

# Modelagem analítica de sobrecarga transiente em sistemas computacionais por meio de parâmetros dinâmicos obtidos empiricamente

Helder Jefferson Ferreira da Luz<sup>1</sup>, Lourenço Alves Pereira Júnior<sup>1,2</sup>,  
Flávio Luiz dos Santos de Souza<sup>1</sup>, Francisco José Monaco<sup>1</sup>

<sup>1</sup>Universidade de São Paulo — USP

{helderfl, ljr, flaviols, monaco}@icmc.usp.br

<sup>2</sup>Instituto Federal de São Paulo — IFSP

ljr@ifsp.edu.br

**Abstract.** *This paper presents an empirical method for analytical modeling of transient overloads in computer systems by means of dynamical modeling. The technique, based on linear time-invariant approximations, aims at identifying the computer system's capacity in absorbing variations on the workload. The methodology relies on the parametrization of a dynamical model obtained from experimental procedures, considering abrupt, long-lasting disturbances, and the results are evaluated through comparison of the model prediction with the simulated system.*

**Resumo.** *Este trabalho apresenta um método empírico para modelagem analítica de sobrecargas transientes em sistemas computacionais por meio de modelagem dinâmica. A técnica baseada em aproximações lineares e invariantes no tempo, tem como objetivo identificar a capacidade de um sistema computacional absorver variações na carga de trabalho. A metodologia baseia-se na parametrização de um modelo dinâmico a partir de ensaios experimentais, considerando perturbações bruscas e de longa duração, e os resultados são avaliados por comparação das previsões do modelo em relação aos obtidos por simulação ou aferição.*

## 1. Introdução

A provisão de QoS em sistemas computacionais tem sido tema de expressivo trabalho de pesquisa, especialmente pela acelerada disseminação das aplicações de Internet em modalidades inovadoras, as quais incluem os avanços na computação móvel e em nuvem. Há diversas formas de atuar sobre os parâmetros de desempenho para o cumprimento do SLA (*Service Level Agreement*) associado, dentre elas o mecanismo de controle de admissão, que busca restringir a carga de trabalho imposta ao sistema de forma a não permitir sobrecarga na utilização de recursos e, em consequência, a violação do acordo de serviço. Com base nas propriedades das requisições recebidas e na capacidade atual do sistema em atender à demanda sem prejudicar os demais usuários, a política de admissão pode prever o descarte ou renegociação da ordem de serviço.

Para o correto projeto de uma política de admissão eficaz é importante o prévio conhecimento da capacidade do sistema frente à carga de trabalho a que deverá ser submetida, a fim de que se possa dimensionar os recursos e estabelecer os limites de aceitação

de novas requisições em tempo de execução. A capacidade de um sistema computacional pode ser estimada por meio de técnicas de avaliação de desempenho, onde há duas possibilidades básicas de abordagem: aferição e modelagem [Jain 1991] . A aferição possibilita a aquisição de resultados realísticos mediante à aplicação de carga sistema físico; no entanto, deve-se atentar a possíveis interferências de fatores externos não considerados, incluindo a própria instrumentação. A modelagem, por sua vez, consiste na abstração do sistema considerando as características que podem influenciar no resultado. De modo geral, o modelo pode ser abordado segundo duas alternativas: analiticamente ou por simulação. A primeira utiliza a representação matemática do sistema relacionando diretamente os estímulos às saídas. A simulação, diferentemente, utiliza uma representação equivalente do sistema por meio de algum outro sistema, de constituição diferente, mas de propriedades supostamente análogas, e que pode ser exercitado e mensurado — tem como vantagem a modelagem dos aspectos de interesse, sob o risco da negligência de fatores adversos que influenciem os resultados no sistema real.

Independentemente do método utilizado para a avaliação de desempenho, aferição ou modelagem, há outro aspecto que define a abrangência das conclusões obtidas a cerca da capacidade do sistema, conforme esse seja considerado em suas propriedades estáticas ou dinâmicas. Tradicionalmente, as abordagens de avaliação de sistemas computacionais extraem a análise estática. As propriedades dinâmicas do sistemas, visíveis por meio de avaliação não-estacionária, não são consideradas. Este artigo discute o emprego da avaliação de desempenho não-estacionário na modelagem de sobrecargas transientes. A técnica baseada em aproximações lineares e invariantes no tempo é capaz de representar o comportamento do sistema em estudo ao absorver variações na carga de trabalho. A metodologia baseia-se na parametrização de modelos auto-regressivos a partir de dados experimentais obtidos de perturbações bruscas e de longa duração. Os resultados são avaliados por meio de comparação entre os dados obtidos por simulação ou aferição em relação aos preditos pelo modelo.

## **2. Avaliação de capacidade estática vs. dinâmica**

Na sua forma mais amplamente explorada no domínio dos sistemas computacionais, a avaliação de desempenho é feita em regime estacionário, i.e., sob carga constante, e após desaparecimento de efeitos da inicialização. Essa forma de avaliação fornece dados para estimar a *capacidade estática do sistema*. Para esse fim, excita-se o sistema com a entrada de interesse, até a estabilização da resposta em seu valor final, seja este um valor constante ou variável dentro de um intervalo que compreende tanto a estocasticidade da resposta quanto flutuações periódicas inerentes ao processo. Essa condição é atingida passados os efeitos do assim referido período de *warm up* (aquecimento), em que o sistema está em estado transitório de inicialização dos seus diversos subsistemas.

