

UNIVERSIDAD DE COSTA RICA
Escuela de Ingeniería Eléctrica
IE0521 – Estructuras de Computadoras Digitales II

Tarea 4

Mariano Segura Chaves C17416
Jimena González Jiménez B83443
Juan Ignacio Hernández Zamora B93826

7 de julio de 2024

Índice

1. Introducción	4
1.1. Desarrollo	4
1.2. Caché de un único nivel	4
1.2.1. Efecto del tamaño del caché	5
1.2.2. Tamaño del bloque	6
1.2.3. Efecto de la asociatividad	7
1.2.4. Política de reemplazo	9
1.3. Caché multinivel	10
1.3.1. Caché de un único nivel	10
1.3.2. Caché con 2 niveles	11
1.3.3. Caché con 3 niveles	13
2. Conclusiones	17

Índice de figuras

1.	Miss rate promedio graficado para cada tamaño de caché.	5
2.	Cantidad de misses vs miss rate.	6
3.	Miss rate promedio graficado para los distintos tamaños de bloque.	7
4.	Cantidad de misses vs miss rates.	7
5.	Miss rate promedio dependiendo de la asociatividad del cache.	8
6.	Cantidad de misses vs miss rate.	9
7.	Miss rate promedio graficado para cada política de reemplazo.	10
8.	Comparación de los AMAT multinivel.	16

1. Introducción

En el curso *IE 0521 – Estructuras de Computadoras Digitales II*, la Tarea 4 tiene como objetivo principal analizar el rendimiento de distintas configuraciones de caché mediante simulaciones en un lenguaje de alto nivel que en este caso fue Python. Este análisis incluye la evaluación de configuraciones de caché multinivel.

El proyecto consiste en desarrollar un simulador de memoria caché que permita obtener métricas de rendimiento utilizando como entrada un "trace" que indica los accesos a memoria, especificando si son lecturas o escrituras. Los datos a utilizar provienen de archivos "trace" con información de alrededor de 1,000,000 de accesos de memoria, generados por aplicaciones de la suite de benchmarks SPEC CPU2000.

El desarrollo de esta tarea se divide en dos partes:

- **Caché de un único nivel:** En esta parte se implementará una clase **Cache** que maneje las lecturas y escrituras del trace, permitiendo modificar parámetros como la capacidad del caché, la asociatividad, el tamaño del bloque y la política de reemplazo. Se espera que el código sea parametrizable y que los resultados obtenidos sean analizados para distintas configuraciones de caché, variando estos parámetros de manera sistemática.
- **Caché multinivel:** Utilizando la clase **Cache** desarrollada en la primera parte, se implementará un simulador de caché multinivel. Aquí se consideran cachés L1, L2 y opcionalmente L3, con configuraciones específicas para cada nivel. El objetivo es analizar cómo diferentes configuraciones de caché multinivel afectan el rendimiento del sistema.

Este reporte documenta el proceso de desarrollo del simulador, la metodología utilizada para la recolección de datos, los resultados obtenidos de las simulaciones y un análisis detallado de estos resultados. Además, se incluyen gráficas y tablas que ilustran el comportamiento del caché bajo diferentes configuraciones, proporcionando una visión completa del rendimiento y las posibles optimizaciones en sistemas de memoria caché.

1.1. Desarrollo

Al igual que con las tareas anteriores, esta consistió en varias simulaciones. Primero, se simuló para una caché de un único nivel y luego una multinivel. Estas simulaciones se llevaron a cabo con una lista de trazes específicos que fueron proporcionadas por el profesor.

Como eran varias simulaciones, fue necesario aplicar lo mismo que en la tarea 3 y automatizarlas. Esto se hizo con un código de Python que toma los parámetros requeridos para cada simulación y los reemplaza en el código en sí que simula la caché.

1.2. Caché de un único nivel

Esta sección de la tarea requirió más simulaciones que la del caché multinivel. Ya que se tuvo que simular con cambios en la capacidad del caché, cambios en el tamaño del bloque, cambios en la asociatividad (cantidad de ways) y por último alterando la política de reemplazo.

En cada sección se explicará más a profundidad las modificaciones realizadas a dichos parámetros y cómo afectaron estas el rendimiento de la caché.

1.2.1. Efecto del tamaño del caché

Las primeras simulaciones se hicieron variando el tamaño del caché. Se usaron los valores 8, 16, 32, 64 y 128; mientras que se mantuvo una asociatividad de 8 ways, un tamaño de bloque de 64 bits, y política de reemplazo LRU.

Capacidad del caché	Miss rate promedio
8	1,790
16	1,346
32	0,905
64	0,671
128	0,525

Tabla 1: Media geométrica del miss rate para las distintas capacidades de cache.

Cabe recalcar que para cada valor del miss rate promedio se tomó el miss rate para cada uno de los trases con los valores solicitados en el enunciado y con base en esto se sacó el promedio. Luego estos valores fueron graficados.

Se puede ver que al aumentar el tamaño del caché reduce la cantidad de misses. Esto se debe a que al tener una memoria caché más grande, se tienen más datos a disposición del procesador. Lo cual reduce especialmente, los misses de capacidad.

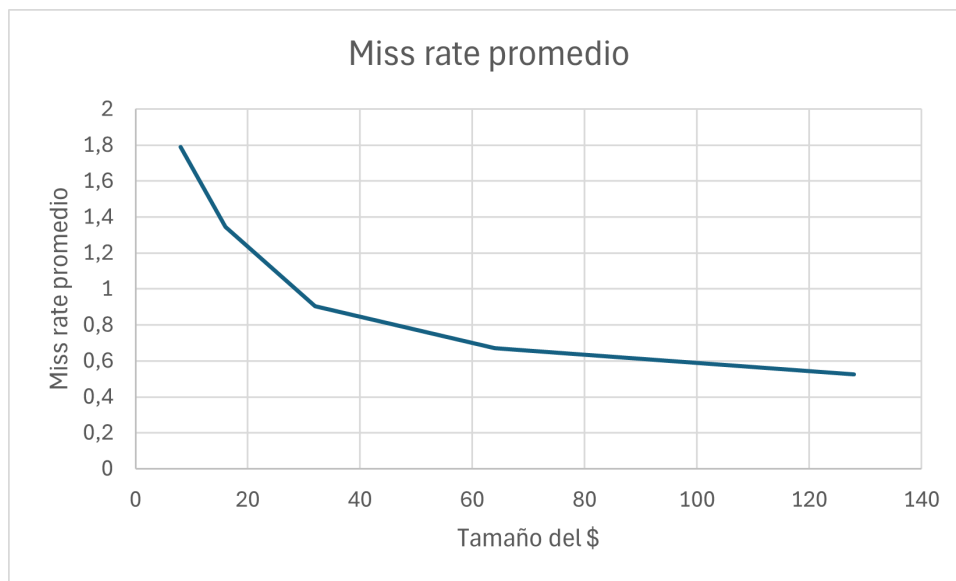


Figura 1: Miss rate promedio graficado para cada tamaño de caché.

