

## Análise e Avaliação de Simuladores de Sistemas Completos para o Ensino de Arquitetura de Computadores

Pedro H. M. M. Penna, Henrique C. Freitas

Grupo de Arquitetura de Computadores e Processamento Paralelo (CARt)

Instituto de Ciências Exatas e Informática (ICEI)

Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

Belo Horizonte, Brasil

pedro.penna@sga.pucminas.br, cota@pucminas.br

**Resumo** — O ensino de arquitetura de computadores em cursos de graduação em computação é fundamental. No entanto, essa tarefa torna-se um grande desafio devido à rápida e contínua evolução da área. Nesse contexto, diferentes metodologias de ensino surgem para facilitar o aprendizado do aluno. Uma delas é o emprego de simuladores de arquitetura como ferramenta de ensino. Este artigo descreve quatro simuladores de sistemas completos populares na academia, com o objetivo de analisar e avaliar suas características para o ensino. Para isso, os simuladores são comparados segundo suas características, e um simulador é escolhido para ser usado em sala de aula e avaliado quantitativamente por meio de um questionário respondido por alunos. Os resultados revelam que alunos aprovam o uso de simuladores como ferramenta de ensino, mas enfrentam dificuldades para entender o simulador utilizado devido à escassez de documentação.

**Palavras-chave:** ensino; arquitetura de computadores; simuladores de sistema completos, ferramentas de aprendizado

### I. INTRODUÇÃO

O ensino de arquitetura de computadores em cursos de graduação na área de computação é fundamental, pois não só capacita o aluno para atuação na área de arquitetura, como também proporciona a ele um melhor esclarecimento dos modelos de programação sequencial e paralelo, conferindo-o pré-requisitos necessários para o desenvolvimento de *software* eficiente. No entanto, ensinar arquitetura de computadores torna-se um grande desafio devido à rápida e contínua evolução da área [1]. Para compreender o funcionamento de arquiteturas *multi-core* e *many-core*, o aluno deve assimilar o conhecimento do projeto de arquitetura de processadores ao de sistemas de memória e de redes de interconexão.

Para isso, uma metodologia teórica de ensino é necessária, uma vez que essa proporciona ao aluno uma compreensão dos conceitos fundamentais. No entanto, o emprego dessa metodologia de maneira isolada, apresenta duas deficiências. Em primeiro lugar, arquitetura de computadores trata-se de um objeto de estudo complexo e que, na perspectiva dos alunos, é desnecessário. Dessa forma, quando o assunto é abordado estritamente de maneira teórica, o aluno se sente desmotivado. Em segundo lugar, essa metodologia não expõe o aluno às implicações práticas do projeto de arquiteturas de computadores, como por exemplo a frequência máxima de ciclo do processador e o tempo de propagação de dados.

Essas duas deficiências levam o professor a buscar abordagens complementares para o ensino de arquitetura de computadores [1, 2, 3]. Uma delas é a de expor o aluno

a uma metodologia de aprendizado prática, com o intuito de tanto despertar o interesse do aluno pelo assunto quanto o de expô-lo às implicações práticas de um projeto de arquitetura. Nesse contexto, o uso de simuladores de arquiteturas vem sendo apontado como ferramenta de aprendizado [1, 4, 5].

Os simuladores de arquitetura existentes podem ser divididos, grosseiramente, em quatro classes: simuladores de processadores, simuladores de memória, simuladores de rede de interconexão e simuladores de sistemas completos. Os simuladores de processadores modelam um processador em nível de micro-código, possibilitando a simulação de programas escritos em linguagem de montagem e a inspeção do estado do processador (ex: conteúdo dos registradores). Os simuladores de memória são capazes de modelar sistemas de memória com diferentes tamanhos, latências, protocolos de coerência, políticas de substituição dentre outras características. Analogamente, os simuladores de redes de interconexão possibilitam a modelagem de redes de interconexão com diferentes largura de banda, políticas de arbitragem, roteadores e protocolos de roteamento. Por fim, os simuladores de sistemas completos são capazes de simular um sistema computacional completo, incluindo elementos de processamento, sistema de memória, redes de interconexão e dispositivos periféricos, possibilitando assim a execução de aplicações ou até mesmo de sistemas operacionais.

