

Web-MHSim: Simulador didático de hierarquia de memória com carregamento dinâmico de módulos

Luiza Maria Novais Coutinho, José Leandro Dias Mendes,
Carlos Augusto Paiva da Silva Martins
Pontifícia Universidade Católica de Minas Gerais
{joseleandrodm, luizamncoutinho}@yahoo.com.br, capsm@pucminas.br

Resumo

O aprendizado de hierarquia de memória utilizando os métodos convencionais de ensino (slides, livros e outros), apresenta algumas dificuldades na visualização e compreensão do que acontece durante os acessos às posições de memória nos diversos níveis hierárquicos. Com o propósito de reduzir essa dificuldade foi proposto e desenvolvido o Web-MHSim, simulador didático de hierarquia de memória. Com o objetivo de adaptar-se à necessidade do usuário foi adicionado o recurso de carregamento dinâmico de módulos.

1. Introdução

O método tradicional de ensino, no qual professor apresenta os vários conceitos de hierarquia de memória utilizando recursos estáticos como quadro-negro, slides e livros, não é suficiente para que a maioria dos alunos tenha uma compreensão precisa do que está sendo ensinado. O docente encontra uma grande dificuldade para demonstrar claramente os diversos níveis da hierarquia de memória utilizando os métodos convencionais [1][2].

Outro problema encontrado é a inexistência de simuladores que sejam flexíveis e possibilitem a incorporação de novas funcionalidades de acordo com a necessidade do usuário e que possuam características didáticas.

O uso de um simulador didático pode facilitar o aprendizado e compreensão do conteúdo por parte do aluno, e a elaboração de exercícios mais objetivos por parte do professor [3][4]. Um simulador flexível permite que o usuário possa simular e analisar uma estrutura não existente na ferramenta. Com a utilização de um simulador didático e flexível o usuário pode comparar resultados de diferentes configurações e

memory traces e visualizar os diversos níveis da hierarquia.

O Web-MHSim [5] destaca-se por ser um simulador didático de hierarquia de memória (*cache*, principal e virtual), com suporte a carga dinâmica de módulos, que tem por objetivo auxiliar o professor no ensino, o aluno no aprendizado e o pesquisador para comprovação dos resultados de suas pesquisas. O simulador Web-MHSim é um módulo de um ambiente denominado Web-MHE [6] e é uma evolução da ferramenta MSCSim [7][8].

Foram considerados alguns simuladores como referência [9]-[15], para definição das principais características e verificação dos resultados. A ferramenta desenvolvida buscou melhorias e novas funcionalidades em relação a estes.

Em síntese o simulador abrange vários níveis de conhecimento, de estudantes iniciantes a pesquisadores avançados.

O restante do artigo está dividido da seguinte forma. Na segunda seção é apresentado o simulador. Em seguida, na terceira seção são apresentados e discutidos os resultados. Na quarta seção são apresentados os trabalhos correlatos, e por fim na quinta seção são mostradas as conclusões e os trabalhos futuros.

2. Web-MHSim

O Web-MHSim (*Web Memory Hierarchy Simulator*) é um simulador web de hierarquia de memória com suporte a carregamento dinâmico de módulos, em que o principal objetivo é a disponibilização de uma ferramenta flexível que possa adaptar-se a necessidade do usuário, através do carregamento dinâmico de módulos.

Com a finalidade de oferecer ao usuário a possibilidade de escolha do tipo de simulação de acordo com o seu nível de conhecimento, foram

adicionados ao simulador o carregamento dinâmico de módulos, diversos parâmetros e simulações, assim, permitindo uma evolução gradual da complexidade das simulações.

Com o simulador é possível realizar simulações de diversas estruturas carregadas dinamicamente (classes criadas pelo usuário) e simulações com estruturas existentes como *cache* unificada ou separada, múltiplos níveis, memória principal unificada ou separada (Arquitetura de Harvard), memória virtual (TLB - *Translation-look aside buffer*, tabela de páginas e memória virtual). As simulações são compostas por uma ou várias das estruturas citadas.

As simulações podem ser configuradas utilizando os seguintes parâmetros: tamanho da palavra, tamanho do bloco, tamanho, tipo de organização e tempo de acesso de cada estrutura, número de *slots*, políticas de substituição (FIFO – *First in First Out* ou LRU – *Last Recently Used*) e escrita (*Write Back* ou *Write Through*), grau de associatividade, níveis de *cache*, tipo de acesso (seqüencial ou paralelo), dentre outros.

As simulações são realizadas com uma grande riqueza de detalhes, mostrando passo a passo cada endereço que é acessado e suas informações como: número do bloco, *slot* e *tag*, *miss/hit*, tipo de falta, tempo de acesso, página virtual e *frame*, auxiliando o usuário na análise e percepção do desempenho da hierarquia de memória configurada.

