

# Análise de resposta em frequência para modelagem e geração de carga de trabalho em aplicações de *e-commerce*

Lourenço Alves Pereira Júnior<sup>1,2</sup>, Edwin Luis Choquehuanca Mamani<sup>1</sup>,  
Regina H. Carlucci Sanatana<sup>1</sup>, Marcos José Santana<sup>1</sup>, Francisco José Monaco<sup>1</sup>

<sup>1</sup>Instituto Federal de São Paulo — IFSP  
ljr@ifsp.edu.br

<sup>2</sup>Universidade de São Paulo — USP  
{ljr,edwin,rcs,mjs,monaco}@icmc.usp.br

**Abstract.** *This paper aims to evaluate the performance of an e-commerce system under a time-varying workload. The approach is based on the Transfer Function Model which enables analysis in the frequency-domain allowing new insights towards workload generation. The main contributions include the novel approach for performance modeling of computing systems and the system's model as parameter for the workload generation.*

**Resumo.** *Este artigo apresenta a avaliação de desempenho de um sistema de e-commerce operando em condições de carga de trabalho variantes no tempo. A abordagem é baseada em Função de Transferência, permitindo a análise no domínio da frequência. As principais contribuições incluem a abordagem inovadora para modelagem de desempenho e a geração de carga que considera o desempenho como parâmetro.*

## 1. Introdução

A computação em nuvem é uma realidade no ciclo de vida de desenvolvimento de sistemas computacionais atuais. Dadas suas características essenciais [Mell and Grance 2011], a potência computacional disponível para uma aplicação (capacidade) é basicamente variável (elasticidade), e a demanda tende a ser flutuante, sendo influenciada por sazonalidades e padrões de acessos, promoções eventuais (*e.g.*: *black friday*) [Yuan et al. 2013, Santos et al. 2016, Silva et al. 2013]. Nesse sentido, a modelagem do desempenho *dinâmico* é importante pois permite que a capacidade computacional seja alterada em tempo de execução em função da carga de trabalho [Mamani et al. 2015].

No contexto de aplicações Web, observa-se a adoção de sistemas de computação em nuvem como plataforma execução [Qu et al. 2016]. A carga de trabalho variante no tempo é um fator importante para sistemas de *e-commerce*, pois percas de desempenho diminuem a taxa de conversão dos clientes, o que ocasiona diminuição de vendas. A variação da carga traduz-se como um elemento que perturba o regime estacionário de um sistema, gerando um comportamento transiente antes do eventual novo regime estacionário. É durante o transiente que sistemas experimentam condições extremas de sua utilização e gargalos são revelados [Veeraraghavan et al. 2016].

Quando relaciona-se a demanda e a capacidade de um serviço em nuvem para modelagem do desempenho, técnicas tradicionais (como na modelagem baseada em Teoria de Filas) apresentam-se desafiadoras dada sua natureza estacionária [Yang and Liu 2012].

Nesse cenário, um requisito importante para o planejamento de capacidade *dinâmica* é entender como o sistema reage a diferentes perturbações na carga de trabalho. Este artigo (1) apresenta uma abordagem de modelagem de desempenho não estacionária baseada em Função de Transferência e (2) analisa o comportamento de um sistema sujeito a diferentes variações na carga de trabalho. As contribuições deste trabalho são a análise de resposta em frequência do desempenho de um sistema de *e-commerce* real e a aplicação do método para a geração de carga de trabalho (com rajadas e baseada em traços). O diferencial de nossa proposta é considerar o modelo do sistema como parâmetro no processo de geração.

A seguir, na Seção 2 são discutidos trabalhos relacionados. A Seção 3 apresenta o sistema em estudo e sua modelagem. A Seção 4 apresenta a análise de resposta em frequência. Nas Seções 5 e 6 aplicam-se os conceitos abordados para a especificação de dois métodos para geração de carga de trabalho. Por fim, na Seção 7 conclui-se este estudo.

## 2. Trabalhos Relacionados

A avaliação de desempenho tradicional utiliza de modelos capazes de prever o desempenho de sistemas em regime estacionário [Jain 1990]. No entanto, conforme apresentado por [Yang and Liu 2012, Pereira 2016], há um esforço para representação analítica do comportamento transiente. Um exemplo da importância da modelagem do transiente pode ser observado em ambiente de computação em nuvem em que flutuações na demanda podem fazer com que Máquinas Virtuais sejam ligadas desnecessariamente [da Luz et al. 2016]. [Hellerstein et al. 2004] apresentam métodos e técnicas para esse fim por meio da abordagem de Função de Transferência (FT).

Com base nesses estudos e na aplicação desses modelos para a avaliação de desempenho transiente, [Pereira et al. 2015b] propuseram um modelo conceitual de requisitos denominado *Monitor, Effector, Demand e Capacity* (MEDC). Sua finalidade é produzir um arcabouço para instrumentação e/ou adaptação de sistemas computacionais para a execução adequada de experimentos para esse tipo de avaliação de desempenho. O modelo conceitual possibilita a modulação da demanda e da capacidade, baseada no monitoramento periódico das métricas de desempenho de interesse e atuações específicas para o sistema em avaliação [Pereira et al. 2015a].

Neste artigo, essa abordagem para avaliação de desempenho é aplicada a um sistema de *e-commerce* baseado no *benchmark* Bench4Q [Zhang et al. 2011]. Seu projeto contempla a análise e produção de resultados estacionários como uma ferramenta orientada a qualidade de serviço (QoS). Trata-se da extensão do TPC-W (*Transaction Processing Performance Council — transactional web benchmark*) [Menasce 2002]. As principais melhorias do Bench4Q incluem o suporte a métricas baseadas em sessões (não apenas em requisições) e a geração de carga de trabalho sensível a QoS para fins de planejamento de capacidade [Zhang et al. 2011]. [Mamani 2016] estendeu o Bench4Q, transformando-o em uma ferramenta de *benchmarks* para avaliação de desempenho não estacionária e [Souza 2016] implementou o módulo gerador de carga de trabalho (*Demand* do modelo MEDC) para que fosse possível a geração de cargas variantes no tempo.

Como um dos desafios apresentados por [Huang et al. 2014] está a adequação de modelos dinâmicos aplicados à implementação de Teoria de Controle com aplicação em gerência de recursos. Isso é importante, pois a avaliação é feita de modo

*offline* com uma carga capaz de expor o comportamento dinâmico do sistema. Avaliações do tipo *what-if* são importantes [Jiang et al. 2016], mas tem-se o ônus de várias execuções e configurações para obter resultados úteis. Modelos autor-regressivos possuem características recursivas e requerem baixo custo computacional [Yang and Liu 2012]. Portanto, há uma tendência de aprimoramento na utilização da Teoria de Controle para o gerenciamento de recursos em ambientes de computação em nuvem [Papadopoulos et al. 2015]. O que motiva a necessidade de diferenciação de serviço em aplicações de Internet multi-camadas com desempenho controlado [Diao et al. 2006].

