Aula 01 - Introdução e Overview

Supercomputação

Prof. Lícia Sales Costa Lima

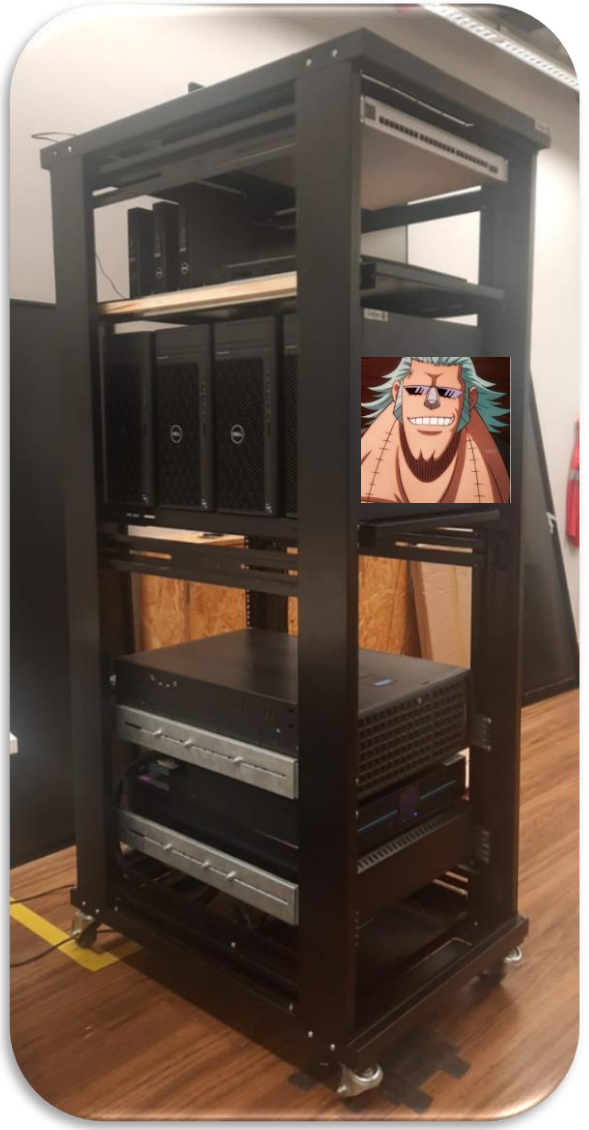
V 1.0

Equipe

Victor Cordeiro
Técnico do lab

Lícia Sales
Professora

Emil Freme
Prof. Assistente



Cluster Franky
Sistema de HPC

Sobre o curso

NOTAS

Atividades - 15%.
APS 1 CPU - 10%
Avaliação Intermediária - 25%
APS 2 GPU - 20%
Avaliação Final - 30%

AULAS:

