

RAINHAS DO CÓDIGO: MULHERES NA SUPERCOMPUTAÇÃO

CONHEÇA O PRINCÍPIO BÁSICO DAS FERRAMENTAS

MAÍRIAN MENDES

Ø T 0 0 T T 0 0 O T T 0 0 0 T 0 T T T Ø T T 0 0 0 0 0 0 T 0 T 0 T T 0 T 0 T 0 T 0 0 T 0 0 T 0 T 0 0 0 T 0 0 T 0 T 0 **T** t T 0 T 0 T T 0 0 T 0 0 T 0 T T T T 0 0 0 T 0 0 T T 0 0 T T T 0 T. 0 T T O O T O O T O O T O 0 0 T O T O T O T O T στ 0 T 0 0 0 t 0 T T T T O O O T O O Ø 0 0 T 0 0 0 T 0 T 0 T 0 0 T T 0 0 T T T Ø T Ø T T 9 0 T 0 T 0 T **Θ** τ **Θ** τ **Θ** τ **Θ** τ **Θ** τ T · F · O T · O T · O T 0 0 T 0 T 0 T 0 T T 0 T 0 0 T 0 0 T 0 T 0 T 0 T 0 T 0 T 0 T 0 T 0 T 0 1 0 T 0 T 0 0 T 0 **Θ**ττ**Θ**Θτ 0 0 T 0 Ø T Ø Ø τ



High-Performance Computing (HPC)

High-Performance Computing, refere-se ao uso de supercomputadores e técnicas de computação paralela para resolver problemas computacionalmente intensivos. Variam desde simulações científicas até a análise de big data e desenvolvimento de inteligência artificial.

0 T O T T T O O 0 T 0 0 O T T 0 T 0 T 0 0 T 0 T 0 0 T 0 T 0 0 0 0 0 0 T T О Т Т 0 0 E T 0 0 T 0 T T 0 0 0 0 0 0 0 T 0 0 T T T 0 T 0 0 T 0 0 0 0 T 0 0 0 T 0 T 0 0 0 0 0 0 0 0 0 0 0 0 0 T T T 0 0 T T T 0 T T T 0 0 T T 0 0 0 0 0 T 0 0 0 0 τ 0 τ 0 0 0 0 0 0 0 T 0 T 0 T T 0 T T T T T T τ T T T 0 0 0 T T 0 0 0 0 0 0 0 T 0 0 T T T T T T T 0 T 0 0 L T 0 0 0 0 T 0 T 0 T T 0 0 0 T 0 0 T T 0 0 0 0 T T T T 0 T 0 T T 0 0 τ T 0 T 0 τ T 0 0 T 0 T T 0 0 T T 0 0 0 0 T T 0 0 0 T 0 T L 0 0 0 T T T T 0 0 τ T τ T 0 0 T 0 0 0 - 1111 - 111 0 0 000000 000 000 lila: i. T 0 000000 000000 1.133.3 . lili .lild Lili _ili T T T τ 111 T T 0 0 T Wikewitk Bauter03 5.11 (a) 1999-2011 http://www.mibrotth.com/ 0 0 0 0 0 0 T 0 mwish has no sopinare kep T T τ 0 0 0 0 You have 23%38s to configure the router to be remotely occurately, and to cater the boy by posting it to a letter window or in Wimbor. See unwinitedlikicon/bey for more details. erpent Installation "zoftwere 80": 8638-2HEE lease press "Ester" to essilauch 0 T ### [m#Hikra#ik] 5 0 0 0 0 T 0 T T 0 T 0 0 T 0 0 0 0 0 0 0 0 0 0 0 0 0 0 T Т T 0 T T 0 0 T 0 T 0 0 0 0 T 0 0 0 T T 0 0 0 T τ 0 0 T 0 0 0 0 0 0 τ 0 0 T 0 T τ τ T T 0 τ T T T 0 0 T 0 0 0 0 0 0 0 T 0 0 0 0 0 0 0 T 0 T τ T 0 0 T T T T T 0 T 0 T 0 0 T 0 0 0 0 0 0 0 0 0 8 0 0 0 0 T 0 0 T 0 0 0 0 T 0 T 0 0 0 0 T 0 0 T 0 0 T 0 T 0 0 0 0 0 0 T 0 0 0 0 0 0 T 0 0 0 0 0 T T 0 T T 0 0 T 0 0 0 0 0 T T T 0 0 0 0 0 T 0 0 0 0 T 0 0 T 0 0 0 0 0 τ 0 0 T 0 0 T T 0 T 0 0 T T 0 0 0 0 0 0 0 0 T 0 0 T 0 T T T 0 0 T T T T 0 0 0 T T 0 0 τ 0 0 T τ T 0 0 0 T 0 0 0 T T 0 T T T 0 0 T 0 T T 0 0 T T 0 T 0 0 τ 0 0 T 0 T T 0 T 0 0 τ L 0 0 0 0 0 0 0 0 T 0 T 0 T 0 T T 0 0 0 T T 0 T



MPI (Message Passing Interface)

O MPI é uma biblioteca padrão para comunicação entre processos em um ambiente de computação paralela. Ele permite que diferentes partes de um programa rodem simultaneamente em vários processadores.

