Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Caio Costa Salgado

Uso de GPGPU na Análise de Buracos Negros

São Paulo
Dezembro de 2017

Uso de GPGPU na Análise de Buracos Negros

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Rodrigo Nemmen da Silva

São Paulo
Dezembro de 2017

Resumo

Aqui vai o resumo que ainda tem que ser feito....

Palavras-chave: GPGPU, CUDA, HPC, Monte Carlo, Transferência radioativa, Buraco Negro.

Abstract

And here will be the english abstract, that still need to be done....

Keywords: GPGPU, CUDA, HPC, Monte Carlo, Rasioactive Transfer, Black Hole.

Sumário

Lista de Abreviaturas

GPU	Unidade de processamento Gráfico (*Graphics Grocessing Unit*)
GPGPU	Unidade de processamento Gráfico de Propósito Geral (*General Purpose Graphics Processing Unit*)
CUDA	Computação em Arquitetura unificada de dispositivos (*Compute Unified Device Architecture*)
HCP	Computação de Alta Performance (*High Performance Computing*)
SIMD	Única Instrução Múltiplos dados (*Sigle Instruction Multiple Dada*)

Capítulo 1

Introdução

Uma grande dúvida dos astrofísicos e também de toda a comunidade científica é o que ocorre em um buraco negro e em suas proximidades. Na busca de respostas programas de computador são feitos com o intuito de simular essa região e talvez trazer alguma luz, um desses programas é o **grmonty**(Dolence *et al.*, 2009).

Programas dessa natureza tendem a ser muito intensos no que diz respeito ao processamento, exigindo muito das CPUs, estas tornam-se assim um limitante, um gargalo, para a velocidade com a qual o programa pode devolver um resultado. É neste contexto que buscamos aplicar métodos de *Computação de Alta Performance* para otimizar ao máximo o uso todos os dispositivos do computador (hardware) que temos disponíveis.

Muitas das técnicas de HPC exploram a paralelização, o que pode ser feito massivamente por um hardware específico as *unidades de processamento gráfico*, GPU. Tais dispositivos são muito populares e já presentes em muitas máquinas domésticas e até em smartphones, eles são confeccionados primordialmente para processamento gráfico em jogos digitais, porém graças aos avanços recentes tais dispositivos tem se tornado mais genéricos e respondendo a uma gama maior de problemas.

Ao analisar o funcionamento do **grmonty** - por sua característica de simulador de partículas - é possível classificar parte de sua execução no modelo SIMD, uma vez que simula a trajetória da cada fóton de maneira independente. Dada essa informação podemos explorar o poder computacional das GPUs afim de paralelizar a execução do código, aumentando drasticamente sua a performance.

Este trabalho tem como objetivo apresentar melhorias a execução do código do **grmonty**, utilizando-se do processador de placas gráficas, as GPGPUs, para massivamente paralelizar e distribuir a carga de trabalho pelos múltiplos núcleos de processamento destas placas. Primeiramente é explicado o que é o **grmonty** e como funciona, depois os paradigmas ao qual sua execução apoia-se, caminhando para a explicação de GPUs e como contribuem para o aumento de performance, assim são apresentadas as otimizações executadas, chegando finalmente nos resultados alcançados e conclusões. Há ainda um capítulo de próximos passos demonstrando que ainda há muito espaço para mais melhorias e mais velocidade na execução.

Capítulo 2

Grmonty: Monte Carlo para Relatividade Geral

2.1 O que Faz

Dolence et al definem o **grmonty** como "software destinado a calcular o espectro de plasmas quentes e opticamente finos a par da completa relatividade geral utilizando um código de transporte radioativo baseado na técnica de Monte Carlo"(Dolence *et al.*, 2009, p.1, traduzido). Em outras palavras o programa estima o espectro de uma simulação de magnetoidrodinâmica relativistica utilizando o método de Monte Carlo para aproximar o espectro.

Utilizando o método de Monte Carlo na geração dos dados, os fótons, e a partir de um dado modelo de plasma fornecido como entrada, o programa busca gerar o espectro de radiação. Para tanto um número próximo a N - fornecido na entrada - de fótons é gerado e para cada fóton sua tragetória é traçada. Nessa trajetória o fóton é espalhado e passa por diferentes interações, podendo ser absorvido, refletido, difratado ou até re-emitido ao percorrer seu percurso e finalmente ser medido.

Depois de algumas iterações um número próximo a N de fótons já foi gerado e rastreado, assim um relatório com o espectograma obtido é gerado e retornado pelo programa que finalmente termina.