Segunda
16h30 -- 18h30
Sexta
14h15 -- 16h15

Atendimento

Início 27/02/26
Sexta
12:30 às 14:00

Recursos para o curso

HARDWARE

- Seu notebook
- Cluster Franky
- Cluster Santos Dumont

SOFTWARE

- C++
- SLURM
- OpenMP
- CUDA
- MPI

O que vamos aprender

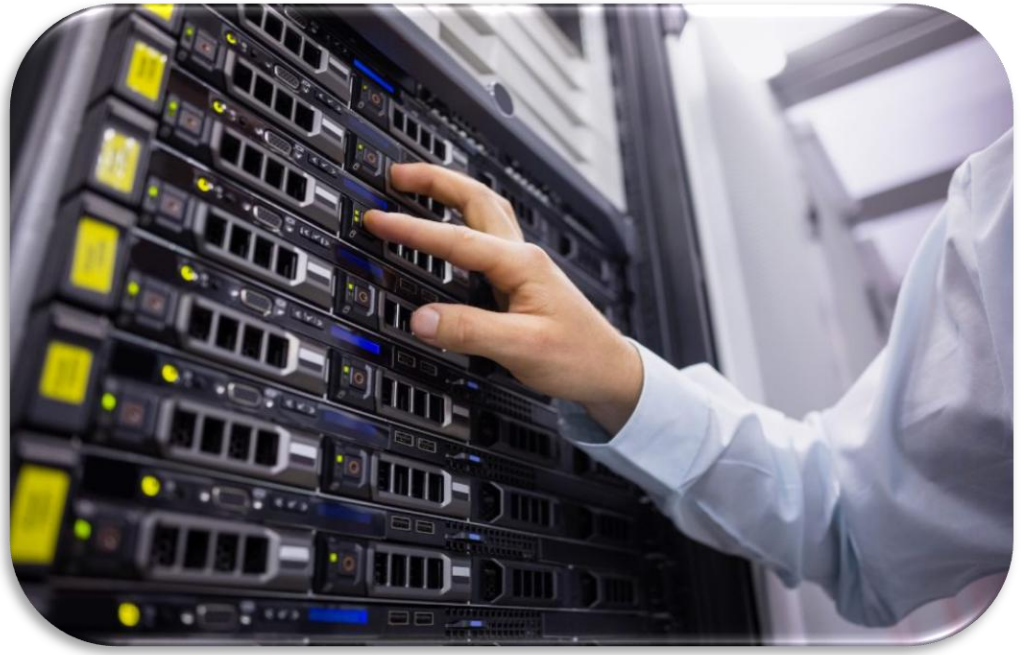
TECNOLOGIA	PARADIGMA	FERRAMENTA
CPU – single core	Sequencial	Compilador C++
SIMD – single core	Paralelismo de dados	Intrínseco
Multicore	Threads	Biblioteca pthreads
NUMA shared memory	Threads	Biblioteca pthreads
GPU	GPU	CUDA
Clusters	Troca de mensagens	MPI

O que vamos aprender

TECNOLOGIA	PARADIGMA	FERRAMENTA
CPU – single core	Sequencial	C++
SIMD – single core	Paralelismo de dados	
Multicore	Threads	
NUMA shared memory	Threads	
GPU	GPU	CUDA
Clusters	Troca de mensagens	MPI

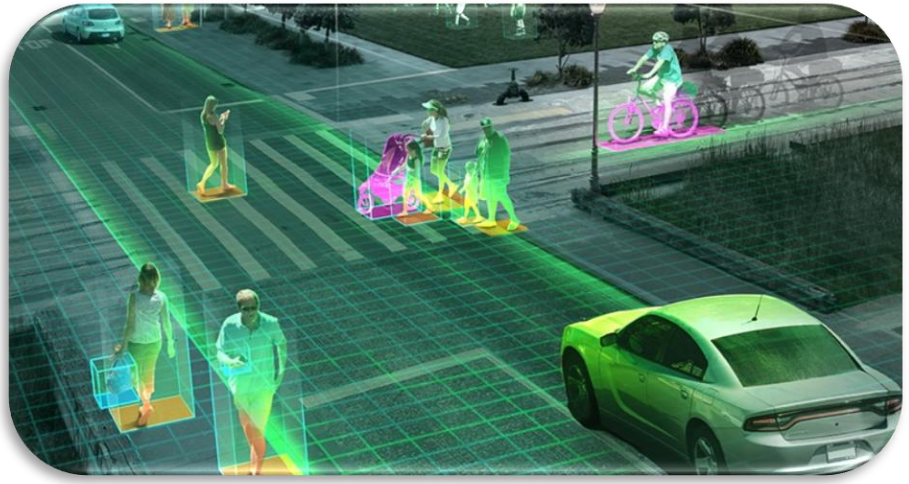
Supercomputação?

- **High-Performance Computing**
- **Computação de alto desempenho**
- **Supercomputação**



Quais são os Problemas de HPC?

- **Grandes:** uma quantidade de dados absurda, que não cabe em um computador de trabalho comum
- **Intensivos:** Realiza cálculos complexos e demorados, demandando horas ou dias de processamento intensivo
- **Combo:** As vezes o problema tem as duas características, tem uma grande quantidade de dados, demanda cálculos intensivos.



Insper

Python x C++

Insper

Compilação vs. Interpretação

- **C++ é compilado;** O compilador traduz o programa para instruções que o processador executa sem intermediários.

