

# Localidades Temporal e Espacial no Acesso à Memória

Gabriel H. Oliveira, Lucas A. Barduíno, Lucas de Oliveira, Pedro M. Rodrigues, Vinícius S. Viana  
Departamento de Ciência da Computação (DCC)  
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)  
Belo Horizonte, Brasil

{1400538, 1382359, 1406610, 1373336, 1387659} @sga.pucminas.br

**Resumo**—Este artigo apresenta um estudo sobre a hierarquia de memória com o objetivo de validar os conceitos de localidade temporal e espacial em sistemas de cache. Utilizando o simulador Amnesia, foram criados cenários de simulação variando parâmetros como tamanho de blocos, associatividade e políticas de substituição de cache. Os resultados demonstram o impacto dessas variações no desempenho, medido em termos de taxas de miss e hit, e comprovam a importância de um projeto otimizado para diferentes tipos de acesso à memória.

**Palavras-chave**—Arquitetura de Computadores, Memória, Cache, Memória Virtual, Localidade Espacial, Localidade Temporal, Simulador Amnesia

## I. INTRODUÇÃO

A hierarquia de memória é um dos componentes mais críticos para o desempenho de sistemas computacionais modernos. A performance da memória é determinada, em grande parte, pela eficiência do cache em explorar os princípios de localidade temporal e espacial. No entanto, o dimensionamento e a configuração de sistemas de cache, como tamanho de bloco, associatividade e políticas de substituição, podem influenciar significativamente a taxa de acertos (hit) e falhas (miss) de memória.

Um problema em aberto, que este artigo busca responder, é: Como diferentes configurações de cache impactam o desempenho do sistema em termos de taxa de acertos e falhas? O objetivo deste trabalho é explorar cenários variados no simulador Amnesia, para verificar como a variação de parâmetros de cache influencia o desempenho geral.

A contribuição científica do presente estudo está na análise quantitativa dessas variações, que é visualizada por meio de gráficos e discussões sobre os resultados obtidos. O artigo está organizado da seguinte forma: a Seção 2 apresenta trabalhos correlatos, a Seção 3 descreve o simulador Amnesia utilizado no trabalho, a Seção 4 mostra a Metodologia e Proposta de Arquiteturas, a Seção 5 relata a avaliação dos resultados e, por fim, a Seção 6 traz as conclusões.

## II. TRABALHOS CORRELATOS

Diversos estudos abordam a análise da hierarquia de memória e o impacto de diferentes configurações de cache no desempenho dos sistemas computacionais. Nesta seção, apresentamos três trabalhos que serviram como base para o desenvolvimento do presente estudo, ressaltando como cada um contribuiu para o avanço do conhecimento em simulações de memória e configuração de caches.

### A. Analysis of the Spatial and Temporal Locality in Data Accesses

O artigo enfatiza a crescente relevância da otimização de cache para atingir alta performance computacional, especialmente em sistemas chip-multiprocessor (CMP), que normalmente exibem uma taxa de erros de Cache superior à dos uni-processadores. Para essa otimização, são necessárias informações sobre a localidade de acesso para auxiliar nas tarefas de alocação de dados, transformação de dados e transformação de código, comumente utilizadas para aumentar a utilização de dados armazenados em cache, visando melhorar a taxa de acertos de cache. A pesquisa apresenta uma ferramenta de análise capaz de identificar a relação espacial e temporal entre os acessos à memória e fornecer informações necessárias para aplicar algumas técnicas de otimização. No contexto do estudo, esta ferramenta de análise pode ser extremamente valiosa para compreender como diferentes técnicas de otimização podem impactar a localidade temporal e espacial e, consequentemente, o desempenho do sistema de memória.

