de la jerarquía de memoria en la determinación de las prestaciones del sistema significa que esta importante área continuará siendo por algunos años un punto de concentración tanto para diseñadores como para investigadores.

5.13 Perspectiva histórica y lecturas recomendadas

La sección de historia proporciona una visión de conjunto de las tecnologías de las memorias, desde las líneas de retardo de mercurio a las DRAM, pasando por la invención de la jerarquía de memoria y los mecanismos de protección, y memoria virtual, y concluye con una breve historia de los sistemas operativos, incluyendo CTSS, MULTICS, UNIX, BSD UNIX, MS-DOS, Windows y Linux.

5.14 Ejercicios

Contribución de Jichuan Chang, Jacob Leverich, Kevin Lim and Parthasarathy Ranganathan (todos de Hewlett-Packard)

Ejercicio 5.1

En este ejercicio se considera la jerarquía de memoria en las aplicaciones mostradas en la siguiente tabla.

a.	Búsquedas en la web
b.	Banca electrónica

5.1.1 [10]<5.1> Suponiendo que tanto los clientes como los servidores están involucrados en el proceso, nombre en primer lugar los sistemas cliente y servidor. ¿Dónde se pueden poner caches para acelerar el proceso?

5.1.2 [10]<5.1> Diseñe una jerarquía de memoria para el sistema. Muestre la capacidad y la latencia típica en varios niveles de la cache. ¿Qué relación hay entre la capacidad de la cache y la latencia de acceso?

5.1.3 [15]<5.1> ¿Cuál es la unidad de transferencia de datos entre niveles de la jerarquía? ¿Cuál es la relación entre posición del dato, tamaño del dato y latencia de la transferencia?

5.1.4 [10]<5.1, 5.2> El ancho de banda en la comunicación y en el procesamiento del servidor son dos factores importantes a tener en cuenta en el diseño de una jerarquía de memoria. ¿Qué anchos de banda pueden ser un factor limitante en este ejemplo? ¿Cómo se pueden mejorar y a qué coste?

5.1.5 [5]<5.1, 5.8> Considere ahora que hay varios clientes accediendo simultáneamente al servidor, ¿se mejorará en este escenario la localidad espacial y temporal?

5.1.6 [10]<5.1, 5.8> Dé un ejemplo en que la cache proporcione datos anticuados. ¿Cómo se puede reducir o evitar este problema?

Ejercicio 5.2

En este ejercicio se explora la localidad de memoria del cálculo con matrices. En el siguiente código C, los elementos de la misma fila se almacenan de forma contigua.

a.
```
for (I=0; I<8000; I++)
  for (J=0; J<8; J++)
    A[I][J]=B[J][0]+A[J][I];
```

b.
```
for (J=0; J<8; I++)
  for (I=0; I<8; I++)
    A[I][J]=B[J][0]+A[J][I];
```

5.2.1 [5]<5.1> ¿Cuántos enteros de 32 bits pueden almacenarse en una línea de cache de 16 bytes?

5.2.2 [5]<5.1> ¿Cuáles de las referencias a variables del código C anterior muestran localidad temporal?

5.2.3 [10]<5.1> ¿Cuáles de las referencias a variables del código C anterior muestran localidad espacial?

La localidad se ve afectada por el orden de las referencias a memoria y por la distribución de los datos. Este mismo cálculo puede escribirse en Matlab, y la diferencia con C es que los elementos de una misma columna se almacenan de forma contigua en memoria.

a.
```
for I=1:8000
  for J=1:8
    A(I,J)=B(J,0)+A(J,I);
  end
end
```

b.
```
for J=1:8
  for I=1:8
    A(I,J)=B(J,0)+A(J,I);
  end
end
```

5.2.4 [10]<5.1> ¿Cuántas líneas de cache de 16 bytes son necesarias para almacenar todos los elementos de 32 bits que se referencian de la matriz?

5.2.5 [5]<5.1> ¿Qué referencias muestran localidad temporal?

5.2.6 [10]<5.1> ¿Qué referencias muestran localidad espacial?

