



Medindo e Modelando o Desempenho de Aplicações em um Ambiente Virtual

Gustavo Lopes Rodrigues

Homenique Vieira Martins

Lucas Santiago Oliveira

Rafael Amauri Diniz Augusto

Sumário

01 – Introdução

- O que é virtualização?
- Vantagem da virtualização
- Mas qual é a proposta do artigo?

02 – Ambiente Virtual Xen

- O Ambiente Virtual Xen
- Arquitetura Xen

03 – Estudo de caso

Validação do modelo

04 – Conclusão

Contribuições e
Trabalhos futuros



01

Introdução

O QUE É VIRTUALIZAÇÃO?

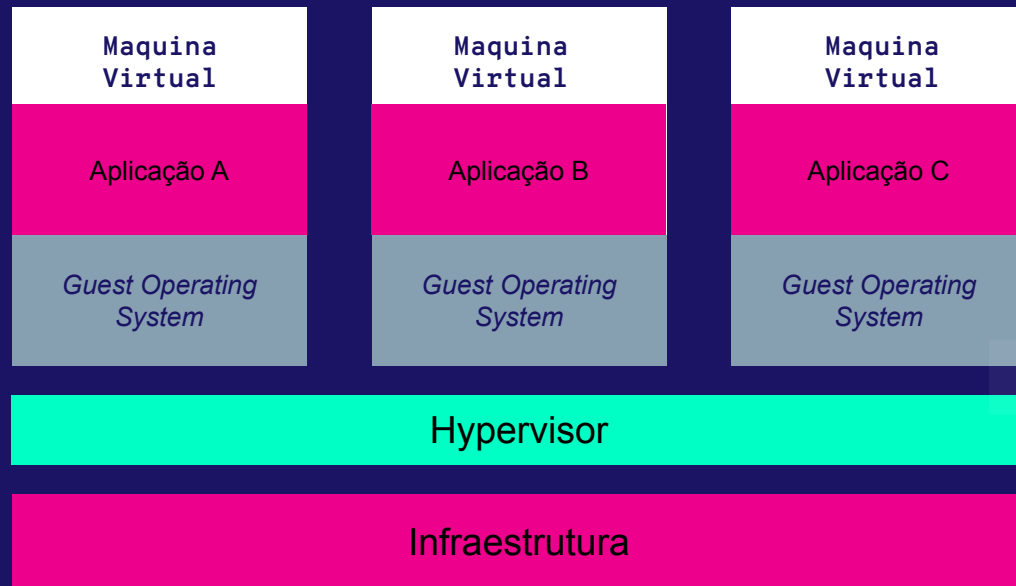
Técnica para particionar recursos de uma máquina em múltiplos ambientes.



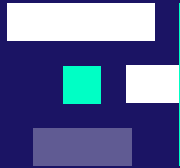
Computador sem Virtualização



Computador com Virtualização

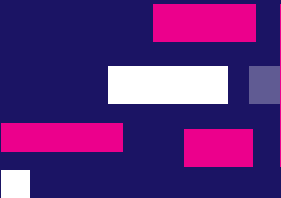
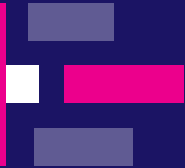
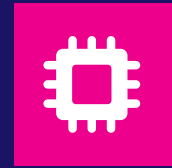


VANTAGENS DA VIRTUALIZAÇÃO



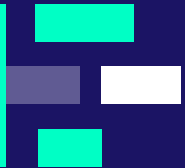
Menor
complexidade

Múltiplas
aplicações



Custos de
gerenciamento e
aquisição

Consumo de
energia





Apesar das vantagens...

Não é algo simples fazer a migração de um ambiente físico para virtual, pois antes é necessário ter uma estimativa do desempenho para suportar os serviços e manter o nível do SLA.

Aspectos a serem vistos antes da migração

- Respeitar acordos de serviço SLA;
- Plataforma de hardware mais adequada para o ambiente virtual;
- Melhor configuração para a plataforma de hardware;

Mas qual é a proposta do Artigo?

No artigo é proposto um modelo analítico para prever o desempenho que aplicações em execução em ambientes reais vão adquirir se forem migradas para o ambiente virtuais.