Esses trabalhos têm em comum a abordagem: (1) baseada em FT para lidar com a carga de trabalho variante no tempo, algo comum no domínio de computação em nuvem e (2) *ad hoc* para resolução de problemas. Percebe-se a falta de iniciativas para especificação de um método genérico para modelagem de desempenho de aplicações de três camadas em ambiente de nuvem, bem como sua análise de resposta em frequência e geração de carga de trabalho que levem em conta seu comportamento dinâmico.

### 3. Sistema em Estudo

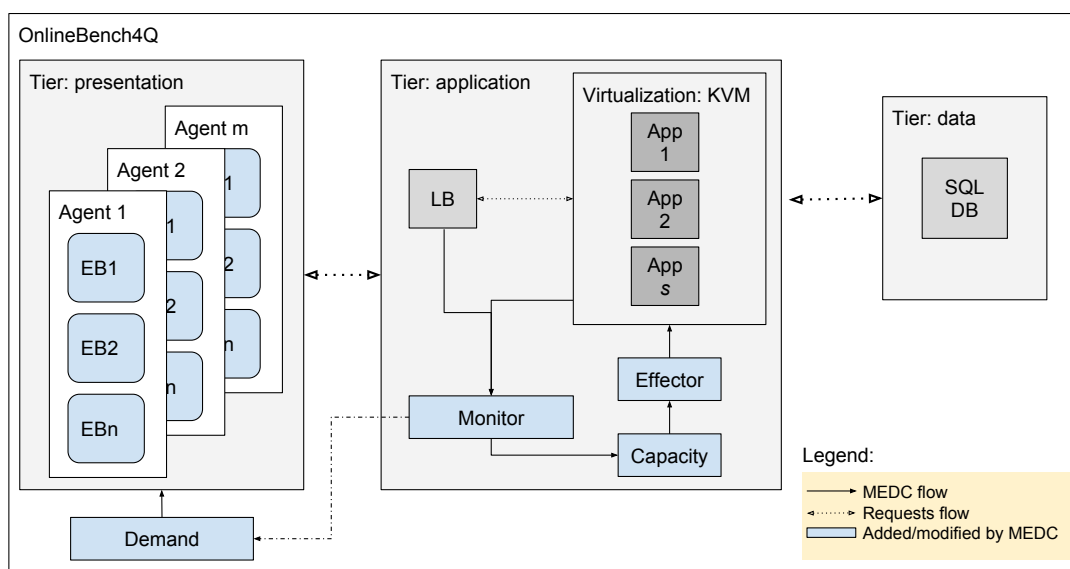
O sistema em estudo é uma aplicação web de três camadas, uma loja virtual (*e-commerce*) parte do Bench4Q [Zhang et al. 2011]. Foram feitas as adaptações necessárias para que esse *benchmark* se adequasse ao modelo conceitual MEDC [Pereira et al. 2015a].

A Figura 1 ilustra a implementação, as interações entre os componentes e a forma de implantação (*deploy*). O nome dado a esta nova versão é OnlineBench4Q, pois é possível alterar as características da carga de trabalho em tempo de execução (e.g: requisições com médias variantes no tempo) e capacidade adaptativa (e.g: *auto scaling*). Na *camada de apresentação*, as requisições são feitas pelos *emulated browsers* (EBs). Antes da execução um arquivo de configuração é passado para o módulo *Demand* para especificar como a carga de trabalho será gerada durante o experimento. Existem três formas básicas de especificação: (1) **carga sintética**, realizada através de uma distribuição de probabilidade (exponencial, normal etc.) e (2) **traço**, um arquivo com a lista de requisições a serem executadas. A *camada de aplicação* é composta por um balanceador de carga implementado pelo HAProxy<sup>1</sup> — que fornece a métrica de quantas requisições por segundo chegam à aplicação — que recebe as requisições e as repassa para os servidores virtualizados (round-robin). Caso a taxa de utilização seja alta/baixa, adiciona-se/remove-se novas VMs. O mecanismo de gerenciamento de recursos é conforme especificado em [Mamani et al. 2015, Pereira et al. 2015a]. O *Monitor* afere a taxa de utilização a cada segundo através do `proc` do Linux. O *Capacity* é implementado por um controlador Proporcional-Integral. O *Effector* executa alterações nas máquinas virtuais (VMs) dessa camada. A *camada de dados* foi implementada pelo PostgreSQL. A Tabela 1 detalha o sistema físico para execução dos experimentos.

O modelo de desempenho adotado relaciona a quantidade de requisições por segundo (req/s) e a média da taxa de utilização dos servidores de aplicação. A carga de trabalho é gerada por *browsers* emulados (*emulated browsers* — EBs) em modo aberto, com sessões configuradas para manter uma média de 47 req/s por EB, e o grafo CBMG arranjado para acesso prioritário a páginas que demandam operações de leitura na camada de dados. O fluxo de requisição de todos EBs é combinado no HAProxy. Foi possível

---

<sup>1</sup>Sítio do HAProxy: <http://www.haproxy.org/>



**Figura 1. Diagrama arquitetural de componentes do OnlineBench4Q, uma combinação de Bench4Q [Zhang et al. 2011] e MEDC [Pereira et al. 2015a].**