### B. Embedded Systems Review

Gonzalez e Souza (2021) concentraram-se na simulação de cache em sistemas embarcados utilizando o simulador DineroIV. Eles exploraram a variação do tamanho dos blocos de cache e mostraram que, para sistemas embarcados, blocos menores (32B e 64B) tendem a oferecer melhor desempenho para programas com localidade espacial limitada, enquanto blocos maiores podem ser vantajosos para acessos sequenciais. A abordagem deste trabalho foi crucial para definirmos nossos cenários de variação do tamanho do bloco. Inspirados por suas descobertas, implementamos um cenário semelhante no simulador Amnesia, variando o tamanho dos

blocos de cache (32B, 64B e 128B). O estudo de Gonzalez e Souza nos orientou a analisar detalhadamente os efeitos da localidade espacial no desempenho, especialmente em aplicações com diferentes padrões de acesso à memória.

### III. SIMULADOR AMNESIA

Para o desenvolvimento deste artigo, as simulações das arquiteturas de memória foram realizadas utilizando o Simulador Amnesia. Trata-se de um recurso educacional gratuito que funciona como um simulador da Hierarquia de Memória na arquitetura de *Von Neumann*. Sua principal função é demonstrar, de forma intuitiva e didática, o processo de busca por instruções, leitura, gravação e escrita nas memórias Cache, Principal, Virtual e Secundária. O simulador foi desenvolvido em 2007 por estudantes do Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP).

Uma das vantagens desse simulador é a possibilidade de configurar a arquitetura através de um arquivo ".xml" (Extensible Markup Language), onde são definidos parâmetros essenciais como: palavras por bloco, tamanhos das memórias e dos dados, ciclos de leitura e escrita, número de vias da cache, políticas de substituição, entre outros.

Além disso, com a arquitetura configurada, é possível carregar um arquivo de texto ".txt" contendo um trace de execução, que será executado conforme os parâmetros especificados anteriormente. Para este trabalho, foram consideradas apenas as instruções do tipo 2, que correspondem à busca de instruções, seguidas de um número hexadecimal que indica a posição de memória onde a instrução simulada se encontra.

### IV. PROPOSTA DE ARQUITETURA

Considerando a necessidade de uma melhor compreensão dos conceitos de Hierarquia de Memória, este estudo foi conduzido em duas fases distintas. Foram então gerados dois *traces* de execução para a busca de instruções em endereços de memória, uma para cada uma delas, variando de 00 a 31 (em hexadecimal, 00 a 1F).

O *Trace 1* será chamado como Trace Temporal, no qual se busca, principalmente, aproveitar as vantagens do Princípio da Localidade Temporal, utilizando repetidas chamadas ao mesmo endereço de memória em um curto intervalo de tempo. De forma semelhante, o segundo trace, *Trace 2*, tem como objetivo aproveitar o Princípio da Localidade Espacial, realizando chamadas a instruções localizadas em endereços próximos na memória, sendo por isso nomeado como *Trace Espacial*.

Trace 1	10 Linhas do Trace Temporal
2 1c	busca no endereço 1c
2 1e	busca no endereço 1e
2 1e	busca no endereço 1e
2 1c	busca no endereço 1c
2 06	busca no endereço 06

Trace 1	10 Linhas do Trace Temporal
2 1e	busca no endereço 1e
2 15	busca no endereço 15
2 06	busca no endereço 06
2 1c	busca no endereço 1c
2 01	busca no endereço 01

Fig. 1. Tabela de testes Trace Temporal

Trace 2	10 Linhas do Trace Espacial
2 18	busca no endereço 18
2 17	busca no endereço 17
2 16	busca no endereço 16
2 03	busca no endereço 03
2 1d	busca no endereço 1d
2 18	busca no endereço 18
2 17	busca no endereço 17
2 0f	busca no endereço 0f
2 15	busca no endereço 15
2 11	busca no endereço 11

Fig. 2. Tabela de teste Trace Espacial

#### A. Memória Cache

Na primeira fase, o estudo focou na utilização de Memória Principal e Memória Cache para o armazenamento de instruções. Para isso, foi criado um Cenário Base de arquitetura, a partir do qual ajustes em certos parâmetros foram aplicados, resultando em três cenários distintos para comparação. O objetivo era aprimorar o uso das Localidades Temporal e Espacial, conforme descrito na Tabela I.

