Análise dos Impactos na Memória Cache

Eduardo Felipe Bauer e Gabriel Reinhardt dos Reis

Resumo

Neste trabalho é abordado e analisada a simulação de uma memória cache, explorando seus impactos positivos e negativos referente às diferentes variações possíveis em seus parâmetros de construção. É possível visualizar graficamente os resultados das simulações, tornando assim mais fácil e divertido a abstração do conhecimento abordado.

Palavras-chave

Memória Cache, bloco, simulação, parâmetro, taxa de acerto.

1. Programa de Simulação

Para a realização das simulações de memória cache e registro dos resultados, desenvolvemos um programa, cujo código foi desenvolvido em C. O programa é desenvolvido em módulos, visando a facilidade para programar em grupo, com sua estrutura principal composta por um struct cache, formado por três variáveis inteiras, uma para o rótulo, uma para o LRU e outra para o dirty bit.

Ao iniciar o programa pelo arquivo main.c, é solicitado parâmetros , pela função “lerParametros”, para a realização da simulação da cache, como política de escrita, tamanho do bloco, tamanho da cache, associatividade do conjunto, política de substituição e o nome do arquivo para preencher com o resultado. Os parâmetros de tempo das memórias foram preestabelecidos, como 5 ns para a cache e 70 ns para a memória principal e o tamanho do endereço como 32 bits.

Após a leitura dos parâmetros de entrada, é alocada a quantidade de blocos presentes na memória cache, utilizando o tamanho da cache / pelo tamanho do bloco. E realizando a limpeza da cache pela função, “limparCache”, com a finalidade de colocar as variáveis rótulo como -1, LRU como 0 e dirty bit como 0.

Em seguida é feito a leitura do arquivo de entrada pela função “lerArquivo”, já estabelecido como “oficial.cache”, contendo 51200 endereços hexas decimais e sua ação, como R para ler e W para escrever. Com seu hexas decimais codificados para binários pela função “decoficicarHexa” e armazenados em um vetor binarios e suas ações no vetor acoes.

Com todas as etapas anteriores realizadas com sucesso, é iniciado a simulação, com a retirada do rótulo e conjunto do endereço binário, se a ação desse endereço for de leitura, é contabilizado mais um entrada de leitura e chamado a função “lerCache”. Ela realiza a verificação do rótulo dentro do conjunto, se ele estiver presente é contabilizado um hit, senão é contabilizado um miss. Com um miss detectado, é realizado uma contabilização de leitura na memória principal e a substituição do bloco conforme a política de substituição e escrita já passadas, para cada tipo de política com suas peculiaridades e diferenças.

Caso a ação do endereço for de escrita, é feito o acréscimo de mais uma entrada de escrita e chamada a função “escreverCache”. Ela realiza verificação do rótulo dentro do conjunto, se ele estiver presente é contabilizado um hit, senão é contabilizado um miss. Com um miss detectado, é realizada a conferência da política de escrita e substituição e é realizada, se necessário, uma leitura na memória principal ou escrita na memória principal, conforme as políticas.

Com todos os endereços do arquivo de entrada já simulados, é feito o cálculo das taxas de acerto, tanto para a escrita, leitura e global, como o tempo médio de acesso. E todos os parâmetros passados para gravar o arquivo de saída com a função “gravarArquivo”

O programa pode ser encontrado no drive: https://drive.google.com/drive/folders/1XNsKbObeSeLSCRqU644o373Zflk8cTT6?usp=sharing.

2. Impacto do Tamanho da Cache

Para este experimento, empregamos uma política de write-through e uma associatividade de 4 blocos. O tamanho do bloco foi mantido constante em 128 bytes, conforme detalhado na Tabela 1. A principal variável modificada foi o número de linhas, o que impactou diretamente o tamanho total da cache. Os dados resultantes dessa variação estão apresentados na Tabela 2, e a análise desses dados gerou o gráfico da Figura 1.

