

Grupo 1:

Daniela Espinosa Albarello – 202022615 – d.espinosaa

Juan Diego Osorio Cárdenas – 202220148 – jd.osorioc1

Sebastián Palma Mogollón – 202222498 – s.palma

INFORME CASO DE ESTUDIO 2 – MEMORIA VIRTUAL

INFRAESTRUCTURA COMPUTACIONAL

Descripción del Algoritmo para Generar las Referencias de Página (Opción 1)

El algoritmo para generar las referencias de página se basa en la lectura de un archivo que contiene líneas de referencias de acceso. Cada línea representa un acceso a una página de memoria, que puede ser de lectura o escritura. El algoritmo realiza lo siguiente:

1. Lectura del Archivo: Utiliza un `BufferedReader` para leer el archivo línea por línea.
2. Filtrado de Líneas: Se descartan las líneas que comienzan con ciertas letras que no representan referencias válidas (P=, NF=, NC=, NR=, NP=).
3. Parsers:
 - Cada línea válida se divide en partes utilizando `split(",")`.
 - Se obtiene el número de página de la segunda parte y se determina si la operación es de escritura o no (el valor se encuentra en la cuarta parte).
4. Almacenamiento de Referencias: Cada referencia se almacena como un arreglo de enteros, donde el primer elemento es el número de página y el segundo elemento indica si es una operación de escritura (1 para escritura, 0 para lectura).
5. Actualización de la Lista de Referencias: Todas las referencias válidas se agregan a una lista llamada referencias.

Descripción de las Estructuras de Datos Usadas para Simular el Comportamiento del Sistema de Paginación

El sistema de paginación se simula utilizando las siguientes estructuras de datos:

1. `ArrayList<int[]>` referencias:
 - Almacena las referencias de página generadas. Cada referencia se guarda como un arreglo de enteros, donde el primer elemento es el número de página y el segundo elemento indica si es una escritura o una lectura.

- Se llena durante la ejecución del método cargarReferencias, donde se agregan nuevas referencias leídas del archivo.

2. HashMap<Integer, int[]> ram:

- Simula la memoria RAM, donde la clave es el número de página y el valor es un arreglo de enteros que contiene dos bits: el primer bit indica si la página está presente en memoria (1) o no (0), y el segundo bit indica si la página ha sido escrita (1) o solo leída (0).
- Se actualiza dentro de las clases ActualizadorReferencias y reemplazarPagina, cuando se accede a una página, se contabilizan hits y fallas, y se reemplazan páginas según sea necesario.

Esquema de Sincronización Usado

La sincronización se utiliza para garantizar la consistencia de los datos cuando múltiples hilos intentan acceder y modificar la estructura de datos ram. Se usa synchronized(ram) cuando se accede a la memoria RAM (el HashMap) desde diferentes hilos (ActualizadorReferencias y ActualizadorBitR).

Clase ActualizadorReferencias: Accede a ram para contar hits y fallas, así como para modificar la estructura al agregar nuevas páginas.

Clase ActualizadorBitR: Accede a ram para actualizar el estado de los bits de acceso de todas las páginas. Sincronizar el acceso aquí asegura que no haya interferencias con los hits y fallas contados por ActualizadorReferencias.

La sincronización es necesaria aquí para prevenir condiciones de carrera, donde múltiples hilos podrían leer o escribir simultáneamente en la estructura de datos, lo que podría llevar a resultados inconsistentes. Por ejemplo, si un hilo está verificando si una página está presente mientras otro hilo está reemplazando esa misma página, podrían surgir inconsistencias.