A determinação da capacidade é feita através de estimulação do sistema até seu limite de desempenho aceitável, quando é determinada a carga estática máxima suportada e, desse modo, podem ser definidos os parâmetros para a política de admissão. Não obstante o amplo emprego da avaliação de capacidade estática em sistemas computacionais, tanto na pesquisa como na indústria, esta se limita aos resultados que podem ser obtidos pela análise em *regime estacionário*, desconsiderando os efeitos do período transitório. No entanto, diversas informações relevantes podem ser obtidas pela análise do comportamento nesse período. De fato, em *regime transiente*, a quantidade de recursos necessários

para garantir a QoS especificada pode variar consideravelmente em relação ao estado estacionário. Observa-se, por exemplo, que embora durante o regime estacionário possa haver recursos suficientes para a absorção da carga de trabalho apurada na avaliação de capacidade estática, durante um período transiente causado por uma perturbação na entrada, esses recursos podem se apresentar temporariamente insuficientes para manter o desempenho desejado. Isso pode ocorrer, por exemplo, devido a um aumento brusco no regime de carga (ou diminuição na capacidade do sistema por indisponibilidade temporária de recursos), resultando em uma falta no comprimento do SLA, mesmo que o aumento da carga a leve para um nível dentro daquele determinado como suficiente pela análise estacionária.

Como ilustração, seja um sistema mecânico onde uma mola mantém suspensa uma massa  $m$ . A lei de Hooke diz que a força  $F$  exercida por uma mola ideal é diretamente proporcional na direção contrária à distensão  $x$ , i.e.  $F = -k.x$ , em que  $k$  é uma característica da mola. Assim sob a ação da gravidade, e livre de outras forças externas, na condição de equilíbrio a mola é esticada da sua posição relaxada, até um alongamento  $x_1$ . Nesse ponto, a força exercida pela mola  $-k.x_1$  se anula perfeitamente com a força peso  $m_1.g$  entrando o sistema em regime estacionário. Nesse sistema, a mola precisa suportar uma massa sem que a distensão ultrapasse determinado limite. Na avaliação em regime estacionário, aumenta-se a carga gradativamente e observa-se a distensão final com a finalidade de observar se o limite estabelecido foi alcançado. Seguindo o procedimento, aumentando gradualmente a massa, chega-se a uma massa  $m_2$ , que estica a mola até o limite especificado, correspondendo ao limite da capacidade estática do sistema. Observe-se, contudo, que se ao invés de aumentar gradualmente o valor da massa, for adicionado um acréscimo súbito, de modo que a massa passe de  $m_1$  para  $m_2$  repentinamente, mesmo sem necessidade de uma análise matemática a partir das equações físicas, intuitivamente pode-se esperar que a mola distenda-se temporariamente além do ponto  $x_2$ , e oscile em torno deste — convergindo eventualmente para o ponto de equilíbrio no caso de um sistema real com perda de energia. Essa distensão transiente pode, inclusive, ultrapassar o limite estipulado, levando o sistema além do seu limite no período transitório.

Assim, uma mesma carga que em regime estacionário é determinada como suportável com base na avaliação de capacidade estática, no regime transiente pode mostrar-se excessiva dependendo da velocidade em que é aplicada ao sistema — o que pode ser antecipado por técnicas de avaliação de desempenho dinâmica. Em um sistema computacional, embora o efeito inercial seja usualmente não relevante para efeito de muitas aplicações convencionais, a literatura científica vem reconhecendo uma tendência no questionamento dessa hipótese, especialmente com a disseminação de sistemas distribuídos de grande escala [Xiong 2012, Hellerstein et al. 2012]. Observa-se, por exemplo, que enquanto a avaliação de capacidade estática determina a máxima carga suportada pelo sistema, a avaliação de *capacidade dinâmica* permite determinar a máxima taxa com que essa carga pode ser recebida [Mamani et al. 2015, Pereira et al. 2015b, Pereira et al. 2015a]. Com essa informação, é possível decidir se um determinado sistema é capaz de suportar determinadas perturbações, bem como calcular a quantidade de recursos necessários para tal. Ainda, a política de admissão pode ser projetada não apenas para aceitar ou rejeitar requisições em função da capacidade estática do sistema, mas também ajustar a taxa de entrada de requisições de modo a cumprir com requisitos de desempenho transiente como máxima queda de desempenho e máximo tempo de

restabelecimento ao nível desejado.

Este trabalho tem como objetivo a formulação de um método empírico para previsão de sobrecargas transientes em sistemas computacionais. Consideram-se perturbações bruscas e de duração prolongada em relação à dinâmica do sistema, por hipótese caracterizada como linear e invariante no tempo, e aproximável por um modelo de segunda ordem. Os resultados são demonstrados por meio de aplicações-exemplos em sistemas computacionais utilizando técnicas de simulação validadas na literatura.

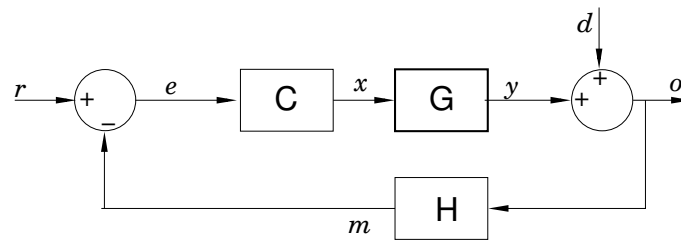
### 3. Trabalhos relacionados

Modelagem dinâmica é uma abordagem essencial para diversos ramos da Engenharia como as que tratam sistemas elétricos, mecânicos, hidráulicos e outros. Em seu instrumental analítico e de simulação conta-se com métodos de análise do transiente de *sistemas dinâmicos*, dos fatores internos e externos que influenciam a variável de saída e medidas, tais como dimensão do impacto, tempo de recuperação e qualidade da estabilização. A utilização de modelos dinâmicos em arquiteturas de sistemas computacionais para otimização de desempenho ou confiabilidade é, todavia mais recente e consideravelmente menos explorada que em outros domínios.