Devido às suas características e capacidades técnicas, o uso de simuladores já se tornou bastante popular na academia, em áreas como o desenvolvimento de sistemas operacionais e o estudo de novas arquiteturas [6]. Já se discute que o emprego de simuladores como ferramenta de aprendizado exige também características didáticas [4, 5]. No entanto, quais características dos simuladores podem beneficiar o ensino de arquitetura de computadores?

Este artigo possui como objetivo a análise e avaliação, para o ensino, de quatro simuladores de sistemas completos populares na academia: Simics [6], Bochs [7], SESC [8] e Gem5 [9]. Os simuladores são comparados segundo suas características principais. O simulador SESC foi escolhido para uso em sala de aula e através deste é feita uma avaliação quantitativa por meio de um questionário respondido por alunos. Portanto, a contribuição desse artigo está relacionada à identificação de características didáticas desejáveis em simuladores completos no ensino de arquitetura de computadores.

O restante do artigo está organizado da seguinte forma. Na Seção II são apresentados os trabalhos relacionados; na Seção III são apresentados os simuladores de sistemas completos analisados; na Seção IV é apresentada a

metologia adotada; na Seção V são apresentados os resultados desse trabalho e na Seção VI suas conclusões.

## II. TRABALHOS RELACIONADOS

No trabalho de Xavier, et al. [4], é feito um estudo comparativo entre simuladores de memória *cache* e sua aplicabilidade no ensino, com o objetivo de facilitar a seleção de ferramentas didáticas para o ensino de arquitetura de computadores. Nesse trabalho, os autores avaliam os simuladores KSH, MSCSim, DCMSim e Dinero IV segundo nove características pertinentes entre eles: simulação de *cache* dividida, simulação de *cache* multi-níveis, tipos de acesso (sequencial ou paralelo), tempo de acesso, erros compulsório/conflito/capacidade, gerador de *memory-trace*, apresentação de estatísticas, interface didática e animação. Ao final, os autores elegem o MSCSim como o simulador mais completo avaliado, mas concluem que existe uma grande carência em ferramentas de simulação, especificamente de memória *cache*, e apontam possíveis soluções para o problema.

No trabalho de Nikolic, et al. [5], é feita uma análise e um estudo avaliativo de simuladores adequados para o ensino de organização e arquitetura de computadores, com o objetivo de auxiliar na escolha de simuladores didáticos nessa área. Nesse trabalho, os autores analisam mais de vinte simuladores, incluindo simuladores de memória e de sistemas completos, e os avaliam segundo dois critérios: características pertinentes e abrangência de conteúdo. No critério características pertinentes, os simuladores são avaliados quanto à sua complexidade, suporte ao projeto de novas arquiteturas, interface gráfica, granularidade de simulação (ciclo de *clock*, instrução ou programa), suporte a simulação iterativa, suporte a *debugging/tracing* e suporte a acesso remoto. No critério abrangência de conteúdo, os simuladores são avaliados em diversos tópicos das unidades de ensino: fundamentos de arquitetura de computadores, organização e arquitetura de sistemas de memória, interface e comunicação, sub-sistema de dispositivos, projeto de processadores e organização da CPU. Dentre os simuladores avaliados, o simulador de sistema completo M5 se destacou, atendendo a 100% e 69,57% dos quesitos, nos critérios características pertinentes e abrangência de conteúdo respectivamente.

O trabalho apresentado neste artigo se diferencia dos trabalhos relacionados discutidos anteriormente em dois aspectos: (I) este trabalho se limita a avaliação de simuladores de sistemas completos e (II) os simuladores selecionados são avaliados de maneira comparativa, segundo suas características pertinentes, e de maneira quantitativa, através da escolha de um simulador para uso em sala de aula e de um questionário respondido por alunos.

## III. SIMULADORES DE SISTEMAS COMPLETOS

Os simuladores Simics [6], Bochs [7], SESC [8] e Gem5 [9] foram selecionados para estudo devido à sua popularidade na academia e a prévia familiaridade dos autores com as ferramentas.

