APPROXSS: UM SIMULADOR DE INJEÇÃO DE ERRO E CONSUMO DE ENERGIA PARA SISTEMAS COM MEMÓRIA APROXIMADA

MATHEUS ISQUIERDO[1]; BRUNO ZATT[2]; DANIEL PALOMINO[3]

[1]*Universidade Federal de Pelotas – mmisquierdo@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas – zatt@inf.ufpel.edu.br*
[3]*Universidade Federal de Pelotas – dpalomino@inf.ufpel.edu.br*

1. INTRODUÇÃO

Memória é parte essencial de sistemas computacionais, mas também um conhecido gargalo de desempenho e foco de consumo de energia (NAIR, 2015). Algumas abordagens para a redução desse consumo de energia no contexto de aplicações envolvem heurísticas e melhorias algorítmicas, trazendo-a como subproduto de menores tempo de acesso, ou implementações em *hardware* especificas, com estruturas de memória dedicadas (CORRÊA, 2015).

Nesse contexto, armazenamento aproximado é um paradigma promissor para lidar com as limitações relacionadas a tecnologias de memória, trazendo ganhos em consumo de energia, tempo de acesso, etc. Entretanto, tais ganhos apenas são atingidos com o sacrifício da precisão dos dados armazenados, submetendo-os a possibilidade de erros e, consequentemente, levando a perdas na qualidade da aplicação alvo. Em aplicações ditas resilientes a erros, essas perdas de qualidade podem ser consideradas aceitáveis (CHIPPA et al., 2013).

Para o emprego eficiente de armazenamento aproximado é necessária a avaliação de resiliência a erros das aplicações, identificando porções de memória mais resilientes a erros e quantificando os seus graus de tolerância para tirar proveito dessa capacidade, enquanto se mantém uma qualidade aceitável. Devido à complexidade e elevados custos de se construir fisicamente sistemas computacionais para testes envolvendo armazenamento aproximado, sua viabilidade e seus benefícios, se faz necessário o uso de simuladores.

De forma geral, atualmente, existe uma escassez de simuladores focados na avaliação de armazenamento aproximado em memórias locais de aplicações, sejam amplamente disponíveis ou na literatura. Nenhuma ferramenta atual se especializa na simulação de aplicações que empregam armazenamento aproximado, no contexto de buffers específicos, com as finalidades de exploração, avaliação de resiliência a erros e estimação de consumo de energia. Portanto, o objetivo deste trabalho é o desenvolvimento de um simulador de memória aproximada, que replique os efeitos colaterais e permita a estimação do potencial de economia de energia quando memória aproximada é empregada.

2. METODOLOGIA

O simulador desenvolvido se chama ApproxSS, para *Approximate Storage Simulator*, sendo constituído por um injetor de erros e um estimador de consumo de energia. Ele é construído em cima da plataforma *Pin* (LUK et al., 2005), um *framework* de instrumentação binária dinâmica desenvolvido pela Intel, com suporte a várias arquiteturas, sistemas operacionais e compiladores. A finalidade do injetor é replicar os efeitos do emprego de armazenamento aproximados sobre aplicações, fazendo injeções aleatórias de erros nos conteúdos de operações de leitura e escrita efetuadas sobre intervalos especificados da memória de uma

aplicação. Além disso, ele obtém dados sobre acessos feitos à memória de tais intervalos para posteriores cálculos de consumo. As injeções de erros são feitas por *bit-flips* controlados por Taxas de Erros de *Bits* (TEBs) e limitadas por comprimentos de *bits*, que determinam até qual *bit* de um acesso haverá injeção.

A ferramenta de injeção possui uma série de funções de controle, permitindo a delimitação de intervalos de endereços como *buffers* aproximados; ativação e desativação da injeção de erros e instrumentação de acessos; e troca de TEBs. Chamadas às essas funções devem ser adicionadas no código-fonte da aplicação alvo como marcadores de instrumentação. O simulador também oferece algumas diretivas de preprocessamento de compilação, a fim reduzir *overhead* quando certas funcionalidades não são necessárias, como múltiplos *buffers* aproximados simultâneos, múltiplos TEBs por configuração e contabilização de erros injetados. Já o estimador de consumo de energia é relativamente simples, realizando cálculos de consumo energético dos *buffers* aproximados com base nos registros de acessos à memória e perfis de consumo de energia.