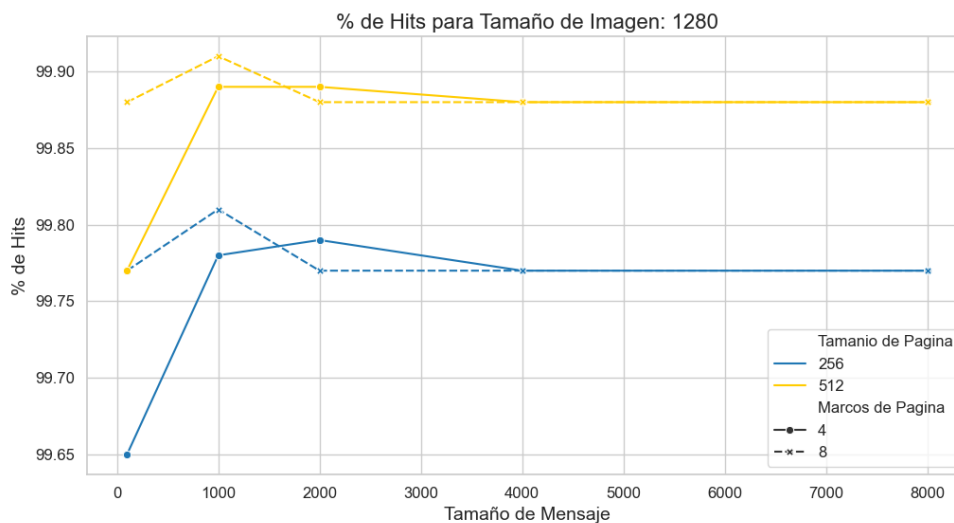
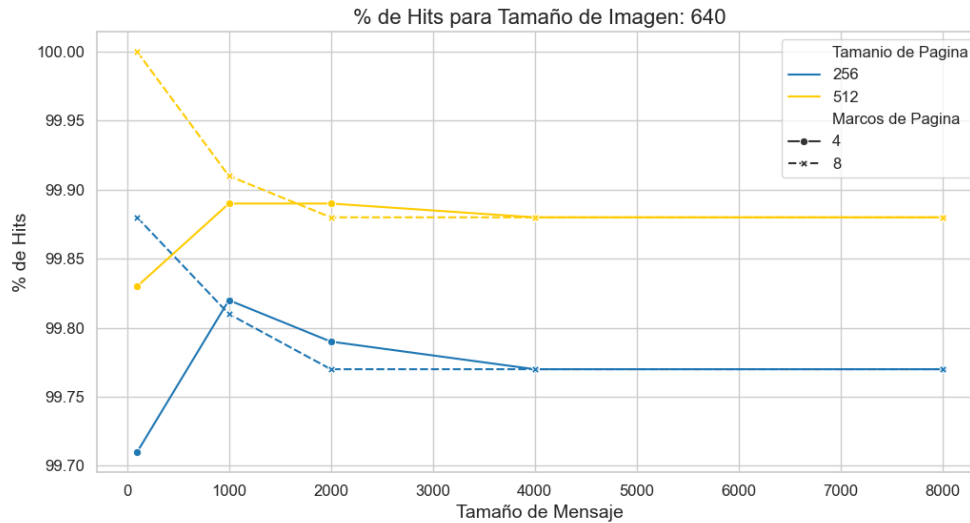
Tabla de los datos recopilados por cada escenario

Tamaño Imagen (px*px)	Tamaño Mensaje (caracteres)	# Marcos de Página	Tamaño de Página (Bytes)	Fallas de Página	% Fallas	Hits	% Hits	Tiempo (ms)
640	100	4	256	5	0,29%	1711	99,71%	50
640	100	4	512	3	0,17%	1713	99,83%	30
640	100	8	256	2	0,12%	1714	99,88%	20
640	100	8	512	0	0,00%	1716	100,00%	0
640	1000	4	256	31	0,18%	16985	99,82%	310
640	1000	4	512	18	0,11%	16998	99,89%	180

640	1000	8	256	33	0,19%	16983	99,81%	330
640	1000	8	512	16	0,09%	17000	99,91%	160
640	2000	4	256	70	0,21%	33946	99,79%	700
640	2000	4	512	38	0,11%	33978	99,89%	380
640	2000	8	256	79	0,23%	33937	99,77%	790
640	2000	8	512	40	0,12%	33976	99,88%	400
640	4000	4	256	158	0,23%	67858	99,77%	1581
640	4000	4	512	79	0,12%	67937	99,88%	791
640	4000	8	256	158	0,23%	67858	99,77%	1581
640	4000	8	512	79	0,12%	67937	99,88%	791
640	8000	4	256	315	0,23%	135701	99,77%	1141
640	8000	4	512	158	0,12%	135858	99,88%	1583
640	8000	8	256	315	0,23%	135701	99,77%	1141
640	8000	8	512	158	0,12%	135858	99,88%	1583
1280	100	4	256	6	0,35%	1710	99,65%	60
1280	100	4	512	4	0,23%	1712	99,77%	40
1280	100	8	256	4	0,23%	1712	99,77%	40
1280	100	8	512	2	0,12%	1714	99,88%	20
1280	1000	4	256	38	0,22%	16978	99,78%	380
1280	1000	4	512	18	0,11%	16998	99,89%	180
1280	1000	8	256	32	0,19%	16984	99,81%	320
1280	1000	8	512	16	0,09%	17000	99,91%	160
1280	2000	4	256	71	0,21%	33945	99,79%	710
1280	2000	4	512	38	0,11%	33978	99,89%	380
1280	2000	8	256	79	0,23%	33937	99,77%	790
1280	2000	8	512	40	0,12%	33976	99,88%	400
1280	4000	4	256	158	0,23%	67858	99,77%	1581
1280	4000	4	512	79	0,12%	67937	99,88%	791
1280	4000	8	256	158	0,23%	67858	99,77%	1581
1280	4000	8	512	79	0,12%	67937	99,88%	791
1280	8000	4	256	315	0,23%	135701	99,77%	1141
1280	8000	4	512	158	0,12%	135858	99,88%	1583
1280	8000	8	256	315	0,23%	135701	99,77%	1141
1280	8000	8	512	158	0,12%	135858	99,88%	1583

