# Inferência de Desempenho: Uma Nova Abordagem para o Planejamento da Capacidade de Aplicações na Nuvem

**Marcelo Gonçalves, Matheus Cunha, Américo Sampaio, Nabor C. Mendonça**

[1]Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Universidade de Fortaleza (UNIFOR)
Av. Washington Soares, 1321, Edson Queiroz, CEP 60811-905 Fortaleza, CE

{marcelocg,mathcunha}@gmail.com,{americo.sampaio,nabor}@unifor.br

***Resumo.*** *Este trabalho propõe uma nova abordagem para apoiar o planejamento da capacidade de aplicações em nuvens que oferecem infraestrutura-como-serviço (IaaS). A abordagem proposta tem como premissa a existência de uma relação de capacidade entre diferentes configurações de recursos de um dado provedor de nuvem IaaS, com a qual é possível prever (ou "inferir"), com alta precisão, o desempenho esperado de uma aplicação para certas configurações de recursos e cargas de trabalho, tendo com base o desempenho da aplicação observado para outras configurações de recursos e cargas de trabalho neste mesmo provedor. Resultados empíricos preliminares, obtidos a partir da avaliação do desempenho de uma popular aplicação de* blogging *(WordPress) em um provedor de nuvem público (Amazon EC2), mostram que a nova abordagem consegue reduzir significativamente (acima de 80%) o número total de cenários de implantação da aplicação que precisam de fato ser avaliados na nuvem.*

***Abstract.*** *This work proposes a novel approach to support application capacity planning in infrastructure-as-a-service (IaaS) clouds. The proposed approach relies on the assumption that there exists a capacity relation between different resource configurations offered by a given IaaS cloud provider, enabling one to predict (or "infer"), with high accuracy, an application's expected performance for certain resource configurations and workloads, based upon its observed performance for other resource configurations and workloads in that same provider. Preliminary empirical results, obtained from evaluating the performance of a well-known blogging application (WordPress) in a public cloud provider (Amazon EC2), show that the proposed approach can significantly reduce (over 80%) the total number of application deployment scenarios that need to be effectively tested in the cloud.*

## 1. Introdução

Um dos principais desafios enfrentados pelos usuários de nuvens que oferecem infraestrutura-como-serviço (IaaS) é planejar adequadamente a capacidade dos recursos da nuvem necessários para atender as demandas específicas de suas aplicações [Menascé and Ngo 2009]. Parte desse desafio envolve tentar descobrir a melhor maneira de implantar a aplicação na nuvem, considerando os vários tipos de recursos (em particular, máquinas virtuais) oferecidos pelo provedor, sob a perspectiva de diferentes requisitos e critérios de qualidade [Gonçalves Junior et al. 2015].

Em geral, provedores de nuvens IaaS cobram seus usuários em função do tempo de utilização dos recursos solicitados, cujos preços variam conforme a capacidade (normalmente medida por características técnicas como quantidade de núcleos de processamento, tamanho de memória e espaço de armazenamento) de cada recurso. Dessa forma, para calcular o custo de operação de uma aplicação na nuvem, é preciso estimar ou medir como a aplicação responderá a diferentes níveis de demanda, em termos de indicadores de desempenho como tempo de resposta ou vazão, quando executada sob diferentes configurações e perfis de máquinas virtuais. Na prática, isso significa que cabe ao usuário da nuvem identificar, dentre as possíveis configurações de máquinas virtuais ofertadas por um ou mais provedores de nuvem, aquelas de menor custo capazes de executar a aplicação mantendo-se níveis satisfatórios para os indicadores de desempenho.

Um grande problema começa a se desenhar para o usuário da nuvem ao seguir essa abordagem: a fase de avaliação da aplicação pode atingir patamares elevados de tempo e custo, em razão das necessidades de variação da demanda, da arquitetura de implantação e das configurações de recursos utilizadas para hospedar cada camada da aplicação [Silva et al. 2013]. Ainda que certos provedores IaaS ofereçam descontos ou pacotes de horas grátis para novos clientes, em geral esses incentivos, por estarem limitados a máquinas de pequeno porte, são insuficientes para suportar a carga de uma aplicação real em produção. Assim, executar uma aplicação real, tipicamente implantada em arquitetura de várias camadas [Jayasinghe et al. 2011], em máquinas virtuais de tamanho considerável e por longos períodos de tempo, apenas para estudar o seu comportamento, pode se traduzir em um custo alto que dificulte ou até mesmo inviabilize o próprio projeto de migração dessa aplicação para a nuvem [Beserra et al. 2012].

Vários trabalhos já foram propostos com o intuito de apoiar o planejamento da capacidade de aplicações em nuvens IaaS. Em linhas gerais, esses trabalhos podem ser classificados de acordo duas abordagens distintas quanto à estratégia de avaliação do desempenho da aplicação. Trabalhos que seguem a primeira abordagem, referenciada neste trabalho como *abordagem preditiva*, visam estimar ou simular o desempenho esperado da aplicação para determinadas configurações de recursos e determinados níveis de carga, sem necessariamente ter que implantá-la na nuvem [CloudHarmony 2014, Malkowski et al. 2010, Fittkau et al. 2012, Li et al. 2011, Jung et al. 2013]. Apesar do baixo custo oferecido aos usuários, que não precisam pagar por recursos de nuvem durante a fase de avaliação, esse trabalhos têm como maior limitação a ainda baixa precisão das técnicas de predição de desempenho, particularmente daquelas baseadas em simulação [Fittkau et al. 2012]. Já os trabalhos que fazem parte da segunda abordagem, aqui referenciada como *abordagem empírica*, tem como objetivo medir o desempenho real da aplicação através de sua efetiva implantação na nuvem e da realização de testes de carga [Jayasinghe et al. 2012, Silva et al. 2013, Cunha et al. 2013a, Scheuner et al. 2014]. Por executarem a aplicação no próprio ambiente de nuvem, esses trabalhos conseguem resultados significativamente mais precisos no que diz respeito à seleção das melhores configurações de recursos para cargas de trabalho específicas. No entanto, uma limitação importante desses trabalhos é a necessidade de se testar exaustivamente uma grande quantidade de configurações de recursos e cargas de trabalho, implicando em altos custos durante a fase de avaliação.

