

# Avaliação de desempenho do sistema de memória heterogênea da arquitetura Intel *Knights Landing* (KNL)

Jefferson Fialho<sup>1,2</sup>, Silvio Stanzani<sup>1</sup>, Raphael Cóbe<sup>1</sup>, Rogério Iope<sup>1</sup>, Igor Freitas<sup>3</sup>

<sup>1</sup>Núcleo de Computação Científica – Universidade Estadual Paulista (UNESP)  
São Paulo – SP – Brasil

<sup>2</sup>Graduando do Curso de Análise e Desenvolvimento na FATEC-SP  
São Paulo – SP – Brasil

<sup>3</sup>Intel - Software and Services Group  
São Paulo – SP – Brasil

{jfialho, silvio, rmcoobe, rogerio}@ncc.unesp.br, igor.freitas@intel.com

**Abstract.** *We present an evaluation of the heterogeneous memory system of the Intel Xeon Phi KNL architecture, using applications with different characteristics. Applications that perform many data transfer operations from/to to main memory, when associated with the efficient use of cache memory, are best candidates to obtain performance gains when mapping data structures to the high bandwidth memory unit.*

**Resumo.** *Neste artigo é apresentada uma avaliação do sistema de memória heterogênea da arquitetura Intel Xeon Phi KNL, usando aplicações com diferentes características. Aplicações que realizam muitas operações de transferências de dados de e para a memória principal, quando associadas ao uso eficiente de memória cache, são fortes candidatas a terem ganhos de desempenho ao mapearem estruturas de dados para a unidade de memória de grande largura de banda.*

## 1. Introdução

Melhorias constantes em sistemas de memória de múltiplos níveis é uma tendência observada no projeto de arquiteturas computacionais atuais. A arquitetura de processadores Intel Xeon Phi de segunda geração, também conhecida como *Knights Landing* (KNL), introduziu um novo nível de memória, o *Multi-Channel Dynamic Random Access Memory* (MCDRAM), que pode ser usado como uma memória cache (*cache mode*), como uma memória endereçável com grande largura de banda (*flat mode*), ou como uma combinação de ambos (*hybrid mode*) [Sodani, 2016]. Um dos desafios encontrados no uso de sistemas de memória heterogênea é como mapear uma aplicação para as diferentes unidades de memória, de modo a obter ganhos de desempenho [Li, 2016].

Este artigo apresenta uma avaliação do uso da MCDRAM no KNL, comparando o tempo de execução e comportamento quanto ao uso do sistema de memória de aplicações com diferentes características.

## 2. Organização do processador Intel KNL

Na arquitetura Intel KNL os núcleos de processamento são organizados em pares chamados *tiles*. Cada unidade de processamento possui um nível de cache L1 e compartilha o cache L2 com o outro núcleo que compõe o *tile*. [Sodani, 2016].

Os *tiles* são organizados de acordo com os seguintes padrões de agrupamento, identificados nesta arquitetura como *cluster modes*:

- *All-to-All*: Os *tiles* não possuem subdivisão e os endereços de memória são distribuídos uniformemente entre eles. Normalmente esse modo deve ser usado apenas para depuração (*debugging*).
- Quadrante / Hemisfério: No modo quadrante, os *tiles* são divididos em 4 partes, cada uma local ao seu respectivo controlador de memória. O modo hemisfério funciona da mesma forma, com a diferença que é subdividido em 2 partes.
- SNC-4 / SNC-2: Semelhante ao Quadrante / Hemisfério, os modos SNC-4 / SNC-2 também subdividem o total de *tiles* em 4 ou 2 partes. Nestes modos, cada subdivisão é vista pelo sistema operacional como um nó independente do tipo Non-Uniform Memory Access (NUMA).

O subsistema de memória de sistemas baseados no processador KNL consiste de uma memória principal *Random Access Memory* (RAM) e uma unidade de memória adicional chamada MCDRAM, que pode ser configurada como um último nível de cache (chamado modo *cache*) ou como uma memória endereçável (chamado modo *flat*) [figura 1]. A memória MCDRAM possui uma capacidade bem menor que a DRAM, porém possui largura de banda cerca de quatro vezes maior. Neste sentido, o uso da MCDRAM pode trazer ganhos de desempenho, armazenando estruturas de dados que sejam acessadas por códigos que exijam grande uso de largura de banda [Li, 2016].

