

Medindo e Modelando o Desempenho de Aplicações em um Ambiente Virtual

Caio Igor
Carolina Lima
Heleny Bessa
Luana Lins

CONSIDERAÇÕES INICIAIS

01

VIRTUALIZAÇÃO

02

PARAVIRTUALIZAÇÃO

03

OVERHEAD

04

SLOWDOWN

05

GARGALOS

01

VIRTUALIZAÇÃO

Particionar
Ambientes
Indireção



02

PARAVIRTUALIZAÇÃO

Hardware
SO
Modificações
Hypervisor



03

OVERHEAD

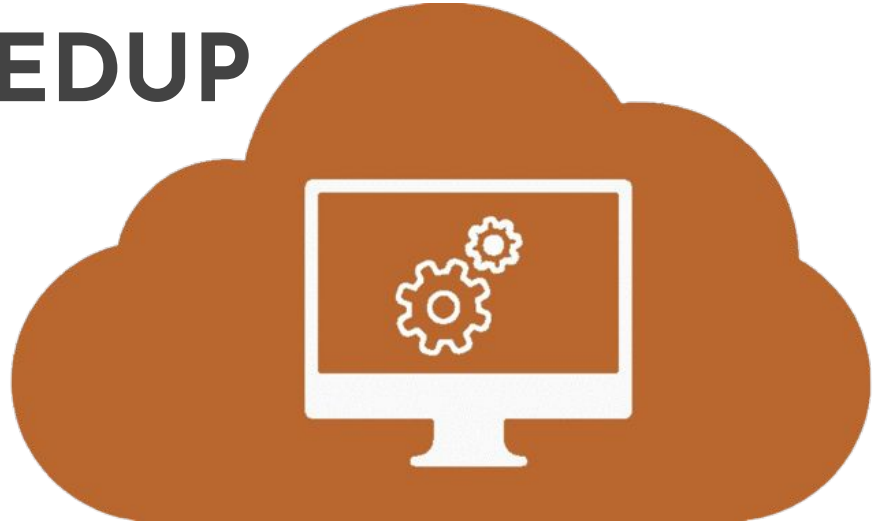
Processo
Armazenamento
Sobrecarga
Recurso



04

SLOWDOWN/SPEEDUP

Deliberar
Simular



05

GARGALOS

Desempenho
Processo
Componente



SUMÁRIO

01 INTRODUÇÃO

02 TRABALHOS RELACIONADOS

03 O AMBIENTE VIRTUAL XEN

04 ARCABOUÇO PARA PREVISÃO
DO DESEMPENHO

05 EXPERIMENTAÇÃO

06 ESTUDO DE CASO E
VALIDAÇÃO DO MODELO

07 CONCLUSÕES E
TRABALHOS FUTUROS

INTRODUÇÃO



VIRTUALIZAÇÃO

redução de
custos

1. Gerenciamento
2. Aquisição
3. Energia

redução de
complexidade

ambiente
flexível

TRABALHOS RELACIONADOS



Desafio anterior: aumentar nível de compartilhamento de mainframes

- Y. Bard. Performance Analysis of Virtual Memory Time-Sharing Systems. Proc. of IBM Systems Journal. 1975.
- Y. Bard. An analytic Model of the VM/370 System. Proc. of IBM Journal of Research and Development. 1978.
- Y. Bard. The VM/370 Performance Predictor. Proc. of ACM Computer Surveys. 1978.

Desafios atuais: segurança, consumo de energia, alto custo administrativo...

TRABALHOS RELACIONADOS

Alocação dinâmica: análise de aplicações já implantadas em ambientes virtuais

- P. Garbacki and V. Naik. Efficient Resource Virtualization and Sharing Strategies for Heterogeneous Grid Environments. 2007.
- G. Rodosek, M. Gohner, M. Golling, and M. Kretzschmar. Towards an Accounting System for Multi-Provider Grid Environments. 2007.