02

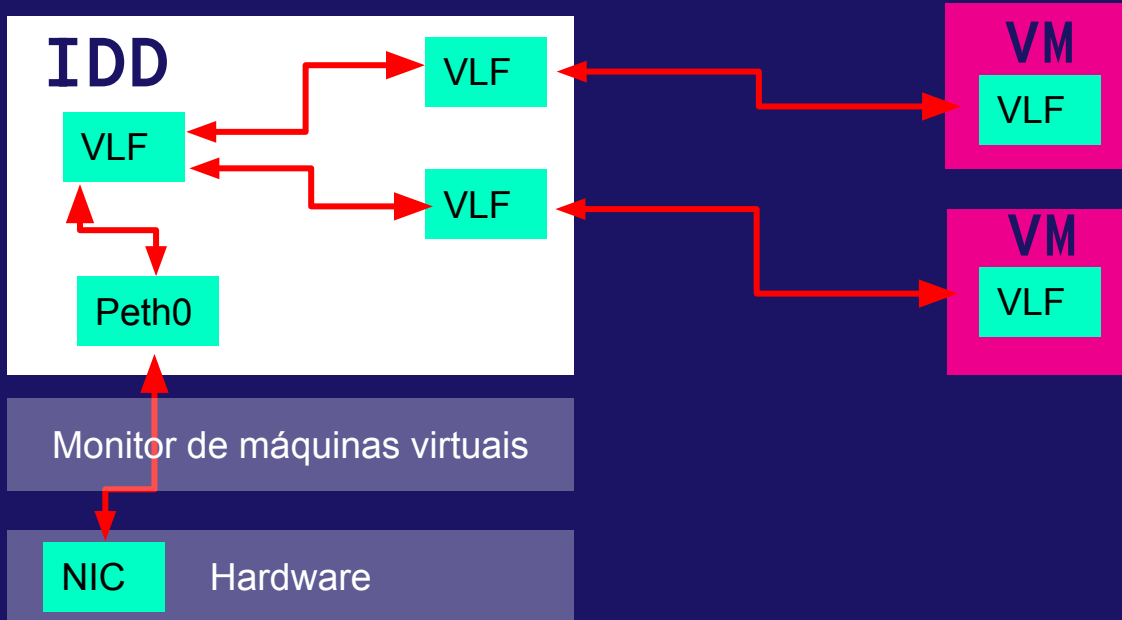
Ambiente Virtual Xen

0 Ambiente Virtual Xen

Um sistema mediador entre máquinas virtuais e o hardware, permitindo diversas máquinas executarem de forma concorrente em um único ambiente físico.



Arquitetura Xen



03

Estudo de caso

Arcabouço para Previsão de Desempenho

- A fim de medir o desempenho são definidas diferentes fórmulas para diferentes aspectos do desempenho da virtualização.
- A partir de um caso base, o modelo calcula tempo de resposta, taxa de saída de requisições e utilização de recursos.

Terminologia

SÍMBOLO	DESCRIÇÃO
N	# de VMs no ambiente virtual
K	# de recursos diferentes no ambiente virtual
N_{cpu}	# de CPUs disponíveis no ambiente virtual
B_k^i	Tempo ocupado do recurso k para a classe i no ambiente real
D_k^i	Demanda do recurso k para a classe i no ambiente real
λ^i	Taxa de chegada de requisições na vm_i
S_k^i	<i>Slowdown</i> da classe i no recurso k
P_k^i	<i>Speedup</i> da classe i no recurso k
cap_i	Porção do total de recursos que vm_i pode utilizar
$B_k^{vm_i}$	Tempo ocupado do recurso k para executar vm_i
$U_k^{vm_i}$	Utilização do recurso k pela vm_i do ponto de vista da vm_i
$D_k^{vm_i}$	Demanda do recurso k pela vm_i
$R_k^{vm_i}$	Tempo de residência médio do recurso k pela vm_i

Terminologia

cp^i	Tempo de CPU para o IDD processar um pacote da classe i
pr^i	Número de pacotes por requisição da classe i
$D_{CPU,i}^{vm_0}$	Demanda de CPU no IDD devido a atividade de E/S na vm_i
$U_{CPU}^{vm_0}$	Utilização de CPU no do IDD
$R_{CPU,i}^{vm_0}$	Tempo de Residência no IDD para a classe de aplicações i
$U_{CPU,i}^*$	Soma das utilizações de todas as VMs, exceto vm_i e o IDD
λ_{max}^i	Taxa máxima de chegada requisições