Essa abordagem de modelagem tem sido empregada, ainda que menos expressivamente, por exemplo, para clusterização dinâmica de processadores paralelos em problemas de particionamento de recursos, e para formatação de *throughput* em mecanismos de controle de congestão em redes de comunicação [Abdelzaher et al. 2003]. Escalonamento retroalimentado (*feedback scheduling*) é considerado um paradigma de ponta no campo de sistemas de tempo-real [Lu et al. 2003], especialmente em aplicações destinadas a ambientes não-determinísticos. Essa perspectiva tem sido também investigada em controle de admissão adaptativo em arquiteturas multicamadas [Kamra et al. 2004], controle de latência em servidores Web [Henriksson et al. 2004], bem como aplicações em computação em nuvem [Lim et al. 2009], redes de computadores [Xia et al. 2008], computação verde (*green computing*) [Wang et al. 2011].

Considerada ainda uma formação mais específica das engenharias, o reconhecimento da ampla aplicabilidade aos sistemas computacionais, especialmente os de grande escala e complexidade, as oportunidades tem inspirado também iniciativas explicitamente dedicadas a tornar mais acessível a teoria a cientistas da computação por meio de publicações especialmente dirigidas a esse fim [Hellerstein et al. 2005]. O projeto de estratégia de controle para implementar propriedades autonômicas de auto-gerenciamiento em resposta à perturbações em tempo de execução, constituem inerentemente exemplos onde o modelo do comportamento é transiente e implícito ao atendimento de requisitos de tempo-real. Este é o caso, por exemplo, de trabalhos como [Lu et al. 2006]. O que possibilita também um abordagem complementar a avaliações de desempenho de sistemas computacionais implantados (*deployed*) nas nuvens UFC [Coutinho et al. 2014, Muchalski and Maziero 2014]

Nobile [Nobile 2013] apresenta um estudo que demonstra que as características dinâmicas de um ambiente de computação em nuvem de larga escala tem impacto apreciável no desempenho dos mecanismos de provisão de QoS, e que a modelagem das propriedades de resposta transiente, bem como a aplicação de técnicas de Teoria de Controle podem ser justificadamente motivadas em projetos de mecanismos de controle



**Figura 1. Diagrama da arquitetura de controle de Nobile [Nobile 2013].**

realimentado em ambiente de recursos elásticos. O trabalho de Nobile fornece um roteiro para o projeto de um controle regulatório que gerencia, adaptativamente, o número de máquinas virtuais (VM) de um *datacenter*, de modo a manter a taxa de utilização do sistema constante, a despeito de variações na demanda.

Tendo o sistema proposto por Nobile como estudo de caso, o método apresentado no presente trabalho de pesquisa é utilizado como exemplo para modelar o comportamento dinâmico daquele ambiente. A comparação da modelagem analítica segundo a abordagem proposta com observações experimentais produzidas por simulação é avaliada e discutida nas próximas seções. O desafio e a contribuição propostos no trabalho incluem investigar se a complexidade de um sistema dessa dimensão, com todas as suas características não determinísticas e não lineares pode ser aproximado adequadamente por um modelo operacional simples e abordável analiticamente.

### 3.1. Arquitetura do sistema de Nobile

Em seu escopo, Nobile [Nobile 2013] propõe uma abordagem de implementação de um mecanismo adaptativo de alocação de recursos em arquitetura de computação em nuvem, a partir das técnicas de análise e projeto inspiradas em conceitos de controle clássico. O objetivo do mecanismo desenvolvido é regular a *taxa de utilização* dos recursos do *datacenter*, definida como a fração da capacidade do sistema que, em média, é utilizada. A taxa de utilização pode ser aumentada ou reduzida pelo respectivo aumento ou diminuição do desempenho do *datacenter*, proporcional ao número de VMs (recurso elástico). Assim, a estratégia de gerenciamento adaptativo de recursos consiste em ativar ou desativar máquinas virtuais a fim de manter a taxa de utilização regulada frente a perturbações na carga de trabalho.

A Figura 1 oferece uma representação em alto nível do sistema proposto. No diagrama, utilizando a notação convencional, os blocos representam as funções de transferência de cada um dos subsistemas, e as setas indicam seus respectivos sinais de entrada e saída. No diagrama o *datacenter* é representado pelo bloco  $G$ , sendo a entrada  $x$  a quantidade de máquinas virtuais desejadas, e a saída  $y$  a taxa de utilização resultante da carga de trabalho imposta ao sistema. O sinal  $d$  é uma perturbação imposta à carga causando o aumento ou diminuição na taxa de utilização final  $o$ . O bloco  $H$  é um sensor que mede a taxa de utilização e aplica uma média móvel exponencial, sendo  $m$  seu resultado. O sinal  $r$  é o valor da taxa de utilização desejada, sendo  $e$  o erro residual de  $r - m$ . O bloco  $C$  é o controlador que utiliza o erro  $e$  para gerenciar a quantidade de máquinas virtuais necessárias,  $x$ . O sistema foi simulado computacionalmente com o auxílio da ferramenta CloudSim [Buyya et al. 2009]. O software permite que ambientes de nuvem de larga escala compostos de vários nós sejam avaliados com a implementação simples

de um programa de simulação. Os resultados obtidos por meio do CloudSim, para a arquitetura desenvolvida, são coerentes com aqueles previstos pela modelagem proposta e, assim, evidenciam a validade da contribuição.

