

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0521 – Estructuras de Computadores Digitales II
I ciclo 2024

Tarea 4

Prof. Erick Carvajal Barboza

Joselyn Barquero Castillo
Paulette Pérez Monge
Susan Jiménez Hernández

7 de julio de 2024

Índice

1. Parte 1: Experimentos	1
1.1. Efecto del tamaño del caché	1
1.2. Efecto de la asociatividad del caché	2
1.3. Efecto del tamaño del bloque en el caché	4
1.4. Efecto de la política de reemplazo del caché	5
2. Parte 2: Experimentos	6
2.1. Caché de un único nivel	6
2.2. Caché con dos niveles	7
2.3. Presencia de L3	7

Índice de figuras

1. Miss rate total promedio vs tamaño del caché	1
2. Miss rate de 465.tonto-1769B.trace.txt.gz vs tamaño del caché	2
3. Miss rate total promedio vs asociatividad del caché	2
4. Miss rate de 470.lbm-1274B.trace.txt.gz vs asociatividad del caché	3
5. Miss rate total promedio vs tamaño del bloque en el caché	4
6. Miss rate de 401.bzip2-226B.trace.txt.gz vs tamaño del bloque en caché	4
7. Miss rate total promedio vs política de reemplazo del caché	5
8. Gráfico del AMAT por trace	6
9. Gráfico de AMAT mejores casos	8

1. Parte 1: Experimentos

A continuación se muestran los resultados de diferentes optimizaciones aplicadas a una caché de un sólo nivel programada en Python.

1.1. Efecto del tamaño del caché

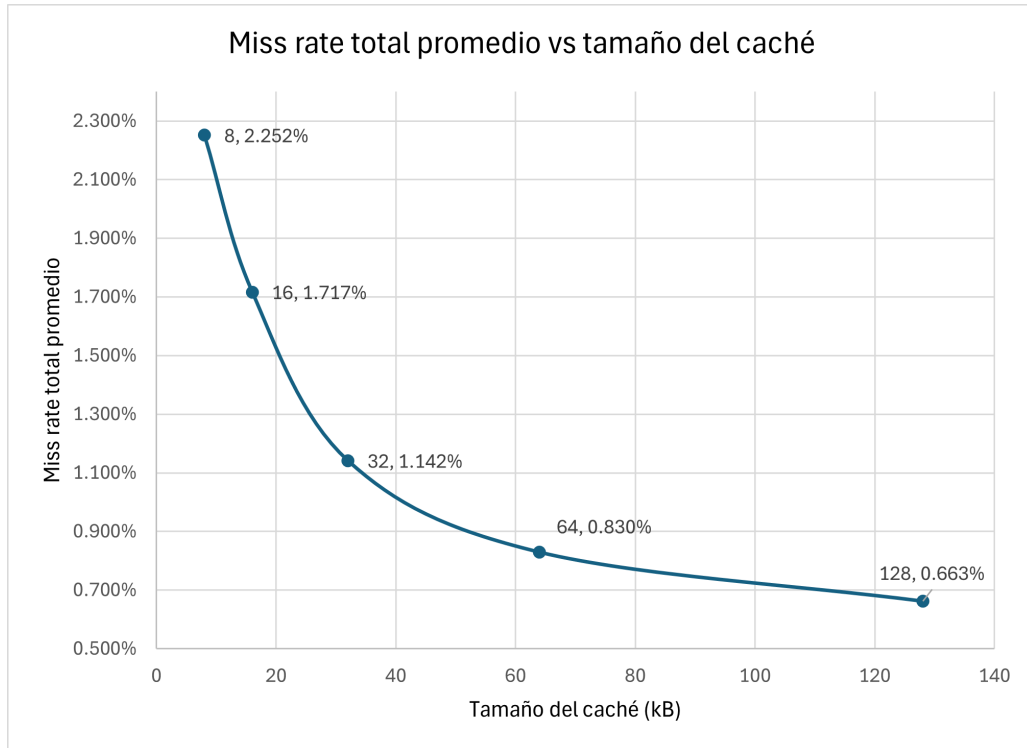


Figura 1: Miss rate total promedio vs tamaño del caché

Según lo visto en clase, una caché más grande disminuye el miss rate, particularmente porque se evitan mises de capacidad al retirar menos bloques para añadir nuevos. La figura 1 verifica este comportamiento, de una forma prácticamente exponencial.