2.2 Para que Faz

Qualquer fonte astrofísica de radiação que seja relativistica, ou seja, qualquer corpo ou fenômeno fonte de radiação eletromagnética, seja do rádio à raios gama e que é relativistica: apresenta uma considerável distorção no espaço-tempo, seja por estar em velocidades próximas a da luz, seja por possuir uma enorme quantidade de massa e/ou energia. Exemplos de objetos são os buracos negros e estrelas de neutrons, fenômenos são os *Gamma Ray Bursts* ou núcleos ativos de galáxias.

Foram desenvolvidas várias técnicas para calcular a transferência radioativa a partir de fontes como as descritas a cima(Dolence *et al.*, 2009), porém poucas levam em conta a relatividade geral como um todo, pricipalmente no quesíto de objetos, fontes, muito massivas ou com velocidades próxima a da luz, o **grmonty** vem para aprimorar esses cálculos.

2.3 Como Faz

No momento de criação e rastreio dos fótons o programa faz o uso da biblioteca **OpenMP** para paralelizar o desenvolvimento dos fótons, graças a esta abordagem é viável o potêncial uso de todos os núcleos disponíveis na CPU da máquina. A bilioteca é utilizada para que cada fóton seja produzido e espalhado de forma independente dos outros e funcionando em paralelo, além disso todas as intruções não dependem do fóton em sí, elas são as mesmas intruções para todos os fótons. Desta forma podemos caracterizar o **grmonty** como tendo uma computação SIMD.

"Única Instrução Multiplos Dados: Nesse tipo de computação podem haver multiplos processadors, cada um operando sobre seu item de dados, mas estão todos executando a mesma instrução naquele item de dados"(Victor *et al.*, 2016, p.84, traduzido)

Assim que um fóton é gerado seu percurso é traçado. O que é evidente ao se observar as linhas 106 a 137 do *grmonty.cu*, aqui copiadas:

```
1     #pragma omp parallel private(ph)
2     {
3       while (1) {
4         /* get pseudo-quanta */
5     #pragma omp critical (MAKE_SPHOT)
6         {
7           if (!quit_flag)
8             make_super_photon(&ph, &quit_flag);
9         }
10        if (quit_flag)
11          break;
12
13        /* push them around */
14        track_super_photon(&ph);
15
16        /* step */
17    #pragma omp atomic
18        N_superph_made += 1;
19        /*mais codigo*/
20      }
21    }
```

Fica claro também - ao observar a linha 8 a 14 - que o processamento do rastreio é feito assim que possível, ao oposto de um processamento em lotes, ou seja, assim que o comandado *make_super_photon* é executado gerando um novo fóton *ph*, o *track_super_photon* é chamado, não havendo algum buffer ou lote, um fóton produzido é um fóton consumido.

Capítulo 3

GPGPU

3.1 História das GPU e GPGPU

era uma vez...

3.2 Bibliotecas: OpenCL e CUDA

o que são e como funcionam porque escolhemos cuda?

Capítulo 4

Otimizacao

4.1 Arquitetura

mostrar gráfico de processamento do grmonty apontar o track super photon como candidato a ser produzido no kernel

4.2 Melhorias e Modificações

4.2.1 Somente uma dimensão

matrix pra vetor

4.2.2 OpenMP e Concorrência

desligar o openmp

4.2.3 math.h

unix math pra nvida math

4.2.4 divisão e trabalho e parelelização

calculo de diviasão de trabalho na GPU

4.2.5 Processar em Lotes

de “assim que possível” “para processamento em lotes”

Capítulo 5

Resultados

5.1 Métricas e Medição

the old

5.2 Comparações

demonstrar o aumento de 100X na velocidade

Capítulo 6

Futuro

6.1 Outras Linguagens de Programação

rust, nim, python

6.2 Single Precision

Usar float ao invés de double

6.3 Novos Disposítivos e Particulariedades dos frabicantes: AMD e NVIDIA

cálculo discreto, artiqueturas diferentes

6.4 Arcabouços: Tensorflow

tensorflow TPU TensorProcessingUnit

6.5 Application-specific integrated circuit chips (ASICs)

O que são? Onde vivem? O que comem?

Capítulo 7

Conclusões

Calculos são importantes e o avanço da ciência depende de artiteturas de alta performance, gpus tem se apresentado competentes na realização de tais tarefas, e sua popular adoção facilita um maior acesso computação astrofísica, aumentando assim a velocidade do progresso cientifico.

Referências Bibliográficas

Dolence *et al.*(**2009**) Joshua C. Dolence, Charles F. Gammie, Monika Mościbrodzka e Po Kin Leung. grmonty: A monte carlo code for relativistic radiative transport. *The Astrophysical Journal Supplement*, 184:387–397. Citado na pág. 1, 3, 4

Victor *et al.*(**2016**) Eijkhout Victor, Chow Edmond e Geijn V. Robert. *Introduction to High Performance Scientific Computing*. Saylor Academy. Citado na pág. 4