Visando combinar as vantagens das abordagens preditiva e empírica, este trabalho

propõe uma nova maneira de apoiar os usuários de nuvens IaaS a identificarem as melhores (i.e., mais baratas) configurações de recursos capazes de satisfazer as demandas específicas de suas aplicações. A nova abordagem tem como premissa a existência de uma relação de capacidade entre diferentes configurações de recursos oferecidas por um dado provedor de nuvem, com a qual é possível prever (ou "inferir"), com alta precisão, o desempenho esperado da aplicação para determinadas configurações de recursos. A predição ou inferência é realizada com base no desempenho observado da aplicação para outras configurações de recursos e cargas de trabalho no mesmo provedor. Por exemplo, se a aplicação atendeu satisfatoriamente a demanda para uma configuração de recursos de determinada capacidade sob uma determinada carga de trabalho, é muito provável que ela também vá atendê-la para outras configurações de maior capacidade sob a mesma carga de trabalho. Analogamente, se a aplicação não atendeu a demanda para uma determinada configuração de recursos sob uma determinada carga de trabalho, muito provavelmente ela também não irá atendê-la para a mesma configuração sob cargas de trabalho maiores. Através do uso de inferência, a abordagem permite avaliar uma ampla variedade de cenários de implantação da aplicação, sendo que apenas uma pequena parte desses cenários precisa de fato ser implantada e executada na nuvem. Dessa forma, a abordagem consegue obter o melhor das duas abordagens previamente citadas, produzindo resultados de alta precisão (característicos da abordagem empírica) mas com significativa redução de custo (característica da abordagem preditiva).

A próxima seção apresenta um novo processo de avaliação de capacidade para aplicações na nuvem, fundamentado no conceito de inferência de desempenho. A Seção 3 descreve os resultados de uma avaliação preliminar do novo processo envolvendo a implantação de uma aplicação real (WordPress) em um provedor de nuvem IaaS público (Amazon EC2). A Seção 4 compara o novo processo com outros trabalhos relacionados. Por fim, a Seção 5 oferece algumas conclusões e sugestões para trabalhos futuros.

## 2. Processo de Avaliação de Capacidade por Inferência de Desempenho

### 2.1. Conceitos e Terminologia

Antes de apresentarmos o processo, é necessário definirmos alguns conceitos importantes relacionados ao domínio da avaliação da capacidade de aplicações na nuvem (ver Tabela 1). A definição desses conceitos também serve para estabelecer a terminologia que será utilizada na descrição do processo, feita a seguir.

### 2.2. Dados de Entrada

O principal dado de entrada esperado pelo processo é o valor de referência (SLA), o qual será usado para determinar se a aplicação atingiu os requisitos mínimos de desempenho exigidos em cada cenário de execução. Além do SLA, o processo precisa também conhecer quais são as cargas de trabalho sob as quais o desempenho da aplicação deverá ser avaliado. Outro dado importante que deve ser passado como entrada para o processo é o espaço de implantação da aplicação. Para isso, o processo deve ser alimentado com três parâmetros: (i) uma lista de tipos de máquinas virtuais fornecidos pelo provedor no qual deseja-se hospedar a aplicação; (ii) a quantidade máxima de máquinas virtuais de cada tipo que irá compor cada configuração a ser avaliada; e (iii) um critério para estabelecimento das relações de capacidade entre as configurações do espaço de implantação.

**Tabela 1. Conceitos e terminologia utilizados no artigo.**

| Conceito | Definição |
|---|---|
| *Aplicação sob teste* | Um sistema computacional, possivelmente implementado em uma arquitetura multicamadas, para o qual se deseja observar o comportamento em um ambiente de computação em nuvem e ao qual estão associadas uma ou mais *métricas de desempenho*. |
| *Métrica de desempenho* | Uma característica ou comportamento mensurável de forma automatizada e comparável a um *valor de referência*, capaz de indicar o grau de sucesso de uma execução da aplicação sob teste. É dependente do domínio da aplicação. Ex.: tempo de resposta, quadros por segundo. |
| *Valor de referência (SLA)* | Um valor predefinido como minimamente aceitável para uma métrica de desempenho após uma execução da aplicação sob teste. Este valor, também referenciado neste trabalho como SLA (*Service Level Agreement*), serve como base de comparação para que se classifique a aplicação como capaz de ser executada em uma certa *configuração* de máquinas virtuais e sob uma certa *carga de trabalho*. |
| *Provedor* | Uma empresa que fornece recursos computacionais como serviço cobrado financeiramente por fração de tempo de utilização. Neste trabalho, o foco será em provedores que disponibilizam recursos de infraestrutura, notadamente *máquinas virtuais*. |
| *Tipos de máquinas virtuais* | Classificam as máquinas virtuais fornecidas por um provedor conforme suas características técnicas (e.g., núcleos de processamento, tamanho de memória, espaço em disco), permitindo que o provedor mantenha uma linha de produtos discreta e finita. |
| *Categorias* | Agrupam os tipos de máquinas virtuais de um provedor de acordo com suas características técnicas, plataforma e/ou arquitetura de hardware e a natureza do uso a que se destinam. Ex.: categorias que priorizam consumo de memória, acesso a disco, processamento gráfico, etc. |
| *Configuração* | Um conjunto de máquinas virtuais de um mesmo tipo e, portanto, de uma mesma categoria. *Configurações* são usadas para implantar uma ou mais camadas arquiteturais (ex.: apresentação, negócio, persistência) da aplicação sob teste. |
| *Espaço de implantação* | Denota um conjunto limitado de configurações de máquina virtuais nas quais a aplicação sob teste será implantada e executada durante uma sessão de avaliação. |
| *Relações de capacidade* | Relativizam o poder computacional das diversas configurações que compõem o espaço de implantação. As *relações de capacidade* definem um grafo orientado sobre o espaço de implantação onde os vértices correspondem às configurações e as arestas indicam a superioridade ou inferioridade (dependendo da direção da aresta) de uma configuração em relação a outra em termos de poder computacional. |
| *Níveis de capacidade* | Estabelecem uma hierarquia sobre as relações de capacidade definidas entre as configurações do espaço de implantação. Nessa hierarquia, configurações classificadas em um mesmo nível de capacidade seriam equivalentes (ou indistinguíveis) em termos de poder computacional. |
| *Carga de trabalho* | Representa o tamanho da demanda que será imposta à aplicação sob teste em uma execução. Sua unidade de medida é dependente do domínio da aplicação. Ex.: tamanho dos arquivos de entrada para uma aplicação de compactação de arquivos, quantidade de usuários concorrentes para uma aplicação web, etc. |
| *Execução* | Corresponde à execução da aplicação sob teste utilizando uma determinada configuração de máquinas virtuais e submetida a uma determinada carga de trabalho. O resultado de uma *execução* fornece indicadores que permitirão avaliar se a aplicação atingiu o valor de referência esperado para uma determinada métrica de desempenho naquele cenário. |