Slowdown: Uma armadilha da virtualização

- Desempenho nunca vem de graça.
- Utilizar uma VM sempre vai trazer algum tipo de *overhead*.
- O Slowdown pode ser calculado pela equação 1 abaixo:

$$S_k^i = \frac{B_k^{vm_i}}{B_k^i}$$

Calculando a Demanda D

- Recursos diferentes de um computador são usados em quantidades diferentes por cada processo.
- Com base no uso no sistema real, no slowdown e no speedup relativo da máquina que vai hospedar a VM com relação à máquina real, podemos calcular a Demanda “D” que um recurso K terá para cada VM i hospedada no sistema. “O speedup é uma variável essencial para o modelo porque permite aos usuários escolherem entre diferentes plataformas de hardware para implantar seus ambientes virtuais”.
- A equação 2 abaixo demonstra isso:

$$D_k^{vm_i} = D_k^i \cdot \frac{S_k^i}{P_k^i}$$

Calculando o Uso U

- O uso de uma VM é definido como o uso de um recurso K.
- A partir da demanda D de uma VM i sobre um recurso K, do número de requisições λ , e do número de recursos da CPU alocados para a VM i (CAPi), é possível calcular o uso. "CAPi" é entendido como todos os recursos da CPU alocados para a VM, que pode ser interpretado na sua forma mais básica como o número de núcleos alocados.
- Com a equação 3 abaixo o uso pode ser estimado:

$$U_k^{vm_i} = \frac{\lambda^i \cdot D_k^{vm_i}}{cap_i}$$

Calculando o Tempo de Residência R

- Com as equações 2 e 3 é possível calcular o tempo de residência de uma VM i sobre o recurso K .
- Reflete sobre a quantidade de tempo que a VM ocupa o recurso K .
- O tempo de residência " R " pode ser calculado com a equação 4 abaixo:

$$R_k^{vm_i} = \frac{D_k^{vm_i}}{1 - U_k^{vm_i}}$$

Sobre o Tempo de Residência R

- “Com o tempo de residência é possível calcular o tempo de resposta de uma aplicação executando em uma VM”.
- O tempo de resposta de uma aplicação “i” pode ser calculado a partir do tempo de residência R de i sobre todos os recursos K e do tempo de residência médio no IDD.
- “Estimar o tempo de resposta médio pode ser bastante útil para manter os níveis do acordo do SLA”.

Entendendo o IDD

- É utilizado para gerenciar as operações de I/O entre as VMs.
- É necessário saber de antemão o custo de processar um pacote i (cp), o número médio de pacotes por requisição (pr) e o speedup da CPU (P_{cpu}).
- A demanda de CPU do IDD é entendida com a fórmula 5 abaixo:

$$D_{CPU,i}^{vm0} = \frac{cp^i \cdot pr^i}{P_{CPU}^i}$$

Uso do IDD

- O uso da CPU pelo IDD (que é entendido como a VM 0) pode ser expresso pela fórmula 6 abaixo:

$$U_{CPU}^{vm_0} = \frac{\sum_{i=1}^N (D_{CPU,i}^{vm_0} \cdot \lambda^i)}{cap_0}$$

- As variáveis envolvidas são a demanda do IDD sobre cada CPU, o número de requisições em cada VM i e o número de CPUs alocadas para o IDD.

Tempo de Residência no IDD

- O tempo de residência do IDD na CPU pode ser expresso pela equação 7 abaixo:

$$R_{CPU,i}^{vm_0} = \frac{D_{CPU,i}^{vm_0}}{1 - U_{CPU}^{vm_0}}$$

- As variáveis envolvidas são a demanda do IDD sobre a CPU, e o uso da CPU pelo IDD.

