.36	42	650	2.234
7	40	3.936	19	150	2.592	31	350	2.573	43	700	2.135
8	45	3.756	20	160	2.498	32	360	2.026	44	750	2.041
9	50	4.363	21	170	2.329	33	370	2.347	45	800	1.8
10	60	4.017	22	180	2.332	34	380	2.354	46	850	2.148
11	70	3.172	23	190	2.55	35	400	2.228	47	870	1.555
12	80	3.054	24	200	2.325	36	450	2.151	48	880	1.365
									49	890	1.541
									50	895	0.985



Conclusión

Revisando el las iteraciones me pude percatar que hubo una oscilación en cuanto el tiempo en el que se ejecutaba el programa, incluso si se corría con el mismo número de threads, encontrando que el mejor tiempo registrado fue 1.365s y con 900 threads ya no jalo el programa.

Bibliografía.

[Shameen Akhter, Jason Roberts \(2006\)](#)

Multi-Core Programming: Increasing erformance through Software Multi-Threading- [Intel Press](#)

[Barry Wilkinson, Michael Allen \(2005\)](#)

Parallel Programing: Techniques and applications using networked workstations- [Pearson](#)