No Cenário 1, a quantidade de palavras por bloco foi modificada. Inicialmente, a Cache e a RAM armazenavam uma única palavra por bloco. Esse parâmetro foi ajustado para 2, 4, 8 e 16 palavras por bloco, buscando avaliar o impacto no acesso às instruções. Espera-se que essa alteração beneficie o Trace Espacial, já que palavras adjacentes são carregadas para a Cache, mesmo que não sejam solicitadas de imediato, favorecendo o acesso a instruções vizinhas.

No Cenário 2, a associatividade da Memória Cache foi alterada. Partindo de uma única via, a Cache foi ajustada para 2, 4, 8 e 16 vias. Com uma Cache de 16 bytes, essa variação vai do mapeamento direto ao totalmente associativo. Acredita-se que essa mudança beneficie o Trace Temporal, ao minimizar colisões de índices e permitir que instruções permaneçam por mais tempo na Cache.

Finalmente, no Cenário 3, foram introduzidos dois níveis de Memória Cache. No Cenário Base, a Cache era unificada com 16 bytes, mas foi dividida em L1 e L2, com variações de tamanho. Espera-se que essa mudança reduza a taxa de falhas (misses), especialmente quando o nível L2 puder armazenar uma quantidade suficiente de instruções, diminuindo o acesso à Memória Principal.

- Nível 1 com 08 bytes e Nível 2 com 16 bytes;
- Nível 1 com 08 bytes e Nível 2 com 32 bytes;
- Nível 1 com 16 bytes e Nível 2 com 16 bytes.

### B. Memória Virtual

Na proposta de arquitetura de Memória Virtual, conforme detalhado na Tabela II, assim como na fase anterior, foi definido um Cenário Base, além de três cenários adicionais com variações específicas a serem analisadas. Este modelo inclui a Memória Principal, a Memória Secundária (Disco) e a TLB. No Cenário 1, o número de blocos por página na Tabela de Páginas foi ajustado. O valor inicial, que era 2 blocos, foi alterado para 4, 8 e 16 blocos. A expectativa é que essa mudança traga resultados similares aos observados quando se altera o número de palavras por bloco na Cache, onde o Trace Espacial tende a se beneficiar mais do que o Temporal, pois ao transferir uma página do Disco para a Memória Principal com instruções contíguas, pode-se diminuir a ocorrência de Page Faults.

No Cenário 2, houve uma modificação na capacidade de armazenamento da TLB. Como essa estrutura atua de forma similar a uma cache para a Tabela de Páginas, espera-se que, ao ampliar sua capacidade, o número de Page Faults diminua. Assim, foram considerados tamanhos de 4, 6, 8, 10 e 12 bytes para a TLB.

Por último, no Cenário 3, a TLB foi removida da arquitetura para investigar o impacto de sua ausência. Como é sabido que a TLB facilita o processo de tradução de endereços virtuais em físicos, ao evitar constantes acessos à Tabela de Páginas, a expectativa é que a exclusão dessa estrutura aumente significativamente o tempo de acesso, além de gerar uma maior frequência de consultas à Tabela de Páginas.

## V. AVALIAÇÃO DOS RESULTADOS

Ao definir as duas arquiteturas base e os seis cenários de teste, os dois traces foram executados com o objetivo de analisar o comportamento dos acessos à memória e o tempo total de execução, levando em consideração o impacto das Localidades Temporal e Espacial. Para isso, o comportamento será apresentado por meio de gráficos e tabelas, conforme os cenários propostos.

Ciclos por acesso para escrita	20	20	20	20
Tempo de ciclo (ms)	1	1	1	1
<b>Memória Cache</b>	Unificada	Unificada	Unificada	Multinível
Palavras por Bloco	1	<b>2, 4, 8, 16</b>	1	1
Tamanho de Dados (Bytes)	16	16	16	<b>L1 = 8 L2 = 12</b>  <b>L1 = 8 L2 = 32</b>  <b>L1 = 16 L2 = 32</b>
Ciclos por acesso para leitura	1	1	1	1
Ciclos por acesso para escrita	2	2	2	2
Tempo de ciclo (ms)	1	1	1	1
Nº de vias	1	1	2, 4, 8, 16	1
Política de escrita	Write Through	Write Through	Write Through	Write Through
Algoritmo de substituição	LRU	LRU	LRU	LRU

