

REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD SIMÓN BOLÍVAR  
CARACAS - 26/05/2014  
SITEMAS DE OPERACIÓN I

# TOYX

Prof. Yudith Cardinale.

Autor

Carnet

Wilmer Bandres

10-10055

Las la realización del programa toyx, se han observó la diferencia de ejecución entre un programa con procesos concurrentes y un programa con hilos.

Los resultados experimentales obtenidos para el programa toyx con la utilización de hilos son:

- Variando las unidades de ejecución:

Número de i-nodos	Número de unidades de ejecución	Número de particiones	Tiempo total del programa	Tiempo de ejecución de todos los hilos	Tiempo promedio de ejecución de una unidad de ejecución	Tiempo promedio de creación de una unidad de ejecución
90	3	3	2434	957	243	35
90	4	3	1802	517	105	38
90	5	3	1519	590	110	33
90	6	3	2511	951	312	22
90	7	3	2200	1000	300	30
100	3	3	2600	1001	256	60
100	5	3	1921	864	250	20
100	7	3	2371	724	200	55
100	9	3	2732	1106	350	30

100	11	3	2743	1055	370	28
-----	----	---	------	------	-----	----

- Variando el número de particiones:

Número de i-nodos	Número de unidades de ejecución	Número de particiones	Tiempo total del programa	Tiempo de ejecución de todos los hilos	Tiempo promedio de ejecución de una unidad de ejecución	Tiempo promedio de creación de una unidad de ejecución
90	3	3	3020	1497	700	20
90	3	4	2205	624	380	24
90	3	5	3662	1655	500	30
90	3	6	2422	470	130	20
90	3	7	2555	531	100	22
100	3	3	2887	1268	600	50
100	3	5	2002	387	130	20
100	3	7	2895	706	130	40
100	3	9	2617	561	150	30
100	3	11	1831	621	100	35

100	31	10	2977	579	90	20
-----	----	----	------	-----	----	----

Los resultados experimentales obtenidos para el programa toyx con la utilización de procesos son:

- Variando las unidades de ejecución:

Número de i-nodos	Número de unidades de ejecución	Número de particiones	Tiempo total del programa	Tiempo de ejecución de todos los procesos	Tiempo promedio de ejecución de una unidad de ejecución	Tiempo promedio de creación de una unidad de ejecución
90	3	3	2372	970	300	50
90	4	3	2427	1036	350	40
90	5	3	2228	917	250	70
90	6	3	2005	607	120	40
90	7	3	2819	1489	400	60
100	3	3	1406	949	300	75
100	5	3	2707	1267	350	120
100	7	3	2216	716	110	120
100	9	3	2061	603	110	70
100	11	3	2196	1444	340	50

- Variando el número de particiones:

Número de i-nodos	Número de unidades de ejecución	Número de particiones	Tiempo total del programa	Tiempo de ejecución de todos los hilos	Tiempo promedio de ejecución de una unidad de ejecución	Tiempo promedio de creación de una unidad de ejecución
90	3	3	2197	542	120	50
90	3	4	1356	772	220	30
90	3	5	1366	880	240	60
90	3	6	2616	1015	160	50
90	3	7	1483	900	100	25
100	3	3	2042	961	240	50
100	3	5	2690	1039	120	120
100	3	7	2179	957	80	60
100	3	9	2665	833	60	70
100	3	11	2546	1828	140	60
100	31	5	1016	518	100	50

Como se observa en la tablas, muchas veces varían los resultados, ya sea con la utilización de hilos o con la utilización de procesos. Lo que sí parece ser cierto, según resultados experimentales, es que a mayor cantidad de particiones, la cantidad de tiempo que dura cada procesos es menor, y eso se debe a que el rango de inodos del que se encarga es mucho menor. Sin embargo, el único problema encontrado es que aunque el tiempo total de ejecución no parece discrepar demasiado, esto pudo haber ocurrido por la ocupación del CPU donde se corrieron los experimentos. A nivel teórico, la ejecución total con hilos debería durar menos que el de procesos, debido a la gran cantidad de tiempo que se gasta duplicando el proceso y haciendo cuestiones de inicialización, mientras que con hilos los recursos son compartidos entre cada uno de los hilos que se van ejecutando.

El tiempo de ejecución de cada proceso, pasa a ser irrelevante luego de su inicialización, ya que ambos ejecutan la misma función para la actualización de los inodos en memoria secundaria. Como se observa en las tablas, el tiempo de inicialización en el caso de los hilos, en promedio, es mucho menor que en el caso de los procesos.

La razón de que los experimentos no varían demasiado con respecto al tiempo total de ejecución, también parece ser la fijación de los parámetros, ya que cuando tenemos una cantidad de particiones fijas y lo que intentamos variar es la cantidad de unidades de ejecución, pasa a ser irrelevante cuando la cantidad de unidades de ejecución supera a la cantidad de particiones. Y viceversa, pues podemos tener muchas particiones pero si la cantidad de unidades de ejecución es pequeña, no parece tener mucho efecto en el resultado.

La situación ideal para los procesos con hilos: parece no importar demasiado la cantidad de unidades de ejecución con respecto a la cantidad de particiones, sin embargo, cuando la cantidad de unidades se acerca a la mitad del número de la cantidad de particiones parece haber una mejora.

La situación ideal para los procesos cuyos hijos son procesos: parece que la cantidad de unidades de ejecución si importa, dado el tiempo que se gasta creando cada unidad, por lo que se

recomienda usar una cantidad pequeña de unidades de ejecución, con una cantidad de particiones alrededor de 6 veces la misma.

La recomendación final para la compañía es que deben utilizar procesos con hilos ya que quizás con números de inodos tan pequeños no se nota, pero con una cantidad grande de inodos la cantidad de unidades de ejecución debería ser un poco alta para que se minimice el tiempo total de ejecución, por lo que es importante que se utilicen hilos, pues la utilización de procesos es muy costosa a gran escala a la hora de crear las unidades. Y tratar de utilizar una cantidad similar a la situación ideal de los procesos con hilos.