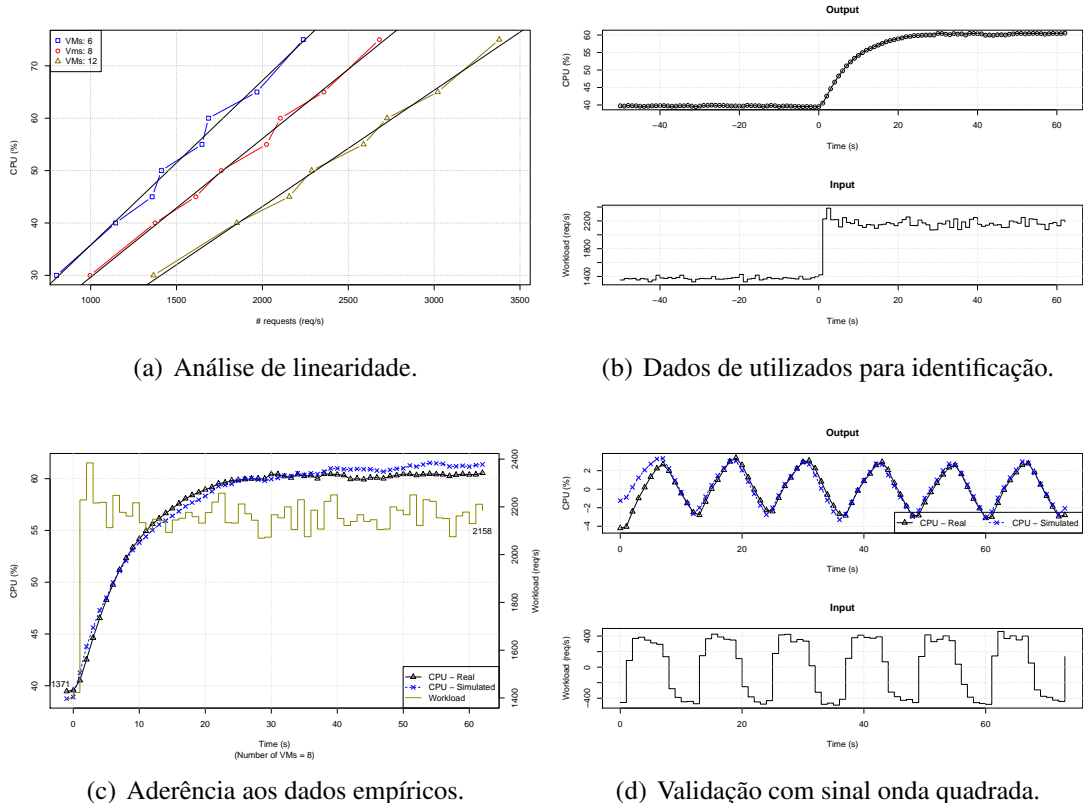
observar que, para esse cenário, quanto maior a quantidade de requisições, maior a taxa de utilização, caracterizando-se como uma relação linear entre essas duas variáveis — comprovada com os experimentos da Seção 3.1. Tal relação linear determina a região de operação desse sistema em conformidade com a carga avaliada. Um parâmetro para o sistema que altera esse comportamento é a quantidade de VMs. Mesmo que a taxa req/s seja constante, ao adicionar VMs, a taxa de utilização diminuirá. Convenciona-se como  $W$  a entrada para esse sistema (req/s),  $U$  para a saída (taxa de utilização média dos servidores de aplicação) e VMs a perturbação. A Seção 3.1 apresenta procedimentos com o propósito de quantificar o comportamento dinâmico entre as variáveis deste sistema.

### 3.1. Identificação do Sistema

A identificação de sistema é uma área de pesquisa que estipula métodos para execução de experimentos e análise de sistemas [Ljung 1999]. O processo de identificação adotado consiste em: (1) determinação do sistema: especificação de variáveis com relação de causa e efeito; (2) determinação do escopo: definição do tempo de amostragem e do gerador de carga de trabalho; (3) excitação do sistema: determinação de um sinal de entrada capaz de expor a dinâmica do sistema; (4) identificação do modelo: execução

**Tabela 1. Configuração do hardware utilizado nos experimentos.**

Nome	CPU (GHz)	Mem. (GB)	HD (Mb/s)	SO	#
<i>Load Balancer</i>	INTEL i5-3330-(3.0) x4	7,8	87,74	ClearOS 6.6	1
<i>DB server (pool)</i>	AMD Q6600-(2.4) x4	7,8	71,21	Ubuntu 14.04.2	8
<i>DB server (master)</i>	FX-8320-(3.5) x8	23	285,58	Ubuntu 14.04.3	1
Clientes	AMD Q6600-(2.4) x4	7,8	76,96	Ubuntu 14.04.2	9
Hipervisor	X5660-(2.8) x12	11	251,98	Ubuntu 15.04	1
Servidor de VMs	X5660-(2.8) x12	11	251,98	Ubuntu 15.04	4
VMs	QEMU Virtual x1	1	178,60	Ubuntu 12.04.5	-



**Figura 2. Resultados obtidos do processo de identificação do sistema.**

de experimentos para gerar dados e parametrização do modelo; (5) verificação: métricas estatísticas para avaliar o modelo; (6) validação: acurácia do modelo em relação outros sinais de entrada. Esses passos são detalhados em [Ljung 1999, Hellerstein et al. 2004, Pereira 2016]. A Figura 2 apresenta os resultados obtidos do processo de identificação.

Os passos 1 e 2 da identificação foram contemplados pela relação de linearidade entre as variáveis e é apresentada na Figura 2(a). Para a geração desses dados configurou-se experimentos da seguinte forma: variou-se a quantidade de VMs, valores  $\{6, 8, 12\}$ . Para cada quantidade de VMs encontrou-se uma taxa de req/s estacionária tal que impactasse na variável  $U$  nos patamares de  $\{30, 40, 50, 60\}$ . Dessa forma, para cada combinação de VMs e  $U$  aplicou-se a carga  $W$  (req/s) correspondente, aumentando  $U$  em 15% no meio do experimento. Mediu-se  $W$  e  $U$  periodicamente a uma taxa de amostragem de 1 s. Descartou-se os dados do período não estacionário. Em suma, conclui-se que o sistema é linear. Para este trabalho fixou-se a quantidade de VMs em 8 (oito) e a região de operação na faixa em que  $W$  está entre 1.371 e 2.158 req/s, fazendo com que  $U$  opere na faixa entre 40% e 60%, respectivamente. O ponto de operação é  $\bar{U} = 50\%$ .

O sinal de excitação adotado foi o de degrau (passo 3). Um sinal em degrau significa que o sistema deve estar em um ponto de operação com uma carga de trabalho estacionária e em determinado instante alterar abruptamente essa carga de modo que o sistema entre em outro patamar estacionário. Uma vantagem da utilização deste sinal é sua simplicidade de implementação e execução, sendo capaz de expor a dinâmica de um sistema computacional linear. A Figura 2(b) apresenta os resultados.  $W$  é mantida a uma taxa estacionária de 1.351 req/s e, no instante de tempo 0 (zero), houve uma mudança

abrupta para 2.158 req/s. Observa-se o comportamento dinâmico entre as duas variáveis, pois  $U$  muda de 40% para 60% não instantaneamente, mas sim gradativamente, semelhante a um modelo exponencial (i.e:  $U = g \cdot (1 - e^{-\lambda}) \cdot W$ ), em que  $g$  é o ganho e  $\lambda$  define a velocidade em que o novo patamar estacionário será atingido).

