Insper

SuperComputação

Aulas 02/03 – Implementação e desempenho

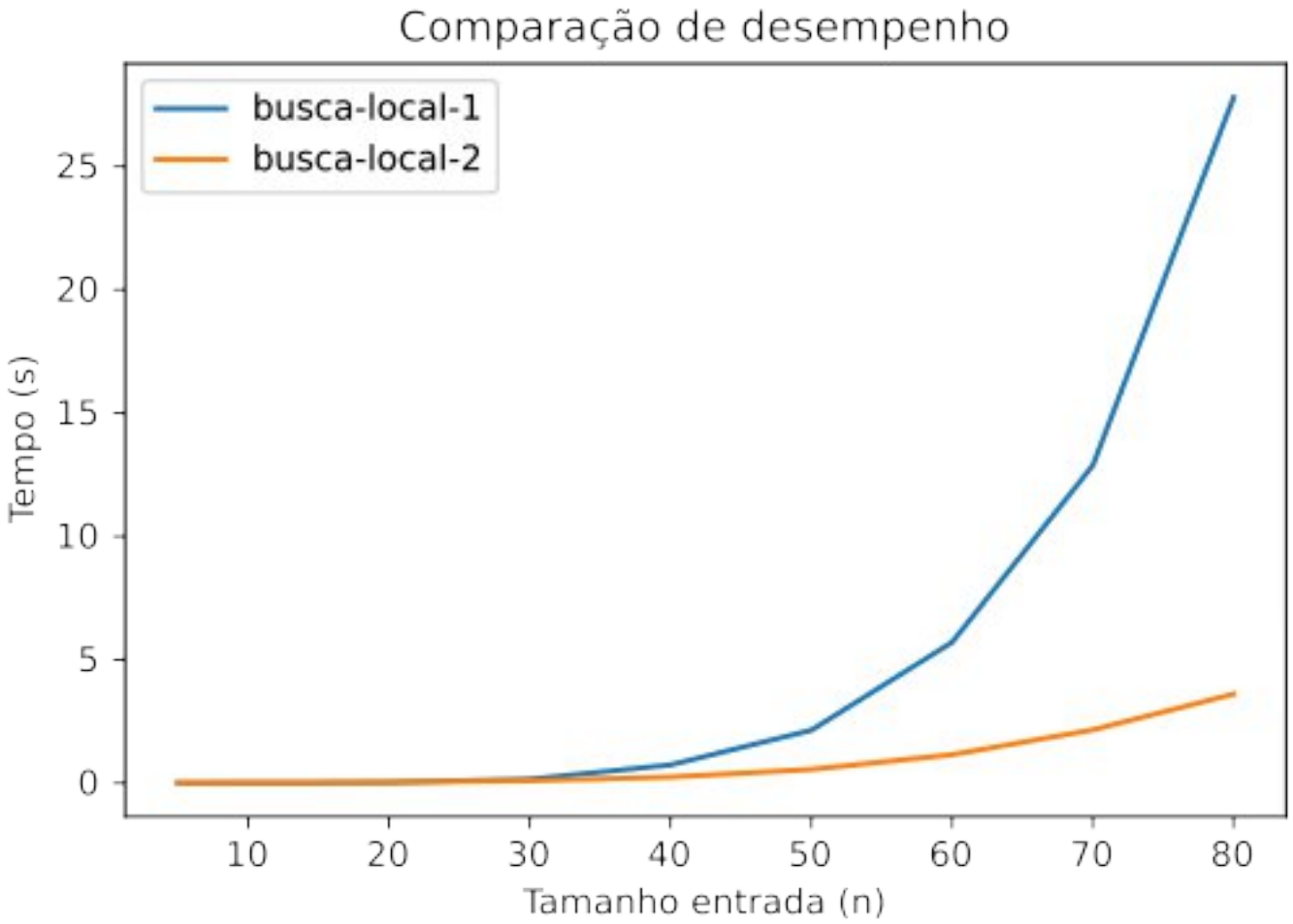
2021 – Engenharia

Igor Montagner <igorsm1@insper.edu.br>
Antônio Selvatici <antoniohps1@insper.edu.br>

