

## UMA PROPOSTA PARA AVALIAÇÃO DE UM MECANISMO PARA CONTROLADORA DE MEMÓRIA EM DESEMPENHO E CONSUMO DE ENERGIA PARA ARQUITETURAS COM MEMÓRIAS NÃO VOLÁTEIS

GIOVANE DE OLIVEIRA TORRES<sup>1</sup>; LAÉRCIO LIMA PILLA<sup>2</sup>; MAURÍCIO LIMA  
PILLA<sup>3</sup>

<sup>1</sup>Universidade Federal de Pelotas – [gdotorres@inf.ufpel.edu.br](mailto:gdotorres@inf.ufpel.edu.br)

<sup>2</sup>Universidade Federal de Santa Catarina – [laercio.pilla@ufsc.br](mailto:laercio.pilla@ufsc.br)

<sup>3</sup>Universidade Federal de Pelotas – [pilla@inf.ufpel.edu.br](mailto:pilla@inf.ufpel.edu.br)

### 1. INTRODUÇÃO

Hoje existem grandes demandas para que múltiplos aspectos de dispositivos eletrônicos obtenham melhorias, já que existe uma constante necessidade por maior poder computacional. Neste contexto, são importantes descobertas e inovações tecnológicas que permitam aperfeiçoamentos em questões como, por exemplo, desempenho e consumo energético. Uma das áreas que necessitam melhorias são as memórias utilizadas atualmente, devido a vários fatores. Um destes fatores ocorre em razão das memórias estarem atingindo o seu limite de escalabilidade (OUKID; KETTLER; WILLHALM, 2017), bem como vazamento de energia (AWAD et al., 2016). Além disto, memórias são consideradas como um dos componentes críticos em computadores (POURSHIRAZI; ZHU, 2016). Assim, para que progressos na área de computação continuem no futuro, torna-se essencial a melhora nas memórias.

Visando as melhorias nas memórias, tem-se a possibilidade de substituir as tecnologias de memórias atuais, como DRAM e SRAM, por memórias não voláteis (MNVs). Este tipo de memórias tem como principais objetivos: (i) prover baixo consumo energético, (ii) melhor escalabilidade e (iii) maior densidade por célula de memória (MEENA et al., 2014; ZHAO et. al., 2015). A característica de não volatilidade faz com que um dado armazenado na célula deste tipo de memórias seja mantido durante um longo período de tempo (variável de acordo com o material da célula de memória), o que não ocorre numa memória volátil, as quais necessitam de operações de *refresh* para que o dado fique retido. Contudo, algumas dificuldades são encontradas quando do uso de MNVs em maior escala. Estes incluem: durabilidade do material das memórias é relativamente pequeno; Maior gasto de tempo e consumo energético para efetuar operações de escritas em células de memórias (WONG; SALAHUDDIN, 2015).

Em algumas categorias de sistemas (e.g., sistemas de tempo real), um conjunto de operações deve finalizar dentro de um determinado tempo (FUJITA, 2014). Como opção para melhorar em especial o consumo energético de sistemas, o uso de MNVs é uma opção bem considerada. Considerando sistemas com uma MNV de memória principal, uma considerável quantidade de conjuntos de operações podem não cumprir seus prazos, acarretando em demoras não aceitáveis nos seus tempos de execução (DASARI; NELIS; MOSSE, 2013). A fim de superar este problema, o emprego de uma política que vise tornar os tempos das requisições de operações de memória mais apertados pode ser útil no sentido de despachar escritas em memórias mais rapidamente, dado que estas operações demandam mais tempo para efetuação.

Assim, podem-se efetuar modificações nos algoritmos da controladora de memória para conseguir atender às operações de memória dentro de tempos aceitáveis. Dentro deste contexto, este trabalho propõe como objetivo efetuar a

implementação de um mecanismo diferente para atender a requisições de memória, considerando um sistema o qual tem uma MNV como memória principal. Com isto, deverá ser avaliado o impacto da implementação deste novo mecanismo nas questões de desempenho e consumo de energia. O mecanismo a ser estudado é descrito largamente no trabalho de DASARI; NELIS; MOSSE (2013). As avaliações descritas neste trabalho são baseadas em traços de instruções gerados por aplicações executadas, sendo então efetuado um pós-processamento nos traços gerados. Já este trabalho visa fazer uma avaliação do mecanismo simulada em tempo de execução.