Gráficas del comportamiento del sistema:

Se crearon gráficos de líneas para mostrar la variación del porcentaje de hits para cada tamaño de imagen, con respecto a los diferentes tamaños de mensaje, números de marcos y tamaños de página.



A lo largo de los diferentes tamaños de mensaje, se observa que el porcentaje de hits tiende a ser bastante alto, generalmente por encima del 99%. Esto indica que, independientemente de las condiciones, el sistema de memoria virtual tiene un rendimiento eficiente en la recuperación de información.

Al comparar los gráficos de los dos tamaños de imagen (640 y 1280), el porcentaje de hits muestra una tendencia similar, pero con ligeras variaciones. En general, el tamaño de imagen más grande (1280) presenta un comportamiento levemente inferior en el porcentaje de hits en comparación con el tamaño de imagen más pequeño (640) en la mayoría de las combinaciones. Esto podría

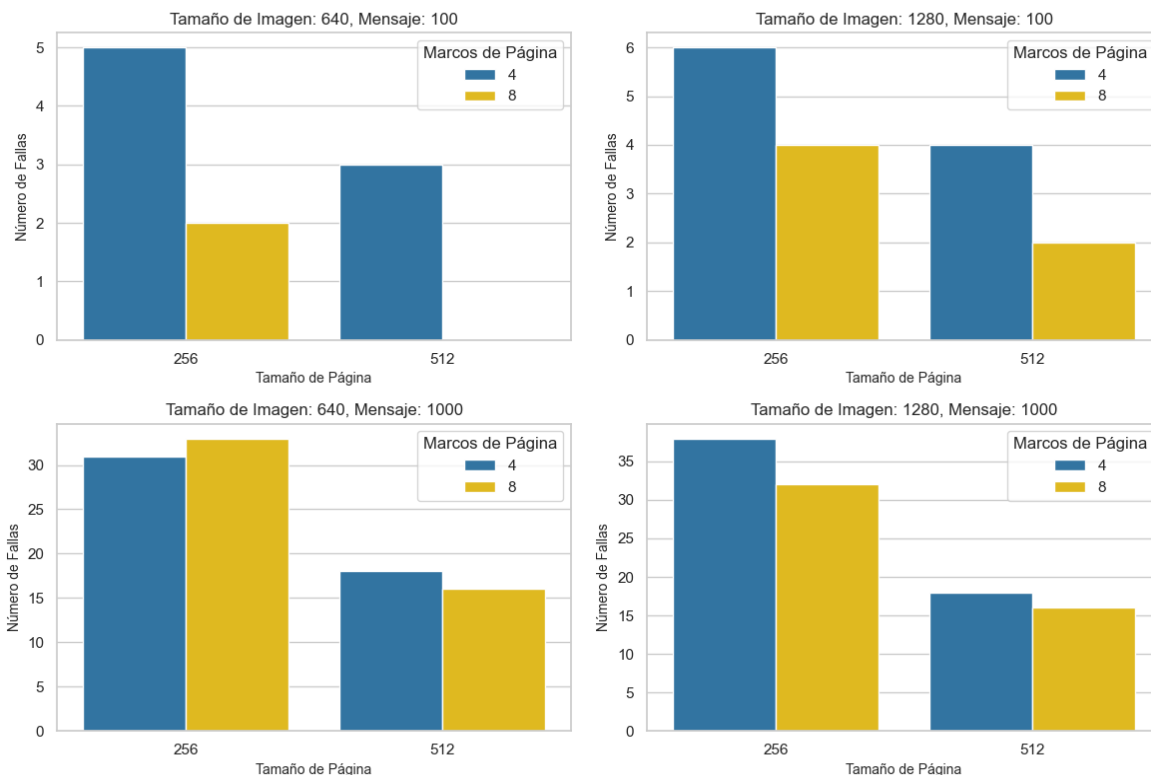
sugerir que a medida que aumenta el tamaño de imagen, puede haber una mayor complejidad en el manejo de los datos, lo que podría afectar la eficiencia del sistema.