Artigo atual: análise de aplicações implantadas em ambiente real

TRABALHOS RELACIONADOS



- D. Menascé. Virtualization: Concepts, Applications, and Performance. 2006.
- D. A. Menascé, L. W. Dowdy, and V. A. F. Almeida. Performance by Design: Computer Capacity Planning By Example. 2004.
- F. Benevenuto, C. Teixeira, M. Caldas, V. Almeida, J. Almeida, J. R. Santos, and G. Janakiraman. Performance Models for Applications on Xen. 2006.

Artigo atual: tem equações detalhadas, parâmetros importantes da configuração e captura características reais das camadas de virtualização

O AMBIENTE VIRTUAL XEN

- Monitor de máquinas virtuais (VMM)
- Permite que múltiplas instâncias operacionais executem concorrentemente em uma única máquina física

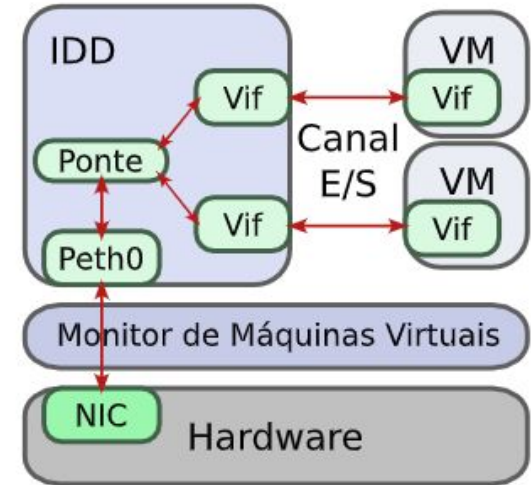
Obs: concorrência é dado ao sistema operacional que é capaz de usar o processador para executar tarefas ao mesmo tempo que outras operações.