Nobile determina experimentalmente que a planta, i.e. *datacenter*, representado pelo bloco  $G$ , tendo como entrada e saída os sinais indicados, pode ter seu comportamento dinâmico aproximado por um sistema linear invariante no tempo (LTI) de primeira ordem, logo, respondendo assintoticamente na forma exponencial a uma entrada na forma de degrau, mudança súbita no número de máquinas virtuais (VMs). O laço de realimentação (sinal  $m$ ), eleva a ordem do sistema, de modo que em malha-fechada, o sistema tem comportamento oscilatório. Os parâmetros do controlador foram ajustados para que a dinâmica seja a de um sistema estável, com oscilações subamortecidas.

#### 4. Modelo empírico LTI no domínio da frequência

Para a realização dos estudos deste trabalho, considera-se a representação do sistema por meio de simulação e sua respectiva análise através de softwares que implementam métodos computacionais que solucionam os modelos dinâmicos a serem empregados. Desse procedimento resultam equações analíticas que representam o comportamento do sistema com respeito ao formato da saída, dada uma determinada entrada. A avaliação da qualidade desse modelo, no presente estudo, é feita comparando-se a previsão com os resultados gerados pelo simulador específico CloudSim [Buyya et al. 2009], que reproduz o comportamento de uma infraestrutura de computação em nuvem.

A formulação analítica da descrição de um sistema consiste basicamente em expressar matematicamente o comportamento de interesse. Quando a relação entrada-saída tem um comportamento dinâmico (i.e. a saída depende da entrada e do estado atual do sistema, caracterizando portanto uma memória “inercial”), o modelo corresponde a equações diferenciais. No caso de sistema que se apresentam lineares (sinais de entrada superpostos são combinados linearmente na saída), e cujos parâmetros são fixos (não alteram-se ao longo do processo), o sistema pode ser representado por um modelo linear invariante no tempo (LTI). Essa condição tem uma razão prática em Engenharia, pois admite soluções analíticas viáveis. No caso de sistemas discretos, a formulação é por meio de equações de diferenças que representam como a saída varia em função do tempo. O arcabouço matemático na teoria clássica de LTI de única entrada e única saída, é conveniente à representação das equações de diferença na sua forma equivalente no domínio da frequência, mediante a transformada Z, análoga à transformada de Laplace para o domínio contínuo. Nessa forma, cada bloco do sistema da Figura 1 denota uma *função de transferência* que converte a entrada na saída, tendo em conta os efeitos inerciais. As funções de transferências tem a propriedade de poderem ser multiplicadas algebricamente quando em arranjos sequenciais, de modo que, no diagrama, a relação  $y = g(c(x))$ , no domínio do tempo, se torna  $Y = FG \cdot X$ , no domínio da frequência complexa discreta Z.

Uma das formas de se obter um modelo de um sistema é a partir dos princípios básicos (i. e. leis da Física) que governam seu funcionamento. Essa forma, conhecida como análise de “caixa branca” é muito útil quando o conhecimento interno de um sistema pode ser convenientemente suposto, entendido e então modelado, o que é válido para alguns sistemas mecânicos ou elétricos, mas que pode ser de difícil aplicação em sistemas computacionais em função da complexidade e dos seus níveis de abstrações que

vão do software ao hardware. Nesse caso, pode ser conveniente a modelagem mediante outras técnicas nas quais o conhecimento tácito do comportamento interno não seja necessário. Técnicas assim são comumente referenciadas como de *caixa preta* e Hellerstein et al. (2004) propôs um fluxograma para aplicação de técnicas desse tipo, em sistemas computacionais.

Tendo sido o escopo bem definido, uma técnica bastante útil para parametrização do modelo, sumarizada por [Parekh et al. 2002], considera a utilização de modelos autor-regressivos para identificar sistemas invariantes no tempo que obedeçam aos princípios de causalidade e superposição. Enquanto o princípio de causalidade diz que uma resposta do sistema deve depender apenas das entradas anteriores a mesma, a superposição diz que se uma resposta do sistema é fruto da aplicação da combinação de duas entradas, então a mesma resposta pode ser obtida da soma das respostas individuais resultantes da aplicação individual das entradas consideradas. Esse princípio é especialmente útil para definição de sistemas lineares.

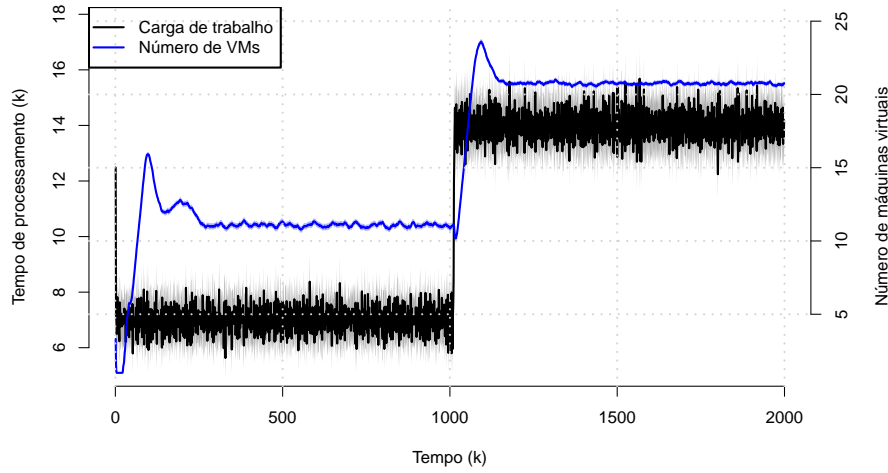
## 5. Estudo de caso

Nesse estudo, a arquitetura de Nobile, esquematizada na Figura 1, é tomada como uma caixa preta. Objetiva-se obter um modelo que informe como o número de máquinas virtuais (VMs), instanciadas e desativadas pelo gerenciador adaptativo de recursos elásticos, varia em função de um aumento da carga de trabalho. Evidentemente há uma relação proporcional entre as duas variáveis (pela ação do controlador), mas este estudo demonstra que essa relação é dinâmica e que existem componentes transientes. Isso significa que, se o sistema opera inicialmente com um número médio de máquinas, uma perturbação na carga de trabalho pode produzir um pico transitório no número de máquinas virtuais, que pode perdurar até que as novas máquinas absorvam a demanda e levem a saída a um novo patamar estacionário.