Para dar um exemplo de como os três parâmetros acima são utilizados na construção do espaço de implantação, considere que o processo recebeu um único tipo de máquina virtual; o valor 3 como sendo a quantidade máxima de máquinas virtuais por configuração; e a quantidade de máquinas virtuais de cada configuração como critério para estabelecimento das relações de capacidade. Nesse caso, o espaço de implantação seria composto por 3 configurações distintas, contendo 1, 2 e 3 máquinas do tipo passado como parâmetro, respectivamente. Além disso, configurações maiores (ou seja, contendo um maior número de máquinas virtuais) seriam classificadas acima de configurações menores na hierarquia de níveis de capacidade estabelecida sobre o espaço de implantação.

## 2.3. Atividades

As principais atividades executadas pelo processo de avaliação de capacidade são ilustradas no diagrama da Figura 1. As atividades destacas com o rótulo "≪A≫" são atividades abstratas, devendo ser customizadas pelos usuários do processo de acordo com diferentes estratégias de avaliação (descritas na seção 2.4). As outras atividades são executadas de

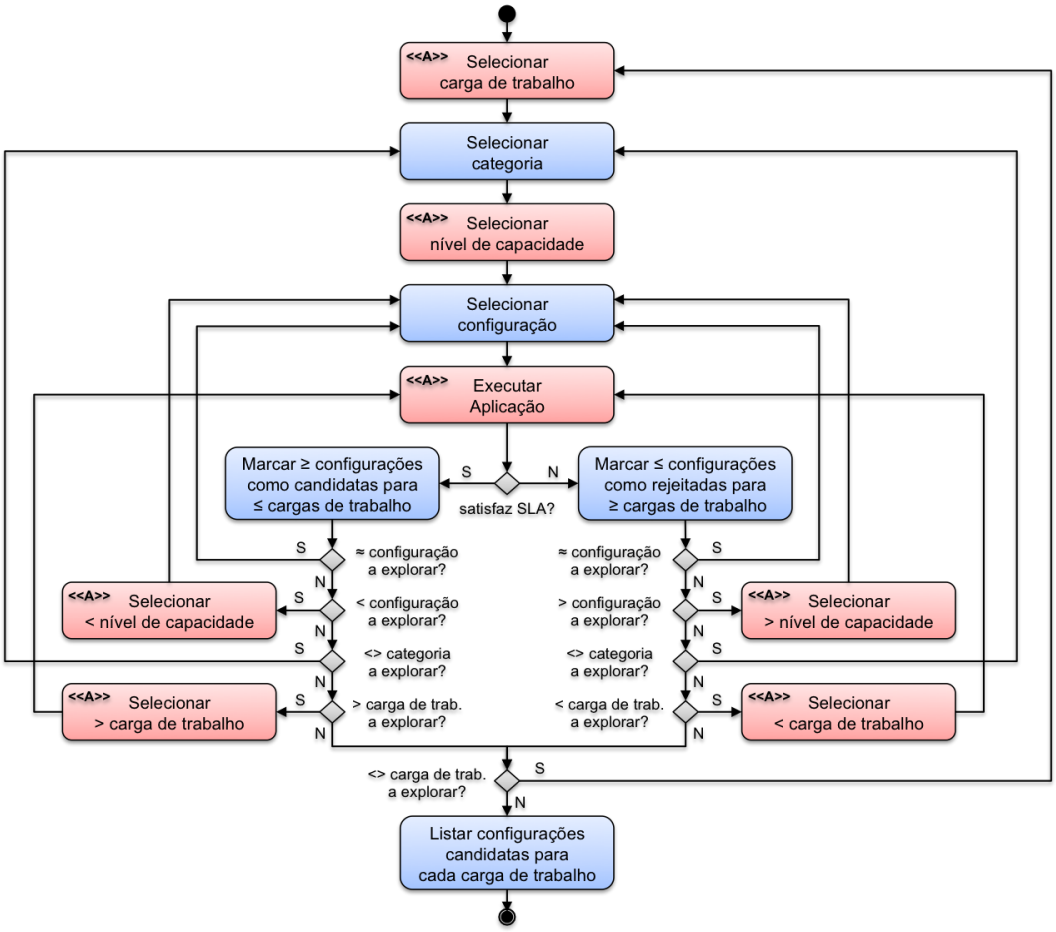



**Figura 1. Diagrama de atividades do processo de avaliação de capacidade.**

maneira idêntica independentemente de qual seja a aplicação sob teste ou de qual seja a estratégia de avaliação utilizada.

A execução do processo acontece em quatro fases bem distintas e cíclicas: seleção do cenário de execução da aplicação; execução da aplicação; inferência de desempenho; e seleção do próximo cenário. Cada uma dessas fases será detalhada a seguir.