Finalmente, se le prestó principal atención al trase 465.tonto-1769B.trace.txt.gz. Por lo tanto, se graficó la cantidad de misses vs miss rate. Lo más notorio del comportamiento de este trase es que a pesar de que se cambia la capacidad del caché no hay cambios en la cantidad de misses. Lo que quiere decir que el tipo de acceso de estos datos no requiere del uso de esos parámetros que nosotros estamos midiendo.

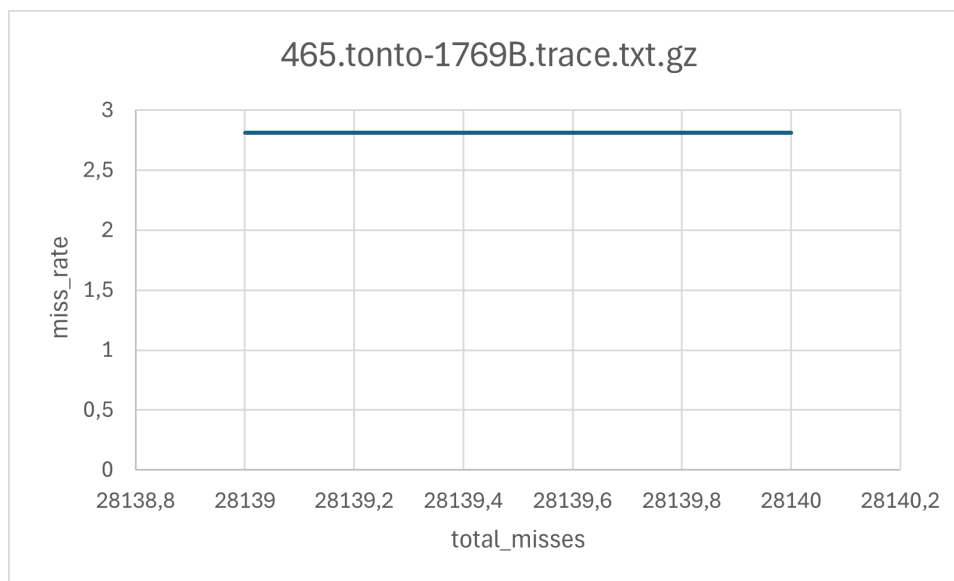


Figura 2: Cantidad de misses vs miss rate.

1.2.2. Tamaño del bloque

Luego, se cambió el tamaño del bloque. Se usaron bloques de tamaño 16, 32, 64 y 128B. La capacidad se mantuvo en 32kB, la asociatividad en 8-way y la política de reemplazo en LRU.

Tamaño del bloque	Miss rate promedio
16	1,983
32	1,307
64	0,905
128	0,708

Tabla 2: Miss rate promedio para los distintos tamaños del bloque.

Al igual que para la capacidad del caché, se calculó la media geométrica para todos los trases con las distintas variaciones en el tamaño del bloque. En otras palabras, se sacó el promedio para un tamaño de bloque 16B en todos los trases. Luego, con los datos obtenidos que se presentan en la tabla 2 se hizo la gráfica de la figura 3.

En dicha imagen y tabla, se puede ver cómo al tener un bloque más grande de caché disminuye el miss rate promedio. Esto se debe a que al aumentar el tamaño del bloque se reduce la cantidad de misses. En el caso de los misses obligatorios, al ser cada bloque (o línea) de caché más grande se pueden traer más datos contiguos a cada acceso. Lo que significa una mayor cantidad de datos y por ende, una menor cantidad de misses compulsivos a futuro.

Aun así, se puede observar como esto contrarresta el efecto negativo que tiene el aumento del tamaño del bloque en los otros dos tipos de misses. Los misses de conflicto y de capacidad. Donde, en el caso de los misses de conflicto, aumentar el tamaño del bloque de caché no necesariamente evita que varios bloques compitan por el mismo lugar en el caché debido a como son mapeados. Por otro lado, los misses de capacidad no necesariamente disminuyen ya que el tener bloques más grandes no significa que hay más espacio para almacenar datos en caché.

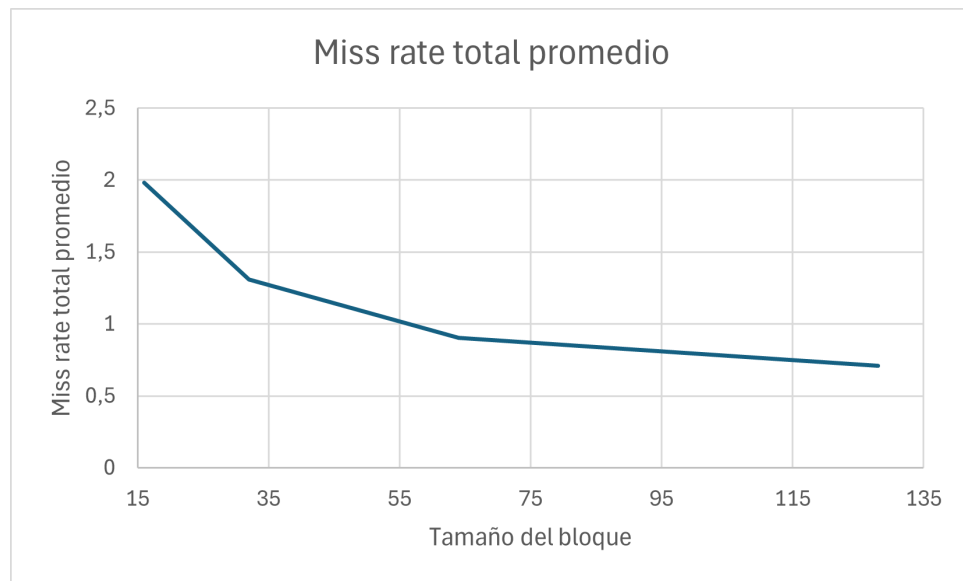


Figura 3: Miss rate promedio graficado para los distintos tamaños de bloque.

