KNC e KNL:
Comparando a Performance de Caches

Rafael Pierri[1], **Liria Matsumoto Sato**[1], **Calebe de Paula Bianchini**[2]

[1]Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica – Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto, Tv. 3, 158 – 05.508-010 – São Paulo – Brasil

[2]Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (Mackenzie)
Rua da Consolação, 896 – 01.302-907 – São Paulo – Brasil

{rafael.pierri,liria.sato}@usp.br, calebe.bianchini@mackenzie.br

Resumo. *Para determinar se um programa tem um bom desempenho, é fundamental conhecer os limites teóricos da arquitetura a qual ele se destina. Este trabalho explora a capacidade de latência e largura de banda das gerações KNC e KNL da arquitetura Xeon Phi™, com a execução de dois micro benchmarks. Em média, nos experiementos executados, o KNL demonstrou uma largura de banda 446% superior e uma latência 33% inferior em relação ao KNC.*

1. Introdução

Ao longo do tempo, processadores mais modernos aumentaram a vazão de consumo de dados, muito além da evolução das memórias DRAM. Dentro deste contexto, se os processadores só fossem capazes de solicitar dados à DRAM, a maior parte do tempo de execução de um programa seria gasto em esperar os dados retornarem da memória.

Para contornar este problema, os fabricantes adotaram uma organização de *cache* de vários níveis. Os níveis mais baixos, mais próximos do processador, têm latência e espaço menores, enquanto os níveis mais altos, mais distantes do processador, possuem latência e espaço maiores. Os dados entre os níveis de *cache* e memória DRAM são transferidos em lotes, de forma que o acesso a dados pertencentes ao mesmo lote previne que novas cargas sejam solicitadas aos níveis mais altos de *cache* ou à memória [Patterson e Hennessy 2013].

Um projeto de algoritmo com boa localidade de dados, consequentemente com maior uso do *cache*, apresentará um desempenho superior evitando que o processador fique ocioso na espera pela transferência dos dados. Assim, para avaliar o desempenho desse algoritmo, é necessário conhecer o limite teórico de desempenho da memória *cache* na arquitetura escolhida. Nesse sentido, este trabalho apresenta os resultados da execução de um conjunto de *micro benchmarks* para *cache* L1 nas famílias de processadores Intel® Xeon Phi™.

2. Arquitetura da família Xeon Phi™

A família de processadores e coprocessadores Xeon Phi™ sofreram mudanças ao longo de suas gerações, principalmente nos tamanhos dos *caches* e em novos mecanismos para

compartilhamento de memória entre os núcleos. Estas mudanças sugerem um esforço para aumentar a capacidade de acesso, no que tange largura de banda e latência de *cache*. A seguir apresentamos as duas gerações do Xeon Phi™ e suas principais diferenças.

2.1. KNC

O KNC (*knights corner*), primeira geração da família Xeon Phi™, é um coprocessador acoplado ao encaixe PCI-e. Ele possui seu próprio sistema operacional embutido e o processamento pode lhe ser delegado pela rede virtual entre o hospedeiro e o coprocessador. Seus mais de 50 núcleos operam a 1GHz de *clock* e são baseados na arquitetura x86 com suporte a instruções 64 *bits*. Seu conjunto de instruções conta com uma extensão AVX-512, que contém instruções SIMD de 512 bits. Cada núcleo possui 32KB de *cache* L1, 512KB de *cache* L2 e 4 *hardware threads*. O conjunto de *caches* L2 é conectado por um anel bidirecional permitindo o compartilhamento do *cache* entre os núcleos [Jeffers e Reinders 2013].

2.2. KNL

A geração KNL (*knights landing*), a depender do modelo, passou a oferecer o produto como um processador ou coprocessador. Em sua arquitetura, houveram mudanças significativas em relação à geração anterior. Manteve-se a compatibilidade com o conjunto de instruções, tornando os programas desenvolvidos para o KNC compatíveis, mas aumentando a compatibilidade com a extensão AVX-512. O KNL está organizado em *tiles*, sendo que cada *tile* contém dois núcleos, duas unidades de processamento vetorial para as instruções SIMD/AVX-512, 32KB de *cache* L1 e 1MB de *cache* L2. Os 38 *tiles* são conectados por uma malha bidimencional de interconexão de *cache* [Jeffers et al. 2016].