Analizando Limites

- Analisar os limites do modelo e como as variáveis interagem pode nos dar pistas sobre possíveis gargalos na máquina, especialmente quando colocando-as em função do número de requisições.
- A partir do modelo apresentado, os gargalos podem ser apresentados em dois locais: no IDD ou em uma VM i .
- Em base, o IDD pode se tornar o gargalo se “a soma da utilização da CPU dedicada a cada VM atingir cap_0 ”, significando que a CPU alocada ao IDD está sendo usada para ciclos de clock em uma VM.
- A VM pode se tornar o gargalo se “o valor máximo de λ_i for limitado pelo recurso com maior valor de demanda de serviço”, significando que a VM está recebendo mais requisições do que um recurso K é capaz de processar. Isso causa sobreuso do recurso e faz a VM virar o gargalo do sistema.

Configurações do Hardware

OS Linux (Xeno Linux)

CPUs Intel Xen 64 bit 3.2 GHz

RAM 2 GB

Disco 7200 RPM 8 MB de cache L2

Configurações da VM

OS Linux (Xeno Linux)

RAM 512 MB

IDD 512 MB

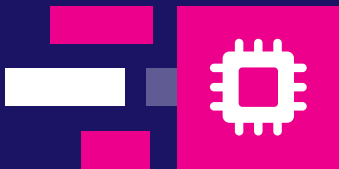
Teste com mais de duas VM

Cada uma fica com
250 MB

Ferramentas utilizadas



`/proc`: para medir tempo ocupado de CPU e o número de pacotes.



Xenoprof: medir número de instruções executadas e falhas (*misses*) nas caches do processador e na TLB1.

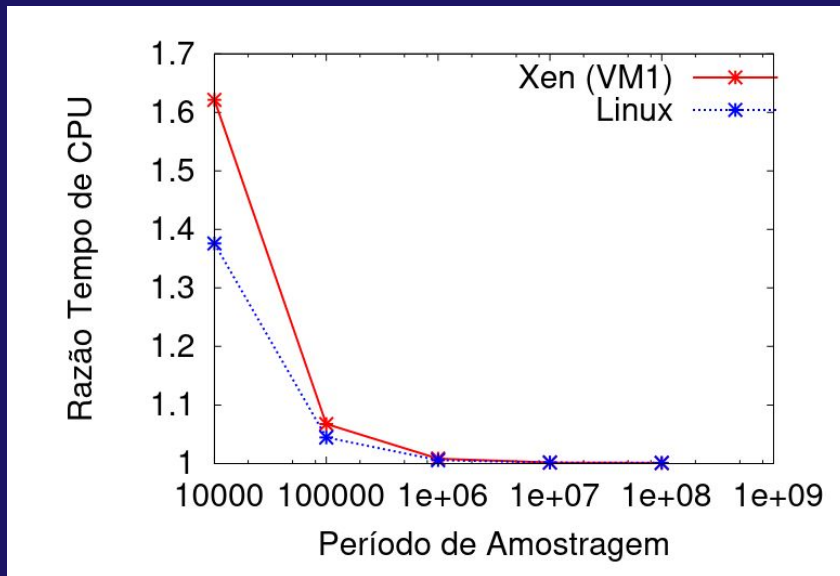
Premissa da Modelagem

- Cada VM executa uma aplicação de classe diferente.
- Classe de aplicações representa um tipo de aplicação para o qual o *overhead* da virtualização foi calculado.
- Cargas geradas no sistema por aplicações de uma mesma classe devem possuir características semelhantes

