Comparação de desempenho de rotinas de multiplicação de matrizes densas em arquiteturas multi-core e many-core

Mateus Silva de Melo[1], Roberto Pinto Souto[2], Icaro Fontes Moreira de Castro[3]

[1]Bolsa de Iniciação Tecnológica – PIBITI/LNCC/Universidade Estácio de Sá (Petrópolis-RJ);

[2]Coordenação de Sistemas e Redes – CSR/LNCC;

[3]Programa de Estágio - LNCC/Universidade Veiga de Almeida (Rio de Janeiro-RJ)

{msmelo, rpsouto, icarofmc}@lncc.br

1. Introdução

As aplicações científicas geralmente utilizam operações de álgebra linear em seu código-fonte, como soma e multiplicação de matrizes, resolução de sistema de equações lineares entre outros. Por esse motivo e pelo grande número de operações que as aplicações científicas devem realizar, é comum vermos bibliotecas especializadas em álgebra linear sendo otimizadas para serem executas em determinadas arquiteturas paralelas. Esse estudo pretende mostrar, entre as arquiteturas multi-core (CPU) e many-core (MIC e GPU), qual obtém o maior desempenho nos testes de multiplicação de matrizes densas.

2. Ambiente Computacional dos Experimentos

Para a realização dos testes, foram utilizados três arquiteturas paralelas: CPU (*central processing unit*), do tipo multi-core, no qual utilizou-se dois processadores Intel Xeon E5-2650v2, totalizando 16 núcleos. MIC (*many integrated core*), do tipo many-core, que foi utilizado o coprocessador Intel Xeon Phi com 61 núcleos e 4 threads por núcleo, somando 244 threads no total. GPU, também do tipo many-core, utilizando o dispositivo NVIDIA Tesla K40 com 2.880 núcleos.

Também foram usadas duas bibliotecas de álgebra linear, otimizadas para cada arquitetura. Da biblioteca Intel MKL (*Intel Math Kernel Library*), empregando-se a rotima SGEMM, a qual foi executada na CPU e na MIC. Na GPU, utilizou-se a rotina cuBLAS SGEMM, nativa da biblioteca NVIDIA cuBLAS. Nas rodadas com MIC e GPU, foi utilizada estratégia *offload* de execução, ou seja, os dados precisaram ser tranferidos da CPU para estas arquiteturas, e vice-versa.

3. Métricas de avaliação

As matrizes utilizas neste estudo comparativo, foram matrizes densas quadradas (N x N), sendo N igual a 4.096, 8.192, 16.384 e 20.480. Os tempos de execução foram medidos na CPU, para rodadas com 1, 2, 4, 8 e 16 threads, e na MIC com 8, 16, 32, 64, 128, 240 threads. A configuração da rodada paralela em GPU (configuração dos blocos de threads, ocupância, uso de memória compartilhada e de registradores) é determinada internamente pela rotina da biblioteca cuBLAS, sendo transparente para quem a utiliza.

Foram adotadas algumas métricas para avaliar os resultados. O tempo de referência (T), é o tempo de execução em um núcleo computacional de CPU, que no caso da MIC, foi neste trabalho definido com sendo o tempo de execução em 8 núcleos. Tempo

paralelo (*Tp*), é o tempo de execução com p núcleos computacionais. Speed-up (*Sp*), é o ganho de desempenho obtido em p núcleos computacionais, sendo a razão entre o tempo de referência e o tempo paralelo. Eficiência (*Ep*), é definida pela razão entre o ganho de desempenho e o números de núcleos computacionais utilizados.

4. Resultados e comentários

Realizados os testes, percebe-se conforme a Figura 1, que as arquiteturas many-core (MIC e GPU) obtiveram um ganho de desempenho considerável em relação a arquitetura multi-core (CPU). Porém, levando em conta o tempo de transferência de dados, houve uma redução deste ganho, destacando a MIC, onde em alguns testes o seu ganho de desempenho foi anulado.

Foi também observado que a escalabilidade depende do tamanho do problema. Quanto maior for a matriz, melhor foi a escalabilidade alcançada. Por exemplo, para a Xeon Phi (MIC), foi obtida eficiência paralela sempre superior a 90%, para uma matriz de tamanho 20.480. Por outro lado, para matrizes de tamanho 16384, 8192 e 4096, a eficiência alcançada utilizando-se 240 threads foi de 87%, de 84% e de 28%, respectivamente. Ao se levar em conta o tempo de transferência dos dados, a eficiência decai consideravelmente.

A GPU, que usa a rotina cuBLAS SGEMM, foi a que obteve o melhor desempenho para todos os tamanhos das matrizes.

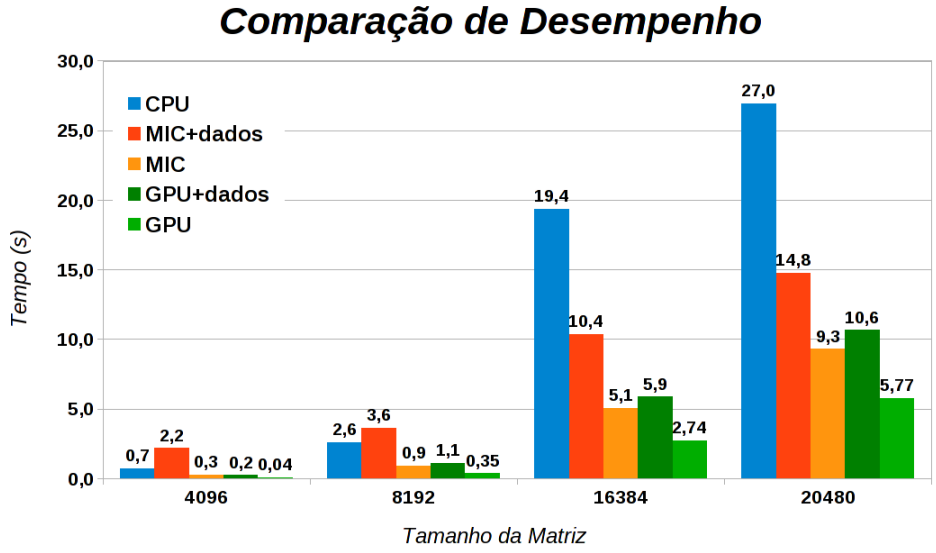

Figura 1. Comparativo de desempenho entre as arquiteturas: CPU (16 threads), MIC (240 threads) e GPU.

5. Referências

[1] NVIDIA. CUBLAS Library User Guide.

[2] Barth, Michaela; Byckling, Mikko; Ilieva, Nevena; Saarinen, Sami; Schliephake, Michael. Best Practices Guide Intel Xeon Phi v1.1. Disponível em http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Intel-Xeon-Phi.pdf .