3. Benchmarks

Para avaliarmos a performance das duas gerações, utilizamos dois programas de *benchmark*. O primeiro, nomeado micbench[1], é uma adaptação do *lmbench* [McVoy e Staelin 1996] para fazer uso de instruções SIMD nos testes largura de banda. O segundo, conhecido como MIC-Meter [Fang et al. 2014], é um benchmark especializado para a família Xeon Phi™ e foi tomado para fins de referência.

O teste de largura de banda, presente em ambos programas, mensura a razão entre quantidade de dados processados e tempo decorrido dos produtos escalares **scale1** ($O[i] = a * A[i]$) e **scale2** ($O[i] = a * O[i]$). O teste de latência, apenas executado com o *micbench*, acessa diferentes posições de um vetor de 1MB e mede o tempo decorrido para acesso ao dado. Todos os testes foram realizados com apenas uma única *thread*, em apenas um dos núcleos do processador.

Ambos *benchmarks* foram compilados utilizando o ICC (*Intel® C++ Compiler*) versão 17.0.1, com *prefetch* de *software* desabilitado. O modelo de referência para o KNC foi o Intel® Xeon Phi™ Coprocessor 5110P e para o KNL foi o Intel® Xeon Phi™ 7250.

4. Experimentos

4.1. Latência

Quando o acesso à memória se restringe aos primeiros 32KB do vetor, por conseguinte contidos no *cache* L1, observa-se uma latência média de 2,9 e 3,0 nanosegundos para o

[1]Disponível em: https://bitbucket.org/pierri/micbench

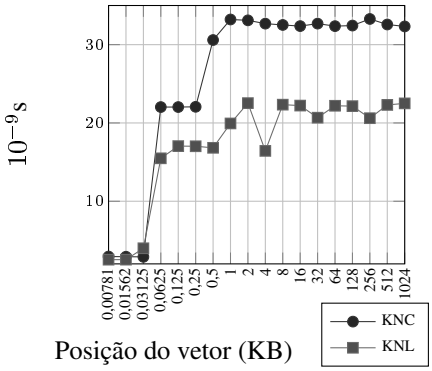

Figura 1. Latência em diferentes posições do vetor

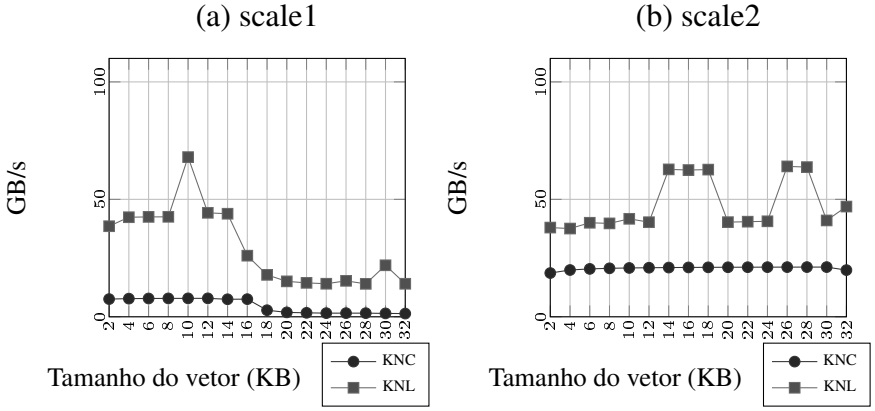

Figura 2. Largura de banda - *micbench*

KNC e o KNL, respectivamente.

Conforme mostra a Figura 1, os acessos feitos em regiões além do *cache* L1 duraram em média 30,4ns para o KNC e 20,0ns para o KNL, corroborando com os resultados obtidos em experimentos semelhantes [Rahman 2013]. Não é possível observar a diferença de acesso entre o *cache* L2 e a memória em virtude do *prefetch* de *hardware* entre ambos [Li 2014].