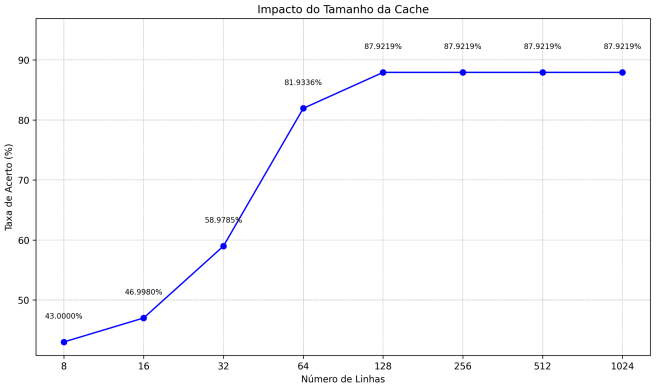

Figura 1. Impacto do tamanho da cache.

É possível notar no gráfico acima que, à medida que o tamanho da cache aumenta, a taxa de acerto também cresce. Esse fenômeno é uma consequência direta do princípio da localidade de referência, um pilar fundamental na arquitetura de computadores. Especificamente, a localidade temporal postula que se um item de dados foi acessado recentemente, é provável que ele seja acessado novamente em breve; uma cache maior retém esses dados por mais tempo, elevando a probabilidade de um acerto. Similarmente, a localidade espacial sugere que se um item de dados foi acessado, é provável que itens próximos a ele na memória também sejam acessados em breve; uma cache de maior capacidade pode armazenar uma porção mais extensa da memória principal, capturando eficazmente esses dados adjacentes e mitigando as falhas compulsórias (cold misses) e as falhas por capacidade (capacity misses). Em suma, quanto maior a cache, mais blocos ela pode comportar, o que, em alinhamento com o princípio da localidade, amplia significativamente a chance de o dado procurado já estar presente na cache [3].

3. Impacto do Tamanho do Bloco

Para este experimento, empregamos uma política de write-through e uma associatividade de 2 blocos. O tamanho da cache foi mantido constante em 8K bytes e o algoritmo de substituição foi o LRU, conforme detalhado na Tabela 3. A principal variável modificada foi o tamanho do bloco, o que impactou diretamente o número total de blocos. Os dados resultantes dessa variação estão apresentados na Tabela 4, e a análise desses dados gerou o gráfico da Figura 2.

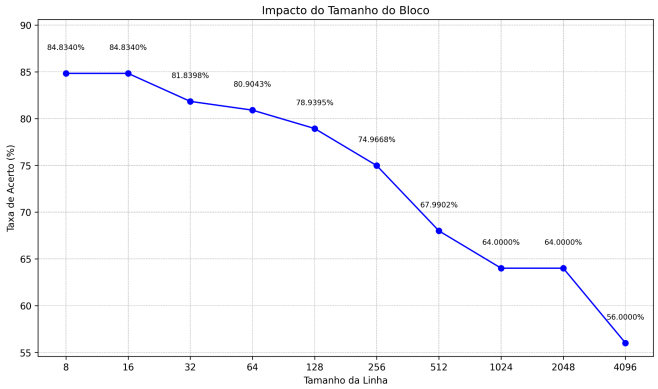

Figura 2. Impacto do tamanho do bloco.

É possível observar no gráfico acima que, à medida que o tamanho do bloco aumenta, a taxa de acerto diminui. Esse fenômeno pode ser explicado por dois fatores principais que operam em oposição ao princípio da localidade espacial. Primeiramente, com um tamanho de cache fixo (8KB, neste caso), blocos maiores significam que menos blocos podem ser armazenados no total. Isso pode levar a mais falhas por capacidade, pois a cache consegue reter uma porção menor dos dados ativos do programa. Em segundo lugar, blocos maiores podem introduzir o problema de "poluição" do cache: se o programa acessa apenas uma pequena porção de um bloco grande, o restante do bloco é carregado desnecessariamente para a cache. Isso desperdiça espaço valioso na cache com dados que provavelmente não serão usados, expulsando prematuramente blocos que seriam necessários e resultando em uma diminuição na taxa de acerto [4]

