

# Informe sobre mutex y busy waiting en el calculo de $\pi$

Luis Alberto Sánchez Moreno Colque

September 13, 2018

## 1 Comparación de Tiempos de BusyWait y Mutex

En la siguiente comparación se tiene el calculo del número de  $\pi$ , del cual se esta utilizando Busy waiting, mutex. En la Figura1 se muestra las comparaciones del libro. [1].

<b>Table 4.1</b> Run-Times (in Seconds) of $\pi$ Programs Using $n = 10^8$ Terms on a System with Two Four-Core Processors		
<b>Threads</b>	<b>Busy-Wait</b>	<b>Mutex</b>
1	2.90	2.90
2	1.45	1.45
4	0.73	0.73
8	0.38	0.38
16	0.50	0.38
32	0.80	0.40
64	3.56	0.38

Threads zs	Busy Waiting	Mutex
1	1.19	1.21
2	0.80	0.82
4	0.55	0.53
8	0.52	0.51
16	0.45	0.44
32	0.50	0.39
64	0.73	0.38

Resultados obtenidos de pi con busy waiting y mutex

## 2 Conclusiones

En conclusión, Vemos que cuando utilizamos busy waiting, el rendimiento puede degradarse si hay más hilos que núcleos pero no vemos mucha diferencia en el tiempo de ejecución general cuando los programas se ejecutan con menos hilos que núcleos. Esto no debería sorprender, ya que cada hilo solo ingresa a la sección crítica una vez. Entonces si empezamos a aumentar el número de subprocesos más allá del número de núcleos, el rendimiento de la versión que usa mutex permanece prácticamente sin cambios, mientras que el rendimiento de la versión de busy waiting se degrada.

## References

- [1] Book. an introduction to parallel programming, nov 2011.