- Foi um dos primeiros a utilizar o conceito de paravirtualização



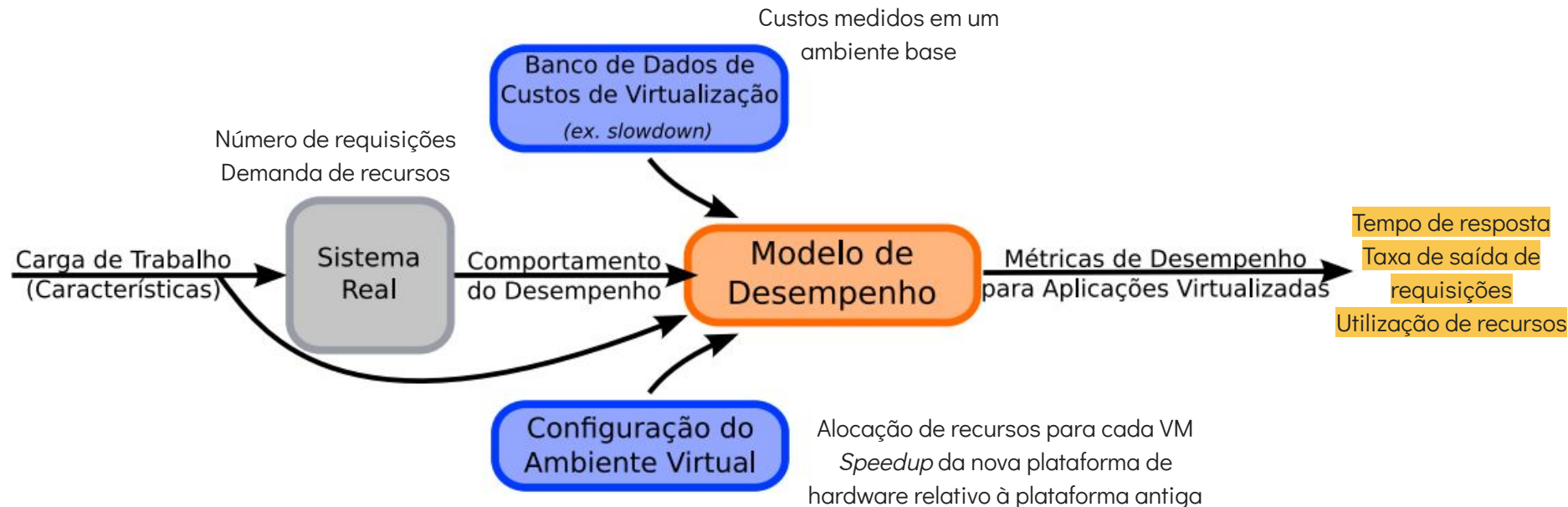
ARQUITETURA DO VMM

- Cada aplicação executando em um S.O acessa dispositivos de hardware através do IDD
- As outras VMs executam dispositivos que se comunicam com o IDD para acessar os verdadeiros dispositivos de hardware
- Uma VM acessa o hardware indiretamente através de um dispositivo virtual no domínio vif conectado ao IDD
- Para evitar cópia de dados, referências às páginas são transferidas através do IDD ao invés dos verdadeiros dados de E/S



- IDD: Vm especial com acessos ao hardware
- NIC: Interface de Rede
- Vif: Dispositivo virtual

ARCABOUÇO PARA PREVISÃO DE DESEMPENHO



MODELOS DE DESEMPENHO

ESTRATÉGIA: Consiste em definir um fator para representar o *overhead* introduzido pela camada de virtualização na execução de uma aplicação em uma VM

PREMISSA: Cada VM executa uma aplicação de classe diferente, onde uma classe de aplicações representa um tipo de aplicação para o qual o *overhead* de virtualização foi calculado

calculado em uma plataforma de hardware base,
para o qual se sabe o *speedup* em relação à outras máquinas

MODELOS DE DESEMPENHO

**MODELANDO O
DESEMPENHO EM
VMs**

**MODELANDO O
DESEMPENHO DE
OPERAÇÕES DE E/S**

**LIMITES
ASSINTÓTICOS**

MODELANDO O DESEMPENHO EM VMs

vm_i , sendo $i = \{0...n\}$

- $i = 0$ é o IDD
- **Todas** as VMs compartilham a mesma plataforma de hardware
- **Cada** VM executa uma aplicação de classe diferente (i)

recurso k , sendo $k = \{1...m\}$

CÁLCULO DO OVERHEAD DE VM_i

SLOWDOWN:

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

DEMANDA DE SERVIÇO:

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

S_k^i	Slowdown da classe i no recurso k
$B_k^{vm_i}$	Tempo ocupado do recurso k para executar vm_i
B_k^i	Tempo ocupado do recurso k para a classe i no ambiente real
$D_k^{vm_i}$	Demanda do recurso k pela vm_i
D_k^i	Demanda do recurso k para a classe i no ambiente real
P_k^i	Speedup da classe i no recurso k

CÁLCULO DO OVERHEAD DE VM_i

UTILIZAÇÃO:

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

TEMPO DE
RESIDÊNCIA
MÉDIO:

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

$U_k^{vm_i}$	Utilização do recurso k pela vm_i do ponto de vista da vm_i
λ^i	Taxa de chegada de requisições na vm_i
$D_k^{vm_i}$	Demanda do recurso k pela vm_i
cap_i	Porção do total de recursos que vm_i pode utilizar
$R_k^{vm_i}$	Tempo de residência médio do recurso k pela vm_i

CÁLCULO DO OVERHEAD DE VM_i

TEMPO DE RESPOSTA EM VM_i:

$$\sum_{k=1}^m R_k^{vm_i} = \frac{D_k^{vm_i}}{1 - U_k^{vm_i}} \quad + \quad R_{CPU,i}^{vm_0} = \frac{D_{CPU,i}^{vm_0}}{1 - U_{CPU}^{vm_0}}$$

MODELANDO O DESEMPENHO DE OPERAÇÕES DE E/S

- O IDD (vm_o) é o componente do Xen responsável por executar operações de entrada e saída (E/S) para outras VMs
- Abordagem para prever utilização de CPU do IDD
 1. Determinar o custo de processar um pacote (cp^i) no ambiente virtual para a classe i de aplicações
 2. Analisar a carga gerada pela aplicação alvo no ambiente real medindo o número médio de pacotes por requisição (pr^i) para a classe i

PREVISÃO DA UTILIZAÇÃO DE CPU DO IDD

DEMANDA
DE CPU:

$$D_{CPU,i}^{vm_0} = \frac{cp^i \cdot pr^i}{P_{CPU}^i}$$

UTILIZAÇÃO
DE CPU:

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

TEMPO DE
RESIDÊNCIA
MÉDIO:

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

$D_{CPU,i}^{vm_0}$	Demanda de CPU no IDD devido a atividade de E/S na vm_i
cp^i	Tempo de CPU para o IDD processar um pacote da classe i
pr^i	Número de pacotes por requisição da classe i
P_{CPU}^i	<i>Speedup</i> da classe i na CPU
$U_{CPU}^{vm_0}$	Utilização de CPU no IDD
λ^i	Taxa de chegada de requisições na vm_i
cap_0	Porção do total de recursos que o IDD pode utilizar
$R_{CPU,i}^{vm_0}$	Tempo de Residência no IDD para a classe de aplicações i

LIMITES ASSINTÓTICOS

Qual o valor máximo de taxa de chegada de requisições que um determinado serviço consegue suportar ao ser migrado para um ambiente virtual?

Depende:

- Das demandas de todos os recursos
- Do parâmetro do escalonador cap_i
- Da taxa de chegada de requisições em cada VM

LIMITES ASSINTÓTICOS

VM É GARGALO: $\lambda^i = \frac{U_k^{vm_i} \cdot cap_i}{D_k^{vm_i}} \quad \lambda^i \leq cap_i / D_k^{vm_i}$

IDD É GARGALO:
$$U_{CPU,i}^* = \sum_{j=1}^N (D_{CPU,j}^{vm_0} \cdot \lambda^j) \quad \forall j \neq i$$

soma das utilizações de CPU no IDD devido a todas as VMs exceto a vmi.

POR FIM:
$$\lambda_{max}^i \leq \min\left(\frac{cap_i}{\max_{k=1}^K D_k^{vm_i}}, \frac{cap_0 - U_{CPU,i}^*}{D_{CPU,i}^{vm_0}}\right)$$

EXPERIMENTAÇÃO

Para todos os experimentos, foram utilizados

Servidor Xen

- Intel Xen 64-bit versão 3.0.4
- Duas CPUs 3.2 GHz com 2GB de RAM
- Disco com 7200 RPM e 8MB de cache L2
- Duas placas de rede Ethernet Gigabit Broadcom Realtek

VMs

- VMs com XenLinux derivado de uma distro Debian, kernel 2.6.16
- Máquina do cliente com a mesma distribuição Linux

Servidor Linux

- Configurado com 1024 MB de RAM
- Cada VM é configurada com 512 MB de RAM
- IDD tal qual a VM

AVALIAÇÃO DE DESEMPENHO

- Para avaliar o desempenho de aplicações em ambientes virtuais, é necessário coletar medidas de desempenho no ambiente virtual.
- Como parte do arcabouço de monitoramento do trabalho, os autores desenvolveram uma aplicação chamada XenCPU para medir o tempo de CPU ocupado no Xen.

AVALIAÇÃO DE DESEMPENHO

Métricas que foram avaliadas:

número de instruções executadas e misses (falhas) nas caches do processador e na TLB1

XEN

Utilização do Xenoprof

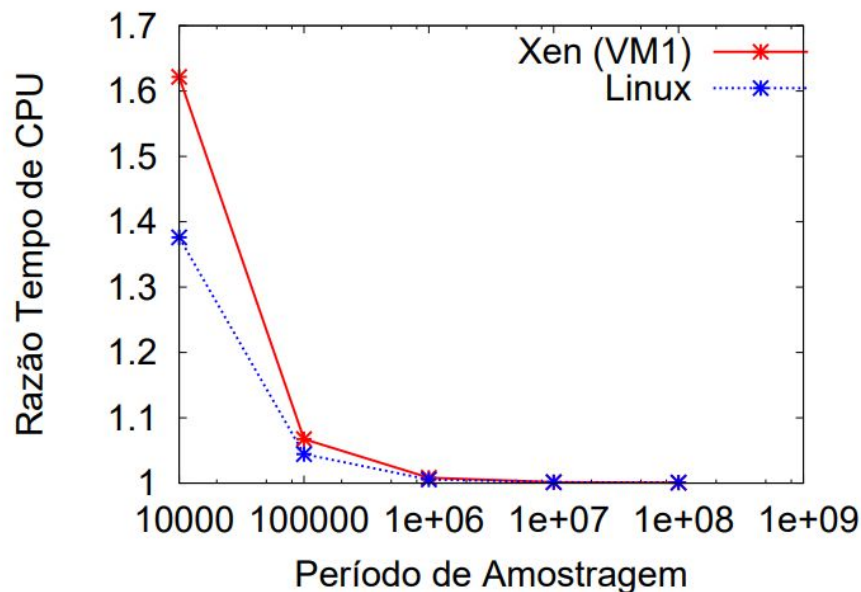
- ➔ ferramenta que, periodicamente, coleta eventos de hardware do sistema