4. Impacto da Associatividade

Para este experimento, empregamos uma política de write-back, tamanho do bloco de 128 bytes. O tamanho da cache foi mantido constante em 8K bytes e o algoritmo de substituição foi o LRU, conforme detalhado na Tabela 5. A principal variável modificada foi a associatividade, o que impactou diretamente o tamanho e o número de conjuntos. Os dados resultantes dessa variação estão apresentados na Tabela 6, e a análise desses dados gerou o gráfico da Figura 3.

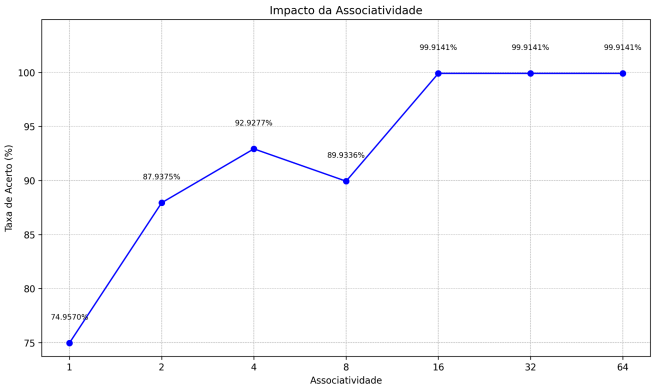

Figura 3. Impacto da Associatividade.

É possível observar no gráfico acima, que a taxa de acerto vai aumentando conforme a associatividade vai aumentando, até se aproximar de 100% de acerto, onde o aumento da associatividade não causa mais influência [5]. De acordo com os conceitos ensinados em [1], é possível analisar que a queda na taxa de acerto da cache com 8 de associatividade pode ocorrer devido a um fenômeno que ocorre e pode ser chamado de conflito de escrita ou “colisões de escrita”. Em sistemas com alta associatividade, vários blocos de memória podem ser mapeados para o mesmo conjunto de linhas de cache. Se o programa acessa frequentemente diferentes dados que mapeiam para o mesmo conjunto, eles podem substituir uns aos outros no cache, mesmo que haja espaço disponível em outros conjuntos. Isso leva a um aumento de falhas de cache (misses) e, consequentemente, a uma queda na taxa de acerto. Logo é uma característica específica que indica que quando mapeamos 8 blocos por conjunto nesta MP e Cache, há um volume muito alto de acessos em um ou alguns conjuntos específicos, necessitando de substituição.

5. Impacto da Política de Substituição

Para este experimento, empregamos uma política de write-through, tamanho do bloco de 128 bytes. A associatividade foi mantida constante em 4 blocos por conjunto. Foi simulado tanto o algoritmo de substituição LRU, quanto o aleatório, conforme detalhado na Tabela 7 e 9 consecutivamente. Os dados resultantes dessa variação estão apresentados na Tabela 8 e 10 consecutivamente, e a análise desses dados gerou o gráfico da Figura 4 e 5.