**TABELA 1 - ARQUITETURAS PROPOSTAS COM MEMÓRIA CACHE**

<b>Memória Principal</b>	<b>Cenário Base</b>	<b>Cenário 1</b>	<b>Cenário 2</b>	<b>Cenário 3</b>
Palavras por Bloco	1	2, 4, 8, 16	1	1
Tamanho de Dados (Bytes)	32	32	32	32
Ciclos por acesso para leitura	10	10	10	10

TABELA 11 - ARQUITETURAS PROPOSTAS COM MEMÓRIA VIRTUAL

Memória Principal	Cenário Base	Cenário 1	Cenário 2	Cenário 3
Palavras por Bloco	1	1	1	1
Tamanho de Dados (Bytes)	16	16	16	16
Ciclos por acesso para leitura	10	10	10	10
Ciclos por acesso para escrita	20	20	20	20
Tempo de ciclo (ms)	1	1	1	1
Memória Secundária	-	-	-	-
Palavras por Bloco	1	1	1	1
Tamanho de Dados (Bytes)	32	32	32	32
Ciclos por acesso para leitura	100	100	100	100
Ciclos por acesso para escrita	200	200	200	200

Tempo de ciclo (ms)	1	1	1	1
Algoritmo de substituição	FIFO	FIFO	FIFO	FIFO
TLB	Sim	Sim	Sim	Não
Tamanho de dados (Bytes)	8	8	4,6,10,12	-
Ciclos por acesso para leitura	1	1	1	-
Ciclos por acesso para escrita	2	2	2	-
Tempo de ciclo (ms)	1	1	1	-
Algoritmo de substituição	LRU	LRU	LRU	-

### A. Memória Cache

Para o Trace Temporal, conforme ilustrado na *Figura 1*, observamos um aumento expressivo na taxa de falhas (miss) na memória cache, que passou de 0.21 para 0.42. Esse comportamento pode ser explicado pela menor eficiência na exploração da Localidade Temporal nessa arquitetura, que favorece a busca por blocos contendo várias palavras próximas. Como as instruções no Trace Temporal tendem a acessar posições de memória mais dispersas no tempo, isso leva a uma maior taxa de falhas sempre que a próxima instrução, embora usada recentemente, se encontra em outro bloco, que pode ter sido removido da cache.

Analogamente, conforme representado na *Figura 2*, o tempo de acesso à memória também reflete essa tendência. No caso do Trace Espacial, o tempo de acesso reduziu substancialmente ao aumentar o número de palavras por bloco até 4. No entanto, à medida que o tamanho dos blocos continuava a aumentar, tanto o tempo de acesso quanto a taxa de falhas aumentaram, devido à menor eficiência da Localidade Temporal. Isso acontece porque a relação entre taxa de falhas e tempo de acesso é diretamente proporcional: quando ocorre uma falha na cache, é necessário buscar a instrução na memória principal, o que aumenta significativamente o tempo de execução.