Durante toda a simulação o usuário pode visualizar informações estatísticas como: número de blocos distintos, taxa de acerto (porcentagem de *caches hits* em relação ao número de acessos), taxa de faltas compulsórias, conflito e capacidade, taxa de ocupação da *cache*, principal, TLB e tabela de página, números de *miss* com substituições na *cache* e TLB, taxa de acerto da *cache*, TLB e tabela de páginas e tempo médio de acesso à memória. Para auxiliar o usuário na verificação do desempenho da estrutura simulada é possível gerar gráficos com as principais taxas obtidas durante a simulação.

Também estão disponíveis recursos como: guia de cálculos (apresentação de todos os cálculos realizados pelo simulador), animações das principais estruturas da hierarquia, interação do usuário com a simulação (possibilidade do usuário parar uma simulação a qualquer momento e informar os dados referentes ao endereço atual, como: número do *slot*, *miss/hit* e etc.) e gerador de *memory trace*.

A possibilidade de carregar módulos dinamicamente permite ao usuário criar novas estruturas de memória (*cache*, principal e virtual) que possam ser carregadas e incorporadas pelo simulador, e em seguida ser simuladas em conjunto com as estruturas existentes

(Figura 1). Estas novas estruturas podem ser utilizadas para verificar o desempenho, reduzir a penalidade e taxa de erro, auxiliar o aprendizado e outros. Nesta versão só é permitido carregar classes referentes às estruturas de memória. Na definição de uma nova classe, o desenvolvedor deve implementar uma interface padrão disponibilizada pelo simulador, de modo a possibilitar que o Web-MHSim possa incorporá-la e disponibilizá-la para execução.

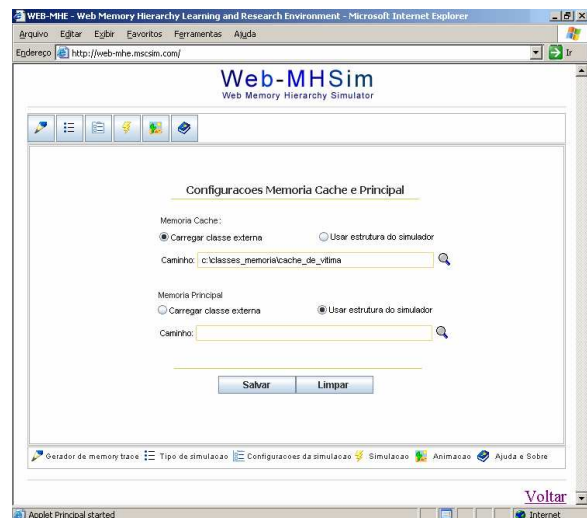


Figura 1. Configuração de uma simulação.

O simulador foi desenvolvido em JAVA utilizando recursos de *Applet*, com o intuito de preservar a portabilidade e a facilidade de disseminação do mesmo.

3. Resultados

O principal recurso do Web-MHSim é o carregamento dinâmico de classes, que o torna bastante flexível permitindo a realização de inúmeras simulações.

Uma das possíveis estruturas que podem ser carregadas pelo simulador é a *cache* de vítimas. O seu principal objetivo é reduzir a penalidade de erro, ou seja, armazenando o que foi descartado, para o caso de ser novamente necessário. A *cache* de vítimas é do tipo completamente associativa e situa-se entre uma *cache* e o seu caminho de reposição. Ela contém apenas blocos que são descartados de uma *cache* devido a uma falha, e são examinados em caso de uma falha para verificar se eles têm os dados desejados, antes de acessar o nível inferior [16].

Esta estrutura foi projetada externamente e implementa a interface padrão do simulador. A classe

foi definida para ser simulada em conjunto com uma *cache* unificada já existente no simulador, ou seja, ao carregar esta classe ela realizará várias chamadas a uma *cache* já existente.

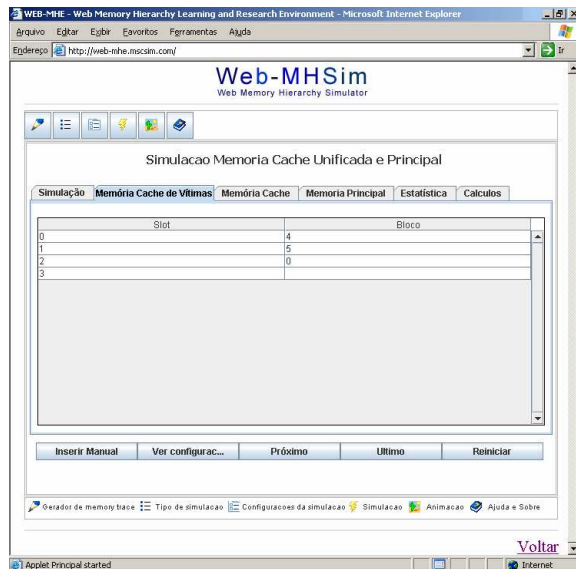


Figura 2. Simulação de *cache* utilizando *cache* de vítimas que foi carregada dinamicamente.

