

ECS e EMCS: Simuladores de Caches para o Apoio Pedagógico no Ensino de Arquitetura de Computadores

Dawson Paiva Lima¹, Nahri Moreano¹

¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)

dawsonpaiva@gmail.com, nahri@facom.ufms.br

Resumo. *Simuladores educacionais são ferramentas fundamentais no ensino das disciplinas da área de Arquitetura e Organização de Computadores, pois facilitam a compreensão do comportamento do sistema modelado e sua avaliação, aprofundando o aprendizado dos conceitos teóricos estudados. Este trabalho apresenta dois simuladores de memórias cache para apoio pedagógico, o Educational Cache Simulator e o Educational Multi-Level Caches Simulator, que modelam um único nível ou dois níveis de memória cache, respectivamente. Ambos possuem uma interface gráfica intuitiva e interativa, possibilitam a escolha de diferentes configurações para as caches e o acompanhamento do comportamento da cache a cada acesso e produzem resultados de desempenho. O primeiro também oferece uma representação gráfica da organização da cache e a visualização dos acessos à mesma. Os simuladores podem ser usados na realização de experimentos em aulas práticas e no estudo extra-classe, promovendo o aprendizado ativo e a integração teoria-prática.*

Abstract. *Educational simulators are fundamental tools for teaching courses in the Computer Architecture and Organization field, as they facilitate understanding the behavior of the modeled system and its evaluation, deepening the learning of the theoretical concepts studied. This work presents two cache memory simulators for educational support, the Educational Cache Simulator and the Educational Multi-Level Caches Simulator, which model a single level or two levels of cache memory, respectively. Both have an intuitive and interactive graphical interface, allow the choice of different cache configurations and monitoring the cache behavior at each access and produce performance results. The first also offers a graphical representation of the cache organization and the visualization of accesses to it. The simulators can be used to carry out experiments in laboratory classes and for extra-class study, promoting active learning and theory-practice integration.*

1. Introdução

As disciplinas da área de Arquitetura e Organização de Computadores (AOC) são essenciais para a formação de estudantes de computação. Tais estudantes não devem enxergar o computador como uma caixa preta capaz de executar programas, e sim compreender o ambiente de hardware no qual a computação é baseada e a interface que ele fornece para as camadas superiores de software. Para isso, precisam conhecer os componentes que compõem um sistema computacional, suas funções, desempenho e as interações entre eles. Além disso, esse conhecimento é fundamental para

desenvolverem programas que usem eficientemente os recursos do hardware e assim alcancem um bom desempenho. Portanto, os currículos de referência dos cursos de graduação em Computação da ACM/IEEE [ACM/IEEE Computer Society 2013, ACM/IEEE Computer Society 2016] e da SBC [Zorzo et al. 2017] incluem AOC como uma de suas importantes áreas de conhecimento.

As disciplinas de AOC apresentam conteúdos extensos e complexos, em função do nível de detalhamento e abrangência dos assuntos abordados. A complexidade das arquiteturas atuais e dos conceitos envolvidos traz desafios ao aprendizado. Por isso, os conteúdos são abordados através de diferentes níveis de abstração que são revelados gradualmente. Um desafio no ensino de AOC é como promover a conexão entre o conhecimento teórico e a experiência prática por parte dos alunos. Por exemplo, ensinar o funcionamento de um processador, um pipeline ou uma memória cache de forma apenas abstrata e estática dificulta a compreensão do comportamento dinâmico desses componentes pelos alunos. A complexidade dos conceitos envolvidos demanda diferentes metodologias e ferramentas computacionais que facilitem o processo de ensino-aprendizagem e promovam a integração teoria-prática. Atividades em laboratório podem ser realizadas com esse objetivo, provendo ambientes de aprendizado que ofereçam aos alunos, de forma intuitiva e interessante, experiência prática e melhor compreensão dos conceitos teóricos estudados.