Finalmente, se graficó el la cantidad de misses vs el miss rate para el trace 401.bzip2-226B.trace.txt.gz. En este caso, se puede ver que el comportamiento de este trace es una función lineal. Lo cual indica que el cuello de botella se relaciona con los accesos a memoria. Es por la localidad espacial, en este caso se están utilizando datos que están muy cercanos entre si. Por lo que el aumento del tamaño del bloque aumenta el rendimiento y reduce la cantidad de misses.

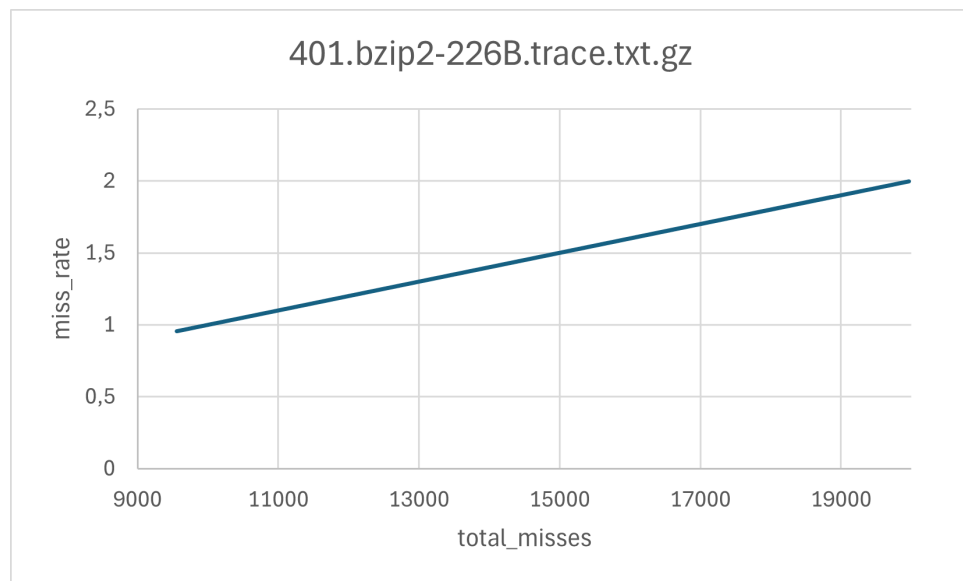


Figura 4: Cantidad de misses vs miss rates.

1.2.3. Efecto de la asociatividad

Para la asociatividad del caché entre mapeo directo y 16-way, se utilizaron las potencias de 2 que hay en medio únicamente i.e 1, 2, 4, 8, 16. Se mantuvo la capacidad en 32kB, el tamaño de bloque en

64B y la política de reemplazo en LRU. Y se procedió a graficar el miss rate total promedio, usando la media geométrica (a través de todos los trace) vs asociatividad del caché. En específico, se graficó el comportamiento del trace “470.lbm-1274B.trace.txt.gz”.

Asociatividad	Miss rate promedio
1	2,023
2	1,254
4	10769,6
8	0,905
16	0,868

Tabla 3: Miss rate promedio para los cambios en la asociatividad.

Para la tabla 3 se tomaron los miss rates de todos los traces y luego se sacó el miss rate promedio para cuando se usaba una asociatividad de 8 ways, 16 ways y así sucesivamente. El miss rate promedio fue calculado con la media geométrica, utilizando el comando `=geomean()` en Excel. Seguidamente se graficaron estos datos.

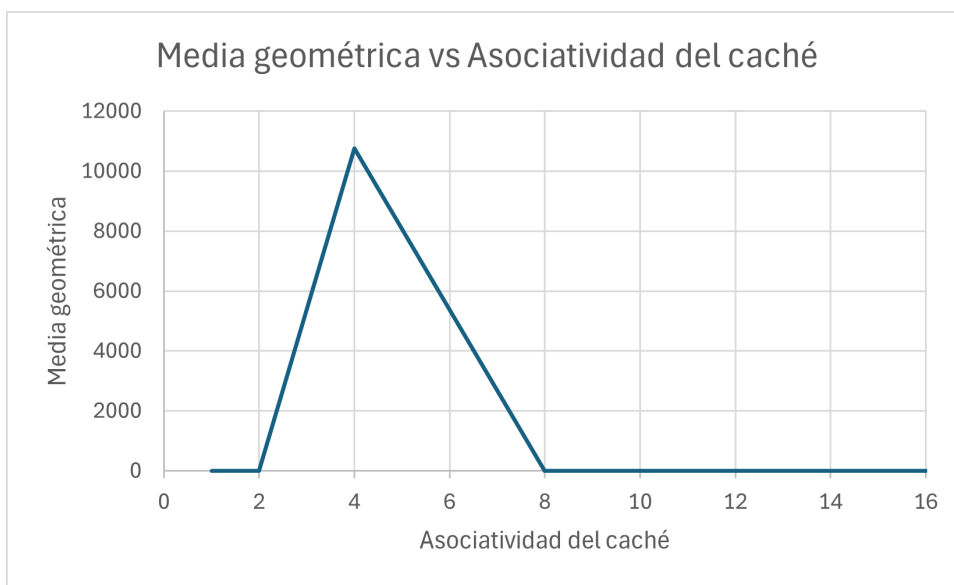


Figura 5: Miss rate promedio dependiendo de la asociatividad del cache.

Finalmente, se le prestó especial atención al trace 470.lbm-1274B.trace.txt.gz. Se graficó la cantidad de misses contra el miss rate. Que al igual que el trace de la figura 4 se comporta como una función lineal. Lo que indica que una mayor asociatividad reduce la tasa de fallos. Esto, debido a que una caché asociativa disminuye la posibilidad que dos bloques compitan por una misma línea de caché, lo cual disminuye misses de conflicto.

