



**PUC Minas**

**LICAP**

Laboratório de Inteligência Computacional Aplicada

# **PLANEJAMENTO DE CAPACIDADE, MODELAGEM E AVALIAÇÃO DE DESEMPENHO DE SISTEMAS COMPUTACIONAIS**

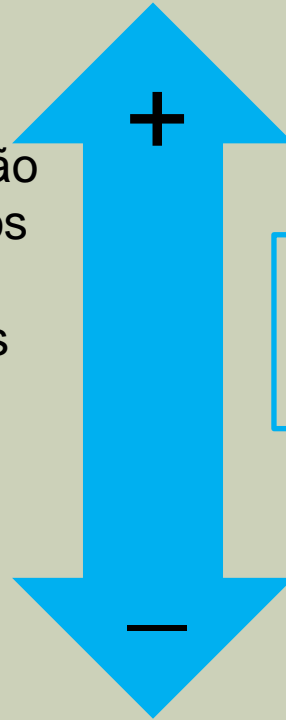
## **TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO DE SISTEMAS COMPUTACIONAIS**

Equipe MAD

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

Diversas são as técnicas que podem ser utilizadas para avaliar o desempenho de sistemas computacionais. Todas as técnicas se complementam, mas uma classificação de acordo com a sua precisão, custo e tempo para sua implementação pode ser apresentada.

- Modelos de simulação
- Modelos matemáticos
- Regressão linear
- Programas sintéticos
- Monitoramento
- Kernel
- Benchmark
- “Eu acho”



- Precisão
- Custo
- Tempo Implantação

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## 1) “Eu acho”

Esta técnica embora mais imprecisa e a mais barata é a mais utilizada na prática.

A precisão desta técnica está associada à experiência dos responsáveis pela implantação do sistema.

Devido a sua imprecisão esta técnica deve ser voltada unicamente para proposta da configuração inicial do sistema. Para proposta de atualização, existem técnicas mais eficazes.

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## 2) Benchmark

É conhecida também como a técnica da comparação.

Tipicamente é destinada para comparar/analisar dispositivos isolados (processadores, placas de vídeo, placas de rede, compiladores, etc.).

Atualmente é aplicada para comparar o desempenho de sistemas nas Nuvens (Google, Microsoft, Amazon etc). Por exemplo para avaliar Scaling up ou Scaling out.

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## ■ Benchmark

A idéia de escalar recursos nas nuvens quando a carga de trabalho muda pode ser intuitivo. Aumentar infraestrutura quando a carga aumenta e diminuindo a infraestrutura quando a carga diminui.

Porém existem duas estratégias para incrementar a infraestrutura :

- **Scaling out (escalonamento horizontal):** consiste em aumentar recursos em paralelo espalhando de forma balanceada a carga de trabalho.
- **Scaling up (escalonamento vertical):** consiste em aumentar a velocidade para gerenciar grandes cargas de trabalho. Isto é, movimentando sua aplicação para um servidor virtual (VM) de 2 CPU para outro de 3 CPU.
- **Scaling down (redução do escalonamento):** refere-se a decrementar os recursos do sistema.

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## ■ Benchmark

A comparação de desempenho de dispositivos deve ser feita em um ambiente controlado.

- **Ex.:** Para avaliar um compilador para diferentes Sistemas Operacionais
  - Mesmo processador
  - Mesmo S.O.
  - Mesmo barramento de memória
  - Mesma otimização do compilador
  - Etc.

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## ■ Benchmark

Esta técnica precisa responder às seguintes questões:

- Qual é o ambiente experimental controlado considerado?
- Quais são as variáveis não controladas?
- Como as variáveis não controladas afetam as conclusões da análise?

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## 3) Kernel

Esta técnica é destinada para avaliar exclusivamente processadores.

Esta técnica é utilizada quando não é disponível ou acessível “fisicamente” o processador.

Podemos considerar que esta técnica possui duas versões: A clássica e a combinada com a técnica dos programas sintéticos.



# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## ■ Kernel – versão clássica

Esta técnica consiste em identificar o módulo kernel da aplicação e obter, deste, seu código assembly.

A partir do código assembly, é possível determinar os tempos de execução de acordo com a especificação do fabricante.

### Procedimentos:

1. Identificar o módulo Kernel da aplicação

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

- Identificando o módulo kernel de uma aplicação.

Início	Interação	Tempo gasto
Modulo 1	10	1s
Modulo 2 <KERNEL	100	200s
Modulo 3	1000	10s
Fim	1	5s

## 2. Codificar o módulo Kernel em assembly

Alto nível	Interação
...	LDD Ax, Hx0018
<b>a = a +1</b>	PST Ac
...	ADD Ax, Ac
...	MOV Ac Hx0018

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

3. Calcular o tempo total de execução do módulo kernel para cada processador em análise de acordo com os manuais do fabricante

Alto nível	Interação	Tempo Gasto Processador 1	Tempo gasto Processador 2
...	...	...	...
	LDD Ax, Hx0018	2,5 ns	2,8 ns
a = a +1	PST Ac	1,2 ns	1,2 ns
	ADD Ax, Ac	3,2 ns	4,5 ns
	MOV Ac Hx0018	2,5 ns	2,8 ns
...	...	...	...
Total		4500 ns	5670 ns

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

## ■ Kernel – versão com Programas Sintéticos

Esta técnica tem surgido pela falta de profissionais com conhecimento profundo de assembly, o que limita a aplicação da versão original.

É proposto então, criar um **Módulo Kernel Sintético (MKS)** que seja similar ao **Módulo Kernel Original (MKO)** e com este realizar rápidos experimentos sobre Hardwares específicos (físicos)

Módulo  
Kernel  
Original

Módulo  
Kernel  
Sintético

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

- Considerando um Processador específico X.
- Calcular o tempo de execução ( $T_{ko}$ ) do MKO, para, por exemplo,  $N = 1000$
- A idéia é construir um programa sintético (MKS) similar ao MKO, respeitando vários critérios:

Alto nível	MKO	condição	MKS
Tempo de Processamento	$T_{ko}$	=	$T_{ks}$
Estrutura de dados	Listas, Vet, Mat,..	$\approx$	Listas, Vet, Mat,..
Tipos de variáveis	Type Varr	$\approx$	Type var
Ordem de complexidade	$O(N)$	=	$O(N)$
Contextos	Contexto	$\neq$	Contexto

# TÉCNICAS PARA AVALIAÇÃO DE DESEMPENHO

- O próximo passo é construir um código executável do programa sintético kernel e utilizar este para realizar testes rápidos em diversos processadores.