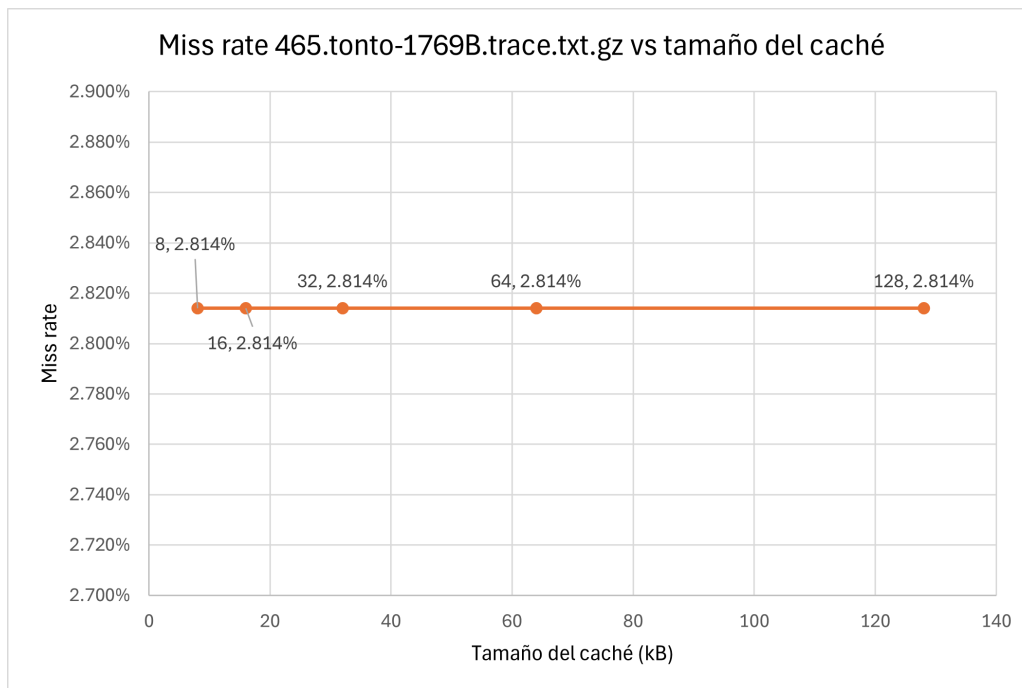


Figura 2: Miss rate de 465.tonto-1769B.trace.txt.gz vs tamaño del caché

En la figura 2 se observa que el trace “465.tonto-1769B.trace.txt.gz” no sigue la misma tendencia que el promedio general, manteniéndose su miss-rate constante. Una posible explicación es que este trace sólo necesita acceder a las posiciones de memoria de unos cuantos bloques distintos y el tamaño del caché original los puede almacenar a todos, por lo que aumentarlo no hizo mucha diferencia.