### A. Simics

O Simics é um simulador de sistema completo proprietário, desenvolvido pela Wind River e escrito em Python e C. Esse simulador é capaz de (I) modelar arquiteturas *single-core*, *multi-core* e *many-core*; (II) simular os conjuntos de instruções Ultra Sparc, Alpha, x86, x86-64, Power PC, IPF, MIPS e ARM, em nível de

instrução; e (III) emular sistemas operacionais inalterados, incluindo: Linux, VxWorks, Solaris, FreeBSD, QNX, RTEMS e Windows.

O sistema simulado por essa ferramenta é parametrizável em nível de componentes e arquitetural ultra-fino. A parametrização em nível de componentes permite a configuração do número de processadores, da quantidade de memória RAM, dispositivos periféricos disponíveis e o tipo de barramento (ISA ou PCI). Já a parametrização em nível arquitetural permite a configuração do tipo (superescalar, *pipelined* e *multithreaded*), frequência de operação e tempos de acesso do processador; do sistema de memória *cache* (níveis disponíveis, tamanho, latência, organização, associatividade, protocolos de coerência e política de substituição de blocos); da memória RAM (tecnologia, latência e frequência de operação) e da rede de interconexão (largura, topologia e tipo de roteador).

Além disso, esse simulador possui suporte a *tracing* e um *debugger* interno, que permite o controle do fluxo de execução da simulação e a inspeção completa do estado do processador, subsistema de memória e dispositivos periféricos. O Simics também suporta a modelagem de sistemas distribuídos e possui integração com linguagens de descrição de *hardware*, possibilitando que novos componentes sejam descritos e integrados ao simulador.

### B. SESC

O SESC é um simulador de sistema completo de código aberto, desenvolvido pelo grupo ia-coma da Universidade de Illinois e escrito em C++. Esse simulador é capaz de simular arquiteturas MIPS *single-core*, *multi-core* e *many-core* em nível de instrução, e de emular a execução de aplicações inalteradas.

Assim como no Simics, o sistema simulado por essa ferramenta é parametrizável em nível de componentes e arquitetural ultra-fino, conforme a descrição realizada na Subseção III. A. Além disso, o SESC possui suporte a *tracing* e, apesar de não possuir *debugger* interno, oferece suporte à *debugging* remoto.

### C. Gem5

O Gem5 é um simulador de sistema completo de código aberto; escrito em Python e C++; desenvolvido a partir dos simuladores M5 [10] e GEMS [11]; e resultado dos esforços de instituições acadêmicas e industriais. Esse simulador é capaz de (I) modelar arquiteturas *single-core*, *multi-core* e *many-core*; (II) simular os conjuntos de instruções Alpha, MIPS, PowerPC, SPARC e x86; e (III) emular a execução tanto de aplicações, quanto de sistemas operacionais, incluindo: Linux, FreeBSD e Solaris.

O sistema simulado por essa ferramenta é parametrizável em nível de componentes e em nível arquitetural fino. A parametrização em nível de componentes permite a configuração do número de processadores, da quantidade de memória RAM disponível e da frequência de operação do processador. Já a parametrização em nível arquitetural permite a configuração da memória RAM (tecnologia e frequência de operação), do sistema de memória *cache* (níveis disponíveis, tamanho, associatividade e protocolos de coerência), do tipo de processador (uni-ciclo, multi-ciclo, *in-order pipelined*, *out-of-order pipelined*) e da rede de interconexão (topologia, tipo de roteador e suporte a *adaptive routing*).

Além disso, o Gem5 suporta simulação iterativa, integração com o simulador de redes de interconexão Topaz [12], possui suporte a *tracing* e oferece suporte a *debugging* remoto.

#### D. Bochs

O Bochs é um simulador de sistema completo de código aberto, desenvolvido por Kevin Lawton e escrito em C++. Esse simulador é capaz de (I) modelar arquiteturas *single-core* e *multi-core*; (II) simular os conjuntos de instruções x86 e x86-64; e (III) emular sistemas operacionais inalterados incluindo: Linux, FreeBSD, Solaris e Windows.

O sistema simulado por essa ferramenta é parametrizável somente em nível de componentes, sendo possível definir: o modelo de processador, quantidade de memória RAM, dispositivos periféricos disponíveis e o tipo de barramento (ISA ou PCI). Além disso, o simulador possui um *debugger* interno que permite o controle do fluxo de execução da simulação, e a inspeção do estado do processador.

