SuperComputação

Aula 4 – Modelo fork-join usando OpenMP

2018 – Engenharia

Igor Montagner, Luciano Soares <igorsm1@insper.edu.br>

Modelo fork-join

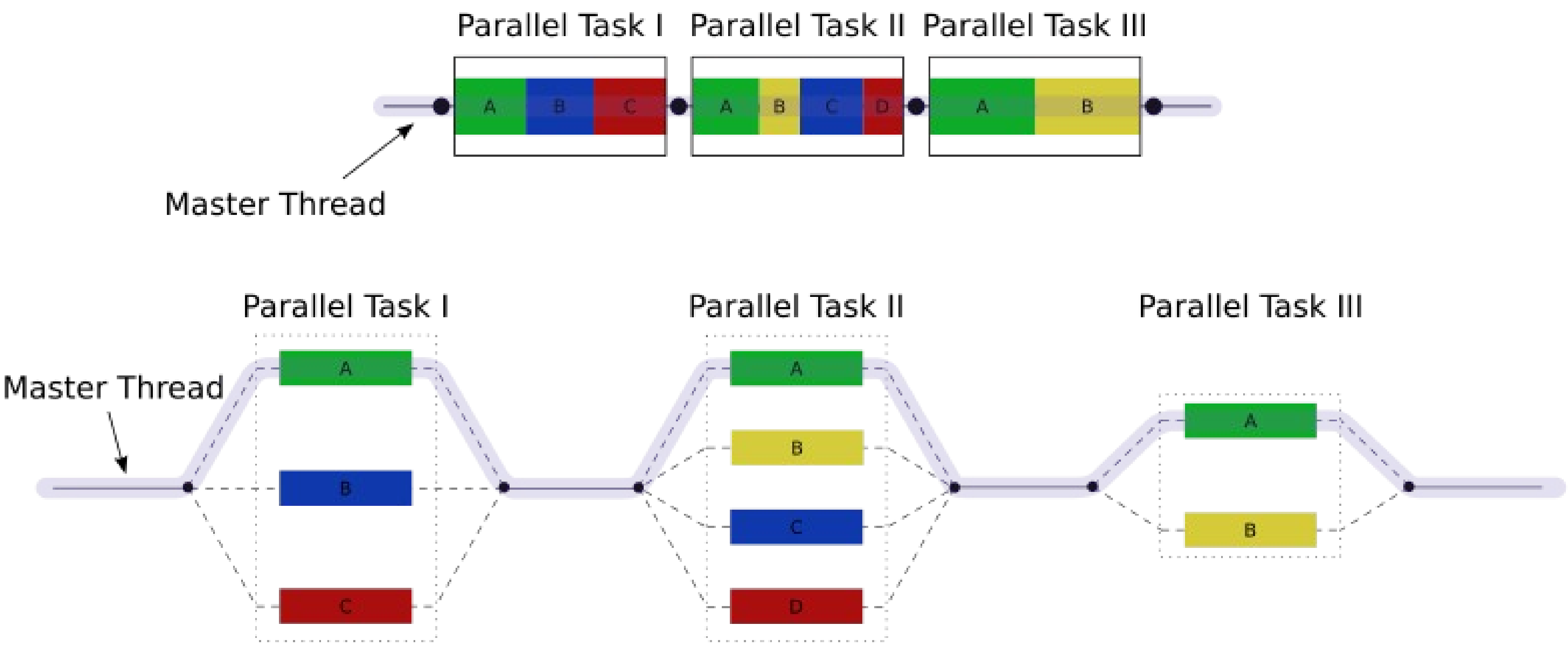


Figura: https://en.wikipedia.org/wiki/File:Fork_join.svg

Roteiro: Modelo fork-join raiz

Parte 1:

```
#include <thread>
#include <iostream>

void thread_f(int id) {
    std::cout << "Thread #" << id << std::endl;
}

int main() {

    int max_threads = std::thread::hardware_concurrency();

    std::thread *array = new std::thread[max_threads];

    for (int i = 0; i < max_threads; i++) {
        array[i] = std::thread(thread_f, i);
    }

    for (int i = 0; i < max_threads; i++) {
        array[i].join();
    }
}
```

