

# Uma proposta de implementação de algoritmos de escalonamento para sistemas com PCM como memórias principais

Giovane de Oliveira Torres<sup>1</sup>, Rodrigo Costa de Moura<sup>1</sup>,  
Laércio Lima Pilla<sup>2</sup>, Maurício Lima Pilla<sup>1</sup>

<sup>1</sup>Centro de Desenvolvimento Tecnológico – UFPel  
Pelotas – RS – Brasil

<sup>2</sup>Departamento de Informática e Estatística – UFSC  
Florianópolis – SC – Brasil

{gdotorres,rcmoura,pilla}@inf.ufpel.br, laercio.pilla@ufsc.br

**Resumo.** *Com as constantes preocupações em consumo de energia, o uso de memórias não voláteis como memórias principais tem sido mais frequente, como a PCM. Um dos problemas destas memórias é a assimetria nos tempos de operações. Neste sentido, o escalonamento de operações torna-se importante, sendo que este trabalho visa adaptar algoritmos estudados para um simulador PCM a fim de avaliar seu impacto no desempenho e consumo de energia.*

## 1. Introdução

Uma das principais preocupações atuais que envolvem sistemas computacionais é a questão de consumo de energia. Em especial, as memórias utilizadas hoje têm esta demanda, dado que estas apresentam diversas possibilidades para melhorias [Wong e Salahuddin 2015]. Dentro deste contexto, outros tipos de memórias podem ser utilizados para substituir a tradicional DRAM. Neste caso, a PCM (*Phase Change Memory*) destaca-se por ser uma memória não volátil, não necessitando de *refresh* periódicos – consequentemente apresentando bom desempenho energético.

Porém, a PCM apresenta alguns desafios, os quais incluem assimetria nas operações de leitura e escrita, bem como baixa durabilidade [Gao et al. 2015]. Com isso, o emprego deste tipo de memórias em sistemas embarcados e de tempo real torna-se difícil, sendo necessário melhorar o gerenciamento do tempo nestes sistemas – tendo em vista que isto impacta significativamente sistemas computacionais.

Dentro deste contexto, o trabalho proposto tem como objetivo a implementação de um método de escalonamento para requisições de memória [Dasari et al. 2013] com simulação propondo melhorias – assim avaliando o impacto causado pelos algoritmos, em questões de desempenho e consumo de energia.

## 2. Método de Escalonamento

O método para escalonamento discutido no trabalho de Dasari et al. (2013) propõe estimar tempos mais apertados para operações de leitura e escrita em vez do método ingênuo. Este último funciona da seguinte maneira: considerando que uma operação leve o pior tempo possível, todas as operações subsequentes terão este tempo para serem executadas.

O método baseia-se na análise da controladora PCM e define dois períodos que esta poderá estar – chamados de **ocupado** e **ocioso**. O primeiro estado denota a fase em que são geradas solicitações de operações de alta prioridade, assim maximizando o tempo para computá-las. Já o segundo estado ocorre quando não existe mais nenhuma solicitação gerada, aguardando até que uma ocorra para voltar ao estado ocupado. É importante observar que o algoritmo encerra-se ao extrapolar o período de *deadline*, i.e., o tempo máximo que uma requisição pode durar. Cada período definido é explicado através de um autômato, podendo ser verificado com mais detalhes em [Dasari et al. 2013].

### 3. Proposta

A proposta deste trabalho é implementar os algoritmos de escalonamento descritos no trabalho de Dasari et al. 2013 para as requisições de operações em uma memória PCM, a fim de avaliar o impacto causado por eles nas questões de desempenho e consumo de energia. A partir dos resultados, pretende-se projetar e avaliar melhorias nos mesmos.

O projeto encontra-se em fase inicial e está sendo desenvolvido com o simulador desenvolvido na Universidade de Pittsburgh que simula uma arquitetura de memória PCM, o qual já foi empregado em trabalhos relacionados anteriores [Ferreira et al. 2010, Zhou et al. 2011]. Atualmente estão sendo feitos estudos tanto nos algoritmos quanto no simulador. Com isto, está sendo avaliada a melhor maneira de inserção dos algoritmos no simulador, o qual está sendo modificado. Assim, é necessário encontrar a parte no código do simulador responsável pela controladora de operações PCM, já que esta é o cerne da proposta.

Os algoritmos já apresentam pseudocódigos indicando que variáveis de controle são necessárias, como por exemplo a capacidade da fila de escritas PCM. Precisa-se então verificar se estas variáveis estão facilmente disponíveis no simulador, caso contrário defini-las no próprio simulador. Com a consolidação do código inserido, deve-se efetuar testes para definir quais são os melhores tempos de *deadline* para as operações, permitindo assim a avaliação final que envolverá desempenho e consumo energético.

### Referências

- Dasari, D., Nelis, V., e Mosse, D. (2013). Timing analysis of pcm main memory in multicore systems. In *2013 IEEE 19th International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 52–61. IEEE, IEEE.
- Ferreira, A. P., Childers, B., Melhem, R., Mosse, D., e Yousif, M. (2010). Using pcm in next-generation embedded space applications. In *2010 16th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 153–162.
- Gao, S., Xu, J., Härder, T., He, B., Choi, B., e Hu, H. (2015). Pcmlogging: Optimizing transaction logging and recovery performance with pcm. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3332–3346.
- Wong, H.-S. P. e Salahuddin, S. (2015). Memory leads the way to better computing. *Nature nanotechnology*, 10(3):191–194.
- Zhou, M., Bock, S., Ferreira, A. P., Childers, B., Melhem, R., e Mosse, D. (2011). Real-time scheduling for phase change main memory systems. In *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 991–998.