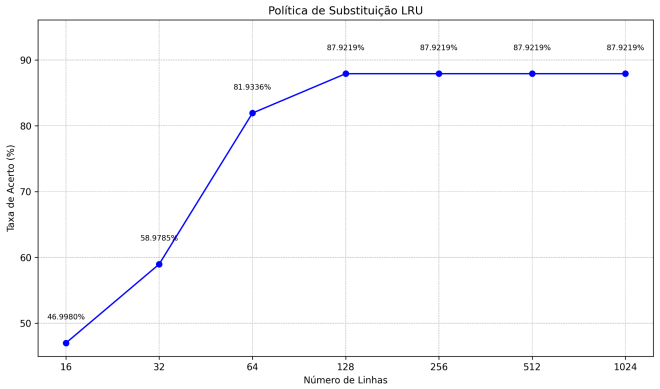

Figura 4. Impacto da Política de Substituição com LRU

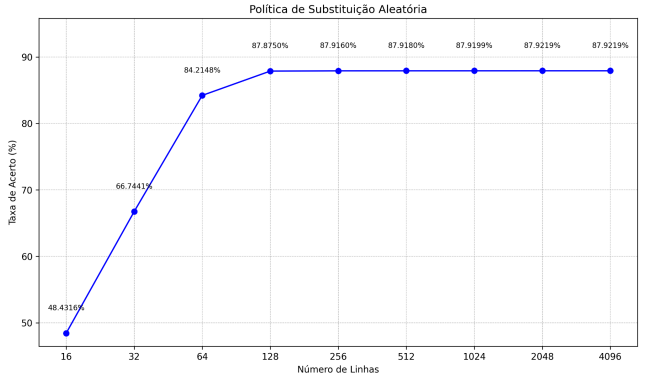

Figura 5. Impacto da Política de Substituição com Aleatório.

É possível notar na figura 4 e 5, que mesmo mudando o método de substituição, quando aumentamos o número de linhas, ambos os métodos possuem uma taxa de acerto semelhante. Porém temos que considerar também que por ser uma memória cache pequena, a estratégia de substituição do LRU: localidade temporal, não se faz muito útil [6].

6. Largura de Banda da Memória

Para este experimento, variamos os parâmetros de escrita em Write-through e Write-Back; Associatividade em 2 e 4; Tamanho da cache em 8K bytes e 16K bytes; Números de linhas de 64 blocos e 128.Conseguimos visualizar os resultados e as variações dos parâmetros na Tabela 11.

Largura de Banda de Memória		Cache 8 Kbytes				Cache 16 Kbytes			
		Associatividade 2		Associatividade 4		Associatividade 2		Associatividade 4	
		64 linhas	128 linhas	64 linhas	128 linhas	64 linhas	128 linhas	64 linhas	128 linhas
leituras totais	write-through	45056	45056	45056	45056	45056	45056	45056	45056
escritas totais		6144	6144	6144	6144	6144	6144	6144	6144
leituras MP		3633	4639	2611	3106	56	1062	56	40
escritas MP		6144	6144	6144	6144	6144	6144	6144	6144
média de leituras MP		1836,5							
média de escritas MP		6144							
leituras totais	write-back	45056	45056	45056	45056	45056	45056	45056	45056
escritas totais		6144	6144	6144	6144	6144	6144	6144	6144
leituras MP		5170	6176	3126	3621	2615	3621	2615	2599
escritas MP		1026	1026	515	515	1026	1026	515	515
média de leituras MP		3173,5							
média de escritas MP		770,5							

Figura 6. Tabela Largura de banda

De acordo com [2], embora a política de write-back ofereça um desempenho superior, sua principal desvantagem reside na maior complexidade de implementação quando comparada à simplicidade da write-through. Conforme ilustrado na Figura 5, a política de write-back resulta em um tráfego de memória ligeiramente reduzido em comparação com a política de write-through, que é inerentemente mais lenta. A lentidão da write-through se deve ao fato de que cada operação de escrita exige um acesso direto à memória principal.

7. Avaliação Global

Foi possível observar no desenvolvimento do sistema uma maior dificuldade na interpretação das características das políticas de escrita e leitura, para fazer a simulação adequadamente em C. Como seus conceitos, demonstrado na figura 7 abaixo.

Leitura:
- acesso a cache para procurar pelo rótulo dentro do conjunto:
  - achou: hit
  - não achou: miss
    se o dirty bit do bloco selecionado está em 1, ele deverá fazer a escrita na memória e depois fazer a substituição?
    - substituição aleatória: escolhe um bloco do conjunto aleatório e faz a substituição
    - substituição LRU: encontra o bloco menos usado e faz a substituição, mas e se tiver dois blocos sendo ambos o menos usado?