Simuladores são ferramentas fundamentais no desenvolvimento das habilidades práticas no ensino de AOC, além de terem um papel importante para a pesquisa em AOC. O objetivo da simulação é modelar um sistema computacional completo ou uma parte dele (por exemplo, o processador ou a memória), reproduzindo o seu comportamento, e estimar o seu desempenho e/ou consumo de energia [Akram and Sawalha 2019]. Dessa forma, simuladores educacionais fornecem modelos simplificados de computadores reais, permitindo que o aluno tenha acesso ao estado interno do sistema simulado e observe os eventos que ocorrem e a interação entre os componentes durante a execução de um programa. Assim, a realização de experimentos usando simuladores facilita a compreensão do comportamento do sistema modelado e sua avaliação, complementando e aprofundando o aprendizado dos conceitos teóricos estudados.

A hierarquia de memórias e em especial as memórias cache têm um papel importante no desempenho de um computador. Embora o uso da hierarquia de memórias nos processadores *multicore* atuais seja transparente para o programador, ele precisa entender o seu impacto no desempenho do sistema, para desenvolver programas que façam um uso adequado da hierarquia e assim obtenham um bom desempenho. Para isso, é necessária a compreensão do funcionamento das memórias cache e da influência da sua organização no desempenho. A utilização de um simulador que permita a visualização dos eventos que ocorrem em um acesso à memória cache durante a execução de um programa auxilia o processo de ensino/aprendizagem desses conceitos.

Este trabalho apresenta dois simuladores de memórias cache, o Educational Cache Simulator (ECS) e o Educational Multi-Level Caches Simulator (EMCS), desenvolvidos com o objetivo de prover apoio pedagógico para as disciplinas de AOC. O primeiro simulador modela um único nível de memória cache, enquanto o segundo representa uma hierarquia com dois níveis de memórias cache (caches L1 e L2). Ambos apresentam uma interface gráfica intuitiva e interativa, possibilitam a escolha de diferentes configurações

para a(s) cache(s) e o acompanhamento do comportamento da cache a cada acesso e produzem resultados de desempenho. O ECS também oferece uma representação gráfica da organização da cache e permite a visualização dos acessos à mesma.

2. Trabalhos Relacionados

Simuladores são fundamentais na pesquisa em AOC para implementar e avaliar novas ideias e auxiliar a exploração do espaço de projeto. Também podem permitir uma compreensão aprofundada do comportamento dos programas e a identificação de gargalos no sistema simulado. As ferramentas diferem bastante em escopo, nível de abstração da simulação e interface com o usuário, entre outros aspectos. Algumas ferramentas simulam um sistema completo, com processador(es), hierarquia de memórias e interconexões, enquanto outras simulam apenas um aspecto do sistema, como as caches por exemplo. Diversos simuladores de caches foram propostos na literatura. Uma compilação e comparação detalhada desses simuladores é apresentada em [Brais et al. 2020]. A Tabela 1 resume as funcionalidades de alguns simuladores e compara-os.

Tabela 1. Comparação entre alguns simuladores de caches

	gem5	MacSim	pycachesim	Amnesia	EDUCache
Objetivo principal	Pesquisa e avaliação de desempenho	Pesquisa e avaliação de desempenho	Educacional	Educacional	Educacional
Aspecto da arquitetura simulado	Processadores, sistema de memórias e interconexões	Processadores e sistema de memórias	Hierarquia de caches	Hierarquia de memórias	Hierarquia de caches
Caches simuladas	L1/L2/L3, instruções/dados, unificada/separada, inclusiva/exclusiva, privada/compartilhada	L1/L2/L3, instruções/dados	L1/L2/L3	L1/L2/L3, instruções/dados, unificada/separada	L1/L2/L3, inclusiva/exclusiva, privada/compartilhada
Origem dos acessos à memória	Programa simulado ou <i>trace</i> de instruções	<i>Trace</i> de instruções	<i>Trace</i> de endereços de memória	<i>Trace</i> de endereços de memória	<i>Trace</i> de endereços de memória
Resultados produzidos	Diversas métricas de desempenho e consumo de energia	Métricas de desempenho e consumo de energia	Métricas de acertos e falhas nas caches	Métricas de desempenho dos acessos aos níveis da hierarquia de memórias	Métricas de acertos e falhas nas caches
Interface	Linha de comando	Linha de comando	Linha de comando	Interface gráfica simples	Interface gráfica simples
Plataforma de execução	Linux	Linux	Multi-plataforma	Multi-plataforma	Multi-plataforma
Disponibilidade	Código aberto	Código aberto	Código aberto	Código aberto	Não disponível