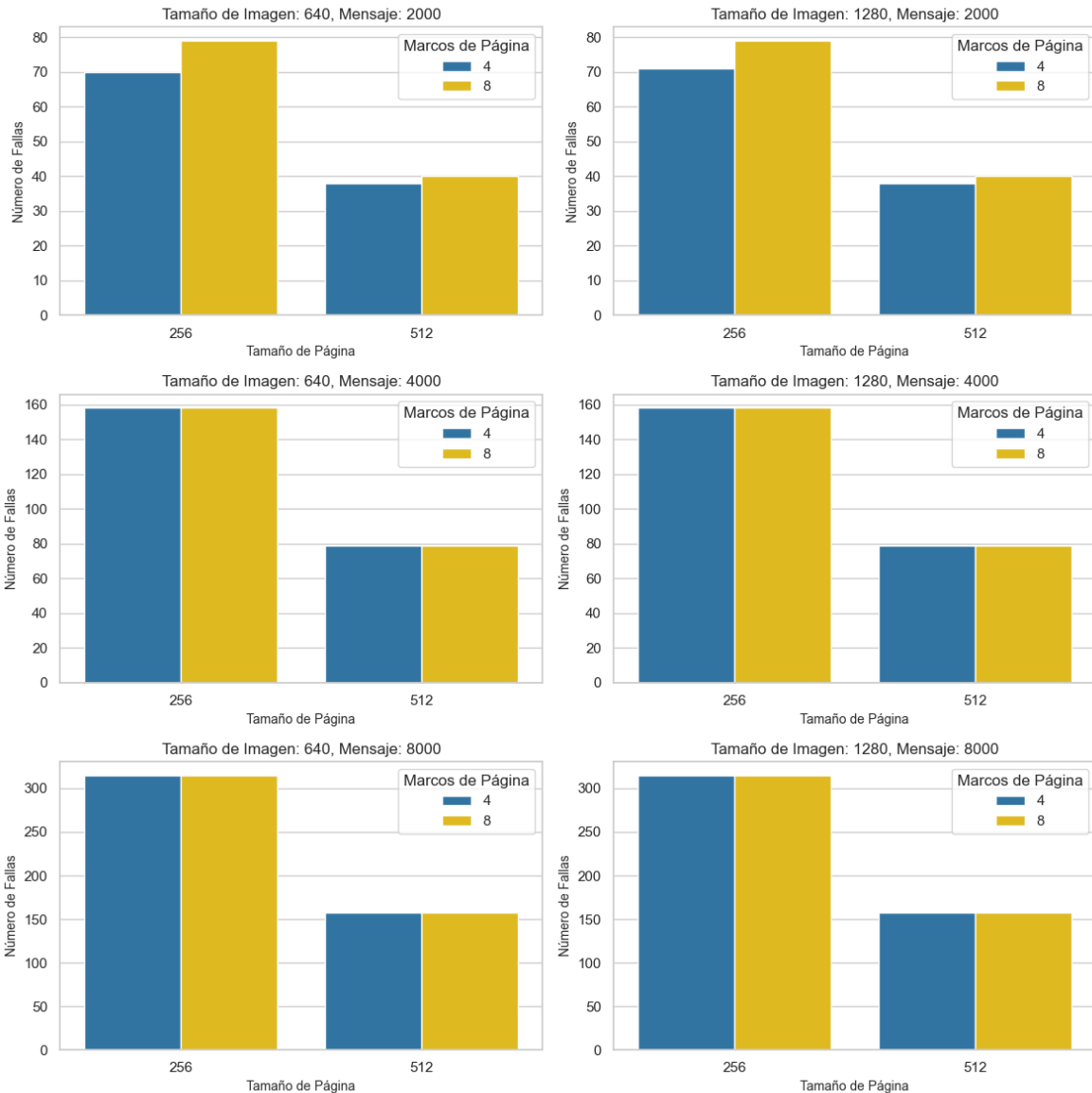
A medida que el tamaño del mensaje aumenta, el porcentaje de hits tiende a fluctuar. En algunos casos, como los mensajes de tamaño 100 y 1000 caracteres, se mantienen altos, mientras que para mensajes más grandes (4000 y 8000 caracteres), el porcentaje de hits se reduce ligeramente. Esto podría indicar que mensajes más grandes presentan más desafíos para ser almacenados y recuperados eficientemente en comparación con mensajes más pequeños.