Roteiro: Modelo fork-join raiz

```
void soma_thread(double *vec, int n, int start, int end, double *res) {
    *res = soma_vec_seq(vec, n, start, end);
}

double soma_vec_par(double *vec, int n) {
    int max_threads = std::thread::hardware_concurrency();
    std::thread *arr = new std::thread[max_threads];
    double *somas_parciais = new double[max_threads];

    int part_size = n / max_threads + 1;

    for (int i = 0; i < max_threads; i++) {
        somas_parciais[i] = 0;
        int start =  i * part_size;
        int end = start + part_size;
        if (end > n) end = n;
        arr[i] = std::thread(soma_thread, vec, n, start, end, &somas_parciais[i]);
    }

    double sum = 0;
    for (int i = 0; i < max_threads; i++) {
        arr[i].join();
        sum += somas_parciais[i];
    }

}
```

Hoje

- Introdução a OpenMP
- Atividade prática: cálculando pi com OpenMP

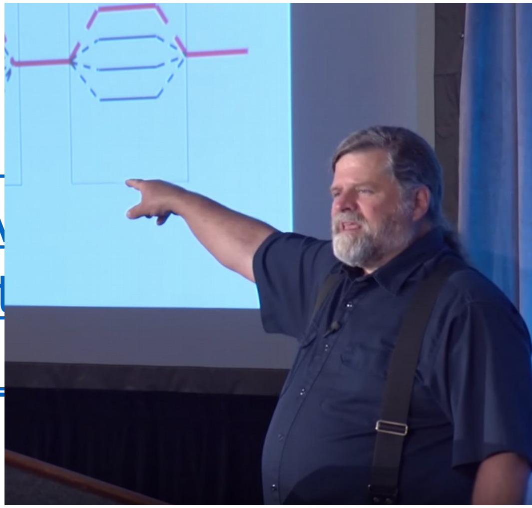
OpenMP

- Conjunto de extensões para C/C++ e Fortran
- Fornece construções que permitem paralelizar código em ambientes multi-core
- Padroniza práticas SMP + SIMD + Sistemas heterogêneos (GPU/FPGA)
- Idealmente funciona com mínimo de modificações no código sequencial

OpenMP (fontes)

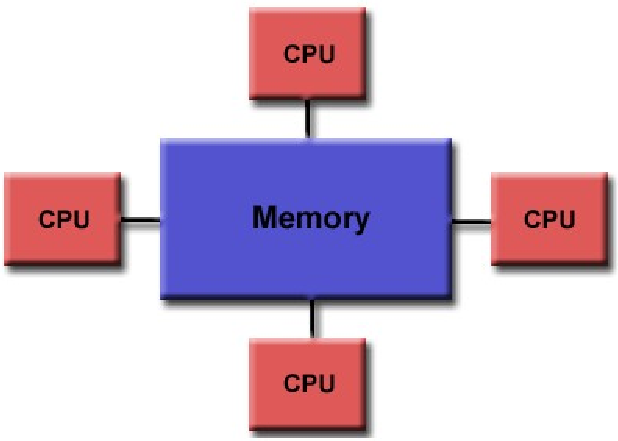
- Material é baseado no curso “A brief Introduction to parallel programming” - Tim Mattson

- Vídeos:
  https://www.youtube.com/watch?v=pRtTIW9-
  https://www.youtube.com/watch?v=LRsQHDA
  https://www.youtube.com/watch?v=dK4PITrQt
  https://www.youtube.com/watch?v=WvoMpG_

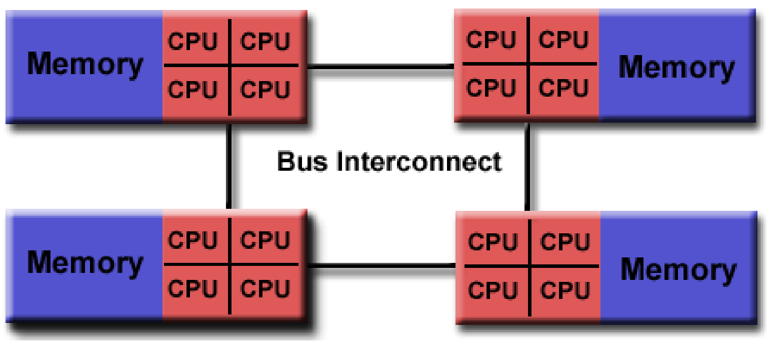


- Slides:
  http://extremecomputingtraining.anl.gov/files/2016/08/Mattson_830aug3_HandsO

