

Laboratorio 3 Sistemas Operativos

Raúl Fuentealba

9 de Diciembre del 2016

1 Introducción

Generalmente durante el desarrollo de un laboratorio a nivel universitario nunca se pide optimizar el código o la solución en ejecución. Esto se debe a que la eficiencia del programa no siempre son medidos en el enunciado, pero en mas de una situación se han entregado laboratorios con tiempos de ejecución gigantescos o que simplemente no logran terminar su objetivo en un tiempo prudente. Si bien este es un tema que llama bastante la atención, nunca es desarrollado o simplemente nunca es tomado en cuenta, dentro de este simple laboratorio se podrá apreciar de forma muy básica pero a su vez bastante completa el como la aplicación se ve afectada por el diseño del algoritmo o dicho de otra manera el como la implementación afecta directamente el tiempo de ejecución.

Dentro de este laboratorio se realizaran 2 programas, uno que explote el principio de localidad espacial-temporal y otro que no lo hace. Esto se realizara a través de un arreglo de gran tamaño que sera recorrido de forma secuencial y otro que sera recorrido de forma aleatoria, esto para mostrar que la aplicación de ciertos principios elementales de la informática o de la arquitectura de software si afecta radicalmente el rendimiento de un programa. En este documento se presentara una introducción, seguido por los resultados obtenidos y finalmente una conclusión respecto a la experiencia.

2 Resultados

A continuación veremos la tabla con los 20 resultados obtenidos de ambos programas, esto se obtuvieron luego de la ejecución de estos programas por separado, sin ningún proceso corriendo en paralelo y todos de forma continua, el tiempo esta medido en segundos.

N Ejecución	Programa 1 [Seg]	Programa 2 [Seg]
1	5.886177	27.778355
2	5.830280	31.178998
3	5.791033	30.425362
4	5.843323	28.273686
5	5.818031	27.517374
6	5.799853	29.038626
7	5.802103	24.421331
8	5.793168	27.319970
9	5.804147	29.177448
10	5.815202	29.393049
11	5.877503	25.983683
12	6.244905	23.073966
13	5.860207	31.326233
14	5.796043	24.971006
15	6.324457	29.327921
16	6.020508	22.727253
17	6.381212	31.362728
18	5.943509	25.846270
19	5.965061	24.719000
20	5.957714	27.204101

Table 1: Tabla de ejecución de los programas.

La media aritmética junto con la varianza de este experimento se muestran continuativo, como se puede apreciar, en el caso de la media en ambos programas es el **Programa 1** el que posee un valor promedio mas pequeño, junto con su varianza que no cambia en mas de 1, en comparación al **Programa 2** cuya varianza cambia en casi 3 puntos.

Programa	Media [Seg]	Varianza
Programa 1	5.912613	0.195274
Programa 2	27.552717	2.602804

Table 2: Tabla de Media y Varianza.

3 Análisis de resultados

Como era de esperarse el **Programa 1** es el que tiene el menor tiempo de ejecución, esto lo hace mas rápido y eficiente a la hora de desarrollar algoritmos de optimización. ¿Pero a que se debe esto? Como se dijo al principio de este documento, uno de os programas explota los principios de localidad espacial y principio de localidad temporal, es precisamente esta propiedad de los computadores lo que hace que este programa sea mucho mas eficiente en cuanto a términos de tiempo.

El **Programa 1** es recorrido en orden desde el primer elemento hasta el ultimo, esto genera que el procesador haga llamados de memoria a elementos mucho muy cercanos (principio de localidad espacial), al acceder de forma mas rápido a estos segmentos de memoria hacen que se puedan paralelizar mas procesos y acceder a estos cuando ya estén listos en memoria. Básicamente, al referencia datos tan juntos el procesador puede trabajar mas de 1 a la vez y con eso tenerlos listos para cuando sean llamados. Esto se comprueba a mirar la varianza del **Programa 1** que se menor a 1 por lo que no existe mucha diferencia entre el peor caso y el mejor caso en el tiempo de ejecución.

En cambio en el **Programa 2** es recorrido de manera aleatoria y si bien se podría pensar que al referencia un dato, es mas probable volver a reverenciarlo (principio de localidad temporal), eso es un error ya que en un universo tan grande no es poco probable que sea reverenciado otra vez y el dato quedara en memoria sin uso, además que el procesador deberá llamar a cada dato de forma distinta por cada ciclo perdiendo muchos de estos ciclos. A diferencia del **Programa 1** que estos datos ya podían estar listos para ser llamados, esto también queda en evidencia mirando la varianza del **Programa 2**, que es de 2 puntos casi 3 por lo que su resultado mas bajo y el mas alto hay una gran diferencia.

4 Conclusión

Ya se había dicho al principio de este documento, pero no esta demás destacar el como se ven afectados los programas por la manera en la que están hechos, esto se debe a que la programación consiste en darle instrucciones al procesador, la forma en que se las damos, los algoritmos que usamos, el como hacemos las cosas, etc. Son factores que siempre afectaran el rendimiento de nuestro programa final, así lo queramos o no. Otro punto destacable de esta experiencia es el hecho de las instrucciones mismas, al principio se coloco en duda que los resultados arrojados por el laboratorio fueran correctos por lo que se corrió el programa mientras imprimía los valores de cada posición en sus respectivos casos, el resultado fue que demoro mas de 903.522019 [seg] en el **Programa 1** y 958.141132 [seg] en el **Programa 2**, esto deja en evidencia que hasta la función mas simple afecta significativamente el rendimiento del programa, mas con volúmenes de datos tan grandes.

Finalmente, mencionar que los dos principios mencionados en al inicio de este laboratorio, que corresponden al principio de localidad espacial y localidad temporal. Son grandes herramientas que sirven de orientación para conocer el verdadero rendimiento de la memoria a la hora de ejecutar un programa y también la realización de este laboratorio ayudo a evidenciar de mejor manera el como funcionan estos principios en la practica mas que en la teoría.