MAC 0219/5742 – Introdução à Computação Concorrente, Paralela e Distribuída

Prof. DR. Alfredo Goldman,
Guilherme Souza S.

1

1. Relatório Geral

Mesmo tendo o ep2 onde tivemos que fazer o mandelbrot sequencial, paralelo com open MP e em nvidia-cuda, ao realizar o mesmo trabalho só que com computação distribuída e efetivar esta distribuição entre mais maquinas, o método de implementação e algumas escolhas sofreram mudanças, as quais o míni ep de processos ajudou imensamente no compreendimento da distribuição, que o lembra um pouco, claro contando com o auxílio de código fornecido pelos monitores e com um boa explicação dos mesmos durante aula, isso fez com que o entendimento fosse um pouco mais claro.

Para tal trabalho e para uma melhor distribuição, optei por mudar a implementação inicial, no caso a optada no ep2. Agora passei a fazer uso diretamente da imagem como um grande vetor, dessa forma pude realizar uma distribuição mais igualitária entre o número de máquinas disponíveis, fazendo uso assim de MPI_Send() e MPI_Recv() para enviar as partes na qual deve ser calculado o mandelbrot, que na minha opnião foi uma boa escolha no tratamento da situação. O que para mim ficou um tanto quanto parecido com a forma que é realizado o tratamento em CUDA, que basicamente foi de onde tirei minha ideia, pois tendo o índice da posição, basta calcular o mandelbrot baseado na transformação do índice em posição x e y, para uso no calculo, dessa forma tornando o trabalho mais simples, alem de ser necessário somente construir um vetor exatamente do tamanho do trabalho recebido para cada maquina, deixando o trabalho de construir o vetor inteiro para o pai, somente puxando os resultados dos filhos com vetores menores.

Em minha situação, a parte de open mp apresentou-se um pouco mais complicado quando comparado a implementação do ep2, pois como estava destruído algumas variáveis nas quais eu tinha dado a opção private, no atual trabalho já não se fazia mais necessário, alem da forma de tratamento ter sido diferente da implementação inicial, com isso levei algum tempo até encontrar a melhor forma de organização do parallel for() de forma que não bagunça-se a imagem e sim a tornando o calculo mais eficiente.

Com isso, cuda também se apresentou como uma dificuldade na realização do cálculo, pois como agora além da divisão realizada por mim, o próprio cuda quebra em blocos, com isso o valor original do índex real da posição do vetor na imagem original, era perdido. Como solução dado o tempo que tinha para solucionar, optei por armazenar o index real dentro do vetor, e fazer uso do valor dentro do vetor para o cálculo do mandelbrot. Outro problema foi no momento de realizar a linkagem, no qual após um pesquisa e com o fórum da disciplina usei a opção do mpicc juntando os pontos objeto, mas tendo que mantes a parte de cuda separado da parte ponto c, assim chamando como extern ”C” ou seja uma função externa ao código, (pois assim compilei e gerei o ponto objeto com código cuda como dito anterior, gerando da mesma forma o ponto objeto da versão C, depois só linkando-os). Na Figura 2 é possível ver a imagem resultante de uma execução.

2. Hardware

O computador no qual os testes foram rodados conta com as configurações apresentadas abaixo, sendo dividida em duas máquinas de execução, pelo fato da necessidade da placa de vídeo NVIDIA exigida pelo trabalho, tal diferença entre hardware foi desconsiderada nos testes. Tais informações foram obtidas com o comando *lscpu* em conjunto com *lshw* e *nvidia-smi*.

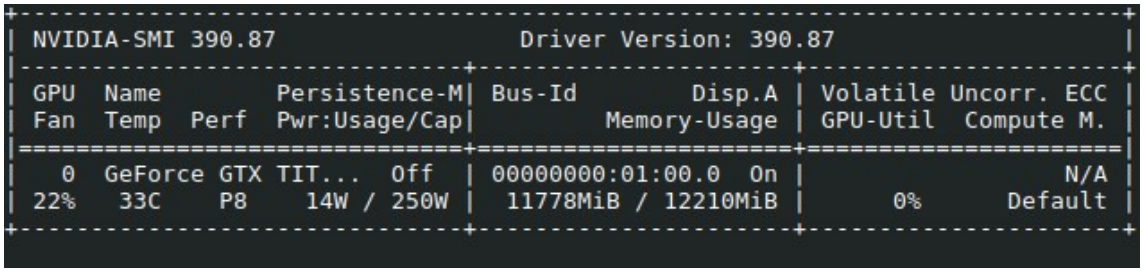

Figure 1. Informações da Placa de Vídeo

```
Arquitetura:            x86_64
Modo(s) operacional da CPU:32-bit, 64-bit
Ordem dos bytes:        Little Endian
CPU(s):                 4
Lista de CPU(s) on-line:0-3
Thread(s) per n cleo: 2
N cleo(s) por soquete:2
Soquete(s):             1
N (s) de NUMA:          1
ID de fornecedor:       GenuineIntel
Fam lia da CPU:         6
Modelo:                 58
Nome do modelo:         Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz
Step:                   9
CPU MHz:                1356.701
CPU MHz m x.:           1800,0000
CPU MHz m n.:           800,0000
BogoMIPS:               3591.71
Virtualiza o:           VT-x
cache de L1d:           32K
cache de L1i:           32K
cache de L2:            256K
cache de L3:            3072K
CPU(s) de n 0 NUMA:     0-3
```