Tipos de memória compartilhada (shared memory)

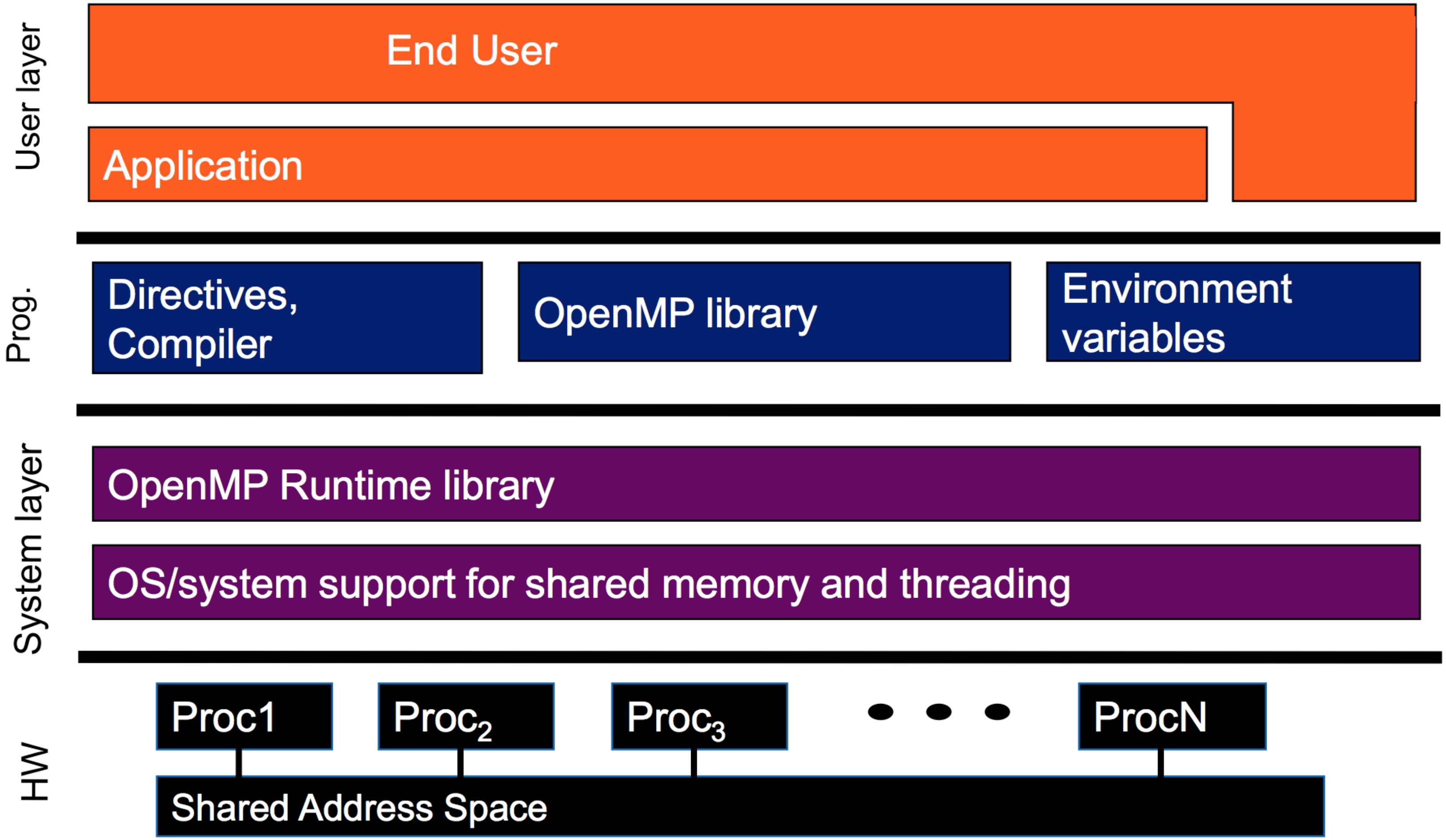


Uniform Memory Access



Non-Uniform Memory Access

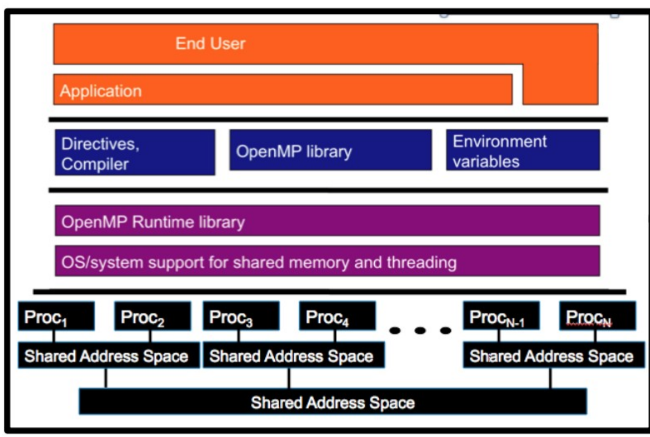
OpenMP (host / NUMA)