### IV. METODOLOGIA

Nesta seção é discutida a metodologia utilizada para a avaliação dos simuladores apresentados.

#### A. Análise de Características

Para realizar uma avaliação comparativa entre os simuladores, um estudo deles foi feito através da análise de sua documentação, disponível em artigos, sites, vídeos e fóruns de discussão. Em seguida, os parâmetros de comparação foram definidos baseando-se em métricas propostas nos trabalhos relacionados e nas características pertinentes entre os simuladores. Os parâmetros de comparação definidos e discutidos a seguir são: tipo de licença, tipos de arquitetura simuláveis, granularidade de parametrização, modos de simulação, suporte a *debugging*, suporte a *tracing* e documentação.

O tipo de licença do simulador é relevante para sua escolha como ferramenta de ensino. Simuladores com licenças abertas, como GPL e BSD, são gratuitos e podem ter seu código fonte modificado para que, por exemplo, uma nova funcionalidade seja adicionada ao simulador. Em contrapartida, simuladores com licença proprietária, quando adquiridos, são dificilmente modificáveis. O parâmetro *tipo de licença* indica o tipo de licença dos simuladores estudados.

Ao final de um curso de arquitetura de computadores, deseja-se que o aluno tenha conhecimento tanto de arquiteturas primitivas como as *single-core*, quanto as mais modernas e complexas arquiteturas *many-core*. O parâmetro *tipos de arquitetura simuláveis*, indica essa capacidade dos simuladores.

Analisando os simuladores de sistemas completos estudados, percebeu-se uma diferença quanto à capacidade de parametrização do sistema simulado. Alguns simuladores permitem somente a parametrização de componentes (ex: número de processadores e dispositivos periféricos disponíveis), enquanto outros também permitem uma parametrização arquitetural fina (ex: tipo do processador e tamanho de *cache*) ou até mesmo ultra-fina (ex: latências de memória e protocolos de substituição de blocos de *cache*). O parâmetro *granularidade de parametrização* expressa essa capacidade do simulador.

Dentre os simuladores de sistemas completos estudados observou-se uma distinção entre dois modos de simulação: (SE) simulação de aplicações de usuário e (FS) simulação de um sistema completo com dispositivos periféricos e sistema operacional. O parâmetro *modos de simulação* indica essa capacidade dos simuladores.

Uma das características que estimulam o emprego de simuladores no ensino de arquitetura de computadores é o suporte a *debugging*, que permite o aluno inspecionar o estado da máquina durante a execução de uma carga de trabalho, e o suporte a *tracing*, que torna possível a análise da execução de uma carga de trabalho na arquitetura simulada. Os parâmetros *suporte a debugging* e *suporte a tracing* expressam essas características.

Por fim, espera-se que um simulador didático possua uma boa documentação de forma a facilitar o uso da ferramenta por alunos e possibilitar uma maximização de aprendizado. O parâmetro *documentação* avalia esse quesito dos simuladores.

#### B. Avaliação em Sala de Aula

Para realizar uma avaliação quantitativa do uso de simuladores, um questionário foi elaborado e repassado a alunos de graduação dos cursos de Engenharia de Computação e Ciência da Computação da PUC Minas. Esses alunos utilizaram o simulador de sistema completo SESC para estudar e aprimorar uma arquitetura exemplo introduzida em sala. Esse simulador foi escolhido como ferramenta de ensino por possuir código aberto, ser capaz de modelar arquiteturas *single-core*, *multi-core* e *many-core*, possibilitar a parametrização ultra-fina do sistema simulado e suportar a execução e *tracing* de aplicações reais.

O questionário foi elaborado de forma a investigar (I) o índice de aprovação dos alunos para o emprego desse tipo de ferramenta no ensino, (II) o grau de dificuldade dos alunos na utilização da ferramenta e (III) o nível de aproveitamento de aprendizado proporcionado pela ferramenta. O questionário, apresentado na Tabela I, é constituído de 11 perguntas de múltipla escolha que permitem uma avaliação em escala (notas) de 1 a 4, sendo 1 a pior nota e 4 a melhor nota. As notas 2 e 3 representam tendência de insatisfação e satisfação, respectivamente.