### 2.3.1. Seleção do cenário de execução

A primeira atividade dessa fase é a escolha de uma carga de trabalho. Essa é uma atividade abstrata, significando que diferentes estratégias podem ser empregadas nessa escolha, por exemplo, selecionando um carga de trabalho maior ou menor dentre aquelas fornecidas como dados de entrada ao processo. Depois de selecionar a carga inicial, o processo seleciona uma categoria de máquinas virtuais. No caso da categoria, a ordem ou método utilizado na escolha é irrelevante para o processo, uma vez que todas as categorias do espaço de implantação deverão ser avaliadas. Em seguida, o processo seleciona um nível de capacidade dentre aqueles presentes no espaço de implantação. Essa também é uma atividade abstrata, uma vez que níveis de capacidade mais acima ou mais abaixo na hierarquia podem ser escolhidos, a depender da estratégia de avaliação utilizada. Por fim, o processo seleciona uma configuração do nível de capacidade previamente selecio-

nado. A ordem de seleção das configurações também é irrelevante, uma vez que todas as configurações daquele nível de capacidade devem ser avaliadas.

### 2.3.2. Execução da aplicação

Uma vez escolhidos uma carga de trabalho, uma categoria, um nível de capacidade e uma configuração, o processo está apto a executar a aplicação na nuvem. A execução da aplicação também é uma atividade abstrata do processo, pois depende de uma série de fatores que são específicos de cada aplicação ou plataforma de nuvem, como as tecnologias necessárias parar implantar os componentes da aplicação na nuvem bem como para submetê-los aos níveis de carga de trabalho desejados. Após a execução da aplicação, o processo analisa o resultado obtido e passa para a fase de inferência de desempenho.

### 2.3.3. Inferência de desempenho

Nesta fase, o processo se bifurca, atingindo seu primeiro ponto de decisão. A partir da análise do resultado da execução, que é feita comparando-se os indicadores obtidos para a métrica de desempenho utilizada frente ao valor de referência (SLA) desejado, o processo determina se a aplicação é ou não capaz de atender à demanda imposta sobre ela com a atual configuração. Se a aplicação satisfaz o SLA, o processo assinala a configuração atual como uma *configuração candidata* para o atual nível de carga. Do contrário, o processo assinala a configuração atual como uma *configuração rejeitada* para esse nível de carga.

É neste momento que a abordagem de inferência de desempenho, proposta originalmente neste trabalho, entra em ação. Com base nas relações de capacidade presentes no espaço de implantação, o processo pode "inferir" o provável desempenho da aplicação para outras configurações e cargas de trabalho ainda não avaliadas. Ora, se o processo identificou que uma certa configuração consegue satisfazer a demanda imposta à aplicação sob uma certa carga de trabalho, intuitivamente qualquer outra configuração de maior poder computacional também será capaz de fazê-lo sob a mesma carga de trabalho. Similarmente, é intuitivo concluir que a mesma configuração também será capaz de satisfazer o SLA da aplicação sob cargas de trabalho menores. Assim, usando as informações sobre as relações de capacidade existentes entre as configurações do espaço de implantação, o processo também assinala como candidatas para o atual nível de carga todas as outras configurações identificadas como sendo de "maior capacidade" que a configuração atual de acordo com o espaço de implantação. Da mesma forma, o processo também assinala a configuração atual como candidata para todos os níveis de carga inferiores ao nível de carga atual.

O caso em que a configuração atual não satisfaz o SLA da aplicação é tratado de modo análogo. Nesse caso, o processo assinala como rejeitadas para o atual nível de carga todas as outras configurações identificadas como sendo de "menor capacidade" que a configuração atual de acordo com o espaço de implantação. O mesmo acontece com a configuração atual, que também é assinalada como rejeitada para todos os outros níveis de carga superiores ao nível de carga atual.

### 2.3.4. Seleção do próximo cenário

Após a fase de inferência de desempenho, o processo seleciona os elementos que comporão o próximo cenário de execução a ser avaliado, ou encerra sua execução, caso não haja mais cenários a explorar. Nesse caso, o processo produz, como saída, uma lista contendo todas as configurações assinaladas como candidatas para cada carga de trabalho avaliada, em ordem crescente de preço.

A seleção do próximo cenário inclui a escolha de uma nova configuração do atual nível de capacidade, a escolha de um novo nível de capacidade (que deverá ser maior ou menor que o nível de capacidade atual, a depender do resultado da execução da aplicação no atual cenário), a escolha de uma nova categoria, ou a escolha de uma nova carga de trabalho (que também deverá ser maior ou menor que o nível de carga atual, novamente a depender do resultado da execução da aplicação no atual cenário). As escolhas do novo nível de capacidade e da nova carga de trabalho também são atividades abstratas, a serem definidas de acordo com a estratégia de avaliação utilizada para customizar o processo.

## 2.4. Estratégias de Avaliação

Conforme mencionado anteriormente, todas as atividades abstratas do processo (com exceção da atividade de execução da aplicação na nuvem) devem ser customizadas de acordo com diferentes estratégias de avaliação. Essas atividades incluem, basicamente, a escolha de cargas de trabalho e níveis de capacidade. Tais escolhas influenciam diretamente a maneira através da qual o processo explora o espaço de implantação, tendo um forte impacto no alcance da inferência de desempenho.

Como exemplo, considere o caso de um espaço de implantação onde nenhuma configuração é capaz de atender a demanda da aplicação sob qualquer nível de carga. Nesse caso, iniciar o processo de avaliação pelas configurações do nível de capacidade mais baixo sob cargas de trabalho maiores não seria uma boa estratégia, uma vez que o número de configurações e cargas de trabalho para os quais o desempenho esperado da aplicação poderia ser inferido seria muito pequeno. Por outro lado, iniciar o processo pelas configurações de nível de capacidade mais alto sob cargas de trabalho menores seria um estratégia muito melhor, já que assim seria possível inferir o desempenho da aplicação para praticamente todas as outras configurações e todas as outras cargas de trabalho, representando uma grande economia de tempo e custo.