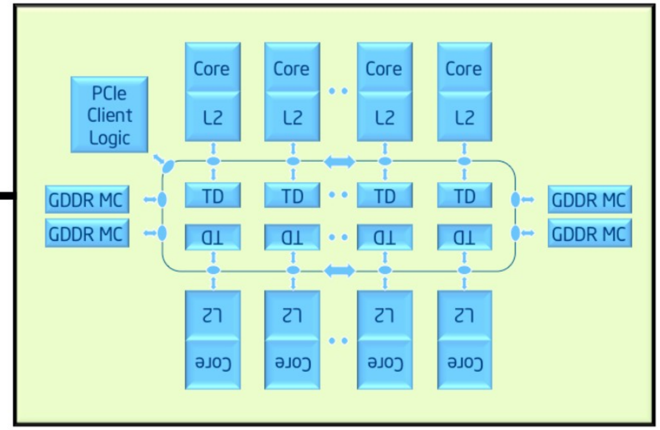
OpenMP (heterogêneo / target)

Version 4.0-4.5

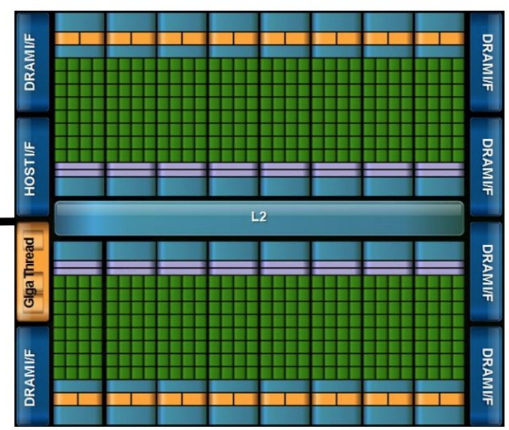
Supported (since OpenMP 4.0) with target, teams, distribute, and other constructs