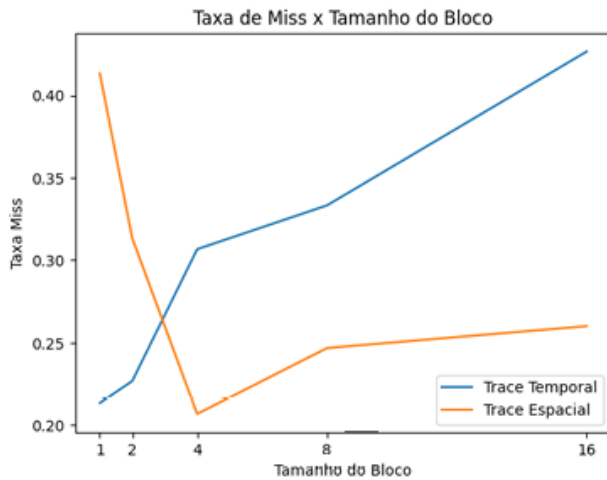


Figura 1: Cenário 1 - Taxa de Falhas (miss) x Tamanho do Bloco

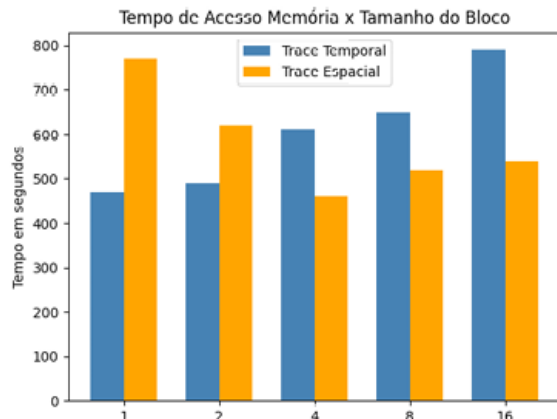


Figura 2: Cenário 1 - Tempo de Acesso à Memória x Tamanho do Bloco

No *Cenário 2*, ao aumentar o número de vias na cache, ocorre um aumento na associatividade, permitindo que mais palavras ocupem o mesmo índice na cache. Isso ajuda a garantir que as palavras acessadas em um curto intervalo de tempo permaneçam na cache por mais tempo, diminuindo as falhas. Assim, o Trace Temporal se beneficia dessa configuração, pois as instruções podem ser reutilizadas antes de serem substituídas.

Como mostrado na *Figura 3*, há uma diferença significativa na taxa de falhas entre os dois traces. No Trace Temporal, a taxa média de falhas foi de aproximadamente 0.23, enquanto no Trace Espacial esse valor foi de 0.40, praticamente o dobro. Isso ocorre porque o aumento de vias favorece a retenção de instruções que exploram a Localidade Temporal, enquanto no Trace Espacial, a substituição rápida de blocos prejudica a localidade.

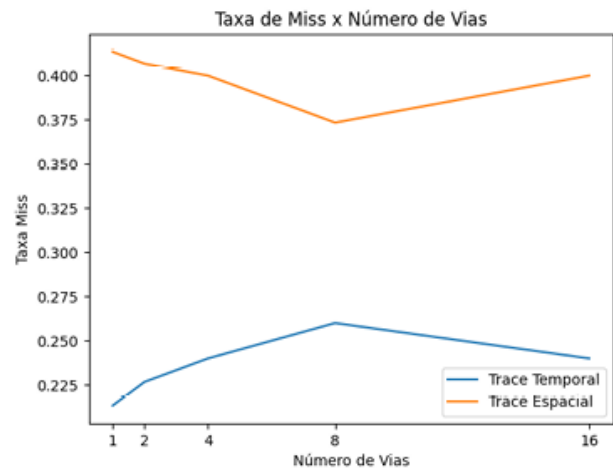


Figura 3: Cenário 2 - Taxa de Falhas (miss) x Número de Vias

A *Figura 4* mostra que o tempo de acesso para o Trace Espacial foi muito maior em comparação ao Trace Temporal, devido à alta taxa de falhas no Trace Espacial, o que requer mais acessos à memória principal.

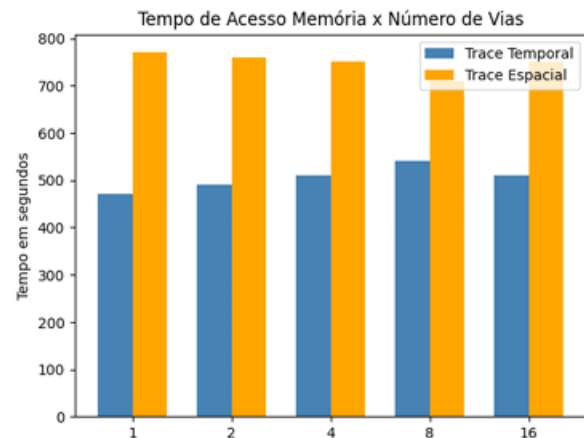


Figura 4: Cenário 2 - Tempo de Acesso à Memória x Número de Vias

No *Cenário 3*, a introdução de um segundo nível de cache (L2), embora seja mais lenta que a cache L1, oferece vantagens ao reduzir o tempo de busca de dados, já que a busca no L2 é mais rápida do que na memória principal. Na *Figura 5*, nota-se que a taxa de falhas no Trace Espacial foi muito mais alta em comparação ao Trace Temporal, como mostrado na *Figura 6*, uma vez que o L2 prioriza dados que podem ser reutilizados (Localidade Temporal).