1.2. Efecto de la asociatividad del caché

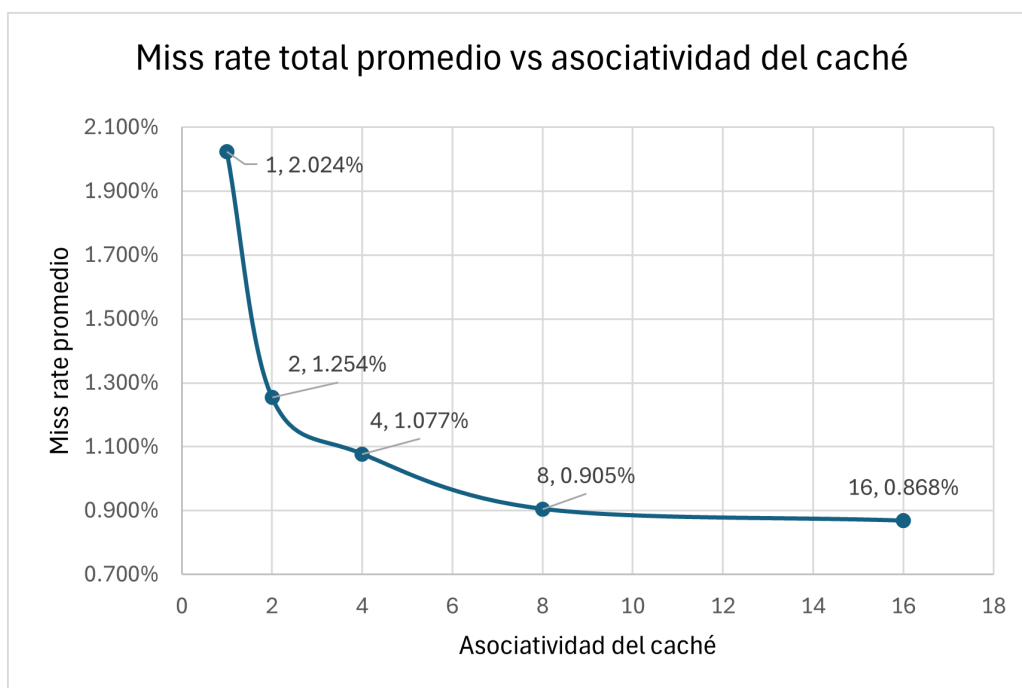


Figura 3: Miss rate total promedio vs asociatividad del caché

Según lo visto en clase, una mayor asociatividad disminuye el miss rate, particularmente los misses de conflicto, pues no se retiran bloques con el mismo index, al haber más ways para almacenarlos. En la figura 3 se observa que ésto mismo sucede, e igual que en la parte anterior, asemeja una función exponencial.

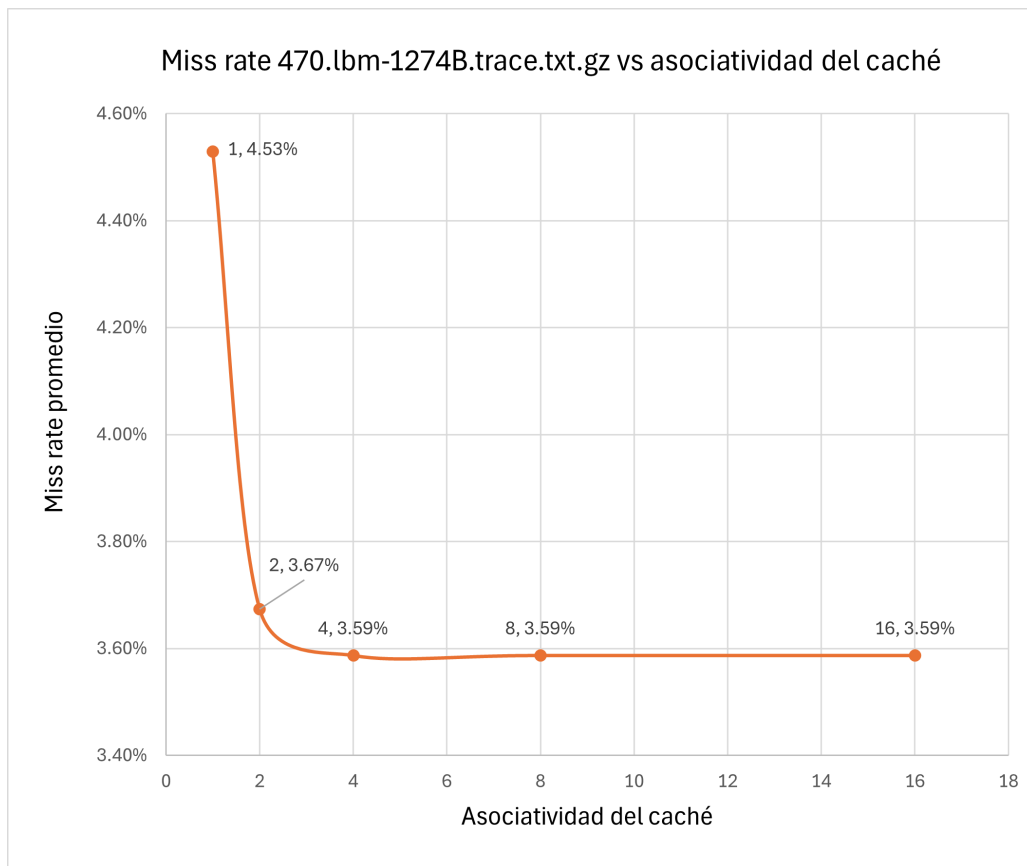


Figura 4: Miss rate de 470.lbm-1274B.trace.txt.gz vs asociatividad del caché

En la figura 4 se observa que el trace “470.lbm-1274B.trace.txt.gz” sigue la misma tendencia que el promedio general, teniendo una disminución prominente en su miss-rate al inicio, pero con la diferencia de que se mantiene constante cuando la asociatividad llega a 4. Una posible explicación es que en 4 ways caben todos los bloques con un mismo index que el trace necesita, y como ya están todos, no hay que reemplazarlos, por lo que más ways serían sólo espacios vacíos.