Ejercicios 5.3

Las caches son importantes para conseguir una jerarquía de memoria de altas prestaciones en los procesadores. En la tabla se muestra una lista de referencias a direcciones de memoria de 32 bits

a.	1, 134, 212, 1, 135, 213, 162, 161, 2, 44, 41, 221
b.	6, 214, 175, 214, 6, 84, 65, 174, 64, 105, 85, 215

5.3.1 [10]<5.2> Dada una cache de correspondencia directa con 16 bloques de una palabra indique, para cada una de estas referencias, la dirección binaria, la etiqueta y el índice. Indique también si es un fallo o un acierto, suponiendo que inicialmente la cache está vacía.

5.3.2 [10]<5.2> Dada una cache de correspondencia directa con bloques de dos palabras y un total de ocho bloques indique, para cada una de estas referencias, la dirección binaria, la etiqueta y el índice. Indique también si es un fallo o un acierto, suponiendo que inicialmente la cache está vacía.

5.3.3 [20]<5.2, 5.3> Optimice el diseño de la cache para las referencias anteriores. Hay tres diseños posibles de una cache de correspondencia directa y una capacidad de ocho palabras: C1 tiene bloques de una palabra, C2 bloques de dos palabras y C3 bloques de cuatro palabras. ¿Cuál es el mejor diseño en términos de frecuencia de fallos? Si la parada por fallo es de 25 ciclos y C1 tiene un tiempo de acceso de 2 ciclos, C2 de 3 ciclos y C3 de 5 ciclos, ¿cuál es el mejor diseño?

Hay muchos parámetros diferentes que son importantes en las prestaciones finales de una cache. En la tabla se muestran algunos parámetros para diferentes caches de correspondencia directa.

	Tamaño de la cache de datos	Tamaño del bloque de cache	Tiempo de acceso a la cache
a.	64 KB	1 palabra	1 ciclo
b.	64 KB	2 palabras	2 ciclos

5.3.4 [15]<5.2> Calcule el número total de bits de la cache suponiendo direcciones de 32 bits. Dado este tamaño total, determine el tamaño más próximo al

tamaño dado de la cache de correspondencia directa con bloques de 16 palabras, siendo este tamaño igual o mayor al dado. Explique por qué la segunda cache, a pesar de tener bloques de mayor tamaño, puede proporcionar peores prestaciones que la primera.

5.3.5 [20]<5.2, 5.3> Genere una secuencia de accesos de lectura con una frecuencia de fallos en una cache de 2 KB asociativa por conjuntos de dos vías menor que en la cache de la tabla. Identifique una posible solución para que la frecuencia de fallos de la cache de la tabla sea menor o igual que la de la cache de 2 KB. Discuta las ventajas y desventajas de esta solución.

5.3.6 [15]<5.2> La fórmula de la página 457 muestra la forma típica para indexar una cache de correspondencia directa, específicamente (dirección del bloque) módulo (número de bloques en la cache). Suponiendo direcciones de 32 bits y 1024 bloques en la cache, considere una función diferente, específicamente (dirección del bloque[31:27]) XOR (dirección del bloque[26:22]). ¿Es posible usar esta función para indexar una cache de correspondencia directa? En caso afirmativo, explicar por qué y discuta cualquier cambio que pueda ser necesario hacer en la cache. En caso negativo, explicar por qué.

Ejercicio 5.4

En una cache de correspondencia directa con direcciones de 32 bits, los bits de la dirección se usan como se indica en la tabla.

	Etiqueta	Índice	Desplazamiento
a.	31-10	9-4	3-0
b.	31-12	11-15	4-0

5.4.1 [5]<5.2> ¿Qué tamaño (en palabras) tiene la línea de cache?

5.4.2 [5]<5.2> ¿Cuántas entradas tiene la cache?

5.4.3 [5]<5.2> ¿Cuál es la relación entre el número total de bits de la cache y el número de bits de almacenamiento?

Inmediatamente después de encender el computador, se producen las siguientes referencias a la cache, expresadas como direcciones de byte

Dirección	0	4	16	132	232	160	1024	30	140	3100	180	2180

5.4.4 [10]<5.2> ¿Cuántos bloques se reemplazan?

5.4.5 [10]<5.2> ¿Cuál es la razón de aciertos?

5.4.6 [20]<5.2> Muestre el estado final de la cache, representando cada entrada válida con <índice, etiqueta, dato>.

Ejercicio 5.5

Recuerde que hay dos estrategias de escritura y dos estrategias de reserva de escritura, y que pueden implementarse varias combinaciones tanto en la L1 como en la L2.