LINUX

Utilização do Oprofile

- ➔ possui a mesma funcionalidade do Xenoprof para máquinas executando Linux.

OVERHEAD DO XENOPROF E OPROFILE



- Foi definido um período de amostragem no qual o Xenoprof e o Oprofile não causasse um grande overhead ao sistema, para realizar uma comparação justa.
- Para o Linux, plotaram valores obtidos executando o Oprofile em relação ao sistema sem executar o Oprofile.
- Para a VM, plotaram valores executando o Xenoprof em relação à VM executando o mesmo benchmark sem executar o Xenoprof.

Imagem: compara o tempo de CPU para executar uma compilação de kernel em uma máquina com Linux e o tempo de CPU na mesma máquina em uma VM do Xen

SLOWDOWN PARA CLASSES DE APLICAÇÕES

Para prover suporte experimental à abordagem de criar um slowdown para representar o custo da virtualização para diferentes tipos de aplicações, eles avaliaram o slowdown de um servidor Web que utiliza apenas conteúdo estático.

SLOWDOWN PARA CLASSES DE APLICAÇÕES

CLIENTE

Httpperf

- Permite várias requisições http

SERVIDOR WEB

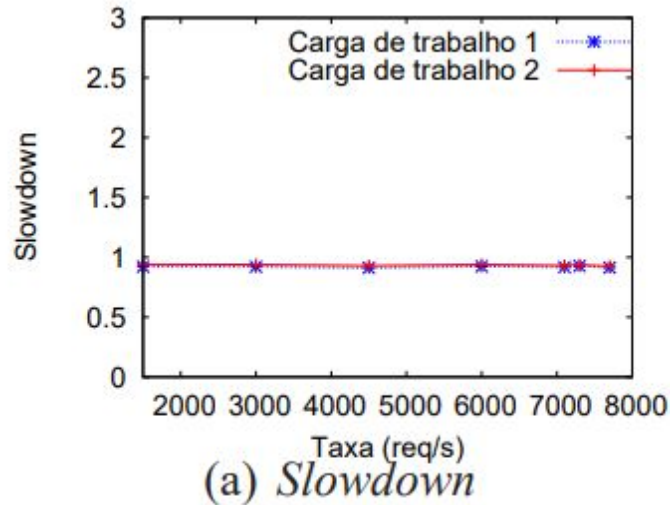
Apache versão 2.0.55

- Software de código aberto e distribuição gratuita

A VM hospeda o servidor apache em uma única CPU e o IDD executa em outra CPU separadamente. As duas cargas de trabalho utilizadas pelos clientes e o conteúdo do servidor foram geradas pelo SPECWeb99.