Escrita:
- acesso a cache para procurar pelo rótulo dentro do conjunto:
  - achou: hit
    - se for write-back: olhar o dirty bit, se estiver 1, escrever na memória principal também, se tiver em 0 só muda pra 1
    - se for write-through: escrever na memória principal e na cache
  - não achou: miss
    - se for write-back: escrevo na memória principal e trago para a cache
      - substituição aleatória: escolhe um bloco do conjunto aleatório e faz a substituição
    - substituição LRU: encontra o bloco menos usado e faz a substituição, mas e se tiver dois blocos sendo ambos o menos usado?
    - se for write-through: escrevo na memória principal apenas

Figura 7. Ilustração representativa da dificuldade de interpretação do processo de funcionamento da cache no programa

Com mais dificuldade na parte de interpretação do conceito quanto acontece um miss, na política de substituição LRU e na política de escrita write-back com o seu dirty bit.

Com todas as simulações feitas, é possível identificar que a cache com associatividade maior ou igual à 16, tamanho do bloco de 128 bytes, tamanho da cache de 8K bytes e o algoritmo de substituição LRU, teve o melhor desempenho. Isso é possível, devido a avaliação do percentual de hits extremamente positivos da simulação em questão.

Claro, a taxa de acerto pode estar a mercê de possíveis conflitos de escrita, porém isso vai depender da sequência de endereços que será acessado na cache, se será um equilíbrio de acessos entre os conjuntos ou mais centralizados em alguns conjuntos específicos, isto considerando os parâmetros atuais [7].

8. Bibliografia

[1] Tanenbaum, A. S., & Austin, T. (2013). *Organização Estruturada de Computadores* (6ª ed.). Pearson.

[2] Stallings, W. (2018). *Arquitetura e Organização de Computadores* (10ª ed.). Pearson.

[3] Tanenbaum, A. S., & Austin, T. (2013). *Organização Estruturada de Computadores* (6ª ed.). Pearson pág. 116.

[4] The Grainger College of Engineering. (n.d.). Cache performance (Slide 3). Disponível em https://courses.grainger.illinois.edu/cs232/sp2010/lectures/L17.pdf

[5] Centro de Informática da UFPE. (n.d.). Memórias cache: uma introdução. Disponível em https://www.cin.ufpe.br/~can/Arquivos/cache.notes-2.pdf

[6] Algoritmos de substituição - Memória cache - FIFO - LRU - LFU. (n.d.). Disponível em https://www.youtube.com/watch?v=vVK6ffd9Aw4

[7] Memória Cache na Organização de Computadores (n.d.). Disponível em https://www.geeksforgeeks.org/cache-memory-in-computer-organization/

9. Anexo A

Impacto do Tamanho da Cache

Tabela 1 - Parâmetros da Simulação

Parâmetro	Valor
Tamanho da linha	128 bytes
Associatividade	4 blocos
Política de Escrita	write-through
Algoritmo de Substituição	LRU

Tabela 2 - Resultados da Simulação

Número de Linhas	Taxa de Acerto (%)	Tempo médio de Acesso (ns)	Leituras na MP	Escritas na MP
8	43	44,9	23040	6144
16	46,998	42,1014	20993	6144
32	58,9785	33,715	14859	6144
64	81,9336	17,6465	3106	6144
128	87,9219	13,4547	40	6144
256	87,9219	13,4547	40	6144
512	87,9219	13,4547	40	6144
1024	87,9219	13,4547	40	6144

IMPACTO DO TAMANHO DO BLOCO:

Tabela 3 - Parâmetros da Simulação

Parâmetro	Valor
Tamanho da Cache	8Kbytes
Associatividade	2 blocos
Política de Escrita	write-through
Algoritmo de Substituição	LRU