	L1	L2
a.	Escritura retardada con reserva de escritura	Escritura directa sin reserva de escritura
b.	Escritura retardada con reserva de escritura	Escritura directa con reserva de escritura

5.5.1 [5]<5.2, 5.5> Para reducir la latencia de acceso se sitúan búferes entre diferentes niveles de la jerarquía de memoria. Para la configuración de la tabla, indique los búferes necesarios entre las caches L1 y L2, y entre la cache L2 y la memoria.

5.5.2 [20]<5.2, 5.5> Describa el procedimiento para procesar un fallo de escritura en L1, considerando los componentes involucrados y la posibilidad de reemplazar un bloque no consistente.

5.5.3 [20]<5.2, 5.5> Describa el procedimiento para procesar un fallo de escritura en L1, considerando los componentes involucrados y la posibilidad de reemplazar un bloque no consistente, para una cache multinivel exclusiva (un bloque puede residir en la L1 o en el L2, pero no en ambas).

Considere las conductas de programa y cache de la tabla.

	Lectura de datos cada 1000 instrucciones	Escritura de datos cada 1000 instrucciones	Frecuencia de fallos de instrucciones	Frecuencia de fallos de datos	Tamaño de bloque (byte)
a.	200	160	0.20%	2%	8
b.	180	120	0.20%	2%	16

5.5.4 [5]<5.2, 5.5> ¿Cuáles son los anchos de banda de lectura y escritura mínimos (medido en bytes por ciclo) necesarios para obtener un CPI = 2 en una cache de escritura directa con reserva de memoria?

5.5.5 [5]<5.2, 5.5> ¿Cuáles son los anchos de banda de lectura y escritura mínimos (medido en bytes por ciclo) necesarios para obtener un CPI = 2 en una cache de escritura retardada con reserva de memoria, suponiendo que el 30% de los bloques de datos de cache reemplazados son no consistentes?

5.5.6 [5]<5.2, 5.5> ¿Cuáles son los anchos de banda de lectura y escritura mínimos necesarios para obtener un CPI = 1.5?

Ejercicio 5.6

Las aplicaciones multimedia que reproducen ficheros de audio y vídeo son parte de un tipo de carga de trabajo que se denominan cargas de trabajo de *streaming*; es decir, utilizan grandes cantidades de datos pero con poca reutilización. Considere una carga de trabajo de este tipo que accede a un conjunto de trabajo de 512 KB secuencialmente con las siguientes direcciones:

```
0, 4, 8, 12, 16, 20, 24, 28, 32, ...
```

5.6.1 [5]<5.5, 5.3> Suponga una cache de 64 KB de correspondencia directa con líneas de 32 bytes. ¿Cuál es la frecuencia de fallos con las direcciones de la tabla? Indique si esta frecuencia de fallos es muy sensible al tamaño de la cache o del conjunto de trabajo. ¿Cómo se clasificarían los fallos con esta carga de trabajo según el modelo de las tres C?

5.6.2 [5]<5.5, 5.1> Recalcule la frecuencia de fallos para líneas de 16, 64 y 128 bytes. ¿Qué tipo de localidad se está explotando?

5.6.3 [10]<5.5, 5.1> La prebúsqueda es una técnica que carga de manera especulativa, y basándose en patrones de direcciones predecibles, líneas adicionales en la cache cuando se accede a una línea de cache particular. Un ejemplo de prebúsqueda sería un búfer de *streaming*, que prebusca secuencialmente líneas de cache adyacentes y las almacena en un búfer separado cuando se trae una línea particular a la cache. Si el dato está en el búfer de prebúsqueda, se considera un acierto, se carga en la cache y se prebusca la siguiente línea. Suponga un búfer de *streaming* de dos entradas y suponga que la latencia de la cache es tal que puede cargarse una línea de cache antes de que se complete el cálculo sobre la línea de cache anterior. ¿Qué frecuencia de fallos se produce con la secuencia de la tabla?

El tamaño del bloque de cache (B) puede afectar a la frecuencia de fallos y a la latencia de los fallos. Tomando las frecuencias de fallos para varios tamaños de bloque de la tabla, y suponiendo un CPI = 1 con una media de 1.35 referencias a memoria (instrucciones y datos) por instrucción, ayude a encontrar el tamaño de bloque óptimo dadas las siguientes tasas de fallos para distintos tamaños de bloque.