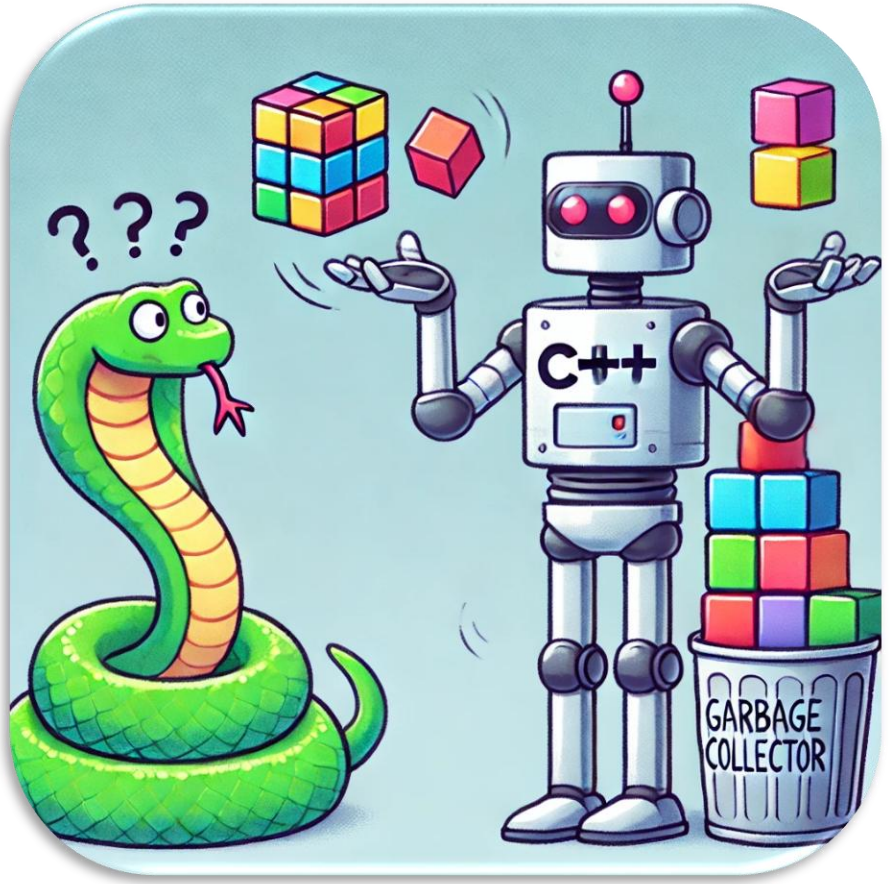
- **Python é interpretado;** Cada instrução é traduzida e executada em tempo real, o interpretador se comunica constantemente com o sistema operacional para solicitar e gerenciar os recursos necessários

Insper

Gerenciamento de Memória

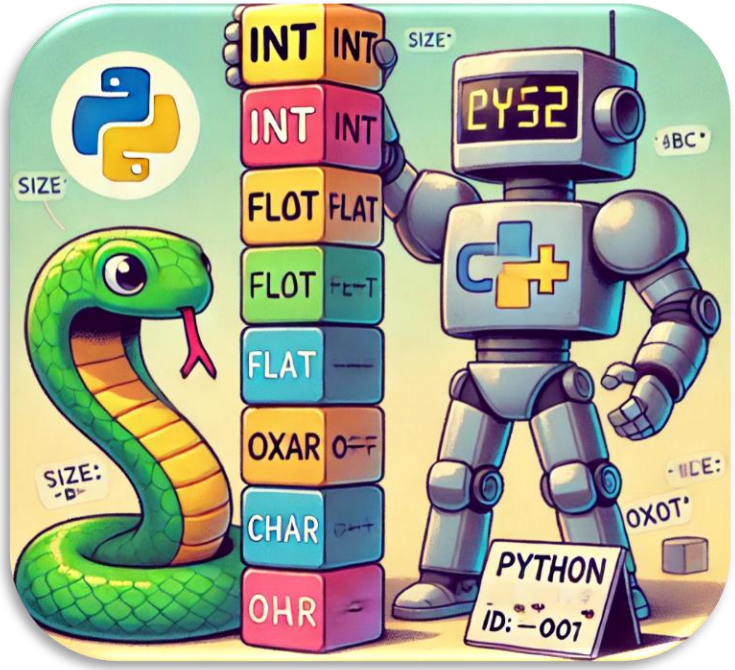
- Em C++, você tem controle explícito sobre alocação e liberação de memória. Isso permite otimizações específicas.