Lo cual también es aplicable al resto de simulaciones y resultados con los cambios en la asociatividad del caché. Ya que al tener mayor asociatividad se reducen los misses de conflicto. Aun así, este comportamiento se sigue al pie de la letra cuando se tiene 4 ways. Aunque para uno y dos ways igual se tiene un miss rate más alto que para 8 y 16, se ve un salto más alto en la cantidad de misses cuando se tiene 4 ways.

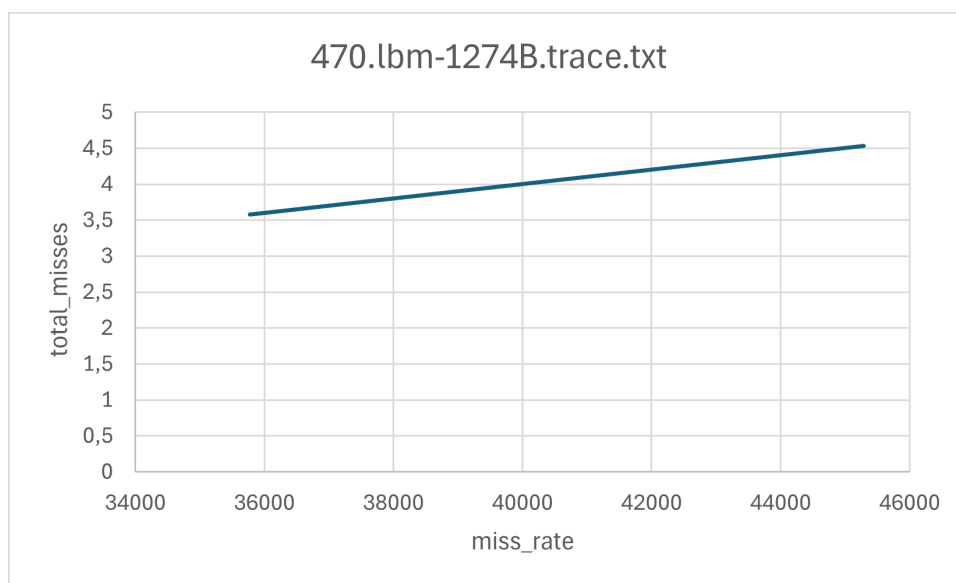


Figura 6: Cantidad de misses vs miss rate.

1.2.4. Política de reemplazo

Se varió la política de reemplazo del caché (LRU y aleatoria). Se mantuvo la capacidad en 32kB, la asociatividad en 8-way y el tamaño de bloque en 64B. Para cada una de esas 2 configuraciones, se simuló para todos los traves brindados, se procedió a graficar el miss rate total promedio, usando la media geométrica (a través de todos los trace) vs política de reemplazo.

Política de reemplazo	Miss rate promedio
LRU	0,905
Random	1,092

Tabla 4: Miss rate promedio para las dos políticas de reemplazo utilizadas.

Para este último grupo de simulaciones, se hizo lo mismo que las pasadas de sacar el promedio por política de reemplazo. Para todos los traves. Por último, se graficó lo apreciado en la tabla 4.

En este caso se tiene un miss rate más alto para cuando se utiliza la política de reemplazo aleatoria. Como se vio en clases, esta es la política de reemplazo menos eficiente ya que literalmente no sigue un criterio lógico para reemplazar datos. Solo reemplaza aleatoriamente los mismos, por lo cual es completamente esperable que el miss rate sea más alto.

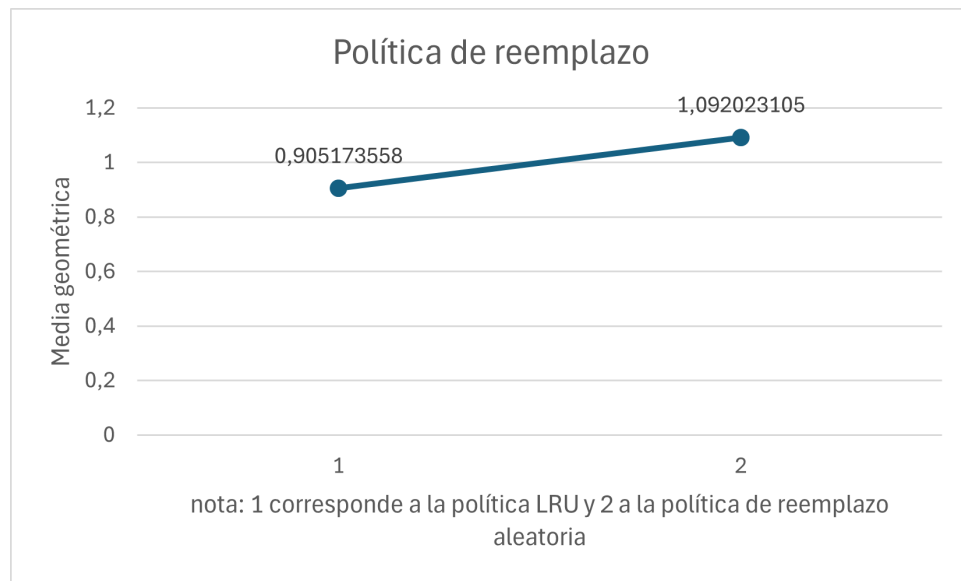


Figura 7: Miss rate promedio graficado para cada política de reemplazo.

1.3. Caché multinivel

Luego, se simuló para un caché multinivel. El cual tiene niveles L1, L2 y L3. En este caso solo se simula para un caché de un único nivel, uno de dos niveles y uno de tres. Para las tres simulaciones únicamente se graficó el AMAT de cada uno de los trances y se sacó el promedio con la media geométrica.

1.3.1. Caché de un único nivel

Para esta primera simulación, fue necesario sacar con una capacidad de 32kB, asociatividad en 8-way y un bloque de 64B y la política de reemplazo LRU. Con un hit time de 4 ciclos de reloj y un miss penalty de 500 ciclos.

Tabla 5: Resultados de la simulación de la caché multinivel con L1.