Tabela 4 - Resultados da Simulação

Número de Linhas	Taxa de Acerto (%)	Tempo médio de Acesso (ns)	Leituras na MP	Escritas na MP
8	84,834	15,6162	1621	6144
16	84,834	15,6162	1621	6144
32	81,8398	17,7121	3154	6144
64	80,9043	18,367	3633	6144
128	78,9395	19,7424	4639	6144
256	74,9668	22,5232	6673	6144
512	67,9902	27,4068	10245	6144
1024	64	30,2	12288	6144
2048	64	30,2	12288	6144
4096	56	35,8	16384	6144

IMPACTO DA ASSOCIATIVIDADE:

Tabela 5 - Parâmetros da Simulação

Parâmetro	Valor
Tamanho do Bloco	128 bytes
Tamanho da Cache	8 Kbytes
Política de Escrita	write-back
Algoritmo de Substituição	LRU

Tabela 6 - Resultados da Simulação

Número de Linhas	Taxa de Acerto (%)	Tempo médio de Acesso (ns)	Leituras na MP	Escritas na MP
1	74,957	22,5301	12822	2048
2	87,9375	13,4438	6176	1026
4	92,9277	9,9506	3621	515
8	89,9336	12,0465	5154	1026
16	99,9141	5,0602	44	4
32	99,9141	5,0602	44	4
64	99,9141	5,0602	44	4

Impacto da Política de Substituição LRU :

Tabela 7 - Parâmetros da Simulação

Parâmetro	Valor
Tamanho do Bloco	128 bytes
Associatividade	4 blocos
Política de Escrita	write-through
Algoritmo de Substituição	LRU

Tabela 8 - Resultados da Simulação

Número de Linhas	Taxa de Acerto (%)	Tempo médio de Acesso (ns)	Leituras na MP	Escritas na MP
16	46,998	42,1014	20993	6144
32	58,9785	33,715	14859	6144
64	81,9336	17,6465	3106	6144
128	87,9219	13,4547	40	6144
256	87,9219	13,4547	40	6144
512	87,9219	13,4547	40	6144
1024	87,9219	13,4547	40	6144

Impacto da Política de Substituição ALEATÓRIO :

Tabela 9 - Parâmetros da Simulação

Parâmetro	Valor
Tamanho do Bloco	128 bytes
Associatividade	4 blocos
Política de Escrita	write-through
Algoritmo de Substituição	Aleatória

Tabela 10 - Resultados da Simulação

Número de Linhas	Taxa de Acerto (%)	Tempo médio de Acesso (ns)	Leituras na MP	Escritas na MP
16	48,4316	41,0979	20259	6144
32	66,7441	28,2791	10883	6144
64	84,2148	16,0496	1938	6144
128	87,875	13,4875	64	6144
256	87,916	13,4588	43	6144
512	87,918	13,4574	42	6144
1024	87,9199	13,4561	41	6144
2048	87,9219	13,4547	40	6144
4096	87,9219	13,4547	40	6144

Largura de Banda da Memória:

Tabela 11 - Resultados da Simulação

Largura de Banda de Memória		Cache 8 Kbytes				Cache 16 Kbytes			
		Associatividade 2		Associatividade 4		Associatividade 2		Associatividade 4	
		64 linhas	128 linhas	64 linhas	128 linhas	64 linhas	128 linhas	64 linhas	128 linhas
leituras totais	write-through	45056	45056	45056	45056	45056	45056	45056	45056
escritas totais		6144	6144	6144	6144	6144	6144	6144	6144
leituras MP		3633	4639	2611	3106	56	1062	56	40
escritas MP		6144	6144	6144	6144	6144	6144	6144	6144
média de leituras MP		1836,5							
média de escritas MP		6144							
leituras totais	write-back	45056	45056	45056	45056	45056	45056	45056	45056
escritas totais		6144	6144	6144	6144	6144	6144	6144	6144
leituras MP		5170	6176	3126	3621	2615	3621	2615	2599
escritas MP		1026	1026	515	515	1026	1026	515	515
média de leituras MP		3373,5							
média de escritas MP		770,5							