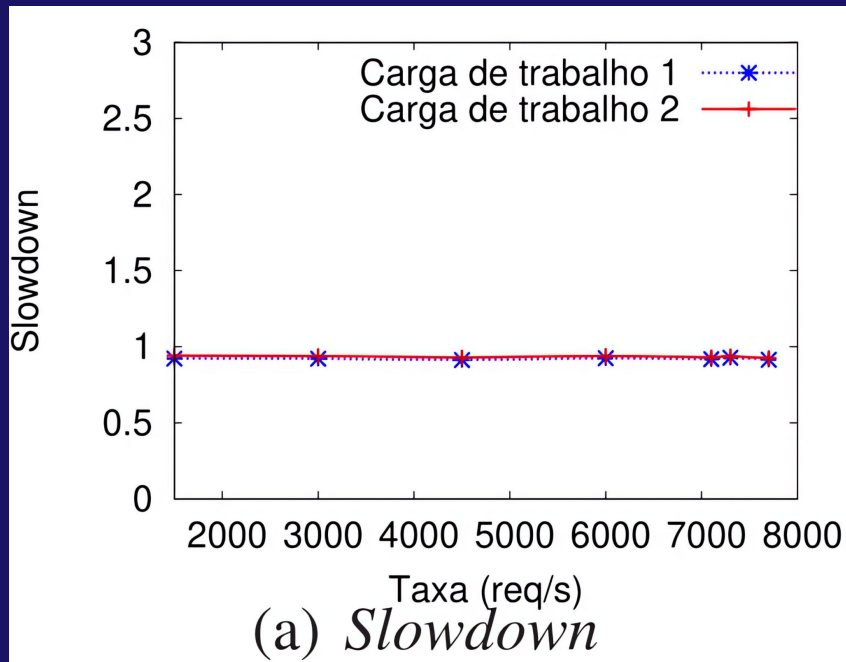
Observações Sobre os Testes

O período de amostragem foi pensado para não ter influência dos softwares usados para as métricas, porém mesmo assim, as máquinas virtuais foram penalizadas devido à virtualização.

