Impacto da Largura do Vetor de Instruções SIMD em Arquiteturas de Processamento Próximo à Memória

Sairo Santos[1], **Marco Antonio Zanata Alves**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba/PR

{srsantos, mazalves}@inf.ufpr.br

Resumo. *Instruções Single Instruction Multiple Data (SIMD) processam múltiplos elementos de dados por vez e podem ser usadas para explorar o paralelismo no acesso aos dados oferecido pelas tecnologias de memória principal. Neste trabalho, discute-se como a largura do vetor usado nessas instruções impacta o desempenho em arquiteturas de processamento próximo à memória.*

1. Introdução

Instruções SIMD, ou instruções vetoriais, oferecem desempenho ao acessar e processar múltiplos elementos de dados por vez usando unidades funcionais vetoriais. Memórias 3D oferecem maior paralelismo no acesso aos dados armazenados devido a sua organização física e lógica, e instruções SIMD podem ser usadas para explorar tal paralelismo, extraindo desempenho. Neste trabalho, observa-se o impacto da largura de vetor usado por instruções vetoriais em uma arquitetura de processamento próximo à memória com base em dados de simulação.

2. Experimentos e Discussão

Os experimentos utilizaram o simulador SiNUCA [Alves et al. 2015], simulando uma arquitetura baseada no processador Intel Skylake. A aplicação simulada percorre a memória sequencialmente, armazenando o mesmo valor em todos os elementos de um *workload*. A comparação contempla três situações diferentes: (i) uma arquitetura de processamento próximo à memória usando instruções SIMD com vetor de 8 KB, (ii) uma arquitetura idêntica à anterior com instruções SIMD de 256 B, e (iii) uma arquitetura x86 tradicional com instruções SIMD AVX-512 da Intel usando dezesseis threads.

A arquitetura com processamento próximo à memória usa um elemento de processamento localizado na camada lógica de uma memória 3D, tendo acesso direto aos 32 *vaults* independentes do dispositivo. Os tamanhos dos vetores usados consideram o *row buffer* de 256 B da memória 3D: instruções com vetor de 256 B acessam um *vault* da memória, enquanto instruções com vetor de 8 KB acessam todos os *vaults*, assim melhor explorando o paralelismo que o dispositivo oferece [Santos et al. 2015]. Os resultados mostrados na Figura 1 se referem (a) ao tempo de execução e (b) à vazão de dados.

Como mostra a Figura 1, o desempenho da arquitetura com instruções SIMD com vetor de 8 KB é superior em ambas as métricas, sugerindo quanto desempenho pode-se extrair ao explorar o paralelismo interno da memória. Atingindo uma vazão de 85GB/s e processando 8 KB por instrução, a arquitetura com processamento próximo à memória tem resultados expressivos: mesmo comparado com a arquitetura tradicional com dezesseis *threads* simultâneos, seu desempenho é superior.

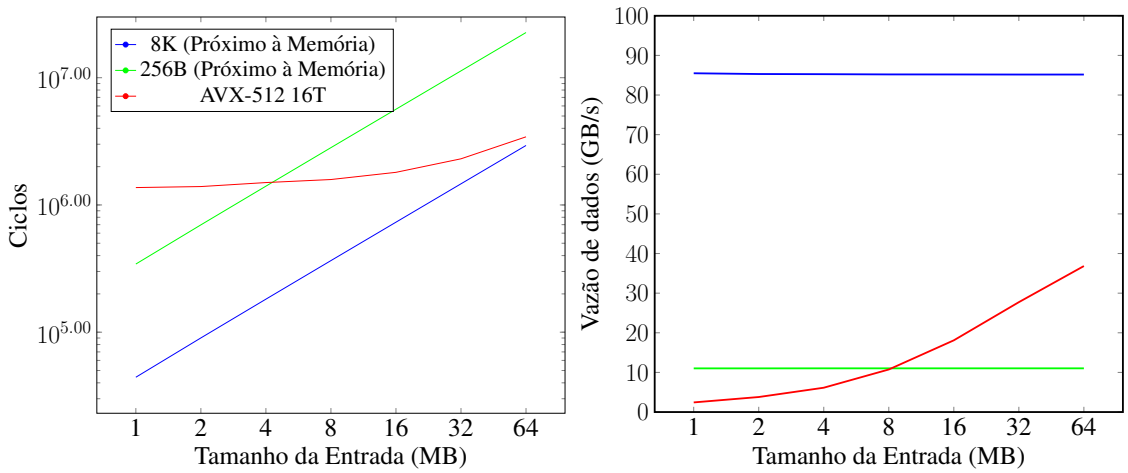

Figura 1. (a) Tempo de execução (em ciclos, quanto menos melhor), (b) vazão de dados (em GB/s, quanto mais melhor)

Por outro lado, os resultados da mesma arquitetura usando vetores de 256 B são os piores dentre os experimentos a partir do *workload* de 8 MB. Apesar da largura de vetor usado pelas instruções AVX-512 na arquitetura x86 ser muito menor, 512 B, arquiteturas tradicionais modernas utilizam-se de técnicas como *multithreading*, execução fora de ordem e paralelismo a nível de instrução, extraindo desempenho da memória. Assim, apesar do tamanho de vetor superior, a arquitetura com processamento próximo à memória é incapaz de melhorar o desempenho.

Deve-se considerar também o quanto o tamanho do vetor pode limitar o uso dos recursos por parte das aplicações. O desempenho com vetores de 8 KB foi superior nos experimentos discutidos neste trabalho, mas um tamanho tão grande pode restringir seu uso por aplicações que não necessariamente fazem acesso sequencial aos dados ou que consideram *workloads* muito menores. É necessário buscar um tamanho de vetor ótimo considerando o *trade-off* entre desempenho e acessibilidade para o uso da arquitetura.

3. Conclusões

Instruções SIMD podem ser usadas por arquiteturas de processamento próximo à memória para explorar um dos principais benefícios desse tipo de arquitetura, o maior paralelismo no acesso ao dados. Os experimentos com um vetor que acessa todos os *vaults* da memória 3D obtiveram resultados expressivos tanto em relação ao tempo de execução quanto à vazão de dados, enquanto vetores que acessam um *vault* por vez tiveram um desempenho insatisfatório. A largura do vetor usado impacta tanto a melhoria de desempenho que poderá ser alcançada quanto a acessibilidade da arquitetura por parte de aplicações, devendo ser selecionada cuidadosamente.

Referências

Alves, M. A. Z. et al. (2015). Sinuca: A validated micro-architecture simulator. In *Int. Conf. on High Performance Computing and Communications*.

Santos, P. C., Alves, M. A., and Carro, L. (2015). Hmc and ddr performance trade-offs. In *International Embedded Systems Symposium*, pages 159–171. Springer.