Além disso, na *Figura 5*, observa-se uma situação favorável ao Trace Espacial quando a cache L1 possui 8 bytes e a L2 possui 32 bytes. Nessa configuração, a cache L2 atua quase como uma cópia da memória principal, reduzindo as falhas. Ao aumentar o tamanho da cache L1 para 16 bytes, a taxa de falhas na L1 diminui, pois há menos substituições de blocos. A aparente alta taxa de falhas na L2 (de 0.31 para 0.51) ocorre porque mais instruções são encontradas na L1, reduzindo a necessidade de acessar a L2.

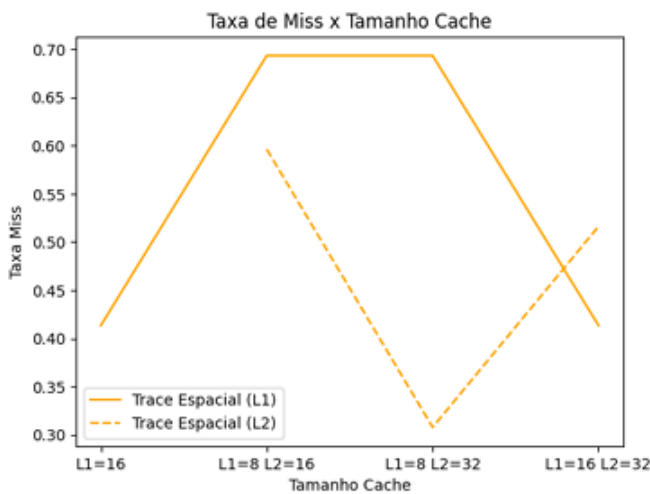


Figura 5: Cenário 3 - Taxa de Falhas no Trace Espacial x Tamanho da Cache

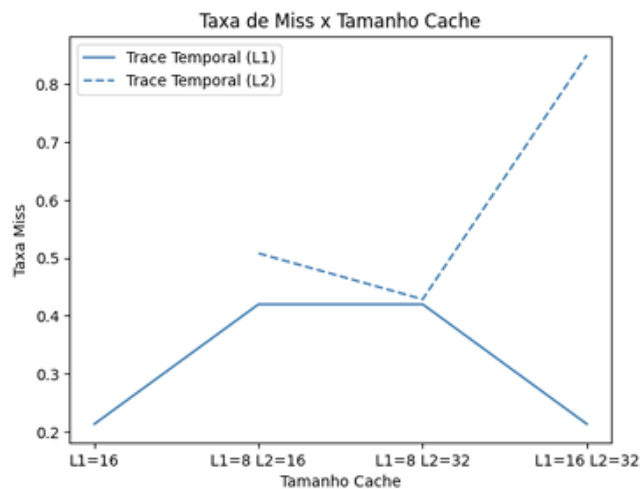


Figura 6: Cenário 3 - Taxa de Falhas no Trace Temporal x Tamanho da Cache

Finalmente, a *Figura 7* mostra que o Trace Temporal teve um tempo de acesso à memória muito maior do que o Trace Espacial. O pior cenário para ambos os traces foi quando a cache L1 tinha apenas 8 bytes e a L2, 16 bytes, resultando em uma alta taxa de falhas tanto na L1 quanto na L2. O tempo de acesso foi, então, uma soma dos tempos das duas caches mais o tempo de acesso à memória principal, elevando o tempo total.