Host



Target Device: Intel® Xeon Phi™ coprocessor



Target Device: GPU

OpenMP - sintaxe

Diretivas de compilação

```
#include <omp.h>
#pragma omp construct [params]
```

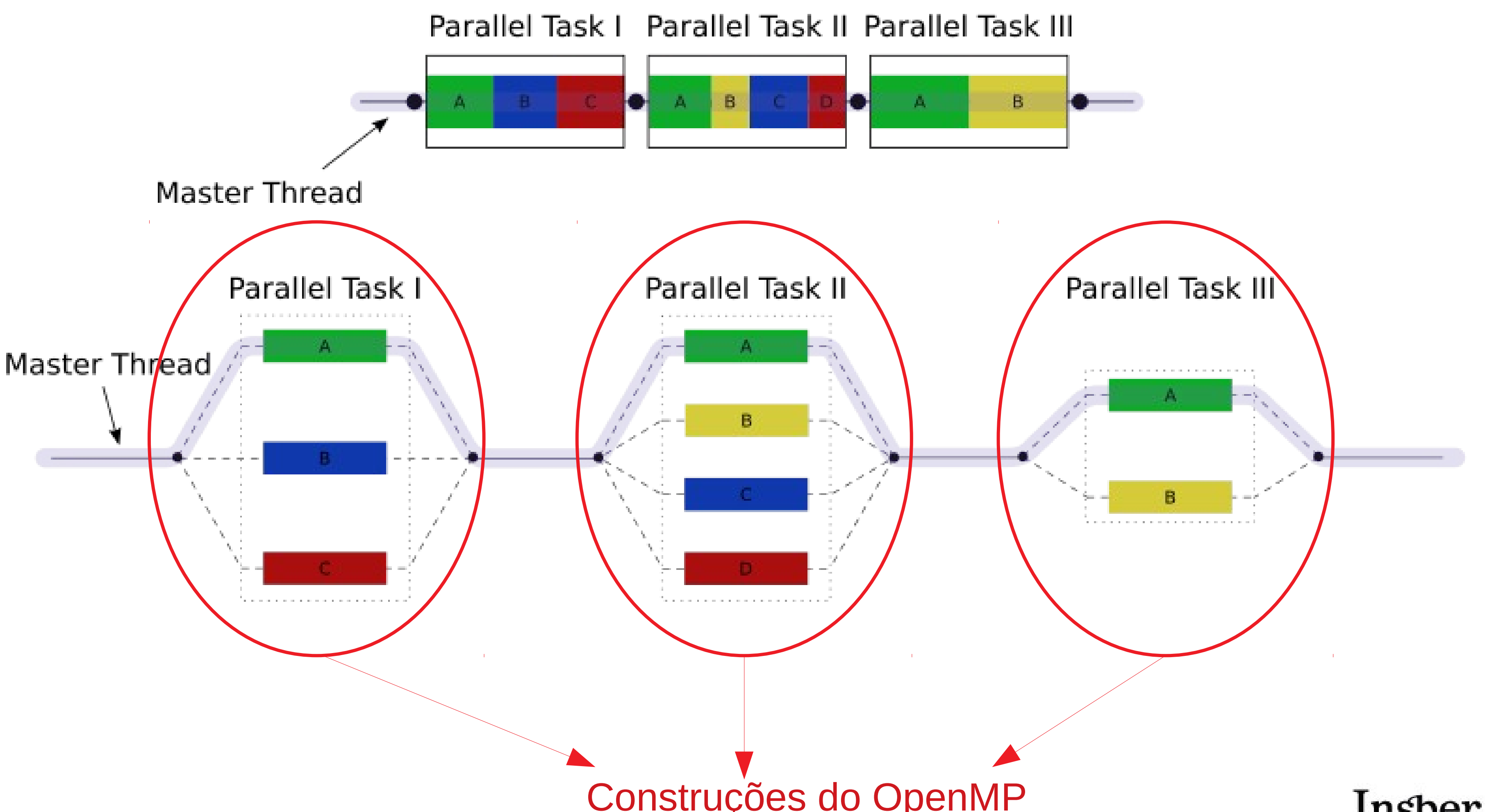
Aplicadas a um bloco de código

limitado diretamente por { }

for (...) { }

Com join implícito

OpenMP – aplicação do modelo fork-join



OpenMP - exemplos

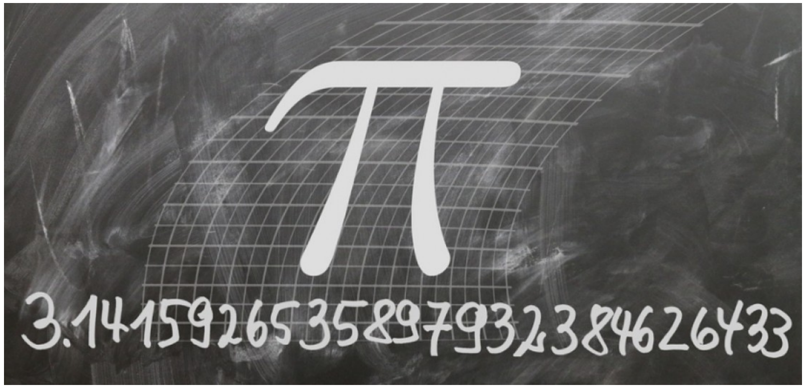
Atividade prática – parte 1

Criação de threads usando OpenMP

API simples para obter/definir

