

Memorando

De: Anizia da Silva Dala Quinguangua

Nº de Matrícula: 20191391

Disciplina: Computação Paralela e Distribuída

Assunto: Análise de Desempenho com OpenMP e Intel VTune

Data: 18/04/2025

1. Introdução

Este relatório documenta o Laboratório 3 da disciplina de Computação Paralela e Distribuída, cujo foco é o uso da ferramenta Intel VTune Profiler para analisar e otimizar o desempenho de algoritmos paralelos implementados com OpenMP. Foram desenvolvidas três versões de um algoritmo de multiplicação de matrizes, com o objetivo de identificar gargalos e comparar a eficiência entre implementações diferentes.

Objetivos: - Analisar o desempenho de aplicações com OpenMP. - Identificar pontos críticos e oportunidades de otimização. - Utilizar o Intel VTune para profiling e análise gráfica.

Restrições: - Utilização do compilador `gcc` com suporte a OpenMP. - Ferramenta Intel VTune instalada localmente.

Observações: - A análise foi realizada com base em códigos criados exclusivamente para este laboratório.

2. Experiências Realizadas

1. Abra mult-sync.c, mult-redux.c e mult-good.c. Quais problemas consegue identificar em mult-sync.c e mult-redux.c?

R: O problema foi o uso da diretiva **#programa omp critical** dentro do loop mais interno (multiplicação), o que causa serialização do código. Isso impede que múltiplas threads escrevam simultaneamente, criando um gargalo de sincronização, tornando assim o paralelismo ineficiente e grande overhead por contenção de threads.

2. Execute a análise de threading com Sampling em modo de utilizador para multsync. Porquê que a utilização efectiva da CPU é tão baixa?

R: O uso do **#programa omp critical** em cada operação de multiplicação cria um ponto de contenção constante entre as threads. Baixo tempo efectivo porque as threads estão

ociosas aguardando acesso à região crítica, e o Intel Vtune mostraria alto tempo em spin time.

3. Mude para a visualização Top-down Tree e expanda a árvore da seguinte forma: Total > clone > start thread > [OpenMP worker]. Clique duas vezes em main.omp fn.0 (ou seja, a função principal) para abrir o código-fonte. Quais directivas do OpenMP apresentam alto overhead? Qual é o principal problema nesta implementação?

R: Diretiva **#programa omp critical** com alto overhead, e o uso desta para atualizar `C[i][j]` aumenta o tempo de execução exponencialmente com o número de threads, anulando os ganhos do paralelismo e cria um efeito similar a uma implementação sequencial.

4. Repita um procedimento semelhante para mult-redux. Onde está o ponto de sincronização no código? Porquê que o programa gastou tanto tempo nele?

R: O ponto de sincronização é implícito no fim de cada bloco **#programa omp for**, onde as threads se sincronizam por padrão. Embora **mult-redux.c** não use critical, cada thread calcula sum local, e há custo de criação e sincronização de threads.

5. Agora, execute a análise de threading para mult-good. Porquê que essa implementação é mais eficiente?

R: Usa **#programa omp paralell for collapse(2)**, o que: paraleliza tanto as linhas, quanto as colunas da matriz; cria mais tarefas menores, melhor distribuídas entre as threads: reduz o tempo de espera e aumenta a eficiência de uso dos núcleos.

3. Desafios

- - Configuração correta do Intel VTune com símbolos de debug.
- - Interpretação de resultados nas visualizações Bottom-up e Flame Graph.
- - Identificação de overheads de sincronização e redução de carga entre threads. - Otimização progressiva do código com base nas análises

Compile os códigos com suporte a OpenMP e símbolos de debug (-g):

- ❖ gcc -fopenmp -g mult-sync.c -o mult-sync
- ❖ gcc -fopenmp -g mult-redux.c -o mult-redux
- ❖ gcc -fopenmp -g mult-good.c -o mult-good

4. Referências Bibliográficas

- Intel VTune Profiler: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html> - OpenMP Specifications: <https://www.openmp.org/specifications/>

- VTune CLI Guide: <https://www.intel.com/content/www/us/en/docs/vtune-profiler/user-guide/2025-1/cli-reference.html>.

5. Repositório GitHub

Link do repositório: <https://github.com/AniziaSilva/openmp-lab3>