Esses dois extremos ilustram bem o desafio de se escolher os cenários de execução mais promissores do ponto de vista da inferência de desempenho. A fim de enfrentar esse desafio, este trabalho introduz o conceito das *heurísticas de seleção*, que agregam táticas a serem observadas no momento em que o processo, via alguma estratégia de avaliação, precisa escolher uma nova configuração ou uma nova carga de trabalho para compor um novo cenário de execução. Nesse sentido, foi inicialmente definido um conjunto de três táticas de seleção, denominadas *otimista*, *conservadora* e *pessimista*, respectivamente, aplicáveis tanto à escolha de novas cargas de trabalho quanto à escolha de novos níveis de capacidade. A combinação dessas três táticas na escolha de novos cenários de execução dá origem a nove heurísticas de seleção, ilustradas na Figura 2.

Na figura, as heurísticas são identificadas por diferentes pares de letras posicionados ao longo da matriz que representa o espaço de implantação. A primeira letra que

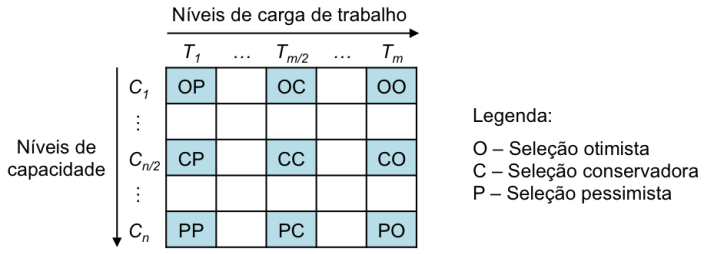



**Figura 2. Heurísticas para seleção de configurações e cargas de trabalho.**

identifica a heurística refere-se à tática usada na escolha da configuração (linha), enquanto a segunda letra refere-se à tática usada na escolha da carga de trabalho (coluna). Como pode-se observar, a tática otimista leva à escolha de configurações menores e cargas de trabalho maiores. Já a tática conservadora leva à escolha de configurações e cargas de trabalho de nível intermediário. Por fim, a tática pessimista leva à escolha de configurações maiores e cargas de trabalho menores.

No contexto do processo de avaliação de capacidade proposto neste trabalho, cada heurística de seleção fornece uma "lógica" diferente para exploração do espaço de implantação, servindo como base para a customização do processo com diferentes estratégias de avaliação. A acurácia e a eficiência do processo proposto, em particular, da abordagem de inferência de desempenho, utilizando cada uma das noves heurísticas de seleção mencionas acima, serão analisadas na próxima seção.

## 3. Avaliação Experimental

Esta seção descreve o experimento realizado como forma de verificação do processo de avaliação de capacidade apresentado anteriormente. Inicialmente, é apresentada a metodologia utilizada para a condução do experimento. Em seguida, são apresentados os resultados obtidos por cada uma das nove heurísticas de seleção propostas. Esses resultados são usados tanto para uma comparação qualitativa das heurísticas entre si, quanto para atestar a eficiência do processo proposto e de sua abordagem de inferência de desempenho.

É importante mencionar que o processo proposto foi implementado e está disponível na forma de uma ferramenta web,[1] a qual foi utilizada para executar o experimento descrito a seguir. Devido a restrições de espaço, os detalhes da implementação do processo bem como de sua ferramenta de apoio estão fora do escopo deste artigo.

### 3.1. Metodologia

O experimento consistiu na realização de sessões de avaliação de capacidade de uma aplicação web real (WordPress [WordPress 2014], escolhida por ser uma dais aplicações de criação e administração de *blogs* mais utilizadas atualmente) implantada em um provedor de nuvem também real (Amazon EC2 [Amazon 2014], escolhido por ser o líder de mercado entre provedores IaaS públicos ). O WordPress foi implantado em duas camadas: uma para o banco de dados MySQL, e outra para o servidor de aplicação, executada pelo servidor Apache HTTPD. Como balanceador de carga, foi utilizada uma máquina dedicada executando o servidor web Nginx.

[1] `http://cloud-capacitor.herokuapp.com/`.

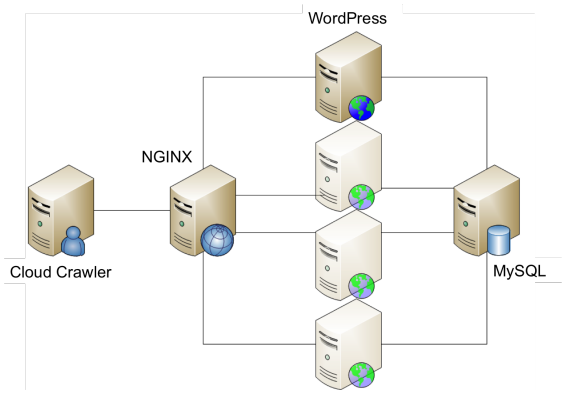



**Figura 3. Arquitetura de implantação do WordPress na Amazon EC2.**

Devido a restrições de custo e tempo, o experimento limitou-se a variar apenas a camada de aplicação, usando de 1 a 4 servidores Apache executando o WordPress. A execução dos testes foi orquestrada pelo ambiente Cloud Crawler [Cunha et al. 2013b, Cunha et al. 2013a], que automatizou as tarefas de iniciar e parar todas as instâncias de máquinas virtuais, configurar o balanceador de carga de acordo com o número de instâncias testadas na camada de aplicação, iniciar e parar a execução dos testes, gerar as cargas de trabalho impostas à aplicação e, finalmente, coletar os dados de desempenho obtidos em cada teste. A Figura 3 mostra um panorama dessa arquitetura de implantação.