TABELA I: QUESTIONÁRIO REPASSADO AOS ALUNOS.

Pergunta 1	A escolha de um simulador de sistema completo para ensino foi acertada?
Pergunta 2	A documentação é necessária para o uso em educação?
Pergunta 3	Fórum de discussão é necessário para o uso em educação?
Pergunta 4	Qual o grau de facilidade para entender o simulador usado?
Pergunta 5	O ambiente facilitou a execução de programas reais?
Pergunta 6	O ambiente facilitou na visualização e geração de estatísticas de desempenho?
Pergunta 7	Você se sentiu motivado a alterar a arquitetura exemplo?
Pergunta 8	Você conseguiu configurar diferentes arquiteturas?
Pergunta 9	Você aprendeu sobre arquiteturas multi-core?
Pergunta 10	Qual a sua motivação em trabalhos futuros?
Pergunta 11	Você recomendaria para seu colega o uso deste ambiente de aprendizagem?

TABELA II: CARACTERÍSTICAS GERAIS PERTINENTES AOS SIMULADORES.

	Tipo de Licença	Arquiteturas Simuláveis	Granularidade de Parametrização	Modos de Simulação	Suporte a Debugging	Suporte a Tracing	Documentação
Bochs	LGPLv2	single/multi	grossa	FS	interno	não	rica
Gem5	BSD	single/multi/many	fina	FS - SE	remoto	sim	rica
SESC	GPL	single/multi/many	ultra-fina	SE	remoto	sim	escassa
Simics	Proprietária	single/multi/many	ultra-fina	FS	interno	sim	escassa

## V. RESULTADOS

Dentre os simuladores analisados (Tabela II), o único que apresenta licença proprietária é o Simics. Os demais apresentam licença aberta e possuem seu código fonte disponível gratuitamente na Internet. Quanto aos tipos de arquiteturas simuláveis, todos os simuladores são capazes de simular arquiteturas *single-core*, *multi-core* e *many-core*; exceto o simulador Bochs que é incapaz de simular arquiteturas *many-core*. Observando o parâmetro de comparação granularidade de parametrização dois simuladores se destacam, o Simics e o SESC, que suportam um ajuste ultra-fino de componentes e da arquitetura. Considerando os modos de simulação, o Gem5 se destaca por ser capaz de emular tanto a execução de aplicações, quanto de sistemas operacionais. Analisando os parâmetros suporte a *debugging* e suporte a *tracing*, percebe-se que todos os simuladores oferecem suporte a *debugging*, mas nem todos eles suportam *tracing*, que é o caso do simulador Bochs. Por fim, observando o parâmetro de comparação documentação, os simuladores Bochs e Gem5 se destacam devido à rica documentação, que está disponível em artigos, sites, fóruns e tutoriais.

Nenhum simulador de sistema completo possui todas as características didáticas desejáveis: licença aberta; simulação de arquiteturas *single-core*, *multi-core* e *many-core*; granularidade de parametrização ultra-fina; suporte aos modos de simulação de aplicativos (SE) e de sistemas completos (FS); suporte à *debugging e tracing*; e documentação rica. No entanto, dois simuladores se destacam: o Gem5, que oferece suporte aos modos de simulação SE e FS; e o SESC, que possibilita uma parametrização ultra-fina do sistema simulado.

O resultado da avaliação quantitativa é apresentado na Figura 1. Um total de 18 alunos, que utilizaram o simulador SESC para estudar e aprimorar uma arquitetura exemplo introduzida em sala, participaram da pesquisa. A análise dos dados obtidos revela que os alunos consideraram que um simulador de sistema completo é uma escolha acertada para o ensino (Pergunta 1) e que o emprego desse tipo de ferramenta os ajudou a aprender (Pergunta 9) sobre arquiteturas *multi-core*. No entanto, observa-se que os alunos tiveram dificuldade para entender o simulador utilizado e julgaram necessário a existência de documentação (Pergunta 2) e fóruns de discussão (Pergunta 3) da ferramenta.

