

16º Congresso Nacional de Iniciação Científica

TÍTULO: COMPARAÇÃO DE OTIMIZAÇÃO PARA PROBLEMAS DIFÍCEIS UTILIZANDO ALGORITMO GENÉTICO PARALELO COM ALGORITMO GENÉTICO CONVENCIONAL

CATEGORIA: EM ANDAMENTO

ÁREA: CIÊNCIAS EXATAS E DA TERRA

SUBÁREA: COMPUTAÇÃO E INFORMÁTICA

INSTITUIÇÃO: UNIVERSIDADE NOVE DE JULHO

AUTOR(ES): JONATHAN FARIAS BORDIGNON, FERNANDO ALISSON NASCIMENTO, LUIS CARLOS DOS SANTOS JÚNIOR

ORIENTADOR(ES): FABIO HENRIQUE PEREIRA

Realização:

Apoio:

1. RESUMO

A proposta do projeto é fazer uma comparação de performance utilizando como teste um Algoritmo Genético feito para encontrar soluções para uma instância de problema da função de Rosenbrock, utilizando a plataforma e arquitetura CUDA como ambiente de desenvolvimento. Os algoritmos genéticos são muito efetivos para resolver os problemas de otimização, pelo fato destes terem um grande universo de possibilidades que visam encontrar uma solução aproximada do ótimo.

Mesmo pelo fato de algoritmos genéticos serem ótimos para resolução de problemas, geralmente eles tendem a ter um custo muito alto computacionalmente. Por esse motivo a ideia é de desenvolver usando programação paralela, diminuindo consideravelmente o tempo de execução desses algoritmos. Um exemplo clássico do que poderia se beneficiar desse paradigma de programação é o problema de Job Shop Scheduling Problem ou Caixeiro Viajante, que requerem um grande esforço computacional para ser solucionados.

2. INTRODUÇÃO

Compute Unified Device Architecture (CUDA) é uma plataforma de programação paralela e um modelo de programação inventado pela NVIDIA, tem como base a linguagem C. O objetivo da plataforma é utilizar o poder de processamento das GPUs que antes era destinado apenas à computação gráfica para resolver problemas computacionais complexos aumentando drasticamente o desempenho dos algoritmos em relação aos métodos tradicionais. Para poder realizar os testes de otimização foi feito uso do problema de Rosenbrock que é uma função desenvolvida por Howard H. Rosenbrock, que é popularmente usado como teste de performance para algoritmos de otimizações. Para poder realizar esse teste do problema de Rosenbrock foi utilizado um algoritmo genético que entra no campo de algoritmos de otimização. Os algoritmos genéticos são métodos meta-heurísticos e pertence ao grupo de algoritmos evolucionários, desenvolvido por John Henry Holland ele usa como metodologia a teoria de Charles Darwin, esses tipos de algoritmos são conhecidos como ótimos métodos para encontrar uma solução para problemas que possuam um grande universo de possibilidades de resolução. Um exemplo clássico de problema de otimização que poderiam se beneficiar desse paradigma são os problemas de Job Shop Scheduling Problem ou Caixeiro Viajante que estão na ordem de NP-HARD, para isso a ferramenta utiliza o método de

computação paralela, onde as tarefas são executadas em cada multiprocessador de forma independente e paralela.

3. OBJETIVOS

O objetivo principal deste trabalho é demonstrar a eficiência de usar a plataforma de quais dos métodos adotados são mais eficientes através de testes realizado plataforma CUDA. Para realizar o teste de otimização foi usado um algoritmo genético implementado para resolver uma instância de problema da função do Rosenbrock em linguagem C, utilizando a plataforma CUDA ou uma plataforma convencional para resolver um problema de otimização.

4. METODOLOGIA

O projeto em seu primeiro momento adotou uma biblioteca GALib disponível para adaptação de algoritmo genético, e para as simulações de performance foi usado uma instância de problema da função de Rosenbrock. Também foi feito pesquisas relacionadas ao tema usando como fonte principal livros, artigos científicos, teses/dissertações e outras fontes de pesquisa que foram bastante importantes para o desenvolvimento desse trabalho. Para os testes foi realizado o uso de uma placa NVIDIA 820M equipada com CUDA 96 cores; memória de vídeo de 2048 MB DDR3; Clock de processamento de 1,5 GHz, um notebook com processador Intel® Core™ i5-5ª geração, 2.20GHz de processamento; memória física 8,0 GB.

5. DESENVOLVIMENTO

Os testes que estão sendo realizados nesse projeto têm como finalidade pôr em discussão as vantagens de utilizar o conceito de programação em CUDA para problemas de otimização e demonstrar de forma empírica os resultados através de testes realizados. Para isso foi feito uma comparação de resultados que se obteve através de testes que ainda estão sendo realizados com um algoritmo genético usado para resolver uma instância do problema de Rosenbrock. O Algoritmo Genético foi desenvolvido e adaptado para executar paralelamente na plataforma CUDA se beneficiando dos recursos disponíveis e o mesmo algoritmo foi adaptado para executar de forma convencional na linguagem C, sendo assim, possível analisar o desempenho ao utilizar os dois conceitos para problemas de otimização de algoritmo. Os parâmetros do algoritmo genético foram definidos depois de 500 (quinhentas) interações/gerações, o algoritmo finaliza encontrando ou não uma

solução aproximada ou ótima para aquela instância de problema. Para cada interação são criados uma população de 100 indivíduos, formando um número total de 50.000 (cinquenta mil) interações para ser comparados. Em um algoritmo convencional cada indivíduo criado entra numa fila que passará por uma avaliação para saber da sua aptidão (fitness) e verificar se está apto ou não para passar o gene para a próxima geração. Já no algoritmo paralelo feito em CUDA, cada indivíduo ao invés de entrar numa fila para ser processado um a um ele se aproveita do processamento paralelo, por exemplo, graças a arquitetura do CUDA podemos executar os 100 (cem) indivíduos de uma só vez, para ter o mesmo efeito, teríamos que ter uma máquina com 100 (cem) processadores ou no pior dos casos 100 (cem) computadores ligado à rede.

6. RESULTADOS PRELIMINARES

Através de alguns testes foi possível notar que o algoritmo genético utilizando a plataforma CUDA obteve um desempenho 70% mais veloz e eficiente em relação ao algoritmo convencional. Mais estudos são necessários afim de otimizar o desempenho do algoritmo genético paralelo.

7. FONTES CONSULTADAS

NVIDIA Corporation, **CUDA Zone**,

<http://www.nvidia.com.br/object/cuda_home_new_br.html>, acesso em agosto de 2016.

NVIDIA Corporation, **CUDA Toolkit Archive**,

<<https://developer.nvidia.com/cuda-toolkit-archive>>, acesso em agosto de 2016.

NVIDIA, Corporation, **NVIDIA Unveils CUDA™-The GPU Computing Revolution Begins**,

<http://www.nvidia.com/object/IO_37226.html>, acesso em agosto de 2006.

H.H. Rosenbrock, **An Automatic Method for Finding the Greatest or Least Value of a Function. The Computer Journal (1960) 3: 175-184.**

KOZA, J.R, **Genetic Programming. On the Programming of Computers by Means of Natural Selection MIT Press (1992).**

Holland, J. H, **Genetic algorithms. Scientific American, 114-116.**

Formatado: Fonte: 12 pt, Cor da fonte: Texto 1, Inglês (Estados Unidos)