Os simuladores gem5 [Lowe-Power et al. 2020] e MacSim [HPArch Research Group 2006] são ferramentas versáteis para a realização de pesquisas e experimentos de avaliação de desempenho e/ou consumo de energia. Todavia, sua utilização como ferramenta de ensino é bastante complexa. Os simuladores pycachesim [Hammer], Amnesia [Tiosso et al. 2014] e EDUCache [Ristov et al. 2013] são ferramentas educacionais interessantes. Almeja-se entretanto nesse trabalho, simuladores didáticos para o ensino dos conceitos fundamentais da organização e do funcionamento das caches, que apresentem interface gráfica visual da organização das caches e permitam tanto a execução de um programa quanto de um *trace* de endereços de memória, entre outros aspectos.

MARS (MIPS Assembler and Runtime Simulator) [Vollmar and Sanderson 2006] é uma ferramenta de uso educacional que permite o desenvolvimento e execução de programas em linguagem de montagem MIPS. Seu simulador funcional implementa o

conjunto de instruções de máquina MIPS, além de pseudo-instruções e chamadas ao sistema. Possui código aberto, desenvolvido em Java, sendo portanto multi-plataforma. Apresenta uma interface gráfica onde é possível executar o programa inteiro ou instrução por instrução, visualizando os valores dos registradores e das posições de memória. É muito utilizada no ensino introdutório de AOC por adotar o conjunto de instruções estudado em livros didáticos amplamente adotados [Patterson and Hennessy 2014].

O MARS disponibiliza várias ferramentas que podem ser utilizadas de forma integrada durante a simulação realizada. Uma dessas ferramentas é o Data Cache Simulator, que permite simular uma cache de dados. Os endereços de memória acessados são provenientes das instruções *load* e *store* executadas pelo programa MIPS simulado no MARS. A ferramenta permite a configuração do tamanho da cache, associatividade, tamanho do bloco e política de substituição e produz métricas de desempenho relativas apenas a acertos e falhas na cache. A interface gráfica é simples e não apresenta detalhes da organização da cache. Novas ferramentas podem ser desenvolvidas e integradas ao MARS para utilização durante a simulação do programa MIPS.

3. Simuladores ECS e EMCS

As ferramentas ECS e EMCS têm por objetivo prover apoio pedagógico no processo de ensino-aprendizagem de AOC. Permitem a simulação de memórias cache, visando a compreensão aprofundada da organização, do funcionamento das mesmas e do impacto da hierarquia de memórias no desempenho do programa. No ECS, a hierarquia modelada consiste de um nível de cache e da memória principal, enquanto que no EMCS essa hierarquia consiste de dois níveis de cache (L1 e L2) e da memória principal. Ambos foram projetados seguindo de maneira bem próxima os conteúdos apresentados nos livros didáticos de referência da área, para facilitar a integração, por parte do aluno, entre os conceitos teóricos estudados e a prática no uso dos simuladores.

Os simuladores foram desenvolvidos como ferramentas acopladas ao MARS e também podem ser executados de forma *stand-alone* independente do MARS. Os acessos de memória (às instruções e/ou dados) simulados são provenientes do programa MIPS executado (como um simulador *emulation-driven*) ou de um *trace* de endereços de memória fornecido (como um simulador *memory trace-driven*). As ferramentas possuem uma interface gráfica intuitiva e interativa, o que contribui para a facilidade de sua utilização. Foram desenvolvidas em Java, sendo portanto ferramentas multi-plataforma e favorecendo sua portabilidade.