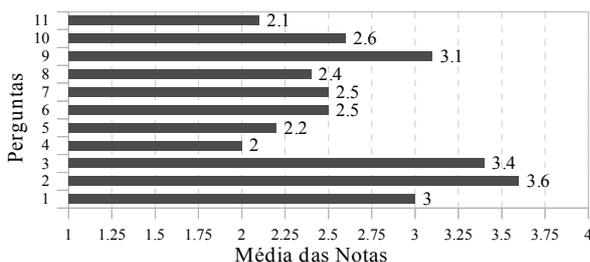


Figura 1: Resultado da avaliação quantitativa dos simuladores estudados.

## VI. CONCLUSÕES

Esse trabalho apresenta uma análise de quatro simuladores de sistemas completos, com o objetivo de analisar e avaliar suas características relevantes ao ensino de arquitetura de computadores.

Para isso, os simuladores são comparados segundo suas características e avaliados quantitativamente através de um questionário respondido por alunos que utilizaram o simulador SESC em sala de aula. A avaliação comparativa revela que, nenhum simulador atende à todas as características desejáveis. No entanto, os simuladores Gem5 e SESC mostram-se os mais indicados. Já a avaliação quantitativa revela a aprovação dos alunos quanto ao uso de simuladores de sistemas completos como ferramenta de ensino. Contudo, foi observado que os alunos tiveram dificuldade para entender o simulador utilizado, possivelmente devido à escassez de documentação da ferramenta.

Como trabalho futuro, o simulador Gem5 será usado em sala de aula com aplicação da mesma metodologia.

## REFERÊNCIAS

- [1] H. C. Freitas, M. A. Z. Alves, N. Maillard e P. O. A. Navaux, "Ensino de Arquitetura de Processadores Multi-core Através de um Sistema de Simulação Completo e da Experiência Baseada de um Projeto de Pesquisa", In: *Workshop sobre Educação em Arquitetura de Computadores (WEAC 2008)*, pg. 1-8, 2008.
- [2] R. O. Duarte, P. F. D. Garcia, "Metodologia de Ensino Orientada a Projetos e Criação de Material Didático aplicada ao Projeto de Processadores", In: *Workshop sobre Educação em Arquitetura de Computadores (WEAC 2011)*, pg. 1-6, 2011.
- [3] R. Ferreira, "Arquitetura de Computadores em uma Simples Folha de Papel", In: *Workshop sobre Educação em Arquitetura de Computadores (WEAC 2011)*, pg. 19-24, 2011.
- [4] M. A. S. Xavier, J. C. Rodrigues, O. A. L. Júnior, "Simuladores de Memória Cache, um Estudo Comparativo Direcionado ao Ensino", In: *Workshop sobre Educação em Arquitetura de Computadores (WEAC 2011)*, pg. 7-12, 2011.
- [5] B. Nikolic, Z. Radivojevic, J. Djordjevic e V. Milutinovic, "A Survey and Evaluation of Simulators Suitable for Teaching Courses in Computer Architecture and Organization", In: *Journal IEEE Transactions on Education*, v. 52-4, pg. 449-458, nov. 2009.
- [6] P. S. Magnusson et Al, "Simics: A Full System Symulation Platform" In: *Computer IEEE*, v. 35-2, pg. 50-58, fev 2002.
- [7] Bochs, <http://bochs.sourceforge.net/>.
- [8] P. M. Ortego e P. Stack, "SESC: SuperEScalar Simulator", Relatório técnico disponível em: <http://iacoma.cs.uiuc.edu/~paulsack/sescdoc/sescdoc.pdf>, dec. 2004.
- [9] N. Binkert et Al, "The gem5 Simulator", In: *Newsletter ACM SIGARCH Computer Architecture News*, v. 39-2, pg. 1-7. 2011.
- [10] N. L. Binkert et Al, "The M5 Simulator: Modeling Networked Systems", In: *Micro, IEEE*, v. 26-4, pg. 52-60, ago, 2006.
- [11] M. M. K. Martin et Al, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset", In: *Newsletter ACM SIGARCH Computer Architecture News*, v. 33-4 pg. 92-99., 2005.
- [12] P. Abad et Al, "TOPAZ: An Open-Source Interconnection Network Simulator for Chip Multiprocessors and Supercomputers", In: *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*, pg. 99-106, 2012.