



DCA3703 PROGRAMAÇÃO PARALELA

Prof. Samuel Xavier
de Souza

HISTÓRICO E EVOLUÇÃO DA PROGRAMAÇÃO

- Objetivos desse tópico
 - Reconhecer a necessidade de maior conhecimento do hardware para alcançar alto desempenho
 - Reconhecer a necessidade da programação paralela para alcançar alto desempenho

HISTÓRICO E EVOLUÇÃO DA PROGRAMAÇÃO

Tarefa 1:

Implemente duas versões da multiplicação de matriz por vetor ($M \times V$) em C: uma com acesso à matriz por linhas (linha externa, coluna interna) e outra por colunas (coluna externa, linha interna). Meça o tempo de execução de cada versão com uma função apropriada e execute testes com diferentes tamanhos de matriz. Identifique a partir de que tamanho os tempos passam a divergir significativamente e explique por que isso ocorre, relacionando suas observações com o uso da memória cache e o padrão de acesso à memória.

- O gargalo de von Neumann e a memória cache
 - Localidade temporal e espacial
 - Row-major vs. column-major

HISTÓRICO E EVOLUÇÃO DA PROGRAMAÇÃO

Tarefa 2:

Implemente três laços em C para investigar os efeitos do paralelismo ao nível de instrução (ILP): 1) inicialize um vetor com um cálculo simples; 2) some seus elementos de forma acumulativa, criando dependência entre as iterações; e 3) quebre essa dependência utilizando múltiplas variáveis. Compare o tempo de execução das versões compiladas com diferentes níveis de otimização (O0, O2, O3) e analise como o estilo do código e as dependências influenciam o desempenho.

- Paralelismo ao nível de instrução
 - Pipelining,
 - Vetorização

HISTÓRICO E EVOLUÇÃO DA PROGRAMAÇÃO

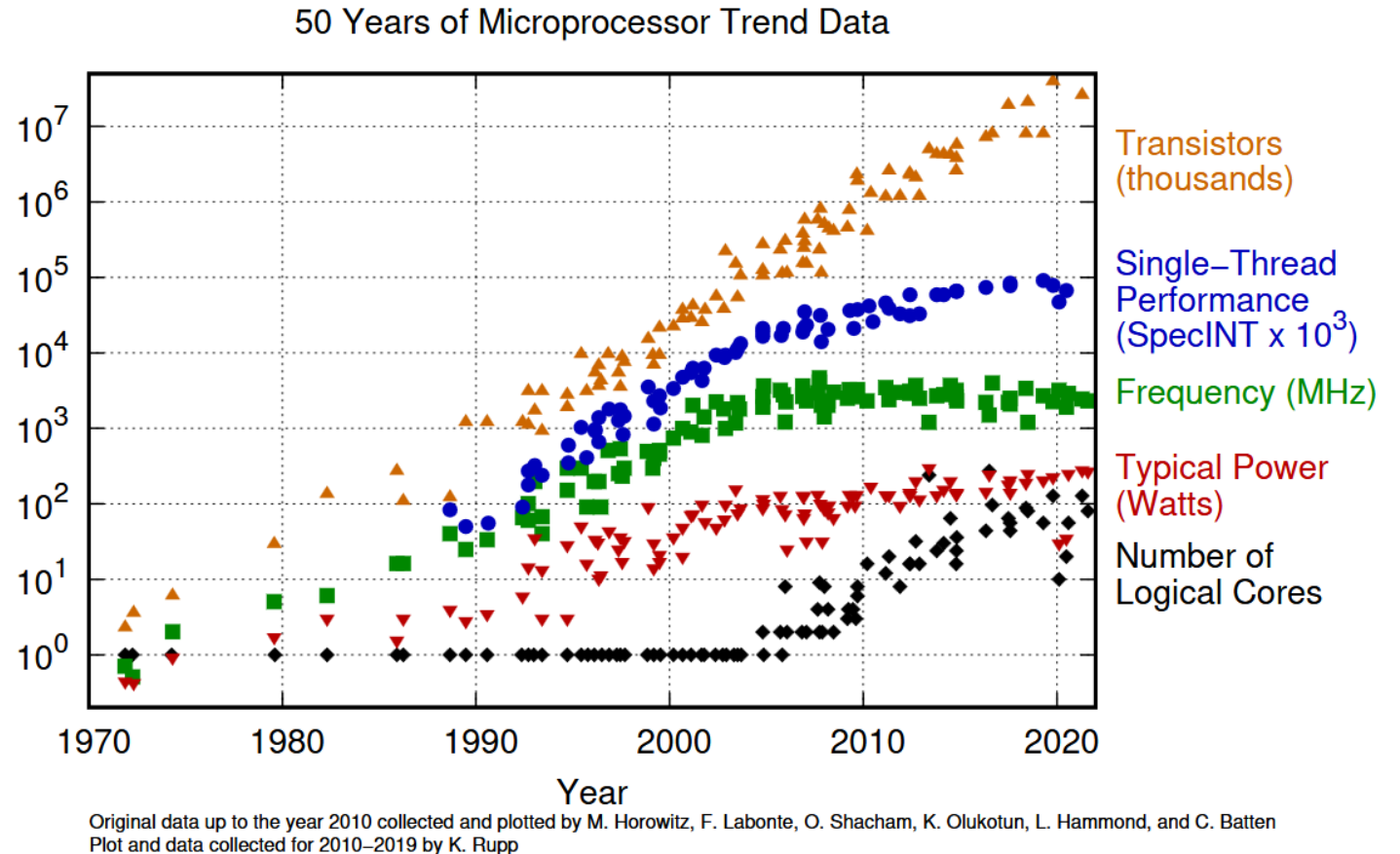
- Por que algumas aplicações requerem maior poder computacional continuamente?
 - Slides sobre aplicações que requerem cada vez mais poder computacional

HISTÓRICO E EVOLUÇÃO DA PROGRAMAÇÃO

Tarefa 3:

Implemente um programa em C que calcule uma aproximação de π usando uma série matemática, variando o número de iterações e medindo o tempo de execução. Compare os valores obtidos com o valor real de π e analise como a acurácia melhora com mais processamento. Reflita sobre como esse comportamento se repete em aplicações reais que demandam resultados cada vez mais precisos, como simulações físicas e inteligência artificial.

- As limitações físicas da fabricação de processadores sequenciais mais rápidos



PROGRAMAS MULTITAREFAS

- Objetivos desse tópico
 - Reconhecer a necessidade de maior conhecimento do software para alcançar alto desempenho
 - Reconhecer a maior dificuldade de se programar em paralelo

PROGRAMAS MULTITAREFAS

Tarefa 4:

Implemente dois programas paralelos em C com OpenMP: um limitado por memória, com somas simples em vetores, e outro limitado por CPU, com cálculos matemáticos intensivos. Paralelize com `#pragma omp parallel for` e meça o tempo de execução variando o número de threads. Analise quando o desempenho melhora, estabiliza ou piora, e reflita sobre como o multithreading de hardware pode ajudar em programas memory-bound, mas atrapalhar em programas compute-bound pela competição por recursos.

- Quais programas são limitados pela memória (gargalo de von Neuman) e quais não são?
 - Como o multi-thread de hardware pode ajudar (ou atrapalhar)