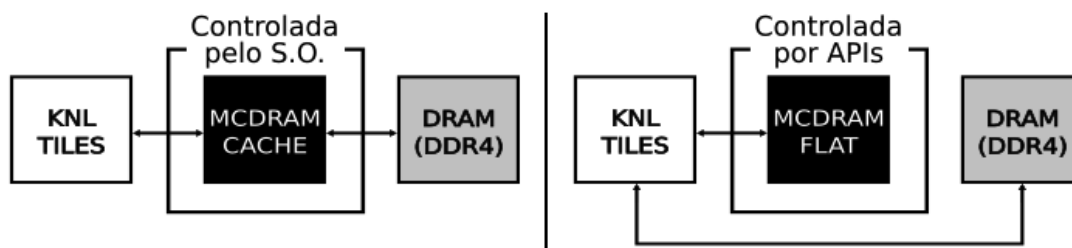


Figura 1: Modos de uso da MCDRAM

## 3. Avaliando o Uso da MCDRAM

Para avaliar o impacto da MCDRAM no desempenho da execução das aplicações, foi escolhido um conjunto de aplicações otimizadas para arquitetura Intel Xeon, que foram compiladas no processador KNL e mapeadas para os dois tipos de memória, sem alterações no código original.

### 3.1. Ambiente de Teste

O ambiente de teste é composto por um servidor *multicore* Intel Xeon e um

servidor *manycore* KNL, cada um com as seguintes configurações:

- Xeon: dois processadores E5-2699v3 @ 2.3 GHz, cada um com 18 núcleos físicos, executando duas *threads* por núcleo e 128 GB de memória principal.
- KNL: processador Intel(R) Xeon Phi(TM) CPU 7250 @ 1.40GHz, composto por 68 núcleos físicos, executando quatro *threads* por núcleo e 192 GB de memória principal. Os núcleos estão configurados no modo SNC-4 e a MCDRAM no modo *flat*. Tais configurações foram escolhidas, pois no modo *flat* é possível acessar a MCDRAM sem concorrência com o SO, permitindo assim o uso total da memória pela aplicação. A modalidade SNC-4 foi adotada em razão do controle e segmentação de memória propiciado pelas configurações NUMA.

Para obter os dados das aplicações durante suas execuções, utilizou-se o *profiler* Intel VTune Amplifier XE 2017.

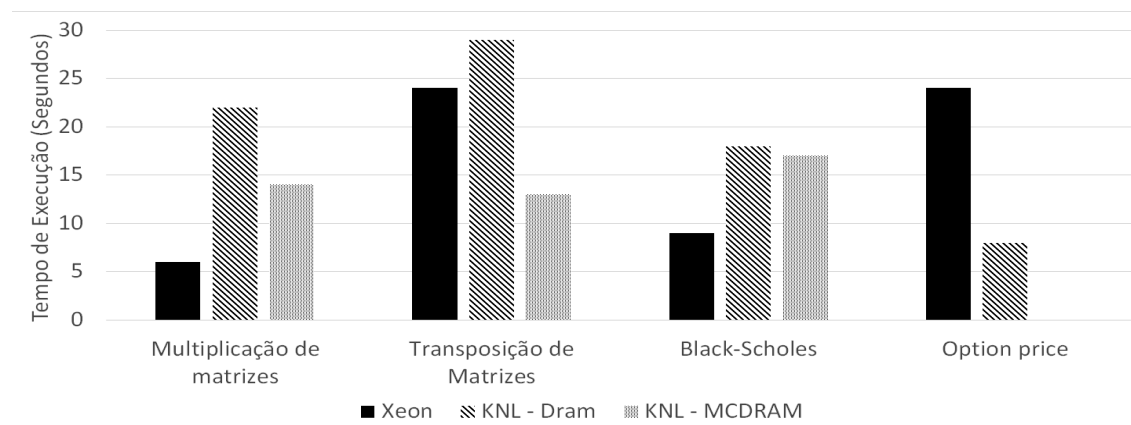
### 3.2. Carga de Entrada de Testes

Para essa avaliação foi utilizado o seguinte conjunto de aplicações: Multiplicação de Matrizes, Transposição de Matrizes [Vladimirov, 2015], *Black-Scholes* [Meyerov, 2015] e *Option Price* [Li, 2015]. Para cada aplicação foi calculada a média do uso de largura de banda [Tabela 1], utilizando todos os núcleos de sua arquitetura durante a execução.