- Em Python, a memória é gerenciada pelo garbage collector, que periodicamente pausa a execução para liberar memória não usada.



Tipagem Estática vs. Dinâmica

- **C++ usa tipagem estática:** os tipos de variáveis são conhecidos e verificados em tempo de compilação.
- **Python usa tipagem dinâmica:** cada variável carrega consigo informações extras sobre o tipo em tempo de execução, o que gera sobrecarga.



Por exemplo, somar dois inteiros em C++ é uma operação direta na ULA da CPU; em Python envolve verificar os tipos, criar objetos temporários e chamar funções internas.

Acesso a Hardware e Paralelismo

- C++ pode acessar diretamente recursos de hardware, instruções SIMD (Single Instruction, Multiple Data), paralelismo via OpenMP, CUDA, MPI.

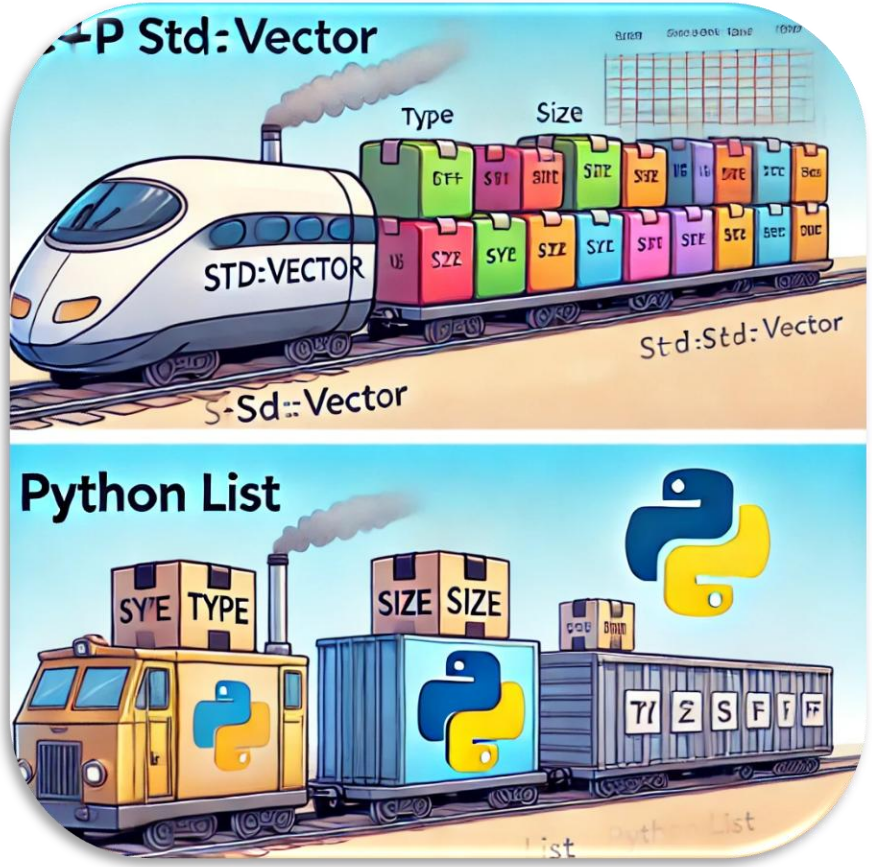
- Python depende de bibliotecas externas (como NumPy, Numba ou PyTorch), que por baixo dos panos usam C/C++ para ter desempenho.

Quando Python é rápido, geralmente está chamando código C++ por baixo dos panos

Overhead de Estruturas Internas

- Em C++, uma lista (std::vector) armazena elementos lado a lado, sem espaços entre eles na memória.

- Em Python, uma lista é um vetor de ponteiros para objetos, e cada objeto inclui metadados (tipo, referência, tamanho) causando maior consumo de memória e mais acessos indiretos.



Site SuperComp

Referências

https://insper.github.io/supercomp/

https://github.com/isocpp/CppCoreGuidelines

AMATH 483/583 Sp 22 U of Washington Xu Tony Liu