Considerando que em sistemas de computação em nuvem não há cobrança apenas de máquinas em execução, mas também das máquinas em processo de inicialização [Amazon 2014], a saída do modelo é tomada como o número de máquinas virtuais (VMs) alocadas, independente do seu estado. A variável de entrada do sistema escolhida é a carga de trabalho. Sendo  $N$  o número de máquinas virtuais e  $W$  a carga de trabalho dada por  $W = \mu/\lambda$ , em que  $\mu$  é o tempo de processamento (em unidades de tempo, *ut*) e  $\lambda$  o intervalo médio entre chegada das requisições (em unidades de tempo). A chegada de requisições é dada por uma distribuição exponencial com média  $\lambda$ .

Para determinar como o controlador responde às variações da carga de trabalho e encontrar uma função que defina a transformação de  $W$  em  $N$ , o sistema (modelo simulado no CloudSim) foi submetido a uma carga de trabalho em formato de degrau. O tempo de processamento foi fixado em  $\mu = 70 ut$ , e o intervalo entre chegadas foi inicialmente estabelecido em  $\lambda = 10 ut$ , produzindo uma carga de trabalho  $W = 7$ . No instante de tempo  $k = 1000$ , o intervalo entre chegadas foi reduzido para  $\lambda = 5 ut$ , o que aumentou bruscamente a carga de trabalho para  $W = 14$ . O experimento foi conduzido até o instante  $k = 2000$ , sendo efetuadas 300 replicações para enaltecer o comportamento do sistema. A Figura 2 mostra a média entre as replicações de  $W$  e  $N$  para cada instante de tempo.

Nota-se que ocorrem *overshoots* no número de VMs iniciadas tanto nos instantes



**Figura 2. Carga de trabalho ( $W$ ) e número de VMs previstas pelo controlador.**

de tempo entre 0 e 240 *ut* quanto entre 1000 e 1200 *ut*.

Pelo comportamento da saída ao degrau demonstrado no gráfico da Figura 2, tendo poucas oscilações e baixo ruído proveniente da estocasticidade do sinal de entrada do sistema, o comportamento pode ser aproximado ao de um sistema de segunda ordem. Utilizando a técnica de identificação de sistemas proposta por [Parekh et al. 2002] é possível definir um modelo autorregressivo com entradas exógenas (ARX) de segunda ordem que represente o comportamento do sistema, demonstrado pela equação (1), em que  $\hat{y}(k + 1)$  é a saída prevista no instante  $k + 1$  proveniente dos valores de entrada  $u$  e saída  $y$  nos instantes  $k$  e  $k - 1$ . Os parâmetros  $a_1$  e  $a_2$  indicam a influência das saídas anteriores em  $\hat{y}$  e  $b_1$  e  $b_2$  a influência das entradas. Considerando que após a modelagem do sistema, para prever  $\hat{y}$  haverá somente a variável de entrada  $W$  e as saídas anteriormente previstas com base na entrada, é necessário normalizar os valores de  $W$  e  $N$  de forma que o estado estacionário inicial fique em torno de 0.

$$\hat{y}(k + 1) = -a_1\hat{y}(k) - a_2\hat{y}(k - 1) + b_1u(k) + b_2u(k - 1) \quad (1)$$

Realizar operações e manipulações no domínio do tempo pode se apresentar como uma tarefa difícil sendo conveniente transformá-lo para um domínio que seja mais fácil trabalhar. Para sinais discretos, um método conhecido é o da transformada Z que atua no domínio da frequência. Aplicando a transformada Z no modelo ARX, obtêm-se a função de transferência (FT) do sistema (Equação 2) de  $W$  para  $N$ .

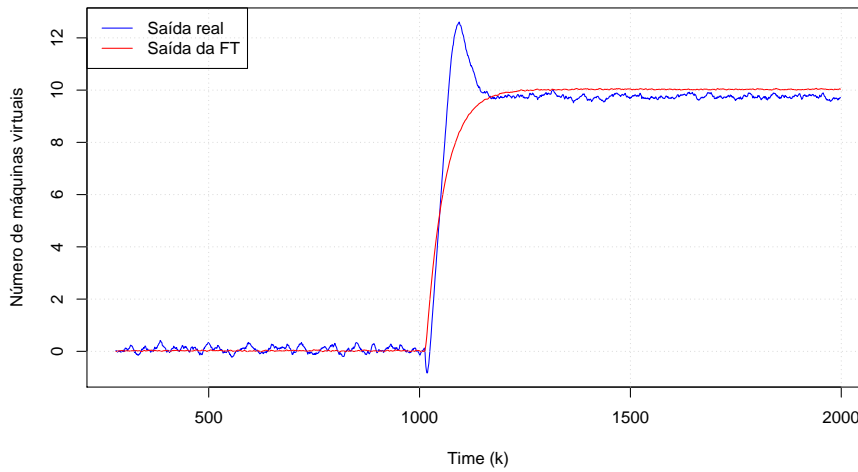
$$F(z) = \frac{b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \quad (2)$$