1.3. Efecto del tamaño del bloque en el caché

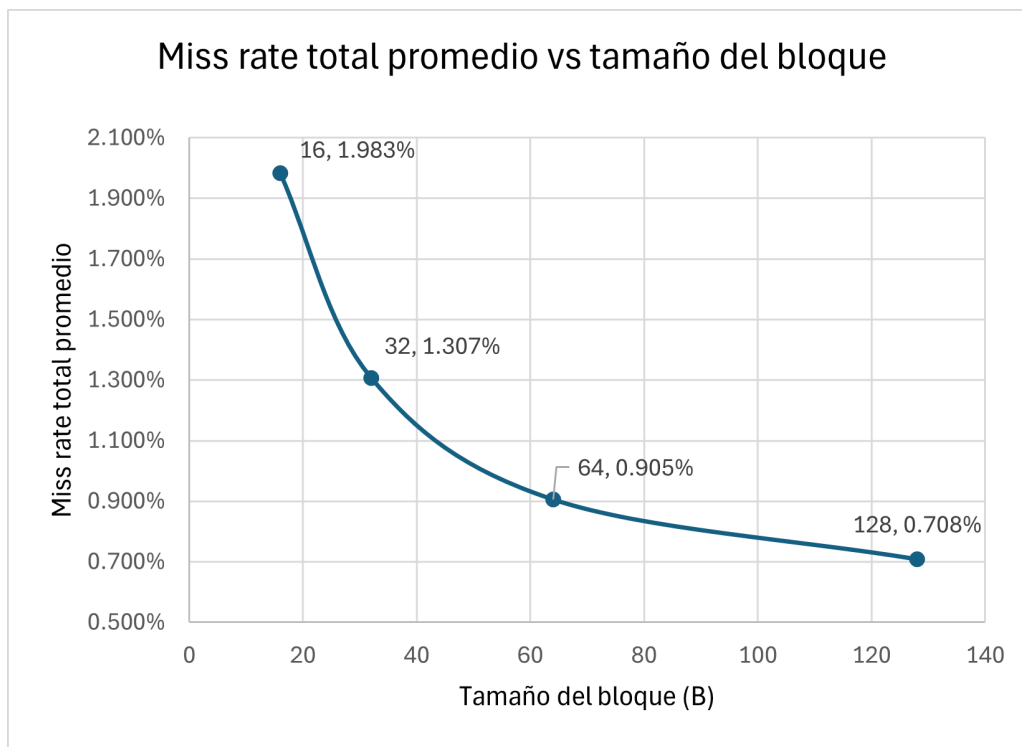


Figura 5: Miss rate total promedio vs tamaño del bloque en el caché

Un tamaño de bloque más grande disminuye el miss rate, por vía de reducir los mises obligatorios, pues más datos cercanos estarán dentro de la caché más pronto, aumentando las posibilidades de encontrar un dato aunque sea la primera vez que se le invoca, basándonos en el principio de localidad espacial. En la figura 5 se verifica que el miss rate disminuye de una forma casi exponencial, como en las dos secciones anteriores.

