

# Implementação de instruções de Memória Transacional em Hardware no OpenMP

Fernando Emilio Puntel<sup>1,2</sup>, Frederico Peixoto Antunes<sup>1</sup> e Gerson Geraldo H. Cavalheiro<sup>1</sup>

<sup>1</sup>Universidade Federal de Pelotas  
Pelotas – RS – Brasil

<sup>2</sup>Faculdade Antonio Meneghetti  
Restinga Sêca - RS - Brasil

{fepuntel,gerson.cavalheiro}@inf.ufpel.edu.br

**Resumo.** *Memória transacional em hardware (HTM) minimiza custos de desempenho implementando mecanismos de detecção e resoluções de conflitos diretamente no hardware. Atualmente algumas ferramentas oferecem recursos para utilização de Memória Transacional em forma de bibliotecas, contudo, não estão diretamente vinculadas as ferramentas de programação multithread. Com essa melhora de desempenho oferecida pelo HTM e não integração do HTM a interfaces de programação, este estudo tem como objetivo o desenvolvimento de uma extensão ao OpenMP para suporte a HTM, diretamente ao kernel da aplicação.*

## 1. Introdução

Memória transacional (TM) é um mecanismo para controle de simultaneidade que executa uma sequência de transações atomicamente. Para isso, uma implementação de TM executa transações e detecta conflitos, desfazendo efeitos causados pelas transações abortadas. É muito semelhante a transações realizadas em banco de dados, onde é necessário a validação e verificação das transações para efetivamente realizar alterações na memória manipulada [Harris et al. 2010].

Existem duas formas de trabalhar com memória transacional, Memória Transacional em Software (STM) e Memória Transacional em Hardware (HTM). O STM detecta e resolve conflitos em software, permitindo suporte para as transações ilimitadas, contudo resultando em uma alta sobrecarga de tempo de execução para resolução de conflitos. Já o HTM detecta e resolve os conflitos diretamente no hardware, resultando em uma baixa sobrecarga, contudo requer um hardware complexo, este recursos esta disponível principalmente em dispositivo que possuem suporte ao Intel Transactional Synchronization Extensions (TSX)<sup>1</sup> [Cai et al. 2021].

Com os benefícios até então conhecidos pela utilização de HTM e baixa integração deste recurso com interfaces de programação, o objetivo deste estudo é estender o estado da arte em interfaces de programação amplamente utilizadas e de Memória Transacional em Hardware. Com isso, pretende-se desenvolver uma extensão ao OpenMP, diretamente no kernel da interface, integrando a integrando-a ao HTM, facilitando a utilização deste mecanismo de controle em aplicações.

---

<sup>1</sup><https://www.intel.com.br/content/www/br/pt/support/articles/000059422/processors.html>

## 2. Estratégias e Experimentos

O ponto de partida será o estudo e implementação dos algoritmos do benchmark STAMP (*Stanford Transactional Applications for Multi-processing*) que utilizam HTM [Minh et al. 2008]. Com isso, será possível avaliar e comparar benchmarks com a utilização do HTM e STM. É importante ressaltar que outros benchmarks também serão utilizados e estudados a fim da validação da utilização do HTM.

Posteriormente, serão analisadas as bibliotecas e serviços já implementadas para utilização de TM, seja ela STM ou HTM, aplicada ao OpenMP, a fim de conhecer e verificar as soluções já disponíveis, como por exemplo, [Bonnichsen and Podobas 2015] e [Bihari et al. 2012]. Também será realizada uma comparação entre as propostas mais conhecidas, para isso, pretende-se utilizar uma aplicação real e benchmarks.

Após ser estudado os conhecimentos iniciais de benchmarks que utilizam HTM, e ser feita a comparação, as partes iniciais de conhecimento de benchmarks que utilizem HTM e a comparação de bibliotecas e serviços que aplicam o HTM para OpenMP, será realizada a implementação da extensão do OpenMP diretamente no kernel. Com isso, espera-se facilitar a utilização de HTM em aplicações que utilizem a interface multithread, principalmente OpenMP.

## 3. Trabalhos Futuros

A partir da identificação de estratégias já aplicadas ao OpenMP, e em geral levando como base a utilização de HTM aplicado ao OpenMP, se dará continuidade ao desenvolvimento e validação da extensão. Primeiramente, pretende-se desenvolver uma extensão a parte do OpenMP para realização de testes e validação. Em seguida pretende-se integrar o desenvolvimento diretamente ao kernel do OpenMP. Para utilização e validação, pretende-se utilizar base de dados ou aplicações reais que utilizem OpenMP e que possa ser aplicado HTM.

## Referências

- Bihari, B. L., Wong, M., Wang, A., de Supinski, B. R., and Chen, W. (2012). A case for including transactions in openmp ii: hardware transactional memory. In *International Workshop on OpenMP*, pages 44–58. Springer.
- Bonnichsen, L. and Podobas, A. (2015). Using transactional memory to avoid blocking in openmp synchronization directives. In *International Workshop on OpenMP*, pages 149–161. Springer.
- Cai, Z., Blackburn, S. M., and Bond, M. D. (2021). Understanding and utilizing hardware transactional memory capacity. In *Proceedings of the 2021 ACM SIGPLAN International Symposium on Memory Management*, pages 1–14.
- Harris, T., Larus, J., and Rajwar, R. (2010). Transactional memory. *Synthesis Lectures on Computer Architecture*, 5(1):1–263.
- Minh, C. C., Chung, J., Kozyrakis, C., and Olukotun, K. (2008). Stamp: Stanford transactional applications for multi-processing. In *2008 IEEE International Symposium on Workload Characterization*, pages 35–46. IEEE.