	8	16	32	64	128
a.	8%	3%	1.8%	1.5%	2%
b.	4%	4%	3%	1.5%	2%

5.6.4 [10]<5.2> ¿Cuál es el tamaño de bloque óptimo para una latencia de fallos de 20 × B ciclos?

5.6.5 [10]<5.2> ¿Cuál es el tamaño de bloque óptimo para una latencia de fallos de 24 + B ciclos?

5.6.6 [10]<5.2> ¿Cuál es el tamaño de bloque óptimo para una latencia de fallos constante?

Ejercicio 5.7

En este ejercicio se explora cómo la capacidad afecta a las prestaciones globales. En general, el tiempo de acceso a la cache es proporcional a la capacidad. Suponga que el tiempo de acceso a la memoria principal es 70 ns y que el 36% de las instrucciones son accesos a memoria. La siguiente tabla muestra datos de caches L1 de dos procesadores P1 y P2.

		Tamaño de la L1	Frecuencia de fallos de la L1	Tiempo de acierto en la L1
a.	P1	1 KB	11.4%	0.62 ns
	P2	2 KB	8.0%	0.66 ns
b.	P1	8 KB	4.3%	0.96 ns
	P2	16 KB	3.4%	1.08 ns

5.7.1 [5]<5.3> Suponiendo que el tiempo de acierto de la L1 determina la duración del ciclo de P1 y P2, ¿cuáles son la frecuencias de reloj?

5.7.2 [5]<5.3> ¿Cuál es el AMAT de P1 y P2?

5.7.3 [5]<5.3> Suponiendo un CPI base igual a 1, ¿cuál es el CPI de P1 y P2? ¿Qué procesador es más rápido?

En los tres problemas siguientes se añade una cache a P1, presumiblemente para completar la capacidad limitada de la L1, con las siguientes características.

	Tamaño de la L2	Frecuencia de fallos de la L2	Tiempo de acierto de la L2
a.	512 KB	98%	3.22 ns
b.	4 MB	73%	11.48 ns

5.7.4 [10]<5.3> ¿Cuál es el AMAT de P1 con la cache L2?

5.7.5 [10]<5.3> Suponiendo un CPI base igual a 1, ¿cuál es el CPI de P1 con la cache L2?

5.7.6 [10]<5.3> ¿Qué procesador es más rápido ahora que P1 tiene una cache L2? Si P1 es más rápido, ¿cuál debería ser la frecuencia de fallos en la L1 de P2 para igualar las prestaciones de P1? Si P2 es más rápido, ¿cuál debería ser la frecuencia de fallos en la L1 de P1 para igualar las prestaciones de P2?

Ejercicio 5.8

En este ejercicio se analiza el impacto de los diferentes diseños de cache, específicamente se compara las caches asociativas con las de correspondencia directa de la sección 5.2. Para este ejercicio se utiliza la tabla con direcciones de referencias a memoria del ejercicio 5.3.

5.8.1 [10]<5.3> Utilizando las referencias del ejercicio 5.3, muestre el contenido final de una cache asociativa por conjuntos de tres vías, con bloques de dos palabras y un tamaño total de 24 palabras. Utilice la estrategia de reemplazo LRU. Para cada referencia indicar los bits del campo índice, del campo de etiqueta, el desplazamiento y si es un fallo o un acierto.

5.8.2 [10]<5.3> Utilizando las referencias del ejercicio 5.3, muestre el contenido final de una cache totalmente asociativa, con bloques de una palabra y un tamaño total de ocho palabras. Utilice la estrategia de reemplazo LRU. Para cada referencia, indique los bits del campo índice, del campo de etiqueta, el desplazamiento y si es un fallo o un acierto.

5.8.3 [15]<5.3> Utilizando las referencias del ejercicio 5.3, ¿cuál es la frecuencia de fallos en una cache totalmente asociativa, con bloques de dos palabras y un tamaño total de ocho palabras, utilizando la estrategia de reemplazo LRU? ¿Cuál es la frecuencia de fallos utilizando una estrategia de reemplazo MRU (usada más recientemente)? ¿Cuál es la mejor frecuencia de fallos posible para esta cache, con cualquier estrategia de reemplazo?