Trace file	AMAT (ciclos)
429.mcf-184B.trace.txt.gz	3576346.5
450.soplex-247B.trace.txt.gz	3732450.5
470.lbm-1274B.trace.txt.gz	3858325.5
465.tonto-1769B.trace.txt.gz	3888851
471.omnetpp-188B.trace.txt.gz	3889514.5
403.gcc-16B.trace.txt.gz	3914526
433.milc-127B.trace.txt.gz	3912772
453.povray-887B.trace.txt.gz	3901246
483.xalancbmk-127B.trace.txt.gz	3918464
401.bzip2-226B.trace.txt.gz	3925503
459.GemsFDTD-1169B.trace.txt.gz	3928987
462.libquantum-1343B.trace.txt.gz	3930286.5
410.bwaves-1963B.trace.txt.gz	3936543.5
482.sphinx3-1100B.trace.txt.gz	3952825
416.gamess-875B.trace.txt.gz	3953670.5
445.gobmk-17B.trace.txt.gz	3954942.5
458.sjeng-1088B.trace.txt.gz	3972650
473.astar-153B.trace.txt.gz	3982426.5
444.namd-120B.trace.txt.gz	3982746.5
436.cactusADM-1804B.trace.txt.gz	3982841
435.gromacs-111B.trace.txt.gz	3983220.5
456.hmmmer-191B.trace.txt.gz	3983817
437.leslie3d-134B.trace.txt.gz	3984326.5
481.wrf-1170B.trace.txt.gz	3984855.5
464.h264ref-30B.trace.txt.gz	3997808
454.calculix-104B.trace.txt.gz	3998341
400.perlbench-41B.trace.txt.gz	3999072
Media geométrica	3925900.72288113

1.3.2. Caché con 2 niveles

Utilizando el mismo caché L1 descrito en la sección anterior se agregó ahora un caché L2 a la simulación. Utilizando bloques de 64B y LRU como política de reemplazo se fue variando la capacidad y la asociatividad del caché y se probaron todas las configuraciones en sus combinaciones de: Capacidad de L2: 64kB y 128kB Asociatividad de L2: 8-way y 16-way Se asumió que el hit time de L1 es 4 ciclos de reloj, el de L2 es 12 ciclos y el miss penalty son 500 ciclos.

Tabla 6: Resultados de la simulación del caché multinivel con L2

Trace file	Tamaño L2	Asociatividad L2	AMAT (ciclos)
400.perlbench-41B.trace.txt.gz	64	8	4095037.812
464.h264ref-30B.trace.txt.gz	64	8	4221657.248
454.calculix-104B.trace.txt.gz	64	8	4166250.658
437.leslie3d-134B.trace.txt.gz	64	8	5565905.6205
481.wrf-1170B.trace.txt.gz	64	8	5512406.7635
456.hmmmer-191B.trace.txt.gz	64	8	5618197.95
435.gromacs-111B.trace.txt.gz	64	8	5676701
473.astar-153B.trace.txt.gz	64	8	5755672.29
444.namd-120B.trace.txt.gz	64	8	5728572.2065
436.cactusADM-1804B.trace.txt.gz	64	8	5712252.012
458.sjeng-1088B.trace.txt.gz	64	8	6735542.348
445.gobmk-17B.trace.txt.gz	64	8	8503814.5515
416.gamess-875B.trace.txt.gz	64	8	8629960.511
482.sphinx3-1100B.trace.txt.gz	64	8	8674631.728
410.bwaves-1963B.trace.txt.gz	64	8	10279570.161
462.libquantum-1343B.trace.txt.gz	64	8	10866359.6725
459.GemsFDTD-1169B.trace.txt.gz	64	8	10994004.149
401.bzip2-226B.trace.txt.gz	64	8	11353439.503
483.xalancbmk-127B.trace.txt.gz	64	8	12005166.464
403.gcc-16B.trace.txt.gz	64	8	12386777.514
433.milc-127B.trace.txt.gz	64	8	12551068.576
453.povray-887B.trace.txt.gz	64	8	13777862.25
465.tonto-1769B.trace.txt.gz	64	8	14830670.714
471.omnetpp-188B.trace.txt.gz	64	8	14882254.5315
470.lbm-1274B.trace.txt.gz	64	8	17696345.5685
450.soplex-247B.trace.txt.gz	64	8	29167761.154
429.mcf-184B.trace.txt.gz	64	8	42221347.5375
Media geométrica			9176227.13863014
400.perlbench-41B.trace.txt.gz	64	16	4095037.812
464.h264ref-30B.trace.txt.gz	64	16	4221657.248
454.calculix-104B.trace.txt.gz	64	16	4166250.658
437.leslie3d-134B.trace.txt.gz	64	16	5565905.6205
481.wrf-1170B.trace.txt.gz	64	16	5512470.3415
456.hmmmer-191B.trace.txt.gz	64	16	5618228.7
435.gromacs-111B.trace.txt.gz	64	16	5676835.3
473.astar-153B.trace.txt.gz	64	16	5755886.5575
444.namd-120B.trace.txt.gz	64	16	5728758.587
436.cactusADM-1804B.trace.txt.gz	64	16	5711502.711
458.sjeng-1088B.trace.txt.gz	64	16	6736004.258
445.gobmk-17B.trace.txt.gz	64	16	8505815.8655
416.gamess-875B.trace.txt.gz	64	16	8637007.895
482.sphinx3-1100B.trace.txt.gz	64	16	8674702.771
410.bwaves-1963B.trace.txt.gz	64	16	10287421.963
462.libquantum-1343B.trace.txt.gz	64	16	10866359.6725
459.GemsFDTD-1169B.trace.txt.gz	64	16	10994004.149
401.bzip2-226B.trace.txt.gz	64	16	11348098.351
483.xalancbmk-127B.trace.txt.gz	64	16	12005166.464
403.gcc-16B.trace.txt.gz	64	16	12386828.368
433.milc-127B.trace.txt.gz	64	16	12551068.576
453.povray-887B.trace.txt.gz	64	16	13789623.5
465.tonto-1769B.trace.txt.gz	64	16	14830670.714
471.omnetpp-188B.trace.txt.gz	64	16	14891467.8495
470.lbm-1274B.trace.txt.gz	64	16	17696345.5685
450.soplex-247B.trace.txt.gz	64	16	29168617.9385
429.mcf-184B.trace.txt.gz	64	16	42244690.5
Media geométrica			9177402.7721011