Los gráficos también destacan cómo los diferentes números de marcos de página (4 y 8) y los tamaños de página (256 y 512) afectan el rendimiento. Por lo general, los tamaños de página más grandes (512) tienden a mostrar un porcentaje de hits ligeramente mejor que los tamaños más pequeños (256) en varios escenarios. Esto sugiere que un mayor tamaño de página puede facilitar una mejor asignación y manejo de la memoria, resultando en una menor cantidad de fallas de página.

La variación entre 4 y 8 marcos de página es menos pronunciada, aunque se puede observar que con 8 marcos de página en general resulta en porcentajes de hits similares o ligeramente mejores, lo que implica que más marcos pueden ofrecer una mejor flexibilidad en la gestión de la memoria.

Para profundizar el análisis, se generaron histogramas para mostrar el número de fallas para cada tamaño de imagen, tamaño de mensaje, número de marcos de página y tamaño de página.





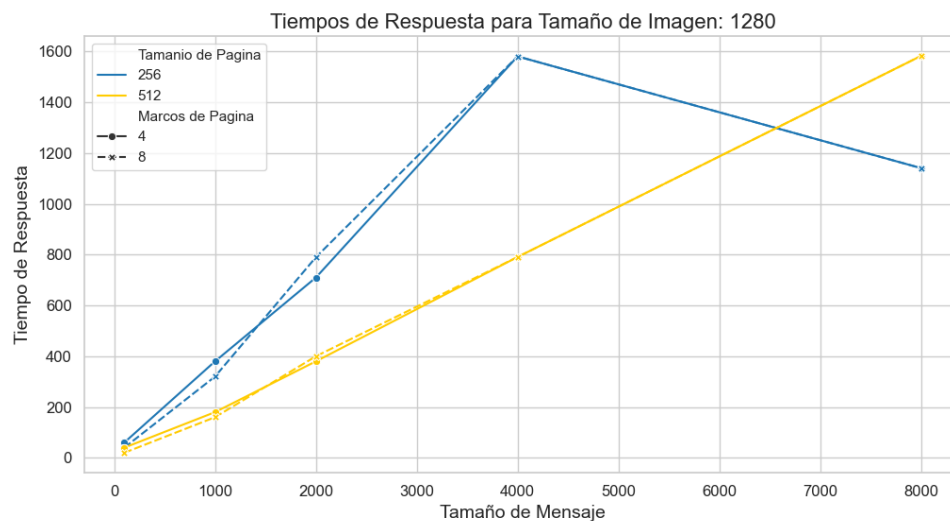
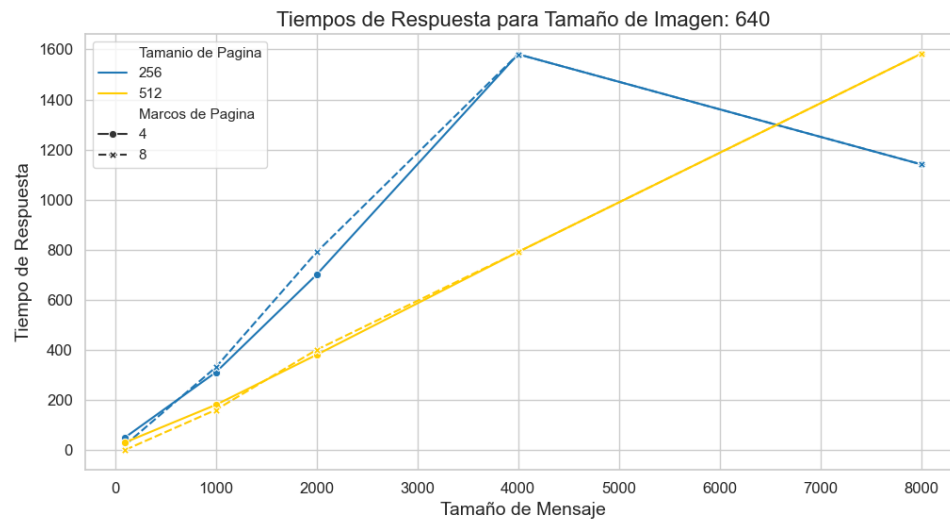
Los histogramas revelan patrones claros en la distribución de las fallas de página. Por ejemplo, se puede observar que los histogramas correspondientes a mensajes de mayor tamaño (4000 y 8000 caracteres) presentan números de fallas significativamente más altos en comparación con los mensajes más pequeños (100 y 1000 caracteres): con los mensajes de mayor tamaño se presentan entre 80 y 300 fallas, mientras que con los de menor tamaño se presentan entre 0 y 30 fallas.