Para compor o espaço de implantação utilizado no experimento, foram escolhidos sete tipos de máquinas virtuais oferecidos pelo provedor Amazon EC2: *m3_medium*, *m3_large*, *m3_xlarge*, *m3_2xlarge*, *c3_large*, *c3_xlarge* e *c3_2xlarge*. Para cada um desses tipos, foram criadas configurações com 1, 2, 3 e 4 instâncias, levando a um total de 28 configurações diferentes no espaço de implantação, divididas em duas categorias distintas, "m3" e "c3". A relação de capacidade entre essas configurações foram definidas separadamente para cada categoria, de modo a refletir o tipo e a quantidade de máquinas virtuais presentes em cada configuração. Assim, configurações com um certo número de máquinas virtuais de um determinado tipo eram consideradas de capacidade superior (inferior) a outras configurações contendo máquinas do mesmo tipo em menor (maior) quantidade. De maneira similar, configurações contendo o um certo número de máquinas virtuais de um certo tipo eram consideradas de capacidade superior (inferior) a outras configurações com a mesma quantidade de máquinas mas de tipos diferentes se estes tipos fossem inferiores (superiores) ao tipo da primeira configuração, de acordo com a classificação dos tipos definidas pelo próprio provedor de nuvem. Por exemplo, uma configuração composta por 3 máquinas to tipo *m3_2xlarge* era considerada superior a outra configuração composta por apenas 2 máquinas deste mesmo tipo. Da mesma forma, uma configuração composta por 2 máquinas do tipo *c3_large* era considerada inferior a outra configuração com a mesma quantidade de máquinas do tipo *c3_xlarge*.

As cargas de trabalho utilizadas no experimento foram quantificadas em número de usuários concorrentes enviando requisições ao WordPress. Foram definidas um total de 10 cargas de trabalho, representando 100, 200, 300, 400, 500, 600, 700, 800, 900 e 1000 usuários concorrentes, respectivamente.

De forma a estabelecer uma *baseline* para comparação da eficiência e da acurácia

do processo proposto, especificamente de suas diferentes heurísticas de seleção, foram coletados dados de desempenho do WordPress na nuvem para cada um dos 280 cenários possíveis, ou seja, foram efetivamente realizados testes de desempenho da aplicação para cada uma das 28 configurações criadas sob cada uma das 10 cargas de trabalho especificadas. A esse conjunto de dados de execuções reais da aplicação foi dado o nome "oráculo" e à estratégia necessária para gerar esses todos esses dados foi dado o nome de heurística "força bruta" (*Brute Force – BF*). As nove heurísticas propostas foram comparadas entre si e com a heurística BF.

Cada teste de desempenho consistiu em executar o WordPress utilizando uma das 28 configurações definidas para o espaço de implantação e então submetê-lo a uma das 10 cargas de trabalho especificadas durante um período de 1 hora. Durante os testes, um gerador de carga criava a quantidade de usuários corresponde à carga de trabalho sendo avaliada. Cada usuário realizava a seguinte sequência de requisições à aplicação: efetuar *logon*; inserir uma nova postagem; consultar a nova postagem; alterar a nova postagem; consultar postagens existentes por palavra-chave; alterar uma postagem existente; efetuar *logoff*.

A métrica de desempenho utilizada no experimento foi o *tempo de resposta total*, ou seja, o tempo total decorrido entre o envio da primeira requisição da sequência acima e o momento em que o usuário recebeu a resposta para última requisição da sequência. Assim, para ser considerada como candidata para uma determinada carga de trabalho, uma configuração devia ser capaz de atender, sem erros, pelo menos 90% das sequências de requisições recebidas dos usuários da aplicação em um tempo total igual ou inferior ao valor do SLA, tal como definido no respectivo parâmetro de entrada do processo.

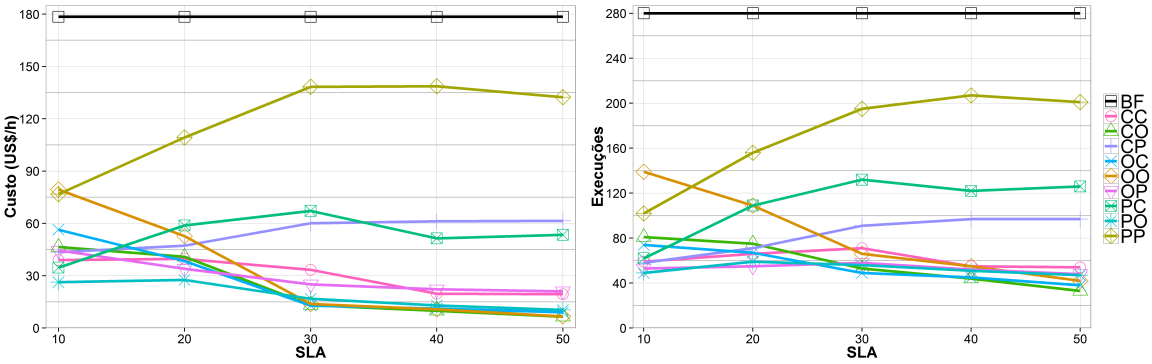
## 3.2. Resultados

### 3.2.1. Eficiência

Esta subseção apresenta os resultados de eficiência atingidos pelas heurísticas de seleção usadas no processo sob dois aspectos distintos: o custo total da avaliação e a quantidade de execuções realizadas por cada heurística. Esse custo foi calculado somando-se o preço da hora de utilização, conforme a tabela de preços do provedor na data realização dos testes, para cada uma das configurações para as quais foram realizadas execuções reais na nuvem.

A Figura 4 mostra os gráficos dos resultados obtidos pelas nove heurísticas em relação a essas duas métricas, considerando SLAs de 10, 20, 30, 40 e 50 segundos. No topo de cada gráfico vê-se uma linha horizontal escura que representa os resultados da heurística BF. Note que, como essa heurística não efetua nenhuma inferência quanto ao desempenho da aplicação, seus resultados, tanto em termos de custo quanto em termos de número de execuções, são sempre constantes, independente do SLA requerido.