Tabla 7: Resultados de la simulación del caché multinivel con L2

Trace file	Tamaño L2	Asociatividad L2	AMAT (ciclos)
454.calcuix-104B.trace.txt.gz	128	8	4166250.658
464.h264ref-30B.trace.txt.gz	128	8	4221657.248
400.perlbenc-41B.trace.txt.gz	128	8	4095037.812
435.gromacs-111B.trace.txt.gz	128	8	5677798.775
437.leslie3d-134B.trace.txt.gz	128	8	5565905.6205
481.wrf-1170B.trace.txt.gz	128	8	5512488.534
456.hmmer-191B.trace.txt.gz	128	8	5618818.28
436.cactusADM-1804B.trace.txt.gz	128	8	5713683.344
444.namd-120B.trace.txt.gz	128	8	5728817.3635
473.astar-153B.trace.txt.gz	128	8	5757166.3775
458.sjeng-1088B.trace.txt.gz	128	8	6736969.252
445.gobmk-17B.trace.txt.gz	128	8	8511886.556
416.gamess-875B.trace.txt.gz	128	8	8639597.879
482.sphinx3-1100B.trace.txt.gz	128	8	8677952.242
410.bwaves-1963B.trace.txt.gz	128	8	10289491.779
462.libquantum-1343B.trace.txt.gz	128	8	10866359.6725
459.GemsFDTD-1169B.trace.txt.gz	128	8	10994004.149
401.bzip2-226B.trace.txt.gz	128	8	11433143.749
483.xalancbmk-127B.trace.txt.gz	128	8	12005182.976
403.gcc-16B.trace.txt.gz	128	8	12387093.458
433.milc-127B.trace.txt.gz	128	8	12551068.576
453.povray-887B.trace.txt.gz	128	8	13871181
465.tonto-1769B.trace.txt.gz	128	8	14830670.714
471.omnetpp-188B.trace.txt.gz	128	8	14893688.6675
470.lbm-1274B.trace.txt.gz	128	8	17867190.7915
450.soplex-247B.trace.txt.gz	128	8	29344814.147
429.mcf-184B.trace.txt.gz	128	8	42528785.025
Media geométrica			9190495.8607761
454.calcuix-104B.trace.txt.gz	128	16	4166250.658
464.h264ref-30B.trace.txt.gz	128	16	4221657.248
400.perlbenc-41B.trace.txt.gz	128	16	4095037.812
435.gromacs-111B.trace.txt.gz	128	16	5677825.7625
437.leslie3d-134B.trace.txt.gz	128	16	5565905.6205
481.wrf-1170B.trace.txt.gz	128	16	5512488.534
456.hmmer-191B.trace.txt.gz	128	16	5618871.785
436.cactusADM-1804B.trace.txt.gz	128	16	5713731.301
444.namd-120B.trace.txt.gz	128	16	5728817.3635
473.astar-153B.trace.txt.gz	128	16	5757166.3775
458.sjeng-1088B.trace.txt.gz	128	16	6737037.414
445.gobmk-17B.trace.txt.gz	128	16	8512012.6365
416.gamess-875B.trace.txt.gz	128	16	8640218.9825
482.sphinx3-1100B.trace.txt.gz	128	16	8677989.853
410.bwaves-1963B.trace.txt.gz	128	16	10290595.788
462.libquantum-1343B.trace.txt.gz	128	16	10866359.6725
459.GemsFDTD-1169B.trace.txt.gz	128	16	10994004.149
401.bzip2-226B.trace.txt.gz	128	16	11439416.585
483.xalancbmk-127B.trace.txt.gz	128	16	12005182.976
403.gcc-16B.trace.txt.gz	128	16	12387110.77
433.milc-127B.trace.txt.gz	128	16	12551068.576
453.povray-887B.trace.txt.gz	128	16	13879929.75
465.tonto-1769B.trace.txt.gz	128	16	14830670.714
471.omnetpp-188B.trace.txt.gz	128	16	14893954.3825
470.lbm-1274B.trace.txt.gz	128	16	17857484.3695
450.soplex-247B.trace.txt.gz	128	16	29349683.934
429.mcf-184B.trace.txt.gz	128	16	42540008.7375
Media geométrica			9190943.74673894

1.3.3. Caché con 3 niveles

Ahora se utilizó el mismo caché L1 y un L2 de 256kB y asociatividad de 8-ways, se agregó un caché L3 a la simulación. Utilizando bloques de 64B y LRU como política de reemplazo se fue variando la capacidad y la asociatividad del caché probando todas sus combinaciones. Capacidad de L3: 512kB y 1024kB. Asociatividad de L3: 16-way y 32-way. Se asumió que el hit time de L1 es 4 ciclos de reloj, el de L2 es 12 ciclos, el de L3 es 60 ciclos y el miss penalty son 500 ciclos.