Solução de alto desempenho

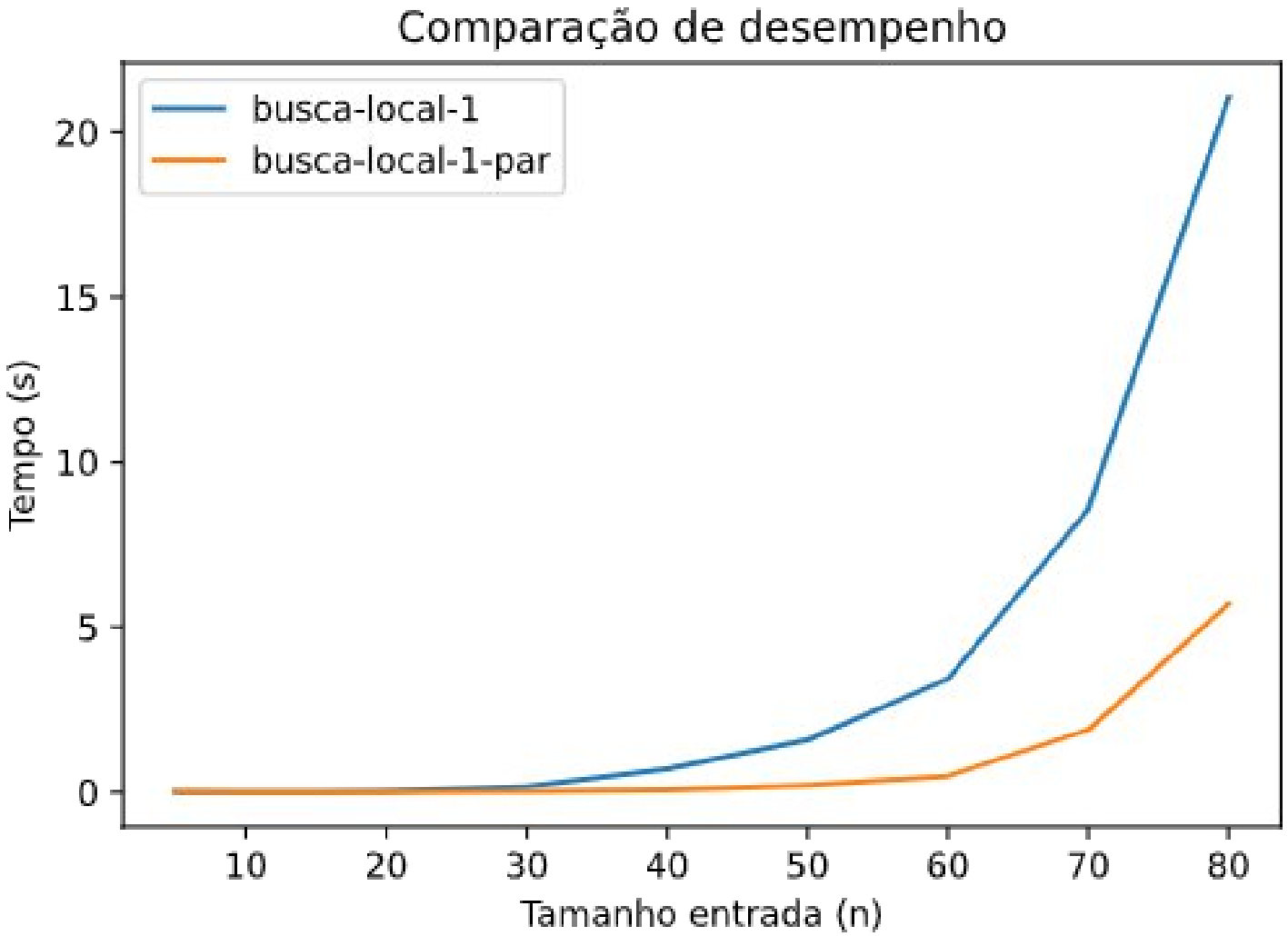
1. Algoritmos eficientes
2. Implementação eficiente
    - Cache, paralelismo de instrução
    - Linguagem de programação adequada
3. Paralelismo

Discussão 1 - algoritmo importa!