3.1. Educational Cache Simulator (ECS)

A ferramenta ECS, desenvolvida inicialmente em [Silva 2019] e estendida no presente trabalho, tem como focos principais: a compreensão da organização de uma cache, incluindo seus blocos de armazenamento e circuitos de acesso; a interpretação dos bits do endereço de memória, de acordo com a configuração da cache, e como ele é usado no acesso à mesma; o entendimento do funcionamento da cache e de como um acesso é realizado na mesma; e a análise do impacto da configuração da cache no desempenho de um programa. Para isso, apresenta uma representação gráfica visual da organização da cache, permite o acompanhamento e visualização de cada acesso realizado durante a execução do programa, mostrando o endereço acessado, o conteúdo da cache, o conjunto acessado, se ocorreu falha ou acerto e a via onde ocorreu o acerto, e apresenta resultados

de desempenho. A ferramenta é organizada em quatro abas, configuração, organização da cache, execução dinâmica e acessos e desempenho, nas quais as suas funcionalidades estão distribuídas.

Configuração Na aba de Configuração (*Settings*), mostrada na Figura 1, é possível selecionar a organização da cache, escolhendo:

- o tipo de armazenamento da cache: apenas dados, apenas instruções ou unificada;
- a configuração da cache, indicando número total de blocos (1, 2, 4, ... ou 1024), associatividade (1, 2, 4, 8 ou 16) e tamanho do bloco (1, 2, 4, ... ou 128 palavras);
- a política de substituição de blocos (LRU, FIFO ou Random);
- os tempos de acesso à cache e à memória principal, para obtenção de resultados de desempenho.

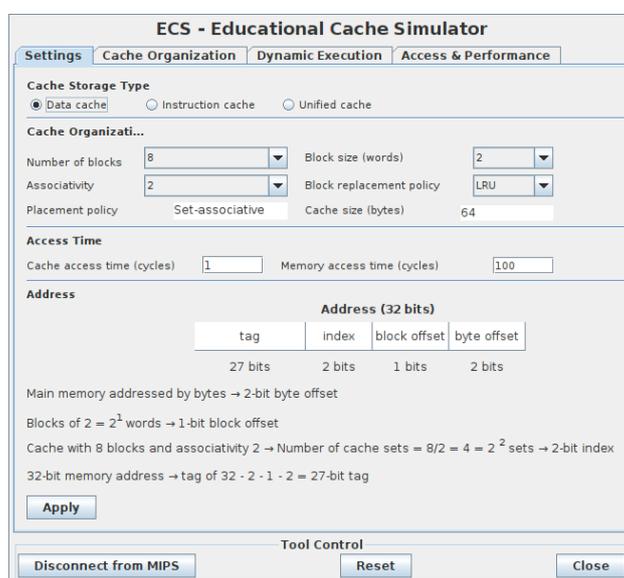


Figura 1. Aba de Configuração no ECS

Os possíveis valores da configuração da cache permitem modelar uma cache mapeada diretamente, totalmente associativa ou associativa por conjunto. As caches modeladas podem ser pequenas e simples, adequadas para exemplos didáticos, até caches maiores e realistas, que possibilitem a realização de experimentos de avaliação de desempenho. Com base na configuração escolhida, a ferramenta informa se a cache modelada é mapeada diretamente, totalmente associativa ou associativa por conjunto. Se for uma cache mapeada diretamente, a escolha da política de substituição é desabilitada. A capacidade total (em bytes) da cache configurada é calculada e apresentada e pode ser usada para estimar o custo do hardware da cache e comparar diferentes configurações. Em relação aos tempos de acesso, valores *default* são fornecidos e podem ser modificados para valores diferentes.

A interpretação dos bits do endereço de memória, de acordo com a configuração escolhida para a cache, também é apresentada nessa aba. A ideia é facilitar a compreensão da relação entre o número total de blocos da cache, a associatividade e o tamanho do bloco e os campos *tag*, *index*, *block offset* e *byte offset* (e sua largura em bits) em que o endereço é dividido.

