Evento	Salão UFRGS 2016: SIC - XXVIII SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2016
Local	Campus do Vale - UFRGS
Título	Explorando a Terceira Lei de Newton para Simulação N-Corpos em uma Plataforma GPU
Autor	VINÍCIUS ALVES HERBSTRITH
Orientador	LUCAS MELLO SCHNORR

Explorando a Terceira Lei de Newton para Simulação N-Corpos em uma Plataforma GPU

Vinícius Herbstrith (bolsista), Lucas Mello Schnorr (orientador)
Instituto de Informática – Universidade Federal do Rio Grande do Sul

Simulações N-corpos são simulações de um sistema de partículas sobre a influência das forças físicas externas e forças geradas pelas outras partículas do mesmo sistema. Simulações deste tipo são largamente aplicadas em astrofísica, física de plasma e dinâmica molecular. A simulação ocorre em intervalos de tempo onde, para cada intervalo e para cada partícula, é calculada a força resultante que uma partícula sofre devido a interação com as outras. Devido a necessidade de calcular a força entre todos os pares de corpos a cada intervalo de tempo, temos uma complexidade O(N^2), onde N é a quantidade de partículas presentes no sistema. A simulação N-corpos pode ser otimizada fazendo-se uso da paralelização. Como não existem dependência de dados, é possível calcular cada corpo em paralelo, para uma complexidade de tempo O(N).

GPUs são uma arquitetura many-core de alto desempenho. Para se tirar proveito deste potencial, os algoritmos geralmente precisam ser reformulados levando em conta as peculiaridades da GPU. Uma proposta de algoritmo para simulação N-corpos é apresentada no livro GPU Gems 3, onde os autores extrairam um excelente desempenho da GPU.

Uma algoritmo N-corpos visando a otimização da paralelização foi proposta por Halpem na CppCon 2014. Ele usa a terceira lei de Newton, onde o efeito de um corpo X sobre outro corpo Y é idêntico ao efeito de Y sobre X. Reduz-se então pela metade o número de interações entre corpos. O autor propõe o uso de recursão paralela para evitar as condições de corrida do algoritmo.

O objetivo deste trabalho é a implementação deste algoritmo em uma GPU, combinando a otimização e a capacidade do paralelismo massivo da GPGPU. O desafio é evitar o problema de condições de corrida do algoritmo, que não pode ser através da recursão, que apresenta um baixo desempenho em um ambiente GPGPU. O algoritmo proposto ataca o problema de uma forma ascendente, começando a partir do que seriam as folhas de uma árvore recursiva. Foi necessário a adição de um esquema mestre-trabalhador para fazer o controle das condições de corrida, tendo em vista que a inconsistência ocorre quando dois nós de níveis diferentes da árvore recursiva são processados em paralelo.

A análise de desempenho do algoritmo foi feita com um conjunto de corpos gerados aleatoriamente. A máquina usada para os testes possui processador Intel i7-4770 CPU (3.40GHz), 8GiB DIMM DDR3(1.6GHz) e uma GPU NVIDIA GTX 760 (1152 cuda cores, 980Mhz). Como base de comparação, usamos o algoritmo N-corpos do livro GPU gems 3 com os mesmos parâmetros de entrada. Para diferentes tamanhos de entrada, foram feitas 15 execuções de forma se obter uma amostra significativa. Comparando os tempos, tivemos um resultado até seis vezes mais lento com a nossa proposta. O baixo desempenho se dá por conta do uso de mutex para realizar o controle mestre-trabalhador, um mecanismo que acaba sendo muito custoso no ambiente GPGPU, custo este mais alto que o processamento reduzido dos cálculos.

Mesmo com este resultado negativo, o algoritmo é promissor. Acreditamos que uma implementação em um ambiente em cluster possa tirar um maior proveito das melhorias do algoritmo, sem ser penalizado fortemente pela condição de corrida, onde podemos ter um esquema mestre-trabalhador para um melhor controle da condição de corrida.