|  |  |  |
| --- | --- | --- |
| **Fecha de entrega:** 01 de diciembre de 2017 | **Título de la tarea:** Principios de localidad temporal y espacial. | **Número de la tarea:** 3.1 |
| **Grupo:** 3CM3 | **Alumno:** Estrada Granados Diego | **Unidad de aprendizaje:** Arquitectura de Computadoras |

**Principio de Localidad de Referencia.**

La propiedad más importante, que regularmente explotamos de un programa, es la **localidad de referencia**: los programas tienden a reutilizar los datos e instrucciones que han utilizado recientemente. Una regla empírica, muy corroborada, es que un programa emplea el 90 por 100 de su tiempo de ejecución en sólo el 10 por 100 del código. Una implicación de la localidad es que, basándose en el pasado reciente del programa, se puede predecir con una precisión razonable qué instrucciones y datos utilizará un programa en el futuro próximo.

Para examinar la localidad, se midieron algunos programas para determinar qué porcentaje de las instrucciones eran responsables del 80 y 90 por 100 de las instrucciones ejecutadas.

La localidad de referencia también se aplica a los accesos a los datos, aunque no tan fuertemente como a los accesos al código. Se han observado dos tipos diferentes de localidad. En la siguiente ilustración se muestra el resultado de los datos mencionados.

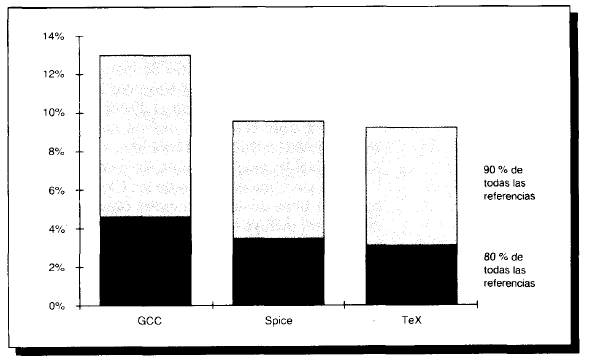


Ilustración Este dibujo muestra el porcentaje de las instrucciones que son responsables del 80 y 90 por ciento de las ejecuciones de las instrucciones.

(Hennessy & Patterson, 2002)

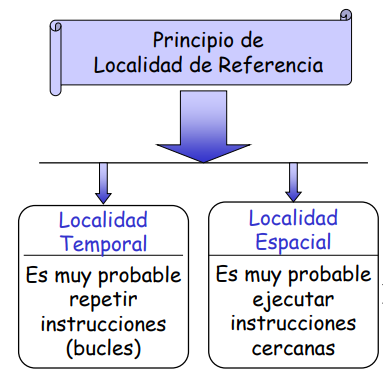
El **Principio de Localidad de Referencia** dice que los accesos a memoria que realiza la CPU no están uniformemente distribuidos por todo el espacio de direcciones, sino que, temporalmente, se concentran en áreas de memoria.

Lo que establece este principio se debe a que el contenido de cada programa no está esparcido por toda la memoria, sino que sus instrucciones y datos están contenidos en una o varias secciones de memora contigua, por lo que los accesos a la memoria se concentran en las áreas donde están las instrucciones o los datos del programa en ejecución.

El principio de localidad se manifiesta en dos aspectos: temporal y espacial.

La **Localidad de Referencia Temporal** se debe a la naturaleza repetitiva de los programas o de las porciones de los programas, lo que representa que una instrucción que se acaba de ejecutar recientemente es muy probable que se vuelva a ejecutar en un futuro próximo. Ahora, la secuencia detallada de la ejecución de las instrucciones no es significativa, lo que realmente nos importa son las instrucciones localizadas en áreas concretas del programa debido a que los bucles que las encierran, se ejecutan rápidamente durante un perdió de tiempo, mientras que al resto del programa o resto de memoria solo se accede muy de vez en cuando.

El aspecto **Espacial** quiere decir que las instrucciones se encuentran en la proximidad de una instrucción recientemente ejecutada (en cuanto a sus direcciones de memoria) también es muy probable que se ejecuten muy pronto esas direcciones cercanas. A los datos u operando del programa les sucede lo mismo, no están dispersos por toda la memoria, sino que están agrupados en una zona de la misma, y ocupando direcciones contiguas.



(Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, 2016)

# Trabajos citados

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos. (septiembre de 2016). *Departamento de Informática Aplicada de la ETSISI.* Obtenido de http://www.dia.eui.upm.es/Asignatu/arq\_com/Paco/7-Cache.pdf

Hennessy, J., & Patterson, D. (2002). *Arquitectura de Computadores Un enfoque cuantitativo.* Madrid: McGraw-Hill.