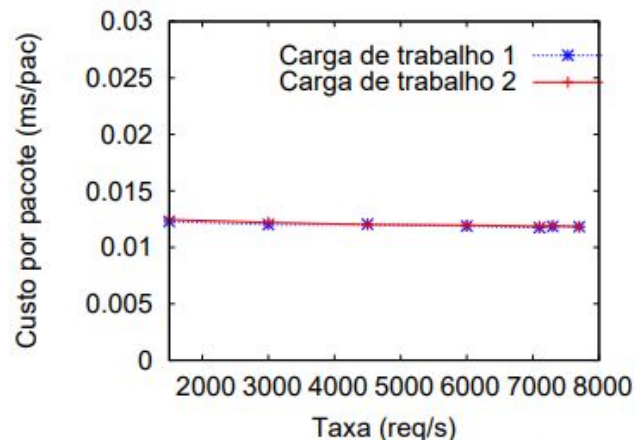
ANÁLISE DE SLOWDOWN E CUSTO POR PACOTE

resultados experimentais que guiaram a elaboração dos modelos



- Slowdown da CPU da VM para as duas cargas de trabalho como uma função da taxa de requisições
- Os valores para as duas cargas de trabalho são bem próximos
- Não dependem da taxa de chegada de requisições
- O slowdown médio para ambas as cargas é aproximadamente 0.92

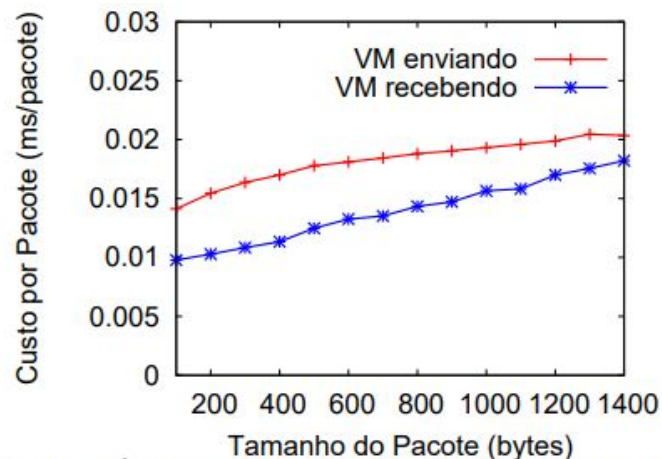
OVERHEAD NO IDD



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

- Os autores proveram uma avaliação experimental para entenderem os principais fatores que envolvem a modelagem do overhead de E/S no Xen.
- A configuração utilizada neste experimento é a mesma utilizada na análise do slowdown
- O gráfico mostra o custo de CPU no IDD por pacote da vm_i (cp^i) como uma função da taxa de requisições para as cargas utilizadas.
- O cpi é constante, mesmo para grandes taxas de chegadas de requisições.

OVERHEAD NO IDD



(c) cp^i para diferentes tamanhos de pacotes

- Esse gráfico mostra o cpi como uma função do tamanho do pacote.
- Podemos ver que o cpi para pacotes de chegada e de saída aumenta com o tamanho do pacote.
- Analisando o número de interrupções por pacote, entenderam que existe menos de uma interrupção por pacote quando pacotes são pequenos, diminuindo o cpi
- O componente principal que varia o número de interrupções por pacote é a taxa entre chegadas de pacotes.

PREMISSA

Sendo assim, uma premissa de modelos é que a carga de trabalho utilizada para medir o número de pacotes por requisição da classe i (pr^i) e calcular $D_{CPU,i}^{vm_0}$ utilizando a equação 5, tenha as mesmas características da carga de trabalho da aplicação alvo.

Ex: distribuição do tamanho, tempo entre chegadas, etc

Equação 5:
$$D_{CPU,i}^{vm_0} = \frac{cp^i \cdot pr^i}{P_{CPU}^i}$$
 } demanda de cpu do idd