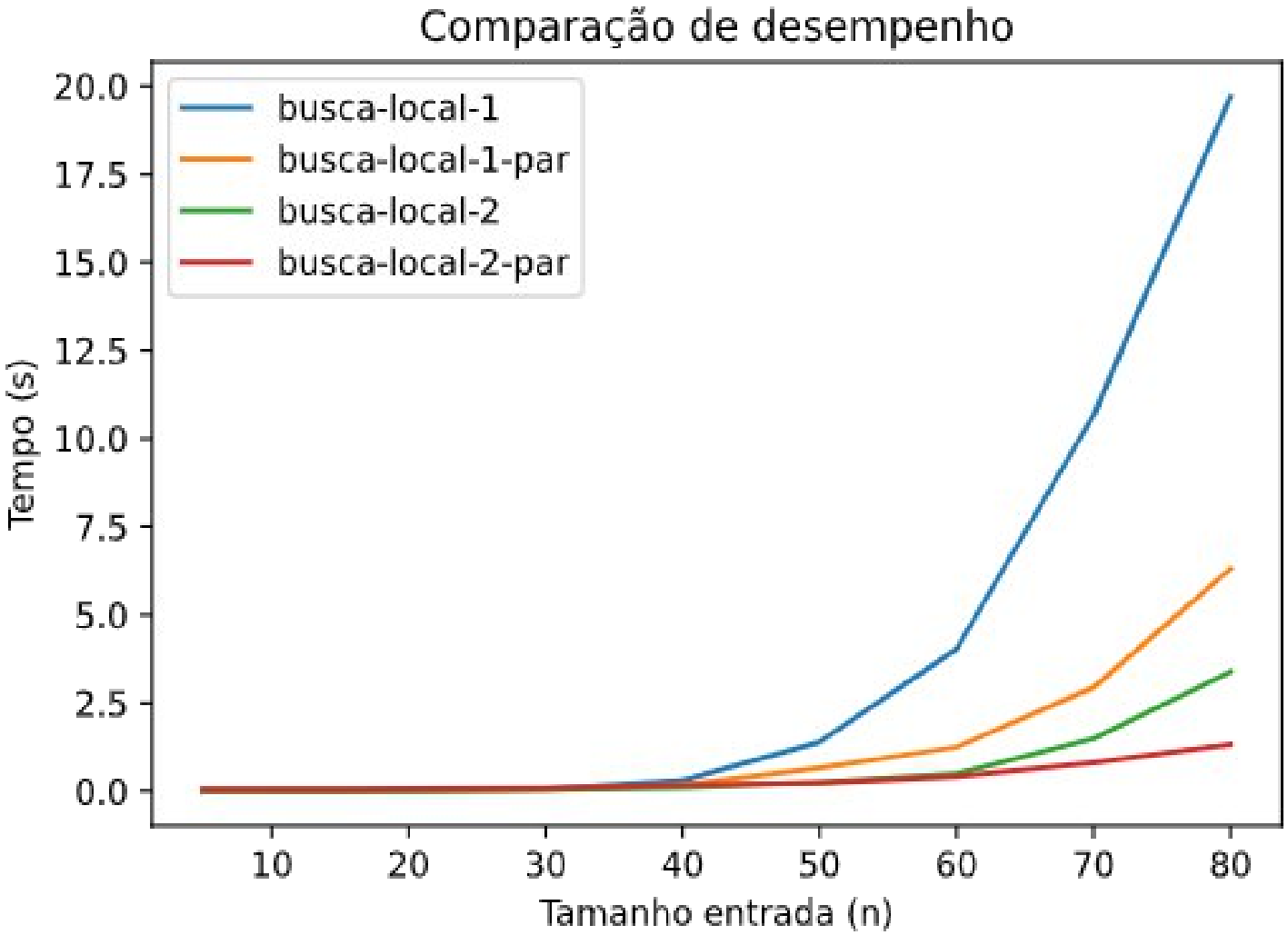
Insper

Discussão 2 - paralelismo importa!



Insper

Discussão 2 - mas não sozinho....



Insper

Visão geral do curso

Estratégias para resolução de problemas difíceis

- Complexidade Computacional
- Problemas NP-completo
- Heurísticas
- Busca local e global
- Algoritmo aleatorizados

- Paralelismo
- Sistemas Multi-core
- GPU
- Projeto de programas paralelos

Processamento paralelo

Insper

Hoje

- Implementação vs algoritmo

Implementação e Algoritmo

Algoritmo

Algoritmo

"Sequência finita de passos executáveis que resolve um problema"

Implementação

Implementação

"Transformação de um algoritmo em um programa executável"

Quanto tempo um programa demora?

Quanto tempo um programa demora?

Algoritmo:

- complexidade computacional (classes de algoritmos)
- estruturas de dados (abstratas)
- **provado matematicamente, não muda**

Implementação:

- medido em segundos, para uma certa entrada
- tecnologia usadas (linguagens de programação, bibliotecas)
- hardware usado
  - Clock de CPU e RAM, tamanho do Cache
  - # de núcleos
- **imprecisão**

Quanto tempo um programa demora?

SuperComputação começa quando Desafios de Programação acaba

Dado um "bom" algoritmo, vamos definir

- **linguagem de programação adequada**
- tipo de paralelismo indicado
- implementação paralela eficiente

Linguagens de Alto Desempenho

- Fortran
- C++
- Julia (computação distribuída)

C++

- Oferece sintaxe mais expressiva que C
- Mantém velocidade
- Flexível

Atividade prática

Iniciando com C++

1. Implementar algoritmos simples usando recursos de C++
2. Definir quando fazer passagem de argumentos por cópia ou por referência
3. Analisar um programa para identificar possíveis pontos de melhorias

Fechamento

Nossa otimização não funcionou, por que?

- Medir quanto tempo cada função demora?

- Nossa função ficou mais rápida? Se sim, quanto? Se não, por que?

- **Como medir "quantidade de trabalho feito"?**

Insper

www.insper.edu.br