Exemplo de Código: Hello World utilizando o MPI

```
# Código MPI em C
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {
    MPI_Init(&argc, &argv);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    printf("Hello world from processor %d out of %d processors\n", world_rank, world_size);

MPI_Finalize();
    return 0;
}
```

0

0

0 0

O T

0

0

τ τ Θ **Θ**

· Ø Ø
· Ø Ø
τ Τ Τ

Θττττ

ΘΤΤΕΕΕΘΘΘ

T

Ø T

0 0

0

0

T T T

0 0 0

T 0

T 0 T

Como Compilar e Executar:

```
woman_HPC - MaÍRIAN MENDES.HPC

mpicc -o hello_mpi hello_mpi.c

mpirun -np 4 ./hello_mpi
```





Open MP (Open Multi-Processing)

Open Multi-Processing) é uma API (Application Programming Interface) que suporta programação paralela em sistemas com múltiplos processadores, permitindo a criação de programas multithread.

Exemplo de Código: Suma Paralela com OpenMP

```
WOMAN HPC - MAÍRIAN MENDES.HPC
# Código OpenMP em C
#include <omp.h>
#include <stdio.h>
int main() {
   int n = 1000;
   int a[n], b[n], sum[n];
   for(int i = 0; i < n; i++) {
       a[i] = i;
       b[i] = i;
   #pragma omp parallel for
   for(int i = 0; i < n; i++) {
       sum[i] = a[i] + b[i];
   }
   printf("Soma dos primeiros 10 elementos:\n");
   for(int i = 0; i < 10; i++) {
       printf("%d ", sum[i]);
   printf("\n");
   return 0;
```

Como Compilar e Executar:

```
WOMAN_HPC - MAÍRIAN MENDES.HPC
gcc -fopenmp -o sum_openmp.c
./sum_openmp
   printf("Soma dos primeiros 10 elementos:\n");
   for(int i = 0; i < 10; i++) {
       printf("%d ", sum[i]);
   printf("\n");
   return 0;
```



CUDA (Compute Unified Device Architecture)

É uma plataforma de computação paralela desenvolvida pela NVIDIA. Permite que os desenvolvedores utilizem a potência total das GPU's (Unidades de Processamento Gráfico) para executar cálculos em paralelo.

Exemplo de Código: Multiplicação de Matrizes com CUDA

```
WOMAN HPC - MAÍRIAN MENDES.HPC
# Código CUDA em C
#include <cuda.h>
#include <stdio.h>
__global__ void matrixMul(int* a, int* b, int* c, int N) {
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;
    int sum = 0;
    for (int i = 0; i < N; ++i) {
        sum += a[row * N + i] * b[i * N + col];
    c[row * N + col] = sum;
int main() {
    int N = 2;
    int a[4] = \{1, 2, 3, 4\};
    int b[4] = \{1, 2, 3, 4\};
    int c[4] = \{0\};
    int *d_a, *d_b, *d_c;
    cudaMalloc((void**)&d_a, N * N * sizeof(int));
    cudaMalloc((void**)&d_b, N * N * sizeof(int));
    cudaMalloc((void**)&d_c, N * N * sizeof(int));
    cudaMemcpy(d_a, a, N * N * sizeof(int), cudaMemcpyHostToDevice);
    cudaMemcpy(d_b, b, N * N * sizeof(int), cudaMemcpyHostToDevice);
    dim3 threadsPerBlock(2, 2);
    dim3 blocksPerGrid(1, 1);
    matrixMul<<<blocksPerGrid, threadsPerBlock>>>(d_a, d_b, d_c, N);
    cudaMemcpy(c, d_c, N * N * sizeof(int), cudaMemcpyDeviceToHost);
    printf("Resultado⊤da multiplicação de matrizes:\n");
    for (int i = 0; i < N * N; i++) {
        printf("%d ", c[i]);
        if ((i + 1) % N == 0) printf("\n");
   cudaFree(d_a);
    cudaFree(d_b);
   cudaFree(d_c);
   return 0;
```

0

0

T

0

T

T

0

T

T

T

0

0

τ

T

Como Compilar e Executar:

WOMAN_HPC - MAÍRIAN MENDES.HPC

nvcc -o matrixMul matrixMul.cu
./matrixMul



Conclusão:

Estas ferramentas são essenciais para o desenvolvimento de aplicações em HPC. MPI, OpenMP e CUDA permitem a programação paralela de maneira eficiente, maximizando o aproveitamento do poder computacional disponível. Com os exemplos fornecidos, você poderá iniciar sua jornada de exploração e experimentação com essas tecnologias.

Agradecimentos

Considerações finais:

Esse e-book foi gerado por I.A. e diagramado por humano.

Este conteúdo foi gerado para fins didáticos e de construção. Não foi realizada uma validação humana cuidadosa do material aqui apresentado, portanto, é possível encontrar erros cometidos pela inteligência artificial.





Link do GitHub:

OBRIGADA POR LER ATÉ AQUI!