O modelo utilizado para a identificação foi o Autorregressivo com entradas exógenas (ARX). Características importantes desse modelo incluem: sua flexibilidade de parametrização, linearidade, discreto e capacidade de representação da estocasticidade. A Figura 2(c) apresenta uma simulação do modelo identificado em relação aos dados empíricos (passo 4). O sistema identificado é descrito como um modelo ARX de primeira ordem com os parâmetros estimados  $a_1 = -0.9037$  e  $b_1 = 0.00273$ , que pode ser representado em formato de Função de Transferência como

$$F(z) = \frac{b}{1 + az^{-1}} = \frac{0.00273}{1 - 0.9037z^{-1}}. \quad (1)$$

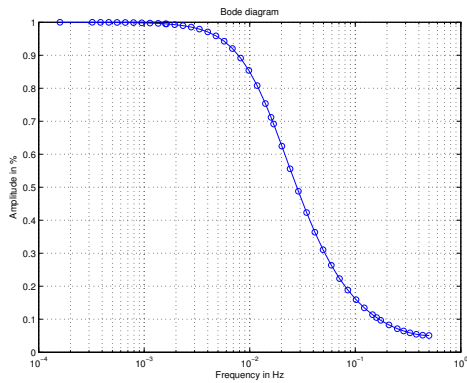
Os valores para RMSE e  $R^2$  foram altos e satisfatórios (passo 5), 0,7924 e 0,8661 respectivamente. A Figura 2(d) apresenta a validação de  $F(z)$  capaz de representar o comportamento do sistema na faixa de operação (passo 6).

#### 4. Análise de Resposta em Frequência

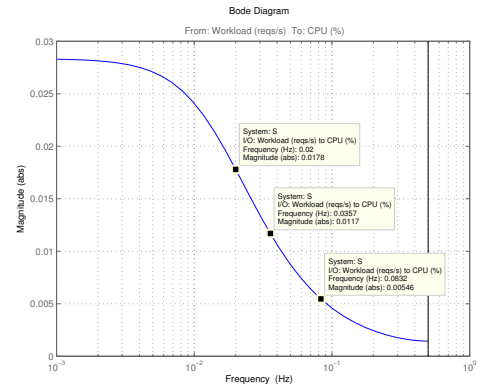
O desempenho do sistema foi modelado como um sistema dinâmico linear e invariante no tempo (*Linear Time-invariant* — LTI) e representado por uma Função de Transferência de primeira ordem (Eq. 1). Isso significa que, respeitada a região de operação, pode-se simular o comportamento dinâmico do desempenho de um sistema face a diferentes cargas de trabalhos. Isso é um ponto importante, pois permite que resultados sejam produzidos por simulação, em vez de experimentos que acarretam em custos. Dada a simplicidade e a característica recursiva de um modelo de FT, os resultados são praticamente instantâneos [Yang and Liu 2012]. Um método a ser explorado nesse tipo de modelagem de desempenho é a resposta em frequência, representado pelo diagrama de Bode.

O diagrama de Bode é um modelo não paramétrico em que utiliza-se a frequência para predição de desempenho (Figura 3). O eixo  $x$  representa a frequência com que eventos ocorrem, e no  $y$  o quanto do desempenho estacionário será atingido. O gráfico é produzido, teoricamente, ao excitar o sistema com uma onda senoidal de frequência  $x$ , obtendo a amplitude do sinal de saída  $y$ . A leitura do gráfico de resposta em frequência é feita ao interpretar o quanto a variável de saída é atenuada em relação a variações na variável de entrada. Por exemplo, seja uma onda quadrada (Figura 4(a)) de período  $P$  segundos e amplitude  $A$ . Se  $P$  tiver comprimento 0,5 s, a frequência será 2 Hz; se tiver 1 s, frequência de 1 Hz; 2 s, frequência de 0,5 Hz; e assim por diante. Dessa forma, define-se frequência como  $f = 1/P$ . Deve-se ressaltar que o período da onda quadrada possui dois patamares, cada um correspondendo a metade de  $P$ ,  $D = P/2$ . Na Figura 3(a), se  $P = 32$  s,  $D = 16$  s e  $f = 0,003$  Hz ( $3 \cdot 10^{-2}$ ), haverá uma atenuação de aproximadamente 50% do valor final atingido em estacionário. Portanto, para o sistema  $F(z)$ , Eq. 1, o impacto no desempenho do sistema é o inverso da frequência com que os eventos ocorrem.

O diagrama Bode serve como referência para quantificação da relação entre frequência e impacto no desempenho (Figura 3). Eventos de alta frequência para o sistema em estudo neste artigo caracterizam como alteração abruptas nos valores, porém com



(a) Diagrama de Bode de atenuação relativa.



(b) Diagrama de Bode.

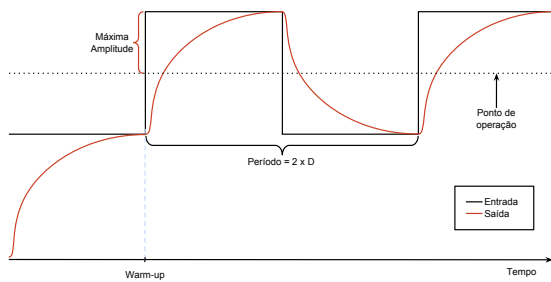
**Figura 3. Resposta em frequência do sistema  $F(z)$ , Eq. 1.**

duraco baixa incapaz de influenciar no desempenho. Seja  $W$  em regime estacionrio e dados coletados de execuo nica — inspecionar Figura 6(b). O sinal  ruidoso pois observa-se que h um patamar mdio de req/s e uma variao em torno desta mdia. A representao desse sinal no domnio da frequncia  composta por uma frequncia 0 Hz sobreposta de outras frequncias mais altas. A frequncia zero corresponde ao valor mdio de  $W$ . Uma concluso importante  que esse rudo influenciar pouco na varivel de sada do sistema identificado (Eq. 1). Os picos e vales causados pela alta frequncia se anulam, caso em que h uma varincia maior na entrada do que na sada.

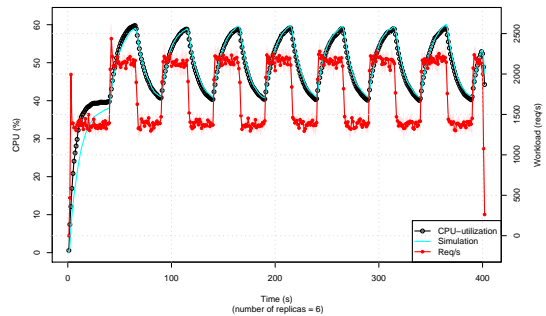
Como prova de conceito, foram escolhidas 3 (trs) frequncias de interesse a fim de analisar se os valores previstos pelo diagrama de Bode representam o desempenho do sistema. A Figura 3(b) apresenta as frequncias anotadas e os valores absolutos de atenuao. Os experimentos foram conduzidos aplicando-se  $W = 1.371$  por 40 s, correspondente ao tempo transiente inicial (*warm-up*), seguidos de uma excitao do tipo onda quadrada, com durao de  $\{6, 14, 25\}$  s. A durao,  $D_i$ , de uma frequncia,  $f_i$ , pode ser obtida pelo clculo  $D_i = 1/2f_i$ . A atenuao em  $U$   observada pela mxima amplitude realizada no experimento e dada por  $A = mag/g$ , em que  $mag$   magnitude, observada na Figura 3(b), e  $g$   o ganho do sistema, neste caso 0,0283. A terminologia adotada  representada na Figura 4(a). Cada experimento foi replicado 6 (seis) vezes e o intervalo de confiana de 95% foi calculado. A Figura 4 apresenta os resultados.