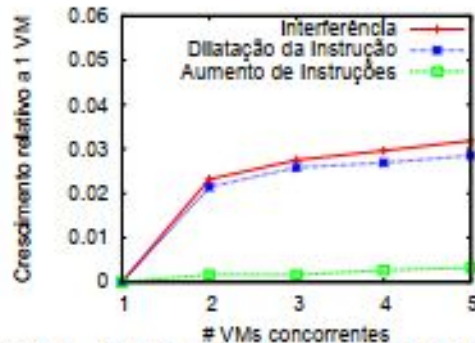
INTERFERÊNCIA ENTRE VMs

- Motivo
- Situações
- Overhead
- Capturar

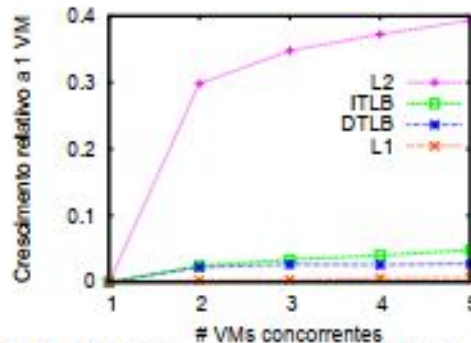
Aumento do Tempo de CPU

$$CPU_Time = Inst_Count * CPI * Clock_Cycle_Time$$

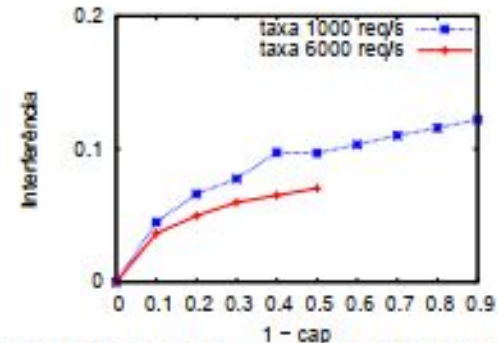
- AI Aumento de Instruções
- DI Dilatação de Instrução
- I Interferência



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



(b) Causas da DI para a compilação do kernel



(c) Impacto do cap na Interferência

ESTUDO DE CASO E VALIDAÇÃO DO MODELO

PASSO 1

Criação da base de dados

PASSO 2

Medição das aplicações alvo no ambiente virtual

PASSO 3

Escolha da plataforma de hardware

PASSO 4

Escolha da classe de aplicações que melhor representa a aplicação alvo

PASSO 5

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

PASSO 6

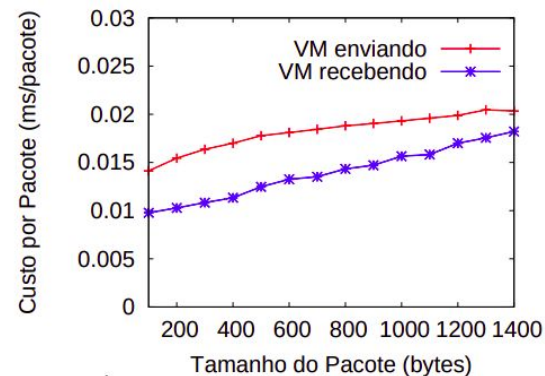
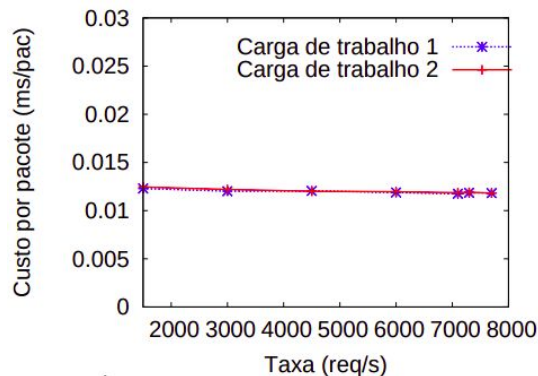
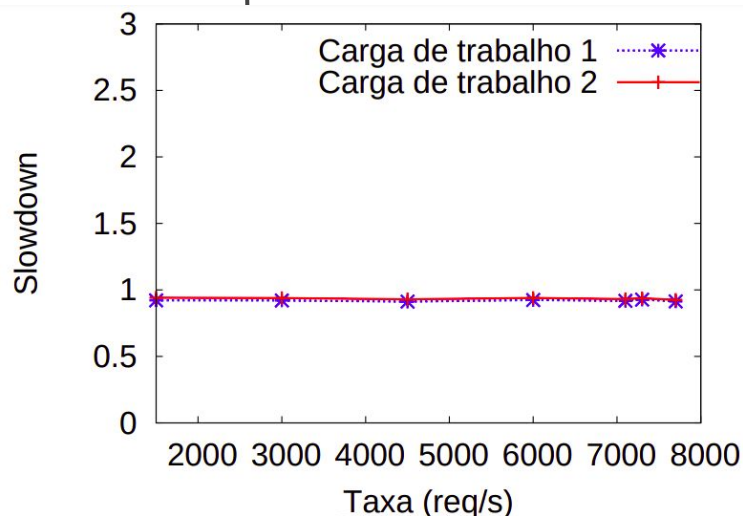
Cálculo de $D_k^{vm_i}$, $U_k^{vm_i}$ e $R_k^{vm_i}$ para cada i e k