A análise do gráfico de custo (Figura 4(a)) mostra que mesmo a heurística com o pior desempenho no que se refere ao custo já apresenta uma redução considerável em relação à heurística BF. Por outro lado, as melhores heurísticas chegam a representar uma economia da ordem de 96% em comparação com o que seria gasto com a execução de todas as combinações de configurações e cargas de trabalho. Embora o comportamento das heurísticas varie em função do SLA, é possível notar que quando a exigência do SLA




(a) (b)

**Figura 4. Eficiência das heurísticas de seleção: (a) custo e (b) execuções.**

é mais moderada, o comportamento de todas as heurísticas se estabiliza, tornando possível identificar que algumas delas tendem a ser mais econômicas que as outras. Ainda que não seja possível afirmar que uma só heurística seja a melhor em todas as situações, pode-se considerar que a heurística Pessimista/Otimista (PO) se mostra como a mais econômica em geral. A heurística Conservadora/Otimista (CO) merece atenção para os SLAs mais brandos, com os menores custos absolutos nessas circunstâncias.

A análise do gráfico de execuções reais na nuvem (Figura 4(b)), por sua vez, mostra uma redução de até 88% em relação aos resultados da heurística BF. Os menores números de execuções são atingidos pelas heurísticas Otimista/Conservadora (OC) e Conservadora/Otimista (CO), sob os SLAs mais brandos. Porém, como não se saem tão bem sob SLAs mais rígidos, como 10 segundos, a heurística PO ganha destaque por ter comportamento mais estável, figurando entre as mais econômicas no aspecto de quantidade de execuções sob a maioria dos SLAs avaliados.

Vale ressaltar que a significativa redução do número de execuções necessárias durante o processo de planejamento de capacidade, decorrente da utilização da abordagem de inferência de desempenho, conforme mostrado nesta seção, implica não apenas em economia de tempo e horas de máquinas para os usuários da nuvem, mas também pode contribuir de forma decisiva para a reduzir outros tipos de custo típicos de qualquer projeto, como esforço e alocação de recursos humanos .

### 3.2.2. Acurácia

Para medir a acurácia do processo de avaliação de capacidade, foram calculados os valores médios de *Precision*, *Recall* e *F-Measure* [Baeza-Yates and Ribeiro-Neto 1999] para os resultados produzidos por cada uma das heurísticas de seleção sob os diferentes valores de SLA avaliados, tomando como base os dados do oráculo. Para isso, os dados do oráculo foram utilizados para determinar se as configurações identificadas como candidatas (resultados positivos) e rejeitadas (resultados negativos) por cada heurística para uma determinada carga de trabalho eram de fato verdadeiras (nesse caso, as predições teriam sido corretas) ou falsas (nesse caso, as predições teriam sido erradas).

Os valores dessas três métricas para uma carga de trabalho $i$, denotados por $P_i$, $R_i$ e $F_i$, respectivamente, são dados pelas seguintes fórmulas:

**Tabela 2. Acurácia das heurísticas de seleção.**

| | SLA | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | | | 20 | | | 30 | | | 40 | | | 50 | | |
| Heurística | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| CC | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,98 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| CO | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,99 | 1,00 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| CP | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,98 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| OC | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,99 | 0,99 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| OO | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,99 | 1,00 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| OP | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,98 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| PC | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,98 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| PO | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,99 | 1,00 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| PP | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,98 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |

$$P_i = \frac{\textit{no. resultados positivos verdadeiros}}{\textit{no. resultados positivos verdadeiros} + \textit{no. resultados positivos falsos}}$$

$$R_i = \frac{\textit{no. resultados positivos verdadeiros}}{\textit{no. resultados positivos verdadeiros} + \textit{no. resultados negativos falsos}}$$

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i}$$

A Tabela 2 mostra os valores médios de $P_i$, $R_i$ e $F_i$, considerando todas as 10 cargas de trabalho, calculados para cada heurística de seleção sob os cinco níveis de SLA. Nota-se que em apenas um dos cinco SLAs o processo deixou de obter 100% de acurácia nas predições, apresentando uma taxa de erro inferior a 3% para os valores de *Precision* e *Recall*, e de aproximadamente 1% para os valores de *F-Measure*, que estabelece uma média ponderada entre as duas primeiras métricas [Baeza-Yates and Ribeiro-Neto 1999]).

Uma investigação mais minuciosa dos dados de desempenho da aplicação na nuvem revelou que essa pequena perda na qualidade das predições foi devida a flutuações ocasionais no desempenho de alguns dos tipos de máquinas virtuais disponibilizadas pelo provedor. Essas flutuações levaram algumas das configurações avaliadas a terem um desempenho superior ao de outras configurações consideradas de maior capacidade de acordo com o espaço de implantação. Tais flutuações afetaram particularmente o desempenho da aplicação para o SLA de 30 segundos, refletindo em erros de predição. De fato, oscilações no desempenho da infraestrutura virtualizada oferecida por provedores de nuvem IaaS são relativamente comuns, como observados por [Iosup et al. 2011] e [Cunha et al. 2011]. O impacto dessas flutuações nos resultados observados no experimento, porém, foi mínimo (afetando um único nível de SLA com taxa de erro médio de 1%), o que reforça a confiança de que a abordagem de inferência de desempenho proposta neste trabalho pode atingir alta acurácia mesmo para aplicações e plataformas de nuvem reais.

## 4. Trabalhos Relacionados

...

Para apoiar os usuários de nuvens IaaS no planejamento de capacidade, os trabalhos fazem uso de diferentes abordagens. Neste artigo serão expostos trabalhos que usam abordagens preditivas e empíricas. Na abordagem preditiva, a aplicação alvo não é executada diretamente no ambiente onde se deseja implantá-la, [CloudHarmony 2014], [Malkowski et al. 2010, Li et al. 2010, Jung et al. 2013, Fittkau et al. 2012] e [Li et al. 2011], fazem uso dessa abordagem. Já em [Jayasinghe et al. 2012, Silva et al. 2013, Cunha et al. 2013b]

e [Scheuner et al. 2014] são apresentados trabalhos que utilizam a abordagem empírica, onde as aplicações alvo são implantadas na nuvem e então submetidas a testes de carga.