O mesmo sinal de entrada  $W$  produzido pelo sistema real foi passado para a FT, resultando em um sinal de sada simulado. Ou seja, o sinal de entrada foi passado tanto para o sistema real quanto para a FT. Observa-se que o modelo de FT para representao do desempenho  adequado. Cada experimento possui o valor mdio de  $\bar{U} = 50\%$ , com o sinal  $U$  excursionando nas amplitudes mnimas e mximas, patamares alto e baixo. A Tabela 2 apresenta a acurcia dos resultados obtidos.

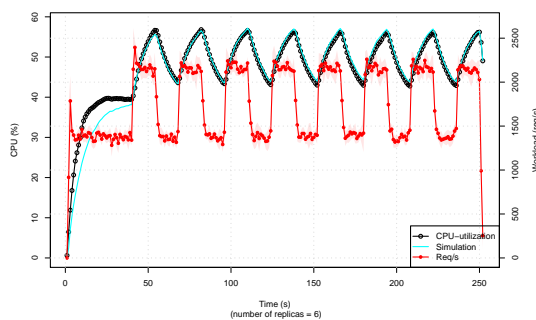
A atenuao de frequncias mais altas podem ser observadas na Figura 5. Neste caso obteve-se o valor de  $W$  estacionrio e com durao de 512 s a partir de um experimento no sistema fsico. A transformada de Fourier permite observar esse sinal no domnio da frequncia. Utilizou-se a transformada de Fourier rpida (FFT) para este fim. O resultado  o espectro do sinal, isto , o conjunto de frequncias que compem aquele sinal. Escolheu-se trs frequncias com a finalidade de observar como elas so atenua-



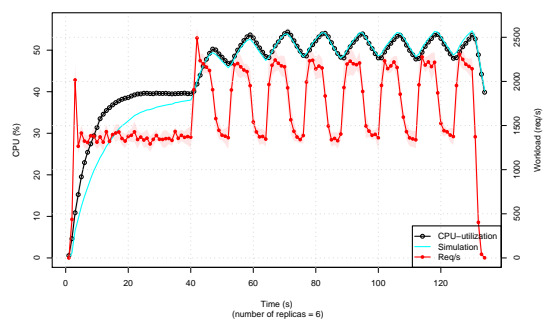
(a) Terminologia adotada para os experimentos.



(b) Resposta em frequência com duração de 25 s.



(c) Resposta em frequência com duração de 14 s.



(d) Resposta em frequência com duração de 6 s.

**Figura 4. Resposta em frequência. Ponto de operação a 50% de CPU. Denomina-se o sinal como onda quadrada ou trem de pulsos por seu formato característico.**

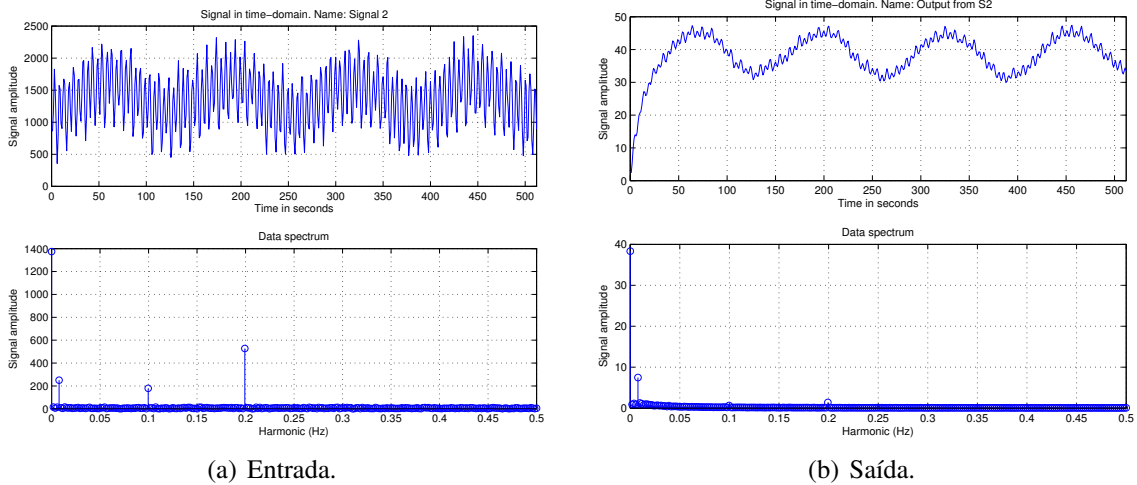
das após passar pelo sistema. As frequências escolhidas foram  $\{0.0075, 0.01, 0.02\}$  Hz. Assim, após a obtenção do espectro, multiplicou-se cada frequência por  $\{1.2, 1.4, 1.6\}$  respectivamente, amplificando-as. O sinal desejado foi computado a partir da transformada inversa de Fourier. Em suma, obteve-se o sinal no domínio do tempo; levou-o para o domínio da frequência; alterou-se algumas frequências; o trouxe para o domínio do tempo novamente. A Figura 5(a) apresenta o sinal após essa alteração (o sinal antes da alteração é apresentado na Figura 6(b)). A frequência alterada mais baixa, 0.0075, é responsável pela oscilação mais observável, e as outras duas frequências correspondem às oscilações de menor duração e com amplitudes contundentes.

Ao passar esse novo sinal de entrada,  $W'$ , pela FT, obtém-se  $U$  conforme observado na Figura 5(b). Ao inspecionar seu espectro, as frequências adicionadas mais

**Tabela 2. Acurácia dos valores baixo e alto preditos pelo modelo.**

Duração Degrau	Predito (Pr) (CPU %)	Empírico (Em) (CPU %)	Resíduo (Pr - Em)
25 alto	56,27	59,095	-2,825
25 baixo	43,73	40,305	3,425
14 alto	54,12	56,372	-2,252
14 baixo	45,88	43,099	2,781
6 alto	51,92	53,406	-1,486
6 baixo	48,08	47,960	0,12