Organização da Cache A aba de Organização da Cache (*Cache Organization*), mostrada na Figura 2, apresenta uma representação gráfica visual da organização da cache, na forma de um diagrama de blocos detalhado. O diagrama contém os blocos de armazenamento da cache (organizados em conjuntos e vias com bits de controle, *tag* e palavras de cada bloco) e também os circuitos de acesso à mesma (decodificador, comparadores e multiplexadores). Também mostra como os campos em que o endereço de memória acessado é dividido são utilizados no acesso à cache. É possível exportar esse diagrama da cache para um arquivo de imagem. Barras de rolagem horizontal e vertical permitem que mesmo caches não muito pequenas sejam visualizadas. Para caches demasiado grandes, essa visualização é desabilitada, entretanto a simulação continua sendo realizada normalmente.

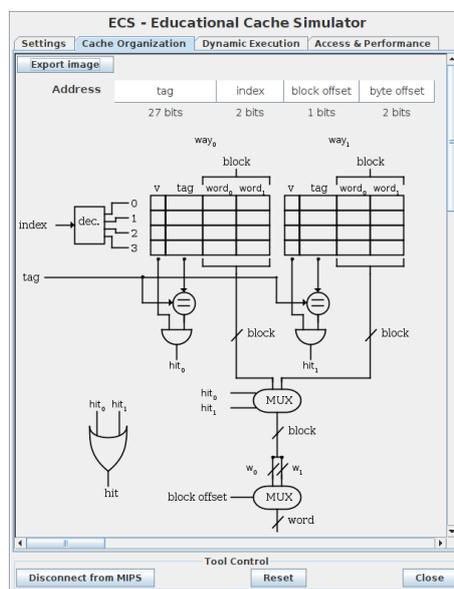


Figura 2. Aba de Organização da Cache no ECS

Execução Dinâmica A aba de Execução Dinâmica (*Dynamic Execution*), mostrada na Figura 3, permite acompanhar os acessos realizados à cache durante a execução de um programa. Usando botão *Connect to MIPS*, a execução do programa MIPS simulado no MARS é associada ao ECS. Esse programa pode ser executado até o fim ou até um *breakpoint* ou instrução por instrução (usando os botões *run* ou *step* do MARS).

Os acessos à memória (a instruções e/ou dados, dependendo da configuração escolhida para a cache) realizados pelo programa MIPS geram acessos à cache (como um simulador *emulation-driven*), e são apresentados no diagrama da organização da cache nessa aba. São mostrados os valores dos campos do endereço acessado e como eles são utilizados no acesso à cache. Também são apresentados os conteúdos das posições da cache (bit de validade, *tag* e palavras do bloco), que são atualizados a cada acesso. Cores são usadas para indicar o conjunto acessado, se ocorreu falha ou acerto e, no caso de acerto, em qual via ocorreu o acerto. Dessa forma, o aluno visualiza graficamente o acesso realizado à cache e pode compreender detalhadamente o funcionamento da mesma.

Acessos e Desempenho A aba de Acessos e Desempenho (*Access & Performance*), mostrada na Figura 4, complementa a aba de Execução Dinâmica, na visualização e compreensão dos acessos realizados à cache. Um *log* dos acessos realizados é