## 2. METODOLOGIA

Para a realização deste trabalho, inicialmente deve ser escolhido uma ferramenta capaz de simular aplicações e memórias não voláteis que permita a implementação do novo mecanismo de tratamento de operações de memória. As ferramentas escolhidas para este trabalho foram o simulador de MNVs NVMain (POREMBA; XIE, 2012) e o simulador de propósito geral Gem5 (BINKERT, 2011).

NVMain é uma ferramenta com objetivo principal de simular tecnologias de memórias emergentes, o que inclui MNVs. Este simulador foi escolhido por causa de diversos fatores, os quais são:

- Código-fonte é disponibilizado de maneira livre e atualizado regularmente;
- Preocupação em manter código legível e modularizado para simulação de tecnologias emergentes de memórias, visto que é uma área sujeita a constantes mudanças;
- Permite a simulação mais próxima de tempo real, se acoplado ao simulador Gem5.

O Gem5 é um simulador de propósito geral, sendo considerada uma plataforma modular para pesquisas no nível de arquitetura de sistema e/ou microarquitetura de processador (GEM5, 2017). O *software* foi escolhido devido a algumas características, as quais incluem:

- Permitir a combinação juntamente ao NVMain para simulação próxima a tempo real;
- Código-fonte disponível livremente e atualizado;
- Capacidade de simulação de múltiplos sistemas com diferentes ISA (*Instruction Set Architecture*), incluindo ALPHA, ARM, SPARC, X86 e outros;
- É utilizado amplamente em diversos trabalhos na academia.

Escolhidas as ferramentas, o próximo passo deste trabalho está na implementação do novo mecanismo da controladora de memória. Para isto, deverá ser criada uma nova controladora de memória no NVMain para dar suporte ao novo algoritmo.

Este algoritmo é baseado em dois estados, os quais são chamados de **ocupado** e **ocioso**, sendo ambos descritos por meio de autômatos (DASARI; NELIS; MOSSE, 2013). Juntamente a estes autômatos, há um algoritmo de suporte que faz o cálculo do WCET (*Worst Case Execution Time* – Tempo de execução do pior caso) de uma determinada tarefa (a qual engloba uma ou mais operações de memória).

O autômato **ocupado** é responsável por atender requisições de memória enquanto as mesmas existirem. Quando não existirem mais requisições, o algoritmo troca o estado para **ocioso**. Este estado ocorre enquanto não houver nenhuma requisição de memória, aguardando até que uma apareça. Quando isto acontece, o algoritmo então volta ao estado **ocupado**.

Com o algoritmo implementado, deve-se então fazer a validação do simulador com a modificação na controladora de memória. Isto será feito comparando as execuções feitas no simulador sem modificações com o simulador modificado. Espera-se então que as aplicações consigam ser executadas no simulador com a nova controladora de memória – com a saída gerada pelos programas sendo similar às aplicações executadas sobre o simulador original.

Após a validação, devem ser então executados testes envolvendo a simulação de *benchmarks* sobre as duas instâncias do simulador (original e o modificado). Estes testes deverão ser executados para verificar o impacto da nova controladora de memória, tanto em questões de consumo de energia quanto em desempenho.

### 3. RESULTADOS E DISCUSSÃO

Como se trata de uma proposta, este artigo não apresenta resultados concretos. O que se espera obter como resultado é uma avaliação sobre os efeitos da implementação de um novo mecanismo para a controladora de memória nas questões de desempenho e consumo de energia, considerando um sistema com MNV como memória principal.