Para atestar o funcionamento do simulador, foi elaborado um estudo de caso em cima do decodificador do *Versatile Video Coding* (VVC) (JVET, 2022), um padrão de codificação de vídeo. O *buffer* escolhido para ser aproximado foi o *Buffer* de Linha de Referência (BLR) da predição intra-quadro. Apenas um vídeo foi decodificado, o *FourPeople*, codificado com o Parâmetro de Quantização (QP) 22 e a configuração *All Intra* (AI), que força o uso do *buffer* instrumentado para a decodificação de todos os píxeis do vídeo. As decodificações se deram conforme a Tabela 1, com dados obtidos a partir de uma SRAM 6T em litografia de 28nm CMOS de 32kb sob 22 °C (FRUSTACI et al., 2015). O vídeo foi decodificado 100 vezes para cada tensão de alimentação aproximada. E a métrica de qualidade de serviço empregada foi o Peak Signal-to-Noise Ratio (PSNR).

Tabela 1 - Consumo por acesso e taxas de erro de bit para operações de escritas e leituras sob diversas tensões para uma memória SRAM 6T CMOS de 28nm (FRUSTACI et al., 2015)

Operação	Leitura				Escrita			
Tensão de Alimentação (V)	0,55	0,6	0,7	0,85	0,55	0,6	0,7	0,8
Taxa de Erro de Bit (TEB)	0,037	0,011125	0,003375	0,0	0,289	0,135	0,01	0,0
Consumo Por *Byte* (pJ)	3,01	3,29	4,59	7,21	2,99	3,27	4,59	5,76

3. RESULTADOS E DISCUSSÃO

A Figura 1 apresenta um *boxplot* dos valores de PSNR obtidos. O primeiro comportamento a se notar é a massiva queda de qualidade da aplicação alvo sob as tensões aproximadas. Isso era esperado em algum grau, devido a altas TEBs impostas por essas tensões. Por exemplo, sob a tensão de 0,7V (a de menores TEBs entre as abordadas) e considerando que cada canal de um píxel possui 8 *bits*, em média 1 em cada 37 amostras lidas acaba sofrendo um *bit-blip*. No caso de escrita, sob a mesma tensão, a questão é consideravelmente mais delicada, com 1 em cada aproximadamente 12 amostras escritas sofrendo um bit-flip, em média. Acompanhando isso, também há fato de que individualmente erros em operações de escrita são potencialmente mais prejudiciais à qualidade do que os em escritas, pois eles são permanentes, ou seja, toda leitura sobre um dado

escrito com erro conterá aquele erro, assim propagando-o. Além disso, conforme a tensão é diminuída, as perdas de qualidade também aumentam, já que o buffer BLR é submetido a TEBs cada vez maiores. Isso acaba aumentando a probabilidade de que qualquer dado *bit* sofra um *bit-flip*, causando um maior desvio em relação ao vídeo original e maior degradação em comparação com aqueles vídeos decodificados com tensões mais próximas da tensão precisa.

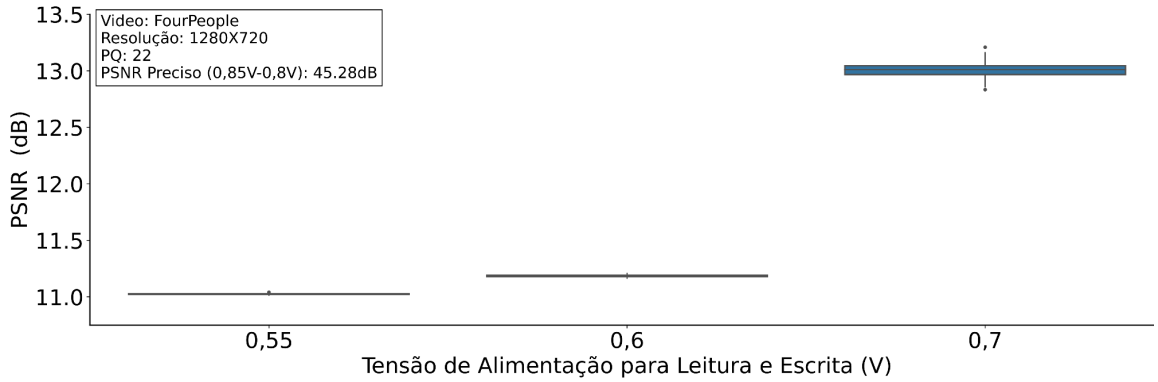

Figura 1 - Medições da qualidade visual em PSNR do vídeo FourPeople codificado com um QP de 22 sob a configuração All Intra do VVC e decodificado sob várias tensões de alimentação