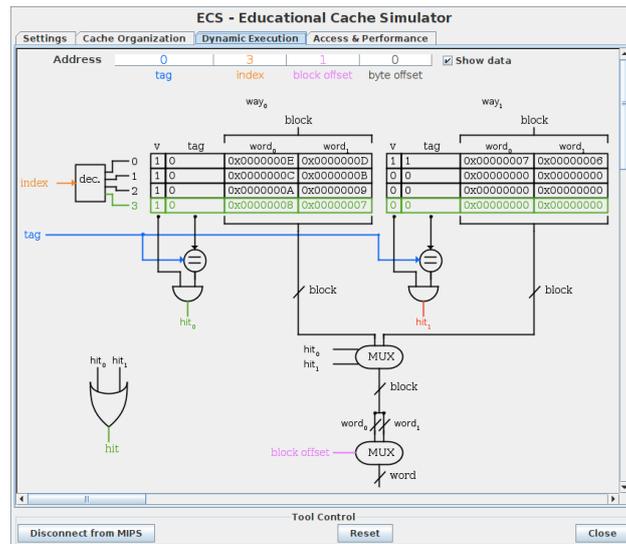


Figura 3. Aba de Execução Dinâmica no ECS

apresentado, na forma de uma tabela, com as seguintes informações de cada acesso: o endereço acessado, o número do bloco de memória acessado, o *block offset*, a *tag* e o índice correspondentes, se ocorreu acerto ou falha e o número do bloco substituído (caso uma substituição tenha sido necessária). Uma barra de rolagem vertical permite visualizar toda a tabela, mesmo para execuções não muito curtas. É possível exportar essa tabela para uma planilha em um arquivo CSV. Durante a execução do programa MIPS (direta até o fim ou instrução por instrução), é possível mudar entre as abas do ECS, intercalando a visualização dos acessos realizados na organização da cache e na tabela de acessos.

Cache Performance

Cache access count	1712	Cache replacement count	176
Cache hit count	1528	Misses per instruction	0,03
Cache miss count	184	Average memory access time (l...	11,75
Cache miss rate	10,7%		

Cache Access

Address	Block offset	Block number	Index	Tag	Hit/Miss	Block replaced
0	0	0	0	0	Miss	
4	1	0	0	0	Hit	
0	0	0	0	0	Hit	
4	1	0	0	0	Hit	
4	1	0	0	0	Hit	
8	0	1	1	0	Miss	
4	1	0	0	0	Hit	
8	0	1	1	0	Hit	

Figura 4. Aba de Acessos e Desempenho no ECS

Essa aba também apresenta vários resultados de desempenho sobre a simulação realizada, de forma dinâmica, durante e ao final da execução do programa MIPS. As métricas de desempenho fornecidas são: número de acessos à cache, número de acertos e falhas, taxa de falhas, número de substituições realizadas, falhas por instrução e tempo médio de acesso à memória. Dessa forma, o aluno pode analisar o impacto da configuração escolhida para a cache no desempenho do programa.

Ao invés de simular acessos à memória oriundos de um programa MIPS simulado, é possível simular acessos obtidos a partir de um arquivo de *trace* de endereços de

memória, usando para isso os botões *Load trace* e *Run trace* presentes nessa aba. A execução do ECS na forma *memory trace-driven* permite sua utilização com *traces* longos e gerados com outras ferramentas, não ficando restrito à execução de programas MIPS. Nesse caso, devido ao tamanho dos *traces*, a simulação é realizada toda de uma só vez (e não passo a passo) e a tabela de acessos não é preenchida. O formato do *trace* é um arquivo texto, com um acesso por linha, contendo o tipo da operação (0, 1 ou 2, para leitura de instrução, leitura de dado e escrita de dado, respectivamente) e o endereço acessado.

3.2. Educational Multi-Level Caches Simulator (EMCS)

A ferramenta EMCS complementa o ECS e tem como focos principais: a compreensão do funcionamento de caches em dois níveis; o entendimento de quando um acesso é realizado em cada nível de cache; e a análise do impacto de um segundo nível de cache no desempenho de um programa. Para isso, permite o acompanhamento de cada acesso realizado durante a execução do programa, mostrando o endereço acessado, informações sobre o acessos nas caches L1 e L2 (se for o caso) e se ocorreu falha ou acerto, e apresenta resultados de desempenho dos dois níveis. O EMCS possui interface similar ao ECS para facilitar sua utilização pelo aluno já habituado ao ECS. A ferramenta é organizada em duas abas, configuração e acessos e desempenho, nas quais as suas funcionalidades estão distribuídas.

Configuração A aba de Configuração (*Settings*) é similar à do ECS, porém permitindo a configuração de caches L1 e L2 separadamente. Novamente, cada cache modelada pode ser mapeada diretamente, totalmente associativa ou associativa por conjunto, o que é informado pela ferramenta. As caches podem ser pequenas ou maiores e a ferramenta valida se a cache L2 não é menor que a cache L1, em relação ao total de blocos e tamanho do bloco. Para uma cache mapeada diretamente, a escolha da política de substituição é desabilitada. A capacidade total (em bytes) de cada cache configurada é apresentada. Em relação aos tempos de acesso, valores *default* são fornecidos e podem ser modificados para valores diferentes.