**Figura 5. Comportamento de atenuação de frequências.**

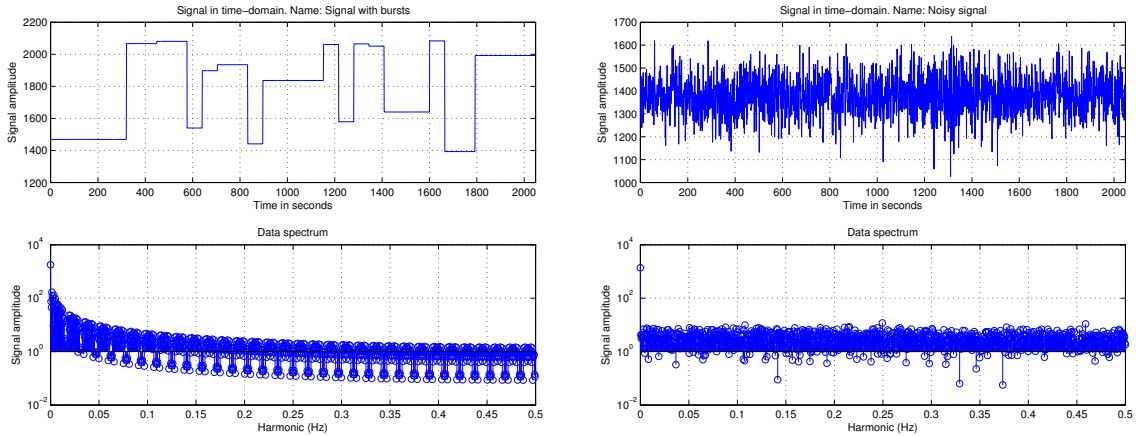
altas, 0.01 e 0.02, foram consideravelmente mais atenuadas se comparadas à frequência mais baixa, 0.0075. Mas todas as frequências mais altas (localizadas à direita) foram atenuadas também. As frequências mais baixas (à esquerda) foram menos atenuadas. A quantificação dessa atenuação é a mesma descrita pelo diagrama de Bode (Figura 3).

## 5. Geração de Rajadas

Uma aplicação que pode ser feita a partir da FT e da análise em frequência é a especificação de um método para a geração de carga de trabalho com rajadas. Uma característica importante dessa abordagem é considerar a carga de trabalho que relaciona o desempenho do sistema, um método diferente daqueles comumente adotados [Centurion et al. 2012]. Devido ao fato de ser um modelo gerado por meio de dados empíricos, pode-se parametrizar como a carga irá impactar no sistema de forma precisa. Seja a especificação de uma rajada tal que haja restrições nos períodos de flutuações no número de req/s, não tão curtas de modo que elas sejam absorvidas pela inércia do sistema, nem tão longas a ponto de se apresentarem como um degrau.

Como já foi ilustrado pela Figura 4(a), o trem de pulso possui dois parâmetros: a amplitude ( $A$ ) e a duração ( $D$ ). Para este estudo essas variáveis terão valores aleatórios dentro da faixa de operação do sistema. Um requisito para a geração da carga de trabalho é o modelo de desempenho do sistema, o qual, por meio da análise em frequência, quantifica a faixa de valores significantes. O tempo de assentamento (*settling time*) do sistema é 39 s (Eq. 1). Portanto,  $D$  deve possuir pelo menos tamanho 40 s de forma que o sistema atinja diferentes patamares de estacionariedade. Também é importante que a rajada ocorra dentro da região de operação ( $U \in [40\%, 60\%]$ ),  $A$  deve estar entre 1.371 e 2.158 req/s.

Um sinal do tipo *Pseudo-Random Binary Sequence* (PRBS) apresenta uma série que se assemelha com o trem de pulso já apresentado, porém com duração ( $D$ ) variável. O comprimento do sinal e a quantidade de mudanças de magnitude dependem do parâmetro  $n$ . No caso do PRBS esse valor corresponderá a  $l = 2^n - 1$  e  $m = 2^n/2$ , em que  $l$  é o comprimento do sinal,  $m$  é a quantidade de variações que ocorrerá no sinal e  $n$  a quantidade de *bits* utilizados para gerar a sequência. Isso significa que, caso  $n$  seja 4 (quatro), o sinal produzirá uma entrada sintética que corresponderá a uma excitação de



(a) Sinal pseudo-aleatório com amplitude variantes dentro da região de operação.

(b) Sinal ruidoso extraído do sistema, oriundo de execução única com carga de trabalho estacionária.

**Figura 6. Sinais para composição do sinal de rajada.**

duração 31 s e a quantidade de vezes em que o sinal alternará será 16.

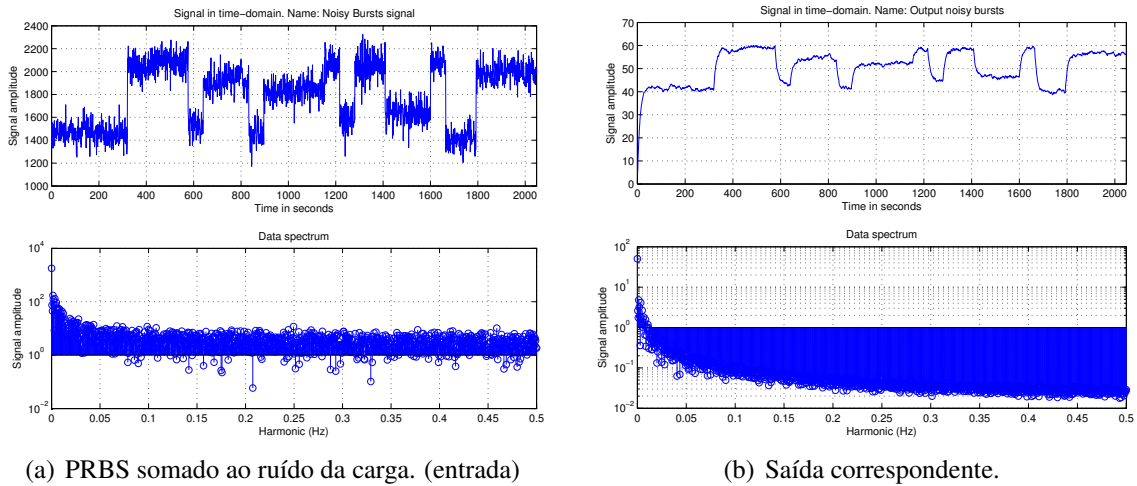
A amplitude pode ser alterada após a geração do sinal PRBS. Uma estratégia é gerar  $m$  valores dentro da faixa de operação e, para cada mudança presente no sinal, atribuir determinada amplitude. Com essa alteração o sinal resultante teria as características aleatórias de mudanças presentes no PRBS, porém não limitado aos valores extremos. A Figura 6(a) apresenta um sinal gerado dessa forma. A Figura 6(b) corresponde a uma execução de 2.048 s em regime estacionário, esse sinal é denominado como um sinal de entrada com ruído inerente da carga de trabalho.

Os espectros apresentados na Figura 6, obtidos pela FFT, podem ser somados. A Figura 7(a) corresponde ao sinal produzido. Ao passar esse sinal para a FT, o resultado produzido é apresentado na Figura 7(b). Ao inspecionar os espectros da Figura 7, percebe-se que parte considerável de frequências são removidas, comportamento típico de um filtro passa baixas. A rajada gerada conforme especificada teve seu objetivo atingido, provocando os patamares estacionários desejados. É possível mudar a duração das alterações e prever o desempenho do sistema por meio de simulação.