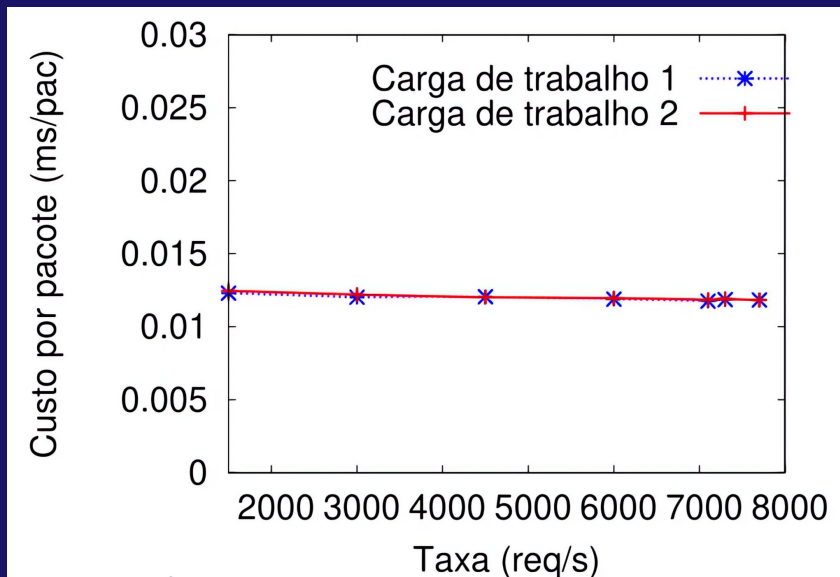
Overhead do Xenoprof e do Oprofile



Análise de *Slowdown* e Custo por Pacote

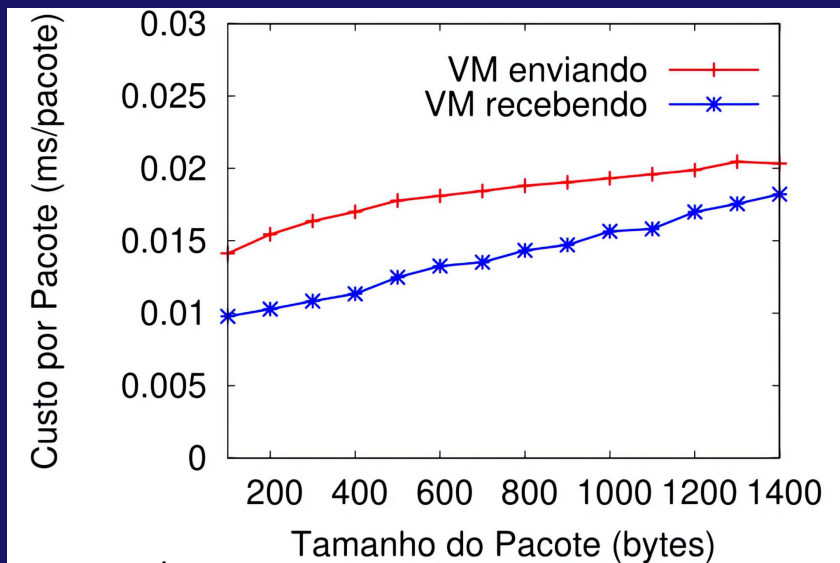


Análise de *Slowdown* e Custo por Pacote



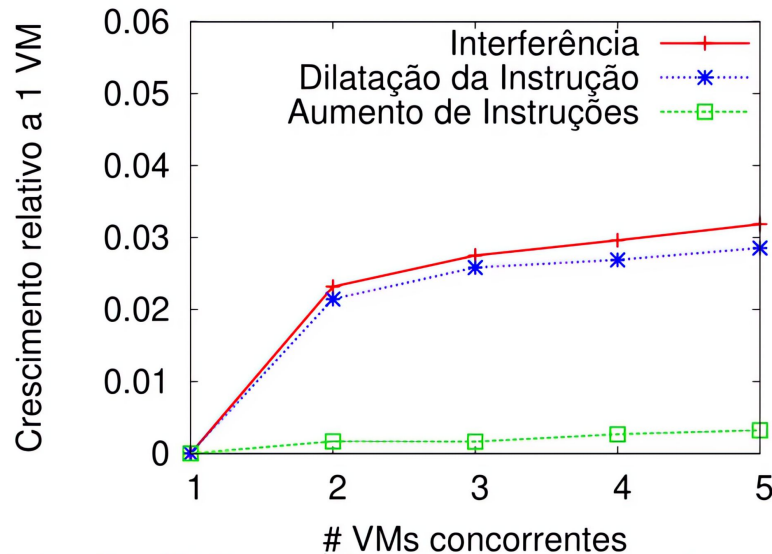
(b) cp^i para as duas cargas de trabalho analisadas

Análise de *Slowdown* e Custo por Pacote



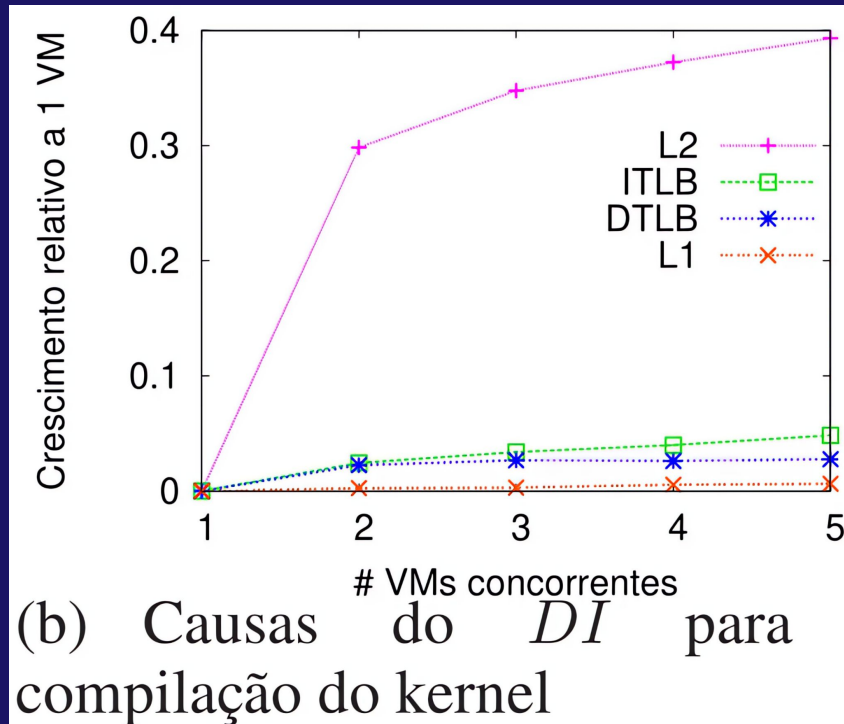
(c) cp^i para diferentes tamanhos de pacotes