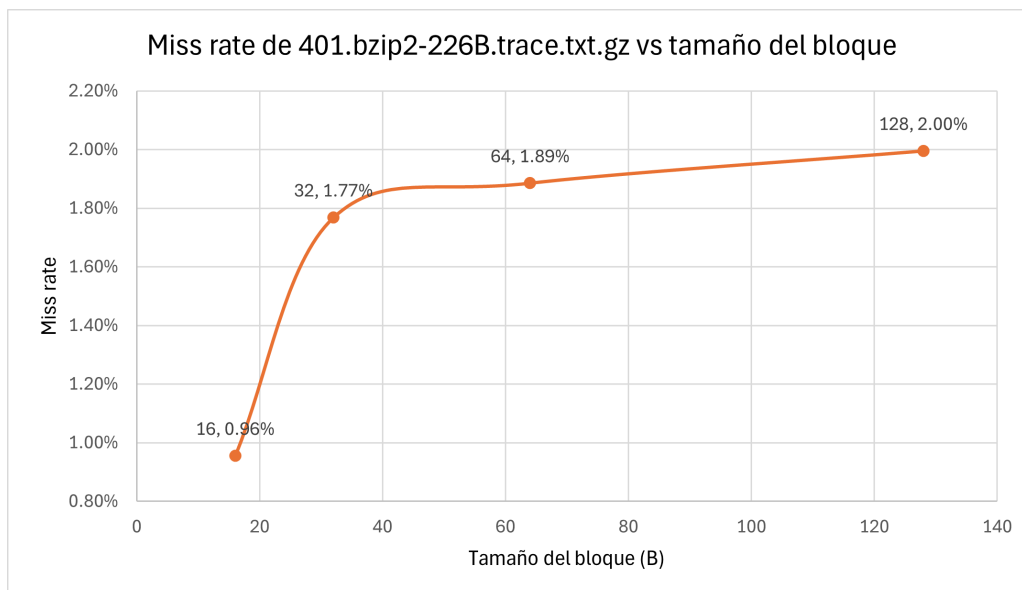


Figura 6: Miss rate de 401.bzip2-226B.trace.txt.gz vs tamaño del bloque en caché

En la figura 6 se observa que el trace “401.bzip2-226B.trace.txt.gz” no sigue la misma tendencia que el promedio general, más bien tiene una inversa. En clases se estudió que el miss

rate puede volver a crecer al aumentar el tamaño del bloque demasiado, al empezar a traer datos innecesarios, lo que indica que este programa tenía los datos que necesitaba en 16 B (improbable porque es demasiado pequeño) o más bien, que estos datos están muy dispersos en la memoria. En el promedio, que es la tendencia mayoritaria, sería muy complicado observar comportamientos como este, porque aplican a trases con características peculiares. En este caso parece que no se cumple el principio de localidad espacial, pero sí lo hace, porque este indica una alta probabilidad de cercanía en los datos, no un 100 % de seguridad. El promedio demuestra que, en general, la probabilidad es alta.

1.4. Efecto de la política de reemplazo del caché

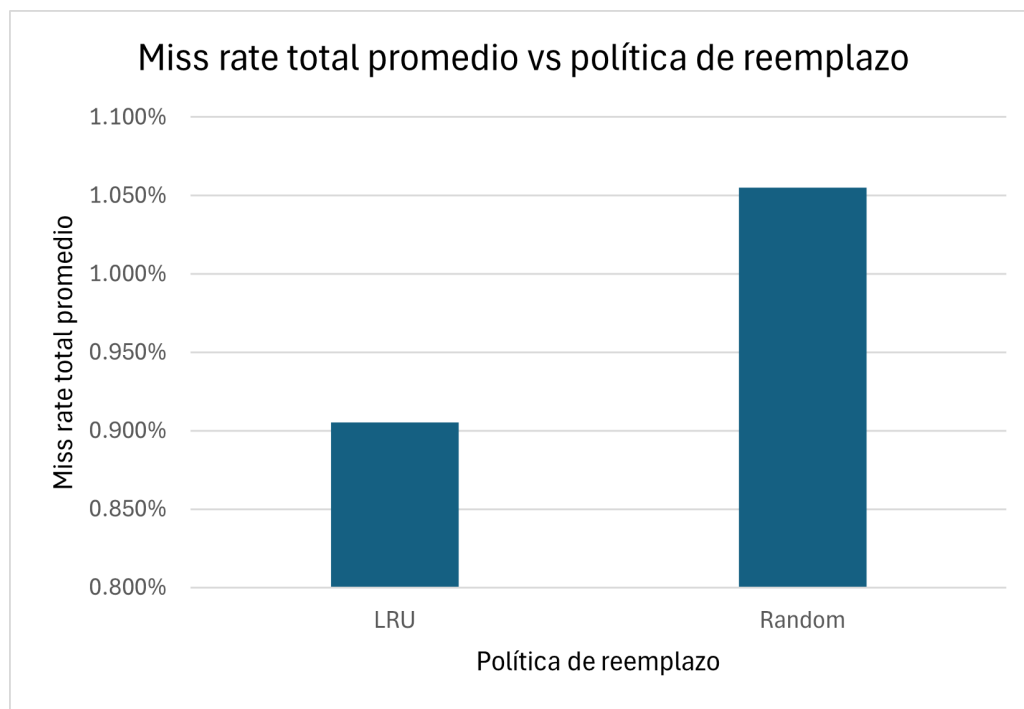


Figura 7: Miss rate total promedio vs política de reemplazo del caché

La figura 7 muestra que la política de eliminar el bloque menos recientemente utilizado es más efectivo para disminuir el miss rate que reemplazar un dato al azar, sin ningpun criterio. Esto viene de analizar la localidad temporal de forma negativa: Si es más probable que una posición de memoria ya referenciada se referencie otra vez en el futuro cercano, si una posición de memoria no se ha referenciado en mucho tiempo, es menos probable que se referencie otra vez.

