Impacto da Precisão Reduzida e Mista na Computação do Método de Lattice-Boltzmann em Múltiplas GPUs*

Gabriel Freytag[1], João V. F. Lima[2], Paolo Rech[1], Philippe O. A. Navaux[1]

[1] Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre – RS – Brasil

[2]Universidade Federal de Santa Maria (UFSM) – Santa Maria – RS – Brasil

{gfreytag,prech,navaux}@inf.ufrgs.br, jvlima@inf.ufsm.br

Resumo. *A heterogeneidade de arquiteturas de computação modernas permite que engenheiros refinem algoritmos para maximizar o nível de afinidade com a arquitetura e, consequentemente, a eficácia da computação. Neste trabalho, investigamos o impacto da precisão reduzida e mista na computação do método de Lattice-Boltzmann em uma plataforma multi-GPU. Usando meia precisão para o armazenamento e simples para as operações aritméticas, obteve-se um speedup de 3.44 consumindo 71% menos energia e uma perda 0.02% de acurácia.*

1. Introdução

Os Sistemas de Computação de Alto Desempenho (HPC) atualmente conectam um grande número de dispositivos com arquiteturas distintas que, por si só, estão tornando-se heterogêneas. No entanto, ao mesmo tempo em que a heterogeneidade dos sistemas aumenta para suportar diferentes características computacionais, a demanda por recursos computacionais excede a disponibilidade atual. Embora alguns problemas exijam resultados exatos, outros admitem resultados aproximados, reduzindo os requisitos de recursos e, consequentemente, melhorando o desempenho. Uma das estratégias mais prevalentes é o dimensionamento de precisão [Mittal 2016], que explora a precisão mista. Assim, é possível compensar a qualidade do resultado com desempenho e eficiência energética.

Neste artigo, avaliamos o impacto da precisão reduzida e mista no desempenho, eficiência energética e precisão da computação estêncil simultaneamente em 4 GPUs NVIDIA P100 com suporte nativo a precisão mista. Como estudo de caso, usamos a amplamente conhecida aplicação de Dinâmica de Fluídos Computacional (CFD) denominada método de Lattice-Boltzmann (LBM) com um modelo tridimensional D3Q19.

2. Implementações

Neste trabalho, foram implementadas três versões distintas do LBM. Na primeira, *multigpu*, as operações aritméticas de ponto flutuante e o armazenamento dos dados na memória utilizam a mesma precisão, selecionada em tempo de compilação. Esta versão executa todas as operações aritméticas com funções matemáticas regulares do CUDA. Já a segunda implementação, *multigpu_float*, executa todas as operações aritméticas de ponto flutuante com precisão simples, usando funções matemáticas intrínsecas do CUDA, e a precisão do armazenamento dos dados é definida em tempo de compilação em dupla, simples ou meia. Por fim, a terceira implementação, *multigpu_half2*, realiza duas

*Apoio financeiro CAPES.

Tabela 1. Resultados experimentais em um modelo 3D de tamanho $352 \times 352 \times 352$.

Implementação	FP	Execução (s)	*Speedup*	Erro FP (%)	Energia (kWh)
multigpu	double	19.47	1.00	0.00	0.3919
	single	11.16	1.75	0.00	0.1418
	half	7.67	2.54	0.22	0.0648
multigpu_float	double	16.81	1.16	0.00	0.2652
	single	8.52	2.28	0.00	0.0734
	half	5.66	3.44	0.02	0.0326
multigpu_half2	double	17.37	1.12	0.06	0.2789
	single	8.72	2.23	0.06	0.0763
	half	5.85	3.33	0.06	0.0354

operações aritméticas de meia precisão em paralelo em uma única Unidade de Ponto Flutuante (FPU) de precisão simples utilizando um tipo de dado vetorizado (*half2*) e funções matemáticas intrínsecas do CUDA, ao invés das funções matemáticas regulares.

3. Resultados

Na Tabela 1, é possível observar o tempo de execução, o *speedup* (com base na versão multigpu utilizando precisão dupla), a perda de precisão e o consumo de energia das três implementações em uma plataforma com 4 GPUs NVIDIA Tesla P100. A medida que reduzimos a precisão das operações aritméticas, o tempo de execução e o consumo de energia reduzem significativamente. Com meia precisão ao invés de precisão dupla, há um *speedup* de 2.54 e uma redução no consumo de energia de 58% com a mesma precisão para o armazenamento dos dados (*multigpu*). Já ao fixar a precisão das operações aritméticas em simples e utilizar funções matemáticas intrínsecas ao invés de funções regulares, é possível obter um *speedup* de 3.44 e reduzir o consumo de energia em 71.4% usando meia precisão para o armazenamento dos dados (*multigpu_float*). Por fim, ao fixar a precisão das operações aritméticas em meia, utilizando dados vetorizados (*half2*) e funções matemáticas intrínsecas (*multigpu_half2*), o desempenho se manteve significativamente próximo à implementação *multigpu_float* devido à falta de afinidade dos procedimentos do método com operações vetoriais, necessitando reestruturação total das mesmas.

4. Conclusão e Trabalhos Futuros

No presente trabalho, mostramos que é possível obter ganhos significativos de desempenho e consumo de energia ao fixar a precisão das operações aritméticas em simples e reduzir a precisão do armazenamento em meia precisão. Como trabalhos futuros, sugerimos a otimização da versão *multigpu_half2* e a análise de outras aplicações de física.

Referências

Mittal, S. (2016). A survey of techniques for approximate computing. *ACM Computing Surveys*, 48(4).