Interferência e suas causas

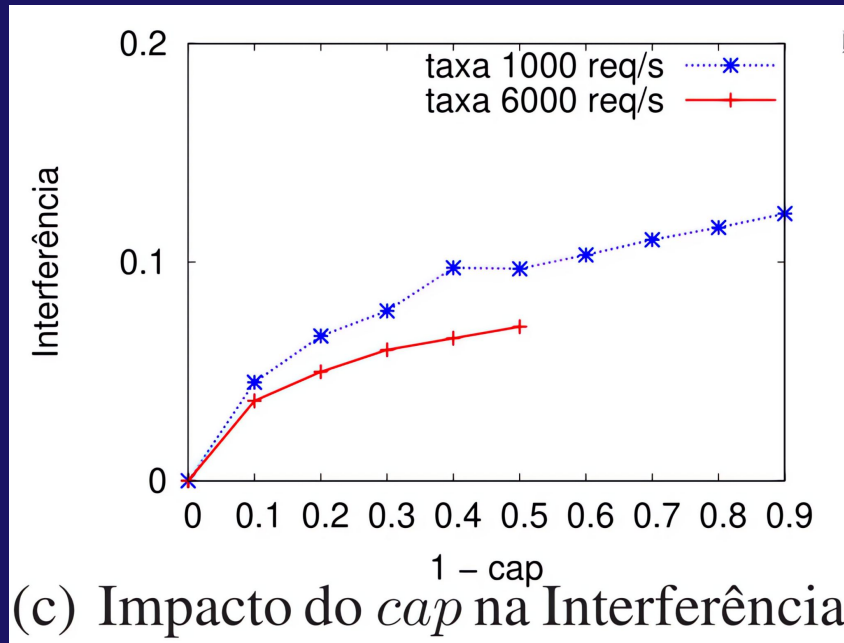


(a) I , DI e AI para a compilação do kernel

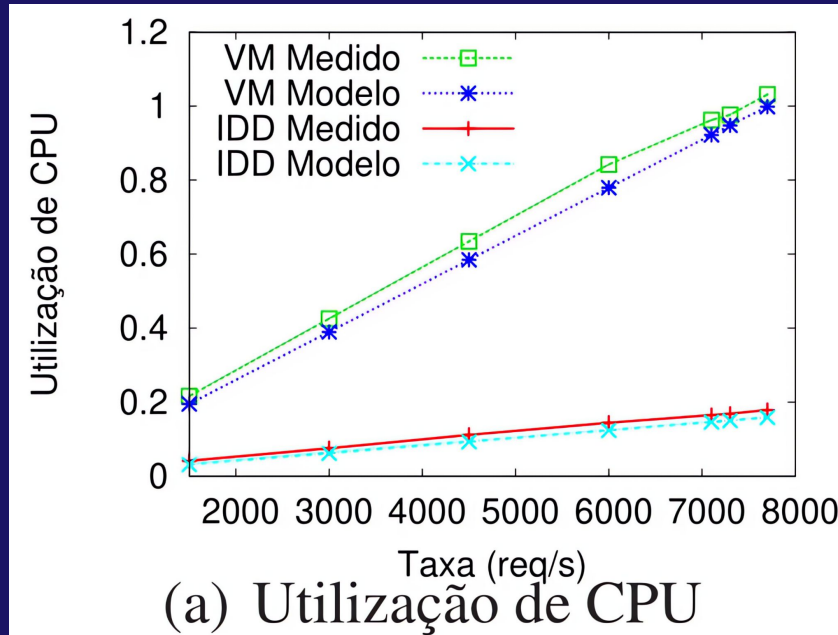
Interferência e suas causas



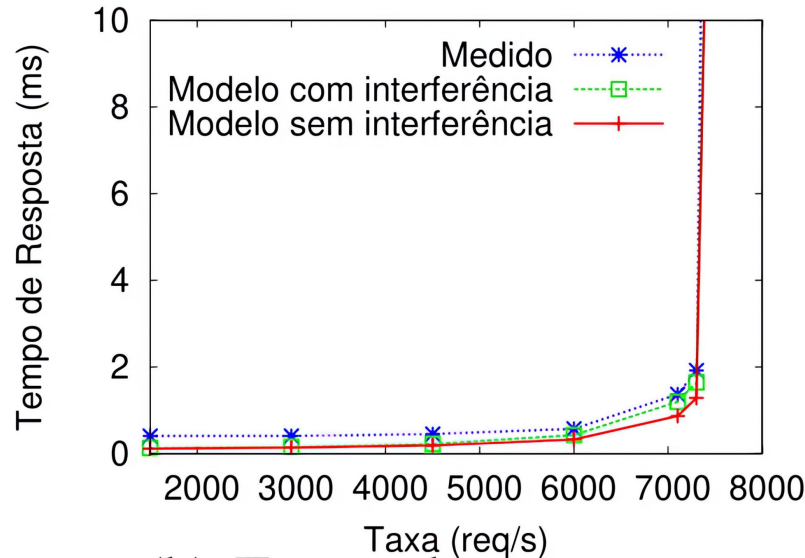
Interferência e suas causas



Sumário da análise de desempenho do servidor Web

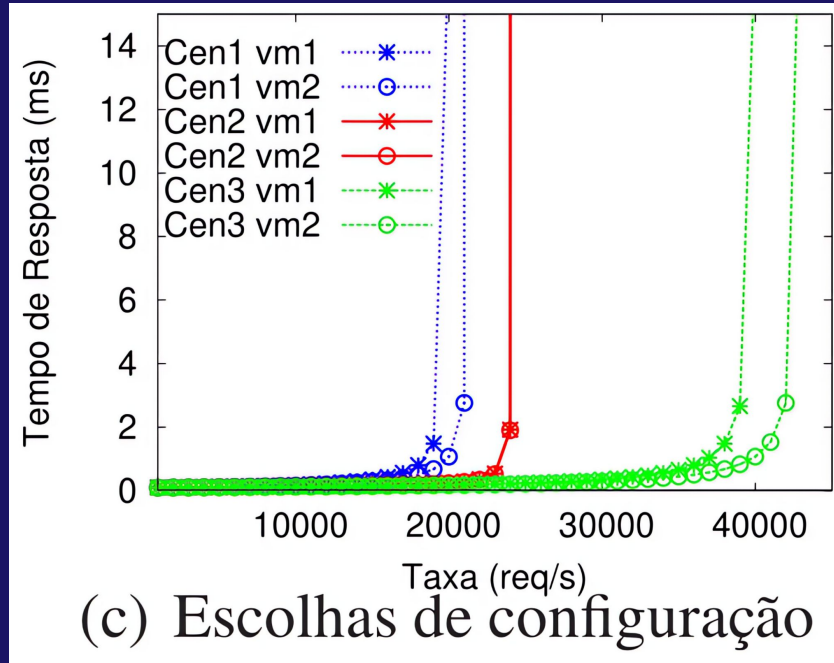


Sumário da análise de desempenho do servidor Web



(b) Tempo de resposta

Sumário da análise de desempenho do servidor Web



Passos para o uso dos modelos de previsão de desempenho

1: Criação da base de dados com:

- S_k^i para $i, 1..N$ e $k, 1..K$
- cp^i e desempenho relativo de hardware para máquinas diferentes

2: Medição das aplicações alvo no ambiente virtual, coletando:

- D_k^i e pr^i
- Taxa de chegada de requisições (λ^i)
- Configuração de hardware para o ambiente não virtual

3: Escolha da plataforma de hardware:

- Definição de N_{cpu}
- Cálculo de P_k^i e ajuste de cp^i baseado em informações de hardware dos passos anteriores

4: Escolha da classe de aplicações que melhor representa a aplicação alvo:

- Definição de S_k^i

5: Definição de cap^i , para $i, 1..N_{CPU}$