Las cache multinivel es una técnica que permite superar las limitaciones en espacio de las caches de primer nivel manteniendo su velocidad. Consideremos un procesador con los siguientes parámetros:

	CPI base, sin paradas de memoria	Velocidad del procesador	Tiempo de acceso a memoria principal	Frecuencia de fallos de la cache L1 por instrucción	Cache de segundo nivel de correspondencia directa, velocidad	Frecuencia de fallos global con una cache de segundo nivel de correspondencia directa	Cache de segundo nivel asociativa por conjuntos de ocho vías, velocidad	Frecuencia de fallos global con una cache de segundo nivel asociativa por conjuntos de ocho vías
a.	2.0	3 GHz	125 ns	5%	15 ciclos	3.0%	25 ciclos	1.8%
b.	2.0	1 GHz	100 ns	4%	10 ciclos	4.0%	20 ciclos	1.6%

5.8.4 [10]<5.3> Calcule el CPI del procesador de la tabla usando: 1) sólo una cache de primer nivel, 2) una cache de segundo nivel de correspondencia directa, 3) una cache de segundo nivel asociativa por conjuntos de ocho vías. ¿Cómo varían estos resultados si se dobla el tiempo de acceso a memoria principal? ¿Y si se reduce a la mitad?

5.8.5 [10]<5.3> Es posible tener una jerarquía de memoria con más de dos niveles de cache. Dado el procesador anterior con una cache de segundo nivel de correspondencia directa, se desea añadir un tercer nivel de cache con un tiempo de acceso de 50 ciclos y que reduce la frecuencia de fallos global al 1.3%. ¿Se mejorarían las prestaciones? En general, ¿cuáles son las ventajas y desventajas de añadir un tercer nivel de cache?

5.8.6 [20]<5.3> En procesadores antiguos, como el Pentium o el Alpha 21264, el segundo nivel cache era externo (situado en un chip diferente) al procesador principal y al primer nivel de cache. Esto permitía disponer de caches de segundo nivel de gran capacidad pero la latencia de acceso era mucho mayor y el ancho de banda típicamente era menor debido a que su frecuencia de reloj era más baja. Suponga una cache de segundo nivel de 512 KB externa y con una frecuencia de fallos global del 4%. Si cada 512 KB adicionales hiciesen disminuir la frecuencia de fallos global en un 0.7% y la cache tuviese un tiempo de acceso total de 50 ciclos, ¿qué tamaño debería tener la cache para igualar las prestaciones de la cache de segundo nivel de correspondencia directa con las características de la tabla? ¿Y de la cache asociativa con conjuntos de ocho vías?

Ejercicio 5.9

En sistemas de altas prestaciones, como por ejemplo el índice de árboles-B de una base de datos, el tamaño de página se determina principalmente en función del tamaño de los datos y las prestaciones del disco. Suponga que, en promedio, la página de índice de árboles-B está llena al 70% con entradas de tamaño fijo. La utilidad de la página es su profundidad de árbol-B, definida como el $\log_2$ (entradas). La siguiente tabla muestra, para entradas de 16 bytes y un disco de hace 10 años con una latencia de 10 ms y un ritmo de transferencia de 10 MB/s, que el tamaño de página óptimo es 16K

Tamaño de página (KB)	Utilidad de la página o profundidad del árbol-B (número de accesos a disco guardados)	Coste de acceso del índice de la página	Utilidad/coste
2	6.49 (o log2(2048/16x0.7))	10.2	0.64
4	7.49	10.4	0.72
8	8.49	10.8	0.79
16	9.49	11.6	0.82
32	10.49	13.2	0.79
64	11.49	16.4	0.70
128	12.49	22.8	0.55
256	13.49	35.6	0.38

5.9.1 [10]<5.4> ¿Cuál es el mejor tamaño de página si las entradas son de 128 bytes?

5.9.2 [10]<5.4> Basándose en el problema 5.9.1, ¿cuál es el mejor tamaño de página si las páginas están llenas al 50%?

5.9.3 [20]<5.4> Basándose en el problema 5.9.2, ¿cuál es el mejor tamaño de página si se utiliza un disco moderno con una latencia de 3 ms y un ritmo de trans-