Acessos e Desempenho A aba de Acessos e Desempenho (*Access & Performance*), mostrada na Figura 5 possibilita a visualização e compreensão dos acessos realizados às caches L1 e L2. Um *log* dos acessos realizados é apresentado, na forma de uma tabela, com as seguintes informações de cada acesso: o endereço acessado e, para a cache L1, o número do bloco de memória acessado, o *block offset*, a *tag* e o índice correspondentes, se ocorreu acerto ou falha e o número do bloco substituído (caso uma substituição tenha sido necessária). Se ocorreu falha na cache L1, as mesmas informações, porém relativas à cache L2 são também apresentadas. Novamente, essa tabela pode ser exportada para uma planilha em um arquivo CSV.

Essa aba também apresenta os resultados de desempenho sobre a simulação realizada, agora considerando os dois níveis de cache. As métricas de desempenho fornecidas são, para cada cache: número de acessos à cache, número de acertos e falhas, taxa de falhas, número de substituições realizadas e falhas por instrução. Para a cache L2, a taxa de falhas é apresentada como global e local, isto é, considerando todos os acessos à hierarquia de memórias ou apenas os acessos realizados à L2, respectivamente. O tempo médio de acesso à memória também é fornecido e permite que o aluno observe como a cache L2 amortiza o tratamento das falhas da cache L1. Da mesma forma que no ECS, é

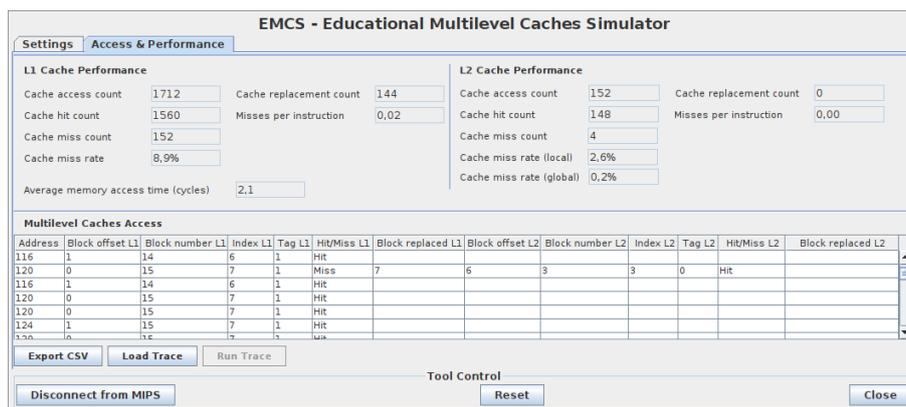


Figura 5. Aba de acessos e desempenho no EMCS

possível simular acessos obtidos a partir de um arquivo de *trace* de endereços de memória. O formato do *trace* é o mesmo usado para o ECS.

4. Implementação

Os simuladores ECS e EMCS foram desenvolvidos como ferramentas acopladas ao MARS, usando a linguagem de programação Java, e suas interface gráfica foram implementadas usando a API Java Swing. Como consequência, os simuladores podem ser executados em diferentes plataformas (Windows, Linux e MacOS), o que é fundamental para uma ferramenta educacional. Os simuladores ECS e EMCS são ferramentas independentes, isto é, são utilizados separadamente um do outro.

O MARS oferece suporte para a criação, externamente, de novas ferramentas e o seu acoplamento ao MARS, de forma que as ferramentas possam interagir com o programa MIPS sendo executado e com os recursos do processador MIPS simulado. Para isso, é necessário implementar uma nova classe estendendo a classe abstrata `mars.tools.AbstractMarsToolAndApplication`, que é parte do pacote `mars.tools` disponibilizado pelo MARS. A nova classe será a classe principal da nova ferramenta e deve fazer parte do mesmo pacote e conter métodos sobrepondo alguns métodos da classe abstrata. Ao executar novamente o MARS, após compilar o projeto completo contendo o MARS e a nova ferramenta, a ferramenta desenvolvida é automaticamente adicionada ao menu *Tools* do MARS e é executada quando selecionada.