As soluções de abordagem preditiva não requerem a alocação de recursos de nuvem para realizarem as predições, com exceção do *CloudProphet*, apresentado em [Li et al. 2011]. Além disso, são soluções de baixa complexidade, com destaque para a apresentada em [CloudHarmony 2014], a qual permite que os testes sejam iniciados e que as pesquisas de resultados anteriores sejam realizadas através de uma interface amigável, sem a necessidade de intervenções do usuário. Por outro lado, essas soluções apresentam limitações na definição da aplicação alvo e dos seus requisitos de desempenho, [Malkowski et al. 2010, CloudHarmony 2014], e os resultados das predições podem divergir dos valores reais. O que ficou evidenciado em [Fittkau et al. 2012], onde os valores da CPU simulada, em comparação com os valores da CPU medida, divergiram na ordem de 30 %.

Essas limitações, apresentadas pelas soluções de abordagem preditiva, são superadas pelas soluções de abordagem empírica que permitem ao usuário definir os componentes da aplicação e os seus resultados são reflexo da execução real da aplicação alvo. Como exemplo de uma solução de abordagem empírica, em [Cunha et al. 2013b] o usuário pode definir toda a pilha de componentes da aplicação alvo, todos cenários utilizados na avaliação de desempenho, as demandas que serão submetidas a cada um dos cenários e o critério que define se o cenário suportou a demanda, ou seja, o SLA. Da mesma forma, as soluções apresentadas em [Jayasinghe et al. 2012, Silva et al. 2013, Scheuner et al. 2014], possuem os seus mecanismos para a realização dessas definições. Porém, elas precisam executar cada um dos cenários definidos pelo usuário e não fazem uso de resultados anteriores para evitar a execução de testes que claramente poderiam ser evitados.

Nesse contexto, o novo processo apresentado neste trabalho segue uma abordagem híbrida, combinando aspectos positivos das abordagens preditiva e empírica. Por exemplo, em uma situação na qual uma demanda é submetida à aplicação que está sendo executada em uma máquina virtual com baixo poder computacional, o processo pode afirmar que essa mesma demanda pode ser atendida por máquinas com perfis computacionais mais robustos, e dessa forma diminuir o número de execuções e consequentemente os custos da avaliação.

## 5. Conclusão e Trabalhos futuros

A tarefa de escolher adequadamente os recursos computacionais (ex: máquinas virtuais) de forma a minimizar os custos dos clientes de nuvens IaaS ainda é um desafio importante que o atual estado da arte não consegue atender de forma satisfatória. Este trabalho apresentou um processo de inferência de desempenho que se mostrou uma solução eficaz (com acurácia medida em cerca de 99%) para planejar a capacidade necessária às aplicações que são implantadas na nuvem. Além disso, nossa solução mostrou-se também muito eficiente, apresentando redução de custo de até 96% e redução do tempo de execução de até 88%, em relação à execução total de todas as combinações de cargas de trabalho e configurações.

Com relação aos trabalhos futuros, há oportunidades de extensão deste trabalho na criação de novas heurísticas baseadas em dados de comprometimento de CPU e memória.

Também pretendemos investigar o impacto da flutuação de desempenho da nuvem sobre a acurácia do processo além de outros perfis de aplicações diferentes da arquitetura web do WordPress.

## Referências


Amazon (2014). Amazon Elastic Compute Cloud (EC2). `http://aws.amazon.com/ec2`.

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Beserra, P. V. et al. (2012). Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud. In *IEEE MESOCA 2012*.

CloudHarmony (2014). CloudHarmony: Benchmarking the Cloud. `http://cloudharmony.com/benchmarks`.

Cunha, M. et al. (2011). Investigating the impact of deployment configuration and user demand on a social network application in the Amazon EC2 cloud. In *IEEE CloudCom 2011*, pages 746–751.

Cunha, M. et al. (2013a). A Declarative Environment for Automatic Performance Evaluation in IaaS Clouds. In *IEEE CLOUD 2013*, pages 285–292.

Cunha, M. et al. (2013b). Cloud Crawler: Um Ambiente Programável para Avaliar o Desempenho de Aplicações em Nuvens de Infraestrutura. In *SBRC 2013*, pages 747–760.

Fittkau, F. et al. (2012). CDOSim: Simulating cloud deployment options for software migration support. In *IEEE MESOCA 2012*, pages 37–46.

Gonçalves Junior, R. et al. (2015). A Multi-Criteria Approach for Assessing Cloud Deployment Options Based on Non-Functional Requirements. In *ACM SAC 2015*. To appear.

Iosup, A. et al. (2011). On the performance variability of production cloud services. In *IEEE/ACM CCGrid 2011*, pages 104–113.

Jayasinghe, D. et al. (2011). Variations in performance and scalability when migrating n-tier applications to different clouds. In *IEEE CLOUD 2011*, pages 73–80.

Jayasinghe, D. et al. (2012). Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds. In *IEEE CLOUD 2012*, pages 73–80.

Jung, G. et al. (2013). CloudAdvisor: A Recommendation-as-a-Service Platform for Cloud Configuration and Pricing. In *IEEE SERVICES 2013*, pages 456–463.

Li, A. et al. (2010). CloudCmp: Comparing Public Cloud Providers. In *ACM SIGCOMM IMC 2010*, pages 1–14.

Li, A. et al. (2011). CloudProphet: Towards Application Performance Prediction in Cloud. In *ACM SIGCOMM 2011*, pages 426–427. Full paper available at `https://www.cs.duke.edu/~angl/papers/cloudprophet_tr.pdf`.

Malkowski, S. et al. (2010). CloudXplor: A tool for configuration planning in clouds based on empirical data. In *ACM SAC 2010*, pages 391–398.

Menascé, D. A. and Ngo, P. (2009). Understanding Cloud Computing: Experimentation and Capacity Planning. In *CMG 2009*.

Scheuner, J. et al. (2014). Cloud WorkBench – Infrastructure-as-Code Based Cloud Benchmarking. *arXiv preprint arXiv:1408.4565*.

Silva, M. et al. (2013). CloudBench: Experiment Automation for Cloud Environments. In *IEEE IC2E 2013*, pages 302–311.

WordPress (2014). Wordpress: Blog tool, publishing platform, and cms. `https://wordpress.org/`.