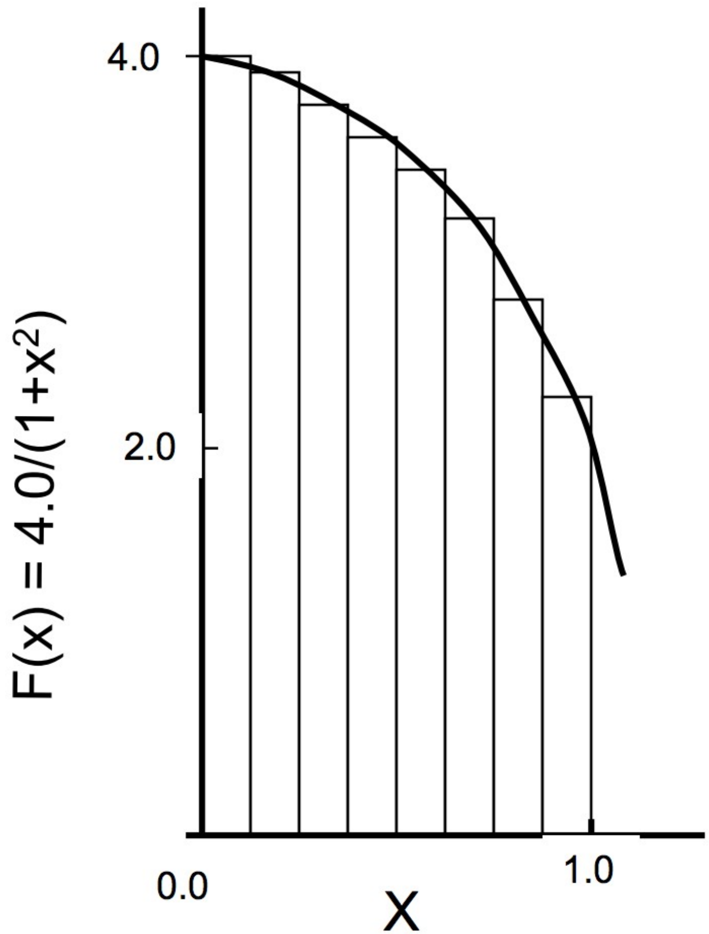
- Número máximo de threads
- Thread atual

Atividade prática – parte 2



- Recorde atual:
  - por Peter Trueb (Dectris)
  - 22,459,157,718,361 dígitos
  - 105 dias de processamento
  - Servidor:
    - Dell PowerEdge R930
      4 hyper-threaded 18-core Intel Xeon E7-8890 v3 @ 2.5 GHz
      1.25 TB RAM
  - Armazenamento
    - 20 x Seagate Enterprise NAS 6 TB
      4 GB/s bandwidth
      60 TB Backup Storage
  - Código usado: y-cruncher por Alexander J. Yee baseado no algoritmo de Chudnovsky.