Também é possível executar os simuladores ECS e EMCS como programas *stand-alone*, que usam o montador e simulador do MARS em *background*, porém sem necessidade de carregar toda a interface do mesmo. Para isso, um método `main()` deve ser implementado dentro da nova classe.

5. Conclusão

Esse trabalho apresentou o desenvolvimento das ferramentas Educational Cache Simulator e Educational Multi-Level Caches Simulator. Elas têm o objetivo de prover apoio pedagógico, facilitar o processo de ensino-aprendizagem e promover a integração teoria-prática dos conteúdos relacionados a hierarquia de memórias e memórias caches nas disciplinas da área de Arquitetura e Organização de Computadores. Os simuladores podem ser utilizados pelos alunos em aulas práticas em laboratório ou no estudo extra-classe, através da realização de diversos experimentos com diferentes níveis de complexidade. O uso dos simuladores propicia aos alunos o aprendizado ativo da

organização e funcionamento das memórias caches e de como desenvolver programas que usem eficientemente a hierarquia de memórias e assim alcancem um bom desempenho.

Os simuladores oferecem a configuração de vários parâmetros das caches, possibilitando que os alunos simulem diferentes organizações de caches, comparem os resultados e identifiquem a influência dos parâmetros no desempenho. A visualização gráfica da organização da cache e dos acessos realizados e a execução passo a passo oferecidos pelos simuladores permitem que o aluno experiencie o funcionamento das caches. Os modos de execução *emulation-driven* e *memory trace-driven* suportados pelas ferramentas possibilitam a simulação dos acessos à memória gerados por programas simples em linguagem de montagem ou até programas complexos executados em outros ambientes.

Referências

- ACM/IEEE Computer Society (2013). Computer science curricula 2013 – Curriculum guidelines for undergraduate degree programs in computer science. ACM/IEEE.
- ACM/IEEE Computer Society (2016). Computer engineering curricula 2016 – Curriculum guidelines for undergraduate degree programs in computer engineering. ACM/IEEE.
- Akram, A. and Sawalha, L. (2019). A survey of computer architecture simulation techniques and tools. *IEEE Access*, 7:78120–78145.
- Brais, H., Kalayappan, R., and Panda, P. R. (2020). A survey of cache simulators. *ACM Computing Surveys*, 53(1):19:1–19:32.
- Hammer, J. The pycachesim simulator. <https://pypi.org/project/pycachesim/>. Acessado em outubro/2020.
- HPArch Research Group (2006). *MacSim: A CPU-GPU Heterogeneous Simulation Framework – User Guide*. Georgia Institute of Technology.
- Lowe-Power, J. et al. (2020). The gem5 simulator: Version 20.0+ – a new era for the open-source computer architecture simulator. <https://arxiv.org/pdf/2007.03152.pdf>. Acessado em janeiro/2021.
- Patterson, D. A. and Hennessy, J. L. (2014). *Computer Organization and Design – The Hardware/Software Interface*. Elsevier, 5th edition.
- Ristov, S. et al. (2013). Using EDUCache simulator for the computer architecture and organization course. *International Journal of Engineering Pedagogy*, 3(3):47–56.
- Silva, L. H. C. (2019). Educational cache simulator: Uma ferramenta para ensino de arquitetura de computadores. Faculdade de Computação, Universidade Federal de Mato Grosso do Sul.
- Tiosso, F. et al. (2014). Amnesia: um objeto de aprendizagem para o ensino de hierarquia de memória. In *Congresso Brasileiro de Informática na Educação*, pages 80–89.
- Vollmar, K. and Sanderson, P. (2006). MARS: an education-oriented MIPS assembly language simulator. *ACM SIGCSE Bulletin*, 38(1):239–243.
- Zorzo, A. F. et al. (2017). Referenciais de formação para os cursos de graduação em computação. Sociedade Brasileira de Computação.