Para parametrizar a equação foi utilizado o método dos mínimos quadrados polinomial [Ljung 1999], obtendo-se os valores dos parâmetros  $a_1$ ,  $a_2$ ,  $b_1$ ,  $b_2$  do modelo ARX e por consequência, da função de transferência (Equação 3).



$$F(z) = \frac{0.005191z^{-1} + 0.009111z^{-2}}{1 - 1.525z^{-1} + 0.5353z^{-2}} \quad (3)$$

Para averiguar a adequação da função de transferência obtida, a carga real do sistema é aplicada ao modelo analítico via função de transferência (modelo ARX), utilizando o software Matlab<sup>1</sup>. Foi utilizado a média do número de máquinas virtuais (N) e carga de trabalho (W) de outras 300 replicações. É possível observar (Figura 3) que o modelo não ajustou corretamente o degrau ocorrido, se comportando como um sistema de primeira ordem. Isso aconteceu devido a estocasticidade do sinal de entrada, demonstrado no gráfico da Figura 2, denotando a insuficiência de um modelo de segunda ordem em se ajustar ao ruído em W.



**Figura 3. Saída prevista pelo modelo e saída gerada pelo controlador.**

Para que o modelo se ajustasse corretamente à saída do controlador, foi necessário a utilização de um filtro para remover a estocasticidade das variáveis de entrada e da saída. Assim, considerou-se os sinais de interesse como processos estocásticos estacionários, cujos os valores correspondem a médias no tempo. Foi utilizado um filtro FIR (*Finite Impulse Response*) passa-baixa, que é bastante utilizado na área de processamento de sinais digitais para redução de ruído. O filtro FIR é descrito por um conjunto de  $p$  coeficientes polinomiais,  $h_0, \dots, h_{p-1}$ , e sua saída é dada exclusivamente pelos últimos  $p$  termos da entrada como mostrado na Equação (4). Utilizamos um *design* baseado em janela de Hamming [Tan and Jiang 2013] com 40 termos e frequência de corte de 10% da frequência máxima, o que eliminou consideravelmente o ruído. Pode-se observar a redução do ruído na entrada  $W$  pela comparação do espectro de frequências na Figura 4.

$$y(n) = \sum_{i=0}^{p-1} h_i x(n-i) \quad (4)$$

<sup>1</sup>Matlab: <http://www.mathworks.com/products/matlab/>

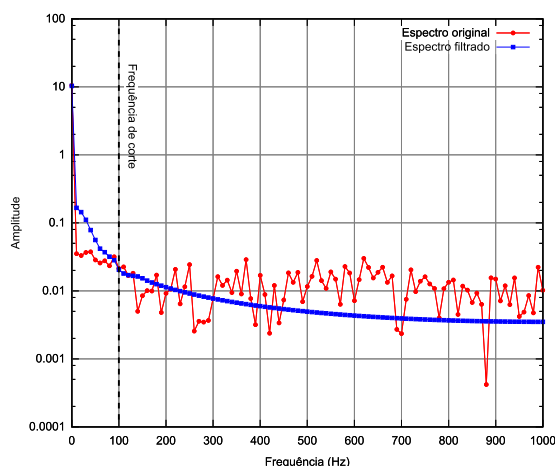


Figura 4. Ruído em W após utilizar o filtro em relação a W sem filtro.

Utilizando as variáveis filtradas foi criada uma nova função de transferência (Equação 5) e verificado o seu ajuste, apresentado na Figura 5. Nota-se visualmente o bom ajuste do modelo nos períodos estacionários e a aproximação da resposta ao degrau no modelo em relação a saída do controlador. O valor na Tabela 1 de  $R^2$  (variabilidade) próximo de 1, do baixo valor do *Root-mean-squared Error* — RMSE e a forte correlação positiva da entrada em relação a saída do modelo também corroboram com o bom ajuste.

$$F(z) = \frac{-0.01024z^{-1} + 0.01312z^{-2}}{1 - 1.96z^{-1} + 0.9617z^{-2}} \quad (5)$$

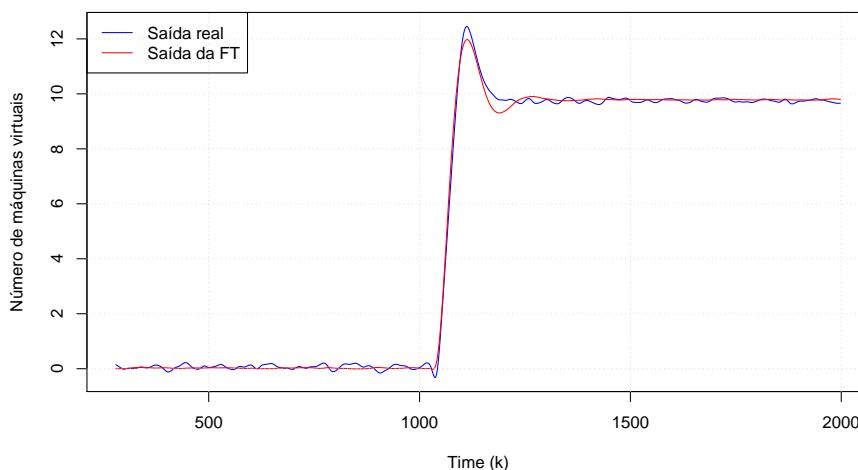


Figura 5. Saída prevista pelo modelo e saída gerada pelo controlador.