4.2. Largura de Banda

Os algoritmos **scale1** e **scale2** do *micbench* obtiveram uma largura de banda média de 4,7GB/s e 20,6GB/s no KNC, conforme mostra a Figura 2. Já no KNL a mesma implementação obteve as marcas de 29,6GB/s para o **scale1** e 47,6GB/s para o **scale2**.

A implementação do *mic-meter* obteve a largura de banda média de 32,4GB/s para o **scale1** e 11,7GB/s para o **scale2** no KNC. No KNL obteve 78,7GB/s e 80,0GB/s respectivamente, conforme mostra a Figura 3.

Apesar dos algoritmos serem os mesmos, em todas as medições o *mic-meter* obteve um desempenho superior ao *micbench*. Para o desempenho se equiparar seriam necessárias otimizações de desempenho, como desdobramento de laços e aquecimento de *cache*, ambas presentes apenas no *mic-meter*. Quanto a diferença entre as gerações do Xeon Phi™, o KNL também obteve desempenho superior em todas as medições, eviden-

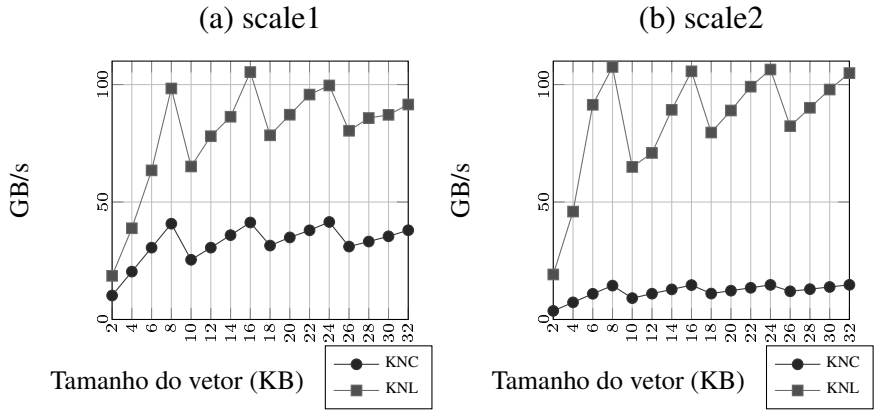

Figura 3. Largura de banda - *mic-meter*

ciando uma largura de banda superior ao seu predecessor.

5. Conclusão

Em todos experimentos o KNL[2] mostrou um desempenho superior no acesso ao *cache*, confirmando o esforço do fabricante em melhorar o desempenho quanto a latência e largura de banda. Nesse sentido, considerando um programa que faz uso intenso de *cache* L1, pode-se esperar um menor tempo de execução no KNL.

Os próximos experimentos considerarão não apenas um maior número de *threads* para benchmaks de cache L1, mas também explorar o barramento circular e a malha de interconexão para acesso de cache L2 do KNC e do KNL, respectivamente.

Referências

Fang, J., Sips, H., Zhang, L., Xu, C., Che, Y., e Varbanescu, A. L. (2014). Test-driving intel xeon phi. In *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering*, ICPE '14, pages 137–148, New York, NY, USA. ACM.

Jeffers, J. e Reinders, J. (2013). *Intel Xeon Phi Coprocessor High Performance Programming*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.

Jeffers, J., Reinders, J., e Sodani, A. (2016). *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition*. Elsevier Science, 2nd edition.

Li, S. (2014). Memory management for optimal performance on intel xeon phi coprocessor: Alignment and prefetching. Intel. Acessado em: 2017-02-06.

McVoy, L. e Staelin, C. (1996). Lmbench: Portable tools for performance analysis. In *Proceedings of the 1996 Annual Conference on USENIX Annual Technical Conference*, ATEC '96, pages 23–23, Berkeley, CA, USA. USENIX Association.

Patterson, D. A. e Hennessy, J. L. (2013). *Computer Organization and Design, Fifth Edition: The Hardware/Software Interface*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition.

Rahman, M. R. (2013). Intel xeon phi core micro-architecture. Intel. Acessado em: 2017-02-06.

[2]Oferecido pela Colfax International, disponível em https://colfaxresearch.com/remote-access/