2. Parte 2: Experimentos

2.1. Caché de un único nivel

Trace	AMAT	Misses	Accesses
400.perlbench-41B.trace.txt.gz	4.12	235	1000000
401.bzip2-226B.trace.txt.gz	13.43	18860	1000000
403.gcc-16B.trace.txt.gz	14.82	21639	1000000
410.bwaves-1963B.trace.txt.gz	12.03	16065	1000000
416.gamess-875B.trace.txt.gz	9.86	11729	1000000
429.mcf-184B.trace.txt.gz	57.63	107254	1000000
433.milc-127B.trace.txt.gz	15.04	22083	1000000
435.gromacs-111B.trace.txt.gz	6.12	4248	1000000
436.cactusADM-1804B.trace.txt.gz	6.17	4344	1000000
437.leslie3d-134B.trace.txt.gz	5.98	3968	1000000
444.namd-120B.trace.txt.gz	6.18	4368	1000000
445.gobmk-17B.trace.txt.gz	9.70	11407	1000000
450.soplex-247B.trace.txt.gz	37.87	67734	1000000
453.povray-887B.trace.txt.gz	16.50	25001	1000000
454.calculix-104B.trace.txt.gz	4.21	420	1000000
456.hmmer-191B.trace.txt.gz	6.05	4097	1000000
458.sjeng-1088B.trace.txt.gz	7.46	6924	1000000
459.GemsFDTD-1169B.trace.txt.gz	12.99	17978	1000000
462.libquantum-1343B.trace.txt.gz	12.82	17649	1000000
464.h264ref-30B.trace.txt.gz	4.28	555	1000000
465.tonto-1769B.trace.txt.gz	18.07	28139	1000000
470.lbm-1274B.trace.txt.gz	21.93	35867	1000000
471.omnetpp-188B.trace.txt.gz	17.99	27971	1000000
473.astar-153B.trace.txt.gz	6.22	4449	1000000
481.wrf-1170B.trace.txt.gz	5.92	3834	1000000
482.sphinx3-1100B.trace.txt.gz	9.97	11943	1000000
483.xalancbmk-127B.trace.txt.gz	14.32	20642	1000000

Tabla 1: Resultado AMAT para los trazes en esta configuración

Media geométrica del AMAT: 10,49

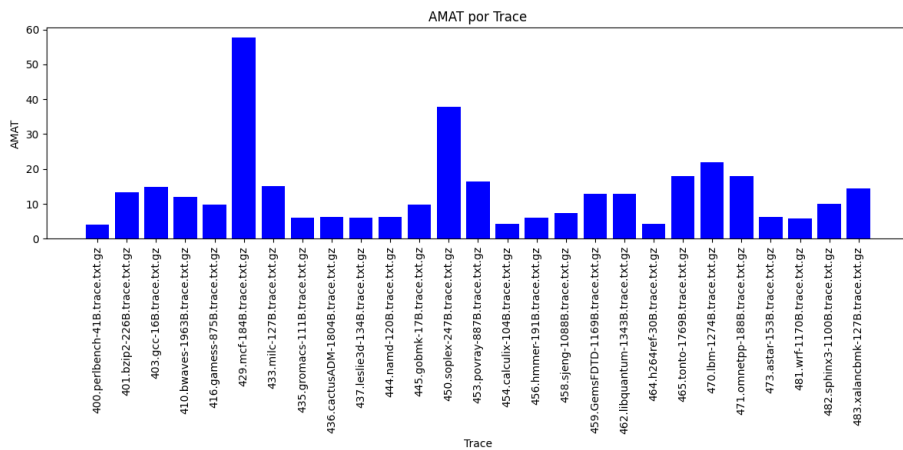


Figura 8: Gráfico del AMAT por trace

Se realizó una simulación utilizando un caché L1 de 32kB, asociatividad de 8-way, tamaño de bloque de 64B y política de reemplazo LRU. Se calcularon los tiempos promedio de acceso a memoria para los traves y el AMAT promedio geométrico a través de todos los traves fue 10.49. Se observó que los traves con alta localización temporal y espacial presentaron un menor AMAT. De forma que los traves con patrones de acceso más dispersos y menos predecibles tuvieron un AMAT significativamente mayor.