## 6. Carga de Trabalho Baseada em Traços

Como demonstrado nas seções anteriores, frequências do sinal de entrada são reduzidas após o processamento realizado pelo sistema. Portanto, ao filtrar o sinal de entrada,  $W_1$ , por um filtro com as mesmas propriedades dinâmicas do sistema, o resultado produzido,  $W_f$ , tende a ter um impacto semelhante ao de  $W_1$ . Isto é, para o sistema  $F(z)$ , as entradas  $W_1$  e  $W_f$  produzem o mesmo resultado. Pode-se derivar um filtro que mantenha a mesma magnitude entre os sinais de entrada e saída (o que corresponde a uma FT de ganho unitário) e remova as frequências conforme informações obtidas a partir do Diagrama de Bode da FT identificada. O equacionamento para esse filtro é  $S_{filtro}(z) = S(z)/dc$ , em que  $dc$  é o ganho da FT. Considerando a FT da Eq. 1 cujo ganho DC é 0,0283, tem-se

$$F_{filtro}(z) = \frac{F(z)}{0,0283} = \frac{0.00273}{0.02835 - 0.02562z^{-1}}. \quad (2)$$



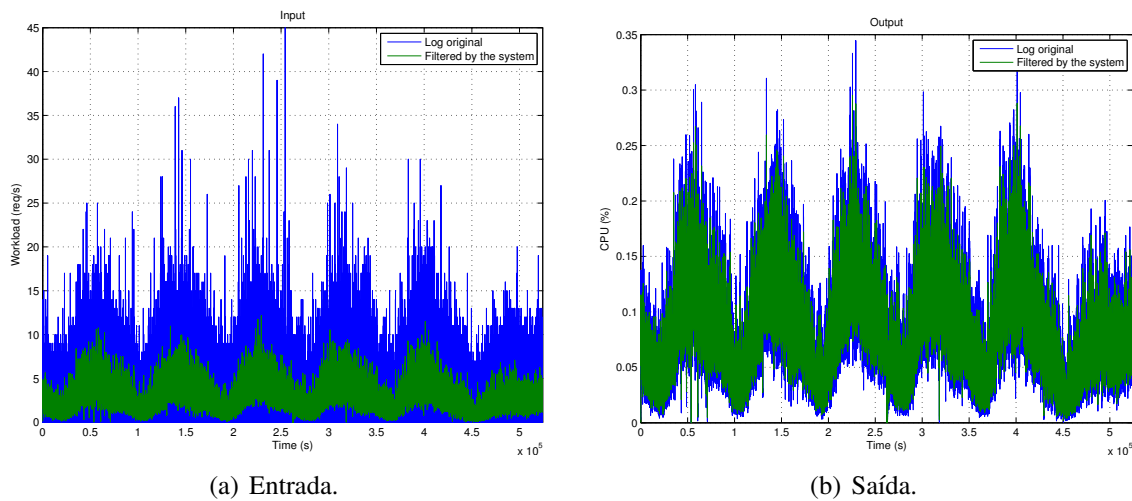
**Figura 7. Rajada produzida e seu impacto no sistema.**

Foram considerados 4 (quatro) traços de execução com a finalidade de verificar o impacto dessas cargas no sistema em teste, a saber: Clarknet, NASA, Copa '98 (obtidos de [LBL 2016] ) e Google [Reiss et al. 2011]. Os *datasets* preparados possuem informações de requisições a servidores Web (Clarknet, NASA e Copa '98) com uma taxa de amostragem de 1 s (1 Hz) e *jobs* para Google com precisão de milissegundos (1 MHz). Processou-se os arquivos de modo que cada requisição presente nos arquivos representasse uma requisição a ser transformada pela FT. No caso do Google, agrupou-se as requisições em períodos de 1 s. Dessa forma, os *datasets* serviram como sinal de entrada para duas FTs  $F(z)$ , Eq. 1, e  $F_{filtro}(z)$ , Eq. 2 — simuladas com o Matlab ©.

A Figura 8 apresenta os resultados obtidos do traço Clarknet. Ao observar os sinais de entrada (Figura 8(a)), eles são diferentes e apresentam correlação de 0,6478. Isso significa que parte considerável do sinal original foi removida, restando apenas as frequências mais baixas. O que se observa na saída (Figura 8(b)) é que há uma semelhança grande, correlação de 0,9509. Uma conclusão para este caso é que duas entradas distintas produzem a mesma resposta para o sistema em estudo. As diferenças nos sinais de entrada impactam de forma irrelevante no desempenho. No entanto, esse não é o caso da carga Copa '98, em que os sinais de entrada apresentam correlação de 0,9980 e saída de 0,9998 — o filtro removeu pouco das frequências existentes no sinal original.

A Tabela 3 apresenta os resultados obtidos de todos os traços estudados. Os traços Clarknet e NASA possuem frequências mais altas, que são absorvidas pela inércia do sistema. Já o traço da Copa '98 possui frequências baixas na alteração da taxa de req/s, isso faz com que os resultados com e sem filtro sejam equivalentes. Os traços da Google possuem altas frequências, mas também não linearidades, fazendo com que a correlação entre as entradas seja de aproximadamente 0,43, no entanto, as saídas com aproximadamente 0,82. Como o modelo é linear, as não linearidades existentes diminuem a correlação entre as saídas obtidas. Deve-se mencionar que essas afirmações são válidas para a FT em estudo (Eq. 1). Para sistemas com dinâmicas diferentes os resultados serão outros.

A carga de trabalho Clarknet tem pouco impacto no sistema, causando uma taxa de utilização de menos de 0,35%. Caso fosse necessário excitar o sistema com essa carga de trabalho, de forma tradicional, dificilmente obteríamos um resultado significativo. No



**Figura 8. Resultados obtidos a partir do traço Clarknet.**

entanto, como há a FT do sistema e sabe-se seu comportamento em resposta a frequência, é possível observar o espectro da carga em estudo e alterar a magnitude das frequências de modo que elas impactem no sistema conforme desejado.

## 7. Conclusões

Este artigo apresentou uma abordagem para a modelagem de desempenho baseada em Função de Transferência (FT), bem como a análise de resposta em frequência de um sistema de *e-commerce*. O modelo utilizado possibilita a previsão do desempenho do sistema em cargas variantes no tempo e permite que cenários do tipo “*what-if*” sejam simulados através da FT sem o ônus de execuções extras. A análise de resposta em frequência permitiu quantificar como a carga de trabalho é absorvida pelo sistema, evidenciando como as frequências são atenuadas. A abordagem adotada proporcionou um novo olhar acerca da geração de carga de trabalho ao incluir o modelo do sistema no processo. Os resultados obtidos permitiram (1) a especificação de um método para a geração de rajadas que pode ser configurado de acordo com a dinâmica de desempenho do sistema e (2) a demonstração de que duas cargas distintas impactaram de forma equivalente no sistema, destacando a importância de considerar o sistema ao utilizar a carga de trabalho no processo de *benchmark*. Os métodos apresentados neste artigo podem ser generalizados para sistemas em que o modelo de desempenho seja em função da carga de trabalho.