Tabla 8: Resultados de la simulación del caché multinivel con L3

Trace file	Capacidad L3	Asociatividad L3	AMAT (ciclos)
400.perlbenc-41B.trace.txt.gz	512	16	4320104,864
454.calculix-104B.trace.txt.gz	512	16	4607905,124
464.h264ref-30B.trace.txt.gz	512	16	4857462,699
444.namd-120B.trace.txt.gz	512	16	11814540,92
481.wrf-1170B.trace.txt.gz	512	16	14892968,38
456.hmm-191B.trace.txt.gz	512	16	15636517,12
473.astar-153B.trace.txt.gz	512	16	15914515,28
435.gromacs-111B.trace.txt.gz	512	16	16189021,7
436.cactusADM-1804B.trace.txt.gz	512	16	16576085,72
458.sjeng-1088B.trace.txt.gz	512	16	17201361,27
437.leslie3d-134B.trace.txt.gz	512	16	18148272,18
416.gamess-875B.trace.txt.gz	512	16	25396472,74
445.gobmk-17B.trace.txt.gz	512	16	27266303,74
401.bzip2-226B.trace.txt.gz	512	16	53483602,23
453.povray-887B.trace.txt.gz	512	16	59753896,33
482.sphinx3-1100B.trace.txt.gz	512	16	89783832,48
410.bwaves-1963B.trace.txt.gz	512	16	132230302,2
462.libquantum-1343B.trace.txt.gz	512	16	208353011
459.GemsFDTD-1169B.trace.txt.gz	512	16	213211413
483.xalancbmk-127B.trace.txt.gz	512	16	278509515,1
403.gcc-16B.trace.txt.gz	512	16	300373570,2
471.omnetpp-188B.trace.txt.gz	512	16	307950864,7
433.milc-127B.trace.txt.gz	512	16	315884151,8
465.tonto-1769B.trace.txt.gz	512	16	498043643,1
470.lbm-1274B.trace.txt.gz	512	16	518880375
450.soplex-247B.trace.txt.gz	512	16	1905753963
429.mcf-184B.trace.txt.gz	512	16	5185454093
Media geométrica			64922463.4317292
400.perlbenc-41B.trace.txt.gz	512	32	4320104,864
454.calculix-104B.trace.txt.gz	512	32	4607905,124
464.h264ref-30B.trace.txt.gz	512	32	4857462,699
444.namd-120B.trace.txt.gz	512	32	11814540,92
481.wrf-1170B.trace.txt.gz	512	32	14892968,38
456.hmm-191B.trace.txt.gz	512	32	15636517,12
473.astar-153B.trace.txt.gz	512	32	15914515,28
435.gromacs-111B.trace.txt.gz	512	32	16189021,7
436.cactusADM-1804B.trace.txt.gz	512	32	16576085,72
458.sjeng-1088B.trace.txt.gz	512	32	17201361,27
437.leslie3d-134B.trace.txt.gz	512	32	18148272,18
416.gamess-875B.trace.txt.gz	512	32	25396472,74
445.gobmk-17B.trace.txt.gz	512	32	27266303,74
401.bzip2-226B.trace.txt.gz	512	32	53483602,23
453.povray-887B.trace.txt.gz	512	32	59753896,33
482.sphinx3-1100B.trace.txt.gz	512	32	89785422,52
410.bwaves-1963B.trace.txt.gz	512	32	132230302,2
462.libquantum-1343B.trace.txt.gz	512	32	208353011
459.GemsFDTD-1169B.trace.txt.gz	512	32	213211413
483.xalancbmk-127B.trace.txt.gz	512	32	278509515,1
403.gcc-16B.trace.txt.gz	512	32	300373570,2
471.omnetpp-188B.trace.txt.gz	512	32	307950864,7
433.milc-127B.trace.txt.gz	512	32	315884151,8
465.tonto-1769B.trace.txt.gz	512	32	497944695,9
470.lbm-1274B.trace.txt.gz	512	32	518880375
450.soplex-247B.trace.txt.gz	512	32	1905859872
429.mcf-184B.trace.txt.gz	512	32	5187331926
Media geométrica			64923032.488244

Tabla 9: Resultados de la simulación del caché multinivel con L3

Trace file	Capacidad L3	Asociatividad L3	AMAT (ciclos)
400.perlbenc-41B.trace.txt.gz	1024	16	4320104,864
454.calculix-104B.trace.txt.gz	1024	16	4607905,124
464.h264ref-30B.trace.txt.gz	1024	16	4857462,699
444.namd-120B.trace.txt.gz	1024	16	11814540,92
481.wrf-1170B.trace.txt.gz	1024	16	14892968,38
456.hmmer-191B.trace.txt.gz	1024	16	15636517,12
473.astar-153B.trace.txt.gz	1024	16	15914555,01
435.gromacs-111B.trace.txt.gz	1024	16	16189021,7
436.cactusADM-1804B.trace.txt.gz	1024	16	16576085,72
458.sjeng-1088B.trace.txt.gz	1024	16	17201361,27
437.leslie3d-134B.trace.txt.gz	1024	16	18148272,18
416.gamess-875B.trace.txt.gz	1024	16	25396472,74
445.gobmk-17B.trace.txt.gz	1024	16	27266303,74
401.bzip2-226B.trace.txt.gz	1024	16	53483602,23
453.povray-887B.trace.txt.gz	1024	16	59753896,33
482.sphinx3-1100B.trace.txt.gz	1024	16	89786724,33
410.bwaves-1963B.trace.txt.gz	1024	16	132230302,2
462.libquantum-1343B.trace.txt.gz	1024	16	208353011
459.GemsFDTD-1169B.trace.txt.gz	1024	16	213211413
483.xalancbmk-127B.trace.txt.gz	1024	16	278509515,1
403.gcc-16B.trace.txt.gz	1024	16	300373847,2
471.omnetpp-188B.trace.txt.gz	1024	16	307955271,4
433.milc-127B.trace.txt.gz	1024	16	315884151,8
465.tonto-1769B.trace.txt.gz	1024	16	500166765,7
470.lbm-1274B.trace.txt.gz	1024	16	518880375
450.soplex-247B.trace.txt.gz	1024	16	1922144315
429.mcf-184B.trace.txt.gz	1024	16	5240194552
Media geométrica			64978678.6104105
400.perlbenc-41B.trace.txt.gz	1024	32	4320104,864
401.bzip2-226B.trace.txt.gz	1024	32	53483602,23
403.gcc-16B.trace.txt.gz	1024	32	300373847,2
410.bwaves-1963B.trace.txt.gz	1024	32	132230302,2
416.gamess-875B.trace.txt.gz	1024	32	25396472,74
429.mcf-184B.trace.txt.gz	1024	32	5240855067
433.milc-127B.trace.txt.gz	1024	32	315884151,8
435.gromacs-111B.trace.txt.gz	1024	32	16189021,7
436.cactusADM-1804B.trace.txt.gz	1024	32	16576085,72
437.leslie3d-134B.trace.txt.gz	1024	32	18148272,18
444.namd-120B.trace.txt.gz	1024	32	11814540,92
445.gobmk-17B.trace.txt.gz	1024	32	27266303,74
450.soplex-247B.trace.txt.gz	1024	32	1922539603
453.povray-887B.trace.txt.gz	1024	32	59753896,33
454.calculix-104B.trace.txt.gz	1024	32	4607905,124
456.hmmer-191B.trace.txt.gz	1024	32	15636517,12
458.sjeng-1088B.trace.txt.gz	1024	32	17201361,27
459.GemsFDTD-1169B.trace.txt.gz	1024	32	213211413
462.libquantum-1343B.trace.txt.gz	1024	32	208353011
464.h264ref-30B.trace.txt.gz	1024	32	4857462,699
465.tonto-1769B.trace.txt.gz	1024	32	500166765,7
470.lbm-1274B.trace.txt.gz	1024	32	518880375
471.omnetpp-188B.trace.txt.gz	1024	32	307955271,4
473.astar-153B.trace.txt.gz	1024	32	15914555,01
481.wrf-1170B.trace.txt.gz	1024	32	14892968,38
482.sphinx3-1100B.trace.txt.gz	1024	32	89786724,33
483.xalancbmk-127B.trace.txt.gz	1024	32	278509515,1
Media geométrica			64979476.8145601