Espera-se que a implementação consiga amortecer os impactos gerados pelas operações de memória, em especial as escritas, as quais consomem grande quantidade de energia além de levar maior tempo para serem executadas. De maneira geral, espera-se que hajam ganhos nos quesitos que deverão ser avaliados, sendo estes: consumo de energia e desempenho.

Outra questão possível de avaliação está no tipo de MNV a ser avaliada com a nova controladora de memória. Testando a nova implementação em diferentes tecnologias de MNV pode-se também verificar qual a tecnologia que pode beneficiar-se melhor do novo algoritmo para a controladora de memória.

### 4. CONCLUSÕES

Este artigo apresentou uma proposta de trabalho para avaliar os efeitos de implementar um mecanismo diferente para a controladora de memória. Para que isso seja possível, propôs-se que fizesse o uso de dois simuladores, sendo os escolhidos o NVMain e o Gem5, os quais darão suporte para a implementação ser avaliada.

A avaliação deste algoritmo se dará através de simulações, comparando com controladoras de memória já implementadas. Assim, verificar-se-á o impacto gerado pela nova implementação, sendo o foco a avaliação nos quesitos de desempenho e consumo de energia. Além disto, a avaliação sobre diferentes tecnologias de MNVs também torna-se importante no sentido de observar quais tipos de MNVs conseguem extrair os melhores resultados.

Com isto, espera-se que a realização do trabalho proposto consiga obter ganhos em desempenho e consumo de energia com uma implementação de nova controladora de memória.

### 5. REFERÊNCIAS BIBLIOGRÁFICAS

AWAD, A. et al. Silent Shredder: Zero-Cost Shredding for Secure Non-Volatile Main Memory Controllers. **SIGOPS Oper. Syst. Rev.**, New York, NY, USA, v.50, n.2, p.263–276, March 2016.

BINKERT, Nathan et al. The gem5 simulator. **ACM SIGARCH Computer Architecture News**, v. 39, n. 2, p. 1-7, 2011.

DASARI, D.; NELIS, V.; MOSSE, D. Timing analysis of PCM main memory in multicore systems. In: IEEE 19TH INTERNATIONAL CONFERENCE ON EMBEDDED AND REAL-TIME COMPUTING SYSTEMS AND APPLICATIONS, 2013., 2013. **Anais... IEEE**, 2013. p.52–61.

FUJITA, M. Highly-Pipelined and Energy-Saved Computing with Arrays of Non-Volatile Memories. In: INTERNATIONAL CONFERENCE ON INTERDISCIPLINARY ADVANCES IN APPLIED COMPUTING, 2014., 2014, New York, NY, USA. **Anais... ACM**, 2014. p.46:1–46:6. (ICONIAAC '14).

GEM5. **The gem5 Simulator**. Acessado em: 29 de Setembro de 2017. Online. Disponível em: [http://www.gem5.org/Main\\_Page](http://www.gem5.org/Main_Page)

MEENA, J. S.; SZE, S. M.; CHAND, U.; TSENG, T.-Y. Overview of emerging nonvolatile memory technologies. **Nanoscale research letters**, New York, NY, USA, v.9, n.1, p.1, 2014.

OUKID, I.; KETTLER, R.; WILLHALM, T. Storage class memory and databases: Opportunities and challenges. **it-Information Technology**, Berlin, Germany, v.3, n.59, p.109–115, March 2017.

POREMBA, Matt; XIE, Yuan. Nvmain: An architectural-level main memory simulator for emerging non-volatile memories. In: **VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on**. IEEE, 2012. p. 392-397.

POURSHIRAZI, B.; ZHU, Z. Refree: A Refresh-Free Hybrid DRAM/PCM Main Memory System. In: IEEE INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS), 2016., 2016. **Anais... IEEE**, 2016. p.566–575.

WONG, H.-S. P.; SALAHUDDIN, S. Memory leads the way to better computing. **Nature nanotechnology**, London, United Kingdom, v.10, n.3, p.191–194, 2015.

ZHAO, J.; XU, C.; CHI, P.; XIE, Y. Memory and storage system design with nonvolatile memory technologies. **IPSJ Transactions on System LSI Design**