Como trabalhos futuros, espera-se estender a modelagem para sistemas não lineares (NARX) e de múltiplas entradas e saídas (MIMO). A resposta em frequência pode ser útil na implementação de políticas de *auto scaling* para evitar a criação de VMs durante o período transiente [da Luz et al. 2016]. Outro trabalho a ser explorado é a comparação

**Tabela 3. Correlações entre os sinais de entradas e saídas.**

Nome	Entradas	Saídas
Clark	0,6478	0,9509
NASA	0,5357	0,9198
Copa 98	0,9980	0,9998
Google	0,4330	0,7931

de técnicas tradicionais de modelagem de carga de trabalho com a abordagem proposta neste artigo, permitindo evidenciar as vantagens e desvantagens de cada técnica.

## Agradecimentos

Agradecemos CAPES, FAPESP, CNPq, LaSDPC/USP e IFSP pelo apoio financeiro, e também aos revisores anônimos por suas contribuições.

## Referências

- Centurion, A. M., Santana, M. J., Santana, R. C., and Bruschi, S. M. (2012). Impacto da carga de trabalho com rajadas no desempenho de serviços web. In *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 422–435.
- da Luz, H. J. F., Júnior, L. A. P., dos Santos de Souza, F. L., and Monaco, F. J. (2016). Modelagem analítica de sobrecarga transiente em sistemas computacionais por meio de parâmetros dinâmicos obtidos empiricamente. In *XIV Workshop de Computação em Clouds e Aplicações*, Salvador, BA. Sociedade Brasileira de Computação.
- Diao, Y., Hellerstein, J. L., Parekh, S., Shaikh, H., and Surendra, M. (2006). Controlling quality of service in multi-tier web applications. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, pages 25–25.
- Hellerstein, J. L., Diao, Y., Parekh, S., and Tilbury, D. M. (2004). *Feedback Control of Computing Systems*. Wiley-IEEE Press.
- Huang, D., He, B., and Miao, C. (2014). A survey of resource management in multi-tier web applications. *IEEE Communications Surveys Tutorials*, 16(3):1574–1590.
- Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.
- Jiang, Y., Sivalingam, L. R., Nath, S., and Govindan, R. (2016). Webperf: Evaluating what-if scenarios for cloud-hosted web applications. In *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference*, New York, NY, USA. ACM.
- LBL (2016). The internet traffic archive. <http://ita.ee.lbl.gov/>. Acessado: 01/12/2016.
- Ljung, L. (1999). *System Identification: Theory for the User*. Pearson Education.
- Mamani, E. L. C. (2016). *Metodologia de benchmark para avaliação de desempenho não-estacionária: um estudo de caso baseado em aplicações de computação em nuvem*. PhD thesis, PPG-CCMC ICMC USP.
- Mamani, E. L. C., Pereira, L. A., Santana, M. J., Santana, R. H. C., Nobile, P. N., and Monaco, F. J. (2015). Transient performance evaluation of cloud computing applications and dynamic resource control in large-scale distributed systems. In *High Performance Computing Simulation (HPCS), 2015 International Conference on*, pages 246–253.
- Mell, P. M. and Grance, T. (2011). Sp 800-145. the nist definition of cloud computing. Technical report, National Institute of Standards & Technology, Gaithersburg, USA.
- Menasce, D. (2002). TPC-W: a benchmark for e-commerce. *IEEE Internet Computing*.

- Papadopoulos, A. V., Maggio, M., Terraneo, F., and Leva, A. (2015). A dynamic modeling framework for control-based computing system design. *Mathematical and Computer Modelling of Dynamical Systems*, 21(3):251–271.
- Pereira, L. A. (2016). *Uma abordagem baseada em resposta em frequência para modelagem e avaliação de desempenho não estacionária em sistemas computacionais*. PhD thesis, PPG-CCMC ICMC-USP.
- Pereira, L. A., Mamani, E. L. C., Santana, M. J., Santana, R. H. C., Nobile, P. N., and Monaco, F. J. (2015a). Non-stationary simulation of computer systems and dynamic performance evaluation: A concern-based approach and case study on cloud computing. In *Computer Architecture and High Performance Computing (SBAC-PAD), 2015 27th International Symposium on*, pages 130–137.
- Pereira, Jr., L. A., Mamani, E. L. C., Santana, M. J., Santana, R. H. C., Monaco, F. J., and Nobile, P. N. (2015b). Extending discrete-event simulation frameworks for non-stationary performance evaluation: Requirements and case study. In *Proceedings of the 2015 Winter Simulation Conference, WSC '15*, Piscataway, NJ, USA. IEEE Press.
- Qu, C., Calheiros, R. N., and Buyya, R. (2016). Auto-scaling web applications in clouds: A taxonomy and survey. *arXiv Computing Research Repository (CoRR)*.
- Reiss, C., Wilkes, J., and Hellerstein, J. L. (2011). Google cluster-usage traces: format + schema. Technical report, Google, Inc. Posted at <https://github.com/google/cluster-data>.
- Santos, B., Carnivali, G., Almeida, W., Vieira, A. B., Ítalo Cunha, and Almeida, J. (2016). Caracterização do comportamento dos clientes de um sistema de vídeo ao vivo durante um evento de larga escala na internet. In *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 468–481.
- Silva, T. H., de Melo, P. O. S. V., Almeida, J., and Oliveira, A. A. F. (2013). Uma fotografia do instagram: caracterização e aplicação. In *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 455–468.
- Souza, F. (2016). Extensão da geração de carga do Bench4Q para benchmark de desempenho em regime transiente. Master's thesis, ICMC/USP.
- Veeraraghavan, K., Meza, J., Chou, D., Kim, W., Margulis, S., Michelson, S., Nishtala, R., Obenshain, D., Perelman, D., and Song, Y. J. (2016). Kraken: Leveraging live traffic tests to identify and resolve resource utilization bottlenecks in large scale web services. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 635–651, GA. USENIX Association.
- Yang, F. and Liu, J. (2012). Simulation-based transfer function modeling for transient analysis of general queueing systems. *European Journal of Operational Research*.
- Yuan, D., Joshi, N., Jacobson, D., and Oberai, P. (2013). Sryer: Netflix's predictive auto scaling engine - part 2. <https://goo.gl/6t1Hdb>. Acessado: 01/12/2016.
- Zhang, W., Wang, S., Wang, W., and Zhong, H. (2011). Bench4q: A qos-oriented e-commerce benchmark. In *Computer Software and Applications Conference (COMP-SAC), 2011 IEEE 35th Annual*, pages 38–47.