Atividade prática – nosso método



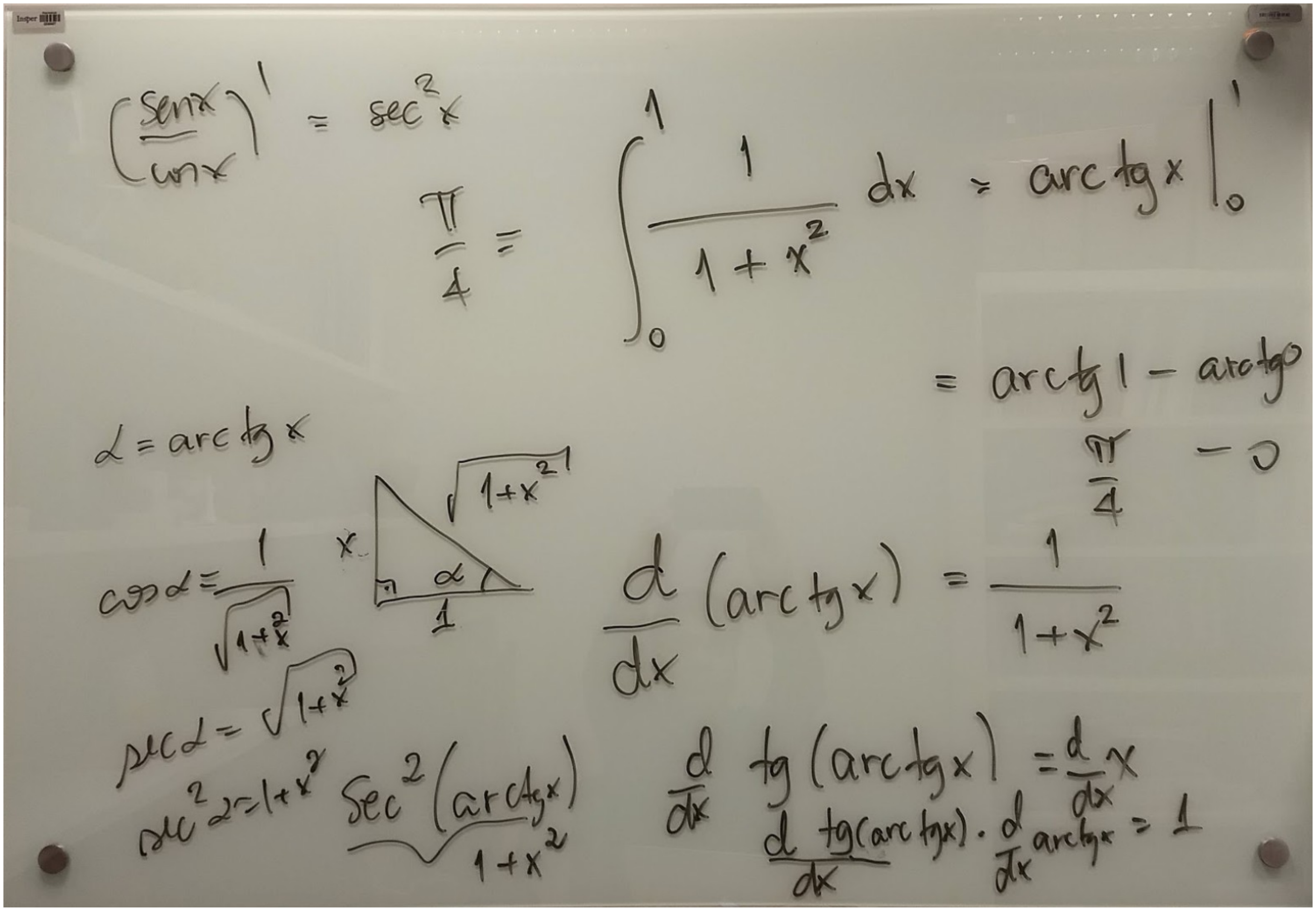
$$\int_{0}^{1} \frac{4.0}{(1+x^2)}\, dx = \pi$$

A integral pode ser aproximada por uma soma de retângulos

$$\sum_{i=0}^{N} F(x_i)\Delta x \approx \pi$$

Cada retângulo tem largura Δx e altura $F(x_i)$ no meio do intervalo i

Explicação da integral by Fábio Orfali



Referências

- Livros:
  - Hager, G. ; Wellein, G. **Introduction to High Performance Computing for Scientists and Engineers**. 1ª Ed. CRC Press, 2010.
- Artigos:
  - Dagum, Leonardo, and Ramesh Menon. "OpenMP: an industry standard API for shared-memory programming." *IEEE computational science and engineering* 5, no. 1 (1998): 46-55.
- Internet:
  - https://www.youtube.com/playlist?list=PLLX-Q6B8xqZ8n8bwjGdzBJ25X2utwnoEG
  - http://www.openmp.org/wp-content/uploads/omp-hands-on-SC08.pdf
  - http://extremecomputingtraining.anl.gov/files/2016/08/Mattson_830aug3_HandsOnIntro.pdf

Insper

www.insper.edu.br