## 6. Caracterização de sobrepasse e tempo de assentamento

Duas propriedades fundamentais de sistemas de segunda ordem são a razão de amortecimento  $\zeta$  e a frequência natural  $\omega$ . Em um sistema massa-mola,  $\zeta$  corresponde ao coeficiente

**Tabela 1. Valores de  $R^2$ ,  $RMSE$  e correlação entre entrada e a saída do modelo estimado**

Teste	Valor calculado
$R^2$	0.99900
$RMSE$	0.15429
Correlação	0.97332

ciente de amortecimento da mola, que define a quantidade de energia que a mola perde por metro; e  $\omega$  corresponde à frequência natural da mola, que ela exibiria na ausência de outras forças [Ogata 2009].

Tais características são definidas para sistemas de tempo contínuo, que são analisados pela transformada de Laplace. Dada a função de transferência  $G(s)$  de um sistema de segunda ordem contínuo, podemos realizar algumas caracterizações teóricas observando seus polos, isto é, as raízes do denominador. Na forma canônica de uma função de transferência contínua de 2ª ordem, observamos

$$G(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{s^2 + as + b} = \frac{N(s)}{s^2 + 2\zeta\omega s + \omega^2}$$

isto é,  $\omega = \sqrt{b}$  e  $\zeta = a/(2\sqrt{b})$ .

Para obter estes parâmetros para um sistema discreto, definido no espaço da transformada Z, é necessário aproximar seu comportamento em tempo contínuo. A transformação bilinear, dada na Equação (6), fornece uma aproximação de 1ª ordem.

$$z = \exp(sT) = \frac{\exp(sT/2)}{\exp(-sT/2)} \approx \frac{1 + sT/2}{1 - sT/2} \quad (6)$$

em que  $T$  é o período de amostragem do sinal discreto.

Quando a razão de amortecimento  $\zeta$  é inferior a um valor crítico, o sistema mostra-se sub-amortecido; quando excitado por uma entrada degrau, o sinal apresenta oscilações decrescentes até atingir um estado estacionário. Podemos determinar características teóricas do sistema a partir da razão de amortecimento  $\zeta$  e da frequência natural  $\omega$ . Neste caso é de interesse caracterizar a sobrepassagem (*overshoot*),  $M_p$ , definida como o máximo valor alcançado pelo sinal, e a sobrepassagem relativa,  $O_p$ , dada como a razão da diferença do valor de pico em relação ao valor estacionário do sinal  $x_\infty$ , dado na equação 7. Esta é uma medida de relevância para sistemas de segunda ordem, pois impacta no dimensionamento de recursos e na máxima perturbação suportada [Ogata 2009].

$$O_p = \frac{M_p - x_\infty}{x_\infty} = \exp\left(-\frac{\pi\zeta}{\sqrt{1 - \zeta^2}}\right) \quad (7)$$

Também, é interessante caracterizar o tempo de assentamento (*settling time*)  $T_s$ , definido como o máximo tempo entre a perturbação e o sinal se apresentar dentro de uma certa faixa ao redor do valor estacionário [Ogata 2009]. Seja uma faixa de tolerância  $\delta$ , podemos obter uma aproximação teórica dada pela Equação (8), válida para  $\zeta \ll 1$ :

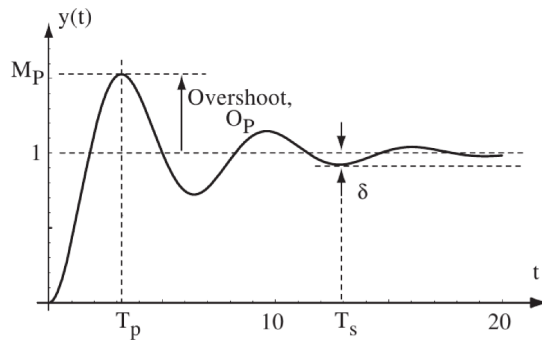


Figura 6. Exemplo de resposta de um sistema de 2ª ordem a uma entrada degrau.

$$T_s = \frac{\ln \delta}{\zeta \omega} \quad (8)$$

A sobrepassagem e o tempo de assentamento estão ilustrados na Figura 6.

Utilizando a transformação bilinear dada na Equação (6), obtivemos o sistema aproximado de tempo contínuo  $\tilde{G}(s)$  equivalente a  $G(z)$ , dado na Equação (9):

$$\tilde{G}(s) = \frac{0.005957s^2 - 0.001338s + 2.938 \times 10^{-5}}{s^2 + 0.003906s + 1.734 \times 10^{-5}} \quad (9)$$

Com isto extrairam-se as características teóricas do sistema, listados na coluna “Teórico” da Tabela 6. Mensuramos a sobrepassagem e tempo de assentamento do sinal de saída real e comparamos com estes valores teóricos, como colocados na Tabela 6. Notamos um excelente ajuste da previsão da sobrepassagem e um bom ajuste da previsão do tempo de assentamento.

Tabela 2. Comparação de características do modelo analítico com o mensurado do sistema

	Teórico degrau unitário	Modelo analítico	Mensurado do sistema	Erro absoluto	Erro relativo
$(\omega)$	$4.164 \times 10^{-3}\text{Hz}$	-	-	-	-
$(\zeta)$	0.469	-	-	-	-
$(O_p)$	18.86%	22.36%	27.76%	5.4%	24.15%
$(T_s)$	2003rtu	1790rtu	1370rtu	420rtu	30.65%

## 7. Conclusões

Este trabalho apresentou os resultados de um estudo sobre modelagem de sobrecarga transiente em sistemas computacionais. A principal contribuição é a formulação de um método para modelagem do sistema na forma de uma representação analítica capaz de reproduzir o comportamento dinâmico do sistema. Um dos resultados deste trabalho é a viabilidade de representar a complexidade de um *datacenter* por um modelo simples, especificamente um sistema dinâmico LTI, possível de ser tratado por técnicas clássicas de análise de sistemas lineares. Válidas as hipóteses sobre o sistema, o modelo analítico

permite prever sobrecargas decorrentes de variações bruscas de carga de trabalho, ainda aquelas de duração transitória, e que se dissipam, mas que podem implicar em um esforço superior àquele exigido do sistema nas condições estacionárias iniciais e finais.