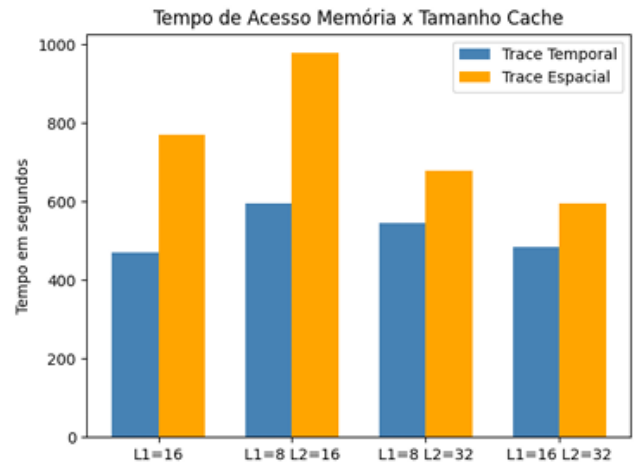


Figura 7: Cenário 3 - Tempo de Acesso à Memória x Tamanho da Cache

### B. Memória Virtual

Agora, para a Memória Virtual, no *Cenário 1*, o aumento do número de blocos por página na memória virtual provocou um aumento na taxa de falhas na TLB (*Translation Lookaside Buffer*) e no tempo de acesso à memória no Trace Temporal, conforme mostrado nas *Figuras 8 e 9*. Em contrapartida, para o Trace Espacial, a taxa de falhas na TLB diminuiu ao aumentar o tamanho do bloco de 2 para 4, reduzindo também o tempo de acesso à memória. Contudo, ao continuar aumentando o número de blocos por página, a taxa de falhas voltou a crescer, prejudicando o desempenho.

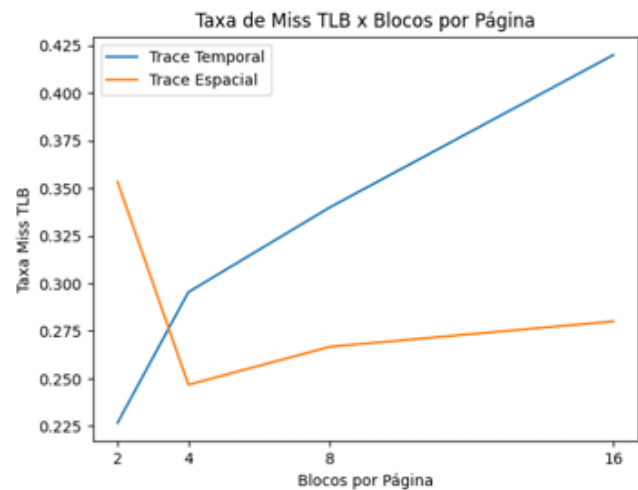


Figura 8: Cenário 1 - Taxa de Falhas na TLB x Blocos por Página

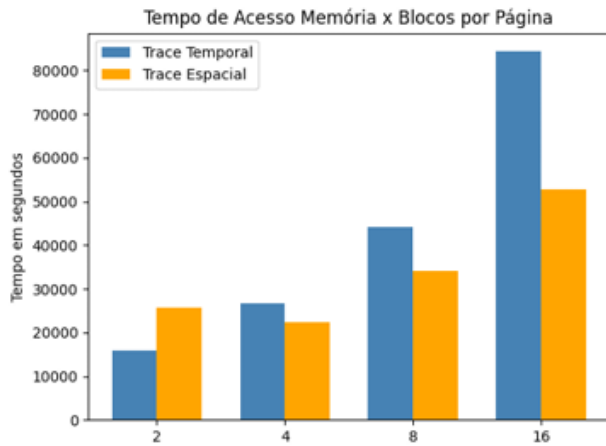


Figura 9: Cenário 1 - Tempo de Acesso à Memória x Blocos por Página

No *Cenário 2*, ao aumentar o tamanho da TLB, observou-se uma redução considerável na taxa de falhas, especialmente ao passar de 4 para 8 blocos por página, como mostrado nas *Figuras 10 e 11*. No entanto, após atingir esse tamanho, a taxa de falhas estabilizou, indicando que um aumento adicional não resultaria em melhorias significativas.