| Aplicações                | Média de Uso de Largura de Banda (GB/s) |
|---------------------------|---|
| <i>Black Scholes</i>      | 13                                      |
| Multiplicação de Matrizes | 16                                      |
| <i>Option Price</i>       | 16                                      |
| Transposição de Matrizes  | 52                                      |

**Tabela 1: Caracterização do uso de largura de banda usando Intel VTune**

Após a caracterização, cada aplicação foi executada no servidor Xeon, no servidor KNL usando DRAM e no mesmo servidor KNL usando MCDRAM [Figura 2]. O tempo de execução da aplicação *Option Price* no KNL usando MCDRAM foi omitido, pois ficou muito mais alto que os demais tempos de execução.



**Figura 2: Avaliação do tempo de Execução de aplicações no Xeon e no KNL**

As aplicações Transposição de Matrizes e *Options Price* tiveram desempenho superior no KNL comparado com o Xeon, e os seguintes aspectos foram observados:

- As duas aplicações conseguiram utilizar a memória cache de modo mais eficiente no KNL do que no Xeon.
- No caso da aplicação *Option Price*, o desempenho da memória DRAM foi superior ao apresentado pela MCDRAM, porque houve um intenso uso de *swap* durante a utilização da MCDRAM.

As aplicações *Black-scholes* e Multiplicação de Matrizes tiveram desempenho inferior no KNL quando comparado com o desempenho apresentado no Xeon, e os seguintes aspectos foram observados:

- O uso da memória cache foi mais ineficiente no KNL do que no Xeon, o que provocou muitos atrasos.
- Ao comparar o tempo de execução das duas aplicações no KNL, é possível verificar que o uso de MCDRAM traz um ganho de desempenho nas duas aplicações. Nesse caso, comprovamos a eficácia do uso de memória com maior largura de banda.

#### 4. Conclusão

Nesse artigo foi apresentada uma avaliação do sistema de memória heterogênea do processador Intel KNL. Para isso foram escolhidas quatro aplicações, que foram executadas em processadores Intel Xeon e no processador KNL, visando a comparação de desempenho. A partir dessa comparação identificou-se que, nos casos em que há um grande uso de banda de memória e uso eficiente de unidade de memória cache, o novo processador Intel Xeon Phi KNL pode apresentar um desempenho superior ao apresentado por processadores Intel Xeon.

#### Referências

- A. Sodani *et al.*, "Knights Landing: Second-Generation Intel Xeon Phi Product," in *IEEE Micro*, vol. 36, no. 2, pp. 34-46, Mar.-Apr. 2016
- S. Li, K. Raman and R. Sasanka, "Enhancing application performance using heterogeneous memory architectures on a many-core platform," *2016 Conference on High Performance Computing & Simulation (HPCS)*, 2016, pp. 1035-1042.
- Andrey Vladimirov, Chapter 24 - Profiling-Guided Optimization, In *High Performance Parallelism Pearls*, edited by James Reinders and Jim Jeffers, Morgan Kaufmann, Boston, 2015, Pages 397-423, ISBN 9780128021187.
- Iosif Meyerov, Alexander Sysoyev, Nikita Astafiev and Ilya Burylov, Chapter 19 - Performance Optimization of Black-Scholes Pricing, In *High Performance Parallelism Pearls*, edited by James Reinders and Jim Jeffers, Morgan Kaufmann, Boston, 2015, Pages 319-340, ISBN 9780128021187.
- Shuo Li, Chapter 8 - Parallel Numerical Methods in Finance, In *High Performance Parallelism Pearls*, Morgan Kaufmann, Boston, 2015, Pages 113-137, ISBN 9780128038192.