A *cache* de vítimas simulada (Figura 2) é composta por 4 *slots* onde é possível observar uma redução significativa da penalidade de erro por conflito, como é descrito em [17], que descobriu que a utilização de *cache* de vítimas (totalmente associativa) de 4 entradas com uma *cache* do tipo mapeamento direto poderia reduzir a penalidade de erro para 20% a 95% dos erros de conflitos.

Além do exemplo citado, várias outras estruturas podem ser implementadas, como *caches* reconfiguráveis. Com o uso do carregamento dinâmico o usuário tem a vantagem de utilizar um simulador didático que já possui várias outras estruturas, assim reduzindo o tempo necessário para implementação de uma simulação de uma nova estrutura.

Outra característica do Web-MHSim é realizar simulações com as estruturas originais, sem a necessidade de carregar novas classes. Dentre essas destaca-se a simulação de múltiplos níveis com *cache* separada, memória principal separada e virtual.

Por abranger diversos níveis de conhecimento o Web-MHSim foi implantado no laboratório de Arquitetura de Computadores na PUC de Minas Gerais. Com a sua utilização os alunos apresentaram uma maior facilidade na compreensão do conteúdo e na realização de exercícios. Os professores também

observaram um melhor aprendizado e uma melhor avaliação com o uso do simulador.

4. Trabalhos relacionados

Com o objetivo de definir as características fundamentais, novos recursos e realizar uma análise comparativa foi realizada uma pesquisa sobre simuladores com propósitos semelhantes [9]-[15].

Dentre os simuladores encontrados foram considerados o KSH [9] por ser o único simulador com suporte a carregamento dinâmico de classes encontrado no estudo do estado da arte, o CacheSim [12] por ser um simulador *web* de memória *cache* e o MSCSim [7][8] para comprovação da evolução do simulador.

O KSH é um simulador *desktop* de memória *cache* com carregamento dinâmico de módulos. Este permite a simulação de memória *cache* unificada do tipo completamente associativa, mapeamento direto e associativa por conjunto, configuração de políticas de substituição FIFO e LRU, políticas de escrita *Write Back* e *Write Through* e outros. Neste simulador não é possível simular estruturas como *cache* separada, múltiplos níveis, memória virtual, acesso paralelo e não é considerado um simulador didático.

O CacheSim é um simulador *web* de *cache* que permite que os usuários especifiquem uma configuração e escolham um dos *memory traces* existentes, para que, em seguida, possam verificar o desempenho da *cache*. Com o CacheSim não é possível acompanhar a simulação passo a passo, o tempo de acesso, gráficos de desempenho, vários tipos de simulação (apenas a simulação de memória *cache* unificada, de instrução ou dado é permitida) e não possui gerador de *memory trace* e interação com o usuário durante a simulação. Os arquivos de *memory trace* são pré-definidos, não permitindo a configuração pelo usuário.

O MSCSim é um simulador didático para *desktop*, onde é possível simular e acompanhar o comportamento de várias combinações de estruturas, como *cache* unificada ou separada, múltiplos níveis e hierarquia de memória (memórias *cache*, principal e virtual). As simulações oferecem uma grande riqueza de detalhes, mostrados passo a passo a medida que cada endereço é acessado. Dentre suas principais características destacam-se gerador de *memory trace*, estatísticas, animação didática das estruturas da hierarquia e interface orientada ao aprendizado. Nesta versão não existe a possibilidade do usuário entrar com dados durante a simulação, realizar simulação com suporte a memória principal separada, visualizar gráficos estatísticos e outros.

Na tabela 1 é possível visualizar a comparação entre as principais características encontradas nos simuladores relacionados e no Web-MHSim.

Tabela 1. Comparação entre o Web-MHSim e os simuladores relacionados

Ferramentas	Web-MHSim	KSH	CacheSim	MSCSim
Carregamento dinâmico	SIM	SIM	NÃO	NÃO
Simulador Web	SIM	NÃO	SIM	NÃO
Simulação de Cache Separada	SIM	NÃO	SIM	SIM
Simulação de Múltiplos Níveis	SIM	NÃO	NÃO	SIM
Simulação Memória Virtual	SIM	NÃO	NÃO	SIM
Simulação de Memória Principal Separada	SIM	NÃO	NÃO	NÃO
Tempo de acesso	SIM	SIM	NÃO	SIM
Política de escrita	SIM	SIM	SIM	SIM
Gerador de <i>memory trace</i>	SIM	NÃO	NÃO	SIM
Gráficos estatísticos	SIM	NÃO	NÃO	NÃO
Interface orientada ao aprendizado	SIM	NÃO	SIM	SIM
Interação com a simulação	SIM	NÃO	NÃO	NÃO
Animação	SIM	NÃO	NÃO	SIM

Com base na tabela 1 é possível observar que o Web-MHSim possui todas as principais características encontradas nos simuladores relacionados e novos recursos.