6: Cálculo de $D_k^{vm_i}$, $U_k^{vm_i}$ e $R_k^{vm_i}$ para cada i e k

04

Conclusão

Conclusão

Proposto arcabouço para planejamento de capacidade prático. Capaz de:

- Determinar a melhor configuração e particionamento dos recursos de uma plataforma de hardware virtualizada.
- Capaz de prever o desempenho de aplicações em ambientes virtuais.

Overhead de virtualização aumenta quando VMs compartilham a mesma CPU. O *overhead* não é considerado por escalonadores.

O estudo atual pode ajudar a criar mecanismos capazes de diminuir o custo da virtualização e direcionar futuros modelos de desempenho.

Trabalhos futuros

- Validar modelos para outras cargas de trabalho.
- Desenvolver modelos para aplicações de classe fechada.
- Avaliar arcabouço para diferentes ambientes virtuais.
- Avaliar o novo hardware da Intel para ambientes virtuais.
- Desenvolver uma ferramenta que utilize o modelo apresentado no artigo.

Bibliografia

Artigo: Medindo e Modelando o
Desempenho de Aplicações em um
Ambiente Virtual

Obrigado pela atenção!

CREDITS: This presentation template was created by
Slidesgo, including icons by Flaticon, and
infographics & images by Freepik.