



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC3524 - Tópicos avanzados de sistemas distribuidos

Tarea 2

27 de junio de 2017

Nicolás Gebauer - 13634941

@negebauer - repo T2

Análisis de rendimiento

Se ejecutó el programa en su versión secuencial y paralela. La versión paralela se corrió con 4 procesos en cada nodo. Los nodos utilizados fueron 3, tripio, trauco y caleuche. Los tiempos de procesamiento para encontrar el camino más corto, junto con los speedups respectivos, se presentan a continuación.

Las tablas describen para cada test el tiempo que tomo ejecutando *time* en los campos *real* y *user*. S son los tiempos secuenciales, P los paralelos. Los tamaños de las pruebas son

t3 :12

t4 :16

t5 :17

t6 :18

Test	S real	S user	P real	P user
t3	0,04	0,04	1,92	0,24
t4	12,39	12,38	4,82	8,77
t5	11,14	11,13	5,27	10,84
t6	270,50	269,34	53,47	169,86

Tiempos de ejecución en segundos

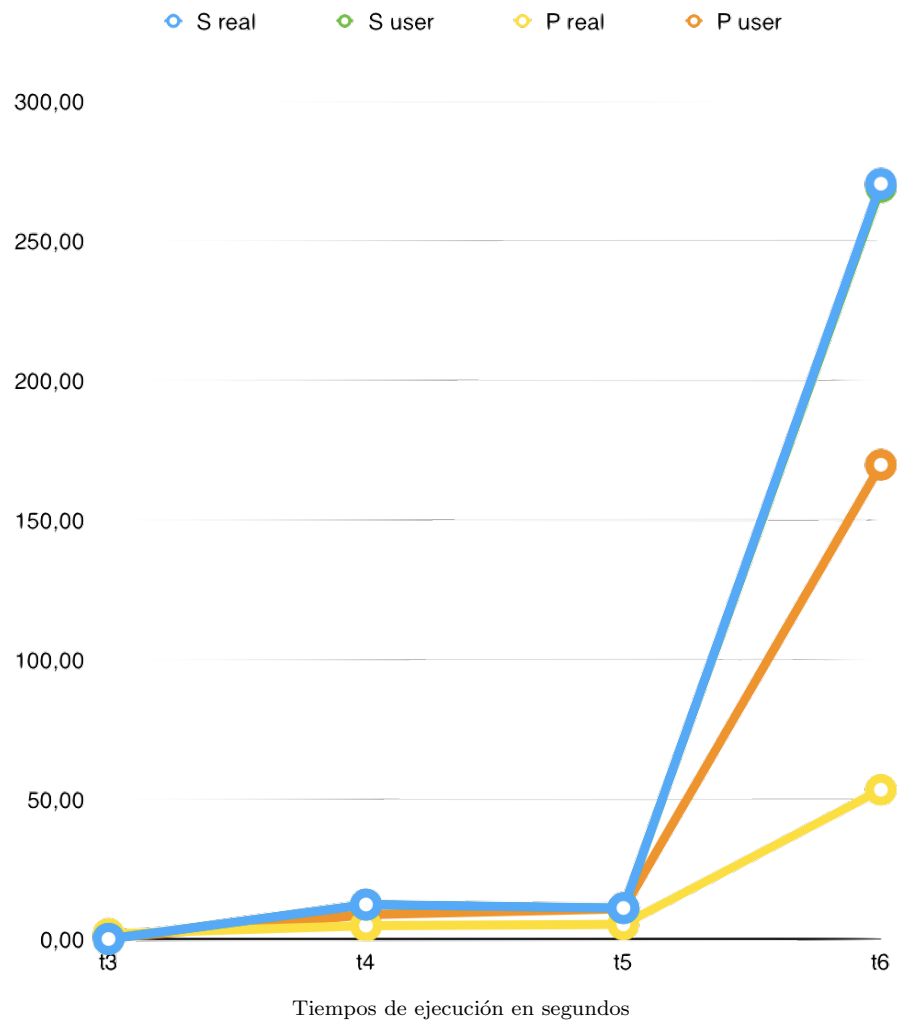
Test	S real	S user	P real	P user
t3	0,00	0,00	0,03	0,00
t4	0,21	0,21	0,08	0,15
t5	0,19	0,19	0,09	0,18
t6	4,51	4,49	0,89	2,83