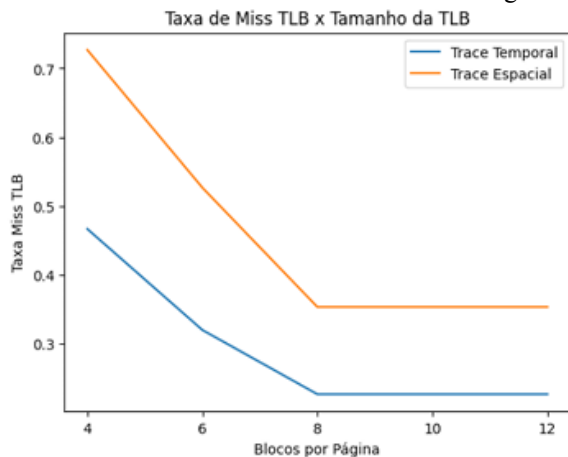


Figura 10: Cenário 2 - Taxa de Falhas na TLB x Blocos por Página

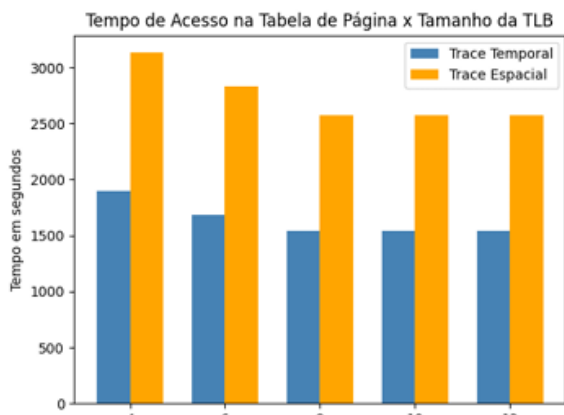


Figura 11: Cenário 2 - Tempo de Acesso à Memória x Blocos por Página

Por fim, no *Cenário 3*, a ausência da TLB resultou em um aumento significativo no número de acessos à memória principal e à tabela de páginas, conforme mostrado na *Figura 12*, evidenciando a importância da TLB para otimizar o processo de tradução de endereços.

	Sem TLB		Com TLB	
	Trace Espacial	Trace Temporal	Trace Espacial	Trace Temporal
Disco	102	60	102	60
Memória Principal	810	600	712	484
Tabela de Página	150	150	53	34
TLB	-	-	150	150
Número de Acessos				

Figura 12: Cenário 3 - Número de Acessos x Presença da TLB

## VI. CONCLUSÕES

Este estudo mostrou a importância de uma correta configuração da hierarquia de memória para explorar os conceitos de localidade temporal e espacial. As variações aplicadas na Cache e na Memória Virtual demonstraram que o tamanho dos blocos e a capacidade da TLB têm um impacto direto nas taxas de hit e miss, afetando o desempenho do sistema.

Os resultados indicam que blocos maiores são benéficos para aplicações que utilizam localidade espacial, mas podem prejudicar o desempenho em cenários com predominância de localidade temporal. Da mesma forma, uma TLB maior reduz a quantidade de page faults, mas sua eficiência se estabiliza após certo ponto, não justificando um aumento indefinido de tamanho.

## REFERÊNCIAS

- [1] Tao, J., Schloissnig, S., Karl, W. (2006). Analysis of the Spatial and Temporal Locality in Data Accesses. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds) Computational Science-ICCS2006. ICCS 2006.
- [2] Gonzalez, M., Souza, L. (2021). "Simulação de memória cache em sistemas embarcados com o DinerioIV". Embedded Systems Review, 12(2), 98-107.
- [3] Silva, J. et al. (2019). "Avaliação de políticas de substituição de cache utilizando o simulador GEM5". Journal of Computer Systems, 45(3), 234-246.
- [4] Amnesia. "Manual do Simulador Amnesia". Disponível em: <http://amnesia.lasdp.icmc.usp.br>.
- [5] L. V. Cargnini, L. Torres, R. M. Brum, S. Senni and G. Sassatelli, "Embedded memory hierarchy exploration based on magnetic RAM," 2013 IEEE Faible Tension Faible Consommation, Paris, France, 2013, pp. 1