A partir da Figura 2, podemos notar o crescente consumo de energia acompanhando a aproximação da tensão de alimentação à tensão precisa, algo esperado, considerando que esse é o comportamento apresentado na Tabela 1. Por exemplo, a tensão precisa traz consumos de 190,43µJ e 78,52µJ para leitura e escrita, respectivamente, enquanto a tensão de 0,55V apresenta reduções de 58,25% e 48,09% para estas operações, com consumos de 79,50µJ e 40,76µJ. As demais tensões causam reduções mais modestas, mas ainda consideráveis, de 54,37% e 43,23% para a tensão 0,6V; e 36,34% e 20,31% para a tensão 0,7V. Além disso, operações de leitura consistentemente apresentam consumos de energia maiores em relação à escritas. Como, de forma geral, o consumo por acesso é apenas ligeiramente maior para leituras, isso acaba se devendo em grande parte aos seus maiores volumes ao longo da aplicação alvo, que favorece a leitura das amostras presentes no *buffer* BLR. Apenas no caso da decodificação precisa que o consumo por acesso individual realmente se difere entre as operações, com o da leitura sendo 25,17% maior do que o da leitura, graças a sua maior tensão precisa.

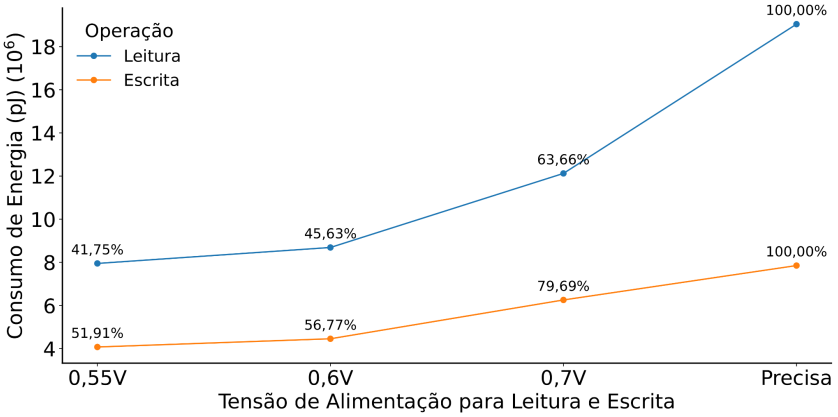

Figura 2 - Consumo de energia do *buffer* BLR aproximado sob várias tensões na decodificação do vídeo *FourPeople* (*All Intra* e QP 22)

4. CONCLUSÕES

Conforme apresentado neste resumo, foi desenvolvido um simulador de armazenamento aproximado, trazendo uma ferramenta de injeção de erros aleatórios em intervalos especificados de memória de uma aplicação alvo e uma ferramenta de estimação de consumo de energia para tais intervalos. A ferramenta de injeção de erros oferece meios de fácil instrumentação de um aplicação alvo através de marcadores, além de diretivas de compilação para a redução do *overhead* imposto pelo seu uso. O simulador desenvolvido tem como objetivo oferecer uma base para futuros trabalhos envolvendo a exploração de armazenamento aproximado em contextos variados, para a avaliação de resiliência a erros e consumo de energia em fases iniciais de desenvolvimento de implementações em *hardware*.

A aplicação alvo do estudo de caso acabou não apresentando resiliência a erros nas condições testadas, devido a uma série de fatores, como altos TEBs, inclusão de erros em escritas, uso exclusivo da predição intra-quadro, efeito direto dos erros sobre a representação dos píxeis a serem consumidos, entre outros. Outros contextos são potencialmente mais favoráveis ao emprego de armazenamento aproximado para trabalhos futuros, principalmente onde os erros ocorridos não possuem impacto direto na qualidade final da aplicação alvo, mas sim causam a geração de resultados sub-ótimos em algoritmos de otimização, como a estimação de movimento em codificadores de vídeo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

CORRÊA, G. R. **Computational Complexity Reduction and Scaling for High Efficiency Video Encoders**. In: ACM PRESS, 2015. Anais. . . [S.l.: s.n.], 2015.

CHIPPA, V. K.; CHAKRADHAR, S. T.; ROY, K.; RAGHUNATHAN, A. **Analysis and characterization of inherent application resilience for approximate computing**. In: Annual Design Automation Conference ON - DAC ’13, 50., 2013, Austin, Texas. Proceedings. . . ACM Press, 2013. p.1.

FRUSTACI, F. et al. **SRAM for Error-Tolerant Applications With Dynamic Energy-Quality Management in 28 Nm CMOS**. IEEE Journal of Solid-State Circuits, [S.l.], v.50, n.5, p.1310–1323, May 2015.

JVET. **VVCSoftware_VTM**. ver. 18.0. Disponível em: <https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-18.0>.

LUK, C.-K. et al. **Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation**. ACM SIGPLAN Notices, [S.l.], v.40, n.6, p.11, June 2005.

NAIR, R. **Evolution of Memory Architecture**. Proceedings of the IEEE, [S.l.], v.103, n.8, p.1331–1345, Aug. 2015.