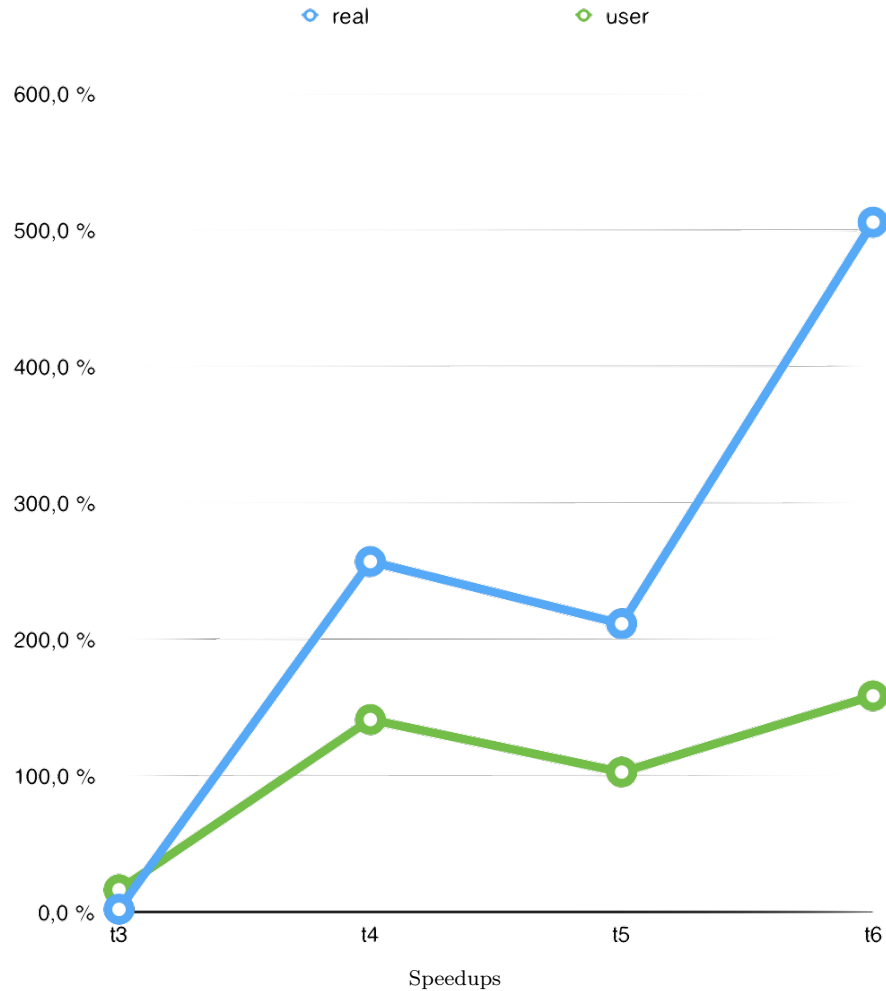
Tiempos de ejecución en minutos

Test	real	user
t3	2,0 %	16,4 %
t4	257,0 %	141,2 %
t5	211,4 %	102,7 %
t6	505,9 %	158,6 %

Speedups

Se grafican las tablas de tiempo y speedup a continuación





Discusión

Debe efectuar un análisis comparando el tiempo de ejecución, speedup y eficiencia para diferentes valores de N y cantidad de procesos MPI, y reportar gráficamente estos valores. Utilice valores de N representativos que permitan apreciar una diferencia en tiempo de ejecución. Para efectos de comparación, la versión baseline debe ser la versión MPI ejecutado usando solamente un proceso local. Debe escribir un informe donde describa sus experimentos y discuta los resultados.

Se aprecia que al aumentar el número de threads disminuye el tiempo que ocupa la CPU cada thread, esto es esperable ya que al dividir la tarea en más procesos cada proceso tendrá que trabajar menos. También el tiempo total de uso de la CPU aumenta al aumentar el número de threads. Esto también es esperable ya que al tener más procesos que corren sobre la CPU el uso total es mayor. Es importante destacar que este tiempo no es cuanto demora el programa, sino más bien cuanto tiempo de cómputo usa de la CPU (en su totalidad). A partir de los 8 threads este valor se estabiliza. Esto puede deberse a que a partir de los 8 threads la CPU se utiliza por completo. Por lo tanto, el tiempo total de uso ya no se puede disminuir. Tener más threads solo hace que cada thread trabaje menos, pero el uso total de la CPU se mantiene.

Lo importante es el speedup, el cual tiene a disminuir al aumentar la cantidad de threads. Se observa que al pasar de 2 a 3 y de 4 a 5 el speedup disminuye. Esto puede deberse a que el paralelismo funciona mejor con potencias de 2.