También se puede notar el claro impacto que genera el tamaño de página, pues las páginas de mayor tamaño (512 en comparación con 256) tiende a reducir el número de fallas de página en la mayoría de los casos, lo que sugiere que un mayor tamaño de página puede mejorar la eficiencia del sistema al reducir la fragmentación.

Además, para los mensajes de menor tamaño, se evidencia que los de 8 marcos presentan en general menos fallas de página, por lo que se puede concluir que una mayor cantidad de marcos contribuye a un rendimiento más eficiente. Sin embargo, esto puede variar, ya que para los

mensajes de 2000 caracteres los de 8 marcos presentaron más fallas, y luego a medida que el mensaje aumenta de tamaño (4000 y 8000), los números de fallas son constantes sin importar el número de marcos.

Por último, se crearon gráficos de líneas para mostrar cómo varían los tiempos de respuesta para cada tamaño de imagen, con respecto a diferentes tamaños de mensaje, tamaños de página y número de marcos.



Se puede observar que en este caso los tamaños de imagen no tienen un impacto tan significativo en los tiempos de respuesta, ya que para ambas imágenes los valores de tiempo son bastante similares, o iguales. El número de marcos tampoco parece tener un impacto significativo en los tiempos de respuesta, ya que en ambos casos se comportan de manera muy similar, con fluctuaciones mínimas.

Por otro lado, el tiempo de respuesta muestra una tendencia a aumentar con el aumento del tamaño del mensaje, excepto por los casos con tamaño de página de 256 y tamaño de mensaje de 8000 caracteres, en los que los tiempos son ligeramente inferiores que los de mensajes con 4000 caracteres. Los tamaños de página también impactan en los tiempos de respuesta, ya que en general las imágenes con tamaño de página de 256 presentan tiempos de respuesta mayores que las de tamaño de página de 512.

Interpretación de los resultados

¿Corresponden a los resultados que esperaba, con respecto al número de marcos asignados?

Sí, los resultados corresponden a lo que se esperaba con relación al número de marcos asignados. Aumentar el número de marcos de página de 4 a 8 claramente ha llevado a una reducción en el número de fallas de página. Esto se debe a que, con más marcos disponibles, el sistema puede mantener más páginas en la memoria, disminuyendo así la necesidad de acceder a la memoria secundaria y, por lo tanto, el número de fallas. La tendencia observada en los datos respalda la teoría de que una mayor cantidad de marcos de página mejora la eficiencia del sistema en la gestión de memoria.

¿Si la localidad del problema manejado fuera diferente cómo variarían los resultados?

Si la localidad del problema fuera mayor, se esperaría que la eficiencia del sistema mejorara, ya que una mayor localidad implica que las referencias a las páginas de memoria son más cercanas entre sí. Esto podría resultar en menos fallas de página, ya que las páginas utilizadas por un proceso tienden a estar más agrupadas, permitiendo que el sistema mantenga estas páginas en la memoria durante más tiempo. En este caso, los histogramas mostrarían una disminución en el número de fallas para las combinaciones de tamaño de imagen y mensaje.

Por otro lado, si la localidad fuera menor, esto implicaría que las referencias a la memoria están más dispersas. En este escenario, el sistema enfrentaría más dificultades para mantener las páginas necesarias en la memoria, resultando en un aumento en el número de fallas de página. Esto podría reflejarse en los histogramas como un aumento en el número de fallas para todas las combinaciones de tamaño de imagen y mensaje, especialmente en aquellos con mensajes grandes, ya que el sistema tendría que acceder a la memoria secundaria con mayor frecuencia.