ESTUDO DE CASO E VALIDAÇÃO DO MODELO

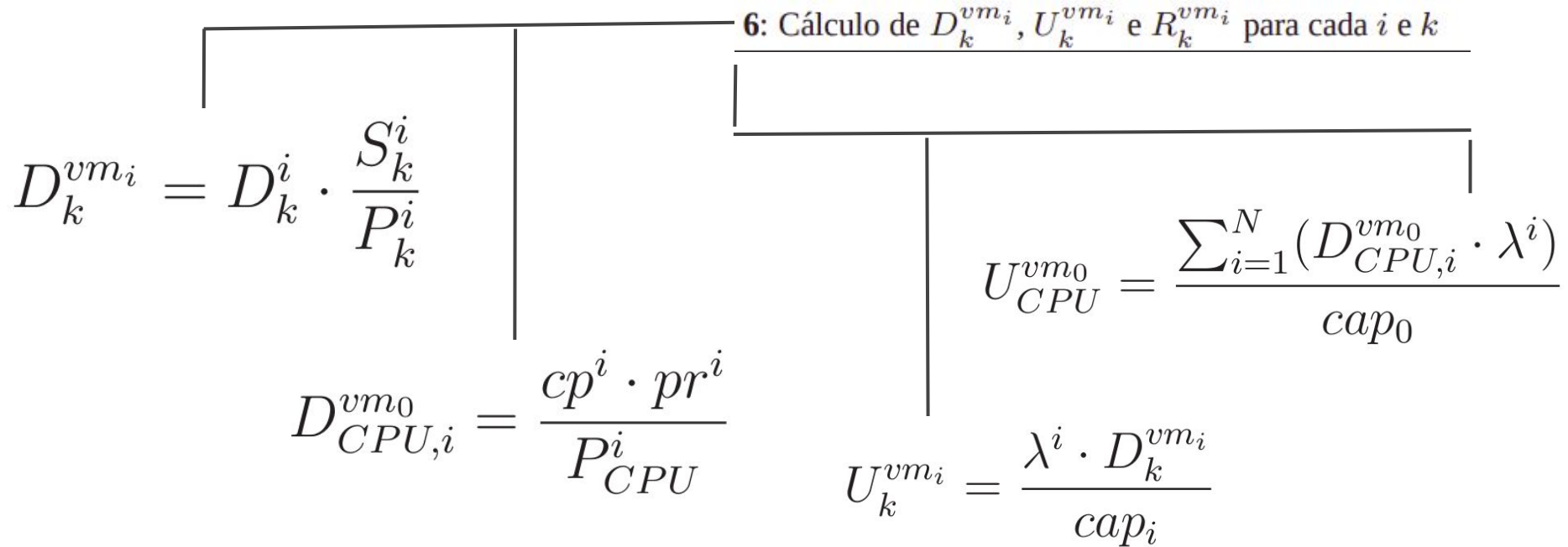
1: Criação da base de dados com:

- S_k^i para $i, 1..N$ e $k, 1..K$

- cp^i e desempenho relativo de hardware para máquinas diferentes



ESTUDO DE CASO E VALIDAÇÃO DO MODELO



CONCLUSÕES E TRABALHOS FUTUROS

Conclusões

- O overhead de virtualização aumenta quando VMs compartilham a mesma CPU

Trabalhos Futuros

- Validar os modelos sugeridos
- Desenvolver modelos para aplicações de classe fechada
- Avaliar o arcabouço para diferentes ambientes virtuais
- Desenvolver uma ferramenta que utilize nosso modelo

REFERÊNCIA

BENEVENUTO, F. et al. Medindo e Modelando o Desempenho de Aplicações em um Ambiente Virtual. [s.l: s.n.]. Disponível em:
<<https://homepages.dcc.ufmg.br/~fabricio/download/sbrc08.pdf>>.