Si ordenamos los trazes de menor a mayor AMAT en el caso de solamente L1. Usando ese orden, si graficamos el AMAT de sólo usar L1, el mejor caso de L1+L2 y el mejor caso de L1+L2+L3, obtenemos:

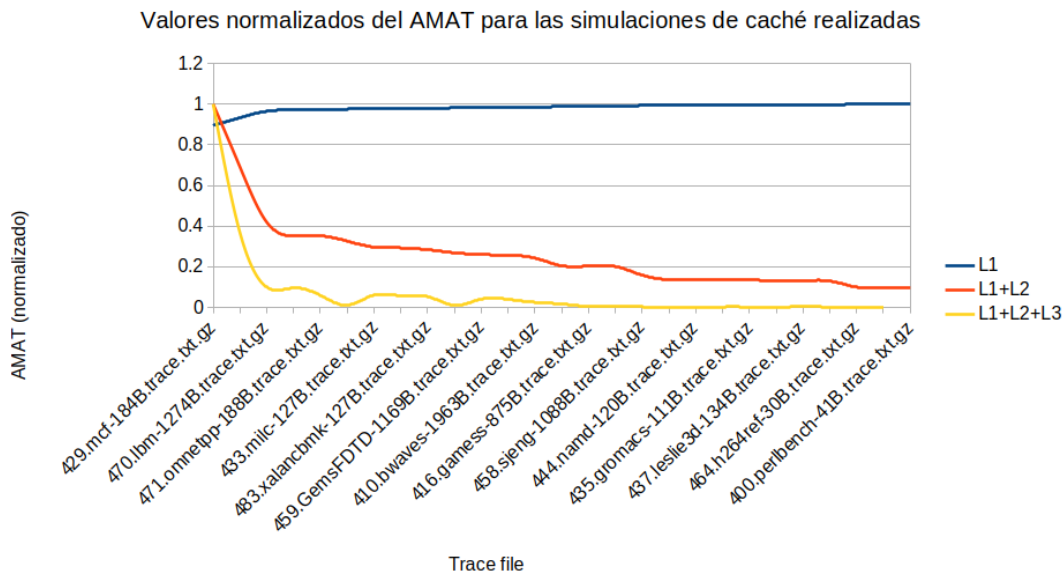


Figura 8: Comparación de los AMAT multinivel.

Al utilizar un solo cache, se llega a saturar, y vemos que se empieza a comportar de manera constante independientemente de qué trace se simule.

Luego, al estudiar las demás gráficas que corresponden a cachés multiniveles de 2 y 3 instancias, encontramos una reducción significativa en el AMAT en ambos casos. Cabe destacar aquí que los normalizamos los valores respecto a los máximos para poder compararlos y utilizamos como referencia el orden ascendente del AMAT para L1 según se solicita en el enunciado.

Estudiando la gráfica a nivel de simulaciones, algunos traces presentan mejoras significativas en AMAT al pasar de una configuración a otra, mientras que en otros la mejora es menor o incluso inexistente. L1+L2+L3 ofrece el mejor rendimiento: Esta configuración tiene el menor AMAT, lo que significa que requiere la menor cantidad de ciclos de reloj para acceder a la memoria.

Además, la reducción del AMAT al pasar de L1 a L1+L2 y luego a L1+L2+L3 demuestra el impacto positivo de la jerarquía de cachés en la reducción del tiempo de acceso a la memoria.

Estos resultados se pueden interpretar como “niveles” o “círculos” de cercanía, donde datos descartados de ciclos anteriores tienen una accesibilidad mayor que si se descartaran y hubiera que llamarlos desde disco o memoria, esto a la vez que no obstruyen el paso a bloques de una necesidad más inmediata, lo que le da mayor fuerza a las razones explicadas con anterioridad.

Esto en cuanto a las tendencias generales que se pueden apreciar en la figura. Se discuten a continuación tres casos específicos en cuanto a su rendimiento respecto a la media:

Trace *464.h264ref-30B.trace.txt.gz*:

Se presenta la mayor mejora en AMAT (90 %) al pasar de L1 a L1+L2+L3. Posiblemente se deba a su alta localidad de datos, lo que significa que accede a un conjunto relativamente pequeño de datos con frecuencia. El tamaño de L2 y L3 parece ser adecuado para almacenar la mayoría de estos datos, lo que explica la significativa reducción del AMAT.

Trace *454.calculix-104B.trace.txt.gz*:

Este trace presenta una mejora modesta en AMAT (40 %) al pasar de L1 a L1+L2+L3. Posiblemente se deba a su patrón de acceso a memoria menos predecible, lo que significa que accede a un conjunto más amplio de datos con menor frecuencia. El tamaño de L2 y L3 podría no ser suficiente

para almacenar todos los datos que se acceden con frecuencia, lo que limita la mejora del AMAT.

Trace *473.astar-153B.trace.txt.gz*:

Este trace presenta una mejora inexistente en AMAT al pasar de L1 a L1+L2. Posiblemente se deba a un tamaño de L2 insuficiente para almacenar los datos que se acceden con frecuencia. La adición de L3 sí mejora el AMAT, pero en menor medida que en otros traces.

Como consecuencia de esto, se encuentra que si bien la mejora en términos generales es muy buena, se sigue teniendo una dependencia a las llamadas específicas a memoria que son inherentes al proceso que se quiere optimizar, dígase el alojamiento ordenado de datos cercanos al procesador.

2. Conclusiones

De la primera parte, podemos concluir que los cambios en la caché no necesariamente afecta el rendimiento de la caché. Ya que puede que se utilicen datos o localidades en la memoria que no se vean afectadas por estos cambios. Por otro lado, esta parte de la tarea también ilustró muy bien lo que es las optimizaciones vistas en clase. Ya que se puede ver muy bien