O trabalho contribui na exploração do emprego de modelos dinâmicos em sistemas computacionais, que começa a tornar-se relevante em função da proliferação de sistemas distribuídos de larga escala. E ainda demonstra que um modelo analítico LTI de baixa ordem prevê com boa aproximação o comportamento dinâmico de um sistema complexo de larga escala.

## Agradecimentos

Os autores agradecem à CAPES, à FAPESP, ao CNPq, ao LaSDPC/ICMC/USP e ao IFSP pelo apoio financeiro e infraestrutura cedida para realização deste trabalho.

## Referências

- Abdelzaher, T. F., Stankovic, J. A., Lu, C., Zhang, R., and Lu, Y. (2003). Feedback performance control in software services. *IEEE Control Systems*, 23(3):74–90.
- Amazon (2014). Amazon ec2 faqs. <http://aws.amazon.com/ec2/faqs>. Acesso: 08/04/2016.
- Buyya, R., Ranjan, R., and Calheiros, R. N. (2009). Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In *2009 International Conference on High Performance Computing & Simulation*. IEEE.
- Coutinho, E. F., Gomes, D. G., and de Souza, J. N. (2014). Uma proposta de *framework* conceitual para análise de desempenho da elasticidade em nuvens computacionais. In *XII Workshop de Computação em Clouds e Aplicações (WCGA)*, pages 43–54.
- Hellerstein, J., Parekh, S., Griffith, R., Kaiser, G., and Phung, D. (2005). A control theory foundation for self-managing computing systems. *IEEE Journal on Selected Areas in Communications*, 23(12):2213–2222.
- Hellerstein, J. L., Kohlhoff, K. J., and Konerding, D. E. (2012). Science in the Cloud: Accelerating Discovery in the 21st Century. *IEEE Internet Computing*, 16(4):64–68.
- Henriksson, D., Lu, Y., and Abdelzaher, T. (2004). Improved prediction for web server delay control. *Proceedings. 16th Euromicro Conference on Real-Time Systems, 2004. ECRTS 2004.*, pages 61–68.
- Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley Computer Publishing, John Wiley & Sons, Inc.
- Kamra, a., Misra, V., and Nahum, E. (2004). Yaksha: a self-tuning controller for managing the performance of 3-tiered web sites. *Twelfth IEEE International Workshop on Quality of Service, 2004. IWQOS 2004.*, pages 47–56.
- Lim, H. C., Babu, S., Chase, J. S., and Parekh, S. S. (2009). Automated control in cloud computing: Challenges and opportunities. In *Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds, ACDC '09*, pages 13–18, New York, NY, USA. ACM.

- Ljung, L. (1999). *System Identification: Theory for the User*. Pearson Education.
- Lu, C., Lu, Y., Abdelzaher, T. F., Stankovic, J. A., and Son, S. H. (2006). Feedback Control Architecture and Design Methodology for Service Delay Guarantees in Web Servers. *IEEE Transactions on Parallel and Distributed Systems*, 17(9):1014–1027.
- Lu, Y., Abdelzaher, T., Lu, C., Sha, L., and Liu, X. (2003). Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers. In *Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE*, pages 208–217.
- 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.
- Muchalski, F. J. and Maziero, C. A. (2014). Alocação de máquinas virtuais em ambientes de computação em nuvem considerando o compartilhamento de memória. In *XII Workshop de Computação em Clouds e Aplicações (WCGA)*, pages 81–92.
- Nobile, P. N. (2013). *Projeto de um Broker de Gerenciamento Adaptativo de Recursos em Computação em Nuvem Baseado em Técnicas de Controle Realimentado*. PhD thesis, ICMC-USP.
- Ogata, K. (2009). *Modern Control Engineering*. Prentice-Hall, 5th edition.
- Parekh, S., Gandhi, N., and Hellerstein, J. (2002). Using control theory to achieve service level objectives in performance management. *Real-Time Systems*, 23:127–141.
- Pereira, L. A., Mamani, E. L. C., Santana, M. J., Monaco, F. J., and Nobile, P. N. (2015a). Extending discrete-event simulation frameworks for non-stationary performance evaluation: requirements and case study. In *2015 Winter Simulation Conference (WSC)*, pages 3150–3151.
- Pereira, L. A., Mamani, E. L. C., Santana, M. J., Santana, R. H. C., Nobile, P. N., and Monaco, F. J. (2015b). 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.
- Tan, L. and Jiang, J. (2013). *Digital signal processing: fundamentals and applications*. Academic Press.
- Wang, Y., Wang, X., Chen, M., and Zhu, X. (2011). Partic: Power-aware response time control for virtualized web servers. *IEEE Transactions on Parallel and Distributed Systems*, 22(2):323–336.
- Xia, F., Sun, Y., and Tian, Y. (2008). Feedback scheduling of priority-driven control networks. *CoRR*, abs/0806.0130.
- Xiong, P. (2012). Dynamic management of resources and workloads for RDBMS in cloud. In *Proceedings of the on SIGMOD/PODS 2012 PhD Symposium - PhD '12*, page 63, New York, New York, USA. ACM Press.