5. Conclusão

Com o Web-MHSim os objetivos propostos foram todos alcançados, pois as principais características dos simuladores relacionados foram implementadas e melhoradas. Em conjunto com estas, novos recursos foram adicionados, obtendo um simulador web que tem como fim auxiliar o aprendizado e a pesquisa.

Como uma das principais contribuições deste trabalho destaca-se a possibilidade de carregar estruturas de memória dinamicamente, assim, facilitando o aprendizado das diversas estruturas da hierarquia e auxiliando no desenvolvimento, análise e verificação do desempenho de novas estruturas.

Como trabalho futuro o recurso de carregamento dinâmico será expandido a outros módulos, como: no *memory trace*, interface e animações. Com esta expansão será possível a substituição do *memory trace* por códigos, tornando o simulador *execution-drive*. Outra futura expansão será suportar novos recursos arquiteturais como, microarquitetura, nível ISA e múltiplos formatos de entrada. Será desenvolvido um *framework* para facilitar a implementação de novos recursos por outros usuários e/ou desenvolvedores e carregamento dinâmico de módulos.

Referências

- [1] Yurcik, W.; Wolffe, G. S.; Holliday, Mark A.; Osborne, H. "Teaching Computer Organization/Architecture With Limited", *ACM SIGCSE Bulletin*, Vol. 34, pp. 176-180, 2002.
- [2] Yurcik, W.; Wolffe, G. S.; Holliday, M. A.; "A Survey of Simulators Used in Computer Organization/Architecture Courses", *Summer Computer Simulation Conference (SCSC)*, Society for Computer Simulation (SCS), 2001.
- [3] Grünbacher H.; "Teaching Computer Architecture / Organization using simulators", *28th Frontiers in Education Conference (FIE)*, pp. 1107-1112, 1998.
- [4] Djordjevic, J.; Nikolic, B.; Mitrovic, M. "A Memory System for Education", *The Computer Journal*, Vol. 48, No. 6, pp. 630-641, 2005.
- [5] Mendes, J. L. D.; Coutinho, L. M. N.; Martins, C. A. S.; "Web-MHSim-Web Memory Hierarchy Simulator", 2006, http://www.msccsim.com/WEB-MHE_SIMULADOR.htm.
- [6] Coutinho, L. M. N.; Mendes, J. L. D.; Martins, C. A. P. S.; "Web memory hierarchy learning and research environment", *WorkShop On Computer Architecture Education (WCAE)*, *33th In Symposium On Computer Architecture (ISCA)*, Boston, United State, 2006.
- [7] Coutinho, L. M. N.; Mendes, J. L. D.; Martins, C. A. P. S.; "MSCSim – Simulador de Memória Cache, Split e Múltiplos Níveis", *VI Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD)*, pp. 193-196, 2005.
- [8] Mendes, J. L. D.; Coutinho, L. M. N.; Martins, C. A. P. S.; "MSCSim – Multilevel and Split Cache Simulator", *36th Frontiers in Education Conference (FIE)*, 2006 (aprovado, em processo de publicação).
- [9] Silva, A. R.; Pinho, R. D., "Simulador de memória cache com carregamento dinâmico de módulos e execução de *script* de configuração" IV Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD), 2003.
- [10] Austin, T., "SimpleScalar LLC", Disponível em <http://www.simplescalar.com/>.
- [11] Hill, M. D., "Dinero IV", Disponível em <http://www.cs.wisc.edu/~markhill/DineroIV/>.
- [12] CacheSimulator, 2003, Disponível em <http://www.ece.gatech.edu/research/labs/reveng/cachesim/index.html>.
- [13] Shivakumar, P.; Jouppi, N. J.; "Cacti 3.0", *An Integrated Cache Timing, Power, and Area Model. Technical Report*, Compaq Computer Corporation, 2001.
- [14] Schintke, F.; Simon, J.; Reinefeld, A.; "LDA-Simulator", 2001, Disponível em <http://www.zib.de/schintke/ldasim/index.en.html>.
- [15] Prima Cache Simulator, Disponível em www.dsi.unimo.it/staff/st36/imagelab/prima.html.
- [16] Hennessy, J. L.; Patterson, D. A.; *Computer Architecture: A Quantitative Approach*, third edition, Morgan Kaufman, 2002.
- [17] Jouppi, N. P.; "Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers", *17th Annual Int'l Symposium on Computer Architecture*, pp. 364-373, 1990.