Listing 1. informações CPU (Sem placa de Video)

```
*-cache:0
             description: L1 cache
             physical id: 6
```

```
             slot: L1-Cache
             size: 32KiB
             capacity: 32KiB
             capabilities: internal write-through instruction
             configuration: level=1
        *-cache:1
             description: L2 cache
             physical id: 7
             slot: L2-Cache
             size: 256KiB
             capacity: 256KiB
             capabilities: internal write-through unified
             configuration: level=2
        *-cache:2
             description: L3 cache
             physical id: 8
             slot: L3-Cache
             size: 3MiB
             capacity: 3MiB
             capabilities: internal write-back unified
             configuration: level=3
```

Listing 2. Informacoes Cache (Sem placa de Video)

```
*-memory
          description: System Memory
          physical id: 29
          slot: System board or motherboard
          size: 4GiB
```

Listing 3. Informacoes Memoria (Sem placa de video)

```
Arquitetura:             x86_64
Modo(s) operacional da CPU:32-bit, 64-bit
Ordem dos bytes:         Little Endian
CPU(s):                  8
Lista de CPU(s) on-line:0-7
Thread(s) per n cleo: 2
N cleo(s) por soquete:4
Soquete(s):              1
N (s) de NUMA:           1
ID de fornecedor:        GenuineIntel
Fam lia da CPU:          6
Modelo:                  58
Nome do modelo:          Intel(R) Xeon(R) CPU E3-1230 V2 @ 3.30GHz
Step:                    9
CPU MHz:                 1985.559
CPU MHz m x.:            3700,0000
CPU MHz m n.:            1600,0000
BogoMIPS:                6585.24
Virtualiza o:            VT-x
cache de L1d:            32K
cache de L1i:            32K
cache de L2:             256K
cache de L3:             8192K
CPU(s) de n 0 NUMA:      0-7
```

Listing 4. informações CPU (Com placa de Video

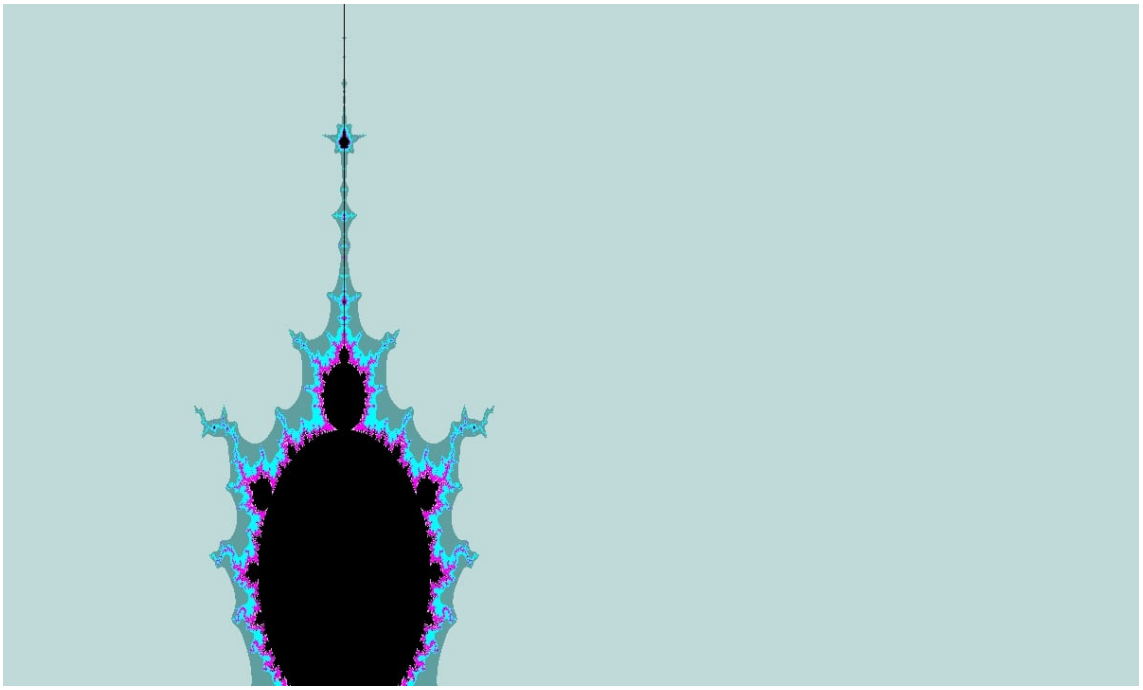

Figure 2. Imagem gerada para testes, com entradas (0, -2,0)(-1.0, 1.0)