

Benchmark de GPU's com simulação N-Body

...

Sumário

1. Escolha de Simulação N-Body como Benchmark
2. Simulação N-Body
3. Algoritmo All-Pairs
 - a. Função potencial
 - b. Como o programa foi paralelizado
 - c. Tile calculation
4. Cálculo de eficiência e score
5. benchmark de GPU's

Escolha de Simulação N-Body como Benchmark

Embora haja uma enorme quantidade de benchmarks para GPUs, no geral esses focam na renderização gráfica ou na habilidade individual da unidade gráfica.

Com a maior popularização do uso de GPUs para aplicações científicas, tornou-se necessário a obtenção de um novo modelo de benchmark, de forma que este apresentasse resultados mais relevantes para aplicações científicas.

A simulação N-Body foi escolhida por representar as necessidades de várias simulações. Este é um problema de ordem N^2 que utiliza padrões de acesso à memória que representam várias simulações

Simulação N-Body

Uma simulação N-Body faz aproximações na evolução de um sistema de corpos no qual cada corpo está em constante interação com todos os outros corpos.

Existem inúmeros problemas computacionais que utilizam N-Body, tais como:

- Simulações astrofísicas em que cada corpo representa uma galáxia ou uma estrela individual e os corpos se atraem por força gravitacional;
- Simulações de iluminação global;
- Turbulência no fluxo de fluidos.

Simulação N-Body

A simulação aplica a lei gravitacional de Newton em cada partícula. Assim, a força total aplicada a uma partícula é calculada somando as forças que as outras partículas aplicam nesta. Após isto utiliza-se um método apropriado de integração para calcular a variação de velocidade e de posição desta partícula ao longo de um intervalo de tempo.

$$F = G \cdot \frac{m_{g_1} \cdot m_{g_2}}{d^2}$$

Algoritmo All-Pairs

O algoritmo utilizado pelo autor do artigo é o All-Pairs, que é o mais simples para calcular a força total em cada partícula. Nele, a força de uma partícula em relação à outra é calculada uma interação por vez e as forças finais são usadas para calcular a mudança da velocidade e posição da partícula.

Função potencial

O potencial gravitacional é usado para ilustrar a forma básica de computar um N-Body em All-Pairs. Sendo N corpos com posição inicial X e velocidade V.

$$\mathbf{f}_{ij} = G \frac{m_i m_j}{\|\mathbf{r}_{ij}\|^2} \cdot \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|}$$

Onde m_i e m_j são as massas dos corpos i e j respectivamente e G é a constante gravitacional

Função potencial

Como os corpos representam galáxias, os corpos colidem-se passando um por dentro do outro. Quando os corpos colidirem, o raio da equação será zero e por isso é usado o fator de balanceamento e^2 para o denominador da equação não ser 0.

$$\mathbf{F}_i = \sum_{\substack{1 \leq j \leq N \\ j \neq i}} \mathbf{f}_{ij} = Gm_i \cdot \sum_{\substack{1 \leq j \leq N \\ j \neq i}} \frac{m_j \mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|^3}$$

Função potencial

Para integrar de acordo com o tempo, é preciso ter a aceleração $\mathbf{a}_i = \mathbf{F}_i / m_i$ para atualizar a posição e velocidade do corpo i e com isso a equação anterior é simplificada.

$$\mathbf{a}_i \approx G \cdot \sum_{1 \leq j \leq N} \frac{m_j \mathbf{r}_{ij}}{\left(\|\mathbf{r}_{ij}\|^2 + \varepsilon^2 \right)^{3/2}}$$

Como o programa foi paralelizado

Na estratégia adotada pelo autor, cada thread fica responsável por calcular a aceleração total na partícula alocada. A aplicação faz uso da memória compartilhada, uma vez que esta resulta em melhores resultados quando comparada com a memória global.

Para o uso correto da memória compartilhada, é utilizada uma estratégia de tile calculation para que os dados sejam transferidos de forma correta para a memória compartilhada.

Tile Calculation

O cálculo das tiles faz uso da memória compartilhada para aumentar a performance da simulação. As threads são alocadas em blocos de P com um total de N/p blocos. Cada bloco irá processar um tile de N/p partículas por vez, usando a memória compartilhada para reduzir a transação de memória global

Cálculo de eficiência e score

Os resultados do benchmark são obtidos após simular em 1000 iterações em quantidades N de corpos, tais que $N = [1024, 2048, 4096, 8192]$, 500 iterações para $N = 16.384$ e 250 iterações para $N = 32.768$. O tempo total realizado para cada tamanho de N é salvo e combinado para a pontuação do benchmark. A simulação irá gerar dois benchmarks, a pontuação e sua eficiência. A pontuação segue a equação(1) e a eficiência a equação(2), onde a variável C é a velocidade de clock e P o número de processadores da GPU.

$$\begin{aligned}\text{score} &= \sum_{N=1024}^{32768} \frac{N^2}{t(N)} \\ \text{efficiency} &= \frac{1}{P} \times \frac{500}{C} \times \sum_{N=1024}^{32768} \frac{N^2}{t(N)}\end{aligned}$$

Benchmark de GPU's

| Testes | GTX 970 | GT 740M |
|--------------------|--------------|--------------|
| Clock | 1114MHz | 980MHz |
| N° de Cuda Cores | 1664 | 384 |
| Score | 20496760 | 7452884,5 |
| Eficiência | 5528,619579 | 9902,323156 |
| Tempo do Benchmark | 214,352654 s | 590,641522 s |

Participantes

Gustavo Ebbo

Ivo Paiva

Rodrigo Vicente