2.2. Caché con dos niveles

L2 Size (kB)	L2 Assoc	AMAT (Geo Mean)
64	8	7.17
64	16	7.12
128	8	6.81
128	16	6.78

Tabla 2: Resultados de AMAT geométrico para diferentes configuraciones de L2

Se agregó un caché L2 con bloques de 64B y política de reemplazo LRU, variando la capacidad y la asociatividad del caché para encontrar la configuración que minimiza el AMAT. Para una capacidad de 64kB, la asociatividad de 16-way mostró una ligera mejora en el AMAT en comparación con 8-way. Para una capacidad de 128kB, tanto 8-way como 16-way proporcionaron una mejora significativa en el AMAT en comparación con 64kB, con 16-way siendo ligeramente mejor. La configuración óptima fue un caché L2 de 128kB con 16-way de asociatividad, con un AMAT promedio geométrico de 6.78.

2.3. Presencia de L3

L3 Size (kB)	L3 Assoc	AMAT (Geo Mean)
512	16	5.81
512	32	5.81
1024	16	5.78
1024	32	5.78

Tabla 3: Resultados de AMAT Geométrico para diferentes configuraciones de L3

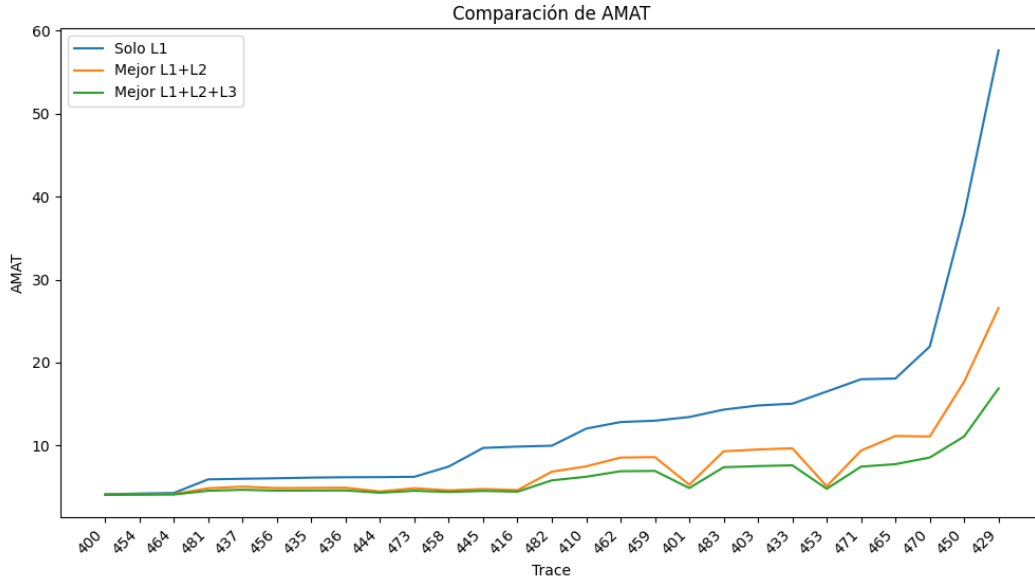


Figura 9: Gráfico de AMAT mejores casos

Finalmente, se añadió un caché L3 a la simulación con un caché L2 de 256kB y asociatividad de 8-ways. Se variaron la capacidad y la asociatividad del caché L3 para encontrar la configuración que minimiza el AMAT. Las configuraciones de 512kB y 1024kB mostraron que la capacidad adicional de L3 tiene un impacto en la reducción del AMAT. Las configuraciones de 32-way asociatividad no proporcionaron una mejora significativa sobre las de 16-way, indicando que aumentar la asociatividad más allá de un cierto punto no es beneficioso. Por lo tanto, la configuración óptima fue un caché L3 de 1024kB con 16-way de asociatividad, con un AMAT promedio geométrico de 5.78.

Referencias